Oracle® Application Integration Architecture

Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management

Release 11.2

E26501-03

September 2012

Documentation that describes the Oracle Communications Order to Cash pre-built integration, which provides communications service providers (CSPs) deployment and integration accelerators that build on industry forward-looking methodology and best practices. The Oracle Communications Order to Cash automates BSS (Business Support Systems) concept to launch and BSS order to activate processes across Siebel Customer Relationship Management (Siebel CRM), Oracle Communications Order and Service Management (Oracle OSM), and Oracle Communications Billing and Revenue Management (Oracle BRM).

ORACLE®

Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management, Release 11.2

E26501-03

Contents

4 PLM - Synchronize Product and Price: Implementation

5 PLM - Understanding the Query Product Classes Business Flow

6 PLM - Query Product Classes: Implementation

7   Understanding the Process Integration for Order Lifecycle Management

8   OLM - Understanding the Process Sales Order Fulfillment Business Flow

9   OLM - Process Sales Order Fulfillment Business Flow: Implementation

10 OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow

11 OLM - Synchronize Fulfillment Order Billing Account Business Flow: Implementation

12 OLM - Understanding the Bill Fulfillment Order Business Flow

13 OLM - Bill Fulfillment Order Business Flow: Implementation

14 OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows

15 OLM - Provision Order and Update Fulfillment Order Business Flows: Implementation

16 OLM - Understanding the Update Sales Order Business Flow

17 OLM - Update Sales Order Business Flow: Implementation

22   OFM - Create Trouble Ticket from Oracle AIA Business Flow: Implementation

23   OFM - Create and Manage Trouble Ticket from Oracle OSM Business Flow: Implementation

Part II   Implementing the Delivered Integrations

24   Configuring the Process Integration for Product Lifecycle Management

25   Configuring the Process Integration for Order Lifecycle Management

26   Configuring the Process Integration for Customer Management

27   Configuring the Process Integration for Order Fallout Management

H Expectations from a COM System for Billing Integration

I Using the Oracle Mediator Resequencer Feature

J OLM - Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant

K Composite Application Validation System Changes

L Reintroducing Enterprise Business Services

List of Examples

List of Tables

List of Figures

Preface

Welcome to the Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management implementation guide.

Oracle AIA Communications Pre-Built Integrations

Oracle AIA Communications pre-built integrations 11.2 includes the following implementation guides:

- Oracle Communications Order to Cash for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management 11.2

- Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care 11.2

- Oracle Communications Billing and Revenue Management Integration Pack for Oracle Business Suite: Revenue Accounting 11.2

Oracle AIA for Communications release 11.2 includes these integrations to automate the Order to Cash process for the Communications industry.

- Oracle Communications Order to Cash Integration Pack for Siebel CRM

- Oracle Communications Order to Cash Integration Pack for Oracle Order and Service Management

- Oracle Communications Order to Cash Integration Pack for Oracle Billing and Revenue Management

For more information about Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM, see Chapter 1, "Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM Overview."

Common Oracle AIA Pre-Built Integration Guides

Oracle Application Integration Architecture Pre-Built Integrations 11.2 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 11.2.

  - This guide is restructured into a general installation chapter with an individual configuration and deployment chapter for each pre-built integration.

–   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 11.2: 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 11.2: 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.

Additional Resources

The following resources are also available:

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

Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.6.0 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: http://tinyurl.com/83q8grw

■   **Release Notes:**

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

- **Documentation updates:**

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

What's New in This Guide for Release 11.2

We have reviewed the entire set of integration flows for the Oracle Communications Order to Cash integration pack and as a result of the reviews, working with customers, and understanding the most common use cases and deployment scenarios, we have made a number of changes. The most significant of these changes is to optimize for the most common deployment scenarios. Oracle believes that these changes result in:

- Reduction in the number of artifacts required to complete the end-to-end integration flows.

- Reduction in the number of "hops" between the various participating applications and middleware components – simplifying operational management.

- Preserving the ability for more complex deployments when and where warranted.

Because of these optimization changes, for release 11.2, this guide has been updated in several ways, including the removal of enterprise business services (EBSs) from integration flows and also the way in which the Composite Application Validation System (CAVS) is enabled. (Previously you used the CAVS UI -- now you are asked to modify the AIA configuration properties file).

EBSs were introduced to simply help route to multiple provider connectors. However, the majority of our customers are using just one source and one target system for most of the integration flows. With this configuration, dynamically identifying a provider system during runtime (content-based routing) is never required and EBSs are therefore, unnecessary. The removal of EBSs, reduces one hop in each flow, reduces the number of artifacts, and in addition, maintenance and debugging becomes simpler. If required, EBSs can be reintroduced.

The following table lists the chapters and sections that have been added or changed.

Sections	Changes Made
Chapter 1 Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM	
Chapter 1, "Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM Overview"	The Oracle Communications Order to Cash integration pack combines the functionality and replaces the *Siebel CRM for Oracle Communications Billing and Revenue Management: Order to Bill* and the *Oracle Order to Activate for Siebel CRM and Oracle Communications Order and Service Management* integration packs, which were previously available as part of the AIA 2.5 release.
Section 1.1, "Oracle Communications Order to Cash Overview"	Section revised to document integration patterns applicable to the Oracle Communications Order to Cash pre-built integration.

Sections	Changes Made
Chapter 3 PLM - Understanding the Synchronize Product and Price Business Flow	
Chapter 3, "PLM - Understanding the Synchronize Product and Price Business Flow"	Chapter has been revised to document the new name for the Cartridge Guide. The new name is *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*
Section 3.2.1, "Synchronization of Billing Products with Pricing Details"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 3.2.2, "Synchronization of Billing Discounts"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Chapter 4 PLM - Synchronize Product and Price: Implementation	
Section 4.2, "Product Synchronization Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 4.3, "Billing Discount Synchronization Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 4.7, "Integration Services"	Section revised to update the functionality of the services delivered with the Synchronize Product and Price business flow.
Chapter 6 PLM - Query Product Classes: Implementation	
Chapter 6, "PLM - Query Product Classes: Implementation"	Chapter has been revised to document the new name for the Cartridge Guide. The new name is *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*
Section 6.2, "Query Product Classes - Implementation Overview"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 6.4, "Query Product Classes Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 6.7, "Integration Services"	Section revised to update the functionality of the services delivered with the Query Product Classes business flow.
Chapter 7 Understanding the Process Integration for Order Lifecycle Management	
Chapter 7, "Understanding the Process Integration for Order Lifecycle Management"	Chapter has been revised to document the new name for the Cartridge Guide. The new name is *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*
Chapter 8 OLM - Understanding the Process Sales Order Fulfillment Business Flow	
Chapter 8, "OLM - Understanding the Process Sales Order Fulfillment Business Flow"	Chapter has been revised to document the new name for the Cartridge Guide. The new name is *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*
Chapter 9 OLM - Process Sales Order Fulfillment Business Flow: Implementation	

Sections	Changes Made
Section 9.2, "Submitting Orders from Siebel CRM to Oracle OSM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 9.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Submitting Orders from Siebel CRM to Oracle OSM integration flow.
Section 9.3, "Supporting Order Priorities"	Section revised to describe new sequence of events for this flow based on removal of EBSs.
Section 9.6, "Integration Services"	Section revised to update the functionality of the services delivered with the Process Sales Order Fulfillment business flow.
Chapter 10 OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow	
Section 10.1, "Synchronize Fulfillment Order Billing Account Overview"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Chapter 11 OLM - Synchronize Fulfillment Order Billing Account Business Flow: Implementation	
Section 11.2, "Interfacing Orders to Create Customer Data in Oracle BRM"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 11.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Interfacing orders to Create Customer Data in Oracle BRM integration flow.
Section 11.5, "Integration Services"	Section revised to update the functionality of the services delivered with the Synchronize Fulfillment Order Billing Account business flow.
Chapter 13 OLM - Bill Fulfillment Order Business Flow: Implementation	
Section 13.2, "Interfacing Orders to Create Transaction Data in Oracle BRM"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 13.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Interfacing Orders to Create Transaction Data in Oracle BRM integration flow.
Section 13.5, "Integration Services"	Section revised to update the functionality of the services delivered with the Bill Fulfillment Order business flow.
Chapter 14 OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows	
Section 14.2, "Creating Provisioning Orders"	Section revised to describe new sequence of events for this flow based on removal of EBSs.
Chapter 15 OLM - Provision Order and Update Fulfillment Order Business Flows: Implementation	

Sections	Changes Made
Section 15.2, "Oracle OSM Fulfillment to Oracle OSM Provisioning Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 15.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Oracle OSM Fulfillment to Oracle OSM Provisioning integration flow.
Section 15.4, "Integration Services"	Section revised to update the functionality of the services delivered with the Provision Order and Update Fulfillment Order business flows.
Chapter 17 OLM - Update Sales Order Business Flow: Implementation	
Section 17.2, "Updating Statuses from Oracle OSM to Siebel CRM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 17.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Updating Statuses from Oracle OSM to Siebel CRM integration flow.
Section 17.5, "Integration Services"	Section revised to update the functionality of the services delivered with the Update Sales Order business flow.
Chapter 18 Understanding the Process Integration for Customer Management	
Section 18.4.1, "Create/Sync Account Integration Flow"	Section revised to update the Create/Sync Account overall flow diagram based on removal of EBSs.
Section 18.4.2, "Update Customer Account Integration Flow"	Section revised to update the Update Customer Account overall flow diagram based on removal of EBSs.
Chapter 19 CM - Synchronize Customer Account: Implementation	
Section 19.2, "Update Customer Accounts Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 19.6, "Integration Services"	Section revised to update the functionality of the services delivered with the Synchronize Customer Account business flow.
Chapter 20 CM - Synchronize Customer Special Rating Profile: Implementation	
Section 20.1, "Synchronize Friends and Family List Updates to Oracle BRM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 20.1.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Synchronize Friends and Family List Updates to Oracle BRM integration flow.
Section 20.5, "Integration Services"	Section revised to update the functionality of the services delivered with the Synchronize Customer Special Rating Profile business flow.
Chapter 22 OFM - Create Trouble Ticket by Oracle AIA Business Flow: Implementation	
Section 22.1, "Creating Trouble Tickets in Siebel CRM from Oracle AIA Overview"	Section revised to describe new sequence of events for this flow based on removal of EBSs.

Sections	Changes Made
Section 22.2, "Creating a Trouble Ticket in Siebel CRM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 22.2.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Creating a Trouble Ticket in Siebel CRM integration flow.
Section 22.5, "Integration Services"	Section revised to update the functionality of the services delivered with the Create Trouble Ticket by Oracle AIA business flow.
Section 22.6, "Business Flow Fallout-Enabled Services"	Section revised to update list of fallout-enabled services based on removal of EBSs.
Chapter 23 OFM - Create and Manage Trouble Ticket by Oracle OSM Business Flow: Implementation	
Section 23.3, "Creating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 23.3.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Creating a Trouble Ticket in Siebel CRM by Oracle OSM integration flow.
Section 23.4, "Updating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow"	Section revised to describe new sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section 23.4.1, "Defining Transaction Boundaries and Recovery Details"	Section added to document transactions involved, the database operations, and what actions to take in case of errors for the Updating a Trouble Ticket in Siebel CRM by Oracle OSM integration flow.
Section 23.7, "Integration Services"	Section revised to update the functionality of the services delivered with the Create and Manage Trouble Ticket by Oracle OSM business flow.
Section 23.8, "Business Flow Fallout-Enabled Services"	Section revised to update list of fallout-enabled services based on removal of EBSs.
Chapter 24 Configuring the Process Integration for Product Lifecycle Management	
Section 24.7.1, "Configuring Properties for the Product Lifecycle Management Feature"	Section has been revised to document configuration properties reorganization.
Section 24.7.2, "Configuring Properties for the Query Product Class Feature"	Section has been revised to document configuration properties reorganization.
Chapter 25 Configuring the Process Integration for Order Lifecycle Management	
Section 25.6, "Configuring the Process Integration for Order Lifecycle Management"	Section has been revised to document configuration properties reorganization.
Chapter 26 Configuring the Process Integration for Customer Management	
Section 26.6, "Configuring the Process Integration for Customer Management"	Section has been revised to document configuration properties reorganization.
Chapter 27 Configuring the Process Integration for Order Fallout Management	

Sections	Changes Made
Section 27.6, "Configuring Properties for Order Fallout Services"	Section has been revised to document configuration properties reorganization.
Appendix A PLM - Cross-References for the Product Integration for Product Management	
Section A.2, "Product Synchronization Flow"	Section revised to describe the sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Section A.3, "Discount Synchronization Flow"	Section revised to describe the sequence of events for this flow based on removal of EBSs. The flow diagram has also been updated.
Appendix C OLM - Mapping Billing Dates	
Section C.1, "How Dates are Set in Oracle BRM"	Table revised to document the change that Oracle AIA uses ADDT on service bundle line (instead of ADDT on charge line) for applying MACD charge. See line 9 of the table.
Appendix F: Configuring Multiple Oracle BRM Instances	
Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations"	Section Configuring Routing Rules for Agent Assisted Billing Care Pre-Built Integration removed based on removal of EBSs.
Appendix H Expectations from a COM System for Billing Integration	
Appendix H, "Expectations from a COM System for Billing Integration"	Table revised to document changes based on the removal of EBSs.
Appendix I Using the Oracle Mediator Resequencer Feature	
Section I.1, "Queues and Flows Enabled for Sequencing"	Section revised to document changes to the queues and flows enabled for sequencing.
Section I.2, "Resolving Errors in Flows with Resequencer"	Section revised to document changes about how errors are resolved in flows using Resequencer.
Appendix K Composite Application Validation System Changes	
Appendix K, "Composite Application Validation System Changes"	Appendix created to describe how CAVS is enabled. (Previously you used the CAVS UI -- now you are asked to modify the AIA configuration properties file).
Appendix L: Reintroducing Enterprise Business Services	
Appendix L, "Reintroducing Enterprise Business Services"	Appendix created to describe how customers can reintroduce EBSs if they need content-based routing.

1

Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM Overview

In today's highly competitive communications industry, the rapid convergence of traditional and IP services, wireless and wire line services, IT and network, and prepaid and postpaid services all present challenges for communications service providers (CSPs) to rapidly launch new and bundled services, and then automate streamlined processes across front-office and back-office applications and networks.

The Oracle Communications Order to Cash pre-built integration provides CSPs integration and deployment accelerators that build on industry best practices and a comprehensive integration methodology. The Oracle Communications Order to Cash automates BSS (Business Support Systems) concept to launch and BSS order to activate processes across Siebel Customer Relationship Management (Siebel CRM), Oracle Communications Order and Service Management (Oracle OSM), and Oracle Communications Billing and Revenue Management (Oracle BRM).

The Oracle Communications Order to Cash pre-built integration consists of these options:

- Oracle Communications Order to Cash - Siebel CRM option
- Oracle Communications Order to Cash - Oracle OSM option
- Oracle Communications Order to Cash - Oracle BRM option

The three pre-built integration options are architected to provide an accelerated integration when using all of the referenced Oracle applications; however, based on the overall design they are loosely coupled enough to allow for third party applications to be leveraged. If you wish to leverage a third party application within the context of the Communications Order to Cash solution, you are required to construct specific application business connector services which conform to the design specified and which allow the third party applications to fulfill the roles and responsibilities specified by the design.

The three pre-built integration options are packaged separately to allow providers to license the options relevant to their deployment.

> **Caution:** This guide provides an overview of the design and implementation instructions for the process integrations available for all three Oracle Communication Order to Cash pre-built integration options. However, if your facility has only one or two of the three options, your own matching systems for the missing applications or connectors must mimic what is outlined in this guide to achieve the same functionality that is delivered with all three options.

1.1 Oracle Communications Order to Cash Overview

Functionally, the Oracle Communications Order to Cash provides the following process integrations:

- Product Lifecycle Management
- Order Lifecycle Management
- Customer Management
- Order Fallout Management

Product Lifecycle Management

The process integration for product lifecycle management enables you to:

- Synchronize and administer products and discounts between Oracle BRM and Siebel CRM.
- Query or import new or updated product classes from Siebel CRM into SCE Studio.

Product classes and transaction attributes are defined within Siebel CRM. SCE queries the product classes from Siebel CRM and maintains the mapping between the Siebel product class and the product specification.

This process integration is used in deployments where Oracle Product Hub is not used. A mutually exclusive alternative is provided with the Product MDM process integration where Oracle Product Hub is included in the deployment topology.

For more information about the Product MDM process integration process, see the *Oracle Application Integration Architecture Oracle Product Master Data Management Integration Implementation Guide*.

Order Lifecycle Management

The process integration for order lifecycle management enables the submission of orders from Siebel CRM to Oracle OSM for order fulfillment. Additionally, Oracle OSM can call the services provided by this integration to:

- Synchronize Fulfillment Order Billing Account(s) to Oracle BRM - Interface orders to create customer data in Oracle BRM.
- Bill Fulfillment Order in Oracle BRM - Interface orders to create transaction data in Oracle BRM.
- Synchronize Provisioning Order from Oracle OSM Central Order Management (COM) to Oracle OSM Service Order Management (SOM) - Downstream provisioning orchestration.
- Update Fulfillment Order from Oracle OSM (SOM) to Oracle OSM (COM) - Provisioning status updates.

- Update Sales Order in Siebel CRM - Update Siebel CRM with the status and other information on the order.

Customer Management

The process integration for customer management enables the synchronization of customer information from Siebel CRM to Oracle BRM. Customer accounts are defined in Siebel CRM and then created in Oracle BRM as part of the order fulfillment process. Once the account is created in Oracle BRM, the process integration ensures that any changes to the account in Siebel CRM are synchronized to Oracle BRM.

Order Fallout Management

The process integration for order fallout management enables you to implement a detection and notification process to handle order failures. Order fallout management uses Siebel trouble ticketing for notification and tracking of order failures.

Figure 1–1 illustrates how these three pre-built integration options work with participating applications to enable these business flows.

Figure 1–1 Oracle Communications Order to Cash Functional Overview
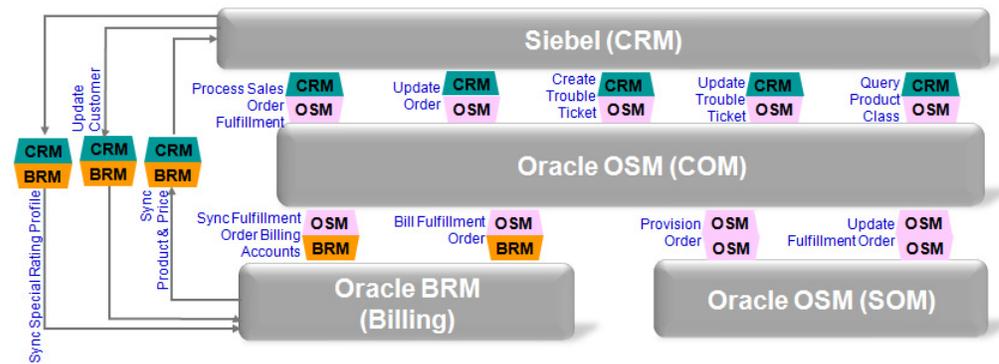
Oracle AIA is based on a service-oriented architecture (SOA) that has a pattern where a request comes from an application, which is translated into an enterprise business message (EBM) payload. The chart below is representative of the integration patterns applicable to the Oracle Communications Order to Cash pre-built integration. Siebel, Oracle BRM, and Oracle OSM participate as providers or requesters in different order to cash processes. Each of the Oracle Communications Order to Cash pre-built integration options package the integration artifacts falling between the subject application and Oracle AIA up to and including the Oracle AIA application business connector service (ABCS) for the respective application.

Figure 1–2   Integration Patterns Applicable to Oracle Comms Order to Cash
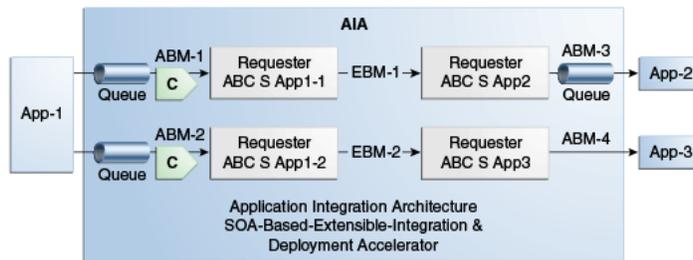
Table 1–1 illustrates the Oracle Communications Order to Cash pre-built integration options required to enable each process integration and business flow combination.

Table 1–1   Process Integration and Business Flow Combinations

Process Integration	Business Flow	Pre-Built Integration Options Enabling Flow
Product Lifecycle Management	Synchronize Product and Price For more information, see Chapter 3, "PLM - Understanding the Synchronize Product and Price Business Flow."	Siebel CRM and Oracle BRM options
	Query Product Class For more information, see Chapter 5, "PLM - Understanding the Query Product Classes Business Flow."	Siebel CRM and Oracle OSM options
Order Management	Process Sales Order Fulfillment For more information, see Chapter 8, "OLM - Understanding the Process Sales Order Fulfillment Business Flow."	Siebel CRM and Oracle OSM options
	Update Sales Order For more information, see Chapter 16, "OLM - Understanding the Update Sales Order Business Flow."	Siebel CRM and Oracle OSM options
	Synchronize Fulfillment Order Billing Account For more information, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."	Siebel CRM, Oracle OSM, and Oracle BRM options
	Bill Fulfillment Order For more information, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow."	Siebel CRM, Oracle OSM and Oracle BRM options
	Provision Order For more information, see Chapter 14, "OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows."	Oracle OSM option
	Update Fulfillment Order For more information, see Chapter 14, "OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows."	Oracle OSM option

Table 1–1   (Cont.) Process Integration and Business Flow Combinations

Process Integration	Business Flow	Pre-Built Integration Options Enabling Flow
Customer Management	Synchronize Customer Account For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."	Siebel CRM and Oracle BRM options
	Synchronize Customer Special Rating Profile For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."	Siebel CRM and Oracle BRM options
Order Fallout Management	Create and Update Trouble Ticket from Oracle OSM For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."	Siebel CRM and Oracle OSM options
	Create Trouble Ticket from Oracle AIA For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."	Siebel CRM and Oracle BRM options

In addition to the individual pre-built integration options, the deployment of multiples of them has also been packaged in out-of-the-box (OOTB) deployment options:

- Oracle Communications Order to Cash Siebel CRM, Oracle OSM, and Oracle BRM pre-built integration options

- Oracle Communications Order to Cash Siebel CRM and Oracle OSM pre-built integration options - this assumes integration with a billing system (other than Oracle BRM)

- Oracle Communications Order to Cash Siebel CRM and Oracle BRM pre-built integration options - this assumes integration with a central order management system (other than Oracle OSM) to manage order decomposition and fulfillment.

> **Note:**   If you have deployed Oracle Communications Order to Cash Siebel CRM and Oracle BRM options, *but not the Oracle OSM option*, a Test Orchestration Process (TOP) is shipped to sanity test the out-of-the-box (OOTB) order flow. This must be replaced by your own order management system.
>
> For more information, see *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*, "Configuring and Deploying the Comms Order to Cash: SBL CRM and BRM Pre-Built Integration," Replacing Test Order Orchestration with your Order Management System.

Table 1–2 illustrates the process integrations and business flows that are enabled by the Oracle Communications Order to Cash - Siebel CRM pre-built integration option:

Table 1–2   Process Integrations and Business Flows Enabled by Oracle Comms Order to Cash: Siebel CRM Option

Enabled Process Integration	Enabled Business Flow and Extent
Product Lifecycle Management	Synchronize Product and Price from Oracle AIA to Siebel; when the Oracle BRM option is in use it provides for Oracle BRM to make the sync request to Oracle AIA.
	For more information, see Chapter 3, "PLM - Understanding the Synchronize Product and Price Business Flow."
	Query Product Class. From Oracle AIA to Siebel and response goes back from Siebel to Oracle AIA; when the Oracle OSM option is in use it provides for SCE Studio to make the query request to Oracle AIA and for Oracle AIA to send the query results to SCE Studio.
	For more information, see Chapter 5, "PLM - Understanding the Query Product Classes Business Flow."
Order Management	Process Sales Order Fulfillment from Siebel to Oracle AIA; when the Oracle OSM option is in use it provides for Oracle AIA to route the request to Oracle OSM COM.
	For more information, see Chapter 8, "OLM - Understanding the Process Sales Order Fulfillment Business Flow."
	Update Sales Order from Oracle AIA to Siebel; when the Oracle OSM option is in use it provides for Oracle OSM COM to make the update request to Oracle AIA.
	For more information, see Chapter 16, "OLM - Understanding the Update Sales Order Business Flow."
Customer Management	Query Customer Account from Oracle AIA to Siebel; enables the Oracle AIA process to query customer accounts from Siebel CRM for new accounts.
	When the Oracle BRM option is in use, it is called as part of Synchronize Fulfillment Order Billing Account request is invoked by an order management system, as part of an order fulfillment flow.
	For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."
	Synchronize Customer Account from Siebel to Oracle AIA; Propagates customer account updates to Oracle AIA; when the Oracle BRM option is in use it provides for taking the account update request from Oracle AIA to Oracle BRM.
	For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."
	Synchronize Customer Special Rating Profile from Siebel to Oracle AIA; when the Oracle BRM option is in use it provides for taking the special rating profile synchronization from Oracle AIA to Oracle BRM.
	For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."

Table 1–2   (Cont.) Process Integrations and Business Flows Enabled by Oracle Comms Order to Cash: Siebel CRM Option

Enabled Process Integration	Enabled Business Flow and Extent
Order Fallout Management	Create and Update Trouble Tickets from Oracle AIA to Siebel; when the Oracle OSM option is in use it provides for Oracle OSM COM to make the create and update trouble ticket requests to Oracle AIA.
	For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."
	Create Trouble Tickets from Oracle AIA to Siebel CRM; when Oracle OSM option is not in use but Oracle BRM option is in use, it provides for Oracle AIA to create trouble tickets in Siebel CRM.
	For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

Table 1–3 illustrates the process integrations and business flows that are enabled by the Oracle Communications Order to Cash - Oracle OSM pre-built integration option:

Table 1–3   Process Integrations and Business Flows Enabled by Oracle Comms Order to Cash: Oracle OSM Option

Enabled Process Integration	Enabled Business Flow and Extent
Product Lifecycle Management	Query Product Class from SCE Studio to Oracle AIA and bring response back from Oracle AIA to Oracle OSM.
	When the Siebel CRM option is in use it provides for Oracle AIA to pass the query request to Siebel CRM and take the response back from Siebel CRM.
	For more information, see Chapter 5, "PLM - Understanding the Query Product Classes Business Flow."

Table 1–3   (Cont.) Process Integrations and Business Flows Enabled by Oracle Comms Order to Cash: Oracle OSM Option

Enabled Process Integration	Enabled Business Flow and Extent
Order Management	Process Sales Order Fulfillment from Oracle AIA to Oracle OSM COM; when the Siebel CRM option is in use it provides for sending the request from Siebel CRM to Oracle AIA.
	For more information, see Chapter 8, "OLM - Understanding the Process Sales Order Fulfillment Business Flow."
	Update Sales Order from Oracle OSM COM to Oracle AIA; when the Siebel CRM option is in use it provides for Oracle AIA to send the update to Siebel CRM.
	For more information, see Chapter 16, "OLM - Understanding the Update Sales Order Business Flow."
	Synchronize Fulfillment Order Billing Account from Oracle OSM to Oracle AIA; when the Siebel CRM option and Oracle BRM option are in use they provide for Oracle AIA to enrich customer account details by querying from Siebel CRM and synchronizing the accounts to Oracle BRM.
	For more information, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."
	Bill Fulfillment Order from Oracle OSM to Oracle AIA (enables both Oracle OSM Order to Cash functions: Initiate Billing and Fulfill Billing); when the Oracle BRM option is in use it provides for Oracle AIA interfacing the order to Oracle BRM.
	For more information, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow."
	Provision Order from OSM COM to Oracle AIA and from Oracle AIA to Oracle OSM Service Order Management (SOM).
	For more information, see Chapter 14, "OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows."
	Update Fulfillment Order from Oracle OSM SOM to Oracle AIA and from Oracle AIA to Oracle OSM COM.
	For more information, see Chapter 14, "OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows."
Order Fallout Management	Create and Update Trouble Tickets from Oracle OSM to Oracle AIA; when the Siebel CRM option is in use, it provides for Oracle AIA to take the request to Siebel CRM.
	For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

Table 1–4 illustrates the process integrations and business flows that are enabled by the Oracle Communications Order to Cash - Oracle BRM pre-built integration option:

Table 1–4   Process Integrations and Business Flows Enabled by Oracle Comms Order to Cash: Oracle BRM Option

Enabled Process Integration	Enabled Business Flow and Extent
Product Lifecycle Management	Synchronize Product and Price from Oracle BRM to Oracle AIA; when the Siebel CRM option is in use it provides for Oracle AIA to pass the sync request to Siebel CRM.
	For more information, see Chapter 3, "PLM - Understanding the Synchronize Product and Price Business Flow."
Order Management	Synchronize Fulfillment Order Billing Account from Oracle AIA to Oracle BRM.
	This uses the Customer Management Synchronize Customer Account process integration to create accounts in Oracle BRM. When the Oracle OSM option is in use, it provides for Oracle OSM to send the request to Oracle AIA. When the Siebel CRM option is in use, it provides for Oracle AIA to enrich the customer account details by querying into Siebel CRM.
	For more information, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."
	Bill Fulfillment Order from Oracle AIA to Oracle BRM; when the Oracle OSM option is in use, it provides for Oracle OSM to send the request to Oracle AIA.
	For more information, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow."
Customer Management	Synchronize Customer Account from Oracle AIA to Oracle BRM.
	This is called from the order management Synchronize Fulfillment Order Billing Account integration to create accounts in Oracle BRM.
	It is also used to propagate accounts updates to accounts interfaced to Oracle BRM. When the Siebel CRM option is in use, it provides for Siebel CRM to send the update request to Oracle AIA.
	For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."
	Synchronize Customer Special Rating Profile from Oracle AIA to Oracle BRM; when the Siebel CRM option is in use, it provides for Siebel CRM to send the request to Oracle AIA. This is used to propagate Special Rating Profile updates to profiles interfaced to Oracle BRM (as part of Bill Fulfillment Order).
	For more information, see Chapter 18, "Understanding the Process Integration for Customer Management."
Order Fallout Management	Create Trouble Tickets from AIA error handling to Oracle AIA; when the Siebel CRM option is in use, it provides for Oracle AIA passing the request to Siebel CRM.
	For more information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

The Oracle Communications Order to Cash pre-built integration is built on top of the Oracle AIA Communications Foundation Pack. Communications customers can easily extend the delivered process integrations and build new ones by leveraging the pre-defined enterprise business objects contained within the Oracle AIA Communications Foundation Pack, which are specifically tailored for the Communications industry.

Part I

Understanding the Delivered Integrations

Part I contains the following chapters:

- Chapter 22, "OFM - Create Trouble Ticket from Oracle AIA Business Flow: Implementation"
- Chapter 23, "OFM - Create and Manage Trouble Ticket from Oracle OSM Business Flow: Implementation"

2

Understanding the Process Integration for Product Lifecycle Management

This chapter provides an overview of the Product Lifecycle Management (PLM) integration process.

This chapter includes the following section:

- Section 2.1, "Product Lifecycle Management Overview"

2.1 Product Lifecycle Management Overview

The process integration for product lifecycle management (PLM) enables you to synchronize billing products and billing discounts between Oracle Billing and Revenue Management (Oracle BRM) and Siebel Customer Relationship Management (Siebel CRM). You can perform synchronization in either real time or batch mode.

Oracle BRM is the master for billing products and billing discounts. As such, this flow assumes that billing products and billing discounts are created in Oracle BRM and enriched in Siebel CRM. Also, any further updates to those products and discounts are also made in Oracle BRM and synchronized to Siebel CRM for further enrichment.

The process integration for product lifecycle management delivers these integration flows, which enable the synchronize product and price and the query product classes business flows:

Synchronize Product and Price

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Billing and Revenue Management (Oracle BRM) pre-built integration options.

- **Synchronization of billing products**: Oracle BRM to Siebel CRM. The product synchronization integration flow enables you to create new products in Oracle BRM and synchronize those products to Siebel CRM. It also enables you to update existing products in Oracle BRM and then synchronize the updated products to Siebel CRM.

- **Synchronization of billing discounts**: Oracle BRM to Siebel CRM. The discount synchronization integration flow enables you to create new discounts in Oracle BRM and synchronize those discounts as products to Siebel CRM. It also enables you to update existing discounts in Oracle BRM and then synchronize the updated discounts to Siebel CRM.

For more information about the Synchronize Product and Price business flow, see Chapter 3, "PLM - Understanding the Synchronize Product and Price Business Flow."

Query Product Classes

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

- **Query Product Classes**: SCE Studio and Siebel CRM. The query product classes integration flow enables you to create new product classes in Siebel and query them from the SCE Studio. The query process includes product classes, associated attributes and valuesets. The new product classes and updates to existing product classes are queried from SCE Studio.

For more information about the Synchronize Product and Price business flow, see Chapter 5, "PLM - Understanding the Query Product Classes Business Flow."

3

PLM - Understanding the Synchronize Product and Price Business Flow

This chapter provides an overview of the synchronize product and price business flow and discusses simple and customizable products, the product bundling methodology, and solution assumptions and constraints.

This chapter includes the following sections:

- Section 3.1, "Synchronize Product and Price Business Flow Overview"
- Section 3.2, "Simple and Customizable Products"
- Section 3.3, "Understanding the Product Bundling Methodology"
- Section 3.4, "Solution Assumptions and Constraints"

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Billing and Revenue Management (Oracle BRM) pre-built integration options.

3.1 Synchronize Product and Price Business Flow Overview

This section includes the following topics:

- Section 3.1.1, "Real-Time Billing Product and Billing Discount Synchronization"
- Section 3.1.2, "Update Real-Time Billing Product and Billing Discount Synchronization"
- Section 3.1.3, "Batch Billing Product and Billing Discount Synchronization"
- Section 3.1.4, "Update Batch Billing Product and Billing Discount Synchronization"
- Section 3.2.1, "Synchronization of Billing Products with Pricing Details"
- Section 3.2.2, "Synchronization of Billing Discounts"

3.1.1 Real-Time Billing Product and Billing Discount Synchronization

In this flow, the Oracle BRM product administrator creates billing products and billing discounts in the Oracle BRM Pricing Center. After a new billing product or billing discount is created, the Oracle BRM product administrator can commit it to the Oracle BRM database. Alternatively, the Oracle BRM product administrator can create a set of billing products and billing discounts and save them in a file. After all of the billing products and billing discounts have been created in the file, the Oracle BRM product administrator commits them to the Oracle BRM database. This instantaneously synchronizes the new billing products or billing discounts to Siebel CRM. The Siebel

CRM product administrator uses these billing products to create service bundles or promotions. The Siebel CRM product administrator can also add charges, such as penalties, to the promotion. After the promotions are created, customers can purchase the promotions.

For more information, see Section 3.3, "Understanding the Product Bundling Methodology."

Figure 3–1 shows the business process flow for synchronization of real-time billing products and billing discounts.

Figure 3–1   Synchronization of Real-Time Billing Products and Billing Discounts
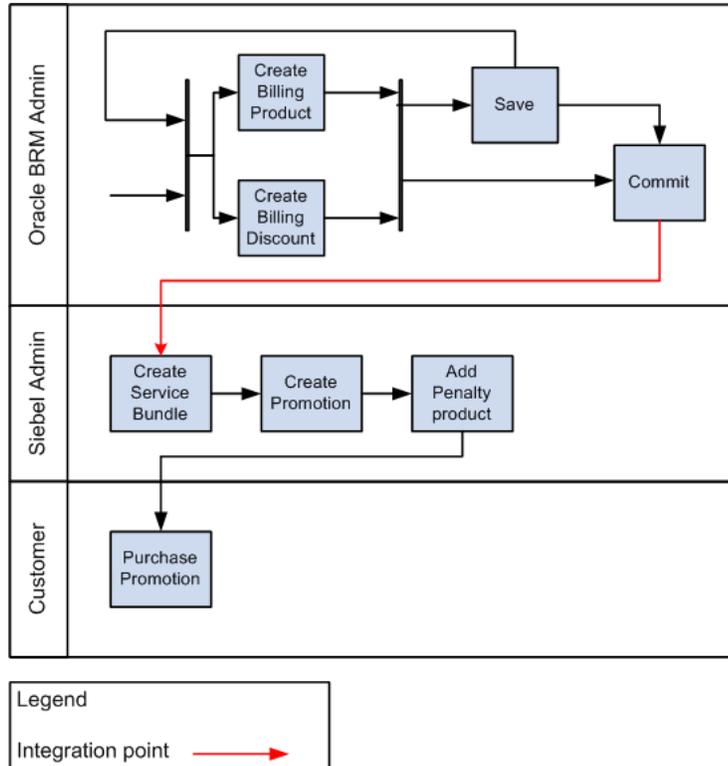
3.1.2 Update Real-Time Billing Product and Billing Discount Synchronization

In this flow, whenever changes occur to a billing product or a billing discount attribute, the Oracle BRM product administrator can update the billing products and billing discounts in the Oracle BRM Pricing Center and commit them to the Oracle BRM database. Alternatively, the Oracle BRM product administrator can update a set of billing products or billing discounts and save all of them in a file. After all of the billing products and billing discounts have been updated in the file, the Oracle BRM product administrator can commit them to the Oracle BRM database. This instantaneously synchronizes the updates to Siebel CRM. The service bundles and the promotions in Siebel CRM are updated to use the latest version of the billing products. The Siebel CRM product administrator makes any necessary changes in Siebel CRM if required. Customers who purchase the promotions receive the latest promotions.

Figure 3–2 shows the business process flow for synchronization of update real-time billing products and billing discounts.

Figure 3–2   Synchronization of Update Real-Time Billing Products and Billing Discounts
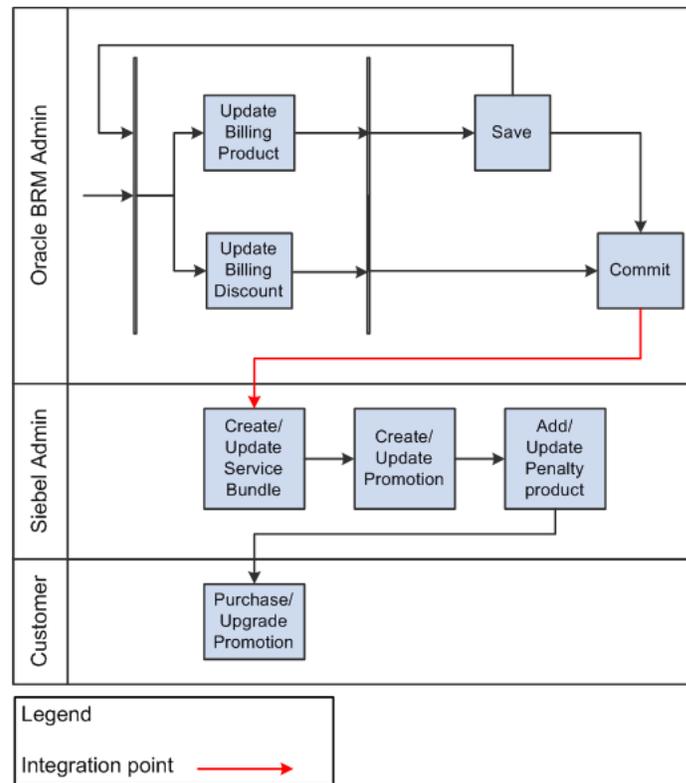
3.1.3 Batch Billing Product and Billing Discount Synchronization

In this flow, the Oracle BRM product administrator disables the event for real-time product synchronization, and then creates a set of billing products and billing discounts. The Oracle BRM product administrator runs a batch utility to store the products in the Oracle BRM database and synchronize the products with Siebel CRM. Alternatively, the Oracle BRM product administrator can create a set of products and save all of them in a file. After all of the billing products and billing discounts are created, the Oracle BRM product administrator runs the batch utility. The Siebel CRM product administrator uses these billing products and billing discounts to create service bundles and promotions. The Siebel CRM product administrator can also add charges, such as penalties, to the promotion. After promotions are created, customers can purchase the promotions.

For more information, see Section 3.3, "Understanding the Product Bundling Methodology."

Figure 3–3 shows the business process flow for synchronization of batch billing products and billing discounts.

Figure 3–3   Synchronization of Batch Billing Products and Billing Discounts
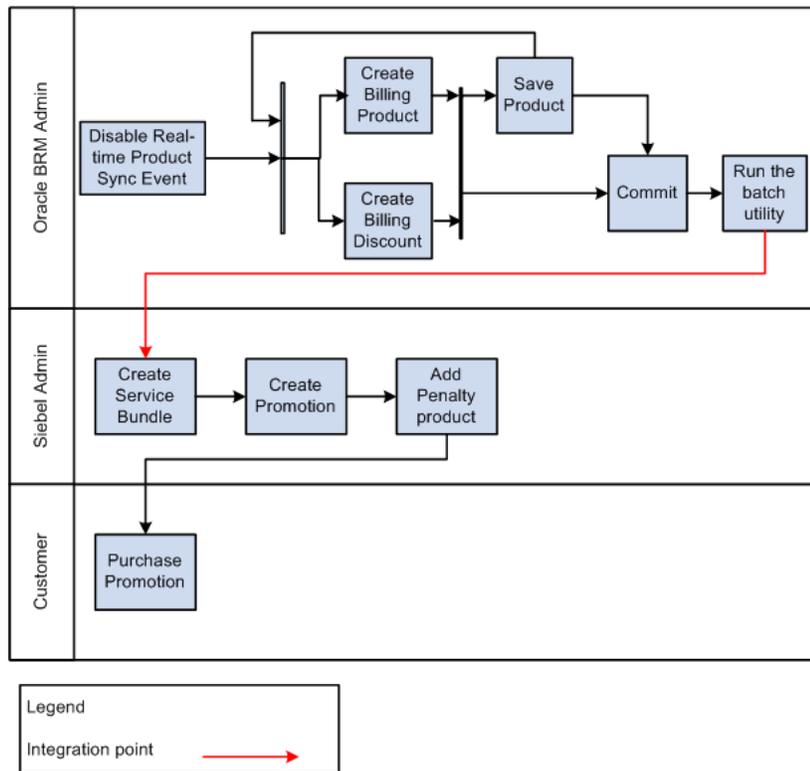
To disable the event for real-time product synchronization, see the Oracle BRM documentation.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "Service Integration Components," Synchronization Queue Data Manager, Installing and configuring the Synchronization Queue DM, Starting and stopping the Synchronization Queue DM.

3.1.4 Update Batch Billing Product and Billing Discount Synchronization

In this flow, the Oracle BRM product administrator disables the event for real-time product synchronization. Whenever changes are made to the products or discount attributes, the Oracle BRM product administrator updates billing products and billing discounts in the Oracle BRM Pricing Center. The Oracle BRM product administrator runs a batch utility to store the updates in the Oracle BRM database and synchronize them with Siebel CRM. Alternatively, the Oracle BRM product administrator can update a set of billing products and billing discounts and save all of them in a file. After all of the billing products and billing discounts are updated, the Oracle BRM product administrator runs the batch utility. The service bundles and the promotions in Siebel CRM are updated to use the latest version of the billing products and billing discounts. The Siebel CRM product administrator makes any necessary changes in Siebel if required. Customers who purchase the promotions receive the latest promotions.

Figure 3–4 shows the business process flow for synchronization of update batch billing products and billing discounts.

Figure 3–4   Synchronization of Update Batch Billing Products and Billing Discounts
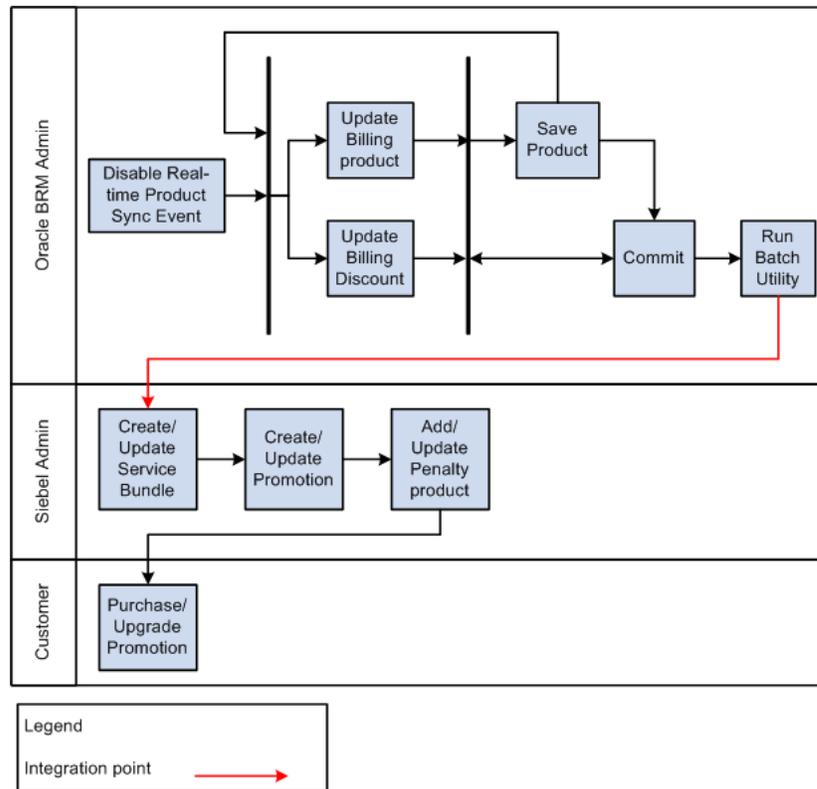
To disable the event for real-time product synchronization, see the Oracle BRM documentation.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "Service Integration Components," Synchronization Queue Data Manager, Installing and configuring the Synchronization Queue DM, Starting and stopping the Synchronization Queue DM.

3.2 Simple and Customizable Products

When products are created in Oracle BRM, they are associated with events that determine how much and how often to charge customers. These events are called billable events. Each product that is created in Oracle BRM is associated with one or more billable events. After the products are synchronized with Siebel CRM, the products that are associated with a single event are synchronized as simple products and products that are associated with multiple events are synchronized as customizable products.

Table 3–1 shows how products are synchronized to Siebel:

Table 3–1   Synchronizing Products to Siebel CRM

In Oracle BRM	In Siebel CRM
Internet	Internet - $25
- Monthly Cycle Forward Event - $25	- Internet Purchase - $30
- Product Purchase Fee Event - $30	
- Delayed Telcom GSM Session Event - 0.40	

3.2.1 Synchronization of Billing Products with Pricing Details

Synchronization of Billing Products with Pricing details

Figure 3–5 shows the synchronization of billing products with pricing details.

Figure 3–5   Synchronizing Billing Products with Pricing Details
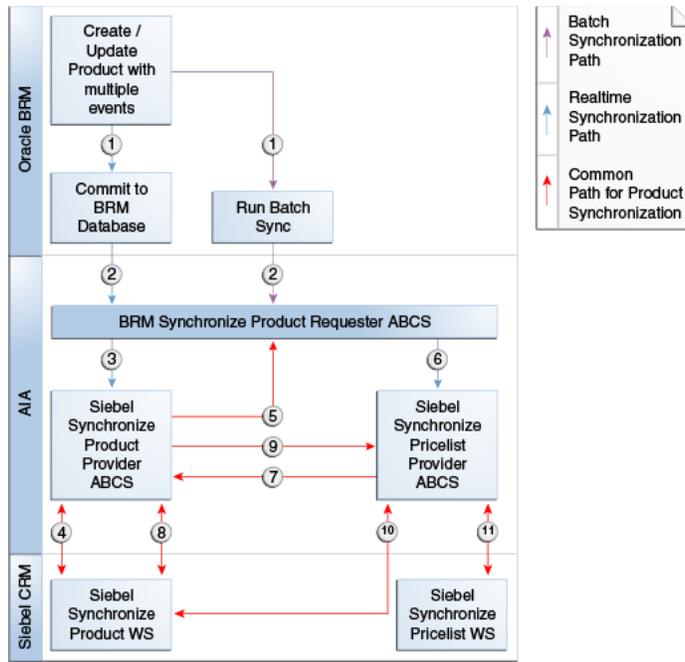
For this flow, the following events occur:

1. You create billing products (single-event multi-event) in the BRM Pricing Center tool. When the new products are created they are synchronized to the target Siebel CRM by either realtime or batch synchronization.

2. The products are committed to the Oracle BRM database and realtime synchronization is invoked (or a batch utility is executed to synchronize the products as a batch). When the realtime or batch synchronization of billing products is invoked, a business event is raised in the Oracle BRM application, which also has the complete definition of the products (ProductABM).

3. The connector service (AQ Consumer), which is subscribed to this business event takes the input (ProductABM) and extracts all the product-related details and passes the message to the Oracle BRM product requester.

    The Requestor service routes the message to the Siebel application-specific connector service (Siebel Synchronize Product Provider).

4. The Siebel Synchronize Product Provider service transforms the standardized product definition (ItemCompositionListEBM) to a Siebel application-specific definition of the product. It invokes the Siebel application web services to create the products in the Siebel application. The status of the web service call (Success or Fail) is returned back to the caller service (Siebel Synchronize Product Provider).

5. The Siebel Synchronize Product Provider service processes the status and sends the details to the Host application connector service (BRM Synchronize Product Requester ABCS) using a standardized response message (ItemCompositionResponseEBM).

6. Once the products are successfully created, the BRM Synchronize Product Requester ABCS extracts the pricing information from the billing products and transforms them into a standardized representation of the pricing (PriceListEBM). The service provides the PriceListEBM as input.

    The Provider service routes the message to the Siebel application-specific connector service (Siebel Synchronize Pricelist Provider).

7. The Siebel Synchronize Pricelist Provider service transforms the standardized pricelist definition (PriceListEBM) to the Siebel application-specific definition of the pricing. If there is multiple charge type associated with the pricing (Events) then simple products are created in the target CRM for each charge type. The pricing related to the charge types are assigned to the corresponding simple product. To create simple products, the connector service transforms the charge types (Events) into a standardized representation of the items (ItemCompositionListEBM).

    The Provider service routes the message to the Siebel application-specific connector service (Siebel Synchronize Product Provider).

8. The Siebel Synchronize Product Provider service transforms the standardized product definition (ItemCompositionListEBM) to a Siebel application-specific definition of the product. It invokes the Siebel application web services to create the simple products for each charge type in the Siebel application. The status of the web service call (Success or Fail) is returned back to the caller service (Siebel Synchronize Product Provider).

9. The Siebel Synchronize Product Provider service processes the status and sends the details to the caller Siebel Synchronize PriceList Provider service using a standardized response message (ItemCompositionResponseEBM).

10. The Siebel Synchronize PriceList Provider service updates the simple products created earlier with the pricing attributes of the product (Price Type) by invoking the Siebel product creation web service. The status of the web service call (Success or Fail) is returned back to the caller service (Siebel Synchronize PriceList Provider).

11. The Siebel Synchronize PriceList Provider service updates the pricelist for all products with the actual pricing information (List Price, Effectivity, and so on) associated with the products. The status of the web service call (Success or Fail) is returned to the caller service (Siebel Synchronize PriceList Provider).

Setting the Billable Flag for Products in Siebel CRM

During the product synchronization from Siebel CRM to Oracle BRM, the billable flag is set for all products of billing type *Subscription*. The billable flag is not set for products of billing type *Event*.

For service bundles, promotions, and simple products of billing type *Special Rating*, the billable flag must be manually set in Siebel CRM.

For more information about setting the billable flag in Siebel, see the *Siebel Communications Guide*, "Profiles in Siebel Communications."

3.2.1.1 Product Attributes

These product attributes are included for all the products in the XML message that is sent to Siebel:

- Product Name
- Product Type

- Purchase Level

- Description

- Billable Events

- Rate Plan

- Effective Start Date and Effective End Date

Rate plan details (charges) go into the price list line while the remaining attributes go into the product.

3.2.1.2 Effective Start and End Dates

The values for the effective start date and the effective end date published by Oracle BRM are communicated and set in Siebel CRM by the Oracle AIA product synchronization process.

For use cases where the effective start date and effective end date are unspecified or has infinite effectivity, the Oracle BRM EAI parameter `infranet.eai.xml_zero_epoch_as_null` must be set to *TRUE*. This ensures that Oracle BRM publishes a null value for the effective start date and the effective end date.

> **Caution:** This is a mandatory step as part of the post installation setup activity.

For more information about the behavior effective dates based on the Oracle BRM EAI parameter, see *Oracle Communications Billing and Revenue Management JCA Resource Adapter*, "Deploying and Configuring the BRM JCA Adapter."

3.2.2 Synchronization of Billing Discounts

Figure 3–6 shows the synchronization of billing discounts.

Figure 3–6   Synchronizing Discounts Flow
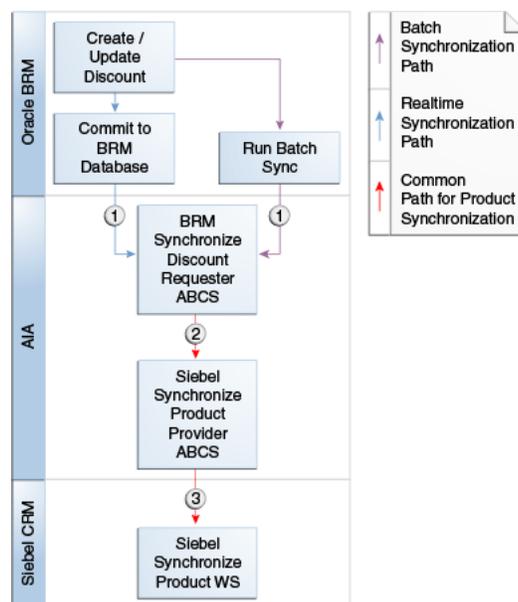
For this flow, the following events occur:

1. You create billing discounts in the BRM Pricing Center tool. Once the products are created, they are synchronized to the target Siebel CRM either realtime or using a batch synchronization. The products are committed to the Oracle BRM database and realtime synchronization is invoked. (A batch utility must be executed to synchronize the discounts as a batch). When the realtime or batch synchronization is invoked, a business event is raised in Oracle BRM, which also has the complete definition of the discount (DiscountABM).

2. The connector service (BRM Synchronize Discount Requestor) that is subscribed to this business event takes the input DiscountABM and extracts all the discount related details and transforms them into a standardized representation of the discount (ItemCompositionListEBM).

   The service routes the message to the Siebel application specific connector service (Siebel Synchronize Product Provider). The discounts are created as simple products in Siebel CRM.

3. The Siebel Synchronize Product Provider service transforms the standardized discount definition (ItemCompositionListEBM) to a Siebel application-specific definition of the product. It invokes the Siebel application web services to create the products in the Siebel application that corresponds to the discount that is published from Oracle BRM. The status of the web services call (Success or Fail) is returned back to the caller (Siebel Synchronize Product Provider service).

3.2.3 Usage Charges on Products

Note the following exceptions.

If a billing product in Oracle BRM has *Delayed Telco GSM Session* as the only event, then the billing product is synchronized with Siebel as a simple product with no pricelist line created in Siebel CRM.

For example, in Table 3–2 *Delayed Telco GSM Session* is an only event.

Table 3–2 Billing Product with Single Event Example

Product in Oracle BRM	Simple Product in Siebel CRM
Wireless Usage	Wireless Usage
Delayed Telco GSM Session Event - 0.40	

If a billing product in Oracle BRM has two events and one of them is *Delayed Telco GSM Session*, then the billing product is synchronized with Siebel CRM as a simple product. The *Delayed Telco GSM Session* event is not synchronized with Siebel CRM. The list price of the simple product in Siebel CRM is set to charge on the other event of the billing product.

For example, in Table 3–3 *Delayed Telco GSM Session* is one of two events.

Table 3–3 Billing Product with Two Events Example

Product in Oracle BRM	Simple Product in Siebel CRM
Call Forwarding:	Call Forwarding - $3.00
- Monthly Cycle Forward Event - $3.00	
- Delayed Telco GSM Session Event - $0.40	

The billing product is synchronized with Siebel CRM as a customizable product if a billing product in Oracle BRM has more than two events and one event is *Delayed Telco*

GSM Session. The *Delayed Telco GSM Session* event is not synchronized with Siebel CRM. The list price of the simple product in Siebel CRM is set to charge on another event of the billing product.

For example, in Table 3–4 *Delayed Telco GSM Session* is one of more than two events:

Table 3–4   Billing Product with More Than Two Events Example

Product in Oracle BRM	Customizable Product in Siebel CRM
Internet:	Internet - $20.00:
- Product Purchase Fee Event - $10.00	- Product Purchase Fee Event - $10.00
- Monthly Cycle Forward Event - $20.00	
- Delayed Telco GSM Session Event - $0.40	

The solution is delivered with the events mapped, as shown in Table 3–5.

Table 3–5   Mapping Events - Solution

Event Name	Event Definition
Product Purchase Fee Event (Activation)	"/event/billing/product/fee/purchase"
Monthly Cycle Arrear Event	"/event/billing/product/fee/cycle/cycle forward arrear"
Monthly Cycle Forward Event	"/event/billing/product/fee/cycle/cycle forward monthly"
Bimonthly Cycle Forward Event	"/event/billing/product/fee/cycle/cycle forward bimonthly"
Quarterly Cycle Forward Event	"/event/billing/product/fee/cycle/cycle forward quarterly"
Annual Cycle Forward Event	"/event/billing/product/fee/cycle/cycle forward annual"
Cycle Forward Arrear Event	"/event/billing/product/fee/cycle/cycle arrear"

You can add more events in the PRICETYPE_EVENT domain value map. Events that are not present in this mapping are not synchronized.

For more information about handling cancel fees (as a result of service, promotion cancellation/upgrade/downgrade, see Section 3.3.11, "Promotion Penalty, Service Activation, and Other MACD Charges."

For more information about DVMs, see Section 24.4, "Working with DVMs."

3.3 Understanding the Product Bundling Methodology

This section describes the methodology for introducing service bundles and marketing bundles in relation to synchronizing billing products from Oracle Communications Billing and Revenue Management (Oracle BRM) to Siebel Customer Relationship Management (Siebel CRM).

This section includes the following topics:

- Section 3.3.1, "Basic Entity Mappings"

- Section 3.3.2, "Defining Products and Discounts in Oracle BRM"

- Section 3.3.3, "Physical Goods"

- Section 3.3.4, "Sales Catalogs"

- Section 3.3.5, "Recommendations for Product Definition in Siebel CRM"

- Section 3.3.6, "Service Bundles"

- Section 3.3.7, "Simple Service Bundles"

- Section 3.3.8, "Marketing Bundles"

- Section 3.3.9, "Balance Groups"

- Section 3.3.10, "Credit Limits"

- Section 3.3.11, "Promotion Penalty, Service Activation, and Other MACD Charges"

- Section 3.3.12, "Supporting Friends and Family"

- Section 3.3.13, "Product Definition Methodology for Friends and Family: Example"

- Section 3.3.14, "Supporting Time-Based Offerings"

3.3.1 Basic Entity Mappings

Table 3–6 shows the mapping between Oracle BRM and Siebel CRM entities.

Table 3–6   Mapping Between Oracle BRM and Siebel CRM Entities

Oracle BRM Entities	Siebel CRM Entities	Description
Product with single event	Simple Product [automatically created]	If a product is associated with a single billable event in Oracle BRM, then a simple product is created in Siebel CRM.
Product with multiple events	Customizable product [automatically created]	If a product is associated with multiple billable events in Oracle BRM, then a customizable product is created in Siebel CRM.
Product Event Binding	Simple Product [automatically created]	Each recurring and nonrecurring type event binding is represented as a simple product.
Discount	Simple Product [automatically created]	A billing discount is represented as a simple product regardless of the number of event bindings.
Balance Impact	Price list Line [automatically created]	A balance impact defined as part of a rate plan in Oracle BRM is mapped to a price list line of a product in Siebel CRM.
Deal	Service Bundle [manually created]	If existing Oracle BRM customers have previously defined deals, those deals are not synchronized as part of the Product Lifecycle Management (PLM) integration. The service bundles must be created manually in Siebel CRM.

Table 3–6   (Cont.) Mapping Between Oracle BRM and Siebel CRM Entities

Oracle BRM Entities	Siebel CRM Entities	Description
Plan	Promotion /Marketing Bundle [manually created]	If existing Oracle BRM customers have been previously defined in a plan, those plans are not synchronized as part of the PLM integration. The Promotion/ Marketing bundles must be created manually in Siebel CRM.
Service Instance	Service Bundle Asset [automatically created]	Purchasing a service bundle results in a service bundle asset that is mapped to an Oracle BRM service instance to support changes to the service.
Purchased Products	Service Bundle Component Asset [automatically created]	Purchasing optional and mandatory components of a service bundle results in asset components that are mapped to Oracle BRM-purchased products.

3.3.2 Defining Products and Discounts in Oracle BRM

When defining the products and discounts in Oracle BRM, use the following guidelines to fully leverage the flexibility and minimize the limitations of this integration:

- Since usage events are *not* synchronized when they are included as a part of multi-event product in Oracle BRM, the name and description of products should include some user-readable identity of the usage. That way the product or price administrator can distinguish the synchronized products on the Siebel side.

- Since the discount value of the Oracle BRM discount objects is *not* synchronized to Siebel CRM, the name and description of the discount objects should include the general intent of the discount to be conveyed on the Siebel order.

- The discountable flag on billing products in Oracle BRM *must be* set to *Y* for all charges that can be discounted when orders are interfaced to billing.

- Oracle AIA does not do time-zone conversion when synchronizing Oracle BRM products and discounts to Siebel CRM.

  The Oracle BRM Enterprise Application Integration (EAI) property infranet.eai.date_pattern controls which time-zone Oracle BRM publishes datetime information in.

  - If the EAI infranet.eai.date_pattern is *not* set, Oracle BRM publishes datetime information in the Oracle BRM local server time zone. (Default)

  - If the EAI infranet.eai.date_pattern property is set, Oracle BRM publishes the datetime information in UTC/GMT time zone.

  For more information about setting this property, see the *Oracle Communications Billing and Revenue Management Developer's Guide*, "Integrating BRM with Enterprise Applications."

3.3.2.1 Using Fixed Amounts versus Scaled Amounts in Oracle BRM

In Oracle BRM, the type of charge associated with a billable event can be either *Scaled* or *Fixed*.

From the user interface (UI) perspective, in the pricing center application of Oracle BRM, when the price must be associated to the event, two fields exist where the charge can be added.

- **Scaled amount**: Specifying the scaled amount allows price overrides and discounts to be applied on the price. When the scaled amount field is used then the fixed amount field must be left empty (null). Zero must not be specified. The scaled amount is specified only for billable events that represent one-time or recurring charges.

- **Fixed amount**: Discount override takes into consideration both fixed and scaled amounts. However price override only overrides the scaled amount. The price overrides can still be applied for the charges but it gets added to the price specified as fixed amount. For example, if the fixed amount on the charge is $5 and a price override is $10 then the price is $15.

Consider the case where both the scaled amount and the fixed amount are specified for the product. The product integration synchronizes the product to Siebel CRM and the list price is the sum of the scaled and fixed amounts. If a discount override is specified for the product, when the order is interfaced to billing the discount override is applied on the sum for the purchased product instance in Oracle BRM.

For example, a billing product has a monthly cycle fee specified as: Scaled = $20 and Fixed = $10.

A discount override of 10% results in a final price of $27 and a discount override of $5 results in a final price of $25.

If a price override is specified for the product, when the order is interfaced to billing, Oracle BRM replaces only the scaled amount with the price override amount for the purchased product instance.

For example, a billing product has a monthly cycle fee specified as: Scaled = $20 and Fixed = $10.

A price override of $15 results in a final price of $25 (Scaled $15 + Fixed $10).

> **Caution:** This behavior for the price override scenario results in a discrepancy between the final price for a product on the order in Siebel CRM and what the customer is actually charged in Oracle BRM. Therefore, it is recommended that you not use fixed amounts for either one-time or recurring charges in Oracle BRM for implementations where the intent is to use the Siebel price override functionality.

For more information about using fixed and scaled amount fields, see *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating*, "About Real-Time Rate Plans."

3.3.3 Physical Goods

Customers can use one of two possible approaches:

- Physical goods can be created as a billing product in Oracle BRM at account-level or at service-level. These are synchronized to Siebel CRM and can be added to the product hierarchy when creating bundles and promotions.

- Physical goods are defined in Enterprise Resource Planning (ERP). In this case, customers are responsible for synchronizing them between ERP and Oracle BRM. The product synchronization process, which is supported by the process integration, is used to synchronize the product from Oracle BRM to Siebel CRM. If

the service or marketing bundle contains one or more physical goods, then those products are passed to Oracle BRM when the order is interfaced to billing.

3.3.4 Sales Catalogs

After all of the Oracle BRM products are synchronized to Siebel CRM, you must add only those products that can be ordered to the catalogs (products whose orderable flag is set). If the customizable products are added to the catalog then the components are automatically added.

Table 3–7   Products Included in the Siebel Catalog

Oracle BRM Entities	Siebel Synchronized Entities	Siebel Catalog
Product: Wireless (Yearly) Event: YCF - $100	Wireless - YCF - $100	It must be added as a component to a service bundle product, which must be added to the sales catalog.
Product: Wireless (Monthly) Event: MCF - $40 Event: Usage - $0.40	Wireless - MCF -$40	The product must be added as a component to a service bundle product, which must be added to the sales catalog.
Product: Wireless Activation Event: Activation - $10	Wireless Activation - $10	The product must be added as a component to a service bundle product, which must be added to the sales catalog.
Product: SMS Activation Event: Activation - $10	SMS Activation - $10	The product must be added as a component to a service bundle product, which must be added to the sales catalog.
Product: SMS Usage Event: Usage - $0.05	SMS Usage	The product must be added as a component to a service bundle product, which must be added to the sales catalog.

3.3.5 Recommendations for Product Definition in Siebel CRM

These are the recommendations for defining products:

- Oracle BRM billing products that are defined with fixed charges should not be discounted in Siebel CRM (using promotion discounts, price overrides, and so forth) because communicating such overrides to Oracle BRM results in a price increase. For this reason it is recommended that only scaled charges be defined for the billing products of type item and subscription with one-time or recurring charges in Oracle BRM.

  For more information, see Section 3.3.2.1, "Using Fixed Amounts versus Scaled Amounts in Oracle BRM."

- The Product Management integration maintains cross-reference information between Oracle BRM billing products and Siebel CRM products. If you delete a billing product in Oracle BRM that is synchronized with Siebel CRM, then the cross-reference data for that billing product is not deleted. This has to be purged manually. It is recommended that instead of deleting the product you inactivate it by specifying an end date.

- If products updated in Oracle BRM result in changing the product structure in Siebel CRM, then you must release the updated product in its respective workspace. This automatically updates the service bundles and the promotions that include the updated product as one of its components.

3.3.5.1 Recommendation for Discounts

This section describes customizable discounts that are time-based or that impact noncurrency resources and multiple event types.

Discounts Defined in Billing Systems

Customizable discounts that are either time-based, or that impact noncurrency resources or multiple event types, must be defined in Oracle BRM. These can be account-level or service-level discounts. Because you can associate general ledger IDs (GLIDs) with them in Oracle BRM, you can account for them in the general ledger in separate accounts if needed.

These discounts are defined in Oracle BRM and synchronized to Siebel CRM as simple products (Structure type = none). The products that represent the discounts are identified using the billing type *Discount*. You manually bundle the service-level discounts into the service bundles.

These can be included or excluded during promotion bundling. The account-level discounts are directly added as components of the promotions and can be made optional based on promotional bundling.

Discounts Defined in Siebel CRM Systems

You can define simple discounts in Siebel CRM when you bundle the billing products into service bundles and promotions. These are usually matrix or promotional discounts. At run time, these discounts getting applied on the order, results in a difference in the start price or list price and the net price.

Defining Overrides on the Product Definition

The following offers you greater control and flexibility in determining how pricing differences between the list price and the selling price are communicated to the billing system. Two new fields are on the Siebel product definition:

- Pricing commit type.

  - The value of the pricing commit type field indicates whether a price override or a discount override is being defined on the product:

  - If the pricing commit type is *Committed*, then a price override has been defined on the product.

  - If the pricing commit type is *Dynamic*, then a discount override has been defined on the product. If a discount override has been defined on the product, then the Dynamic discount method field identifies the discount type.

- Dynamic discount method.

  - If the dynamic discount method is *Amount*, then an amount is defined as the discount value.

  - If the dynamic discount method is *Percent*, then a percent discount has been defined as the discount value.

In Oracle BRM, discount overrides can be tracked in a separate sub-bucket within the GL code that is tied to the product. With discount overrides, mass price changes can also be supported because the list price on the product remains unchanged.

3.3.6 Service Bundles

Billing products (with single or multiple events) are created in Oracle BRM and are synchronized with Siebel CRM. Whenever billing products have to be bundled, one

must manually create a customizable product and set the billing type to *Service Bundle* in Siebel CRM. This product is called a service bundle product. You must add the billing products that are synchronized from Oracle BRM as child components to the service bundle product. Service bundles map to the run-time entity called service instances in Oracle BRM.

The service bundle products are created manually in Siebel CRM.

You can also create a simple service bundle when only one billing product is applicable for a given service.

For more information, see Section 3.3.7, "Simple Service Bundles."

The process integration for order management uses the service bundle to construct the billing service instance for billing management.

The Oracle BRM discounts that are synchronized as products with Siebel CRM can be included in the service bundle. If they are included in the service bundle, then at run time, when the service bundle is purchased and interfaced to billing, those discounts apply to the products within the service instance. If they are not included in the service bundle, but purchased on the order, then they get applied as account-level discounts.

> **Note:** Any product (Oracle BRM product or discount) whose immediate parent is not a service bundle at run time gets purchased as an account-level product or discount in Oracle BRM.

The product bundling methodology gives Siebel CRM product administrators more flexibility when creating service bundles and promotions. Product administrators can nest service bundles and these nested service bundles do not need the same billing service type as the parent or root service bundle. However, within a service bundle, all of the component products must be of the same billing service type. The methodology supports a nested structure in which service bundles can be included as a component of another service bundle.

In the case of multiple billing system instances connected to the same Siebel CRM system instance, all component products within a service bundle reference products from the same billing system. Siebel CRM does not store the target billing instance details.

For more information about service bundles in Siebel CRM, see the *Siebel Communications Guide*.

For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."

In addition to billing products and nonbilling products, the methodology gives a Siebel CRM product administrator the option to include child service bundles and nonservice bundle customizable products as components of a service bundle.

Here are some definitions of the components:

- Billing products are created by the product synchronization. They can be defined as simple products or customizable products based on the number of events. Products with one billable event are synchronized as simple products and products with multiple events are synchronized as customizable products in Siebel CRM.

- Nonbilling products are products that are not originated or synchronized from Oracle BRM. A billing service type should not be specified for nonbilling products.

- Account-level products are associated at the account level and are not associated with any service instance in Oracle BRM; for example, a $2 monthly charge for a hard copy of the bill is charged to the account. The product definition methodology recommends not including account-level products within a service bundle.

- Service bundles can include another service bundle or nonservice bundle customizable products as a component. No limit is placed on the number of levels in the hierarchy. Child service bundles are not required to have the same billing service type as the root bundle.

- Nonservice bundle customizable products are customizable products that group service bundles. Nonservice bundle customizable products can have account-level products and non-billing products as components. They do not have a billing service type.

3.3.6.1 Working with Products and Nested Service Bundles

Consider the following example:

Billing Product - A, Billing Product - B, Billing Product - C, Billing Product - D, Billing Product - E, and Billing Product - F are all synchronized from Oracle BRM to Siebel CRM.

1. Create Service Bundle - SB2 with Billing Product - E as the component.

2. Enrich Billing Product - F so that it becomes a Simple Service Bundle - SSB1

3. Create a Nonservice Bundle Customizable Product (CP) with the following components:

   - Billing Product - D

   - Service Bundle - SB2

   - Simple Service Bundle - SSB1

4. Create Service Bundle - SB1 with the following components:

   - Billing Product - A

   - Billing Product - B

   - Nonservice Bundle CP

Figure 3–7 shows the example of a nested service bundle described previously.

Figure 3–7   Example of Nested Service Bundle Levels
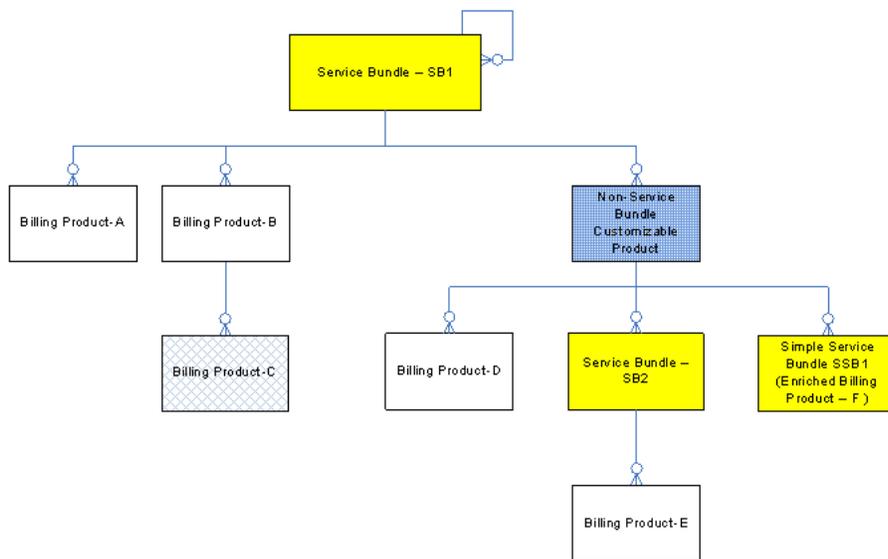
When an order is interfaced to billing, the service bundle gets purchased as a service and the immediate children of the service bundle become purchased product or discount instances for that service instance. *Any product whose immediate parent is not a service bundle gets purchased at the account-level.*

> **Note:**   Dynamic or relationship classes do not get instantiated on the order and are therefore irrelevant in terms of determining a service bundle parent.

Therefore, for the above example, when Service Bundle - SB1 is purchased and the order is interfaced to billing, the following data is created in Oracle BRM:

1. Service instance for Service Bundle - SB1 with purchased product instances for Billing Product - A and Billing Product - B.

   This would be true even if Product Billing - A and Product Billing - B were members of a dynamic or relationship class.

2. Service instance for Service Bundle - SB2 with purchased product instance for Billing product - E.

3. Service instance for Simple Service Bundle - SSB1 with purchased product instance for Billing Product - F.

   Billing Product - D is purchased at the account-level because its immediate parent is not a service bundle.

   Billing Product - C is purchased at the account-level because its immediate parent is not a service bundle.

> **Note:**   If the intention was for Billing Product - C to be purchased for Service Bundle - SB1, then it should be modeled as a sibling of Billing Product - A and Billing Product - B.

3.3.6.2 Working with Nonservice Bundle Customizable Products

Nonservice bundle customizable products can group service bundles (including nested service bundles), simple service bundles, and billing products or discounts. They serve as re-usable components for use across promotions or as is.

Figure 3–8   Using Nonservice Bundle Customizable Products as Root Products in Siebel CRM
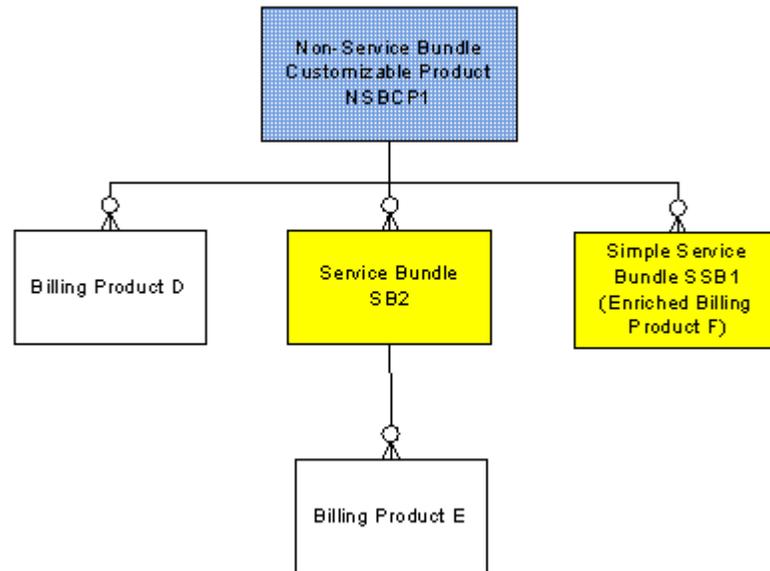
> **Note:**   Using nonservice bundle customizable products are optional. The main benefit of using them is when you are creating promotion variants. The nonservice bundle customizable product can group relevant products. When creating promotions you can add it as a component and include or exclude the components based on the promotion definition. This saves the additional overhead of adding all the components each time a new promotion is created.

When a nonservice bundle customizable product is purchased, the same rules that are listed in the previous section apply. Therefore, using the previous diagram as an illustration, when the Nonservice Bundle Customizable Product - NSBCP1 is purchased and the order is interfaced to billing, the following data is created in Oracle BRM:

1. Service instance for Service Bundle - SB2 with purchased product instance for Billing Product - E.

2. Service instance for Simple Service Bundle - SSB1 with purchased product instance for Billing Product - F.

   Billing Product - D is purchased at the account-level because its immediate parent is not a service bundle.

3.3.6.3 Working with Service Bundles with a Child Non-Service Bundle Customizable Product

Figure 3–9 shows the hierarchical relationships of service bundles with a child nonservice bundle customizable product in Siebel CRM:

Figure 3–9   Service Bundles with a Child Nonservice Bundle Customizable Product
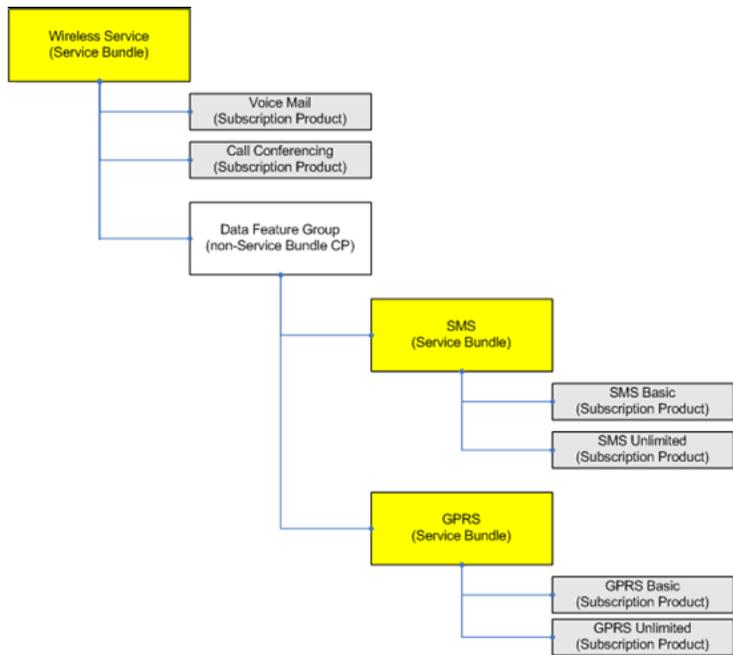
For this diagram:

- No limits are imposed on the number of levels of service bundles. The diagram depicts three levels of bundles.

- The billing products, Voice Mail and Call Conferencing, within a service bundle share the same billing service type as the parent, Wireless Service. Similarly, general packet radio service (GPRS) Basic and GPRS Unlimited share the same billing service type as GPRS, and short message service (SMS) Basic and SMS Unlimited share the same billing service type as SMS.

- If a service bundle does not contain any billing products (for example, if it contains only a child service bundle that has component products or account-level products), then no service instance is created in Oracle BRM for the service bundle. According to the product definition methodology, at least one subscription-based billing product must be a component product of a service bundle. If no billing products are component products, then customers can use a nonservice bundle customizable product. Service instances are not created in Oracle BRM for nonservice bundle customizable products.

Depending on the commercial strategy (rules, usability, and user journey), the service in the preceding diagram can also be modeled and ordered differently. Figure 3–10 is an example of using a nonservice bundle customizable product as the root product.

Figure 3–10   Using a Nonservice Bundle Customizable Product as the Root Product
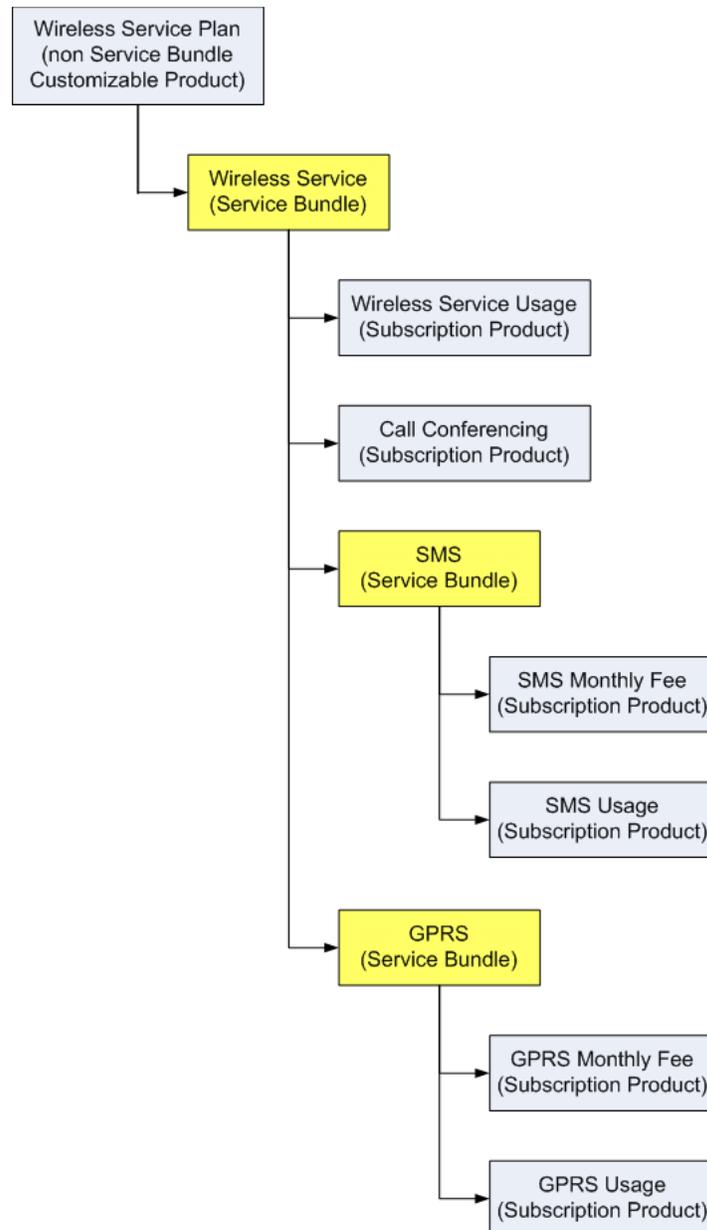
> **Caution:**   Service bundles must not combine products from different billing systems. The order processing integration fails because it expects all billing products within a service bundle to be interfaced to a single billing system.

3.3.7 Simple Service Bundles

Using the service bundle methodology, Siebel CRM product administrators must define a new customizable product of billing type *Service Bundle* and then bundle the subscription product inside it.

The simple service bundle methodology obviates the additional service bundle product definition. This alternate methodology does *not* replace the current one, but is supported in addition to the current one.

The simple service bundle can be a root-level product or can be nested within another service bundle (of billing type *Service Bundle*) or nested within a regular customizable product in Siebel CRM.

This can be achieved by setting the service instance flag (serv_instance_flag in Siebel CRM) to *Y* for a subscription product that has been synchronized from Oracle BRM.

Subscription products can be either:

- A customizable product (this represents a multi-event product in Oracle BRM) with billing type *Subscription*.

- A simple product (this represents a single-event product in Oracle BRM) with billing type *Subscription*.

At run time, when a simple service bundle is purchased, the integration creates both a service instance and a purchased product instance in the billing system.

Also:

- In release 2.4, any subscription product whose immediate parent is not a service bundle is processed as an account-level product at run time when interfaced to billing.

- In release 2.5, any subscription product whose immediate parent is not a service bundle *and* is not service instance-enabled is an account-level product.

Note: The Product Lifecycle Management (PLM) sync neither sets this flag when the product is synchronized from Oracle BRM to Siebel CRM as part of product creation nor updates or overwrites it as part of product updates synchronized from Oracle BRM to Siebel CRM. The Siebel CRM product administrator sets the Service Instance flag manually.

3.3.7.1 Guidelines for Using Service Bundles or Simple Service Bundles

Essentially the CSP's product bundling requirements determine whether the Siebel CRM product administrator uses the classic or the new model to define service bundles.

The simple service bundle model can be used when only one billing product is applicable for a given service. It does not have any service-level billing discounts tied to it, nor does a need exist to be switching from one product variant to another while retaining the same service. No need exists for special rating for this product either.

For more information, see Section 3.3.7.3, "Assumptions and Constraints."

Once a product is defined using the simple service bundle methodology, you cannot switch to using the other one (and vice versa) because that adversely affects processing of change orders for existing assets. If the product bundling requirements change, requiring the use of the other methodology, then you must define another product in billing, synchronize it to CRM, and bundle it differently.

Also, because you have a single asset representing both the service instance and billing product, you cannot upgrade a customer from a service modeled in this manner to one modeled based on the other methodology while retaining the same service instance. You can do the upgrade using a service cancellation and repurchase.

3.3.7.2 Service Bundle versus Simple Service Bundle Example

Here is an example of the service bundles versus simple service bundles. Remember that both are supported.

Table 3–8   Service Bundles versus Simple Service Bundles

Hierarchy	Service Bundle	Hierarchy	Simple Service Bundle
1	CP: Internet Access Service (SB)	1	CP: Internet-MCF (SSB)
1.1	----- CP: Internet - MCF (SBO)	1.1	----- Internet - Activation (SBO)
1.2	----- Internet - Activation (SBO)		
	Note: The internet product is mapped to multiple events in Oracle BRM.		

Table 3–9   Service Bundle versus Simple Service Bundle (Cont'd)

Hierarchy	Service Bundle	Hierarchy	Simple Service Bundle
2	CP: Internet Service (SB)	2	CP: Internet Service (SSB)
2.1	---- Dynamic Class	2.1	---- Dynamic Class
Only 1 of	*Basic High Speed Internet MCF (SBO)*	Only 1 of	*Basic High Speed Internet MCF (SBO)*
these 3 is	*Premium High Speed Internet MCF (SBO)*	these 3 is	*Premium High Speed Internet MCF (SBO)*
selected	*Elite High Speed Internet MCF (SBO)*	selected	*Elite High Speed Internet MCF (SBO)*
2.2	----- Internet Secure Firewall (SBO)	2.2	----- Internet Secure Firewall (SBO)
2.3	----- CP: High Speed Internet Features (NSB-CP)	2.3	----- CP: High Speed Internet Features (NSB-CP)
2.3.1	-------- CP: Internet email (SB)	2.3.1	---------- Internet email (SSB)
2.3.1.1	------------- Internet email (SBO)	2.3.2	---------- Internet Instant Chat (SSB)
2.3.2	-------- CP: Internet Instant Chat (SB)	2.3.3	---------- Internet Conference Chat (SSB)
2.3.2.1	------------- Internet Instant Chat (SBO)	2.3.4	---------- CP: Internet Media (SB)
2.3.3	-------- CP: Internet Conference Chat (SB)	2.3.4.1	--------------- Internet Content on Demand (SBO)
2.3.3.1	------------- Internet Conference Chat (SBO)	2.3.4.2	--------------- Internet Video on Demand (SBO)
2.3.4	-------- CP: Internet Media (SB)	2.3.4.3	--------------- High Speed Internet First Month-Free Discount (SBO)
2.3.4.1	------------- Internet Content on Demand (SBO)		**Note:** The NSB-CP is optional; without it the four-feature SBs have the Internet Service SB as the parent
2.3.4.2	------------- Internet Video on Demand (SBO)		
2.3.4.3	------------- High Speed Internet First Month- Free Discount (SBO)		
	Note: The NSB-CP is optional; without it the four-feature SBs have the Internet Service SB as the parent.		

Legend:

SBO - Service bundle component product synced from Oracle BRM.

SB - Service bundle manually created in Siebel CRM, billing type set to *Service Bundle*.

SSB - Subscription product synced from Oracle BRM, whose Service Instance flag is set to *Y* (SSB - Simple Service Bundle).

3.3.7.3 Assumptions and Constraints

These are the assumptions and constraints:

1. Only products of type *Subscription* can be updated to be a simple-service bundle. This is enforced using Siebel validation.

2. Existing products that have pending quotes, orders, or assets in Siebel CRM or Oracle BRM referencing them cannot be enabled as simple-service bundles. This is because doing so impacts existing asset cross-references, and vice versa in products modeled as simple service bundles. You cannot switch from these to using the traditional one (that is, becoming a service bundle component). This is enforced using Siebel validation.

3. To switch from one methodology variant to another (that is, service bundle versus simple service bundle), you must define new products in Oracle BRM, synchronize them, and bundle them using the desired methodology.

4. Disconnecting the simple service bundle results in disconnecting both the service instance and the product in Oracle BRM. In other words, customers cannot upgrade or downgrade from one simple-service bundle to another while retaining the same service instance.

5. Currently, the service ID is required on service bundle lines for the integration to successfully interface the purchase to Oracle BRM. Similarly, the service ID is also required on the simple service bundle.

6. The methodology does *not* allow bundling additional billing product and discounts, special rating products, or other service bundles within a simple service bundle. The only subcomponents that the simple-service bundle can have are products of billing type *Event*; these are synchronized from Oracle BRM. This is enforced using validations in Siebel.

> **Note:** The order billing integration supports the simple-service bundle methodology for all supported features, within the constraints listed previously.

For more information, see Chapter 7, "Understanding the Process Integration for Order Lifecycle Management."

3.3.8 Marketing Bundles

After all of the service bundles are defined, the marketing manager can create marketing bundles or promotions to group services and products that are to be sold as promotions. The promotions definition offers the flexibility to be upgraded to other promotions.

Table 3–10 is an example of a marketing bundle for a wireless promotion with SMS.

Table 3–10   Marketing Bundle for a Wireless Promotion Example

1	Nation 550 Minutes
1.1	-- Wireless Plan
1.1.1	----- Wireless Service
1.1.1.1	-------- Basic Wireless 550
1.1.1.2	-------- Friends
1.1.1.3	-------- Wireless Voice Service Feature
1.1.1.3.1	---------- Wireless Voice Mail

Table 3–10   (Cont.) Marketing Bundle for a Wireless Promotion Example

1.1.1.3.2	---------- Wireless Call Conference
1.1.1.3.3	---------- Wireless Caller ID
1.1.1.3.4	---------- Wireless Call Waiting
1.1.1.3.5	---------- Wireless Call Forwarding
1.1.1.4	-------- Text Messaging
1.1.1.4.1	---------- Text Messaging SMS 200
1.1.1.4.2	---------- Text Messaging Usage
1.2	-- 50% Activation Discount

The definition of marketing bundles is also used as a grouping for balance groups. For example, each promotion defines the boundaries of a balance group such that each included service bundle's service uses shared resources.

By using the communications product bundling methodology, promotion variants can be created by reusing the same nonservice bundle customizable products or service bundles if the bundles have options as components.

> **Note:**   Options are defined as a class-type relationship with the product that represents the options that are included in the relationship domain in Siebel CRM.

The same service bundle can create promotion variants. This ensures that the service is not disconnected during promotion upgrade or downgrade.

For more information, see Section 3.3.13, "Product Definition Methodology for Friends and Family: Example" for more promotion variants created by reusing the service bundles

The following are defined in context of the Promotion in Siebel CRM.

- **Upgrades**: Specify promotions to which the original promotion can be upgraded.
- **Pricing adjustments**: specify the price or discount overrides for the component products at any level in context of the Promotion.

For more information about Price and Discount overrides, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow."

For more information about Promotion definition, see your Siebel documentation.

3.3.9 Balance Groups

Balance groups are defined at the plan level in Oracle BRM, and plans are not synchronized with Siebel CRM. As delivered, the solution does not provide design-time support for balance groups. When the order is interfaced to Oracle BRM for billing, it uses the default account-level balance group.

3.3.10 Credit Limits

Because credit limits are typically defined at the billing-plan level in Oracle BRM, and such plans are not synchronized, customers can optionally define the default credit limits for each separate service type. As delivered, the solution does not support overrides of credit limits at either bundling or order capture time.

3.3.11 Promotion Penalty, Service Activation, and Other MACD Charges

Penalty charges for promotion cancellation, upgrade, or downgrade must be defined as an item type product with a charge in Oracle BRM and synchronized to Siebel CRM. The Siebel CRM promotion disconnect workflow process (ISS Promotion Disconnect Process) must be modified to use the product synchronized from Oracle BRM.

For more information about ISS Promotion Disconnect Process, see the *Siebel Order Management Guide Addendum for Communications*, "Workflows for Employee Asset-Based Ordering."

You can additionally define proration plans in Siebel CRM to prorate the penalty charge. During the order process when a promotion is Canceled, upgraded, or downgraded, Siebel CRM automatically adds the product (for the penalty charge) with the appropriate charge amount onto the order. To support the application of charges for Move, Add, Change, and Disconnect (MACD) actions such as service suspend, resume, move, and cancel, the solution does not rely on the native Oracle BRM event mappings and charge application for such events. Instead, it simulates the application of such charges by relying on Siebel Related Product functionality. This facilitates visibility of the charges on the MACD and change order. Therefore, charges for suspend, resume, move, or disconnect must be defined as item type products in Oracle BRM (for every service type that you enable such a charge application) and synchronized to Siebel CRM. You must then associate these products to the respective service bundles for the various actions (suspend, resume, move, disconnect) as related products.

For more information about Related Product functionality in Siebel, see the *Siebel Order Management Guide Addendum for Communications*, "Employee Asset-Based Ordering".

When a service is suspended, resumed, moved, or disconnected, Siebel CRM automatically adds the appropriate product (for the MACD charge) onto the order.

To model activation charges for a service:

1. Go to Oracle BRM and define an item type product with a one-time charge.

2. Synchronize this item type product to Siebel CRM.

3. Go to Siebel CRM and set the **Track as Asset** flag to *Y*.

3.3.12 Supporting Friends and Family

The Friends and Family feature supports the ability to rate calls to certain phone numbers differently from others.

Special rating products and special rating profile lists in Siebel CRM are used to associate friends and family lists to services. Discounted rating for friends and family lists is defined in Oracle BRM.

Special rating products must be manually defined in Siebel CRM, included in the service bundle along with the usage-based subscription product, and eventually added into the promotion during product modeling. When a promotion is purchased, the customer service representative (CSR) associates lists to the special rating products and optionally adds numbers to the lists. After the order is fulfilled and completed, the customer can update their friends and family lists.

For more information about how the lists are created and associated with the list product during run time, see Section 12.7, "Supporting Friends and Family Lists" and Chapter 20, "CM - Synchronize Customer Special Rating Profile: Implementation."

Figure 3–11 shows the business process task flow for friends and family.

Figure 3–11   Business Process Task Flow
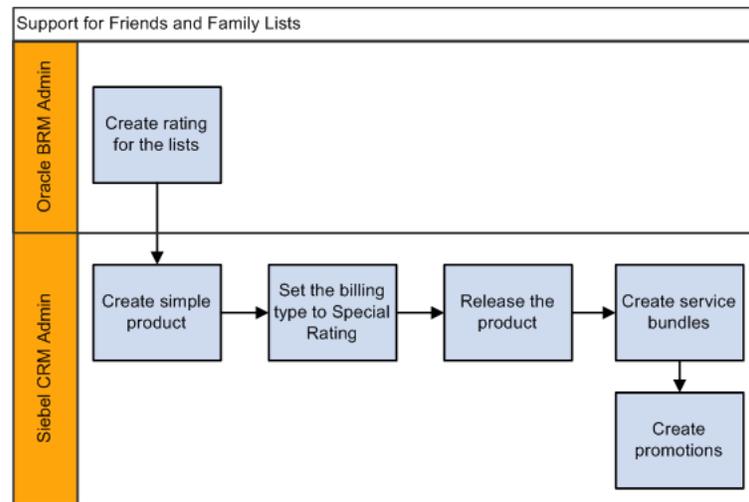
3.3.12.1 Design Time Setup

To enable the Friends and Family support:

You must perform the following in Oracle BRM:

■   Define discounted pricing for friends and family lists. This involves specifying a label name for each list type defined in billing.

> **Caution:**   The solution does not use the Oracle BRM Provisioning Tag Framework to support the Friends and Family feature.

For more information, see the Oracle *BRM Documentation* for "Working with Extended Rating Attributes" and "About rating based on Friends and Family ERA."

You must perform the following in the Siebel CRM Project Workspace:

1.   Create a simple product with a name that is identical to the list label name used in Oracle BRM (while defining the discounted pricing for the lists).

2.   Set the billing type of the product to be *Special Rating*.

3.   Leave the billing service type blank.

> **Tip:**   This allows the use of the same special rating product across different types of services (such as Wireless and VoIP) for which you want to enable Friends and Family.

4.   Set the billable flag to *Y*

5.   Set the track as asset flag to *Y*

6.   Add the special rating products to the service bundle that represents the service that supports friends and family lists. This service bundle must include a usage-based subscription product that is used to rate service usage.

7.   Include the service bundle in the desired promotion(s) and release all the entities.

For more information on friends and family plans, see *Siebel Communications Guide*, "Profiles in Siebel Communications."

3.3.13 Product Definition Methodology for Friends and Family: Example

Table 3–11 and Table 3–12 are examples of the product definition methodology.

Oracle BRM Definition

Table 3–11   Oracle BRM Definition

Products in Oracle BRM	Service Type
Basic Wireless 550	/service/telco/gsm/telephony
------ Monthly Cycle Forward Event	
------ Delayed Telco GSM Event	
Premium Wireless 800	/service/telco/gsm/telephony
------ Monthly Cycle Forward Event	
------ Delayed Telco GSM Event	
Unlimited Wireless Voice	/service/telco/gsm/telephony
------ Monthly Cycle Forward Event	
------ Delayed Telco GSM Event	
Wireless Add On Line	/service/telco/gsm/telephony
------ Monthly Cycle Forward Event	
------ Delayed Telco GSM Event	
------ Product Purchase Fee Event	
Wireless Voice Activation	/service/telco/gsm/telephony
------ Product Purchase Fee Event	
Wireless Voice Mail	/service/telco/gsm/telephony
Wireless Call Conference	/service/telco/gsm/telephony
Wireless Caller ID	/service/telco/gsm/telephony
Wireless Call Waiting	/service/telco/gsm/telephony
Wireless Call Forwarding	/service/telco/gsm/telephony
Text Messaging SMS 200	/service/telco/gsm/sms
Text Messaging SMS 400	/service/telco/gsm/sms
Text Messaging SMS Unlimited	/service/telco/gsm/sms
Text Messaging Usage	/service/telco/gsm/sms
50% Activation Discount	/account

Define discounted pricing in Oracle BRM for rating phone numbers on the Special Rating lists. Use the labels *Friends* and *Family*.

Siebel CRM Representation

Table 3–12   Siebel CRM Representation

Product Name	Service Type	Billing type	Comments
Basic Wireless 550	/service/telco/gsm/telephony	Subscription	Automated
Premium Wireless 800	/service/telco/gsm/telephony	Subscription	Automated
Unlimited Wireless Voice	/service/telco/gsm/telephony	Subscription	Automated
Wireless Add On Line	/service/telco/gsm/telephony	Subscription	Automated
------ Product Purchase Fee Event	/service/telco/gsm/telephony	Event	Automated
Wireless Voice Activation	/service/telco/gsm/telephony	Item	Automated
Wireless Voice Mail	/service/telco/gsm/telephony	Subscription	Automated
Wireless Call Conference	/service/telco/gsm/telephony	Subscription	Automated
Wireless Caller ID	/service/telco/gsm/telephony	Subscription	Automated
Wireless Call Waiting	/service/telco/gsm/telephony	Subscription	Automated
Wireless Call Forwarding	/service/telco/gsm/telephony	Subscription	Automated
Text Messaging SMS 200	/service/telco/gsm/sms	Subscription	Automated
Text Messaging SMS 400	/service/telco/gsm/sms	Subscription	Automated
Text Messaging SMS Unlimited	/service/telco/gsm/sms	Subscription	Automated
Text Messaging Usage	/service/telco/gsm/sms	Subscription	Automated
50% Activation Discount	/account	Discount	Automated
Friends	Not applicable	Special Rating	Manually Created
Family	Not applicable	Special Rating	Manually Created

Table 3–13 contains some examples of the service bundles that include special rating products.

Service Bundles (SB)

Table 3–13   Service Bundles (SB)

Service Bundles	Comments
Wireless Service	Nested Service Bundle
------- Voice Access Options	Relationship of domain type = "Dynamic Class" and the components represent the options
-------------- Basic Wireless 550	--
-------------- Premium Wireless 800	--
-------------- Unlimited Wireless Voice	--
-------------- Wireless Add On Line	--
------- Wireless Voice Activation	--
------- Special Rating Options	Relationship of domain type = "Dynamic Class" and the components represent the options
-------------- Friends	--
-------------- Family	--

Table 3–13   (Cont.) Service Bundles (SB)

Service Bundles	Comments
------- Wireless Voice Service Feature	--
------------- Wireless Voice Mail	--
------------- Wireless Call Conference	--
------------- Wireless Caller ID	--
------------- Wireless Call Waiting	--
------------- Wireless Call Forwarding	--
------- Text Messaging	--
------------- Text Messaging Options	Relationship of domain type = "Dynamic Class" and the components represent the options
-------------------- Text Messaging SMS 200	--
-------------------- Text Messaging SMS 400	--
-------------------- Text Messaging SMS Unlimited	--
------------- Text Messaging Usage	--

> **Note:**   If multiple special rating products are bundled within the same service bundle, it is recommended that they be first grouped into a dynamic class and then included in the service bundle.

Here are some examples of the promotion definition:

Promotions

Table 3–14   Promotions

Nation 550 Minutes
--------- Wireless Plan
------------- Wireless Service
------------------ Basic Wireless 550
------------------ Wireless Voice Activation
------------------ Friends
------------------ Wireless Voice Service Feature
------------------------ Wireless Voice Mail
------------------------ Wireless Call Conference
------------------------ Wireless Caller ID
------------------------ Wireless Call Waiting
------------------------ Wireless Call Forwarding
------------------ Text Messaging
Text Messaging SMS 200
Text Messaging Usage
------------- 50% Activation Discount

Table 3–14 (Cont.) Promotions

Nation 800 Minutes	*Different Promotion Variant created from the same bundle.*
--------- Wireless Plan	
------------- Wireless Service	
------------------ Premier Wireless 800	
------------------ Wireless Voice Activation	
------------------ Friends	*Friends and Family lists added to the Wireless Service*
------------------ Family	
------------------ Wireless Voice Service Feature	
-------------------------- Wireless Voice Mail	
-------------------------- Wireless Call Conference	
-------------------------- Wireless Caller ID	
-------------------------- Wireless Call Waiting	
-------------------------- Wireless Call Forwarding	
------------------ Text Messaging	
-------------------------- Text Messaging SMS 400	
-------------------------- Text Messaging Usage	
------------- 50% Activation Discount	

3.3.14 Supporting Time-Based Offerings

The time-based offerings (TBO) feature enables customers to define and use products and discounts in Siebel CRM that are valid only for a specific period, and expire after that.

Consider a use case in which a service provider wants to offer two promotions:

■ Gold Plan, which provides 50% discount on monthly cycle fee for the first three months

■ Silver Plan, which provides 50% discount on monthly cycle fee for the first two months

Service providers creates a product in Oracle BRM that grants 1000 free minutes. This product is added to the Gold Plan by setting the attribute value for Duration to *3* and UOM to *months*. In case of the Silver Plan, the attribute values for Duration and UOM are set to *2* and *months* respectively.

Similarly, using time-based offering, you can model absolute or percentage-based discounts with limited validity. In case of a TBO upgrade or a downgrade scenario, the attribute ValidityDurationStart is used to calculate the new service end date.

For more information about TBOs, see Section 12.6, "Supporting Time-Based Offerings."

3.3.14.1 Time-Based Offerings Methodology

The two components of the solution for time-based offerings are design time and order time. They are defined as follows:

■ Design Time

- – A custom product class with three validity attributes (Duration, DurationUnitOfMeasure, and DurationValidityStart) is created in Siebel CRM.

  These attribute names must exactly match the names specified previously.

- – Products and discounts that have validity requirements are manually changed from simple products to customizable products.

- – The product type for the products in Siebel is *Time Based Offer*.

- – The validity product class is associated with the products and discounts.

- – The validity attribute values are supplied for bundles or promotions in Siebel CRM.

- Order Time (for a new order or a revision order).

Passing the End Date Value to Oracle BRM

In Siebel CRM, a product class with new attributes must be created. The product class contains three new attributes:

- **Duration**: Used to define the number of days, months, or years.

- **DurationUnitOfMeasure**: The unit used to measure the duration (days, months, or years).

- **DurationValidityStart**: Indicates which date to calculate the end date from. The valid values for this attribute are: *Original Start*, *Order Time*, and *Original End*.

The DurationValidityStart determines how the end date is calculated in a change order scenario. During a change order, if the duration of the product is changed, the new duration is calculated based on the original start date, the current date, or the original end date based on the value of the attribute.

For more information about Oracle Service Management (Oracle OSM), see *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture*.

For more information, see *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating*, "Real-time rating based on date and time."

3.3.14.2 Using Time-Based Offerings

Here is the methodology to use TBOs:

- The products that are valid should be changed manually to a customizable product in Siebel CRM. This is because simple products cannot be associated with a product class.

- A new class with three attributes must be created within Siebel CRM. This class must be made available for use with any charge or discount type product. The attribute names must exactly match what is specified in Table 3–15.

Table 3–15   Product Class - Attributes

Attribute Name	ValueSet	Attribute Type	LOV
Duration	--	int	--
DurationUnitOfMeasure	UnitOfMeasure	enum	0 - none
			1 - seconds
			2 - minutes
			3 - hours
			4 - days
			5 - months
DurationValidityStart	ValidityStart	enum	0 - Original Start
			1 - Now
			2 - Original End

For more information about billing dates, see Appendix C, "OLM - Mapping Billing Dates."

- Associate the previously created class with time-based products and discounts in Siebel CRM.

- Provide values for the validity attributes for products and discounts for promotions and bundles.

- Siebel supports product class hierarchy. The methodology covers the aspects of using the validity attributes in the class hierarchy scenario.

> **Note:**   In Siebel CRM, a product can be associated with a single product class. If class inheritance is not used, validity class may clash with other product classes that are required to support other features.

- The implementer in Oracle OSM must use the attribute names used in Siebel CRM to retrieve the values of the validity attributes.

For more information about how to create products and discounts in Siebel, see the Siebel product documentation.

3.4 Solution Assumptions and Constraints

1. Oracle BRM deals and plans are not synchronized from Oracle BRM to Siebel CRM. The service bundles and promotions are manually defined in Siebel CRM.

2. Credit limits are not synchronized from Oracle BRM to Siebel CRM.

3. Sharing groups are not synchronized from Oracle BRM to Siebel CRM.

4. Multiple brands defined within a single instance of Oracle BRM are not supported by the integration.

5. The synchronization of billing products and billing discounts is one-way. Billing products created or updated in Siebel CRM are not synchronized back to Oracle BRM. Oracle BRM is the product master.

6. The integration does not support multiple price lists or multiple currencies. All products are created under a single price list and a single currency in Siebel CRM.

The price list (with its specified currency) used is the one specified in the AIAConfigurationProperties.xml file.

For more information about configuration properties, see Chapter 24, "Configuring the Process Integration for Product Lifecycle Management."

7. All of the billing products created by this synchronization are associated with one business unit in Siebel CRM. This is the business unit that is specified in the AIAConfigurationProperties.xml file.

   For more information about business units, see the Siebel CRM product documentation.

   For more information about configuration properties, see Chapter 24, "Configuring the Process Integration for Product Lifecycle Management."

8. All of the billing products synchronized to Siebel CRM are created in a single workspace in Siebel CRM. This is the workspace specified in the AIAConfigurationProperties.xml file.

   For more information about workspaces, see the Siebel CRM product documentation.

   For more information about configuration properties, see Chapter 24, "Configuring the Process Integration for Product Lifecycle Management."

9. If a product in Oracle BRM has multiple rate plans or multiple tiers, the integration does not synchronize the pricing information. The price is set to $0 in Siebel CRM for such products.

   For more information, see Section 3.3, "Understanding the Product Bundling Methodology."

10. We recommend you use Siebel discounts for discounting purchase fees on products. Based on the pricing commit type, Siebel discounts get applied as price or discount overrides when the order is interfaced to billing.

    For more information about pricing commit type, see "Defining Overrides on the Product Definition".

    For Oracle BRM purchase fee discounts to get applied consistently, the discount must be purchased before the product that it applies to. Both the order management system and the AIA connector service that interfaces the order to billing must recognize this and currently, the AIA connector service does not handle this sequencing requirement.

    In cases where discounts are defined in Oracle BRM and synchronized to Siebel CRM, they must not be used in the bundling of products to create offers or promotions. Also, products, bundles, or promotions, which have purchase fee discounts must not be used to create quotes or orders.

    > **Note:** This guide does not address upgrade issues for customers that have in-flight orders or transaction data with purchase fee discounts interfaced to billing.

11. The lists associated with the Special Rating products (such as Friends and Family) are defined in Siebel CRM. An external communication must exist between the Siebel CRM product administrator and the Oracle BRM pricing administrator to communicate the name of the lists that were created in Siebel CRM. The Oracle BRM product administrator creates the labels for the corresponding list names in

Oracle BRM. Oracle BRM uses labels to identify the friends and family type lists. The labels are used to associate special pricing models in Oracle BRM Pricing.

12. When a billing product is deleted in Oracle BRM, it does not publish any message. The corresponding billing product in Siebel CRM is not deleted or inactivated automatically. You must inactivate this billing product manually in Siebel CRM. If you delete a billing product in Oracle BRM that is synchronized with Siebel CRM, then the cross-reference data for that billing product is not deleted. This must be purged manually. We recommend that you not delete products in Oracle BRM but instead inactivate the product in Oracle BRM by setting the product end date.

13. The billable events that are associated with billing products in Oracle BRM must be included in the PRICETYPE_EVENT domain value map. If an event is not included in the DVM, the process integration ignores the event. In other words, the process integration does not create a corresponding simple product that represents the event (billing type *Event*) in Siebel CRM. The process integration does not end in error, nor does it send a notification that an event was not found in the DVM.

14. Oracle BRM is the master for usage pricing. When billing products with only one usage event are synchronized from Oracle BRM, a simple product with a price type of *One-Time* is created in Siebel CRM. The pricing information for such products must not be changed in Siebel CRM. For example, a price override or discount must not be specified in Siebel CRM. If the price is updated in Siebel CRM the changes are not propagated to Oracle BRM or applied when the order is interfaced to billing.

15. A service bundle can have another service bundle as a component product. A service bundle that does not have another service bundle as one of its component products must have the same billing service type as its component products. Violation of this assumption can result in Oracle BRM grouping the billed charges under the wrong bucket (bill-item).The product synchronization sets the asset-trackable flag to *Y* for Oracle BRM products of type *Subscription* and *N* for products of type *Item* or *System*.

16. The product synchronization process ignores the effective start date and effective end date that are specified on the rate tier of the billing products. The effective start date on the price line in Siebel CRM is set to the creation date and time and the effective end date is not set.

17. Because BPEL flows are transactional in nature, they must not be used for either initial data loads or considerable-sized data loads. Instead, you should create your own data loading capability using appropriate tools or scripts. You must also create scripts to populate cross-reference data.

18. Out of the box in Siebel CRM, a single penalty product can be associated to the promotion disconnect workflow process (ISS Promotion Disconnect Process). When a promotion is violated (early termination), the penalty charge gets added to the quote or order and the penalty can be processed or applied in the billing system from where the penalty product was originally synchronized to Siebel CRM.

Siebel CRM must be extended to support scenarios where multiple penalties are applied and processed in different billing systems (such as mutli-play promotions where products are billing fulfilled in different billing systems).

4

PLM - Synchronize Product and Price: Implementation

This chapter provides an overview of the synchronize product and price business flow and discusses Siebel Customer Relationship Management (Siebel CRM) and Oracle Billing and Revenue Management (Oracle BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 4.1, "Synchronize Product and Price Business Flow Overview"
- Section 4.2, "Product Synchronization Integration Flow"
- Section 4.3, "Billing Discount Synchronization Integration Flow"
- Section 4.4, "Oracle BRM Interfaces"
- Section 4.5, "Siebel CRM Interfaces"
- Section 4.6, "Industry Oracle AIA Components"
- Section 4.7, "Integration Services"

4.1 Synchronize Product and Price Business Flow Overview

The Synchronize Product and Price integration between Oracle BRM and Siebel Customer Relationship Management (Siebel CRM) supports the following integration flows (in real time or batch mode):

- Product Synchronization enables you to create new products in Oracle BRM and then synchronize those products in Siebel CRM.
- Billing Discount Synchronization enables you to create billing discounts as billing products in Oracle BRM and then synchronize those billing discounts with Siebel CRM.

4.2 Product Synchronization Integration Flow

The product synchronization integration flow enables you to create new products in Oracle BRM and then synchronize those products in Siebel CRM. The products created are used by the Order Capture and Asset Tracking modules in Siebel CRM. The product synchronization integration flow also enables updates to existing products in Oracle BRM. The updates are then synchronized in Siebel CRM.

When products are created in Oracle BRM, those products have multiple events, each with a price, which differs from Siebel CRM, which has only one product and a price for that product.

The product synchronization integration flow takes multiple events with recurring prices first. If no recurring price event exists, then the integration takes the first event and makes that the main (parent) product with a price.

After synchronizing the product data from Oracle BRM to Siebel CRM, Siebel returns a message containing both the product and the price data. The integration separates the product data from the price list data, and then synchronizes the price data in a separate process.

This integration flow delivers these services:

- SyncProductBRMCommsReqABCSImpl with operation SyncProduct
- SyncItemCompositionListSiebelCommsProvABCSImpl with operation SyncItemCompositionList
- ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl with operation ProductOptimizedSyncPriceListList

Figure 4–1 is the sequence diagram for the product synchronization.

Figure 4–1   Product Synchronization Sequence Diagram

When this process starts, the following events occur:

1. In the Oracle BRM Pricing Center, go to **File, New**. Drag the product to edit to the right pane. Double-click the product name and edit appropriate fields. To commit the changes, click **File, Commit to BRM Database**. The modified product is published to the Oracle BRM product queue.

2. Dequeue the Oracle BRM product queue. The adapter SyncProductInfoChangeBRMAQ polls the Oracle BRM product queue.

    It is dequeued whenever it sees a message in the queue and invokes the SyncProductBRMCommsReqABCSImpl with the operation SyncProduct.

3. The SyncProductBRMCommsReqABCSImpl first transforms the Oracle BRM product message into an ItemCompositionEBM and routes the message to the SyncItemCompositionListSiebelCommsProvABCSImpl

4. The SyncItemCompositionListSiebelCommsProvABCSImpl transforms the ItemCompositionEBM into the Siebel product message and then calls the Siebel product web service on operation SWIProductImportUpsert. The Siebel web service completes the request and returns a response message. The SyncItemCompositionListSiebelCommsProvABCSImpl then transforms the Siebel response message to an ItemCompositionResponseEBM and sends it back to the SyncProductBRMCommsReqABCSImpl.

5. The SyncProductBRMCommsReqABCSImpl transforms the Oracle BRM product message into the PriceListListEBM and routes the message to the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl.

6. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl first identifies the event to be associated with the main product and then transforms the SyncPriceListListEBM to a SyncItemCompositionListEBM and routes the message to the SyncItemCompositionListSiebelCommsProvABCSImpl.

7. The SyncItemCompositionListSiebelCommsProvABCSImpl transforms the ItemCompositionEBM to the Siebel product message and then calls the Siebel product web service on operation SWIProductImportUpsert. The Siebel web service completes the request and returns a response message. SyncItemCompositionListSiebelCommsProvABCSImpl then transforms the Siebel response message to an ItemCompositionResponseEBM and returns it to the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl.

8. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl transforms the PriceListEBM to a Siebel price list message and then calls the Siebel price list web service on operation Price_spcList_spcItem_spcInsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl transforms the PriceListListEBM to a Siebel product message and then calls the Siebel product web service on operation SWIProductImportUpsert. The Siebel web service completes the request and returns a response message. SWIProductImportUpsert then transforms the Siebel response message to a PriceList ListResponseEBM.

4.3 Billing Discount Synchronization Integration Flow

The billing discount synchronization integration flow enables you to create billing discounts as billing products in Oracle BRM and then synchronize those billing discounts with Siebel CRM. The billing discounts created are used by the Order Capture module in Siebel CRM.

The billing discount synchronization flow also enables updates to billing discounts in Oracle BRM. The updates are then synchronized in Siebel CRM.

The billing discount synchronization integration flow synchronizes only the basic billing discount attributes. It does not synchronize any price information. Add the billing discount detail information in the description of the billing discount when creating billing discounts

This integration flow delivers these services:

- SyncDiscountBRMCommsReqABCSImpl with operation SyncDiscount

- SyncItemCompositionListSiebelCommsProvABCSImpl with operation SyncItemCompositionList

Figure 4–2 is the sequence diagram for the billing discount synchronization:

Figure 4–2   Billing Discount Synchronization Sequence Diagram
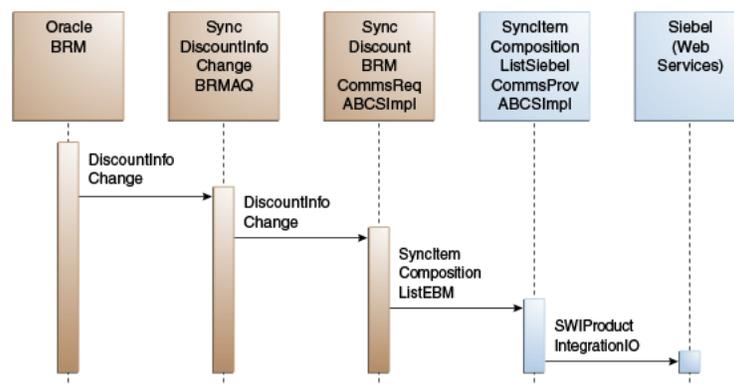
When this process starts, the following events occur:

1. In the Oracle BRM Pricing Center, go to **File, New**. Drag the discount to edit to the right pane. Double-click the product name and edit the appropriate fields. To commit the changes, click **File, Commit to BRM Database**. The modified discount is published to the Oracle BRM discount queue.

2. Dequeue the Oracle BRM discount queue. The adapter SyncDiscountInfoChangeBRMAQ polls the Oracle BRM discount queue. It is dequeued whenever it sees a message in the queue and invokes the SyncDiscountBRMCommsReqABCSImpl with the operation SyncDiscount.

3. The SyncDiscountBRMCommsReqABCSImpl first transforms the Oracle BRM discount message into the ItemCompositionEBM and routes the message to the SyncItemCompositionListSiebelCommsProvABCSImpl.

4. The SyncItemCompositionListSiebelCommsProvABCSImpl transforms the ItemCompositionEBM into the Siebel product message and then calls the Siebel product web service on operation SWIProductImportUpsert. The Siebel web service completes the request and returns a response message to the SyncItemCompositionListSiebelCommsProvABCSImpl.

4.4 Oracle BRM Interfaces

The process integration for product management uses these services:

- SyncProductInfoChangeBRMAQ: The adapter SyncProductInfoChangeBRMAQ polls the BRM Product queue. It dequeues whenever it sees a message in the queue and invokes SyncProductBRMCommsReqABCSImpl with the operation SyncProduct.

- SyncDiscountInfoChangeBRMAQ: The adapter SyncDiscountInfoChangeBRMAQ polls the BRM Discount queue. It dequeues whenever it sees a message in the queue and invokes SyncDiscountBRMCommsReqABCSImpl with the operation SyncDiscount.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "Service Integration Components," Synchronization Queue Data Manager.

4.5 Siebel CRM Interfaces

The process integration for product management uses these Siebel CRM interfaces:

- SWIISSPriceListItemIO: This web service is used to perform basic operations on a price list such as insert or update (upsert).

- SWIProductImport: This inbound web service is used to support Insert or Update of product details, which allows the import of external product information into the Siebel database.

For more information, see the *Siebel CRM Web Services Reference*, "Siebel CRM Primary Web Services."

4.6 Industry Oracle AIA Components

The process integration for product management uses the following delivered Industry AIA components:

- ItemCompositionEBO

- SyncItemCompositionListEBM

- SyncItemCompositionListResponseEBM

- PriceListEBO

- SyncPriceListListEBM

- SyncPriceListListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

4.7 Integration Services

These services are delivered with this integration:

- SyncProductBRMCommsReqABCSImpl

- SyncDiscountBRMCommsReqABCSImpl

- SyncItemCompositionListSiebelCommsProvABCSImpl

- ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

Some of these services have been enabled to use Session Pool Manager.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

4.7.1 SyncProductBRMCommsReqABCSImpl

The SyncProductBRMCommsReqABCSImpl has the operation SyncProduct and performs all of the Product/Item-related actions such as Create Product/Item, Update Product/Item, and Sync Product/Item.

This service accepts a BRM product message as a request and does not return a response. An Oracle BRM product message has two sets of information:

- Standard product attributes.

- Pricing information that can be mapped to a PriceLine of a PriceList.

Because it has two sets of information, the Oracle BRM product message is transformed into two EBMs: one for the product (SyncItemCompositionListEBM) and another for the PriceLine (SyncPriceListList EBM).

The program first prepares the SyncItemCompositionListEBM with the basic product information.

After the SyncItemCompositionList is complete, it prepares a SyncPriceListListEBM with the pricing information of the Oracle BRM message. It fetches the PriceList name from a configuration parameter.

The configuration parameter is located in the AIAConfigurationProperties.xml file.

4.7.2 SyncDiscountBRMCommsReqABCSImpl

The SyncDiscountBRMCommsReqABCSImpl is a BPEL service and it is the Oracle BRM discount request ABC implementation. It has the operation SyncDiscount. This accepts an Oracle BRM discount message as a request and does not return a response.

The SyncDiscountBRMCommsReqABCSImpl service accepts an Oracle BRM discount message. An Oracle BRM discount is created as a product for all of the recipients. An Oracle BRM discount message has basic discount attributes and does not contain any pricing information. The Oracle BRM discount message is transformed into the SyncItemCompositionListEBM with the basic discount information.

4.7.3 SyncItemCompositionListSiebelCommsProvABCSImpl

The SyncItemCompositionListSiebelCommsProvABCSImpl process accepts the SyncItemCompositionListEBM. It transforms SyncItemCompositionListEBM into the Siebel product application business message (ABM). It then invokes the Siebel Product web service to create products and product structures in Siebel.

This service is SPM enabled.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

4.7.4 ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

The ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service performs all of the PriceList-related actions such as Create PriceList, Update PriceList, Sync PriceList, and Sync PriceListList. This operation has the standard create, read, update, delete (CRUD) operations.

This service transforms the PriceListEBM into a Siebel price list message and then calls the Siebel price list web service on operation Price_spcList_spcItem_spcInsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl transforms the PriceListListEBM into a Siebel product message and then calls the Siebel product web service on operation SWIProductImportUpsert. The Siebel web service completes the request and returns a response message. SWIProductImportUpsert then transforms the Siebel response message to a PriceList ListResponseEBM.

This service is SPM enabled.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

5

PLM - Understanding the Query Product Classes Business Flow

This chapter provides an overview of product classes in Siebel Customer Relationship Management (Siebel CRM) and discusses the design-time setup in Service Creation Environment (SCE) Studio and the support for effectivity.

This chapter includes the following sections:

- Section 5.1, "Product Classes in Siebel CRM Overview"
- Section 5.2, "Design-Time Setup in SCE Studio"
- Section 5.3, "Solution Assumptions and Constraints"

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

5.1 Product Classes in Siebel CRM Overview

In Siebel CRM, product classes provide a way to organize and maintain product attributes. Product attributes describe the characteristics of products. Examples include color, size, speed, and so on. You define product attributes and associate them with one or more product classes. Product attributes have a range of values associated with them called the attribute domain. For example, an attribute bandwidth has a range of values, such as 2Mbps, 3 Mbps, 5 Mbps, and so on.

Siebel CRM supports product class hierarchies using inheritance; that is, the attributes that are associated with a class are automatically inherited by all of the subclasses in the hierarchy. The subclasses can be nested as deeply as needed. The standard recommendation is to use three levels in the hierarchy.

5.2 Design-Time Setup in SCE Studio

SCE Studio is an Eclipse plug-in-based application that maintains data in file format. It provides a data dictionary to store the metadata definitions, and data models to define various entities that are used by Oracle Order and Service Management (Oracle OSM). It provides a platform to define rules, functions, and dependencies that can be stored as files and deployed on a deployment server. These files are used during the order fulfillment process.

The current version of SCE Studio does not offer web services that you can invoke from an external service to create entities supported by it.

5.3 Solution Assumptions and Constraints

These are the assumptions and constraints:

1. The SCE Studio queries the product classes from Siebel CRM. Any updates made to the product specification in SCE Studio are not synchronized back to Siebel CRM.

2. SCE Studio maintains the mapping between the Siebel product class and the product specification.

3. Only product class and the associated attributes are imported into the SCE Studio. Structures, constraints, properties, user interface definitions, and so on, which can be associated with the product class in Siebel CRM, are not sent to query from SCE Studio.

6

PLM - Query Product Classes: Implementation

This chapter provides an overview of the Query Product Classes business flow and discusses Siebel Customer Relationship Management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 6.1, "Query Product Classes Overview"
- Section 6.2, "Query Product Classes - Implementation Overview"
- Section 6.3, "Support for Effectivity during Design-Time - Methodology"
- Section 6.4, "Query Product Classes Integration Flow"
- Section 6.5, "Siebel CRM Interfaces"
- Section 6.6, "Industry Oracle AIA Components"
- Section 6.7, "Integration Services"

6.1 Query Product Classes Overview

Product administrators define product classes and transaction attributes in Siebel CRM. The SCE product-to-service mapping specialist queries and imports the product class and the associated transaction attributes into a cartridge from SCE Studio. The product-to-service mapping specialist then maps the product class to a product specification in the cartridge. They use the product specification to associate decomposition rules, fulfillment functions, and their dependencies. After all the design time setup is completed, they deploy the cartridge to Oracle OSM. The cartridge defines various fulfillment topologies to process order lines during order processing.

Figure 6–1 illustrates the flow.

Figure 6–1   Business Process Model Diagram
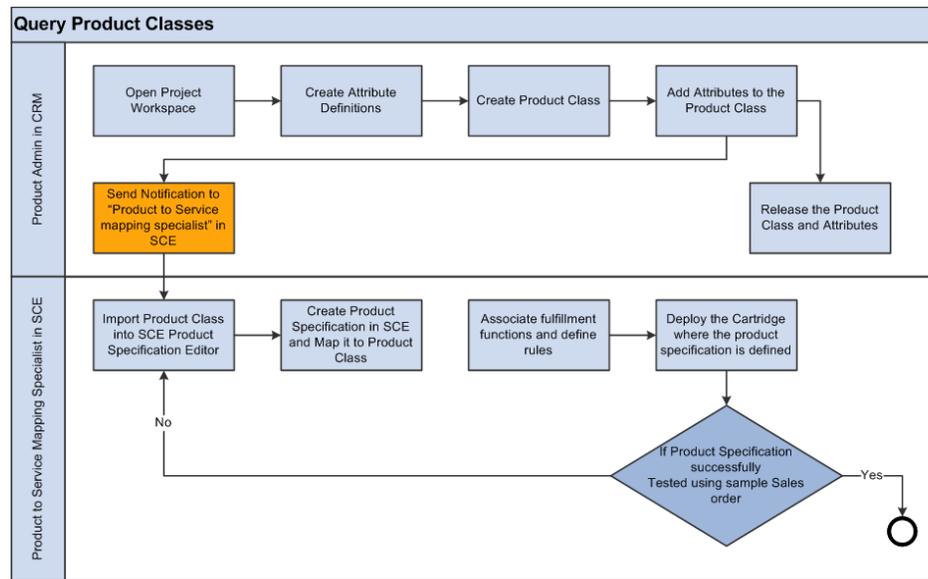
The product administrator performs the following operations in Workspace projects in the Siebel application:

1. Creates a new product class and transaction attributes and then associates attribute definitions to them.

2. Updates the product class by adding or deleting transaction attributes.

3. Updates the attribute definitions (value sets) by adding or removing values from them.

   For this step, select any class that has the transaction attribute that is associated with the valueset and send the notification.

For more information about creating classes, attributes, and attribute definitions in Siebel, see the *Siebel Product Administration Guide.*

Whenever these operations are performed, the Siebel product administrator notifies the product-to-service mapping specialist in SCE Studio. The product administrator can send the notification in multiple ways:

■ Use the notify menu function provided by Siebel. This function is provided in the Product Class UI in the Siebel application. The notification uses fixed templates to communicate the product class details and requires additional configuration in the Siebel application. One or more classes can be selected, and a single notification can be sent from Siebel.

   For more information about setting up the notification in Siebel, see the *Siebel Quick Fix Installation Guide,* "Email Notification for the Product Class" and "Attribute Query Feature."

■ Use a company's email application. This method provides additional flexibility to add more required information. This method is not dependent on a template and does not require additional configuration steps.

> **Note:** Regardless of the mechanism used to send the notification, you must include the product class name. The product-to-service mapping specialist uses the product class name to query in SCE Studio. If you use the notify menu function, the template adds the product class name.

The integration solution provides services that the product-to-service mapping specialist can use to query or import new or updated product classes from Siebel into SCE Studio.

The SCE product-to-service mapping specialist logs in to SCE Studio and queries or imports the product class and the transaction attribute details from Siebel into a cartridge. The import process uses the product class name. SCE Studio also offers the flexibility to query all the classes in the hierarchy associated with a product class. For more information about using SCE Studio, refer to the SCE documentation

For more information about using the SCE Studio, see the design studio section in the "OSM System Components" chapter of the *Oracle Communications Order and Service Management Concepts Guide*. See also the Design Studio online help.

After you import the product classes into a cartridge, the system stores all the transaction attributes and the associated valuesets in the SCE Studio data dictionary. Map the product class to a product specification in the cartridge. You can create a new product specification or map an existing product specification to the product class.

After you map the product class and the product specification, the product-to-service mapping specialist defines and associates fulfillment functions and their dependencies to the product specification. After defining the validation and decomposition rules, they deploy the cartridge to the studio environments.

SCE Studio provides a function that tests the design-time configuration after you submit a sample test order. If issues occur with the product specification in the cartridge, then you create a new product specification or import the product class again. If you create a new product specification, you must perform the configuration again. You must redeploy the cartridge for your changes to take effect. Oracle OSM uses the deployed cartridge to process order lines that are submitted for fulfillment.

6.2 Query Product Classes - Implementation Overview

Figure 6–2 illustrates how you query the product class and attributes from SCE:

Figure 6–2   Querying Product Class and Associated Attributes from SCE
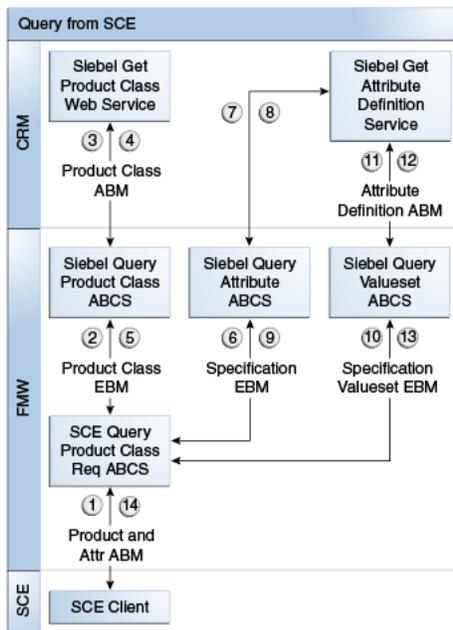
When this process initiates, the following events occur:

1.   The product-to-service mapping specialist must include all the necessary product class names in the query dialogue in SCE and invoke the query operation. The SCE Client calls the SCE Query Product Class Application Business Connector Service (ABCS) and provides one or more Class details as input. SCE must be configured to provide the input in the Product Class EBO structure that has only the Product Class names.

2.   The SCE Query Product Class ABCS provides the Product ClassEBM.

3.   The SCE Query Product Class ABCS routes the Product ClassEBM to Siebel Query Product Class ABCS.

4.   The Siebel Query Product Class ABCS transforms the Product ClassEBM, extracts the product class name, and invokes the Siebel getProductClass web service.

5.   The Siebel getProductClass web service returns the complete product class information for all input product classes to the Siebel query product class ABCS.

6.   The Siebel Query Product Class ABCS transforms the response message into the Product ClassEBM.

7.   The Siebel Query Product Class ABCS routes the Product ClassEBM to the SCE Query Product Class ABCS.

8.   The SCE Query Product Class ABCS identifies that the ProductClass has references to one or more attribute definition entities. It has to query all the attribute definitions associated with all of the product classes. It aggregates all of the attributes that are associated with all of the product classes. It provides all attribute IDs that are associated with all the product classes.

9.   The SCE Query Product Class ABCS routes the request to the Siebel Query Attribute ABCS.

10.   The Siebel Query Attribute ABCS invokes the getAttributeDefinition web service provided by Siebel.

11. The getAttributeDefinition service returns the complete attribute information for all attribute IDs to the ABCS.

12. The Siebel Query Attribute ABCS performs the transformation, filters the necessary fields, and constructs the AttributeEBM.

13. The Siebel Query Attribute ABCS routes the Attribute enterprise business message (EBM) back to the SCE Query Product Class ABCS.

14. The SCE Query Product Class ABCS identifies that the attributes have references to the valueset entity. It must query all of the valuesets associated with all the attributes. It aggregates all of the attributes that are associated with all the valuesets.

15. The SCE Query Product Class ABCS routes the request to the Siebel Query Valueset ABCS.

16. The Siebel Query Attribute ABCS invokes the getAttributeDefinition web service provided by Siebel.

17. The getAttributeDefinition service returns the complete attribute information to the ABCS.

18. The Siebel Query Valueset ABCS performs the transformation, filters the necessary fields, and constructs the ValuesetEBM.

19. The Siebel Query Valueset ABCS routes the ValuesetEBM back to the SCE Query Product Class ABCS.

20. The SCE Query Product Class ABCS composes the Product ClassEBM, the AttributeEBM, and the ValuesetEBM and returns it to the SCE Client application.

6.2.1 Logical Data Model in Siebel

Figure 6–3 illustrates the logical data model.

Figure 6–3 Logical Data Model of Product Class Structure in Siebel
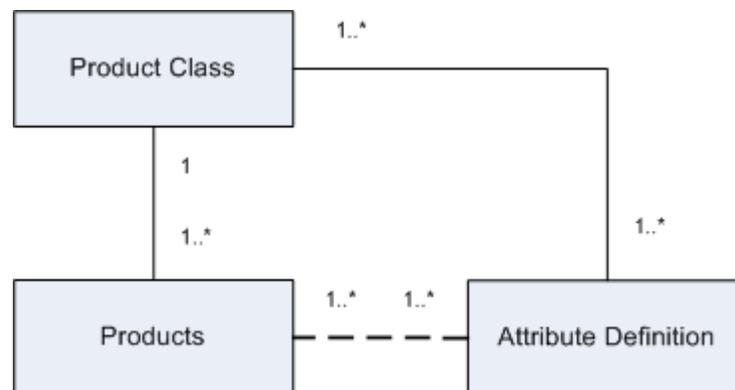
6.2.2 Updating the Attribute Valueset

The Siebel product administrator can update the attribute valueset associated with the attribute definition in the project workspace. This action automatically updates all the classes and their subclasses. The product administrator selects a class that is associated with the updated attribute definition and sends the notification. The product-to-service mapping specialist queries the product class from SCE Studio, which updates the corresponding valueset metadata in the data dictionary.

6.3 Support for Effectivity during Design-Time - Methodology

You can use product classes in Siebel (or the equivalent ICC in Product Hub) to represent unique product specifications. Product specifications represent a type of product offering and can only be sold through a product offering. Product specifications represent the unique entities that must be fulfilled. Not all product classes are created for this purpose. For the subset of product classes created to represent product specifications, at design time you must map the product class name and the corresponding fulfillment pattern name in Oracle OSM in its Central Order Management role. Consequently, changes to these product classes affects the mapping for both design time and run time order management handling. This section describes an important aspect of this mapping effectivity: how the time of a new mapping or a change to an existing mapping takes effect in coordination across Order Capture (Siebel) and Central Order Management (Oracle OSM).

References to product classes in this section are limited to those used to represent product specifications. To distinguish these, it is recommended to use a naming convention, such as ending the name with the term *ProdSpec*. You can create and update product classes directly in Siebel or from a Product Master, such as Oracle Product Hub for Communications. In this section, whenever you see Siebel design time product class changes, it is also applicable to product master when one is used.

Specify the effectivity for the product class in Siebel when:

- You create a new product class.

- You update an existing product class.

- You make an existing product class inactive.

Whenever these scenarios or any combination of these scenarios occur, you must query the product classes in the Design Studio, which is the design time tool for Oracle OSM from the Siebel application. After the product classes are successfully queried, product classes and the product specification are updated manually, and a following condition occurs:

- The product class is mapped to a new product specification. The new product specification definition involves defining the fulfillment metadata.

- The product class is mapped to an existing product specification.

- The product class mapping is changed to a different product specification.

Product class effectivity must be the same as the product specification effectivity. You update product specification effectivity manually in the Design Studio after the product class is successfully queried and the product specification is mapped to it.

When the mapping between the product class and product specification is updated (when the effectivity of the product class and product specification changes), the cartridge version in SCE Studio must be updated and the cartridge must be redeployed to the environment.

For more information about updating the cartridge version and the various deployment options, see the *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*

6.3.1 Deciding on Effectivity and Cartridge Deployment

Whenever the previously mentioned scenarios or their combination occur, you must create a new version of the cartridge and redeploy it. All orders the Oracle OSM system has submitted for processing uses the existing cartridge version. Any

subsequent new orders (including revision orders, follow-on orders, and change orders) or existing orders that are not yet submitted for processing by the Oracle OSM system uses the new version of the deployed cartridge. You should group product class changes and set effectivity for a date-time that is suitable for deploying a new cartridge version, such as an off-peak hour.

> **Note:** You deploy a new version of the cartridge only if you introduced a new mapping or changed an existing mapping.

The support for effectivity on the product class and the effectivity on the product specification is manifested in Oracle OSM by the effectivity of deployment of the cartridge. Effectivity is defined and controlled by the deployment dates for the cartridge. To manage effectivity across the applications, you must consider the following issues:

- The effective dates on the product class are the same as the deployment dates on the cartridge.

  In this case, the Siebel product administrator and the product-to-service mapping specialist in SCE Studio must reach a consensus on the effectivity dates of the product class and the deployment dates of the cartridge, respectively.

- The effective dates on the product class differ from the deployment dates on the cartridge.

  In this case, the deployment dates on the cartridge control effectivity. You can handle effectivity as follows:

  - Create a new product class or update an existing product class. Import the product class and define the mappings to the product specification in the cartridge. Deploy it based on when the changes must be applied to the order lines in the Oracle OSM fulfillment system.

  - Create a new product class; the mappings to the product specification are not yet done in the cartridge. If the product class is subsequently updated, Siebel queries the updated version of the product class and defines the mappings in the cartridge before deploying it. In this case, multiple versions of the product class may be defined with different effective dates and queried before the mappings are defined and the cartridge is deployed. Again, the deployment is based on when the changes must be applied to the order lines in the Oracle OSM fulfillment system.

  > **Note:** If you create a new product specification in the cartridge, you must configure the product specification before deploying the cartridge.

6.4 Query Product Classes Integration Flow

This integration flow uses the following interfaces:

- QueryProductClassAndAttributesSCECommsReqABCSImpl
- QueryClassificationListSiebelCommsProvABCSImpl
- QuerySpecificationListSiebelCommsProvABCSImpl
- QuerySpecificationValueSetListSiebelCommsProvABCSImpl

Figure 6–4 illustrates the query product classes integration flow.

Figure 6–4   Query Product Classes and Attributes Sequence Diagram
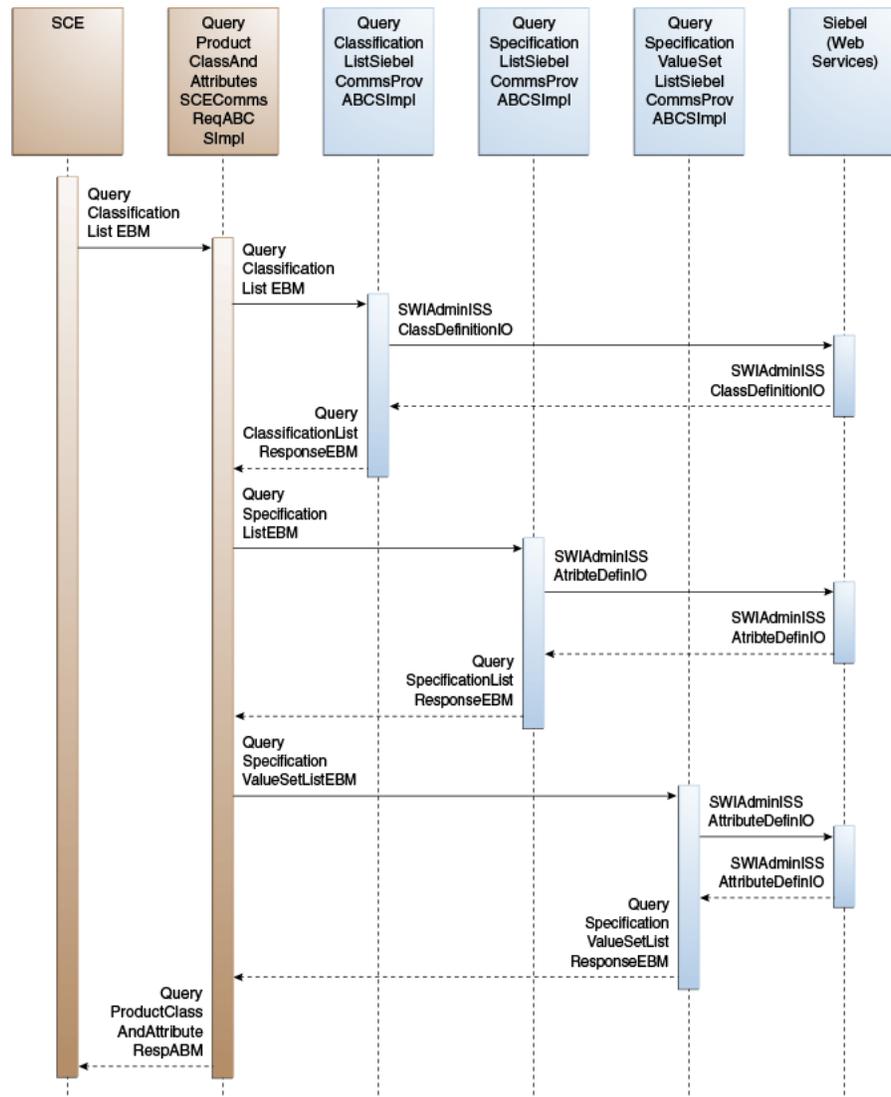
When this process initiates, the following events occur:

1. An SCE user (a product-to-service mapping specialist) triggers the Query Product Class and Attribute definitions integration flow using the SCE client, which provides the Class Code and the Oracle Fusion Middleware (FMW) URL. In the case of an update, SCE also provides the Class Codes for all of the subclasses.

2. The SCE invokes the QueryProdClassAndAttributesSCEReqCommsABCSImpl with the QueryClassificationListEBM, which contains the Product Class codes.

3. QueryProdClassAndAttributesSCEReqCommsABCSImpl passes through the QueryClassificationListEBM to the QueryClassificationListSiebelCommsProvABCSImpl.

4. QueryClassificationListSiebelCommsProvABCSImpl transforms the QueryClassificationListEBM into the Siebel Class Definition IO application business message (ABM), along with the Workspace Name from

AIAConfigurationProperties.xml file and invokes the Siebel GetProductClass web service.

5. The Siebel GetProductClass web service returns the complete product class information and associated Attribute IDs for all input product Class Codes. The QueryClassificationListSiebelCommsProvABCSImpl transforms the Siebel GetProductClassResponseABM into the QueryClassificationListRespEBM.

6. The QueryClassificationListRespEBM goes all the way back to the QueryProdClassAndAttributesSCEReqCommsABCSImpl.

7. Next, the QueryProdClassAndAttributesSCEReqCommsABCSImpl picks up the Attribute IDs from the QueryClassificationListRespEBM and maps these IDs to the QuerySpecificationListEBM and QuerySpecificationValueSetListEBM.

   One attribute may appear in multiple class definitions. While mapping Attribute IDs from the QueryClassificationListRespEBM to the QuerySpecificationListEBM and QuerySpecificationValueSetListEBM, you must take the union of these attribute definitions.

8. The QueryProdClassAndAttributesSCEReqCommsABCSImpl invokes the QuerySpecificationListSiebelCommsProvABCSImpl with the QuerySpecificationListEBM.

9. The QuerySpecificationListSiebelCommsProvABCSImpl transforms the QuerySpecificationListEBM into the Siebel Attribute definitions IO (ABM) along with the Workspace Name from the AIAConfigurationProperies.xml file and invokes the Siebel GetAttributeDefinition web service.

10. The GetAttributeDefinition service returns the complete attribute information for one or more attribute IDs. The QuerySpecificationListSiebelCommsProvABCSImpl transforms the response into the QuerySpecificationListRespEBM.

11. The QueryProdClassAndAttributesSCEReqCommsABCSImpl then invokes the QuerySpecificationValueSetListSiebelCommsProvABCSImpl with the QuerySpecificationValueSetListEBM. The QuerySpecificationValueSetListSiebelCommsProvABCSImpl transforms the QuerySpecificationValueSetListEBM into the Siebel Attribute definitions IO (ABM) along with the Workspace Name from the AIAConfigurationProperies.xml file and invokes the Siebel GetAttributeDefinition web service.

12. The GetAttributeDefinition service returns the complete attribute valueset information for one or more attribute IDs. The QuerySpecificationValueSetListSiebelCommsProvABCSImpl transforms the response into the QuerySpecificationValueSetListRespEBM.

13. The QuerySpecificationListRespEBM and QuerySpecificationValueSetListRespEBM go all the way back to the QueryProdClassAndAttributesSCEReqCommsABCSImpl. The QueryProdClassAndAttributesSCEReqCommsABCSImpl merges the QuerySpecificationListRespEBM, QuerySpecificationValueSetListRespEBM and the QueryClassificationListRespEBM and maps them to the QueryProdClassAndAttributesRespABM.

14. The QueryProdClassAndAttributesSCEReqCommsABCSImpl replies to SCE along with the QueryProdClassAndAttributesRespABM, which contains the class and attribute details for all the Class Codes provided by the product to the service mapping specialist.

6.5 Siebel CRM Interfaces

The process integration for Product Lifecycle Management uses these Siebel CRM interfaces:

- SWIAdminISSClassDefinitionIO: This web service is used to perform query operations on product class definitions.

- SWIAdminISSAttributeDefnIO: This web service is used to perform query operations on product attribute definitions.

For more information, see the *Siebel CRM Web Services Reference*, "Siebel CRM Primary Web Services."

6.6 Industry Oracle AIA Components

The integration uses these industry components:

- ClassificationEBO

- QueryClassificationListEBM

- QueryClassificationListResponseEBM

- SpecificationEBO

- QuerySpecificationListEBM

- QuerySpecificationListResponseEBM

- SpecificationValueSetEBO

- QuerySpecificationValueSetListEBM

- QuerySpecificationValueSetListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

6.7 Integration Services

These services are delivered:

- QueryProdClassAndAttributesSCEReqCommsABCSImpl with operation QueryProdClassAndAttributes

- QueryClassificationListSiebelCommsProvABCSImpl with operation QueryProductClass

- QuerySpecificationListSiebelCommsProvABCSImpl with operation QuerySpecificationList

- QuerySpecificationValueSetListSiebelCommsProvABCSImpl with operation QuerySpecificationValueSetList

Some of these services have been enabled to use Session Pool Manager.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

6.7.1 QueryProdClassAndAttributesSCEReqCommsABCSImpl

The QueryProdClassAndAttributesSCEReqCommsABCSImpl is a synchronous BPEL process and is the SCE requestor ABC implementation and performs all of the Product Class related actions like Create Product Class, Update Product Class, Query Product Class, and so on. This service follows all of the standards of a requester ABCS. This service has one operation: QueryProdClassAndAttributes. This accepts a QueryClassificationListEBM as a request and returns QueryProdClassAndAttributesRespABM as a response.

6.7.2 QueryClassificationListSiebelCommsProvABCSImpl

This is the Siebel Classification List Provider ABC Implementation. This service follows all the standards of a Provider ABCS implementation. This service has one operation: QueryProductClass.

6.7.3 QuerySpecificationListSiebelCommsProvABCSImpl

The QuerySpecificationListSiebelCommsProvABCSImpl is the Siebel attribute provider ABC implementation and performs all of the Specification List related actions like Query Specification List, Create Specification List, Update Specification List, and so on. This service follows all the standards of a provider ABCS implementation. This service has one operation: QuerySpecificationList.

6.7.4 QuerySpecificationValueSetListSiebelCommsProvABCSImpl

The QuerySpecificationValueSetListSiebelCommsProvABCSImpl is the Siebel attribute value set provider ABC implementation and performs all of the SpecificationValueSet List related actions like Query SpecificationValueSet List, Create SpecificationValueSet List, Update SpecificationValueSet List, and so on. This service follows all the standards of a provider ABCS implementation. This service has one operation: QuerySpecificationValueSetList.

7

Understanding the Process Integration for Order Lifecycle Management

This chapter provides an overview of the Order Lifecycle Management (OLM) integration process, discusses a typical topology and order capture flow. It describes both the Deliver and Qualify customer order subflows, and also design considerations for product definition and mapping.

This chapter includes the following sections:

- Section 7.1, "Order Lifecycle Management Overview"
- Section 7.2, "Typical Topology"
- Section 7.3, "Order Capture Overview"
- Section 7.4, "Describing the Deliver Customer Order Subflow"
- Section 7.5, "Describing the Qualify Customer Order Subflow"
- Section 7.6, "Product Definition and Mapping Design Considerations"
- Section 7.7, "Data Requirements"

7.1 Order Lifecycle Management Overview

The process integration for order lifecycle management (OLM) is at the core of business and operational support systems for any communications service provider (CSP). The process extends from the time a quote or order is created to the time when the goods and services are delivered and properly billed.

The Oracle Communications Order to Cash pre-built integration works with participating applications to accomplish this process as it relates to Customer Relationship Management (CRM), Order Management, Billing, and up to passing a customer order to Service Fulfillment (also commonly known as service fulfillment). Integration to other fulfillment system types such as supply chain management and workforce management can be added as an extension project at implementation time.

Figure 7–1 illustrates the functional flow.

Figure 7–1   Order Lifecycle Management Functional Flow
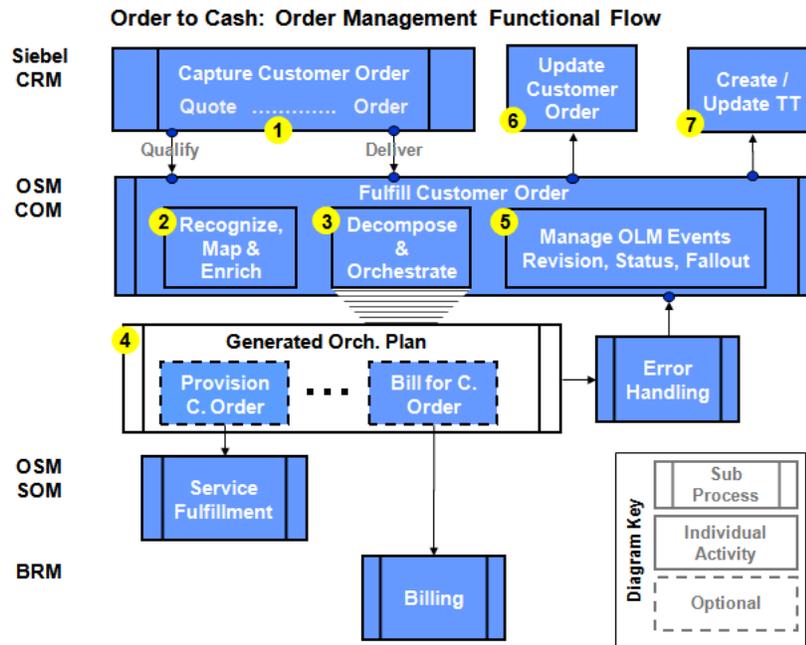
Here are the steps:

1. A customer order is captured in Siebel CRM. For some orders, the order may require technical qualification, such as validating that the network has enough capacity to offer the purchased products. After an order capture is complete and the order is validated in Siebel CRM, the system submits it to Oracle Order Service and Management Central Order Management (Oracle OSM COM) for delivery. The two arrows from Capture Customer Order to Fulfill Customer Order show the Qualify scenario and the Deliver scenario.

2. Customer orders (both *Qualify* and *Deliver* request types) received in Oracle OSM are first recognized (as Oracle AIA Customer Orders), mapped to fulfillment patterns, and enriched with fulfillment metadata.

3. Oracle OSM decomposes and orchestrates the customer order. Oracle OSM divides the order into suborders, called order components, which have cross-order components, cross-order lines, and cross-order dependencies that reflect the specific demands of the communications service provider (CSP).

4. The outcome is an order orchestration plan. The fulfillment flow that is produced orchestrates fulfillment requests to different fulfillment providers (such as fulfillment system instances or stacks) using preconfigured fulfillment functions, like sync customer, initiate and fulfill billing, and provision order. The Oracle OSM Order to Activate PIP cartridge product provides out of the box ready to use automatic integration to Oracle Application Integration Architecture (Oracle AIA) web services. When the Oracle Billing and Revenue Management (Oracle BRM) pre-built integration option is in use it takes the billing related requests (Sync Customer, Initiate and Fulfill Billing) made by Oracle OSM to Oracle AIA, from Oracle AIA to Oracle BRM. The Sync Customer Oracle AIA process integration also uses the Siebel pre-built integration option to get customer account details.

5. Oracle OSM manages OLM events. For cancel and revision requests, Oracle OSM generates and executes compensation plans to efficiently match a change. OLM manages order data and status updates, and order fallout.

6. Throughout the fulfillment process, Oracle OSM maps fulfillment function responses to common statuses, which are then aggregated into order line statuses and order header status values. The status management capability updates Siebel CRM with relevant customer status and milestone values. Oracle OSM updates Siebel CRM when order lines reach their point-of-no-return (PONR) to prevent the submission of new revisions. It also updates Siebel CRM with any enrichment to order lines that may have occurred during fulfillment.

   Errors may occur for many reasons. Oracle AIA reports such errors to Oracle OSM for fallout management. Additionally, validation logic in Oracle OSM may raise fallout incidents.

7. Oracle OSM detects, reports, and resolves order fulfillment fallout incidents such as system, validation, and fulfillment errors. The Oracle approach creates trouble tickets in Siebel CRM to take advantage of the rich notification, reporting, and management capabilities of Siebel CRM.

For more information about Oracle OSM, see the *Oracle Communications Order and Service Management Concepts Guide.*

> **Caution:** A PIP in Oracle OSM terms differs from a PIP in Oracle AIA terms. In Oracle OSM, a PIP in the name of a cartridge indicates that the cartridge is designed to work with Oracle AIA Communications Order to Cash PIP whereas in Oracle AIA, it is a collection of processes.

Oracle OSM delivers these pre-built cartridges for use with the Oracle Communications Order to Cash pre-built integration:

- CommunicationsSalesOrderFulfillmentPIP
- CommnicationsProvisioningOrderFulfillmentPIP
- OracleCgbuCommunicationsORPFalloutPIP
- SIFalloutPIP

Additionally, Oracle OSM provides an Oracle AIA Emulator, which you can use to emulate an order.

For more information about how to install and deploy the delivered cartridges and the emulator, see the *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture*.

> **Note:** The focus of this guide is the automated integration points among Siebel CRM, Oracle OSM COM, Oracle OSM Service Order Management (Oracle OSM SOM), and Billing. This guide does not cover process details within Oracle OSM SOM, for example, service design, assign, and activation.

7.1.1 Order Lifecycle Management Business Flows

The order lifecycle management process integration enables the following business flows:

Process Sales Order Fulfillment business flow:

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle OSM pre-built integration options.

- Submitting orders from Siebel CRM to Oracle OSM for order fulfillment processing.

For more information about the Process Sales Order Fulfillment business flow, see Chapter 8, "OLM - Understanding the Process Sales Order Fulfillment Business Flow."

Synchronize Fulfillment Order Billing Account business flow:

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM, Oracle OSM, and Oracle BRM pre-built integration options.

- Interfacing orders to create customer data in Oracle BRM

For more information about the Synchronize Fulfillment Order Billing Account business flow, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."

Bill Fulfillment Order business flow:

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM, Oracle OSM, and Oracle BRM pre-built integration options.

- Interfacing orders to create transaction data in Oracle BRM

For more information about the Bill Fulfillment Order business flow, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow."

Provision Order and Update Fulfillment Order business flows:

These business flows are enabled using the Oracle Communications Order to Cash Oracle OSM pre-built integration option.

- Provisioning orders in Oracle OSM SOM.

- Updating orders and statuses in Oracle OSM COM.

  You do this through explicit order updates coming from Oracle OSM SOM.

For more information about the Provision Order and Update Fulfillment Order business flows, see Chapter 14, "OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows."

Update Sales Order business flow:

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle OSM pre-built integration options.

- Sending order updates from Oracle OSM COM to Siebel CRM.

For more information about the Update Sales Order business flows, see business flow, see Chapter 16, "OLM - Understanding the Update Sales Order Business Flow."

> **Note:** Information about managing order fallout in Oracle OSM and creating trouble tickets in Siebel CRM is discussed in the Order Fallout Management (OFM) chapters.
>
> For information, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

7.2 Typical Topology

Traditionally, CSPs deployed stovepipe business support system (BSS) and operational support system (OSS) solutions with middleware-based custom order orchestration solutions. Deployment consolidation for cost savings, convergent bundling, and time-to-market demands are fostering increasingly complex requirements for the orchestration solution. These requirements include sophisticated order mapping, order decomposition, status composition, fallout management, changes to in-flight orders, future-dated orders, and cross order dependencies, among others. You cannot easily meet these requirements using middleware-based custom solutions.

However, Oracle and a large group of leading CSPs concluded that a prominent and distinct role exists for a commercial ready-to-use OLM solution. We recognize this concept as the Order Management solution responsible for central fulfillment functionality and therefore, the CFS (not to be confused with the SID abbreviation for Customer Facing Service) references throughout this document. Oracle OSM refers to this concept as Central Order Management.

Figure 7–2 illustrates a typical Oracle Communications Order to Cash deployment topology. The Order Management system is at the center of this topology.

Figure 7–2 Typical Oracle Communications Order to Cash Deployment Topology
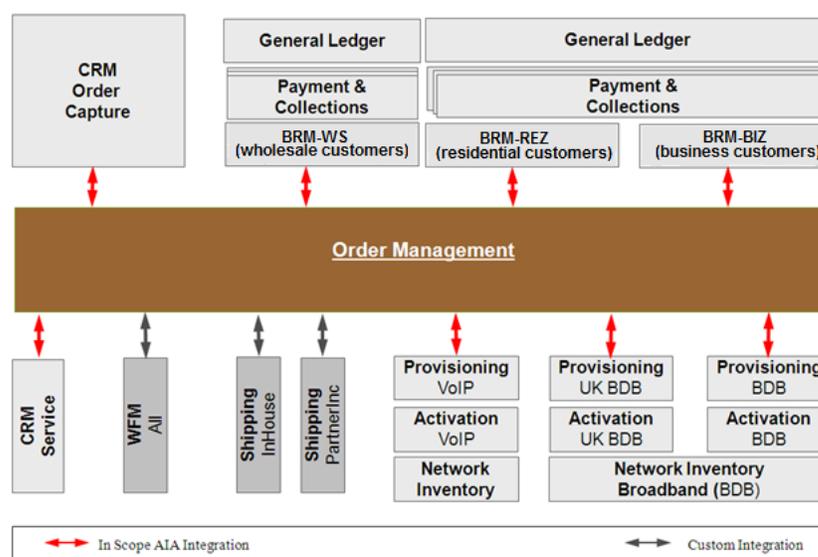
The topology shown is typical of most CSPs, although many could include more fulfillment system types (for example, billing or workforce management) and fulfillment system stacks. Order management is at the center of the Oracle Communications Order to Cash deployment, with order capture systems passing orders to the order management (OM) system. The OM system decomposes the order into suborders, each of which targets a particular fulfillment provider (that is, system instance) called order components. The topology shown uses three billing providers based on customer segment: wholesale, residential, and business. It uses three provisioning stacks based on service family and geography: VoIP, UK Broadband, and Broadband. It uses two shipping providers, one for in-house products and another for partner supplier products. Finally, it uses one workforce management provider and one separate Siebel CRM service provider (for trouble ticketing).

7.3 Order Capture Overview

Figure 7–3 illustrates a typical order capture flow. This flow varies by CSP and may vary by service family, customer segment, line of business, and other considerations. Two important integration points between Siebel CRM and OM are illustrated for a *Qualify* customer order and a *Deliver* customer order. In Siebel CRM, a customer order is known as a sales order. In general, order-based system interactions between different BSSs and OSSs require that order decomposition and orchestration go through the OM layer. For the Oracle Communications Order to Cash flow, at least two system interactions exist: *Qualify* customer order to validate the availability of a service design and the capacity to fulfill the customer order; and *Deliver* customer order to fulfill the products and services purchased by the customer or fulfill actions on existing customer assets.

Figure 7–3 Order Capture Sub-Flow
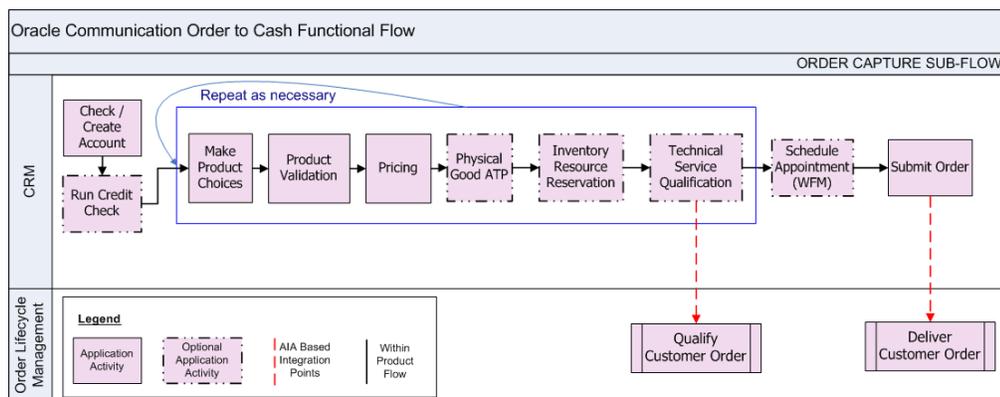
Figure 7–3 shows two swim lanes, one for Siebel CRM and another for OLM. Each swim lane includes the typical application activities and user interactions that are part of that application. Arrows between such activities represent the typical sequence of events within the same application. Arrows across swim lanes represent system interactions across applications. See the legend in Figure 7–3 for other details.

A typical flow starts with creating new customers and updating existing customer information. Depending on customer segment, line of business, or another consideration, you might capture customer information earlier, for example, when you create an opportunity or quote that you update during order capture. Depending on its business policy, some customers may pass through a credit check before starting the process of making product choices. While making product choices and at other points in the process, such as while capturing an order, the Siebel CRM system performs several validations. You price selected products and product options using relevant pricing logic. When physical goods are involved, the order capture process typically checks availability to purchase. For some services, resource reservation (for example, a phone number) also occurs during order capture. Before you submit an order and depending on the business practices of the CSP, the order may be required to pass a technical service qualification. Some CSPs also require scheduling an engineer (when needed) at the time of order capture to synchronize both the availability of an engineer and a customer. After completing an order and having it validated, you submit it to start the delivery process.

> **Caution:** The Siebel Copy Orders feature does not regenerate the identifiers (asset integration Id) that uniquely identify the customer purchases on the copied order. This makes the copied orders invalid to back-end systems. Therefore, copied orders are not supported by Oracle AIA. Instead of copying orders, It is recommended that you use the Siebel Favorites feature.

7.4 Describing the Deliver Customer Order Subflow

Figure 7–4 shows six swim lanes, one for each of the following applications: Siebel CRM, Oracle OSM, Oracle BRM, Provisioning, Network Inventory (Service and Resource Inventory), and Activation. Each swim lane includes the typical application activities and user interactions that are part of that application. Arrows between such activities represent the typical sequence of events within the same application. Arrows across swim lanes represent system interactions across applications. The O2C hexagons between swim lanes represent existing Oracle Communications Order to Cash pre-built integration points. See the legend in Figure 7–4 for other details.

Figure 7–4   Deliver Customer Order Subflow
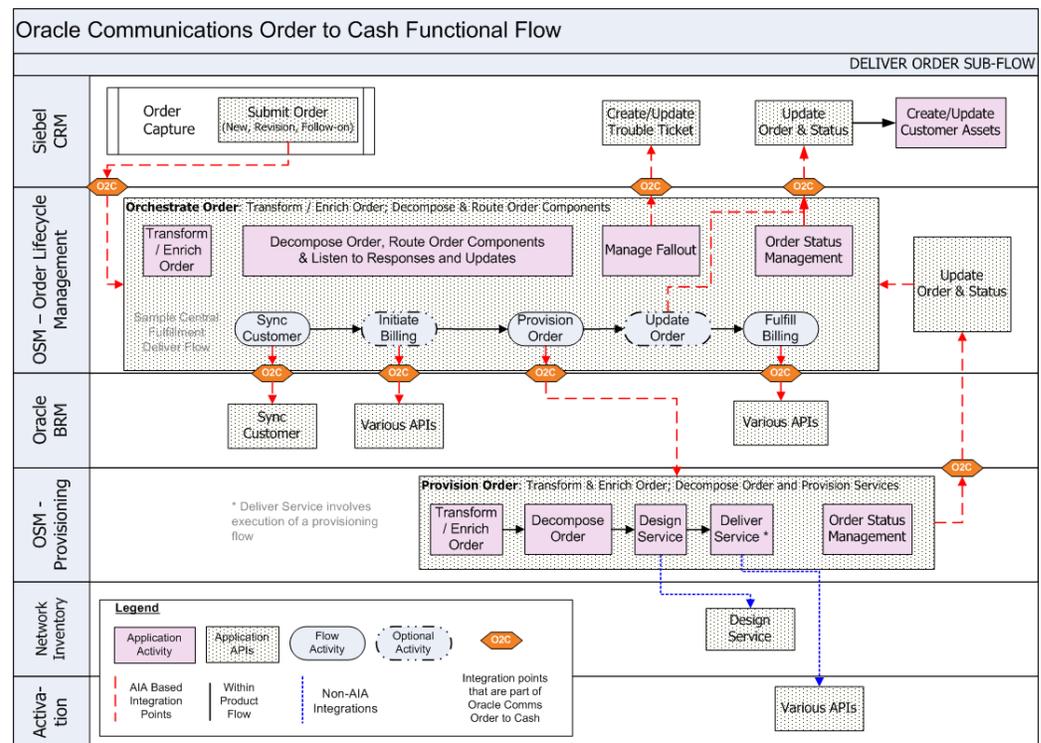
This flow starts with a new order, an order revision, future-dated order, or a follow-on order submitted from Siebel CRM to Oracle OSM. OM performs these key functions:

1. Transforms and enriches the order.

   It maps order lines to fulfillment flows and enriches it with fulfillment metadata and other relevant data.

2. Decomposes and routes the order.

It divides the order into suborders, which are called order components. Order components have cross-order components, cross-order lines, and cross-order dependencies that reflect the specific needs of the CSP. The outcome is an order orchestration plan that is executed at the computed fulfillment start time to meet the requested delivery date. Figure 7–4 illustrates a simple flow; however, the flow is typically more complex as shown in Figure 7–5. The produced fulfillment flow orchestrates fulfillment requests using preconfigured fulfillment functions, such as sync customer into Oracle BRM, initiate and fulfill billing, provision order, ship order, and install order. The Oracle OSM decompose and route order function also generates compensation plans that are associated with revision orders.

Figure 7–5   Complex Deliver Customer Order Subflow
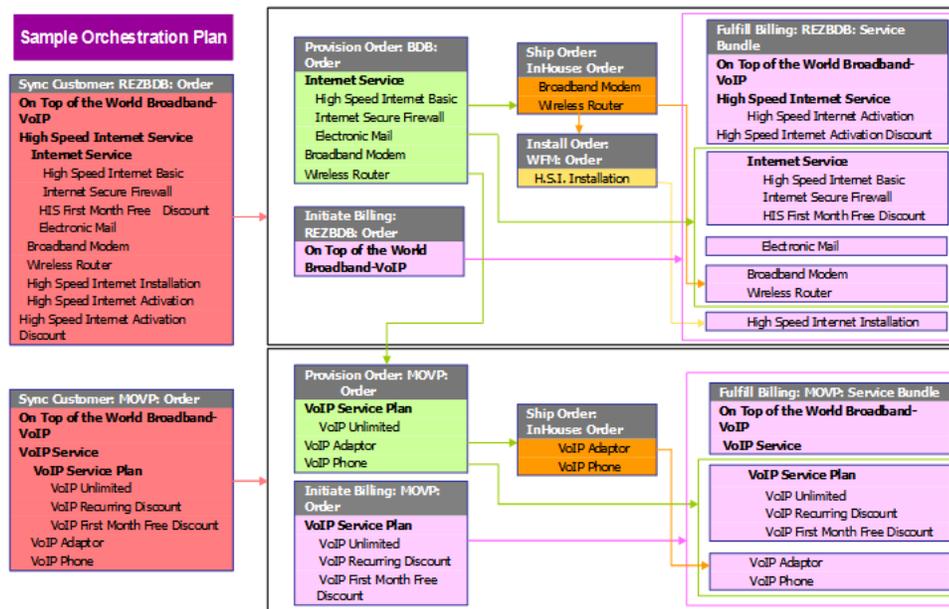
3. Manages fallout.

The integration provides for detection, reporting, and resolution of order fulfillment fallout conditions such as validation, and fulfillment errors. The Oracle approach is to create trouble tickets in Siebel CRM to take advantage of its rich notification, reporting, and management capabilities. System errors (such as an unreachable system), is handled differently.

For more information, see See Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

4. Manages status.

It maps fulfillment function responses to common statuses, which are then aggregated into order line statuses and order header status values. The status management capability updates Siebel CRM with relevant customer status and milestone values. It also updates Siebel CRM when order lines reach their PONR to prevent the submission of new revisions.

7.5 Describing the Qualify Customer Order Subflow

Figure 7–6 shows six swim lanes, one for each of the following applications: Siebel CRM, Oracle OSM, Oracle BRM, Provisioning, Network Inventory (Service and

Resource Inventory), and Activation. Each swim lane includes the typical application activities and user interactions that are part of that application. Arrows between such activities represent the typical sequence of events within the same application. Arrows across swim lanes represent system interactions across applications. The O2C hexagons between swim lanes represent existing Oracle Communications Order to Cash pre-built integration points. See the legend in Figure 7–6 for other details.

Figure 7–6   Qualify Customer Order Subflow
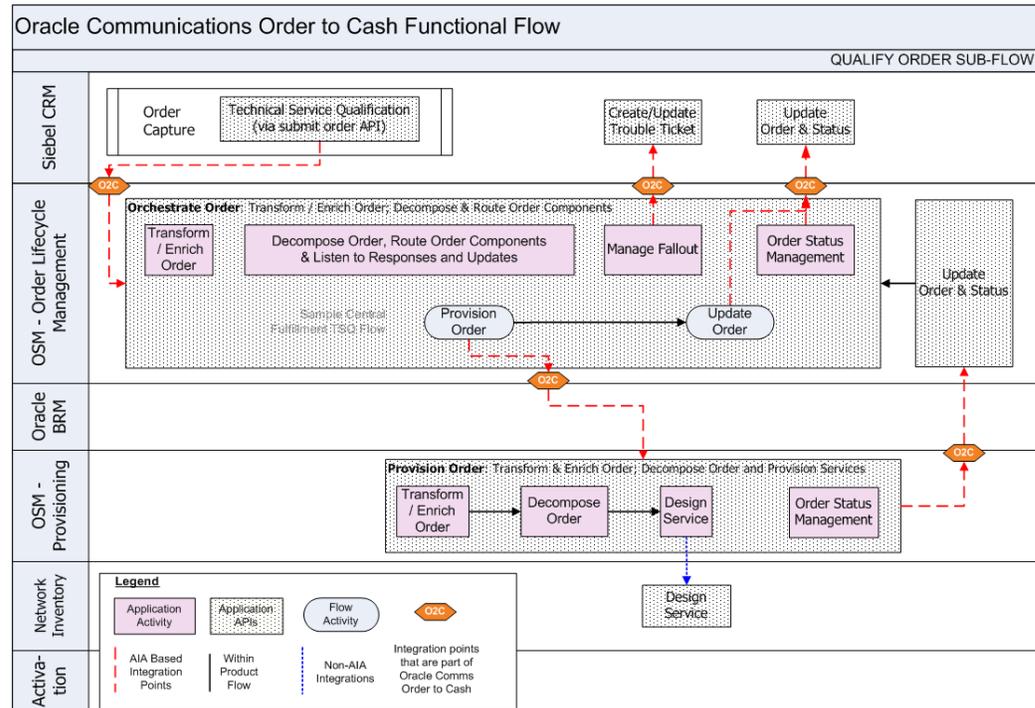
This flow starts with a request to qualify the technical validity of a customer order submitted from Siebel CRM to Oracle OSM. Oracle OSM performs the same four functions detailed for the *Deliver* customer order with one key distinction: the metadata used and the fulfillment flow produced is for qualifying the customer order rather than delivering the customer order. *Deliver* order flows and *Qualify* order flows produce different order and order line status updates.

7.6 Product Definition and Mapping Design Considerations

The product and service definition methodology has the greatest effect on time to market and on the cost of an Oracle Communications Order to Cash deployment. Often, CSPs define products and services in different departments to serve the best interests of individual departments. This approach creates a challenge for bridging the gaps at run time. A balanced approach that requires departments to make calculated compromises that result in simplified overall product life cycle and order life cycle business flows is recommended.

Figure 7–7 aligns with Tele Management Forum (TMF) terminology and guidelines.

Figure 7–7   Product Model
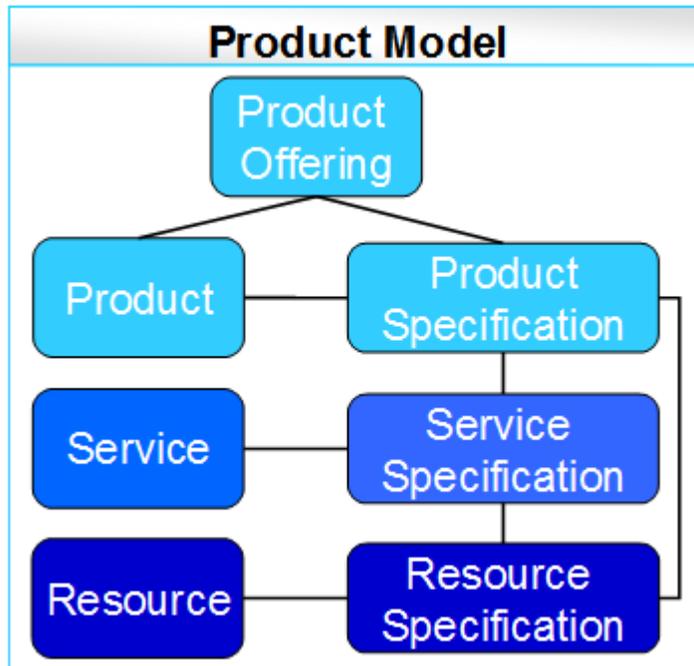
A balanced model produces a catalog with product specifications represented by the least number of entities. Product specifications represent unique capabilities with commercial value but only sold through product offerings. A more technical definition is that product specifications are types of products.

The product model shown covers the three TMF SID key entities: product, service, and resource.

Product offerings represent tangible and intangible goods and services made available for a certain price to the market in the form of product catalogs. Product offerings take one of three possible forms-simple offerings, bundled offerings, and promotional offerings:

- Simple offerings are product offerings of a single good or service.

- Bundled offerings are a grouping of two or more simple offerings into a single offer.

- Promotional offerings are time-bound, contract-bound, or discounted combinations of simple and bundled offerings.

A key element of the Oracle methodology is a one-to-one mapping of every order line to a product specification. This approach is key to achieving fast time-to-market and low-cost operations. The Oracle solution facilitates this mapping by associating product offerings with a product class in Siebel or Product Hub for Communications through the Fulfillment Item Code attribute.

OM acts on customer orders. Customer orders are composed of order lines. Each order line is represented by an action and a subject. Actions are verbs that represent the nature of the customer request, such as ADD to purchase an offering, UPDATE to modify a customer's subscription to an offering (for example, Customer Asset), and so on. A subject is the target of the action and can represent an offering, an asset, a discount, and so forth.

In the service fulfillment layer, a product specification can map to one or more technical services. A technical service is composed of one or more technical services and resources. The mapping from a customer order to a service order requires specific metadata modeled on products, product specifications, and service and resource configurations.

Figure 7–8 illustrates how the OM system takes advantage of the product model to map customer order lines to fulfillment flows according to the Oracle methodology. Other approaches may be plausible, but you must maintain a balanced approach that facilitates achieving the business objectives of fast time-to-market, and low-cost operations.

Figure 7–8   Mapping Customer Order Lines to Fulfillment Flows
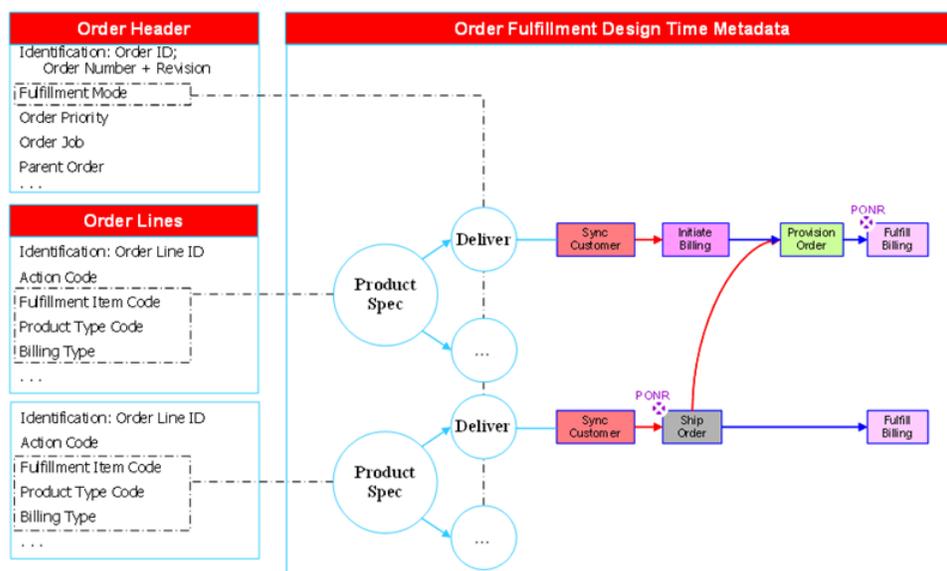
At run time, order capture copies key product offering attributes to each order line. These attributes include Fulfillment Item Code, Product Type Code, and Billing Type. OLM uses these attribute values to determine the corresponding product specification. The order header Fulfillment Mode attribute value determines the fulfillment requested type (for example, *Deliver* or *Qualify*). The intersection of a product specification and fulfillment request type determines the fulfillment actions and dependencies involved. When combined for all order lines in an order, an order fulfillment plan is generated dynamically.

7.7 Data Requirements

These are the data requirements for the OLM process integration. These apply to Siebel orders submitted for processing:

- An order must be of type Sales Order.

- The price list specified on the order must match the one created for the process integration for product lifecycle management. It is created in Siebel CRM and configured in the AIAConfigurationProperties.xml file.

- Service bundle lines or account-level product lines must have a service account, a billing account, and a billing profile.

- Service bundle lines and Simple Service bundle lines must have a service ID before they are interfaced to a billing system.

- Order lines referencing the same service account cannot reference different billing accounts. Refer to the solution constraint about having a single parent for subordinate accounts.

  For more information, see Section 12.8, "Solution Assumptions and Constraints."

- On any new order or change order for a service account, if the billing account is different from the billing account used on a previous order for the same service account, then all existing services paid for by the original billing account must appear on the order as updates to be paid by the new billing account.

- For Oracle OSM, Table 7–1 lists mandatory attributes:

Table 7–1   Oracle OSM Mandatory Attributes

Order Header EBO Attributes	Order Line EBO Attributes
Order ID	Line ID
Order Number	Base Line ID
Revision	Action Code
Fulfillment Mode	Product Name
Order Type	Product Type

> **Tip:**   The Sales Order enterprise business object (EBO) includes a vast set of attributes that are sufficient for most fulfillment systems, and it is extensible.

8

OLM - Understanding the Process Sales Order Fulfillment Business Flow

This chapter provides an overview of the Process Sales Order Fulfillment business flow and discusses order priorities and solution assumptions and constraints.

This chapter includes the following sections:

- Section 8.1, "Process Sales Order Fulfillment Overview"
- Section 8.2, "Supporting Order Priorities"
- Section 8.3, "Solution Assumptions and Constraints"

This business flow is enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM) and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

8.1 Process Sales Order Fulfillment Overview

The process integration for order lifecycle management (OLM) provides the following integration flow, which enables the Process Sales Order Fulfillment business flow.

- Submitting orders from Siebel CRM to Oracle OSM Central Order Management (COM) for order fulfillment processing.

A typical sales call center flow goes like this: a customer contacts a customer service representative (CSR) to place orders for new services or to make changes to existing services. The CSR must first determine whether the caller is an existing customer. If the customer is new, the CSR must set up an account for the customer before placing an order. If a customer is calling to change an existing service, the CSR can query the asset representing the customer's existing service and then use what is known in Siebel as asset-based ordering to modify or add to it. In this scenario, the CSR creates an order that references existing assets. When a CSR has captured an order, it is submitted for processing. Alternative sales channels follow a similar pattern.

In Siebel CRM, the submit order event enqueues the Siebel order message (Siebel order ABM or Application Business Message) in a Java Message Service (JMS) queue. After Siebel drops the message in the queue, the control is given back to the CSR, making the submit order event an asynchronous process. A JMS Consumer that listens to this queue, dequeues the message, and then invokes the Siebel Application Business Connector Service (ABCS).

Oracle OSM recognizes four kinds of customer orders:

- New orders:

These are orders for new purchases or changes to delivered products. Products that have been delivered are known as customer assets.

- Revision orders:

These are changed versions of orders that are still in fulfillment also referred to as in-flight orders. You can submit revision orders to fulfillment while the revised order (also known as base order) is in a fulfillment state that allows for order changes.

- Follow-on orders:

These are orders that have a fulfillment completion dependency on other orders.

- Future-dated orders:

These are orders that have a time-based dependency for the start of the fulfillment flow.

8.1.1 New Orders

New orders include first time purchases and changes to existing (asseted) service subscriptions and products. Siebel Order Capture captures new orders and submits it to Oracle OSM COM to deliver on the promises made to the customer.

Sales orders are primarily composed of two key parts: the order header and the order line. The order header includes attributes applicable to the customer and to all order lines. Order lines are composed of an action and a subject.

Order lines can include any combination of order line actions supported in Siebel CRM. Possible order line actions are:

- Add
- Delete
- Update
- Suspend
- Resume
- Move-Delete
- Move-Add
- Existing (no change is required)

Order lines can include a variety of subjects, including but not limited to simple product offerings, discounts (modeled as simple product offerings), bundled product offerings, promotional product offering, and pricing event products (used with multi-event billing products).

The key function of the Oracle Application Integration Architecture (Oracle AIA) integration is to pass enough order header and order line attributes to facilitate order fulfillment and to establish the necessary cross-references.

Notice that an order in Siebel may be revised several times before it is submitted for fulfillment for the first-time; all such revisions are only internal to Siebel such that each revision supersedes the prior revision completely and for Oracle OSM these do not count as revision orders.

8.1.2 Revision Orders

The fulfillment of some services may take days and weeks, and some business-to-business (B2B) and infrastructure projects may take months to complete.

During this period, customers change their minds and request changes to their orders, which then become revision orders in Siebel CRM. In many cases, continuing the base order when a revision is submitted is costly for the communications service provider (CSP), and sometimes the operation cannot be fully undone. For these reasons, support for revision orders provides the following benefits:

- Enhances customer satisfaction by allowing customers to change their orders within an agreed-upon limit.

- Reduces the costs associated with fulfilling unwanted goods and service requests and wasting system capacity, nonrecoverable resources, acquired stock, and so on.

- Reduces human intervention to manually retrofit data records when recovery cannot be automated.

Revision orders are changes made to a previously submitted order. Siebel CRM allows users to revise an order line if the order line has not reach the point-of-no-return (PONR) or complete. A PONR is configured on the fulfillment flow of each product specification in Oracle OSM and is propagated to Siebel CRM to indicate that an order line cannot be revised beyond that point in time. Not all revisions are submitted to fulfillment; only submitted revisions factor into fulfillment.

To avoid problems associated with stale revisions (that is, revisions that do not progress in Siebel CRM and become out of sync with their underlying asset); Siebel allows only one pending revision for each order.

After a revision is submitted, Oracle OSM Order Change Management (Oracle OSM OCM) takes three actions:

1. Suspends the fulfillment flows associated with the revised order.

2. Computes the delta changes for each order line.

3. Leverages the metadata configured for the flow to devise a compensation plan for fulfillment activities that have occurred and that are affected by the revision. The compensation plan is woven into the fulfillment plan for the revision order, and the revision fulfillment does not begin until completion or another revision is submitted.

In Siebel CRM, for the sales order that is to be revised, a CSR navigates to the Sales Order screen, revises a base order, makes the required changes, and then submits the revision.

8.1.3 Follow-On Orders

As mentioned previously with revision orders, the fulfillment of some services may take days and weeks, and some B2B and infrastructure projects may take months to complete. During this period, customers change their minds and request order changes that become revision orders in Siebel CRM if the subject order lines did not reach the PONR or otherwise become follow-on orders. In many cases, not taking an order pending the completion of in-flight orders is not acceptable; therefore, Siebel simulates the future state of in-flight orders and allows for the creation and submittal of follow-on orders that are nothing more than change orders based on the projected future state of a customer's assets.

Follow-on orders are change orders that involve a dependency on the future fulfillment of at least one other order line in an order that is currently in flight. The follow-on order line may change another in-flight order line that is beyond the hard PONR or that depends on the future asset state of that line, as through an explicit dependency established in Siebel CRM.

Follow-on orders are created and submitted to Oracle OSM immediately, and Oracle OSM provides for managing the fulfillment dependency between the follow-on order and other base orders. This responsibility is similar to the responsibility for determining the correct processing time for future-dated orders.

In Siebel CRM, a CSR navigates to the Sales Order screen (for the sales order that is supposed to undergo follow-on), and creates and submits the follow-on order.

After the follow-on order start-fulfillment dependencies are resolved, the follow-on order becomes like any other change order. It is also subject to revisions and other follow-on orders.

8.1.4 Future-Dated Orders

A variety of reasons require a CSP to take or place an order with a future-requested delivery date. Future-dated orders are submitted immediately to Oracle OSM when they are ready. Oracle OSM is responsible for computing the fulfillment start date-time.

When a CSR receives a request from the customer to submit an order on a future date, they set the Due Date attribute to the specified date before submitting the order.

For more information about handling current, past, future, and requested but not provided delivery date-time values, see the *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*

Avoid creating multiple future-dated orders against the same asset because they create a complex future asset state that is difficult for both the CSR and the customer to comprehend. We recommend that only a trained CSR be allowed to enter multiple future-dated orders against the same asset and only when required. When introducing an order line against the same asset with a Requested Delivery Date sooner than another created order, you must revise the latter to ensure that the order is based on an updated future state of the asset.

8.2 Supporting Order Priorities

Order fulfillment priority is specified in Siebel CRM and honored by message queues, Oracle AIA, and Oracle OSM unless data integrity dictates a different processing sequence, such as with update sales orders from Oracle OSM to Siebel CRM.

Order priority affects the sequence in which orders are picked up from queues and processed in Oracle AIA and Oracle OSM. Orders with a higher priority take precedence over orders with a lower priority that have not yet started fulfillment.

Order priorities work as follows:

1. The submission process for orders is the same for new orders, revision orders, and follow-on orders. The CSR selects a priority for the order when they submit it.

2. As delivered, Siebel provides and maps these priority values:

Table 8–1   Order Priority Values

Order Priority	JMS Priority
Low	3
Medium	5
High	7
Urgent	9

The integration supports 10 priority values, 0-9, as dictated by JMS queuing technology. Implementers can extend Siebel to support priority values other than the four that are supported when delivered.

8.3 Solution Assumptions and Constraints

These are the solution assumptions and constraints for this business flow.

1. Service points in Siebel are implemented as assets and are typically uploaded into Siebel from external sources. Ideally, service points are mastered in a common place and shared between Siebel CRM and Network Inventory (Service and Resource Inventory). The integration assumes that at least one following statement is true:

    ■ The determination of service point in Siebel CRM is irrelevant to Service and Resource Inventory.

    ■ The determination of service point in Siebel CRM is replicated in Service and Resource Inventory (for example, the same result is achieved).

    ■ The service point attribute value is unique and common across Siebel and Service and Resource Inventory, such that Service and Resource Inventory can use the value directly.

    ■ The service point attribute value is a cross-reference that is understood by Service and Resource Inventory; no Oracle AIA cross-reference exists for this attribute.

2. In Siebel CRM, order revisions are created as a copy of the previous revision and then changes are made to the revision. When created, the first order reflects the customer assets at the time. Revisions sometimes stay for a long period in Siebel CRM without submittal and may become stale if the customer assets change in the interim. The expectation for Siebel CRM is that it ensures that the revision order data is up to date with the customer assets at the time the order is submitted. Any customization of Siebel CRM or integration to a different CRM system must ensure that revision orders are brought up to date with the customer assets state before submitting the order to Oracle OSM.

3. Multiple future-dated orders require special care from the CSR to ensure that orders are submitted in the correct sequence and that new orders do not invalidate formerly submitted orders. We recommend that providers limit future orders to one per customer.

4. Follow-on orders, if submitted before base orders, are processed as base orders. CSRs must make sure they submit base orders first for the follow-on orders dependency on base orders to take effect in Oracle OSM.

5. Mixing future-dated, follow-on, and revision orders requires a well-trained CSR because some scenarios could produce unintended results. Ensure that:

    ■ You create follow-on events only when base orders are past the PONR.

    ■ You create and submit revisions as soon as they are firm; when revisions are pending, you do not create follow-on orders before you discard pending (not submitted) revisions.

    ■ You can create future-dated orders against the same asset if you create them in chronological order.

6. Siebel CRM does not guarantee correct assets if follow-on orders are created before modified order lines reach the PONR. You should create follow-on orders only

after modified order lines reach the PONR and any pending revisions are discarded.

7. Siebel CRM can capture revisions to order Due Date in Siebel CRM (Requested Delivery Date in Oracle AIA) and submit them to Oracle OSM.

8. Revising the requested delivery date for an order only affects Oracle OSM if the base order did not start fulfillment by the time the revision was received in Oracle OSM.

9. While in Siebel CRM, you can create an Oracle AIA follow-on order even before an order reaches the PONR. Oracle OSM only accepts follow-on orders when the base order is past the PONR.

10. Oracle OSM does not support revisions to base orders with follow-on orders.

For more information, see the *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture.*

9

OLM - Process Sales Order Fulfillment Business Flow: Implementation

This chapter provides an overview of the Process Sales Order Fulfillment business flow and discusses the implementation of Siebel Customer Relationship Management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 9.1, "Process Sales Order Fulfillment Business Flow Overview"
- Section 9.2, "Submitting Orders from Siebel CRM to Oracle OSM Integration Flow"
- Section 9.4, "Siebel CRM Interfaces"
- Section 9.5, "Industry Oracle AIA Components"
- Section 9.6, "Integration Services"

9.1 Process Sales Order Fulfillment Business Flow Overview

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

> **Note:** If you have deployed Oracle Communications Order to Cash Siebel CRM and Oracle BRM options, *but not the Oracle OSM option*, a Test Orchestration Process (TOP) is shipped to sanity test the out-of-the-box (OOTB) order flow. This must be replaced by your own order management system.
>
> For more information, see *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*, "Configuring and Deploying the Comms Order to Cash: SBL CRM and BRM Pre-Built Integration," Replacing Test Order Orchestration with your Order Management System.

The following integration flow involves submitting orders to Oracle OSM:

- Submitting orders from Siebel CRM to Oracle OSM

9.2 Submitting Orders from Siebel CRM to Oracle OSM Integration Flow

This integration uses the following interfaces:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer

- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

Figure 9–1 illustrates an order submitted from Siebel CRM to Oracle OSM Central Fulfillment System (CFS) using Oracle AIA.

Figure 9–1   Siebel CRM to Oracle OSM CFS Sequence Diagram
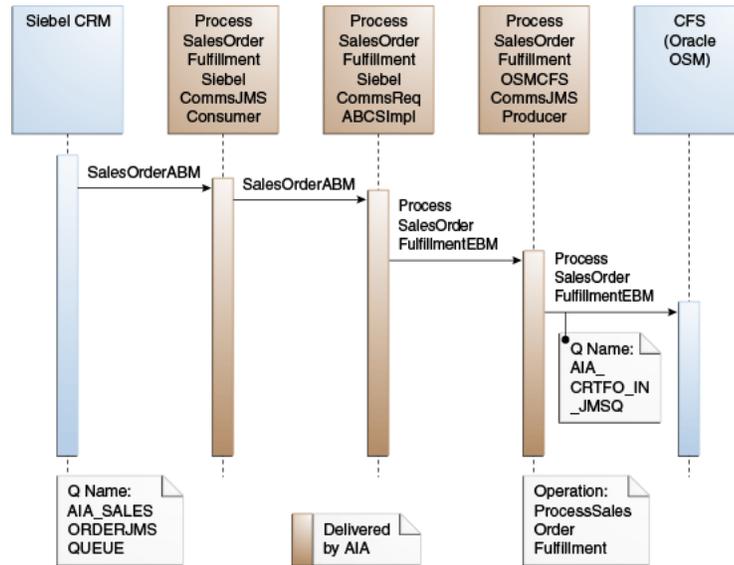
When a new order process is initiated, the following events occur:

1. In Siebel CRM, a user navigates to the Sales Order screen and clicks **Submit** to submit a new order. This action triggers Siebel to create a Siebel application business message (ABM) with all the captured details. Siebel then enqueues this ABM in the AIA_SALESORDERJMSQUEUE JMS Queue.

2. After the Siebel ABM is enqueued in the AIA_SALESORDERJMSQUEUE, the ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer dequeues the message and passes it on to the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl. The ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer has a JMS adapter service that polls for any messages in the AIA_SALESORDERJMSQUEUE.

3. The ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl invokes the transformation logic to transform Siebel ABM to the ProcessSalesOrderFulfillmentEBM, and routes the message to the ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer.

   The Siebel order structure does not support multiple charge types for a single order line, while the order EBO structure does. For this reason, the order lines referencing a Complex Product (CP) of billing type *Subscription* and its component products of billing type *Event*, representing a multi-event billing product, are transformed into a single EBO order line referencing a product with multiple charge types.

4. The ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer interface wraps ProcessSalesOrderFulfillmentEBM into CreateOrder message format and enqueues the message into the AIA_CRTFO_IN_JMSQ.

The CreateOrder message is then wrapped in a SOAP envelope. Oracle OSM expects the following additional JMS properties to be set when the JMS payload:

- JMSPriority

- URI -- /osm/wsapi

- _wls_mimehdrContent_Type -- text/xml; charset=utf-8

5. Once the message is dropped to the Queue, the store and forward (SAF) mechanism forwards the message from Oracle AIA WebLogic to Oracle OSM WebLogic.

   Oracle OSM picks up the CreateOrder message and then further decomposes the order for fulfillment and provisioning.

For more information about how Oracle OSM processes the order, see Chapter 7, "Understanding the Process Integration for Order Lifecycle Management."

9.2.1 Defining Transaction Boundaries and Recovery Details

For this flow, there is one transaction boundary. Table 9–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

If order submission from Siebel causes a system or business error, any further order to the account does not get processed until the error is fixed. All order submissions for that account are locked in the sequencer table. If the error is a business error then the message must be removed from the sequencer table and if the error is a system error then the message must be resubmitted.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer

- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

Table 9–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer passes the Siebel message to ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl, which invokes transformation logic to convert that Siebel message into an EBM. The EBM is then routed to ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer.	AIA cross-reference entries.	Rollback cross-reference transactions. Rollback JMS message to AIA_SALESORDERJMSQUEUE_ErrorQ	Resubmit the order from either AIA_SALESORDERJMSQUEUE_ErrorQ or from Siebel.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

9.3 Supporting Order Priorities

Customers can add other order priority values in Siebel CRM. Additionally, customers can use the SWI_ORDER_JMS_PRIORITY mapping, which maps these string values to integers.

You are required to set up some JMS compatibility properties on the Siebel queue and to make manual changes to seeded priority values.

For more information about priority values, see the *Siebel Order Management Guide Addendum for Communications, Employee Asset-Based Ordering*, "Modifying the Order Priority Mapping."

The integration supports 10 priority values, 0-9, as dictated by JMS queuing technology. Implementers can extend Siebel to support priority values other than the four that are supported when delivered.

For more information, see the Siebel product documentation for Lists of Values.

These steps describe how the integration handles order priorities:

1. When the order is submitted, Siebel sets the JMS Priority message in the JMS headers and also populates the following field: ListOfSWIOrderIO/SWIOrder/OrderPriority.

2. Based on the priority, the ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer consumes the message and routes it to the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl. This process does a lookup on the SALESORDER_PRIORITY domain value map (DVM) and populates the ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/FulfillmentPriorityCode.

3. The ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer looks for the FulfillmentPriorityCode and does a lookup on SALESORDER_PRIORITY DVM for the JMS column and populates the priority in the JMS Headers in JMSPriority field.

4. The store and forward (SAF) mechanism honors the JMS priority and picks up the message with high priority first and passes it to Oracle OSM CFS.

5. Oracle OSM CFS and Oracle OSM Provisioning honor the priority through internal mechanisms. Higher priority orders are fulfilled and provisioned first, followed by lower priority orders.

6. Oracle OSM is expected to maintain the priority of the orders and must populate the FulfillmentPriorityCode element in all outbound messages.

7. The population of JMS priority from the FulfillmentPriorityCode is done using the BPEL assign activity as follows:

Look up the DVM column for priority value and copy it to the JMSPriority field in JMS headers:

```
<assign>
            <copy>
                <from
expression="orcl:lookup-dvm('oramds:/apps/AIAMetaData/dvm/SALESORDER_
PRIORITY.dvm','COMMON',bpws:getVariableData('priority_value'),'JMS',null)"/>
                <to variable="msg_priority"/>
            </copy>
            <copy>
                <from variable="msg_priority"/>
```

```
          <to variable="jmsHeaders" part="outboundHeader"

query="/ns5:JMSOutboundHeadersAndProperties/ns5:JMSOutboundHeaders/ns5:JMSPrior
ity"/>
          </copy>
      </assign>
```

- jmsHeaders is a variable which is of type JMSOutboundHeadersAndProperties.

- msg_priority is a string variable.

The ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer follows this logic to populate the JMSPriority.

Other producers need not do a DVM lookup since the priority value is an integer and is directly populated in the EBMs in the FulfillmentPriorityCode. The integration can use this value to populate the JMS priority. Oracle OSM and Oracle AIA, unlike Siebel, follow the same values for JMS priorities.

9.4 Siebel CRM Interfaces

This integration flow uses the following Siebel interface:

- SISOMBillingSubmitOrderWebService

This is the outbound Siebel web service used to submit orders.

For more information about web services, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

9.5 Industry Oracle AIA Components

These integration flows use these industry components:

- SalesOrderEBO

- ProcessSalesOrderFulfillmentEBM

The industry enterprise business object (EBO) and EBM XSD files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry EBS WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

9.6 Integration Services

The following services are delivered with these integration flows:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

9.6.1 ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer

The ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer is implemented as a Mediator process.

This consumer listens over the AIA_SALESORDERJMSQUEUE into which Siebel enqueues the simple object access protocol (SOAP)-wrapped Siebel Order application business message (ABM). This consumer dequeues the messages from this queue, unwraps the message from the SOAP envelope, and routes the Siebel ABM to the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl.

9.6.2 ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

The ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl is implemented as a business process execution language (BPEL) process with a single operation: Initiate.

This service is invoked when an order is submitted in the Siebel application. This service is the Siebel ABCS implementation, which converts the Siebel ABM into the Sales Order EBM. The service looks up the cross-reference values for the customer account ID, billing profile ID, pay profile ID, and product ID to find common IDs to appropriately populate the Sales Order EBM. In the case of promotions and service bundles, if the cross-reference values are not present, new cross-reference values are created.

This service also creates the requisite cross-reference values for the order ID, order line ID, installed product ID, account ID, bill profile ID, pay profile ID, contact ID, and address ID between Siebel values and generated common values.

9.6.3 ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

The ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer is a BPEL process that has a JMS Adapter Service, which enqueues the message ProcessSalesOrderFulfillmentEBM into AIA_CRTFO_IN_JMSQ after wrapping it into SOAP envelope (for putting WS-security information for Oracle OSM) and Oracle OSM's CreateOrder envelope.

This service has one operation: ProcessSalesOrderFulfillment. It takes the

ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducerRequestMessage as input.

10

OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow

This chapter provides an overview of the Synchronize Fulfillment Order Billing Account business flow and discusses solution assumptions and constraints.

This chapter includes the following section:

- Section 10.1, "Synchronize Fulfillment Order Billing Account Overview"
- Section 10.2, "Solution Assumptions and Constraints"

This business flow is enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM), Oracle Order and Service Management (Oracle OSM), and Oracle Billing and Revenue Management (Oracle BRM) pre-built integration options.

10.1 Synchronize Fulfillment Order Billing Account Overview

Communications service providers (CSPs) do not want to overburden Oracle BRM with all of the customer information in their Siebel CRM system. Instead, they want the ability to create the necessary customer data in Oracle BRM only as it is needed; that is as part of the order fulfillment process.

To Synchronize Fulfillment Order Billing Account(s), the process integration for order lifecycle management (OLM) provides the following service:

- CommsProcessFulfillmentOrderBillingAccountListEBF: This creates customer data in Oracle BRM when called as part of the order fulfillment process.

This service takes an order as input and collates order data and then calls other enterprise billing services (from the Customer Management Process Integration) to create accounts and their components (such as billing preferences and payment methods) referenced on an order in a target Oracle BRM instance. This service can be invoked from an order orchestration flow from within an order management system, such as Oracle OSM, to create customer data in Oracle BRM.

For more information about calling this service, see Appendix H, "Expectations from a COM System for Billing Integration."

For more information about the Customer Management process integration, see Chapter 18, "Understanding the Process Integration for Customer Management."

For more information, see Appendix C, "OLM - Mapping Billing Dates" and Appendix E, "OLM - Examples of Changing the Paying Parent on Subordinate Accounts."

Figure 10–1 shows the order interface to customer data in Oracle BRM.

Figure 10–1   Synchronize Fulfillment Order Billing Account - Interfacing Customer Data to Billing
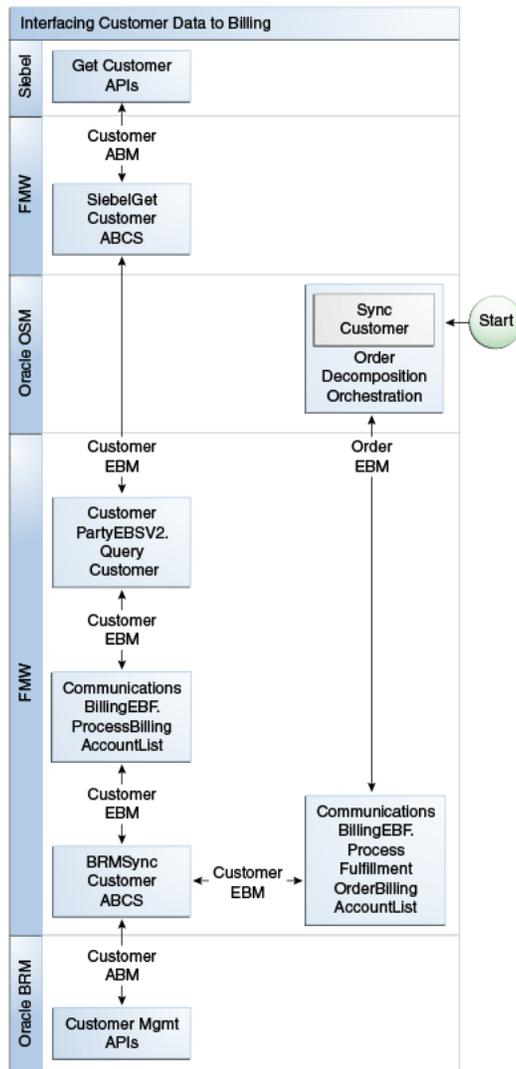
The CommsProcessFulfillmentOrderBillingAccountListEBF service processes *only* lines with the actions of ADD, UPDATE, and MOVE-ADD and ignores the others. This service considers the following kinds of order lines for customer data collation:

- For lines whose billing type is *Service Bundle, Item, Subscription*, or *Discount*, it considers Service Account, Billing Account, and Billing Profile.

- For lines whose product type is *Promotion*, it includes only Billing Account.

- All other lines are ignored.

The result of calling this service is the creation of customer data such as accounts, bill-infos, and pay-infos in Oracle BRM.

Customer creation that occurs in Oracle BRM as part of order fulfillment using the service InterfaceOrderToCustomerEBF cannot be undone:

- The service does not support the ability to inactivate or delete accounts, bill-infos, or pay-infos in Oracle BRM.

- Calling the CommsProcessFulfillmentOrderBillingAccountListEBF again with the same input as before has no effect.

- If the service is called with references to different customer data than before, the service detects the delta and creates just the account, bill-infos, and pay-infos that do not exist in Oracle BRM.

When the service account on a service bundle or account-level product line is different from the bill-to account, the service account is created as a nonpaying subordinate account under the bill-to account in Oracle BRM; that is, it results in the creation of a paying hierarchy in billing:

- A paying hierarchy, when created, cannot be undone simply by the cancellation of the original Siebel CRM order.

- If the service is called to update an existing paying hierarchy (for example, to set the paying account for a subordinate account to a different paying account), to undo that update (because the Siebel CRM order requesting the change was canceled), the order management system must rework the message such that it is a call to update the hierarchy to a previous state.

For information about what Siebel account information is sent to Oracle BRM see Section 18.4.1, "Create/Sync Account Integration Flow."

Table 10–1 summarizes what is expected from the order management system (that is calling this service) in terms of action on the line. *Oracle OSM and OSM AIA cartridges obey these expectations*.

Table 10–1   Actions on Order Line Expectations Summary

Original Action on Order Line	Is this the first time the order line is being processed by customer sync or is it a revision?	What is occurring on the revision (that is relevant to customer sync)	Expected Action on compensation order line, set by Order Management	Comments
ADD	First time	Not applicable	ADD	--
ADD	Revision	No changes to service account, billing account, or billing profile	NONE	No changes for customer sync to process.
ADD	Revision	Changes to service account, billing account, or billing profile	UPDATE	From a customer sync perspective, the fact that it is a revision is irrelevant in that it just checks whether the customer data referenced on the order exists in Oracle BRM; if not, it creates it.
				If customer sync is using the original ADD line, a billing hierarchy is created, and on the revision the attributes that affect the hierarchy are changing, then it makes the required change.
				The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.
				Prior value fields are specifically used in flagging and determining that a paying hierarchy change has occurred.
ADD	Revision	Cancellation. Manifests as a missing line on the revision.	DELETE	This action is ignored by the customer sync.
				If the ADD line added a new account, bill-info, and pay-info, and then the request for a new purchase was canceled, then those entities are not inactivated or deleted.
				If the ADD line created a paying hierarchy, and then the request for new purchase was canceled, then the paying hierarchy stays in place.
UPDATE	First time	Not applicable	UPDATE	Expects prior value fields to be populated.
UPDATE	Revision	No changes to service account, billing account, or billing profile	NONE	No changes for customer sync to process.

Table 10–1 (Cont.) Actions on Order Line Expectations Summary

Original Action on Order Line	Is this the first time the order line is being processed by customer sync or is it a revision?	What is occurring on the revision (that is relevant to customer sync)	Expected Action on compensation order line, set by Order Management	Comments
UPDATE	Revision	Changes to service account, billing account, or billing profile	UPDATE	From a customer sync perspective, the fact that it is a revision is irrelevant in that it just checks whether the new set of customer and billing profiles exist in Oracle BRM; if not, it creates it.
				If customer sync is using the original UPDATE line a billing hierarchy is created or updated, and on the revision the attributes that affect the hierarchy are changing, then it makes the required change.
				The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.
UPDATE	Revision	Cancellation. Manifests as a missing line on the revision or the action changing to a "-" (NONE).	UPDATE	If the original update line created a new account and billing profile in Oracle BRM, then it cannot be undone.
				For the attributes that *have changed* on the original line, the order management system flips the values (old, new) on the compensation line. For the case in which a hierarchy has been updated, this in essence reverts that update.
MOVE-ADD	First Time, but can change billing account and billing profile as part of a move-add.	Not Applicable	MOVE-ADD	Expects prior value fields to be populated for values that are changing from an existing asset.

Table 10–1   (Cont.) Actions on Order Line Expectations Summary

Original Action on Order Line	Is this the first time the order line is being processed by customer sync or is it a revision?	What is occurring on the revision (that is relevant to customer sync)	Expected Action on compensation order line, set by Order Management	Comments
MOVE-ADD	Revision	No changes to service account, billing account, or billing profile*	NONE	No changes for customer sync to process.
MOVE-ADD	Revision	Changes to service account, billing account, or billing profile*	MOVE-ADD	From a customer sync perspective, the fact that it is a revision is irrelevant in that it just checks whether the new set of customer and billing profiles exist in Oracle BRM; if not, it creates it. If customer sync is using the original UPDATE line, a billing hierarchy is created or updated, and on the revision the attributes that affect the hierarchy are changing, then it makes the required change. The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.
MOVE-ADD	Revision	Manifests as a missing line on the revision or the action changing to a "-" (In essence the line is canceled).	MOVE-ADD	If the original MOVE-ADD line created a new account and billing profile in Oracle BRM, then it cannot be undone. For the attributes that *have changed* on the original line, the order management system flips the values (old, new) on the compensation line. For the case in which a hierarchy has been updated, this in essence reverts that update.

* Billing integration supports only changes to billing account and billing profile as part of MOVE-ADD.

> **Caution:** The process integration for billing management (delivered in the Agent Assisted Billing Care pre-built integration) assumes that a given billing profile is synchronized to a single billing system. It does not support the ability to query data for the same billing profile from multiple billing system. For that reason, if that process integration is in use, then the same billing profile must not be used on an order for services that are fulfilled in different billing systems.
>
> For more information about this assumption, see the billing management chapter in the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

10.2 Solution Assumptions and Constraints

For solution assumptions and constraints for this business flow, see Section 12.8, "Solution Assumptions and Constraints."

11

OLM - Synchronize Fulfillment Order Billing Account Business Flow: Implementation

This chapter provides an overview of the Synchronize Fulfillment Order Billing Account business flow and discusses the implementation of Oracle Billing and Revenue Management (Oracle BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 11.1, "Synchronize Fulfillment Order Billing Account Business Flow Overview"
- Section 11.2, "Interfacing Orders to Create Customer Data in Oracle BRM"
- Section 11.3, "Oracle BRM Interfaces"
- Section 11.4, "Industry Oracle AIA Components"
- Section 11.5, "Integration Services"

11.1 Synchronize Fulfillment Order Billing Account Business Flow Overview

This business flow is enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM), Oracle Order and Services Management (Oracle OSM), and Oracle BRM pre-built integration options.

The following integration flow involves interfacing order customer accounts to one or more Oracle BRM instances:

- Interfacing orders to create customer data in Oracle BRM.

This flow leverages the *Create/Sync Account* integration flow, which enables the synchronization of customer information from Siebel CRM to Oracle BRM.

For more information about the Create/Sync Account integration flow, see Chapter 18, "Understanding the Process Integration for Customer Management."

11.2 Interfacing Orders to Create Customer Data in Oracle BRM

This integration flow uses the following interfaces:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF

- ■ * CommunicationsCustomerPartyEBSV2Resequencer

- ■ * QueryCustomerPartyListSiebelProvABCSImplV2

- ■ * SyncCustomerPartyListBRMCommsProvABCSImpl

- ■ ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSProducer

* - Indicates integration artifacts that are from the Customer Management process integration (Create/Sync Account flow). They are described in more detail in Section 19.6, "Integration Services."

Figure 11–1 Interfacing Orders to Create Customer Data in BRM Sequence Diagram
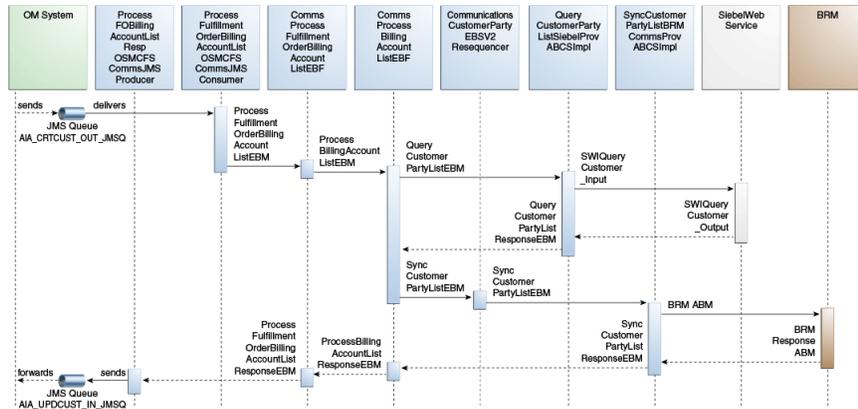
This is the sequence of events:

1. The order management system drops the message into the AIA_CRTCUST_OUT_ JMSQ JMS queue, which is picked up by the ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer and routed to CommsProcessFulfillmentOrderBillingAccountListEBF.

2. The CommsProcessFulfillmentOrderBillingAccountListEBF extracts the relevant customer data (ProcessBillingAccountListEBM) and routes it to CommsProcessBillingAccountListEBF.

   This leverages the *Create/Sync Account* flow.

3. The CommsProcessBillingAccountListEBF prepares the QueryCustomerPartyListEBM message.

   This enterprise business message (EBM) is required for querying the entire account data from Siebel CRM before creating the account in Oracle BRM

4. The CommsProcessBillingAccountListEBF routes the message to QueryCustomerPartyListSiebelProvABCSImplV2 (a core service implemented in the MDM pre-built integration).

5. The QueryCustomerPartyListSiebelProvABCSImplV2 prepares the application business message (ABM), which is required to invoke Siebel's SWI_Customer_ Party_Service.

   This query service invokes the Siebel database and fetches the account details and replies to QueryCustomerPartyListSiebelProvABCSImplV2 with response ABM.

6. This response ABM is then transformed to the QueryCustomerPartyListResponseEBM and is sent back to the CommsProcessBillingAccountListEBF.

7. The CommsProcessBillingAccountListEBF service instantiates SyncCustomerPartyListBRMCommsProvABCSImpl.

8. This SyncCustomerPartyListBRMCommsProvABCSImpl invokes Oracle BRM on opcode PCM_OP_CUST_COMMIT_CUSTOMER to create an account. To update an existing account, either opcode PCM_OP_CUST_UPDATE_CUSTOMER or PCM_OP_CUSTCARE_MOVE_ACCT is called, whichever is applicable.

9. If an account is successfully created or updated, an appropriate response (SyncCustomerPartyListResponseEBM) is sent back to CommsProcessBillingAccountListEBF in an asynchronous delayed response mode.

10. The CommsProcessBillingAccountListEBF then sends the ProcessBillingAccountListResponseEBM response message to CommsProcessFulfillmentOrderBillingAccountListEBF in an asynchronous delayed response mode.

11. CommsProcessFulfillmentOrderBillingAccountListEBF drops a message into AIA_UPDCUST_IN_JMSQ store and forward (SAF) queue where order management is notified of the SyncCustomer status.

11.2.1 Defining Transaction Boundaries and Recovery Details

For this flow there are two transaction boundaries. Table 11–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

If any account creation causes a system or business error, any further updates to the account (and thereby processing of other orders for that account) do not occur until the error is fixed. All updates for that account are locked in the sequencer table. If the error is a business error then the message must be removed from the sequencer table and if the error is a system error then the message must be resubmitted.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCSImplV2
- CommunicationsCustomerPartyEBSV2Resequencer
- SyncCustomerPartyListBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingAccountListRespOSMCFSCommsJMSProducer

Table 11–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer routes the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which extracts relevant customer data and then routes message to CommsProcessBillingAccountListEBF. The message is then routed to QueryCustomerPartyListSiebelProvABCSImplV2, which fetches account details and a response is sent back to CommsProcessBillingAccountListEBF that invokes CommunicationsCustomerPartyEBSV2Resequencer.	AIA cross-reference entries for some of the Siebel entries. Message goes into the sequencer table.	Rollback JMS message to the originating queue AIA_CRTCUST_OUT_JMSQ_ErrorQ.	Resubmit the order from AIA_CRTCUST_OUT_JMSQ_ErrorQ.
CommunicationsCustomerPartyEBSV2Resequencer instantiates SyncCustomerPartyListBRMCommsProvABCSImpl, which invokes Oracle BRM to either create or update an account. If successful a response is sent back to CommsProcessBillingAccountListEBF. A response message is then routed to CommsProcessFulfillmentOrderBillingAccountListEBF. ProcessFulfillmentOrderBillingAccountListRespOSMCFSCommsJMSProducer produces the response message to AIA_UPDCUST_IN_JMSQ.	AIA cross-reference entries. Message goes to queue AIA_UPDCUST_JMSQ.	Rollback cross-reference transactions. Rollback data created in Oracle BRM. Message goes back to the sequencer table.	Resubmit the message from the sequencer table.

> **Note:**   If any order contains more than one account and a failure occurs after any account is processed successfully but the subsequent account fails, then error recovery may become difficult based on the point of failure. Customers must first examine the point of failure and then determine if it's necessary to recover the BPEL instance from the recovery console.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

11.3 Oracle BRM Interfaces

This integration flow uses these services:

- PCM_OP_CUST_COMMIT_CUSTOMER

- PCM_OP_CUSTOMER_UPDATE_CUSTOMER

- PCM_OP_CUSTCARE_MOVE_AACT

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation,* "BRM Documentation," Reference, API reference."

For information about the Oracle BRM interfaces used by the Create/Sync Account integration flow, see Section 19.3, "Oracle BRM Interfaces."

11.4 Industry Oracle AIA Components

This integration flow uses these industry components:

- FufillmentOrderEBO

- ProcessFulfillmentOrderBillingAccountLstEBM

- ProcessBillingAccountListEBM

- ProcessFulfillmentOrderBillingAccountListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

For more information about the industry Oracle AIA components used by the Create/Sync Account integration flow, see Section 19.5, "Industry Oracle AIA Components."

11.5 Integration Services

These services are delivered with the *Interfacing Orders to Create Customer Data in Oracle BRM* integration flow:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer

- CommsProcessFulfillmentOrderBillingAccountListEBF

- CommsProcessBillingAccountListEBF

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSProducer

For more information about the integration services delivered with the Create/Sync Account integration flow, see Section 19.6, "Integration Services."

11.5.1 ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer

The ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer is a Mediator process that has a JMS Adapter Service, which continuously polls the Oracle AIA queue AIA_CRTCUST_OUT_JMSQ.

The ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer dequeues the ProcessFulfillmentOrderBillingAccountListEBM message and routes it to the CommsProcessFulfillmentOrderBillingAccountListEBF.

This service has one operation: Consume_Message.

11.5.2 CommsProcessFulfillmentOrderBillingAccountListEBF

The CommsProcessFulfillmentOrderBillingAccountListEBF is implemented as an asynchronous BPEL process. It performs these operations:

- Receives the ProcessFulfillmentOrderBillingAccountListEBM from the Oracle OSM with the target Oracle BRM instance identified.

- Transforms the message into the ProcessBillingAccountListEBM appropriately.

- Invokes the CommsProcessBillingAccountListEBF.

- Awaits response from CommsProcessBillingAccountListEBF.

- On receipt of response, drops a message into AIA_UPDCUST_IN_JMSQ store and forward (SAF) queue where order management is notified of the SyncCustomer status.

This process has the following operations.

- Operation: initiate

  This is an asynchronous operation to start the CommsProcessFulfillmentOrderBillingAccountListEBF.

- Operation: CallbackResponse

  This is an asynchronous callback operation. It makes a call back to the calling process, and passes a FaultMsg in the EBMHeader in case of any error received from CommsProcessBillingAccountListEBF.

For error scenarios, a response message can be optionally sent back to the order management system. The decision whether to send a response message back to the order management system is done based on the responseCode attribute of the DataArea of the incoming EBM (ProcessFulfillmentOrderBillingAccountListEBM) from the order management system.

If the responseCode value in the incoming EBM is REQUIRED_FOR_BUSINESS_ AND_SYSTEM_ERRORS, the response message is sent back to the order management system for all errors. However, if the responseCode value is REQUIRED_FOR_ BUSINESS_ERRORS, the response message is only sent back to the order management system for business errors.

> **Caution:**   with errors, Oracle OSM and the OSM AIA cartridges do not expect a response back. Instead, they use the Oracle AIA order fallout notification to both generate a trouble ticket and change the order and line status to indicate failure.

For more information about EBFs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Flows."

11.5.3 CommsProcessBillingAccountListEBF

The CommsProcessBillingAccountListEBF is implemented as an asynchronous BPEL process. it performs these operations:

- Receives ProcessBillingAccountListEBM from CommsProcessFulfillmentOrderBillingAccountListEBF.

- Constructs a QueryCustomerPartyListEBM payload and queries the Siebel web service with this payload through QueryCustomerPartyListSiebelProvABCSImpl.

- Receives a response QueryCustomerPartyListResponseEBM, constructs a SyncCustomerPartyListEBM message and then invokes and routes the message to QueryCustomerPartyListSiebelProvABCSImplV2.

11.5.4 ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer

The ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer is a BPEL process that has an adapter service, which produces the customer response messages to AIA_UPDCUST_IN_JMSQ.

This process has one operation: Produce_PFOBALResponse to produce the message into the AIA_UPDCUST_IN_JMSQ queue. This operation is called after the account or customer is interfaced in Oracle BRM.

12

OLM - Understanding the Bill Fulfillment Order Business Flow

This chapter provides information on how orders from Siebel Customer Relationship Management (Siebel CRM) are interfaced to Oracle Billing and Revenue Management (Oracle BRM), through an order management system like Oracle Order and Service Management (Oracle OSM). It lists various expectations of an order management system. *Oracle OSM and the OSM AIA Cartridges obey these expectations*.

This chapter includes the following sections:

- Section 12.1, "Bill Fulfillment Order Overview"
- Section 12.2, "Interfacing Orders to Oracle BRM"
- Section 12.3, "Supporting Simple Service Bundles"
- Section 12.4, "Supporting Single Phase versus Two-Phase Billing"
- Section 12.5, "Supporting Revisions"
- Section 12.6, "Supporting Time-Based Offerings"
- Section 12.7, "Supporting Friends and Family Lists"
- Section 12.8, "Solution Assumptions and Constraints"

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM, Oracle OSM, and Oracle BRM pre-built integration options.

12.1 Bill Fulfillment Order Overview

A customer must be billed for the services purchased and their usage. 'The process integration for order lifecycle management (OLM) provides a service that can be called by an order management system (such as Oracle OSM) to interface the order to Oracle BRM. This creates the required transaction data so that Oracle BRM can bill the customer.

For more information about calling this service, see Appendix H, "Expectations from a COM System for Billing Integration."'

As part of interfacing a new order or change order to Oracle BRM, the process integration for OLM supports purchasing the following in Oracle BRM:

- Products of type *Item* that apply to an account (for example, promotion penalty charges).
- Products of type *Item* that apply to a service (for example, one-time charges).

- Products of type *Subscription* that apply to an account (for example, charges for mailing a monthly paper invoice).

- Products of type *Subscription* that apply to a service (for example, wireless service).

- Discounts of type *Subscription* that apply to an account (for example, account-level discounts).

- Discounts of type *Subscription* that apply to a service (for example, a free minutes discount).

Oracle BRM products and discounts design time data is synchronized to Siebel CRM by the process integration for product lifecycle management (PLM).

For more information about the process integration for PLM, see Chapter 2, "Understanding the Process Integration for Product Lifecycle Management."

For more information and examples of supported products, see Appendix D, "OLM Bill Fulfillment Order - Matrix of MACD Actions Supported Per Billing Product Type."

12.2 Interfacing Orders to Oracle BRM

As part of interfacing a new order or change order to Oracle BRM, the process integration:

- Creates or updates service instances and purchased product and discount instances in Oracle BRM as part of the order interface to Oracle BRM.

  The integration supports the following actions: ADD, DELETE, UPDATE, SUSPEND, RESUME, MOVE-ADD, and MOVE-DELETE.

  It supports communicating updates to the service identifier, billing account, billing profile, and price changes on existing services.

  As part of service cancellations or promotion upgrade or downgrades, when an old product is canceled, whether the customer gets a refund for (billed) monthly charges or whether the refund is prorated depends on product-level controls in Oracle BRM.

  As part of a *Move* transaction, the integration supports changing Service Identifier, Billing Account, and Billing Profile. The integration does not support purchasing new products or canceling existing products as part of a *Move* transaction.

  > **Tip:** Transferring a service from one location to another in Siebel CRM results in lines with the action of MOVE-ADD and MOVE-DELETE. (This was previously referred to as *Move*).

  For more information, see Appendix D, "OLM Bill Fulfillment Order - Matrix of MACD Actions Supported Per Billing Product Type," Appendix E, "OLM - Examples of Changing the Paying Parent on Subordinate Accounts," and Appendix C, "OLM - Mapping Billing Dates."

  The solution supports account-level default balance groups alone.

  – The account-level balance group is created when the service account or billing account referenced on the order is created in Oracle BRM. A balance group in Oracle BRM can reference a single bill-info. When the account-level balance group is created, it uses the first billing profile referenced on the first order processed for an account. Thus all account-level products and services for a given account on the same order or subsequent orders must reference the

same billing profile. An order violating this assumption fails billing integration with an Oracle BRM error.

- The solution does support updating an existing billing profile in Siebel CRM; such changes are synchronized to billing outside of the order integration flow.

For more information, see the *Siebel CRM Integration Pack for Oracle Communications BRM: Agent Assisted Billing Care Implementation Guide*.

Because the solution supports only an account-level balance group, transfer of services (or account-level products) from one account to another is not supported.

Change orders that update the service account on existing services fails billing integration with an Oracle BRM error.

■ Communicates pricing information such as price or discount overrides, discounts, and onetime and penalty charges as part of the order interface to Oracle BRM.

For price changes that occur mid-cycle, the integration passes the price or discount overrides on a purchased product as is, the new price goes into effect from the following billing period, and no credits or debits are issued for the current period. If the latter is desired, then the Siebel CRM user must explicitly disconnect and add the product with the new price versus changing the price on an existing product.

Onetime charges*, for actions such as suspend and resume, are applied as service-level charges. Penalty charges incurred for compromising a promotion agreement are communicated to Oracle BRM as account-level charges.

As delivered, Siebel CRM supports defining charges for any of these actions: *Suspend, Resume, Move*, and *Delete*. One can extend Siebel to define charges for other actions such as *Update*.

For example, a communications service provider (CSP) charges a customer a fee for requesting a change to their phone number or billing profile. The order billing integration generically supports such charges regardless of the action that triggered the charge.

The integration expects order lines representing such charges to be tied to the service bundle line using the related asset integration ID and due date (on the Siebel order line) and using the charge parent line (on the order enterprise business message (EBM)). Therefore, any lines on the order that are tied to the service bundle line (regardless of the action on that line) using the related asset integration ID and due date (on the Siebel order**) and using the charge parent line (on the order EBM) are processed by the billing interface and applied to the respective service instance.

If the application business connector service (ABCS) that transforms the Siebel Order application business message (ABM) to the order EBM is unable to resolve the base line that a new order or change order onetime charge maps to, it does not populate the charge parent line and the charge is applied to the account when the charge line is interfaced to billing.

* Refer to the product bundling methodology for defining onetime charge products in Oracle BRM and synchronizing them to Siebel for tying them to new order or change order actions.

** The onetime charge points to the service bundle line using the related asset integration ID. The integration assumes that the due date on the charge line equals the service bundle line with the new order or change order action that triggered the charge. For example, service is suspended and resumed by the same order and two different charges are applied. The charge line applied for the suspend action

points to the service bundle line with the SUSPEND action, and the due date on both the lines are the same. The charge applied for the resume action points to the service bundle line with the RESUME action, and the due date on both the lines are the same.

For more information about service bundles, see Section 3.3, "Understanding the Product Bundling Methodology."

The pricing commit type on the order line controls whether the difference between the list and the selling price (due to promotion bundling discounts, matrix discounts, or manual price overrides) on a purchased product is communicated as a price or discount override to billing. Price overrides cannot be accounted for in General Ledger (GL) in Oracle BRM but discount overrides can be.

- If the pricing commit type is set to *Committed*, then the integration sets a price override when purchasing the product in billing.

- If the pricing commit type is set to *Dynamic*, then the integration sets a discount override when purchasing the product in billing.

- The Dynamic Discount method on the line controls whether the discount override is of type *Percent* or *Amount*.

- In the case in which the intent is to use Oracle BRM pricing as is, the pricing commit type on the order line must have a value of *Dynamic*, and neither the discount amount nor the discount percent are set. In this case, the integration sets neither a price nor a discount override for the product purchased.

Note: At most, for a charge type within a given product, Oracle BRM allows a single override price. In other words, if an Oracle BRM product is mapped to multiple events of the same type and is synchronized to Siebel CRM as a complex product with multiple simple products, the Siebel CRM application cannot override the price for the charge type that has multiple charges defined. If it does, it is applied as the override value for all charges of that charge type. This same constraint also applies to discount overrides.

For more information about using the pricing commit type and dynamic discount method, see the Siebel product documentation.

- Communicates service identifiers (for example, phone number for land-based or wireless phone service) to the billing system as part of the order interface to billing. The service identifier on the service bundle line in Siebel CRM is communicated to Oracle BRM. For telephony services, it is used as the phone number. For nontelephony service, it is used as the login and password.

- Communicates Siebel promotion information for invoice display.

To allow Oracle BRM to display promotion information on the invoice, the integration communicates the following information about the promotion when interfacing an order for billing:

- For new promotion purchases, the integration creates bundle instances (under the billing account on the order line) with the following information:

  Promotion name

  Promotion description

Effective start date (purchase date from Promotion Order line, if available, else request date if available, else Oracle BRM defaults current date).

– The integration creates the purchased product and discount instances for the respective purchased bundle instance. Such references are not created for products of type *Item*.

– As subsequent orders are processed, the integration creates new references as needed and maintains existing references such that the purchased products and discounts point to the bundle instance that is current.

– When a purchased promotion is canceled as part of a downgrade, upgrade, or cancellation, the integration cancels the bundle instance in Oracle BRM by specifying an effective end date. The integration uses the actual delivery date (on the order line canceling the promotion). If the actual delivery date is not available, it uses the request date.

> **Caution:** No support is provided for translation of promotion name or description. Changing the name and description of the promotion (design time data) in Siebel CRM does not have any effect on transactions that have been submitted for processing and interfaced to billing.

■ The service that interfaces the order to Oracle BRM either processes all of the lines on the incoming message or none of them. If an error occurs while it is processing the lines, then the entire transaction is rolled back.

For more information about order fallout, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

12.3 Supporting Simple Service Bundles

The Oracle Communications Order to Cash pre-built integration supports two product bundling methodologies: service bundles and simple service bundles.

For more information about the bundling methodologies, see Section 3.3, "Understanding the Product Bundling Methodology."

Order billing integration supports the simple service bundle methodology for all supported features, within the listed constraints.

Here is a summary of how the integration supports purchases of simple service bundles:

■ Purchasing a simple service bundle creates both a service instance and a purchased product instance in Oracle BRM. If the service was purchased within the context of a promotion, the product instance in Oracle BRM is tied to the purchased promotion (or bundle) instance. See Section 12.3.1, "Cross-Reference Impact."

■ The quantity (if > 1) on a simple service bundle line applies to the product purchase alone. Therefore, a single simple service bundle line creates:

– A single service instance *and*

– A single purchased product instance with a quantity as specified on the order line.

■ Both single-phase billing and two-phase billing are supported for the simple service bundle.

Here is a summary of how the integration supports changes to purchased simple service bundles:

- Suspending or resuming the asset that represents a simple service bundle suspends or resumes the service and product on Oracle BRM.

- Disconnecting the asset that represents a simple service bundle cancels the service and product instance in Oracle BRM.

  When using a simple service bundle, you *cannot* cancel the product without canceling the service.

- Transferring the asset that represents a simple service bundle in Siebel (Move-add or Move-delete) results in the cross-reference being adjusted for both the service and purchased product instance.

- Updates to service instance attributes (for example, Service ID, billing account/billing profile) on the asset that represents a simple service bundle results in the appropriate updates to the service instance in Oracle BRM.

- Updates to product attributes* (for example, pricing changes, promotion reference) on the asset that represents a simple service bundle results in the appropriate updates to the purchased product instance in Oracle BRM. Changes to billing dates as part of two-phase billing are honored.

  * - Quantity changes are not propagated to Oracle BRM for this release.

- If a onetime charge was defined and applied for a Move, Add, Change, and Disconnect (MACD) action in Siebel, it is applied in Oracle BRM to the balance group that the service instance points to.

12.3.1 Cross-Reference Impact

With simple service bundles, a single Siebel asset (for the simple service bundle product) is mapped to both the service instance and the purchased product instance in Oracle BRM. To manage mapping to both instances, the integration creates an additional cross-reference entry in the InstalledProduct cross-reference, as shown in Table 12–1.

Table 12–1   Cross-References Example

Cross-Reference Type	Siebel_01	Common	BRM_01
InstalledProduct_Id	Siebel-S01	C-ON-01	BRM-A01
InstalledProduct_Id	--	C-ON-01+Child	BRM-B01

In this example, BRM-A01 is the Oracle BRM portal object ID (POID) for the service instance and BRM-B01 is the Oracle BRM POID for the purchased product instance. The common ID for the purchased product instance is the same value as the common ID for the service instance with the string "+Child" appended to it.

12.4 Supporting Single Phase versus Two-Phase Billing

The solution supports both single-phase and two-phase billing. In single-phase billing, the order is interfaced to billing (or billing-fulfilled) after the service is provisioned. In two-phase billing, the order is billing-initiated before the service is provisioned, and is billing-fulfilled after service activation.

12.4.1 Considerations for using the Single Phase versus the Two Phase Billing Pattern

Billing fulfillment scenarios lead to one of two fulfillment patterns, each of which must be supported by the order management implementation.

Single-Phase Billing

In this pattern, a service is interfaced to billing through Fulfill Billing toward the end of the fulfillment flow, after the order is delivered and the actual delivery date is known.

The following business scenario requires this pattern:

- **All at Once**

  This scenario is the most common. Here the CSP does not have the concerns mentioned below for two-phase billing. In this case interfacing to Billing takes place after the service or product is *made available** to the customer.

  * - The interpretation of *made available* may vary among CSPs, based on jurisdiction and based on whether the subject is a service or a physical good. For example, physical goods that require no network activation or on-site installation might be billed immediately after the goods are shipped. The exact timing is built into the fulfillment flows associated with the underlying product specification through the Actual Delivery Date and other billing date attributes.

Two-Phase Billing

In this pattern, a service is interfaced to billing twice:

Initiate Billing: The service and purchased products are interfaced early in the fulfillment flow and before actual delivery dates are known.

Fulfill Billing: Accurate billing dates are updated in billing after the order is delivered and the actual delivery date is known.

The following business scenarios require this pattern:

- **Phased for Time Latency**

  In this scenario, the CSP has these concerns:

  Operational or deployment conditions produce a time lag between the time a service is made available for customer use and the time the service is interfaced into billing. Therefore, usage records can go into error logs and the CSP may lose revenue. CSPs attempt to plan fulfillment of future-dated orders to meet the requested delivery date, often using a safe margin that produces a time lag between the time a service is made available for customer use and the requested delivery date.

  In these cases, the usage cycle must start sooner than the billing cycle date. The fulfillment flow must be constructed such that the Usage Start Date is set to the current date during Initiate Billing, and the Cycle Start Date is set to a distant future date. At the time of Fulfill Billing, the Cycle Start Date is then reset to match the Actual Delivery Date or Requested Delivery Date, depending on business practices and legal requirements.

- **Phased for Validation**

  In this scenario, the CSP has these concerns:

  - Inadequate controls are in place to guarantee that valid orders interface to billing. Therefore, the CSP faces a high rate of invalid orders.

- The costs associated with delaying order line validation for interfacing to billing are prohibitive.

In these cases, orders must be interfaced to billing early in the fulfillment flow to ensure that the order can be interfaced successfully later. The fulfillment flow must be constructed such that the Purchase Start Date, the Usage Start Date, and the Cycle Start Date are set to a distant future date during Initiate Billing. At the time of Fulfill Billing, the Purchase Start, Usage Start Date, and Cycle Start Date are reset to match the Actual Delivery Date or Requested Delivery Date, depending on business practices and legal requirements.

12.4.2 Using the Single Phase versus the Two Phase Billing Pattern

To support various fulfillment latency requirements, the order billing interface can be called in two modes (by setting the ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentModeCode):

INITIATE BILLING

FULFILL BILLING

To enable single-phase billing, the order management system calls the order billing interface using only the FULFILL BILLING mode.

To enable two-phase billing, the order management system calls the order billing interface using the INITIATE BILLING mode before the service is provisioned and then after service activation, calls it using the FULFILL BILLING mode.

12.4.2.1 INITIATE BILLING Mode

An implementer can design an order orchestration flow such that it first interfaces the order to billing before the order is sent to provisioning. Calling the interface in this mode is optional. In this mode, the billing interface is called with either the whole order* or order components such as promotion lines, service bundles**, and account-level products. Depending on the requirements, the implementer should set some or all of the following dates on new purchases of products*** to the future (in essence they are treated as inactive when interfaced to billing):

Purchase Date (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderS chedule/PurchaseDate)

Cycle Start Date (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderS chedule/CycleStartDate)

Usage Start Date (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderS chedule/ServiceUsageStartDate)

Therefore, to support the scenario in which a fulfillment latency exists between service activation and billing, and you want to ensure that service usage is rated as soon as the service is activated but you want to start cycle fees only as of the date that the service was requested by the customer, you must have your order management system set the purchase and usage start dates to current and the cycle start date alone to the future when calling this service. See the subsequent section for certain modeling recommendations.

In this mode, the order interface to billing processes only new purchases of services or account-level products, or new purchases of products for existing services.

If a promotion is purchased as part of the new purchase, then that is also processed. Onetime charges for actions such as Suspend, Resume, Move, and Disconnect and promotion penalties are not processed in this mode.

* - All of the lines on the order that are intended for a certain target billing system and related lines such as promotion lines.

** - Service bundle means the service bundle line and all its component lines. The solution does not support a scenario in which some service bundle component lines are sent for billing initiation and billing fulfillment, while others are sent only for billing fulfillment. In such a scenario, the service bundle component lines that are sent only for billing fulfillment do not get processed.

*** - A product referenced on the Siebel CRM order line may result in the purchase of a product or a discount based on how it was originally defined in Oracle BRM. For the promotion line, only the purchase date is relevant.

For more information about how dates are set in Oracle BRM, see Appendix C, "OLM - Mapping Billing Dates."

Handling of Revision Orders

Oracle BRM has validation that prevents the caller from resetting purchase and cycle start dates when they become current. The integration does not reset the purchase date as part of billing-initiation revision processing. It does reset the cycle start and usage start date if asked by the caller. However, if billing initiation is called to process a revision on order lines that are billing-initiated, and billing initiation is asked to reset the cycle start date* if the previously set date is current, then billing initiation fails due to the Oracle BRM validation error.

* - In this case, the order management system sets the prior values for the billing dates to indicate to the billing integration that the dates are being reset.

Modeling and Implementation Recommendations

Here are some modeling and implementation recommendations:

- General

  The interface validates that the cycle date is set to the future for products of type subscription/discount. For products of type item, the interface validates that the purchase date is set to the future. As a best practice, it is recommended that when calling billing initiation, the caller set the billing date that is being set to the future to a year ahead of the due date.

  Oracle AIA deems the purchase, cycle start, or usage start dates as being in the future if the billing date in question is > (Fusion Middleware (FMW) current time converted to UTC + (25 or XX hours, whichever is greater)).

  XX is the value of the Oracle AIA configuration property: FutureTimeThresholdForBillingDates. This property has a default value of 8640 hrs (360 days in hours).

  If an implementer is highly confident of the lead time required to activate the service, then they can lower the value of the 'FutureTimeThresholdForBillingDates' property such that the order management system does not have to call fulfill billing to reset the dates (that were set in initiate billing). This also allows the billing dates to naturally become current soon after the service is activated. This property is settable per Oracle BRM instance level.

If the property 'FutureTimeThresholdForBillingDates' is not specified for a given billing instance, then the integration assumes the default value of 8640 hours (360 days).

> **Tip:** Products of billing type *Item* must be purchased with a future date in billing initiation to enable the integration to cross-reference them and therefore avoid repurchasing them in billing fulfillment. The 25-hour minimum threshold is hard-coded to enable this.

Oracle BRM requires that the purchase date be before or equal to usage and cycle start dates. If the caller does not follow this for any line*, then the billing interface (Oracle BRM ABCS) errors.

- Purchase Fees or Activation Charges

  Oracle BRM requires that the purchase date on a product be the same as or earlier than the usage start date. If activation (purchase fees) and usage charges were modeled on the same product to support the fulfillment latency scenario, you must set both the purchase date and start usage date to current. However, if the customer cancels their order before the service was provisioned, you must manually process a refund of the activation charges to them. To avoid this manual process, you must model the activation (purchase) fee on a product of type *Item*, which is a separate product from the one on which the usage and cycle charges are modeled. Now to support the fulfillment latency scenario, you set the purchase date for products of type *Item* to the future and set the purchase and usage start dates for the subscription products to current.

- Discounts

  If the service bundle includes products representing purchase or usage discounts, then to ensure that the customers get the discount, the purchase and usage start dates for the discount products must also be set to current when you are modeling the flow that sets the purchase and usage start dates to current for the subscription products.

* - A product referenced on the Siebel CRM order line can result in the purchase of product or a discount based on how it was originally defined in Oracle BRM. For the promotion line, only the purchase date is relevant.

12.4.2.2 FULFILL BILLING Mode

After provisioning is complete, the order orchestration flow can interface the order to billing in this mode. This is the default mode that the interface supports and is required to interface an order to billing.

In this mode, the interface processes all order lines that are sent (new and change orders). Onetime charges for actions such as Suspend, Resume, Move, and Disconnect and promotion penalties are processed in this mode.

For orders (order lines) that have been interfaced in the INITIATE BILLING mode, the caller can now set a specific date* (based on the actual delivery date) for those new purchases whose billing dates were earlier set to the future. Therefore, for the case in which only the cycle start date was set to the future during billing initiation, it must now be reset to the actual delivery date (date when the service was delivered). For the case in which the purchase, cycle start, and usage start dates were set to the future, the caller must now set them to the actual delivery date.

> **Caution:** Billing dates that are set to current in billing initiation must not be reset in billing fulfillment because it causes Oracle BRM to end in error.

The following prior values must be supplied:

PurchaseDate:
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
PurchaseDate

CycleStartDate:
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
CycleStartDate

ServiceUsageStartDate:
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
ServiceUsageStartDate

* - The interface relies on the population of prior value fields to indicate that an attribute on the line has changed. So your order management system must set the prior value fields for the billing dates.

12.4.3 Assumptions and Constraints for Two-Phase Billing

1. For multi-event billing products, the integration honors billing dates (purchase start date -nrc_start_date, cycle start date - rc_start_date, usage start date - usage_start_date in Siebel) on the parent complex product alone.

2. Billing Initiation is optional, but Billing Fulfillment is mandatory for an order (or order lines) to be interfaced to billing.

   (Billing Initiation is defined as the billing interface called in Initiate Billing mode. Billing Fulfillment is defined as the billing interface called in Fulfill Billing mode.)

3. The product that an order line references does not change after the line has been billing-initiated.

4. The order management system sends the onetime charge associated with a MACD action (Suspend, Resume, Move, Disconnect) with the service bundle on which the action is being performed.

5. Every MOVE-ADD line on a Siebel order has a matching MOVE-DELETE (and vice versa). The order management system sends MOVE-ADD lines along with the MOVE-DELETE lines to billing.

6. After order lines are submitted for Fulfill Billing, they are assumed to have hit a hard point of no return (PONR) and cannot be revised in Siebel CRM.

7. Service ID is always sent as input to the billing interface (Initiation or Fulfillment).

For more information about how dates are set in Oracle BRM, see Appendix C, "OLM - Mapping Billing Dates."

12.5 Supporting Revisions

To provide support for revisions after order lines are billing-initiated but not yet billing-fulfilled, the order interface to Oracle BRM expects the order management system to pass in a fulfillment mode at the line-level.

- The first time that billing initiation is called for order lines, the fulfillment mode should be set to *DO*.

- If an order line is successfully billing-initiated and subsequently the order line is revised in Siebel CRM and the order resubmitted, then the order management system compares the revised line against what was submitted to billing initiation, determines whether any changes must be processed, and calls billing initiation with a fulfillment mode of *REDO* to process the delta*.

  - Changes to the following attributes on a revised promotion line results in updates to billing: Billing Account, Purchase Date.

  - Changes to the following attributes on a revised account-level product line results in updates to billing: Billing Account, Bill Profile, Promotion reference, Pricing Information****, Billing Dates**.

  - Changes to the following attributes on a revised service bundle line results in updates to billing: Billing Account, Bill Profile, Promotion reference, Service ID.

  - Changes to the following attributes on a revised service bundle component line results in updates to billing: Pricing Information****, Billing Dates**.

  Caution: Revisions to order lines for products of type *Item* can be interfaced to Oracle BRM if the billing date is not current. When it is current, the call to update Oracle BRM fails.

  For more information about these attributes, see Appendix D, "OLM Bill Fulfillment Order - Matrix of MACD Actions Supported Per Billing Product Type."

- If an order line is successfully billing-initiated***** and subsequently the order line is canceled in Siebel CRM*** and the order resubmitted, then the order management system calls billing initiation with a fulfillment mode of *UNDO*.

- If no changes are made to an order line as part of a revision, but it must still be submitted for context (for example, the service bundle component line is revised but the service bundle line is not, the service bundle line is still sent because the service bundle as a whole is sent to Oracle BRM), then the order management system calls billing initiation with a fulfillment mode of *NOOP*.

Notes

* - Old attribute values are supplied only for delta changes.

** - Only cycle start and usage start dates should be changed if they are not yet current. The integration ignores requests to reset the purchase date.

For more information, see Section 12.4, "Supporting Single Phase versus Two-Phase Billing."

*** - On a Siebel revised order, this manifests as lines being dropped.

**** - Pricing information includes list price, selling (or net) price, pricing commit type, dynamic discount method, discount amount, and discount percent.

***** - The Oracle AIA service that interfaces order messages to Oracle BRM processes all lines or none of the lines. It does not do partial processing. Therefore, when an order is successfully billing-initiated, if any subsequent revisions for lines on the base order have to be processed, then the order management system must trigger compensation as described previously (using *REDO, UNDO*, or *NOOP* mode). If the order fails billing initiation (and triggers Order Fallout), a subsequent revision should be sent as is for billing initiation (DO mode).

> **Caution:** As delivered, the integration does not check for changes to the Special Rating List reference on revision orders when the List product has been billing-initiated.

Table 12–2 summarizes revision actions.

Table 12–2   Revision Actions

Action on Order Line	Fulfillment Mode (expected from the calling order management system)	Processed As	Comments
ADD	DO	ADD	Billing initiation processes only new purchases (lines with action of ADD).
ADD	REDO	UPDATE	Because billing initiation processes only new purchases (lines with action of ADD), changes to those lines are processed as updates. Prior value fields are set only for attributes that have changed on the revision.
ADD	UNDO	DELETE	Because billing initiation processes only new purchases (lines with action of ADD), cancellations to those lines are processed as deletes or disconnects.
ADD	NOOP	Ignored	Billing initiation processes only new purchases (lines with action of ADD); if on revision, those lines have not changed (from original order), then they are ignored.

12.5.1 Assumptions and Constraints for Revisions

1. Order lines are assumed to hit the PONR after they have been interfaced to Oracle BRM in the Fulfill Billing mode. Support for revisions is provided only for the case in which order lines have been billing-initiated (interfaced to billing in the Initiate Billing mode) but not yet billing fulfilled (interfaced to billing in the Fulfill Billing mode).

2. Only new purchases (lines with action ADD) are processed by billing initiation; hence billing initiation processes only revisions for new purchases.

3. The billing interface detects a changed attribute by the presence of an old attribute value for that attribute on the message. This is true for change orders and revisions.

12.6 Supporting Time-Based Offerings

The time-based offerings feature allows the definition and usage of products and discounts in Siebel CRM that are valid only for a specific duration, and expire after that.

The solution for time-based offerings has two components: design time and order time.

Design Time

For more information about the design time component, see Section 3.3.14, "Supporting Time-Based Offerings."

Order Time

- New Purchase - When an order for a time-based offering is placed and processed the following occurs:

  1. Siebel CRM calculates the end date, taking into account the start date (defaulted from due date) and the Duration, DurationUOM and DurationValidityStart transaction attribute values.

  2. The order is then submitted to the order management system for fulfillment. When the order is fulfilled, Oracle OSM AIA Cartridges set the purchase, cycle start, and usage start dates based on service actual delivery date and re-calculates the end date.

  3. When the order is billing fulfilled, the integration communicates the end date for the purchased product or discount to Oracle BRM.

  4. As part of the order update back to Siebel CRM, the order management system through the integration communicates the actual start and end dates to Siebel CRM.

- Change Order (such as a promotion upgrade or downgrade that results in changes to the duration validity of a previously purchased time based discount):

  1. Siebel CRM re-calculates the end date based on taking into account the Duration, DurationUOM and DurationValidityStart transaction attribute values.

  2. The order is then submitted to the order management system for fulfillment. When the order is fulfilled, Oracle OSM AIA Cartridges re-calculates the end date based on the actual delivery date. The end date is recalculated *only if any* of the validity attributes have changed on the order (by comparing against prior values) as follows.

     DurationValidityStart = Original End: Service End Date = Prior Value for Service End Date + Duration

     DurationValidityStart = Now: Service End Date = Actual Delivery Date Time + Duration

     DurationValidityStart = Original Start: Service End Date = Service Start Date + Duration

  3. When the order is billing fulfilled, the integration communicates the new end date for the purchased product or discount to Oracle BRM.

  4. As part of the order update back to Siebel CRM, the order management system through the integration communicates the changed end dates to Siebel CRM.

> **Tip:** It is recommended that end dates not be set during Billing Initiation since it is not required and avoids the requirement to manage them as part of revisions. Oracle OSM AIA Cartridges do not set end dates during Billing Initiation.

12.6.1 Assumptions and Constraints for Time-Based Offerings

1. When using an order management system other than Oracle OSM, it must behave as described above to enable support for time based offerings.

2. The Implementer must schedule a recurring job (daily or some other frequency based on their requirements) in Siebel to execute the workflow (SWI Asset Status Update Workflow) to inactivate such assets (time based offering products whose end date has passed). This is required to ensure that change orders for services that include time based offering products are successfully processed.

3. To ensure that the purchased products and discounts reflect the correct status after the expiration date is passed, the Implementer must periodically run the Oracle BRM utilities pin_cycle_fees -cancel and pin_discount_cleanup in Oracle BRM.

4. When a subscription product that is duration-based (Time Based Offering) and is also marked as a simple service bundle in Siebel CRM, is purchased and fulfilled, the Siebel CRM asset changes to inactive on the duration expiring. This results in scenario where the service instance is still active in Oracle BRM, but the corresponding asset in Siebel is inactive (because the same Siebel asset is mapped to both the service and the purchase product or discount instance in Oracle BRM). To handle such cases, the implementer must develop custom scripts to inactivate the respective service instances in Oracle BRM.

12.7 Supporting Friends and Family Lists

The friends and family feature enables end customers to call certain phone numbers at discounted rates. The feature requires special rating products to be defined in Siebel CRM and included in a service bundle.

For more information about how special rating products are supported and the methodology, see Section 3.3.12, "Supporting Friends and Family."

When orders for such service bundles are placed, the customer service representative (CSR) can create the lists, optionally add numbers to the lists, and associate the lists with the special rating products.

For more information, see the sections on friends and family plans in the "Profiles in Siebel Communications," chapter of the *Siebel Communications Guide*.

When the order is interfaced to Oracle BRM, the integration creates a list profile for every order line that has a special rating product. These list profiles are associated with the service instance in Oracle BRM. For the list profile to get created during order billing integration, a list (special rating profile list) must be associated to the special rating product on the order.

When the order is successfully interfaced to Oracle BRM and is auto-asseted, the special rating product used to capture the list is tracked as an asset in Siebel.

> **Caution:** The solution assumes that if the same special rating list is referenced by multiple services, (for example, VOIP and Wireless Voice) those services are fulfilled in the same Oracle BRM instance.

For more information, see Section 3.3.12, "Supporting Friends and Family" and Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."

12.7.1 Using Change Orders and Special Rating Products

Here are some recommendations for using change orders and special rating products.

- Changing Special Rating list entries:

  You can use either of the following two options to achieve this:

  – Tying a completely different list to the special rating product: You can use a change order to update the special rating list reference on the existing special rating product asset to a different list reference. When the integration processes the change, it updates the list profile in billing with contents from the new list.

  – Adding or removing entries from a list currently referenced by a special rating product: You can use the Siebel Special Rating Profile user interface (UI) to make changes to the list and synchronize them to Oracle BRM. This synchronization is enabled by the following integration services:

    ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

    ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

    ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

- Promotion upgrades and downgrades:

  Promotion upgrades or downgrades can result in the cancellation or addition of Special Rating products for an existing service.

  Cancellation: When such orders are processed, the integration deletes the respective list profile in Oracle BRM.

  Addition: When such orders are processed, the integration creates new list profiles in billing for the given service instance.

- Service cancellations:

  Service cancellation results in the deletion of the list profile in Oracle BRM.

12.7.2 Modifying Friends and Family List

After a service that supports special rating has been purchased and the order fulfilled and asseted, the customer can use the Siebel Special Rating Profile UI to make changes to their list, and then update and synchronize the list to Oracle BRM.

The flow uses the operation ProcessInstalledProductSpecialRatingSetList on the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl composite for this purpose. The specification group on the installed product EBM is used for Communicating the list entries.

For more information, see Chapter 20, "CM - Synchronize Customer Special Rating Profile: Implementation."

12.8 Solution Assumptions and Constraints

These are the solution assumptions and constraints for this integration flow:

1. The solution does not support an integration scenario in which multiple brands are defined within a single instance of Oracle BRM.

2. After an order in Siebel CRM is submitted for processing and successfully interfaced to billing, it *cannot be changed and resubmitted*. You must enforce this by defining rules in the Siebel state model. The order can be revised and resubmitted for processing if it has not reached a point-of-no-return (PONR). The solution assumes that the order line reaches the PONR after the line has been sent for billing fulfillment.

3. The Siebel Copy Orders feature does not regenerate the identifiers (asset integration Id) that uniquely identify the customer purchases on the copied order. This makes the copied orders invalid to back-end systems. Therefore, copied orders are not supported by Oracle AIA. Instead of copying orders, it is recommended that you use the Siebel Favorites feature.

4. Regarding quantity support for service bundles and account-level products, the solution assumes that the auto-explode flag on service bundle products is set to *Yes* and that the customer is using Siebel Asset Based Ordering processes to enforce service item instantiation.

    ■   The service bundle line always has a quantity of *1* when the order is handed off from Siebel CRM to the integration with the integration creating a single service instance in Oracle BRM (per service bundle line on the Siebel order).

        No special handling exists for order quantity > *1* for products whose auto-explode flag in Siebel is set to *No*.

    ■   Quantity (and not extended quantity) on service bundle components or account-level products is interfaced to Oracle BRM; this creates purchased product or discount instances (one instance per product or discount purchased) with the specified quantity, which is used to determine charge calculation.

    ■   When an order line is interfaced to Siebel CRM assets it creates a single asset with the specified quantity.

        Additionally, the integration does *not* look at quantity changes on revisions, or change orders (for existing products) and therefore such changes are not communicated to Oracle BRM.

5. No special handling exists for shippable goods. No support is available for returns or credit orders.

6. Order lines that must be sent to different billing systems have different billing profiles.

    > **Note:** This is a limitation only if the customer is also using the Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care pre-built integration. The Billing Management flows available as part of that integration, do not support the ability to display information from multiple billing systems for the same billing profile.

7. Order lines are interfaced to billing only after they have been provisioned.

    Based on this assumption, the service that interfaces the lines with billing creates the service instances, purchased product instances, purchased discount instances,

or a combination of these as active. This applies to scenarios of single-phase billing, in which billing interface is called one time in Fulfill Billing mode.

8. The service account, billing account, and billing profiles are the same on all order lines (components) in a service bundle.

    For service bundles, any integration logic that works on these fields looks only at the service bundle line. This constraint also applies to onetime charges that are added for MACD actions such as suspending or resuming a service, in that the integration ignores the service account, billing account, and billing profiles on such a line. The charge generated by such an order line is applied to the balance group that the service instance points to.

    This is an Oracle BRM limitation and is enforced in Siebel CRM.

9. The solution supports account-level default balance groups alone. A balance group in Oracle BRM can reference a single bill-info. This is the first billing profile that is referenced on the first order processed for an account.

    It follows that all services for a given account on the same order or subsequent orders must reference the same billing profile; an order violating this assumption fails billing integration with an Oracle BRM error.

    ■ If the order message contains multiple service being purchased, the integration (because of optimization), uses the billing profile on the first service for processing all of the services. In this case, the Oracle BRM validation and error are not raised.

    It follows that all account-level product purchases for a given account reference the same billing profile. Violation of this assumption does not result in failure because order billing integration ignores the billing profile specified on order lines for such products.

    The solution does support updating an existing billing profile in Siebel; such changes are synchronized with billing outside of the order integration flow.

10. In the case where an account is paying for its own services (and account-level products), the solution does not support changing the billing profile on existing services or account-level products to a different one using a change order:

    ■ Changing from one billing profile to another for a self-paying account is not supported.

    ■ Changing from one paying parent to another for a subordinate account is supported.

    ■ Changing from one billing profile to another (while retaining the same paying parent) for a subordinate account is supported.

    ■ Changing from self-paying to nonpaying subordinate is not supported*.

    ■ Changing from nonpaying subordinate to self-paying is not supported.

    This is an Oracle BRM limitation with account-level balance group usage. Order integration to billing fails with an Oracle BRM error for the preceding scenarios that are not supported.

    * - This specific scenario does not error but is not supported since it results in data that breaks the billing management integration flows.

11. Oracle BRM does not support a subordinate account having multiple paying parent.

Any order changing the paying parent for a subordinate account using a new purchase must include lines to change all the other services (and account-level products) for the subordinate account that was paid for by the old parent so that it can successfully interface customer data to Oracle BRM.

> **Caution:** Transactions that do not obey this assumption fails with an Oracle BRM error when an order is interfacing customer data to Oracle BRM.

Any order changing the paying parent for an existing service on a subordinate account changes the paying parent for all the other services (and account-level products) under that subordinate account. To ensure that Siebel CRM assets are synchronized with Oracle BRM, it is recommended that the change order to update the paying parent include an update for all the services (and account-level products) for a given subordinate account.

For more information, see Appendix E, "OLM - Examples of Changing the Paying Parent on Subordinate Accounts."

12. Transfer of services (or account-level products) from one account to another is not supported.

    For more information, see Section D.2, "Table B."

13. All lines within a service bundle reference products from the same billing system.

    Based on this assumption, a single Siebel CRM asset can be mapped to a service instance or a purchased product or discount instance in only one billing system.

14. The integration assumes that the service bundle product and its component products reference the same billing service type. This assumption applies only to component products that represent Oracle BRM products of type Subscription or BRM discounts. Violation of this assumption can result in Oracle BRM grouping the billed charges under the wrong bucket (bill-item). Nested service bundles do not have to have the same service type as the root parent service bundle.

13

OLM - Bill Fulfillment Order Business Flow: Implementation

This chapter provides an overview of the Bill Fulfillment Order business flow and discusses the implementation of Oracle Billing and Revenue Management (Oracle BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 13.1, "Bill Fulfillment Order Business Flow Overview"
- Section 13.2, "Interfacing Orders to Create Transaction Data in Oracle BRM"
- Section 13.3, "Oracle BRM Interfaces"
- Section 13.4, "Industry Oracle AIA Components"
- Section 13.5, "Integration Services"

13.1 Bill Fulfillment Order Business Flow Overview

This business flow is enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM), Oracle Order and Services Management (Oracle OSM), and Oracle BRM pre-built integration options.

The following integration flow involves interfacing orders to create transaction data in one or more Oracle BRM instances:

- Interfacing orders to create transaction data in Oracle BRM

13.2 Interfacing Orders to Create Transaction Data in Oracle BRM

This integration flow uses the following interfaces:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer

Figure 13–1 illustrates the integration components used by Oracle OSM to interface orders to create transaction data in Oracle BRM.

Figure 13–1 Interfacing Orders to Create Transaction Data in Oracle BRM
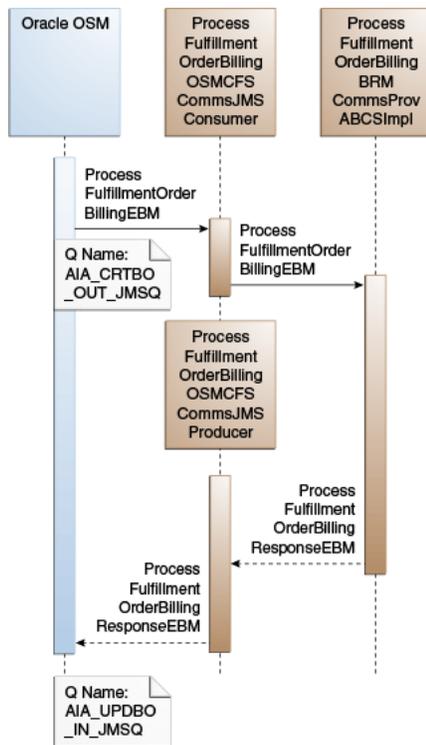
When this flow is initiated, the following events occur:

1. Oracle OSM composes a ProcessFulfillmentOrderBillingEBM and places it in a JMS Queue. The store and forward (SAF) mechanism pushes the message to the AIA_CRTBO_OUT_JMSQ messaging queue.

2. The ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer picks up this message, and routes this message to the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl.

3. The ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl service orchestrates the ProcessFulfillmentOrderBillingEBM into creating billing artifacts, service instances, purchased products, purchased discounts, and so on in Oracle BRM.

4. The ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl constructs the ProcessFulfillmentOrderBillingResponseEBM and sends it back to the ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer.

5. The ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer forwards the ProcessFulfillmentOrderBillingResponseEBM in the AIA_UPDBO_IN_JMSQ using SAF to an Oracle OSM messaging queue on the WebLogic server.

13.2.1 Defining Transaction Boundaries and Recovery Details

For this flow there is one transaction boundary. Table 13–1 describes the transaction involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer

- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

- ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer

Table 13–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer passes message to ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl, which creates billing artifacts and call one or more subprocesses. The response message is then routed to ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer.	AIA cross-references created. Oracle BRM data created. Message goes to queue AIA_UPDBO_IN_JMSQ.	Rollback AIA cross-references. Rollback data created in Oracle BRM. Message goes back to the originating queue AIA_CRTBO_OUT_JMSQ_ErrorQ.	Resubmit the order from AIA_CRTBO_OUT_JMSQ_ErrorQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

13.3 Oracle BRM Interfaces

This integration flow uses these services:

- PCM_OP_CUST_MODIFY_CUSTOMER

- PCM_OP_CUST_CREATE_PROFILE

- PCM_OP_CUST_DELETE_PROFILE

- PCM_OP_CUSTMODIFY_PROFILE

- PCM_OP_CUST_SET_STATUS

- PCM_OP_CUST_UPDATE_SERVICES

- PCM_OP_SUBCRIPTION_PURCHASE_DEAL

- PCM_OP_SUBCRIPTION_CANCEL_PRODUCT

- PCM_OP_SUBCRIPTION_CANCEL_DISCOUNT

- PCM_OP_SUBSCRIPTION_TRANSFER_SUBCRIPTION

- PCM_OP_SUBSCRIPTION_SET_PRODINFO

- PCM_OP_SEARCH

- PCM_OP_READ_FLDS

- PCM_OP_SUBSCRIPTION_SET_BUNDLE

- PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS

- PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation,* "BRM Documentation," Reference, API reference."

13.4 Industry Oracle AIA Components

This integration flow uses these industry components:

- FufillmentOrderEBO

- ProcessFulfillmentOrderBillingEBM

- ProcessFulfillmentOrderBillingResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

13.5 Integration Services

These services are delivered with this integration flow:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer

- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

  - ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

  - ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

  - ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

  - ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

  - ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer

13.5.1 ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer

The ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer is a Mediator process that has a JMS Adapter Service, which continuously polls the AIA_CRTBO_OUT_JMSQ. The ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer dequeues the ProcessFulfillmentOrderBillingEBM message and routes it to the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl.

This service has one operation: Consume_Message.

13.5.2 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

The ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl consists of a BPEL process with one operation: ProcessBilling. It receives the Order EBM and then converts the message into a BRM-specific message based on which opcode must be invoked.

This service communicates with Oracle BRM using the custom Java EE Connector Architecture (JCA) adapter provided by Oracle BRM. It uses the default capability of the custom JCA adapter to define unit transactions for every order. (Do all or none.)

The routing to the right Oracle BRM instance is done using dynamic end point binding in the BPEL process using the target application that is decided.

This service accepts the appropriate ProcessFulfillmentOrderBillingEBM and is responsible for transforming to the relevant Oracle BRM ABM and invoking the corresponding opcode.

ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl mainly performs the following activities

- Evaluates the product type of the order line and the action code. If the particular order line is a ServiceBundle/Subscription/Discount/Item and if this line has never been interfaced to Oracle BRM, then it proceeds to call the subprocesses:

  - For ActionCode = 'ADD' and BillingMode = 'INITIATE BILLING' or 'FULFILL BILLING', ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is called.

  - For ActionCode = 'SUSPEND' or 'RESUME' and BillingMode = 'FULFILL BILLING', ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess is called.

  - For ActionCode = 'DELETE' and BillingMode = 'FULFILL BILLING', ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is called.

  - For ActionCode = 'UPDATE' or "MOVE-ADD" and BillingMode = 'FULFILL BILLING', ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess is called.

  - For ActionCode = 'MOVE-ADD' and BillingMode = 'FULFILL BILLING', ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is called.

  - For ActionCode = "ADD", FulfillmentModeCode = "REDO" and BillingMode ="INITIATE BILLING", ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess is called.

  - For ActionCode = "ADD", FulfillmentModeCode = "UNDO" and BillingMode ="INITIATE BILLING", ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is called.

- For friends and family orders, where the order has one or multiple SpecialRatingProduct as an OrderLine, the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl Process calls an Oracle BRM opcode from the following list depending on the nature of the action to be performed:

  - For New Order, the PCM_OP_CUST_CREATE_PROFILE opcode is called. Afterwards, the Oracle BRM POID is cross-referenced and populated in the AIA XREF database.

  - For Deleting the Special Rating Product, the PCM_OP_CUST_DELETE_ PROFILE opcode is called. After the call, the Oracle BRM POID is cross-referenced and deleted from the AIA XREF database.

  - For Deleting the Special Rating Product, the PCM_OP_CUST_MODIFY_ PROFILE opcode is called.

- For Promotion on Invoice, ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl calls the PCM_OP_ SUBSCRIPTION_SET_BUNDLE opcode and different values are passed depending on the particular functional operation.

- After all of these activities, the data is cross-referenced to the AIA XREF database.

  Figure 13–2 shows the data that is cross-referenced to the AIA XREF database:

Figure 13–2   Data Cross-Referenced to the AIA XREF Database
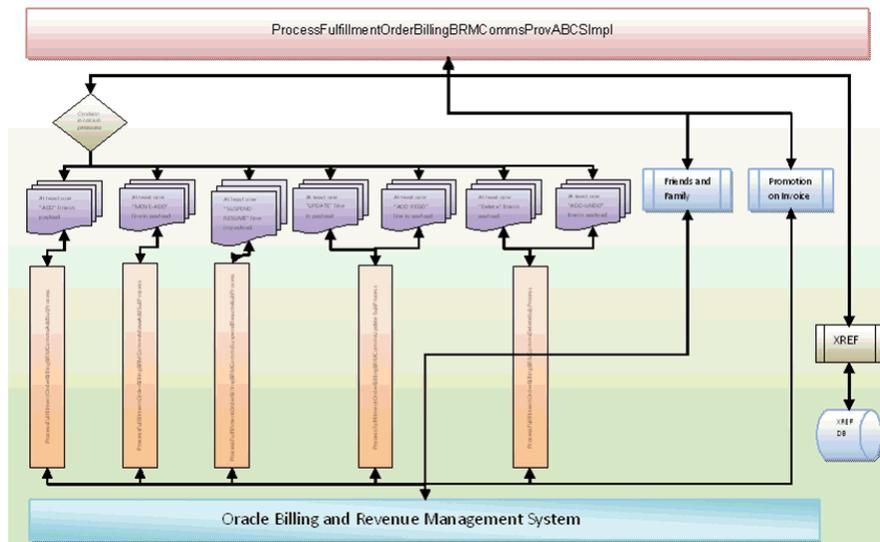
This service calls the following subprocesses in a synchronous fashion to perform various billing-related activities:

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

For error scenarios, a response message can be optionally sent back to the order management system. The decision whether to send a response message back to the

order management system is done based on the responseCode attribute of the DataArea of the incoming EBM (ProcessFulfillmentOrderBillingEBM) from the order management system.

If the responseCode value in the incoming EBM is REQUIRED_FOR_BUSINESS_ AND_SYSTEM_ERRORS, the response message is sent back to the order management system for all errors. However, if the responseCode value is REQUIRED_FOR_ BUSINESS_ERRORS, the response message is only sent back to the order management system for business errors.

> **Caution:**   With errors, Oracle OSM and the OSM AIA cartridges do not expect a response back. Instead, they use the Oracle AIA order fallout notification to both generate a trouble ticket and change the order and line status to indicate failure.

13.5.3 ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product.

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is called for a service bundle, account-level product, or account-level discount that is being newly added either as a part of a new order or an update order and that has an action code of ADD.

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is never called for any onetime penalty charges that also have an action code of ADD, but are being added as a part of the MACD operation performed on a service bundle or a promotion.

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess receives a custom message that has the ProcessFulfillmentOrderBillingEBM, XREFPopulate, and XREFDelete DataStructure.

The structure of the message coming in to the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess comprises:

- ProcessFulfillmentOrderBillingEBM

- XREFPopulate

- XREFDelete

Depending on the type of product for every OrderLine, the following operations are performed in the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess:

1. The incoming payload is tunneled through two transforms. The first transform groups all the ServiceBundles per service account and the second transform groups all the account-level purchases.

2. When the product type is service bundle, then this BPEL process accumulates all of the children inside this service bundle and calls the PCM_OP_CUST_MODIFY_ CUSTOMER opcode. During this call, the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess also transforms the ProcessFulfillmentOrderBillingEBM into a BRM-specific message. All the Service Bundles per service account are passed in one single PCM_OP_CUST_MODIFY_ CUSTOMER opcode call.

3. When the product type is an account-level subscription, discount, or item, then this BPEL process calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL

opcode. During this call, the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess also transforms the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message.

4. After these two calls are successfully carried out, this BPEL process captures the POID (ObjectIdentifier) returned by Oracle BRM and populates the XREFPopulateData.

5. For ITEM, the POID (ObjectIdentifier) is returned by Oracle BRM only during INITIATE BILLING mode.

This service communicates with Oracle BRM using the custom JCA adapter provided by Oracle BRM. It uses the default capability of the custom JCA adapter to define unit transactions for every order. (Do all or none.)

This service supports two modes of billing:

- initiate billing

- fulfill billing

13.5.4 ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

The ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingMove.

The structure of the message coming in ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess comprises:

- ProcessFulfillmentOrderBillingEBM

- XREFPopulate

- XREFDelete

When the action code on the order line is MOVE-ADD and the product type is a service bundle, an account-level product, or an account-level discount, the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is called.

- As part of Move-Add, Update of ServiceID, Price Override, and Discount Override, changes can be performed. To process the updates, after ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is invoked, BRM Main Provider - ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl invokes ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

- Simple Move-Add of the service bundles from one location to another.

  During this scenario, only the XREFs are repointed from. No Oracle BRM interaction happens in this operation.

- All the preceding Move-Add scenarios can be accompanied with or without a onetime penalty charge.

  When a onetime penalty charge is associated, the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL Oracle BRM opcode.

13.5.5 ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

The structure of the message coming in the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess comprises:

- ProcessFulfillmentOrderBillingEBM

- XREFPopulate

- XREFDelete

The ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingSuspendResume.

When the action code is SUSPEND or RESUME and the ProductType is a service bundle or an account-level subscription or account-level discount, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess is called.

The following operations are done by this process:

- When the action code is SUSPEND or RESUME and the product type is a service bundle.

  ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_CUST_SET_STATUS Oracle BRM opcode.

  When the action code is SUSPEND, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the Flag= 10102.

  When the action code is RESUME, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the Flag= 10100.

- When the action code is SUSPEND or RESUME and the product type is Account Level Discount:

  ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS Oracle BRM opcode.

  When the action code is SUSPEND, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the Flag= 2.

  When the action code is RESUME, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the Flag= 1.

- When the action code is SUSPEND or RESUME and the product type is Account Level Subscription.

  ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS Oracle BRM opcode.

When the action code is SUSPEND, then the
ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the
Flag= 2.

When the action code is RESUME, then the
ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess passes the
Flag= 1.

For Operation 1, a onetime penalty charge may or may not be associated.

■ When a onetime penalty charge is associated with the service bundle, then
depending on the action code, the onetime charge gets added in the following
manner:

When the action code is SUSPEND, the onetime charge gets added first.

ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts
the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and
calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL Oracle BRM opcode.

After the onetime charge is added, then Operation 1 is run to SUSPEND the
service bundle.

When the action code is RESUME, the onetime charge gets added after the service
bundle is resumed.

Operation 1 is run to RESUME the service bundle.

Afterwards, the onetime charge gets added:

ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts
the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and
calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL Oracle BRM opcode.

13.5.6 ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

The ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess is a synchronous
BPEL process that is called by the
ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the
action code present on the order line and also the type of product. It has one operation:
processBillingUpdate.

The structure of the message coming in
ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess comprises:

■ ProcessFulfillmentOrderBillingEBM

■ XREFPopulate

■ XREFDelete

When the action code is UPDATE and the product type is a service bundle or an
account-level subscription or account-level discount, then the
ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess is called.

This process supports the following update scenarios:

■ Update of the service ID for a particular service bundle.

During this scenario, users can update the service ID for one or more service
bundles as part of regular modify orders or as part of Move-Add.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the
ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and
calls the PCM_OP_CUST_UPDATE_SERVICES Oracle BRM opcode.

- Price Override

  During this scenario, users can change the PriceOverride on a product line.

  ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO Oracle BRM opcode.

- Discount Override

  During this scenario, users can change the DiscountOverride on a product line.

  ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO Oracle BRM opcode.

- TBO End Date

  During this scenario, users can change the EffectiveEndDate on a product line.

  ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO Oracle BRM opcode in case of SUSCRIPTION products and calls PCM_OP_SUBSCRIPTION_SET_ DISCOUNTINFO in case of DISCOUNT products.

13.5.7 ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

The ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is a synchronous BPEL process that is called by ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingDelete.

When the action code is DELETE and the product type is a service bundle or an account-level subscription or account-level discount, then the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is called.

The following operations are done by this process:

- When the action code is DELETE and the product type is Service Bundle, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_CUST_SET_STATUS Oracle BRM opcode.

- The ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess passes the StatusFlag=4 and Status = 10103 in this case.

- When the action code is DELETE and the product type is Discount, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT Oracle BRM opcode.

- When the action code is DELETE and the product type is Account Level Subscription, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and calls the PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT Oracle BRM opcode.

- During these operations, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess, also checks for the existence of any onetime penalty charge. If present, then the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the

ProcessFulfillmentOrderBillingEBM into an Oracle BRM-specific message and
calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL Oracle BRM opcode.

13.5.8 ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer

The ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer is a BPEL
process that has an adapter service, which produces the order response messages to
the AIA_UPDBO_IN_JMSQ.

This process has only one operation: Produce_ProcessFOBResponse to produce the
message into the AIA_UPDBO_IN_JMSQ AIA queue. This operation is called after the
order is interfaced into Oracle BRM.

14

OLM - Understanding the Provision Order and Update Fulfillment Order Business Flows

This chapter provides an overview of order provisioning, describes how provisioning orders are created, and how orders and statuses are updated in the Oracle Order and Service Management Central Order Management (Oracle OSM COM) system.

This chapter includes the following sections:

- Section 14.1, "Order Provisioning Overview"
- Section 14.2, "Creating Provisioning Orders"
- Section 14.3, "Updating Fulfillment Orders"
- Section 14.4, "Solution Assumptions and Constraints"

The provision order and update fulfillment order business flows are enabled using the Oracle Communications Order to Cash: Oracle OSM pre-built integration option.

14.1 Order Provisioning Overview

Customer order fulfillment requests, (both *Qualify* and *Deliver* types), received in Oracle OSM COM from Siebel Customer Relationship Management (Siebel CRM) are decomposed into suborders called order components depending on the fulfillment topology and the fulfillment dependencies. Some of the order components are targeted for a provisioning (Service Order Management) stack. Oracle OSM COM uses Oracle Application Integration Architecture (Oracle AIA) services to pass provision order requests to Oracle OSM Service Order Management (Oracle OSM Provisioning) instance or any third-party OSM system.

For more information about customer order fulfillment request types, see Section 7.3, "Order Capture Overview."

When Oracle OSM is also used as the Service Order Management application, Oracle OSM manages the order lifecycle management (OLM) events of the service order. For *Cancel* and *Revision* requests, Oracle OSM generates and executes compensation plans to efficiently match the change. OLM also manages order data and status updates, and fallout incidents. Throughout the fulfillment process, Oracle OSM Provisioning sends status and data updates to Oracle OSM COM.

14.2 Creating Provisioning Orders

For interacting with Oracle OSM Provisioning, OSM COM pushes the ProcessProvisioningOrderEBM message, which includes most of the SalesOrderEBO attributes, into AIA_CRTFO_OUT_JMSQ using sore and forward (SAF) mechanism. When it errors, the response comes through the Oracle AIA common error schema, otherwise, there is no response for this message. Provisioning Service operation responses are made through ProcessFulfillmentOrderUpdate service operations in all cases except an interface error or request failure. In these cases, the responses are passed to an Oracle AIA Error Handling service, which passes order failure information from provisioning to customer order management for order fallout handling.

For more information about this sequence of events, see Chapter 15, "OLM - Provision Order and Update Fulfillment Order Business Flows: Implementation."

For more information about order fallout, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

14.3 Updating Fulfillment Orders

This feature provides the ability to update Oracle OSM COM with Oracle OSM Provisioning milestones, status, and data.

Order Status Management is an integral capability of Oracle OSM COM. Oracle OSM COM provides for a configurable order status management across different fulfillment systems, including Oracle OSM Provisioning.

In addition, order milestones are configured to track order fulfillment progress. Fulfillment system responses and status updates are used to trigger evaluation of rules that progress the order item status and realize new milestones and in turn trigger aggregation rules that update the order-level status.

Also, several attributes are populated during design and assign that are critical to pass to other fulfillment systems and Siebel CRM.

14.4 Solution Assumptions and Constraints

One or more OSM Provisioning Cartridges must be deployed. They preserve the Oracle AIA interfaces.

For more information about product-specific assumptions and constraints, see the Oracle Order and Service Management product documentation.

15

OLM - Provision Order and Update Fulfillment Order Business Flows: Implementation

This chapter provides an overview of the Provision Order and Update Fulfillment Order business flows and discusses industry Oracle Application Integration Architecture (Oracle AIA) components and integration services.

This chapter includes the following sections:

- Section 15.1, "Provision Order and Update Fulfillment Order Business Flows Overview"
- Section 15.2, "Oracle OSM Fulfillment to Oracle OSM Provisioning Integration Flow"
- Section 15.3, "Industry Oracle AIA Components"
- Section 15.4, "Integration Services"

15.1 Provision Order and Update Fulfillment Order Business Flows Overview

These business flows are enabled using the Oracle Communications Order to Cash Oracle Order and Service Management (Oracle OSM) pre-built integration option.

The following integration flow involves passing provision order requests to Oracle OSM Service Order Management (Oracle OSM Provisioning) and then providing the ability to update Oracle OSM Central Order Management (Oracle OSM COM) with Oracle OSM Provisioning milestones, status, and data.

- Oracle OSM Fulfillment to Oracle OSM Provisioning

15.2 Oracle OSM Fulfillment to Oracle OSM Provisioning Integration Flow

This integration flow uses the following interfaces:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

Figure 15–1 illustrates the sequence of events for requesting provisioning of an order to Oracle OSM Provisioning and receiving updates back:

Oracle OSM Fulfillment (COM) to Oracle OSM Provisioning sequence diagram

Figure 15–1   Oracle OSM Fulfillment (COM) to Oracle OSM Provisioning Sequence Diagram
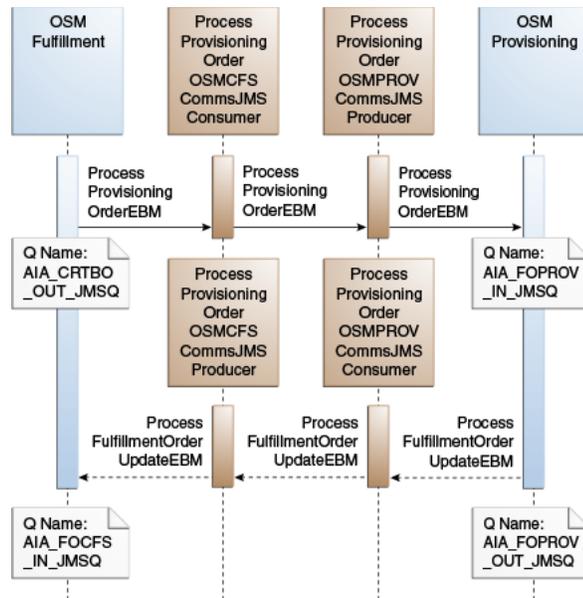
When this process is initiated, the following events occur:

1. Whenever a new order is created in Provisioning, a ProcessProvisioningOrderEBM message is created by the Oracle OSM Central Fulfillment System (Oracle OSM CFS) (for OSM COM). The message is enqueued in the AIA_CRTFO_OUT_JMSQ using the store and forward (SAF) mechanism.

2. The ProcessProvisioningOrderOSMCFSCommsJMSConsumer monitors the AIA_CRTFO_OUT_JMSQ queue and dequeues the message from the Oracle AIA queue using a JMS adapter service whenever a message exists. The message is then routed to the ProcessProvisioningOrderOSMPROVJMSProducer.

3. The routing mentioned in the previous step produces the message into the AIA_FOPROV_IN_JMSQ. SAF puts the message into Oracle OSM. Oracle OSM Provisioning then picks up the message from the queue and processes it accordingly.

4. During provisioning, one or more update messages are enqueued by Oracle OSM Provisioning into Oracle OSM WebLogic and eventually moves to AIA_FOPROV_OUT_JMSQ using SAF. The ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer dequeues the ProcessFulfillmentOrderUpdateEBM message and routes the message to the ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer.

5. The ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer then enqueues the ProcessFufillmentOrderUpdateEBM to the AIA_FOCFS_IN_JMSQ using SAF. Oracle OSM picks up this message to update the status of the order.

For more information about the events that occur when this process initiates, see Chapter 7, "Understanding the Process Integration for Order Lifecycle Management."

15.2.1 Defining Transaction Boundaries and Recovery Details

For this flow there are two transaction boundaries. Table 15–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer

- ProcessProvisioningOrderOSMPROVCommsJMSProducer

- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

Table 15–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessProvisioningOrderOSMCFSCommsJMSConsumer routes the message to ProcessProvisioningOrderOSMPROVCommsJMSProducer, which produces message into AIA_FOPROV_IN_JMSQ. SAF puts message in Oracle OSM.	None.	Rollback JMS message to originating queue AIA_CRTFO_OUT_JMSQ_ErrorQ	Resubmit the message from AIA_CRTFO_OUT_JMSQ_ErrorQ.
During provisioning, update messages are dequeued by Oracle OSM and eventually moves to AIA_FOPROV_OUT_JMSQ. ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer routes the message to ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer, which produces the message to AIA_FOCFS_IN_JMSQ.	None.	Rollback JMS message to the originating queue AIA_FOPROV_OUT_JMSQ_ErrorQ.	Resubmit the order from AIA_FOPROV_OUT_JMSQ_ErrorQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

15.3 Industry Oracle AIA Components

The process integration uses these industry components:

- ProvisioningOrderEBO

- ProcessProvisioningOrderEBM

- FulfillmentOrderEBO

- ProcessFulfillmentOrderUpdateEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry /Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

15.4 Integration Services

These services are delivered with this integration:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer

- ProcessProvisioningOrderOSMPROVCommsJMSProducer

- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

15.4.1 ProcessProvisioningOrderOSMCFSCommsJMSConsumer

For interacting with Oracle OSM Provisioning, OSM COM pushes ProcessProvisioningOrderEBM message into AIA_CRTFO_OUT_JMSQ using SAF.

The ProcessProvisioningOrderOSMCFSCommsJMSConsumer is a Mediator process that has a JMS Adapter Service. This Mediator service continuously polls the AIA_ CRTFO_OUT_JMSQ. The ProcessProvisioningOrderOSMCFSCommsJMSConsumer dequeues the ProcessProvisioningOrderEBM and routes it to the ProcessProvisioningOrderOSMPROVCommsJMSProducer.

This service has one operation: Consume_Message.

15.4.2 ProcessProvisioningOrderOSMPROVCommsJMSProducer

The ProcessProvisioningOrderOSMPROVCommsJMSProducer is a BPEL process that has a JMS Adapter Service. This BPEL process is responsible for pushing the ProcessProvisioningOrderEBM message into the AIA_FOPROV_IN_JMSQ using SAF. Oracle OSM Provisioning then consumes this message and processes it further.

This service has one operation: Initiate.

15.4.3 ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

For interacting with Oracle OSM COM, Oracle OSM Provisioning pushes ProcessFulfillmentOrderUpdateEBM message into AIA_FOPROV_OUT_JMSQ using SAF.

The ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer is a Mediator process with a JMS Adapter Service. The ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer dequeues the ProcessFulfillmentOrderUpdateEBM from AIA_FOPROV_OUT_JMSQ and routes it to ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer.

This service has one operation: Consume_Message.

15.4.4 ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

The ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer is a BPEL process that has a JMS Adapter Service. This BPEL process is responsible for pushing the ProcessFulfillmentOrderUpdateEBM Message into the AIA_FOCFS_IN_JMSQ. Using the SAF mechanism, it gets into the appropriate Oracle OSM queue.

This service has one operation: Initiate

16

OLM - Understanding the Update Sales Order Business Flow

This chapter provides an overview of the Update Sales Order business flow and discusses updating the sales order data and status, and describes how installed assets are created or updated in Siebel Customer Relationship Management (Siebel CRM).

This chapter includes the following sections:

- Section 16.1, "Update Sales Order Overview"
- Section 16.2, "Updating the Sales Order Data"
- Section 16.3, "Updating the Sales Order Status"
- Section 16.4, "Creating or Updating Installed Assets in Siebel CRM"

This business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

16.1 Update Sales Order Overview

The Update Sales Order business flow is used for two purposes:

To update sales order data:

Updating sales order data enables Oracle OSM to enrich the sales order with data coming from downstream systems, such as provisioning. An example of such data is the service instance ID in cases when the service instance ID is determined during service provisioning.

To update sales order status:

Updating sales order status enables Oracle OSM to send order and order line-level status updates to keep the customer service representative (CSR) and self-service customer updated on the progress made as the order is fulfilled. Oracle OSM optimizes the number of updates to Siebel CRM and limits updates to those that are significant to the Siebel CRM user.

16.2 Updating the Sales Order Data

When making data updates to an order line in Siebel CRM, an order management implementation must avoid sending the data updates before the order line reaches the point-of-no-return (PONR). If a revision is created before the data update is sent to Siebel CRM and then the revision is submitted, the data updates may be lost. Fulfillment flows in order management must delay sending data updates if possible

but no later than when the *Complete* status value is sent to Siebel CRM. If any data update occurs after the *Complete* status value is propagated to Siebel CRM, then the updated data is not saved for the asset.

These practices are the default behavior of the Oracle OSM cartridges for Oracle Application Integration Architecture (Oracle AIA).

For more information about assets, see Section 16.4, "Creating or Updating Installed Assets in Siebel CRM."

16.3 Updating the Sales Order Status

Oracle OSM facilitates configurable and streamlined order fulfillment statuses and propagation across the fulfillment systems and Siebel CRM. Decomposition of an order into order components and multiple fulfillment steps, places an extra burden on the order management system to manage the translation of fulfillment function responses to common status attribute values. Each response may contribute to different order line and order header status values, which are also the responsibility of the status management function of the order management system.

A single status attribute is not sufficient to provide comprehensive visibility into the fulfillment process. Oracle has adopted the extended set of attributes listed in the following table as part of its methodology to implement the Oracle Communications Order to Cash business process.

Table 16–1   Extended Set of Status Attributes

Functional Attribute Name	Usage
Order Header / Fulfillment Status	Updates Siebel CRM on the current status of order fulfillment at a high level. This is different from the Siebel Status attribute. The Fulfillment Status attribute tracks the order status while in fulfillment. Values can include *In Progress*, *Complete, Canceled, Failed*, and so on. The Status attribute tracks the order status across order capture and order fulfillment. Only *Complete* and *Canceled* fulfillment status values are reflected (internally by Siebel) on Status.
	Fulfillment status values are configurable by the implementer in the OSM cartridge.
Order Header / Status Context	Provides details about the current status. Implementers can configure this value. The Fulfillment Status attribute tracks the order status while in fulfillment. The Status attribute tracks the order status across order capture and order fulfillment. Only *Complete* and *Canceled* fulfillment status values are reflected (internally by Siebel) on Status.
Order Line / Fulfillment Status	Provides a high-level update of the current status of order line fulfillment to order management and Siebel CRM, such as I*n Progress, Complete, Canceled, Failed*, and so on.
	Fulfillment status values are configurable by the implementer in the OSM cartridge.
Order Line / Milestone	The last reached fulfillment milestone, such as *Shipped, Provisioned, Installed*, and so on.
	Milestone values are configurable by the implementer in the OSM cartridge.

Table 16–1   (Cont.) Extended Set of Status Attributes

Functional Attribute Name	Usage
Order Line / Status Context	Provides details about the current status. An implementer can configure this value. You can use Context Text to indicate: ■ Required customer interaction. ■ If delivery is expected to be delayed. ■ Milestone/fulfillment function in which a failure occurred. ■ Cause of a cancellation or who canceled an order.
Order Line / Point-of-no-return	Indicates if Siebel CRM should allow revisions to an order line or submission of previously created revisions to an order line. Oracle OSM fulfillment flows allow configuration of setting a hard point of no return (PONR) when a condition is met for a particular order line. When a hard PONR is established for an order line in Oracle OSM, an update is issued to reflect the same in Siebel CRM. Siebel CRM uses the PONR to block users from revising order lines.
Order Line / Actual Delivery Date-Time	Determines the date when the purchased product or service is considered available to the customer. This date may be the date physical goods are shipped, delivered, or their receipt acknowledged. For service-based products, this date is when the service is activated. This date is computed in the fulfillment flow.
Order Line / Expected Delivery Date-Time	Provides the expected delivery date for an order line. When Siebel CRM creates the order, the system provides this value by default. Oracle OSM uses this date to communicate changes for specific order line dates to Siebel CRM.

When referring to order or order line status in this guide, it is referring to values for all of the previous attributes. Some communications service providers (CSPs) do not realize the processing complexity that is introduced when different fulfillment status values are used for different services. You may be required to configure additional status values, but it is recommended you use a streamlined set of status values across product specifications. This practice has two advantages:

■ Enhances understanding for both the customer service representative (CSR) and the customer.

■ Maximizes fulfillment flow reusability and enhances the time to market.

In addition to using streamlined statuses, you can optimize the propagation of status changes by considering the following:

■ Not all status changes are relevant to the CSR or the customer in Siebel CRM. Therefore, do not propagate all changes to Siebel CRM.

■ Not all status changes must be reflected instantly, therefore, a throttling mechanism should be provided.

Some statuses, however, must be reflected instantly, such as point of no return (PONR) being reached. Careful analysis is required to determine which status

changes require instant propagation and which can wait. Too many status updates may cause performance and throughput problems.

- Some status attribute values drive specific logic in Siebel CRM and must be preserved. For Siebel CRM, these values are *Complete* and *Canceled*. Both affect the asset maintenance logic in Siebel CRM.

  The *Complete* status value drives the logic to create and update Siebel Assets. The order management implementation must turn the status value to *Complete* for a parent order line only after the order line and all of its subordinate order lines (within the order hierarchy) have completed fulfillment successfully.

  A *Canceled* order status excludes the order from a Siebel calculation of the future state of the asset when creating follow-on or future-dated orders.

16.4 Creating or Updating Installed Assets in Siebel CRM

An installed asset is initially created when a customer orders a new service and that order is fulfilled and asseted. From then on, if the customer requests a change to the existing services, the CSR initiates what is known as asset-based ordering. An asset-based order (also known as change order or MACD order) has references to an existing installed asset and actions indicating how it must be modified to match the customer's request. After a change order is fulfilled, the installed asset is updated to reflect the new desired state.

The process integration for order lifecycle management relies on Siebel CRM auto-asset functionality. Siebel CRM is configured so that assets are automatically created or updated when the order line status is set to *Complete*.

17

OLM - Update Sales Order Business Flow: Implementation

This chapter provides an overview of the Update Sales Order business flow and discusses the implementation of Siebel Customer Relationship Management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 17.1, "Update Sales Order Business Flow Overview"
- Section 17.2, "Updating Statuses from Oracle OSM to Siebel CRM Integration Flow"
- Section 17.3, "Siebel CRM Interfaces"
- Section 17.4, "Industry Oracle AIA Components"
- Section 17.5, "Integration Services"

17.1 Update Sales Order Business Flow Overview

This business flow is enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM) and Oracle Order and Service Management (Oracle OSM) pre-built integration options.

The following integration flow involves updating order statuses from Oracle OSM back to Siebel CRM.:

- Updating statuses from Oracle OSM to Siebel CRM

17.2 Updating Statuses from Oracle OSM to Siebel CRM Integration Flow

This integration flow uses the following interfaces:

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCSImpl

Figure 17–1 illustrates how to use Oracle OSM to send a sales order data and status update to Siebel:

Updating Statuses from Oracle OSM to Siebel CRM sequence diagram

Figure 17–1   Updating Statuses from Oracle OSM to Siebel CRM Sequence Diagram
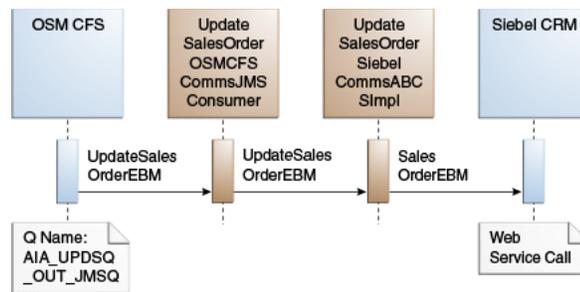
When this process is initiated, the following events occur:

1. Oracle OSM creates a message UpdateSalesOrderEBM and enqueues the message to the Oracle AIA queue called AIA_UPDSO_OUT_JMSQ using SAF. The UpdateSalesOrderOSMCFSCommsJMSConsumer consumes this message and routes the message to UpdateSalesOrderSiebelCommsProvABCSImpl.

> **Caution:**   The UpdateSalesOrderOSMCFSCommsJMSConsumer also has a sequencer. In other words, if any update to Siebel causes a system or business error, any further updates to the account does not happen until the error is fixed. All updates for that account are locked in the sequencer table. If the error is a business error then the message must be removed from the sequencer table and if the error is a system error then the message must be resubmitted.

---

2. This process converts the enterprise business message (EBM) into a Siebel ABM and invokes the Siebel web service to update the order.

   Oracle AIA always copies fulfillment status to the DeliveryStatus ABM attribute. Internally, Siebel CRM reflects the end state status values (*Canceled* and *Complete*) on the Status field. The DeliveryStatus is mapped in the Siebel UI to the Fulfillment Status.

   In other words, the Status field in the Siebel UI represents the overarching status throughout order capture and order fulfillment. Fulfillment Status is a sub-status to Status = Open. Fulfillment Status indicates the status of the order in OSM COM while the order is in being fulfilled.

   For more information about using sequencing logic to make updates to Siebel, see Section 8.2, "Supporting Order Priorities" and Appendix I, "Using the Oracle Mediator Resequencer Feature."

17.2.1 Defining Transaction Boundaries and Recovery Details

For this flow there are two transaction boundaries. Table 17–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

If any update to Siebel causes a system or business error, any further updates to the account does not happen until the error is fixed. All updates for that account are locked in the sequencer table. If the error is a business error then the message must be removed from the sequencer table and if the error is a system error then the message must be resubmitted.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderOSMCFSCommsJMSConsumer_RS
- UpdateSalesOrderSiebelCommsProvABCSImpl

Table 17–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
UpdateSalesOrderOSMCFSCommsJMSConsumer consumes the message and puts it in the sequencer table defined at the Routing Service UpdateSalesOrderOSMCFSCommsJMSConsumer_RS.	Message goes into the sequencer table.	Rollback JMS message to AIA_UPDSO_OUT_JMSQ_ErrorQ	Resubmit the order from AIA_UPDSO_OUT_JMSQ_ErrorQ.
UpdateSalesOrderOSMCFSCommsJMSConsumer_RS routes the message to UpdateSalesOrderSiebelCommsProvABCSImpl, which invokes the Siebel web service to update the order.	AIA cross-reference entries.	Rollback the message to the sequencer table.	Resubmit the order from the sequencer table.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

17.3 Siebel CRM Interfaces

This integration flow uses the following Siebel interfaces:

- SWIOrderUpsert
- SWIOrderUpsertSubProcess

These are inbound Siebel web services used to update the order information back to Siebel CRM.

For more information about web services, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

17.4 Industry Oracle AIA Components

These integration flows use these industry components:

- SalesOrderEBO
- UpdateSalesOrderEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_
HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry
/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference
Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA
Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application
Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise
Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are
protected and remain intact after a patch or an upgrade, so long as the extensibility
guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle
Application Integration Architecture Foundation Pack*, "Working with AIA Design
Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

17.5 Integration Services

The following services are delivered with these integration flows:

- UpdateSalesOrderOSMCFSCommsJMSConsumer

- UpdateSalesOrderSiebelCommsProvABCSImpl

Some of these services have been enabled to use Session Pool Manager.

For more information about Session Pool Manager, see *Oracle Application Integration
Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

17.5.1 UpdateSalesOrderOSMCFSCommsJMSConsumer

The UpdateSalesOrderOSMCFSCommsJMSConsumer is a Mediator process with a
JMS Adapter Service. This process dequeues the UpdateSalesOrderEBM message from
the AIA_UPDSO_OUT_JMSQ.

This service has one operation: Consume_Message.

17.5.2 UpdateSalesOrderSiebelCommsProvABCSImpl

The UpdateSalesOrderSiebelCommsProvABCSImpl is a BPEL process with one
operation: UpdateSalesOrder. It accepts the UpdateSalesOrderEBM as the input from
the UpdateSalesOrderOSMCFSCommsJMSConsumer, and uses the order information
in the input message to update the orders in Siebel CRM.

The main functions of this service are:

- Updating the order line status: updates the order line status back to Siebel CRM.

- Enriching the order: enriches the information back to Siebel CRM from a central
  fulfillment system to facilitate customer care, service, and asset-based ordering. It
  is also used to update or enrich the order line items with fulfillment attributes
  back to Siebel CRM. Among these attributes are service IDs and allocated
  resources such as port number and IP address.

- Updating the order header: enriches the order header to Siebel CRM.

This process is an asynchronous, one-way service.

This service is SPM enabled.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

18

Understanding the Process Integration for Customer Management

This chapter provides an overview of the customer management process integration and describes the synchronize customer account and synchronize customer special rating profile business flows.

This chapter includes the following sections:

- Section 18.1, "Customer Management Process Integration Overview"
- Section 18.2, "Solution Assumptions and Constraints"
- Section 18.3, "Data Requirements"
- Section 18.4, "Synchronize Customer Account Business Flow"
- Section 18.5, "Synchronize Customer Special Rating Profile Business Flow"

These business flows are enabled using the Oracle Communications Order to Cash Siebel Customer Relationship Management (Siebel CRM) and the Oracle Billing and Revenue Management (Oracle BRM) pre-built integration options.

18.1 Customer Management Process Integration Overview

The process integration for customer management enables the synchronization of customer information between Siebel CRM and Oracle BRM. Customers are created in Siebel CRM and sent to Oracle BRM. Siebel CRM is the customer master. Customer data updated in Siebel CRM is synchronized to Oracle BRM through the customer management process integration and is a one-way synchronization process.

The process integration for customer management provides the following integration flows, which enable the Synchronize Customer Account and Synchronize Customer Special Rating Profile business flows.

Synchronize Customer Account Business Flow
- The create/sync customer account integration flow interfaces customers to Oracle BRM (as part of the Order Management processing flow).

  For more information about the Order Management processing flow, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."

- The update customer account integration flow, which updates account information (such as address, name, contact, and status) from Siebel CRM to Oracle BRM.

Synchronize Customer Special Rating Profile Business Flow

■ This flow synchronizes friends and family list updates to Oracle BRM.

18.2 Solution Assumptions and Constraints

Here are the solution assumptions for the process integration for customer management:

1. Siebel CRM is the customer master and manages all aspects of the lifecycle from creation to updates for a customer. The customer management process integration is a uni-directional flow from Siebel CRM to Oracle BRM.

2. Initial loading of customer data is not supported for this release.

3. An order line can have only one bill-to account.

4. If order line items reference a service account that is different from the billing account, then the Siebel CRM billing account is propagated as a paying account in Oracle BRM, while the Siebel CRM service account is propagated as a non-paying sub-ordinate account in Oracle BRM.

5. Customer accounts and billing profiles are first synchronized to Oracle BRM during order processing, and not before.

6. Once synchronized to a particular billing system, a customer account is kept synchronized by real-time integration flows.

   The Customer Account Sync integration that occurs during order processing can assume that if an account has been created in Oracle BRM, it is current and up-to-date.

7. The Customer Account Sync process during order processing synchronizes accounts to one billing system/instance (Oracle BRM) at a time. The order management system can synchronize the same customer to additional billing systems/instances by calling the Customer Account Sync service multiple times.

   For more information about configuring multiple billing instances, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."

8. The Siebel CRM account hierarchy is not synchronized to Oracle BRM. Instead, the billing account and service account relationship on a Siebel order line is sent to Oracle BRM as a parent account and child account, respectively. Oracle BRM supports a single parent for a child account.

18.3 Data Requirements

The process integration for customer management requires the following data to successfully create customer data in Oracle BRM:

■ Accounts must be of type *Residential* or *Business* and the account class must be *Customer, Service*, or *Billing*.

■ In Siebel CRM, accounts can have any number of contacts or addresses associated with them, but account creation in Oracle BRM requires:

   – The primary contact (must be explicitly set) and address for the account.

   – The contact and address that is associated with the billing profile that is used in the order.

   – For an account's primary address, the city, state, country, and zip code.

- For an account's primary contact, the last name.

- For a bill profile, all bill profiles that are synchronized for an account and its related parent and child accounts must have the same value for Bill Frequency.

- For a bill profile address, the city, state, and zip code.

- For a credit card bill profile, the credit card number, expiration month and year, and cardholder name are required. Card verification value (CVV) number is optional.

- For an automatic debit bill profile, the bank routing number and account number are required.

18.4 Synchronize Customer Account Business Flow

Account information is captured at the beginning of the order process. When a customer places an order, the first step of the process is to determine whether the customer is new or existing. If this is an existing customer, the customer record can be found and selected, and the customer order details are captured. If this is a new customer, a new account is created.

The billing preferences (bill medium, bill frequency, payment type, billing type, billing contact, bill cycle data, and so on) are also captured. After the account information is captured, the order details are captured. The order is submitted to the order management system for processing. Customer data is created in billing as part of the Order Fulfillment flow.

For more information about the Order Fulfillment flow, see Chapter 10, "OLM - Understanding the Synchronize Fulfillment Order Billing Account Business Flow."

Subsequently, customers can call in to request changes to their contact information, address, and so on. These changes and updates to other attributes are supported through the Update Customer Account integration flow.

18.4.1 Create/Sync Account Integration Flow

Figure 18–1 illustrates the overall flow for the create/sync customer account integration flow.

Figure 18–1   Create/Sync Customer Account Overall Flow
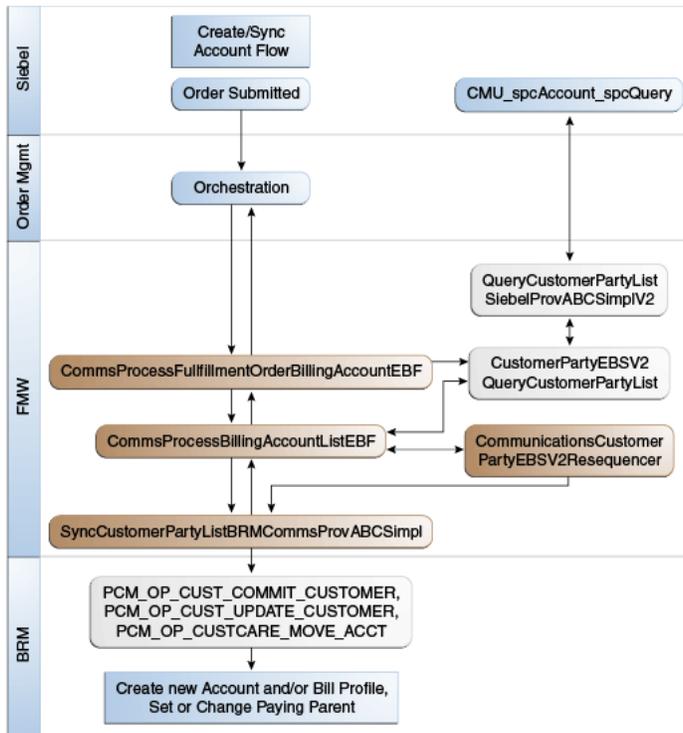
Table 18–1 provides information on Siebel CRM attributes mapped to Oracle BRM as part of the create/sync account integration flow.

Table 18–1   Siebel Entities Created or Synchronized to Oracle BRM

Siebel Entity/Attributes (as labeled in Siebel UI)	BRM Entity/Attributes (as labeled in Oracle BRM Customer Center)	Comments
Account	**Account**	--
	Account Number	Integration sets this to the *Common ID*.
Account Type	Business Type	Only Siebel Account Type of *Residential* or *Business* is supported.
		Uses the CUSTOMERPARTY_TYPECODE DVM.
Name	Company Name	Only set for Account Type of *Business.*
Currency	Currency	Uses the CURRENCY_CODE DVM.
--	--	**Notes:**
		Integration does not explicitly set the status when creating the customer account in Oracle BRM. Oracle BRM defaults the status to *Active*.
		The integration creates a two-level hierarchy in Oracle BRM with a paying parent and a subordinate service account when the billing account and the service account on the order line are different.
		For more information, see Section 10.1, "Synchronize Fulfillment Order Billing Account Overview."
Contact	--	The integration only syncs the primary contact that is tied to the Account in Siebel CRM to Oracle BRM.

Table 18–1 (Cont.) Siebel Entities Created or Synchronized to Oracle BRM

Siebel Entity/Attributes (as labeled in Siebel UI)	BRM Entity/Attributes (as labeled in Oracle BRM Customer Center)	Comments
Mr/Mrs	Salutation	Uses the CONTACT_SALUTATION DVM.
First Name	First Name	--
Last Name	Last Name	--
Phone	Phone Number	The integration maps different Siebel CRM phone number types (home, work, fax, mobile) to Oracle BRM Phone Type and Number using the PHONENUMBER_TYPE DVM.
		The phone number format should match the supported format in Oracle BRM.
		For more information about phone number formats, see *Oracle Communications Billing and Revenue Management Concepts*, "Using BRM with Oracle Application Integration Architecture", Validating Customer Contact Information.
Job Title	Job Title	--
Email	Email	--
Address	--	The integration only synchronizes the primary address that is tied to the Account in Siebel CRM to Oracle BRM.
Address	Address	In addition to *Address*, the following fields are also mapped:
		City, State, Postal Code, Country.
		Uses the following DVMs:
		ADDRESS_COUNTRYID, ADDRESS_COUNTRYSUBDIVID, PROVINCE, STATE.
Billing Profile	**BillInfo**	--
Name	Name	--
Frequency	Billing Frequency in Months	Uses the CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE DVM
--	Currency	Integration passes account-level currency.
		Uses the CURRENCY_CODE DVM
Billing Schedule	Billing Day of Month	If the Billing Schedule is not set in and sent from Siebel CRM, then Oracle BRM defaults the Billing Day of Month.
		For more information about the billing schedule, see the *Oracle Communications Billing and Revenue Management Configuring and Running Billing Guide*, "Setting Business Policies for Billing."
--	**PayInfo**	--
Payment Method	Payment Method	Only *Bill Me*, *Credit Card* or *Auto-Debit* is supported.
		Uses the CUSTOMERPARTY_PAYPROFILE_PAYMETHODECODE DVM.

Table 18–1   (Cont.) Siebel Entities Created or Synchronized to Oracle BRM

Siebel Entity/Attributes (as labeled in Siebel UI)	BRM Entity/Attributes (as labeled in Oracle BRM Customer Center)	Comments
Contact Last Name, First Name	Name	When the payment method is *Bill Me*, the Contact Name on the Siebel Billing profile is mapped to Oracle BRM PayInfo Contact Name.
		When the payment method is *Credit Card* or *Auto-Debit*, either the Credit Card owner name or Debit Account name is mapped to Oracle BRM PayInfo Contact Name.
Bill Media	Delivery Preference	Applicable only when the payment method is *Bill Me*.
		Uses the CUSTOMERPARTY_PAYPROFILE_ DELIVERYPREF DVM.
Email Bill To	Email Address	Applicable only when the payment method is *Bill Me*.
Address	Address	In addition to *Address*, the following fields are also mapped:
		City, State, Postal Code, Country.
		Uses the following DVMs:
		ADDRESS_COUNTRYID, ADDRESS_ COUNTRYSUBDIVID, PROVINCE, STATE.
Credit Card #	Credit Card Number	Applicable only when the payment method is *Credit Card.*
Expiration Month & Year	Credit Card Exp	Applicable only when the payment method is *Credit Card.*
Security Code	Security ID	Applicable only when the payment method is *Credit Card.*
Account #	Debit Num	Applicable only when the payment method is *Auto-Debit*.
Bank Routing #	Bank No	Applicable only when the payment method is *Auto-Debit*.
Bank Account Type	Type	Applicable only when the payment method is *Auto-Debit*.

18.4.2 Update Customer Account Integration Flow

Customers can call in to make changes to their account information. The customer service representative (CSR) uses Siebel CRM as the front-end application to capture these customer data updates. The Customer Management process integration synchronizes these customer updates to Oracle BRM through the update customer account integration flow.

> **Note:**   Updates are synchronized to Oracle BRM only for accounts that have been created through the order fulfillment flow.

Over time customer attributes such as name, address, contact information, billing, and payment information can change. As and when customer data is changed in Siebel CRM, the process integration ensures that these changes are synchronized to Oracle BRM in real time, thereby ensuring the customer data is both consistent and current between both the applications.

A provision exists for optionally synchronizing account status updates from Siebel CRM to Oracle BRM. For more information about the synchronization of the account status, see Section 18.4.2.1, "Account Status Synchronization Methodology."

Figure 18–2 illustrates the overall flow for the update customer account integration flow.

Figure 18–2   Update Customer Account Overall Flow
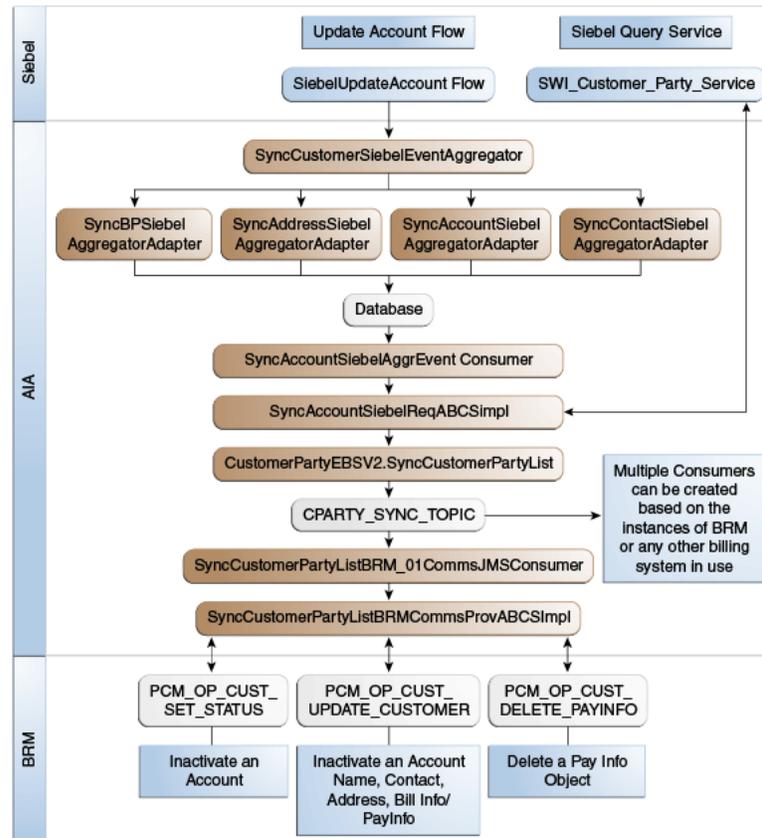
18.4.2.1 Account Status Synchronization Methodology

The account status synchronization feature enables propagation of account status changes from Siebel CRM to Oracle BRM.

As delivered, the account status propagation to Oracle BRM is disabled. If implementers choose to use this feature, they must explicitly enable it by changing a configuration setting in the AIAConfiguration.xml file.

For more information about this configuration setting, see the EnableAccountStatusSync property in Section 26.6, "Configuring the Process Integration for Customer Management."

The account status synchronization feature is designed as part of the collections process integration and should ideally be used with it and not as an independent or standalone feature.

For more information about collections, see the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*

To support collections, the integration synchronizes collection actions generated by Oracle BRM as credit alerts in Siebel CRM. Various actions such as notifying the customer regarding unpaid dues or suspending or canceling services due to delinquency are delegated to Siebel.

Siebel can be extended to automate the generation of change orders for suspending or canceling services based on the generated credit alerts. Alternatively, the Siebel collection agent can manually submit change orders for suspending or canceling services. Either of these approaches ensures that changes in the service asset state are communicated correctly to Oracle BRM and both the applications are synchronized considering the service state.

As part of the collections lifecycle, if the customer continues to be delinquent and must be written off and his account inactivated, this feature (if enabled) ensures that the account status change in Siebel CRM is propagated to Oracle BRM.

It is recommended that the account in Siebel CRM be inactivated only after all the services (and account-level subscription products) have been canceled. This is because inactivating an account in Siebel CRM that has active services propagates that account status change to Oracle BRM resulting in the cancellation of services in Oracle BRM. This is because Oracle BRM cascades the status change from the account to all its bill-infos and services and products. An important practice is to inactivate the account in Siebel CRM only after all the services (and account-level subscription products) have been canceled (the cancellation orders fulfilled and asseted).

As delivered, Siebel does not have logic to restrict changes to account status. Therefore, it is also recommended that the ability to inactivate an account be restricted to authorized users and roles in Siebel CRM because inadvertently inactivating accounts with active services can result (when the account status propagation is enabled) in those services being canceled in Oracle BRM.

18.5 Synchronize Customer Special Rating Profile Business Flow

Once a service that supports special rating has been purchased and the order fulfilled and asseted, the customer can use the Siebel Special Rating Profile to make changes to their friends and family list. Updates are then synchronized to Oracle BRM.

The Synchronize Customer Special Rating Profile business flow uses the operation ProcessInstalledProductSpecialRatingSetList on the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl composite for this purpose. The specification group on the installed product enterprise business message (EBM) is used to communicate the list entries.

For more information about purchasing services that support special rating, see Section 12.7, "Supporting Friends and Family Lists."

19

CM - Synchronize Customer Account: Implementation

This chapter describes the create/sync customer account and update customer accounts integration flows and discusses Siebel Customer Relationship Management (Siebel CRM) and Oracle Billing and Revenue Management (Oracle BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 19.1, "Create/Sync Customer Account Integration Flow"
- Section 19.2, "Update Customer Accounts Integration Flow"
- Section 19.3, "Oracle BRM Interfaces"
- Section 19.4, "Siebel CRM Interfaces"
- Section 19.5, "Industry Oracle AIA Components"
- Section 19.6, "Integration Services"

The Synchronize Customer Account business flow is enabled using the Oracle Communications Order to Cash Siebel CRM and Oracle BRM pre-built integration options.

19.1 Create/Sync Customer Account Integration Flow

The *Create/Sync Customer Account* integration flow enables the synchronization of customer information from Siebel CRM to Oracle BRM. This flow is called during the *Interfacing Orders to Create Customer Data in Oracle BRM* integration flow

For information about the sequence of events for these integration flows, see Section 11.2, "Interfacing Orders to Create Customer Data in Oracle BRM."

19.2 Update Customer Accounts Integration Flow

This flow is initiated to propagate updates to accounts in Siebel CRM to Oracle BRM.

This integration flow uses the following interfaces:

- SyncCustomerSiebelEventAggregator
- SyncAcctSiebelEventAggrConsumer
- SyncAccountSiebelReqABCSImpl
- CustomerPartyEBSV2

- SyncCustomerPartyListBRM_01CommsJMSConsumer

- SyncCustomerPartyListBRMCommsProvABCSImpl

Figure 19–1 illustrates the update customer accounts flow.

Figure 19–1   Update Customer Account Flow Sequence Diagram
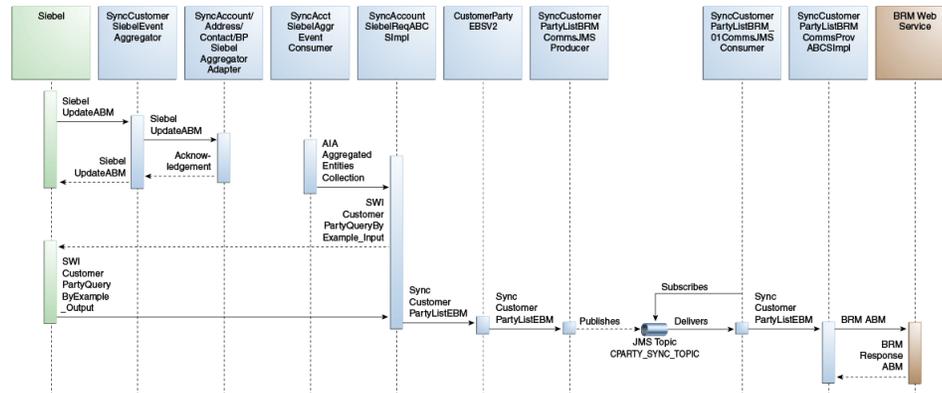
When this process is initiated, the following events occur:

1. In Siebel CRM, a user navigates to the Accounts screen, queries an account, and updates an account attribute (for example, address, contact, or the billing profile).

    This causes Siebel CRM to invoke the SyncCustomerSiebelEventAggregator, with the SiebelUpdateABM message containing the details of the account that has been updated. Depending on the type of update, one of four kinds of Siebel messages can be generated: ListOfSWICustomerIO, ListOfSWIBillingProfileIO, ListOfSWIContactIO, or ListOfSWIAddressIO.

2. The SyncCustomerSiebelEventAggregator then calls a database adapter (account, address, contact, or billing profile) that executes a pl/sql script that extracts and stores the relevant IDs (for example, account, contact, or billing profile) in a database table AIA_AGGREGATED_ENTITIES and sends an acknowledgment to Siebel CRM as a reply.

3. The IDs in the database table are stored in the same manner as the hierarchy of IDs is maintained (for example, BillingProfileID is always the child of some account ID).

    The account ID, along with its entire child IDs, is picked up from the database table by the SyncAcctSiebelEventAggrConsumer process. This consumer is sequencing-enabled to ensure that the updates for the same customer are sent in the appropriate sequence.

4. The Consumer process then calls the SyncAccountSiebelReqABCSImpl process.

    This process takes all the IDs, constructs a Siebel Query Input ABM, and calls the Siebel Query web service to get the entire account data from Siebel CRM. After getting the data, the Siebel Query Input ABM is transformed into the SyncCustomerPartyListEBM and routed to SyncCustomerPartyListBRMCommsJMSProducer through CustomerPartyEBSV2, which publishes the message to the JMS topic named CPARTY_SYNC_TOPIC.

5. Depending on the instances of Oracle BRM or any other billing system, consumers can be defined that have subscribed to the CPARTY_SYNC_TOPIC topic.

One such consumer for the default implementation is available, named SyncCustomerPartyListBRM_01CommsJMSConsumer, which listens to the topic for messages, picks up the arriving message, and passes it on to the process SyncCustomerPartyListBRMCommsProvABCSImpl after duly checking whether the message should go to the ensuing provider ABCS, and accordingly stamping the target ID.

6. The SyncCustomerPartyListBRMCommsProvABCSImpl process then calls the PCM_OP_CUST_UPDATE_CUSTOMER, PCM_OP_CUST_DELETE_PAYINFO, or PCM_OP_CUST_SET_STATUS opcode as required to synchronize the updated data to Oracle BRM.

19.3 Oracle BRM Interfaces

Table 19–1 lists the Oracle BRM interfaces used by the Synchronize Customer Account business flow.

Table 19–1   Oracle BRM Interfaces

API / Opcode	Description	Used by
PCM_OP_CUST_COMMIT_CUSTOMER	Create a new account with one or more bill-infos and pay-infos in Oracle BRM.	Oracle BRM Sync Customer Provider application business connector service (ABCS) as part of the Create/Update Customer Account flow.
PCM_OP_CUST_UPDATE_CUSTOMER	Update account information (name, address, phone), contact information, and billing and pay information.	Oracle BRM Sync Customer Provider ABCS as part of the Update Customer Account flow.
PCM_OP_CUST_DELETE_PAYINFO	Delete a payinfo object from an account.	Oracle BRM Sync Customer Provider ABCS, as part of the Update Customer Account flow.
PCM_OP_CUSTCARE_MOVE_ACCT	Move an account to a new parent account.	Oracle BRM Sync Customer Provider ABCS, as part of Order flow to manage paying parent changes (account hierarchy change).
PCM_OP_CUST_SET_STATUS	Used to modify the account status in Oracle BRM.	Oracle BRM Sync Customer Provider ABCS as part of the Update Customer Account flow.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "BRM Documentation," Reference, API reference, PCM opcode libraries.

19.4 Siebel CRM Interfaces

Siebel CRM Web Service Interfaces

Table 19–2 describes the Siebel CRM web service interface.

Table 19–2   Siebel CRM Web Service Interface

Web Service	Description	Used by
Query Account- (SWICustomerParty)	Retrieves account, bill profile, contact, and address data from Siebel CRM.	Used by the Siebel Query Account Provider ABCS as part of creating and adding a new billing profile to an existing customer. Also used by the Siebel Sync Account Requester ABCS to retrieve the most current account data from Siebel.

For more information about web services, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

Siebel CRM Workflow Event Interfaces

describes the Siebel CRM workflow event interfaces.

Table 19–3   Siebel CRM Workflow Event Interfaces

Event	Description	Consumed by
SWI Account Updated	This workflow event is started when an account is updated in Siebel CRM.	This event message is consumed by the SyncCustomerSiebelEventAggregator. aggregateaccountevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ ENTITIES).
SWI Bill Profile Updated	This workflow event is started when a bill profile is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator. aggregatebpevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ ENTITIES).
SWI Contact Updated	This workflow event is started when a contact is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator. aggregatecontactevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ ENTITIES).
SWI Address Updated	This workflow event is started when an address is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator. aggregateaddressevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ ENTITIES).

For more information, *see Siebel Order Management Guide Addendum for Communications*, "Workflows for Employee Asset-Based Ordering."

19.5 Industry Oracle AIA Components

This is the list of the enterprise business objects (EBOs) and enterprise business messages (EBMs) used by the process integration for customer management:

CustomerPartyEBO

QueryCustomerPartyListEBM

QueryCustomerPartyListResponseEBM

SyncCustomerPartyListEBM

SyncCustomerPartyListResponseEBM

ProcessBillingAccountListEBM

ProcessBillingAccountListResponseEBM

FulfillmentOrderEBO

ProcessFulfillmentOrderBillingAccountListEBM

ProcessFulfillmentOrderBillingAccountListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Commu nications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry /Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

19.6 Integration Services

These services are delivered with the customer management integration flow:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommunicationsCustomerPartyEBSV2Resequencer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- SyncCustomerSiebelEventAggregator
- SyncAccountSiebelAggregatorAdapter
- SyncContactSiebelAggregatorAdapter

- SyncAddressSiebelAggregatorAdapter

- SyncBPSiebelAggregatorAdapter

- SyncAcctSiebelAggrEventConsumer

- SyncAccountSiebelReqABCSImpl

- CustomerPartyEBSV2

- QueryCustomerPartyListSiebelProvABCSImplV2

- SyncCustomerPartyListBRMCommsProvABCSImpl

- SyncCustomerPartyListBRM_01CommsJMSConsumer

19.6.1 ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer

This process listens to the AIA_CRTCUST_OUT_JMSQ JMS queue and as soon as a message is picked up, forwards it to the CommsProcessFulfillmentOrderBillingAccountListEBF enterprise business flow, which extracts the relevant customer data. The ProcessBillingAccountListEBM is then routed to the CommsProcessBillingAccountListEBF.

19.6.2 CommunicationsCustomerPartyEBSV2Resequencer

The CommunicationsCustomerPartyEBSV2Resequencer enterprise business service sequences the account message from CommsProcessBillingAccountListEBF. The messages are grouped by account ID. This process receives the customer EBM and passes it to SyncCustomerPartyListBRMCommsProvABCSImpl for routing to Oracle BRM.

For more information about this EBS, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Constructing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services," Enterprise Business Flow Processes.

For more information about the Oracle Mediator Resequencer, see Appendix I, "Using the Oracle Mediator Resequencer Feature."

19.6.3 CommsProcessFulfillmentOrderBillingAccountListEBF

This enterprise business flow (EBF) extracts the Customer Data from OrderEBM. The process loops through every order line and extracts any customer account or billing profile that it encounters.

This service has two operations. One accepts the ProcessFulfillmentOrderBillingAccountListEBM and is used by the process to order data. The other is used by the process to send the response back to the calling process (using the ProcessFulfillmentOrderBillingAccountListEBM).

The transformations include:

- ProcessFulfillmentOrderBillingAccountList to ResponseEBM.xsl

- ProcessFulfillmentOrderBillingAccountListEBM to ProcessBillingAccountListEBM.xsl

The CommsProcessFulfillmentOrderBillingAccountListEBF enterprise business flow is implemented as an asynchronous delayed response Business Process Execution Language (BPEL) process.

For more information about EBFs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Constructing Enterprise Business Flows" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services," Enterprise Business Flow Processes.

19.6.4 CommsProcessBillingAccountListEBF

This EBF service creates or synchronizes all the customer accounts and billing profiles in an appropriate billing system. The Order Processing integration flow invokes this service with a list of customer account IDs, billing profile IDs, and the target system ID. When the process is complete, a response is sent back to the order flow confirming that all accounts have been set up in the target billing system, and the order processing can continue.

This service provides two operations. One accepts the ProcessBillingAccountListEBM and is used by the process to accept the customer data to be synchronized. The other one is used by the process to send the response back to the calling process (using the SyncCustomerPartyListResponseEBM). The data area of the message contains one or more customer account IDs. For each account, one or more bill profile IDs must be synchronized to the target billing system. The customer data indicates both the hierarchical and the paying relationships between the accounts.

This service creates or synchronizes one or more customers (identified by ID only) and their billing profiles to a particular target billing system (identified in the EBM header).

Therefore, the responsibilities of this service include:

- Determining whether the customer exists and is up to date in the target billing system.

  If so, optimize and do not try to create or synchronize.

- Retrieving the customer data from the appropriate Siebel CRM system using the provided IDs, if necessary.

- Optimizing, if possible, the number and size of queries back into Siebel CRM for the customer data.

- Creating or updating the customers and billing profiles in the target billing system, reflecting the customer hierarchy and paying relationships among the customers.

For more information about EBFs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Constructing Enterprise Business Flows" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services," Enterprise Business Flow Processes.

19.6.5 SyncCustomerSiebelEventAggregator

This service is responsible for receiving Siebel CRM update account events and collating them into an Oracle AIA database table.

Figure 19–2 illustrates the relationship of the SyncCustomerSiebelEventAggregator with the other services in the integration flow.

Figure 19–2 SyncCustomerSiebelEventAggregator
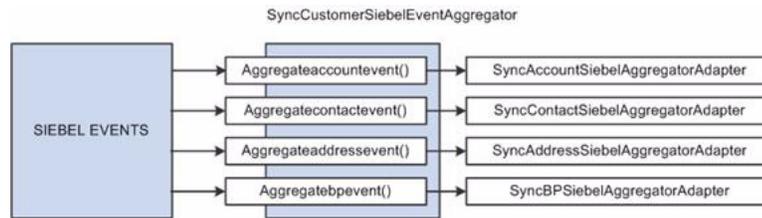
This service provides four operations, one for each of the object types that are updated:

- Aggregateaccountevent:

  Receives the Account Updated Siebel message. Extracts the account ID, contact IDs, and address IDs from the message. Invokes the SyncAccountSiebelAggregatorAdapter to store these IDs into the AIA_ AGGREGATED_ENTITIES database table.

- Aggregatecontacttevent:

  Receives the Contact Update Siebel message. Extracts the account IDs, bill profile IDs, and contact IDs from the message. Invokes the SyncContactSiebelAggregatorAdapter to store these IDs in the AIA_ AGGREGATED_ENTITIES database table.

- Aggregateaddresstevent:

  Receives the Address Update Siebel message. Extracts the account IDs, bill profile IDs, and address IDs from the message. Invokes the SyncAddressSiebelAggregatorAdapter to store these IDs into the AIA_ AGGREGATED_ENTITIES database table.

- Aggregatebpevent:

  Receives the BillingProfile Updated Siebel message. Extracts the BillingProfile ID and the associated account ID from the message. Invokes the SyncBPSiebelAggregatorAdapter to store these IDs in the AIA_AGGREGATED_ ENTITIES database table.

For more information about the Event Aggregation programming model, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Describing the Event Aggregation Programming Model."

19.6.6 SyncAccountSiebelAggregatorAdapter

This service aggregates the account events generated in Siebel CRM when an account is created or updated. This service invokes a PL/SQL procedure, AIA_ AGGREGATOR_PUB.SIEBEL_AGGREGATE_ACCOUNT, which does the actual aggregation in the AIA aggregator table.

19.6.7 SyncContactSiebelAggregatorAdapter

This service aggregates the account events generated in Siebel CRM when an account is created or updated. This service invokes a PL/SQL procedure, AIA_ AGGREGATOR_PUB.SIEBEL_AGGREGATE_CONTACT, which does the actual aggregation in the AIA aggregator table.

19.6.8 SyncAddressSiebelAggregatorAdapter

This service aggregates the account events generated in Siebel CRM when an account is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_ADDRESS, which does the actual aggregation in the AIA aggregator table.

19.6.9 SyncBPSiebelAggregatorAdapter

This service aggregates the account events generated in Siebel CRM when an account is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_BP, which does the actual aggregation in the AIA aggregator table.

19.6.10 SyncAcctSiebelAggrEventConsumer

This service extracts the account IDs stored in the AIA_AGGREGATED_ENTITIES database table and sends them forward to the SyncAccountSiebelReqABCSImpl service.

Sequencing is enabled for this service. When this consumer calls the requestor for further processing and the requestor fails, any subsequent update for that customer is not processed until proper action is taken on the messages in the sequencer. If the failure is due to a business error then messages must be removed from the queue for the subsequent messages to process. If the failure is system related then messages in the resequencer can be retried to move the message from the resequencer queue and thereby enabling subsequent messages to be processed. Any updates for other errors are processed as usual.

For more information about the resequencer, see Appendix I, "Using the Oracle Mediator Resequencer Feature."

19.6.11 SyncAccountSiebelReqABCSImpl

This service is responsible for transforming the Siebel message into the SyncCustomerPartyList EBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

The process checks whether the incoming message has a target system identifier. If the target system identifier is not present, then the delivered rule assumes multiple Oracle BRM systems and routes the incoming requests to a Java message service (JMS) producer service SyncCustomerPartyListBRMCommsJMSProducer::Produce_Message.

If the implementing customer does not have multiple Oracle BRM systems and operates only on a single Oracle BRM system, then they can change the routing rule to route incoming requests to the SyncCustomerPartyListBRMCommsProvABCSImpl_1_0::SyncCustomerPartyList directly. Additionally, customers must apply a transformation before routing to stamp the target system identifier in the EBM. The transformation file name is esb:///ESB_Projects/Customer_CustomerPartyEBSV2/AddTargetID_BRM01.xsl.

19.6.12 CustomerPartyEBSV2

CustomerPartyEBSV2 exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object.

CustomerPartyEBSV2 service uses the following operations:

- SyncCustomerPartyList

- QueryCustomerPartyList

Figure 19–3 illustrates the relationship of QueryCustomerPartyListSiebelProvABCSImplV2 with the other services in the integration flow.

Figure 19–3   CustomerPartyEBSV2
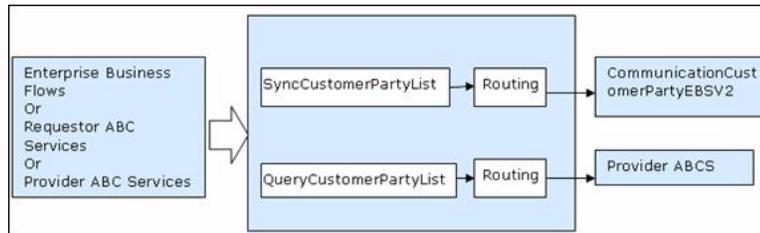
The CustomerPartyEBSV2 is implemented as a lightweight EBS routing service.

For more information about this EBS, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services," Enterprise Business Flow Processes.

19.6.13 QueryCustomerPartyListSiebelProvABCSImplV2

CustomerPartyEBSV2 invokes the QueryCustomerPartyListSiebelProvABCSImplV2 service when the routing rules determine that Siebel CRM is to be the service provider for the QueryCustomerPartyList EBS operation.

This service has one synchronous request and reply operation, QueryCustomerPartyList.

19.6.14 SyncCustomerPartyListBRMCommsProvABCSImpl

The CommsProcessBillingAccountListEBF or SyncAccountSiebelReqABCSImpl service invokes SyncCustomerPartyListBRMCommsProvABCSImpl. It performs the following actions:

1. Receives the SyncCustomerPartyListEBM.

2. Loops through each data area:

    If the current account is a child account, it checks whether the parent account has been synchronized. The child account should be synchronized only when the parent has been synchronized.

    Based on the action code associated with each account, it goes to the Create block (used when a new account must be synchronized) or the Update block (used when an existing account is to be updated).

3. Creates the block:

    Transforms the SyncCustomerPartyListEBM to the Oracle BRM-specific ABM (PCM_OP_CUST_COMMIT_CUSTOMER_Inmsg).

    Calls the PCM_OP_CUST_COMMIT_CUSTOMER opcode with the Oracle BRM ABM.

Transforms the response from the PCM_OP_CUST_COMMIT_CUSTOMER opcode call to SyncCustomerPartyListResponseEBM.

While transforming, the service populates the following cross-reference tables with the Oracle BRM IDs obtained:

- CUSTOMERPARTY_ACCOUNTID
- CUSTOMERPARTY_ADDRESSID
- CUSTOMERPARTY_CONTACTID
- CUSTOMERPARTY_BILLPROFILEID
- CUSTOMERPARTY_PAYPROFILEID

4. Updates the block:

If the account is a child account:

- If the parent obtained from the EBM is different from the parent obtained from the opcode call, then it moves the child account to the new parent (as directed by the EBM) by calling the PCM_OP_CUSTCARE_MOVE_ACCT.

Creates the PCM_OP_CUST_UPDATE_CUSTOMER input message by a transformation from the SyncCustomerPartyListEBM.

Checks whether the AIAConfiguration property EnableAccountStatusSync is set to *True*. If set to *True*, then it creates the PCM_OP_CUST_SET_STATUS input message from the SyncCustomerPartyListEBM. Calls the opcode PCM_OP_CUST_SET_STATUS to synchronize the status mentioned in the EBM to Oracle BRM.

If the result of an account update, in which the PayProfile of the account is changed, is SyncCustomerPartyListEBM, then after calling the PCM_OP_CUST_COMMIT_CUSTOMER, it calls the PCM_OP_CUST_DELETE_PAYINFO to delete the earlier PAYINFO object from Oracle BRM.

Transforms the SyncCustomerPartyListEBM to SyncCustomerPartyListResponseEBM.

19.6.15 SyncCustomerPartyListBRM_01CommsJMSConsumer

This process listens to the topic CPARTY_SYNC_TOPIC and as soon as a message is picked up, forwards it to the SyncCustomerPartyListBRMCommsProvABCSImpl.

This service performs the following actions:

- Receives the SyncCustomerPartyListEBM.
- Does an cross-reference lookup to determine whether for the given common ID, the corresponding Oracle BRM ID (for BRM_01 or BRM_02, based on the consumer name) exists.

  If it exists, then the service stamps the message with the particular target system ID and passes it forward to the SyncCustomerPartyListBRMCommsProvABCSImpl.

This process is implemented as a Mediator process. This consumer process is intended for a multiple Oracle BRM system type installation. If multiple Oracle BRM systems exist, then for each system one such consumer must be deployed.

For more information about multiple BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."

20

CM - Synchronize Customer Special Rating Profile: Implementation

This chapter describes the synchronize friends and family list updates to Oracle BRM integration flow and discusses Siebel Customer Relationship Management (Siebel CRM) and Oracle Billing and Revenue Management (Oracle BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 20.1, "Synchronize Friends and Family List Updates to Oracle BRM Integration Flow"

- Section 20.2, "Oracle BRM Interfaces"

- Section 20.3, "Siebel CRM Interfaces"

- Section 20.4, "Industry Oracle AIA Components"

- Section 20.5, "Integration Services"

20.1 Synchronize Friends and Family List Updates to Oracle BRM Integration Flow

The synchronize friends and family list updates to Oracle BRM integration flow uses the following interfaces:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

Figure 20–1 displays the synchronizing friends and family list updates to the Oracle BRM sequence of events.

Figure 20–1   Synchronizing Friends and Family List Updates
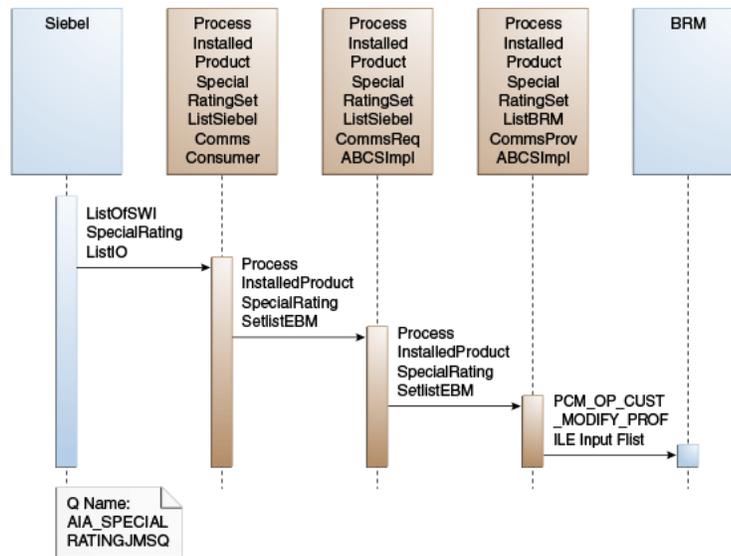
This flow has the following activities. It has a one-way asynchronous pattern.

1. This flow starts when, because of updating the Special Rating List in Siebel CRM for an account; Siebel CRM pushes the ListOfSWISpecialRatingListIO message into an Oracle Advanced Queuing (AQ) named AIA_SPECIALRATINGJMSQ.

2. The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer picks up this message and routes it to the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl.

3. The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl transforms this message into the ProcessInstalledProductSpecialRatingSetListEBM and routes this message to the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl.

4. The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl calls the Oracle BRM opcode PCM_OP_CUST_MODIFY_PROFILE to update this information in Oracle BRM.

20.1.1 Defining Transaction Boundaries and Recovery Details

For this flow there is one transaction boundary. Table 20–1 describes the transaction involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

Table 20–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer picks up message and routes it to ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl, which transforms message and routes to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl. The opcode is called to update information in Oracle BRM.	AIA cross-references updated.	Message goes back to the originating queue AIA_SPECIALRATINGJMSQ_ErrorQ.	Resubmit from AIA_SPECIALRATINGJMSQ_ErrorQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

20.2 Oracle BRM Interfaces

This integration flow uses the following service:

- PCM_OP_CUST_MODIFY_PROFILE

    This service is used to update the special rating profile in Oracle BRM.

20.3 Siebel CRM Interfaces

This integration uses this Siebel CRM workflow event interface:

- SWI Special Rating List Updated

    This workflow event is started when the Special Rating List is updated in Siebel CRM. The event message is pushed into an Oracle Advanced Queuing (AQ) named AIA_SPECIALRATINGJMSQ.

20.4 Industry Oracle AIA Components

This is the enterprise business message (EBM) used by this integration:

- ProcessInstalledProductSpecialRatingSetListEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located here: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application*

Integration Architecture Foundation Pack, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

20.5 Integration Services

These services are delivered with this integration:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

20.5.1 ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer is implemented as a Mediator process.

This consumer listens over the AIA_SPECIALRATINGJMSQ into which Siebel enqueues the SOAP-Wrapped Siebel Special Rating List ABM. This consumer dequeues the messages from this queue, unwraps the message from the SOAP envelope, and routes the Siebel ABM to ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

20.5.2 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl is a BPEL process with one operation: ProcessInstalledProductSpecialRatingSetList. This service accepts as input the Siebel SWISpecialRatingListIO and converts it to the ProcessInstalledProductSpecialRatingSetListEBM structure before routing the message to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl.

This service is invoked when an existing customer (an account exists in Siebel CRM and is synchronized to Oracle BRM) modifies the existing special rating (friends and family) profile in Siebel.

The service looks up the cross-reference values for the customer account ID and installed product ID to find common IDs to appropriately populate the EBM.

20.5.3 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl is implemented as a BPEL process with a single operation: ProcessInstalledProductSpecialRatingSetList.

This service is invoked when an existing customer (an account exists in Siebel CRM and is synchronized to Oracle BRM) modifies the existing special rating (friends and family) profile in Siebel CRM. This service synchronizes the changes in a special rating profile to Oracle BRM.

This service is the Oracle BRM ABCS implementation, which converts the ProcessInstalledProductSpecialRatingSetList into the Oracle BRM ABM before invoking the Oracle BRM opcode PCM_OP_CUST_MODIFY_PROFILE.

21

Understanding the Process Integration for Order Fallout Management

This chapter provides an overview of the order fallout management process integration and discusses capturing faults, order fallout management process integration business flows, and how to extend fault messages to capture order fallout information.

This chapter includes the following sections:

- Section 21.1, "Order Fallout Management Process Integration Overview"

- Section 21.2, "How Oracle AIA Error Handling Framework Captures Faults"

- Section 21.3, "Order Fallout Management Process Integration Business Flows"

- Section 21.4, "Extending Fault Messages to Capture Order Fallout Information"

21.1 Order Fallout Management Process Integration Overview

Orders that have been submitted in Siebel Customer Relationship Management (Siebel CRM) to reflect a customer's intent to use or purchase services provided by a communications service provider (CSP) are passed to downstream systems for fulfillment and provisioning. Because an order is likely to traverse multiple stages before completion, it may fail during the process. The process integration for order fallout management provides a comprehensive, delivered solution that handles such exceptions by implementing a detection and notification process, making the Oracle Communications Order to Cash pre-built integration more robust. Order fallout uses trouble ticketing for notification and tracking of order failures.

The order fallout process is broadly categorized into these three subprocesses:

1. Order Fallout Detection

2. Order Fallout Notification

3. Order Correction

If an error occurs at one Oracle AIA service calls (enterprise business service (EBS), application business connector service (ABCS), and so on), then the service creates an error by invoking the services provided by the Oracle AIA Error Handling Framework to generate a fault message that contains information about the error and also order-specific information that can then be used to create a trouble ticket. The Oracle AIA order fallout management services are then called to create a trouble ticket in Siebel CRM using a Siebel web service. After the trouble ticket is available within Siebel CRM, an order fallout specialist or customer service representative (CSR) opens

the trouble ticket and addresses it either by resubmitting the order after correcting it, or by canceling the order.

For more information about the Oracle AIA Error Handling Framework, see *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack,* "Introduction to Oracle AIA Error Handling."

During the execution of the integration processes, an error may be thrown because of either a system failure or a business failure.

System failures include, but are not limited to, the participating application being down, the network going down, or the Fusion Middleware (FMW) engine going down. Business failures are caused by business reasons and have nothing do with the infrastructure. For example, missing required data is a business error.

The main difference between a system error and a business error is that, for a system error, there is nothing inherently wrong with the original message and can therefore be resubmitted as is for processing. However, for a business error, the original message is flawed (data missing, bad data, and so on) and cannot be resubmitted and reprocessed as is. For business errors, the message must be corrected in the source system and then resubmitted for processing. (For example, a sales order message fails while being interfaced to Oracle Billing and Revenue Management (Oracle BRM) because it has bad data. In this case the sales order must be revised and resubmitted from Siebel CRM to Oracle Order and Service Management (Oracle OSM) for fulfillment, and then from Oracle OSM to Oracle BRM for billing fulfillment.

As part of order fallout management, it only deals with business errors. For system errors, since the message can be retried as is, it is outside the scope of order fallout management.

For more information about how to configure the process integration for order fallout management, see Chapter 27, "Configuring the Process Integration for Order Fallout Management."

21.1.1 Order Fallout Detection

The order can fail in any of the application tiers shown in Figure 21–1. However, this chapter discusses order failure only within Oracle AIA. Other applications and systems are outside the scope of this solution.

Figure 21–1 illustrates the detection subprocess within the order fallout process.

Figure 21–1   Detection Flow

Order Fallout - Detection flow

Lane	Activities
CRM	Create Trouble Ticket
OFM EH Listener	Initiate Create Trouble Ticket Process
AIA Error Handling	Stamp Enhanced Fault Message w/ Correlation ID
AIA ABC	BRM ABC Prepare Order Failure Notification · 3rd Party ABC Prepare Order Failure Notification · 3rd Party ABC Prepare Order Failure Notification
BRM	Identify Order Failure
3rd Party Provisioning System	Identify Order Failure
3rd Party Fulfillment System	Identify Order Failure
AIA Generic	Identify Order Failure · Prepare Order Failure Notification

21.1.2 Order Fallout Notification

When an error occurs within any of the order services, the ABCS (in this case) creates an error in Oracle AIA that is detected by the Oracle AIA Error Handling framework. The framework then creates an enhanced fault message that contains information about the fault and the failed order and publishes it to the AIA Error Java Message Service (JMS) topic. The Oracle AIA Order Fallout Management Error Handling Listener detects the AIA Error Handling Enhanced Fault Message, picks up the message from the queue, and submits it to the order fallout function within Oracle AIA for further processing (creation of trouble ticket).

The AIA Enhanced Fault Message has some following key error and order failure information:

- Faulting Service
- Error Code
- Error Severity
- Error Text
- Time Of Failure

- Order ID

- Order Number

- Order Originating System Code

- Account ID

- Account Name

For more information about extending fault messages, see Section 21.4, "Extending Fault Messages to Capture Order Fallout Information."

Figure 21–2 illustrates the notification subprocess within the order fallout process.

Figure 21–2   Notification Flow
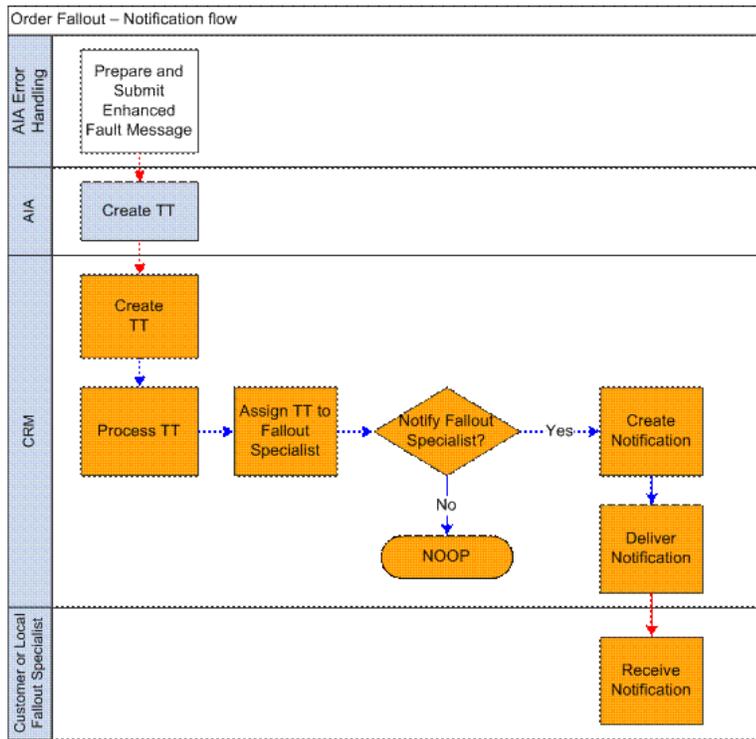
21.1.3 Order Correction

After the trouble ticket is created in Siebel CRM, the request is assigned to a fallout specialist by an assignment rule set in Siebel CRM. The fallout specialist can then log in to the system, pick up the trouble ticket from the queue, and resolve the ticket. After the specialist identifies the failure aspects of the order, they can create a new order to correct the failed order and then submit it for processing.

Order fallout can be caused by one of the following two categories of errors:

- Errors that can only be resolved by changing the order.

  To resolve this type of error, the order fallout specialist must submit a revision order to recover the order from fallout. The OSM Cartridge for Oracle AIA closes any trouble tickets created to report the order fallout and proceed with fulfilling the revision order.

- Errors related to data setup in local fulfillment systems, such as bad inventory data.

To resolve this type of error, the order fallout specialist must correct the cause of the error in the local fulfillment system and resume the order from the Oracle OSM Central Order Management (COM) system.

Submitting a revision order for this type of error does not recover the order from fallout because the revision order will be identical to the base order and therefore ignored by Oracle OSM.

> **Note:** Siebel CRM and Oracle BRM act as local fulfillment systems when participating in the fulfillment of an order. For example, if there is a bad billing profile causing an order to fail, the error must be corrected by fixing the billing profile in Siebel CRM, which triggers synchronization of the corrected billing profile to Oracle BRM. The order then resumes in Oracle OSM.

Figure 21–3 illustrates the Siebel CRM correction flow subprocess within the order fallout management process.

Figure 21–3   Siebel CRM Correction Flow
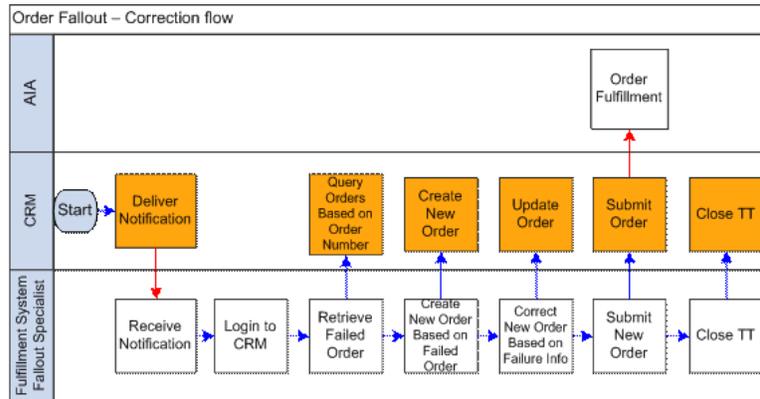
Figure 21–4 illustrates the local correction flow subprocess within the order fallout management process that must take place to undo, compensate, or otherwise fix changes that were committed locally within a fulfillment system for a failed order.

Figure 21–4   Local Correction Flow
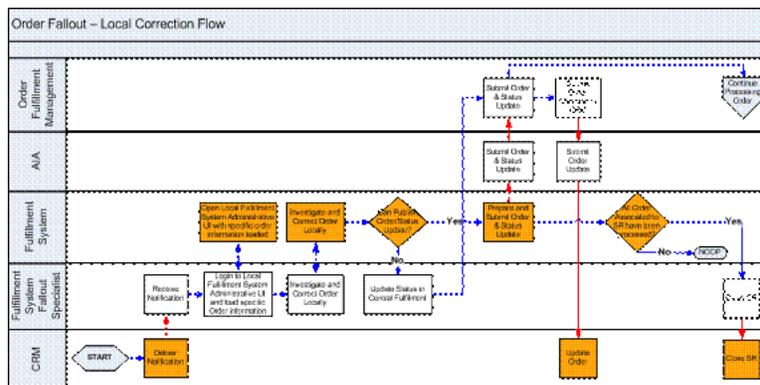
21.2 How Oracle AIA Error Handling Framework Captures Faults

The Oracle AIA Error Handling Framework is used to capture faults across order processing.

Figure 21–5 illustrates the interactions taking place when an order failure is detected by a fulfillment system, such as provisioning and Oracle BRM.

Figure 21–5   Capturing the Fault Sequence Diagram
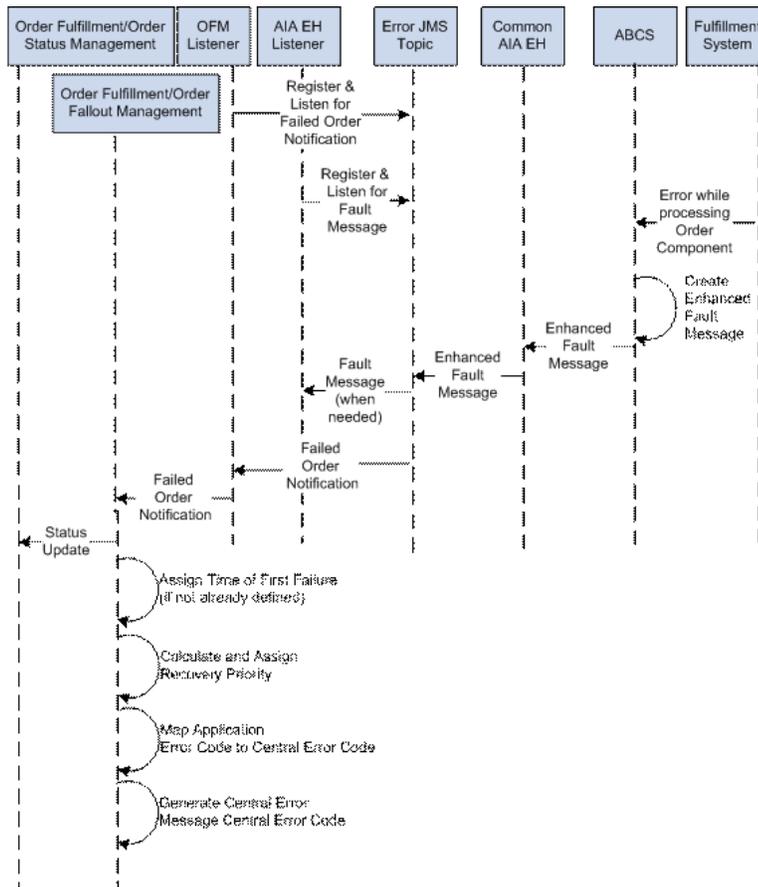
The Oracle AIA Error Handling Framework:

- Allows custom enrichments to the fault message.

- Publishes the enriched fault message to the AIA Error topic.

- Provides a mechanism by which the Order Fallout Listener process picks only the messages that are relevant to the order failure.

Figure 21–6 illustrates how the Oracle AIA Error Handling Framework is leveraged to submit an order failure notification to the AIA Error Topic.

Figure 21–6   Creation and Submission of a Fault Message to the AIA Error Topic
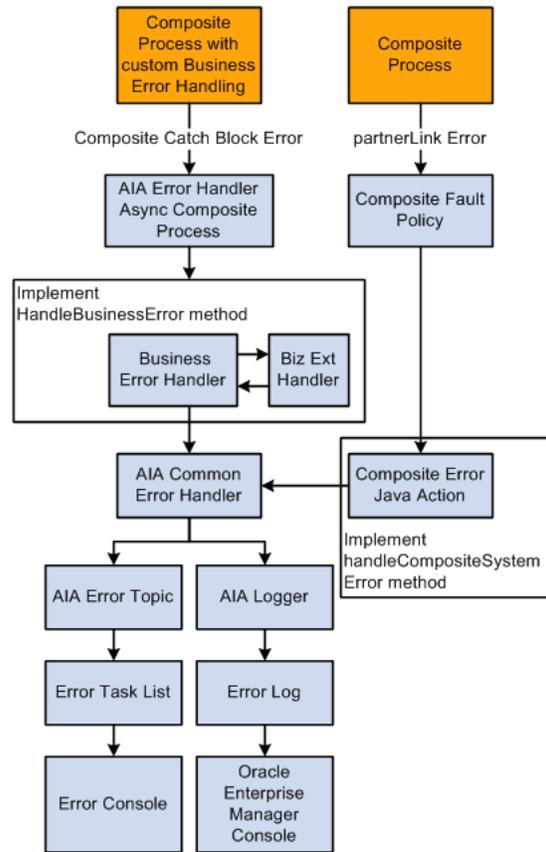
The custom listener selectively picks up the messages from the AIA Error Topic and initiates the appropriate Create Trouble Ticket Business flow, as shown in Figure 21–7.

Figure 21–7   Initiating Appropriate Create Trouble Ticket Flow
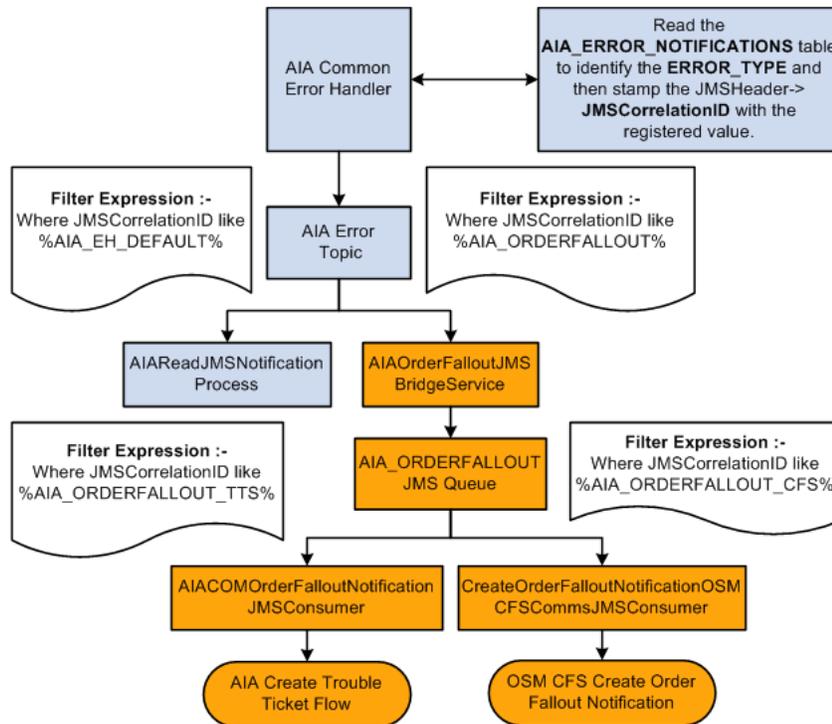
Here are the steps:

1. All of the enriched fault messages with the order failure details are posted to the AIA Error Topic (AIA_ERROR_TOPIC).

2. Messages that are specific to order failure are stamped with a JMS Correlation ID like AIA_ORDERFALLOUT.

3. The AIAOrderFalloutJMSBridgeService consumes the messages from the AIA_ERROR_TOPIC with JMSCorrelationID like AIA_ORDERFALLOUT and publishes them to the AIA_ORDERFALLOUT_JMSQ queue. (This queue is introduced to persist the order failure messages and ensure the messages are not lost if there are errors.)

4. Messages that are specific to order failure have a JMS Correlation ID of either AIA_ORDERFALLOUT_TTS or AIA_ORDERFALLOUT_CFS, depending on whether the trouble ticket is created directly from Oracle AIA or the order failure notification is sent to Oracle OSM CFS.

   For more information about how to set up the seed data so that the trouble ticket is created either from Oracle AIA or from Oracle OSM, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

5. The AIACOMOrderFalloutNotificationJMSConsumer picks up the fault messages and initiates the appropriate Create Trouble Ticket business flow. For the Create Trouble Ticket business flow:

6. If the JMSCorrelationID = AIA_ORDERFALLOUT_TTS, the trouble ticket is directly created from Oracle AIA. (This is the default configuration.)

7. If the JMSCorrelationID = AIA_ORDERFALLOUT_CFS, the order failure notification is sent to Oracle OSM CFS and Oracle OSM CFS initiates the Create Trouble Ticket request.

21.3 Order Fallout Management Process Integration Business Flows

The process integration for order fallout management provides the following integration flows, which enable the *Create Trouble Ticket from Oracle AIA* and the *Create and Manage Trouble Ticket from Oracle OSM* business flows.

Create Trouble Ticket from Oracle AIA

This business flow is enabled by the Oracle Communications Order to Cash - Siebel CRM pre-built integration option with the Oracle Communications Order to Cash - Oracle BRM pre-built integration option.

For this business flow, the JMS Correlation ID = AIA_ORDERFALLOUT_TTS and the request to create a trouble ticket is initiated from Oracle AIA.

The following integration flow enables this business flow:

- Creating a trouble ticket in Siebel CRM integration flow

Create and Manage Trouble Ticket from Oracle OSM

This business flow is enabled by the Oracle Communications Order to Cash - Siebel CRM pre-built integration option with the Oracle Communications Order to Cash - Oracle OSM pre-built integration option.

For this business flow, the JMS Correlation ID = AIA_ORDERFALLOUT_CFS and the request to create a trouble ticket is initiated from Oracle OSM.

The following integration flows enables this business flow:

- Order Failure Notification to Oracle OSM integration flow
- Creating a trouble ticket in Siebel CRM from Oracle OSM integration flow
- Updating a trouble ticket in Siebel CRM from Oracle OSM integration flow

21.3.1 Create Trouble Ticket from Oracle AIA Business Flow

The Create Trouble Ticket from Oracle AIA business flow provides an alternative solution for order fallout management in which Oracle Service Management (Oracle OSM) is not the central fulfillment system and is not used for order fulfillment and fallout management. The approach adopted for this alternate solution assumes that as delivered, the integration handles a subset of order fallout management functionalities by providing delivered services and artifacts that handle order fallout detection and notification.

Also discussed is the functional design required to implement trouble ticket creation in Siebel CRM by the integration when an order fails and an error is detected by the Oracle AIA Error Handler.

Figure 21–8 illustrates the high-level flow of order fulfillment and order fallout management within the capacity of the integration. As illustrated in the diagram, orders can fail at various stages while in process.

Figure 21–8   High-Level Order Fulfillment and Order Fallout
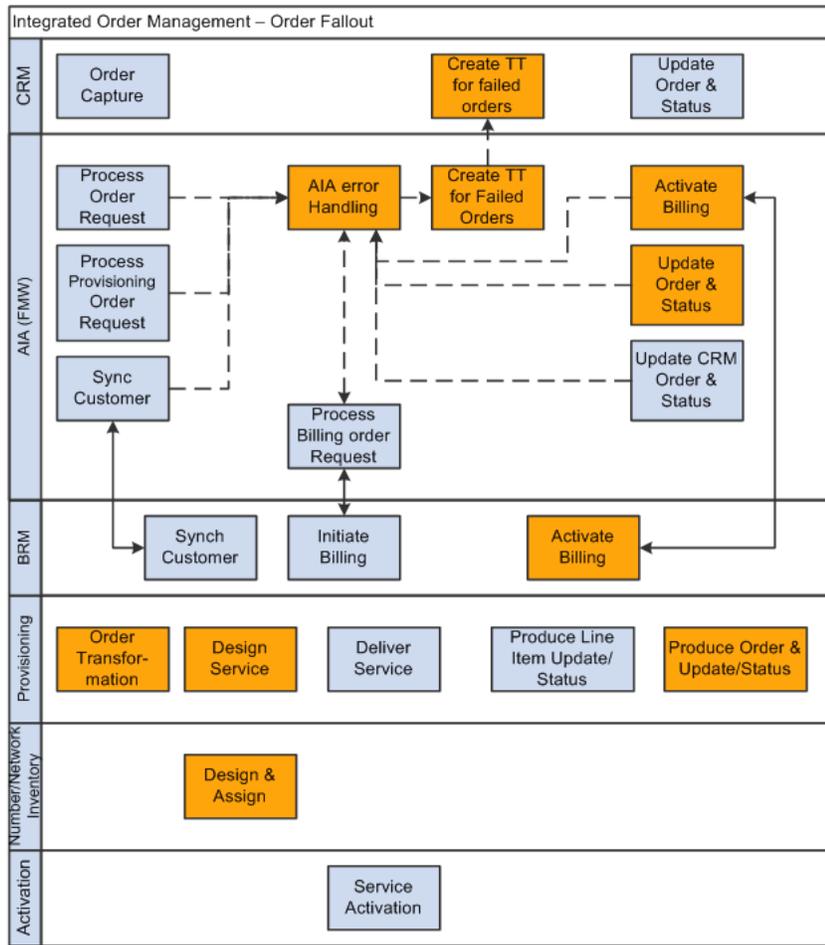
> **Note:**   Figure 21–8 shows only the interactions for order fallout.
> Additional interactions are part of the order fulfillment.

This is a high-level description of the flow:

1. The fault message containing the failed order information is created and submitted within an Oracle AIA service (EBS or application business service (ABS)). If the order fails within a fulfillment application, this returns an error to its ABCS, which produces the fault message.

2. The fault message is then submitted to the AIA Common Error Handler, which recognizes that the fault message is related to an order failure and posts it to the AIA Error JMS Topic (AIA_ERROR_TOPIC) with JMSCorrelation set to AIA_ORDERFALLOUT_TTS (as indicated in the ERROR_TYPE column in the AIA_ERROR_NOTIFICATION page).

3. The Oracle AIA order fallout listener (AIAOrderFalloutJMSBridgeService) picks up the fault message from the AIA Error Topic and pushes it to the Fallout Queue (AIA_ORDERFALLOUT_JMSQ)

4. The AIACOMOrderFalloutNotificationConsumer process picks up the fault message from the Fallout Queue and invokes Oracle AIA order fallout services to create the order failure notification within Oracle AIA.

For more information about extending error handling, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling and Extending Fault Messages.

21.3.1.1 Solution Assumptions and Constraints

These are the assumptions and constraints for the Create Trouble Ticket from Oracle AIA business flow:

- The order fallout management functionality manages orders that fail after being submitted by Siebel CRM.

- One trouble ticket is created in Siebel CRM for every fault message notification. The process flow must ensure that no multiple notifications are generated for the same order failure.

21.3.2 Create and Manage Trouble Ticket from Oracle OSM Business Flow

Oracle AIA or Oracle OSM can initiate the creation of trouble tickets. This is configurable. Installing both the Oracle Communications Order to Cash - Siebel CRM and Oracle Communications Order to Cash - Oracle OSM pre-built integration options automatically configures order fallout to occur in the Oracle OSM Central Fulfillment System (Oracle OSM CFS).

With the combination of Siebel CRM and Oracle OSM:

- Trouble tickets are created in Siebel CRM from Oracle OSM on a per-order or per-system basis. The failure of different orders in the same system generates different trouble tickets, and the failure of the same order in a different system generates a different trouble ticket, but multiple order line item failures for the same order in the same system generates only one trouble ticket. The additional order line item failure information is appended.

- If the cancellation of a failed order is required as part of the recovery flow, the fallout specialist should cancel the order from Oracle OSM.

- Any custom process flow that invokes the creation of an order failure notification must ensure that no multiple notifications are generated for the same order failure.

Figure 21–9 illustrates the high-level process flow involved in using Oracle OSM for order fallout management. It identifies the possible sources of failed orders, capturing these faults using the Oracle AIA Error Handling Framework and the creation of the trouble ticket from Oracle OSM in Siebel CRM for the failed order:

Figure 21–9   Order Fallout Process Flow Using Oracle OSM
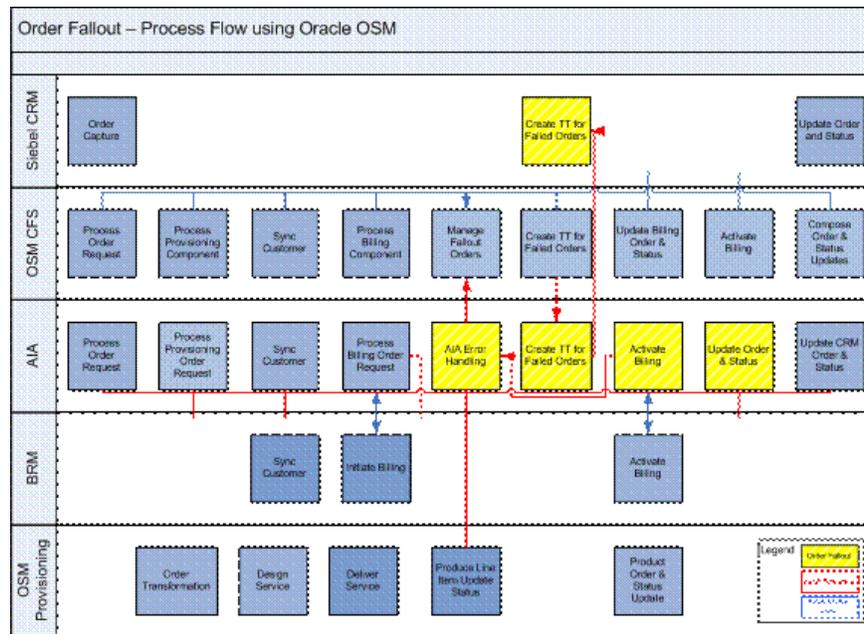
21.3.2.1 Solution Assumptions and Constraints

These are the assumptions and constraints:

- Order fallout management functionality handles orders that fail after being submitted by Siebel CRM.

- When an order revision fails upon arrival in Oracle OSM, a new trouble ticket for the revision is created, and any existing trouble ticket for the base order is preserved. In this case, the trouble ticket acts as an important notification of the failed on arrival condition. The side effect is that the fallout specialist must manually close the trouble ticket for the revision that failed upon arrival.

21.4 Extending Fault Messages to Capture Order Fallout Information

The order fallout management solution leverages the existing Oracle AIA Error Handling Framework to capture order failure notifications when an ABCS or an Oracle AIA service ends due to error.

A fault message is created when an order fails in an AIA service, an ABCS, or in the fulfillment system. The fault message is enhanced with additional information to capture pertinent data about the order failure.

The messages used by the Oracle AIA Error Handling Framework to capture the errors must be extended to capture order failure information. The following two tables describe additional fields that must be added to the Oracle AIA error handling messages to capture order fallout information.

For more information about extending error handling, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling.

If a fault happens within Oracle AIA, the fault message has all the required details of the failed order and does not require additional enrichment by the Oracle AIA Error Handling Framework. In this case, the common error handler stamps the correlation

ID to the fault message and publishes it to the Error Topic (JMS Correlation ID is set to the value indicated in the AIA Error Notification table) so that it can be uniquely identified as an order fallout fault message.

For more information about extending error handling, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Fault Messages.

Table 21–1 lists the order header-level data that is passed from a fulfillment system or Oracle AIA service to the Order Fallout Management functionality over the Oracle AIA Error Handling Framework (order header-level fields).

Table 21–1   Order Header-Level Data

Field Name	Type	Description	Source	Optional
Order Originating System Code	ID	The system code of the Siebel CRM system from which the order was placed. It is required to cross-reference the IDs back to the appropriate Siebel CRM IDs.	Oracle AIA service	No
Sales Order Number	Alphanumeric	Alphanumeric identifier for the sales order number (Siebel CRM value).	Siebel CRM	Yes
Sales Order Revision Number	Numeric	Numeric field storing the sales order number (Siebel CRM value).	Siebel CRM	Yes
SalesOrderID	ID	Siebel CRM Sales Order ID. Required to create trouble tickets for the orders that fail even before hitting the central fulfillment system.	Siebel CRM	Yes
Account Name	AlphaNumeric	AlphaNumeric value identifying the Siebel CRM account name.	Siebel CRM	Yes
Account ID	ID	Siebel CRM Account ID. Required to create trouble tickets for the orders that fail even before hitting the central fulfillment system.	Siebel CRM	Yes
SalesOrderID (Common)	ID	Common Order ID. (Required when Oracle AIA creates the trouble tickets).	Oracle AIA service	No
AccountID (Common)	ID	Common Account ID.	Oracle AIA service	Yes
Order ID	ID	Alphanumeric identifier for the order. Assigned by fulfillment system to the order. The fulfillment system uses it to correlate the order back to the common order ID received for the original order. The common order ID is then mapped to the Siebel order ID by the Siebel ABCS.	Fulfillment System	No
Order Number	AlphaNumeric	User-friendly identifier for the order in the fulfillment system.	Fulfillment System	Yes
ProductID	AlphaNumeric	Alphanumeric identifier for the product used for the failed line or the product for the first order line in case of multiple line failures.	Siebel CRM or Oracle AIA service	Yes

Table 21–1   (Cont.) Order Header-Level Data

Field Name	Type	Description	Source	Optional
Fulfillment System of Failure for Order	LOV	Part of the enterprise business object (EBO) header. Set to the fulfillment system in which the order failed. The Oracle AIA identifier for the fulfillment system is used.	Fulfillment system of Failure or Oracle AIA service	No
Service of Failure / FailureSubSystem	LOV	Identifies the Oracle AIA service, web service, application programming interface (API), or SubSystemCode (if available) where the order failed.	Fulfillment System of failure or Oracle AIA service	Yes
Message	Alphanumeric	Used for the message (error, warning, or other). It can also be used to return notification to customers or other systems. Not to be confused with the original input order message.	Fulfillment System of failure or Oracle AIA Service	Yes
Error Code	Alphanumeric	Used to return the error code from the downstream fulfillment system (if any).	Fulfillment System of failure or Oracle AIA service	No
Error Severity	LOV	Used to return the error severity from the downstream fulfillment system (if any).	Fulfillment System of failure or Oracle AIA service	Yes
Processing Number	ID	Identifier of the job ID assigned in case of batch or bulk orders.	Siebel CRM	Yes
Processing Type Code	Code	Code to identify the job type.	Siebel CRM	Yes
Processing Quantity	Quantity	Job cardinality - Total number of orders within the job.	Siebel CRM	Yes

For more information about how to pass this information, see Appendix J, "OLM - Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant."

This table shows the order fallout information passed from a fulfillment system or Oracle AIA service to the order fallout management functionality over the Oracle AIA Error Handling Framework (order-line item-level fields). This is supplied only if the Oracle AIA service or the fulfillment system identifies a particular order line item as responsible for the order failure. For system faults caused by network issues or system unavailability, the order lines may not actually add value to the trouble ticket and in those cases you need not populate these fields.

Table 21–2   Order-Line Item-Level Data

Field Name	Type	Description	Source	Optional
Order Line Item ID	ID	Unique identifier for the order item.	Siebel CRM	No
Message	Alphanumeric	Used for error message. It can also be used to return notification to customers or other systems.	Fulfillment system of failure or Oracle AIA service.	Yes
Error Code	Alphanumeric	Used to return the error code from the downstream fulfillment system (if any).	Fulfillment system of failure or Oracle AIA service.	No
Error Severity	Alphanumeric	Used to return the error severity from the downstream fulfillment system (if any).	Fulfillment system of failure or Oracle AIA service.	Yes
StatusContext	LOV	Used to capture status-related display information or status-related information that is product-dependent. It can also be used to capture the current milestone within the provisioning system for the service associated with the order item.	Fulfillment system of failure or Oracle AIA service.	Yes
FailureSubSystemCode	LOV	Subsystem code or API where the order line has failed. Applicable for participating applications. If the fault is within Oracle AIA, the service which faulted is assumed as the subsystem of failure.	Fulfillment system of failure or Oracle AIA service.	Yes

The overall solution includes:

1. Extending the Oracle AIA fault message to be able to capture the additional information identified in the tables described previously.

2. Extending the common error handler to be able to:

   ■ Identify when a fault message is related to order failures.

   ■ Stamp the error type in the fault message as a correlation ID and invoke the appropriate fault extension handlers (in case of a partner link fault).

   ■ Publish to the AIA Error JMS Topic.

3. Creating the Oracle AIA order fallout listener (AIAOrderFalloutJMSBridgeService), which:

   ■ Listens to all messages published to the AIA Error JMS Topic.

   ■ Picks up the messages that are specific to order fallout by looking at the correlation ID that contains the error type stamped by the Oracle AIA Common Error Handler.

   ■ Persists the fault message into a fallout queue (AIA_ORDERFALLOUTJMSQ).

4. Creating a listener to the Order Fallout Queue, AIACOMOrderFalloutNotificationConsumer that routes the fault message appropriately to the process integration for order fallout management to create the trouble ticket.

For more information about extending error handling, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling.

21.4.1 Exception Handling

The types of operation conducted by the AIA Order Fallout Listeners are quite straightforward; therefore, the exception handling is also straightforward: If an error occurs while the listeners are preparing the message for the invocation of the Oracle AIA service, then a standard Oracle AIA Error Handling Framework notification is posted to the Oracle AIA Error Handling Framework.

22

OFM - Create Trouble Ticket from Oracle AIA Business Flow: Implementation

This chapter provides an overview of the Create Trouble Ticket from Oracle AIA business flow and discusses Siebel Customer Relationship Management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, integration services, and fallout-enabled services for this business flow.

This chapter includes the following sections:

22.1 Creating Trouble Tickets in Siebel CRM from Oracle AIA Overview

After the Order Fallout Listener (AIACOMOrderFalloutNotificationConsumerProcess) picks up the fault message from the Oracle AIA Error JMS Topic based on an error notification from a downstream system or Oracle AIA service that has ended due to an error, an Oracle AIA Requestor Service provides an interface to invoke a Provider for the creation of trouble tickets in Siebel CRM.

This feature is composed of the following services:

- CreateTroubleTicketAIACommsReqImpl - Oracle AIA Requestor application business connector service (ABCS)
- CreateTroubleTicketSiebelCommsProvABCSImpl - Oracle AIA Provider ABCS invoked to create a trouble ticket in Siebel CRM.

On an error, the order fallout process (detection) within Oracle AIA passes the order fault message that is queued in the Oracle AIA Error JMS Topic to the CreateTroubleTicketAIACommsReqImpl ABCS. The service then routes the Oracle AIA message to the Siebel provider, which in turn calls the Siebel web service to create the trouble ticket in Siebel.

22.2 Creating a Trouble Ticket in Siebel CRM Integration Flow

This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCSImpl

Figure 22–1 illustrates the create trouble ticket integration scenario.

Figure 22–1   Create Trouble Ticket Sequence Diagram
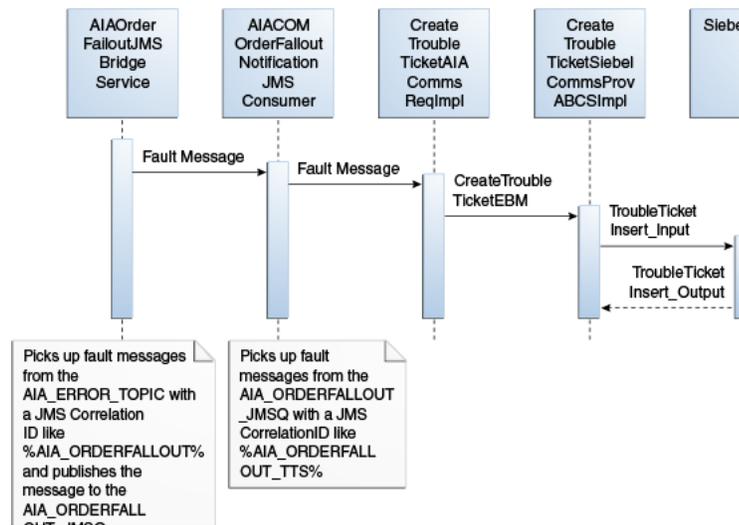
This flow creates a trouble ticket and has the following set of activities:

1. The enriched fault messages that contain the details of the order are pushed to the AIA_ERROR_TOPIC using the Oracle AIA Error Handling Framework. These messages are stamped with a JMS Correlation ID = AIA_ORDERFALLOUT_TTS in case the trouble tickets are created from Oracle AIA directly, based on the ERROR_TYPE set in the AIA Error Notifications page.

2. The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID such as AIA_ORDERFALLOUT (AIA_ORDERFALLOUT_TTS in this case) and publishes them to the AIA_ORDERFALLOUT_JMSQ JMS Queue.

3. The AIACOMOrderFalloutNotificationJMSConsumer picks up the messages stamped with the JMS Correlation ID AIA_ORDERFALLOUT_TTS from the AIA_ORDERFALLOUT_JMSQ.

4. The AIACOMOrderFalloutNotificationJMSConsumer invokes the CreateTroubleTicketAIACommsReqImpl.

5. The CreateTroubleTicketAIACommsReqImpl service parses the fault message, prepares the CreateTroubleTicketEBM, and routes the message to the CreateTroubleTicketSiebelCommsProvABCSImpl.

6. The CreateTroubleTicketSiebelCommsProvABCSImpl synchronously invokes the Siebel web service (SWITroubleTicketIO.wsdl: SWITroubleTicketInsert) and the response trouble ticket ID is received in the form of SWITroubleTicketInsert_Output message. This application business message (ABM) is transformed to the CreateTroubleTicketResponseEBM depending on the Response Code set in the EBM.

22.2.1 Defining Transaction Boundaries and Recovery Details

For this flow there are two transaction boundaries. Table 22–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- AIAOrderFalloutJMSBridgeService

- AIACOMOrderFalloutNotificationJMSConsumer

- CreateTroubleTicketAIACommsReqImpl

- CreateTroubleTicketSiebelCommsProvABCSImpl

Table 22–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID and publishes to AIA_ORDERFALLOUT_JMSQ.	Message enqueued in AIA_ORDERFALLOUT_JMSQ.	Rollback JMS message to AIA_ERROR_TOPIC.	Resubmit from AIA_ERROR_TOPIC.
AIACOMOrderFalloutNotificationJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_TTS and invokes CreateTroubleTicketAIACommsReqImpl, which parses fault message and routes to CreateTroubleTicketSiebelCommsProvABCSImpl.	AIA cross-reference entries.	Rollback the message to AIA_ORDERFALLOUT_JMSQ.	Resubmit from AIA_ORDERFALLOUT_JMSQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

22.2.2 Exception Handling

These are the exception handling notes for creating trouble tickets in Siebel CRM:

- If validation of the message fails because of missing mandatory data, incorrect formatting, or other problems, then an error message identifying the validation issue is returned to the invoking application.

- In case of any errors in the flow, a standard Oracle AIA Error Handling Framework notification is posted to the Oracle AIA Error Handling Framework.

22.3 Siebel CRM Interfaces

The Create Trouble Ticket from Oracle AIA business flow uses this Siebel CRM interface:

- **SWI Trouble Ticket Service:** This service is invoked by the Siebel ABCS to create or update a trouble ticket in Siebel CRM. If the request is for creating a new trouble ticket, a new trouble ticket is created and the trouble ticket number is returned. If the request is to update a particular trouble ticket, typically to close the trouble ticket, the trouble ticket is updated.

For more information, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

22.4 Industry Oracle AIA Components

The Create Trouble Ticket from Oracle AIA business flow uses these industry components:

- TroubleTicketEBO
- CreateTroubleTicketEBM
- CreateTroubleTicketResponseEBM
- CommunicationsTroubleTicketEBSV1.wsdl

The industry EBO and EBM XML schema (XSD) files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1/

The industry EBS web service description language (WSDL) files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/TroubleTicket/V1/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

22.5 Integration Services

These services are delivered with this integration:

- CreateTroubleTicketSiebelCommsProvABCSImpl
- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- AIAOrderFalloutErrorHandlerExtension.java

Some of these services have been enabled to use Session Pool Manager.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations Utilities Guide*, "Session Pool Manager."

22.5.1 CreateTroubleTicketSiebelCommsProvABCSImpl

The CreateTroubleTicketSiebelCommsProvABCSImpl is implemented as an asynchronous business process execution language (BPEL) process. This service takes CreateTroubleTicketEBM as the input. It invokes the Siebel web service to create the trouble ticket and after the trouble ticket is created in Siebel CRM, the trouble ticket ID is passed back to this service.

This process acts either as a fire-and-forget one-way flow or a request response flow depending on a couple of configurable parameters. CreateTroubleTicketSiebelCommsProvABCSImpl creates a trouble ticket response message (creates a cross-reference for the trouble ticket ID with the Siebel ID) if the property TroubleTicket.GenerateTroubleTicketResponse is set to True or if the response code attribute (CreateTroubleTicketEBM/DataArea/Create/@responseCode) is not null. Otherwise, this service just acts as a fire-and-forget flow and ignores the response.

This service is SPM enabled.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

22.5.2 AIAOrderFalloutJMSBridgeService

The AIAOrderFalloutJMSBridgeService service is a Mediator service that picks up the fault message from the AIA Error Topic and publishes the message to the AIA_ORDERFALLOUT_JMSQ. This service is introduced to persist the enhanced fault message into a fallout queue and retry in case of errors in the downstream process. The message can either be picked from this queue by Oracle AIA to directly create a trouble ticket in Siebel CRM or to send an order failure notification to Oracle Order and Service Management Central Fulfillment System (Oracle OSM CFS).

22.5.3 AIACOMOrderFalloutNotificationJMSConsumer

The AIAOrderFalloutNotificationJMSConsumer service is implemented as a Mediator service and picks up the fault message from the AIA Error Topic. The fault message is passed to the CreateTroubleTicketAIACommsReqImpl process. This service acts as the consumer, listening to the messages produced in the AIA Error Topic.

22.5.4 CreateTroubleTicketAIACommsReqImpl

The CreateTroubleTicketAIACommsReqImpl service is implemented as a one-way asynchronous BPEL process. This service picks up the fault message from the AIACOMOrderFalloutNotificationJMSConsumer. The fault message is parsed and then the CreateTroubleTicketEBM is constructed.

22.5.5 AIAOrderFalloutErrorHandlerExtension - Java Class

This module is the Java action that is specified for enhancing the fault message. In case of a Java action in the bpel/esb fault policy, the control is handed to this application module to enrich the fault message with business-specific content. The enriched fault message is returned to the AIA Error Handling Framework Common Error Handler.

- oracle.apps.aia.industry.comms.eh.AIAOrderFalloutErrorHandlerExtension.java implements oracle.apps.aia.core.eh. IAIAErrorHandlerExtension interface.

- This class implements the IAIAErrorHandlerExtension interface, which has two methods exposed: one for treating a BPEL fault and the other for a Mediator fault.

This class constructs the ApplicationFaultData element of the fault message with the order-related details.

22.6 Business Flow Fallout-Enabled Services

The following Create Trouble Ticket from Oracle AIA business flow services are fallout-enabled:

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCSImplV2
- SyncCustomerPartyListBRMCommsProvABCSImpl

23

OFM - Create and Manage Trouble Ticket from Oracle OSM Business Flow: Implementation

This chapter provides an overview of the Create and Manage Trouble Ticket from Oracle OSM business flow and discusses Siebel Customer Relationship Management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, integration services, and fallout-enabled services for this business flow.

This chapter includes the following sections:

- Section 23.1, "Creating Trouble Tickets in Siebel CRM from Oracle OSM Overview"
- Section 23.2, "Order Failure Notification to Oracle OSM Integration Flow"
- Section 23.3, "Creating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow"
- Section 23.4, "Updating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow"
- Section 23.5, "Siebel CRM Interfaces"
- Section 23.6, "Industry Oracle AIA Components"
- Section 23.7, "Integration Services"
- Section 23.8, "Business Flow Fallout-Enabled Services"

23.1 Creating Trouble Tickets in Siebel CRM from Oracle OSM Overview

The Create and Manage Trouble Ticket from Oracle OSM business flow enables the following integration flows:

- Order Failure Notification to Oracle OSM:

  Listens to the common error topic used in the Oracle AIA Error Handling Framework (AIA_ERROR_TOPIC) for errors or faults specific to orders, enriches these fault messages, and then publishes them to the central order fallout management (OFM) in Oracle Order and Service Management (Oracle OSM).

- Creating a Trouble Ticket in Siebel CRM from Oracle OSM:

  Creates trouble tickets in Siebel CRM for individual and batch or bulk orders from Oracle OSM.

- Updating a Trouble Ticket in Siebel CRM from Oracle OSM:

Updates trouble tickets in Siebel CRM from Oracle OSM.

23.2 Order Failure Notification to Oracle OSM Integration Flow

The Oracle AIA order fallout listener (AIAOrderFalloutJMSBridgeService), listens to all messages published to the Oracle AIA Error JMS Topic (AIA_ERROR_TOPIC) for errors or faults specific to order fallout by looking at the correlation ID. These fault messages are enriched and published to the central OFM in Oracle OSM.

This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

Figure 23–1 illustrates how Oracle OSM initiates the request to create a trouble after receiving an order failure notification.

Figure 23–1   Order Failure Notification to Oracle OSM
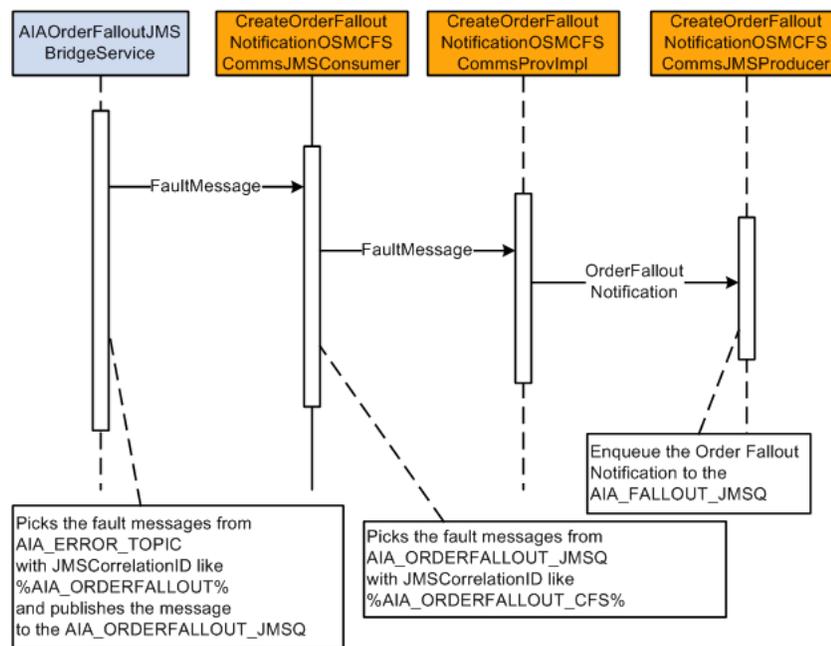
When this process initiates, the following events occur:

1. The Enriched Fault Message containing the details pertaining to orders are pushed to the AIA_ERROR_TOPIC using the Oracle AIA Error Handling Framework. These messages are stamped with a JMS Correlation ID, for example, AIA_ORDERFALLOUT.

2. AIAOrderFalloutJMSBridgeService picks up the messages with JMSCorrelationID, for example, AIA_ORDERFALLOUT and publishes them to the AIA_ORDERFALLOUT_JMSQ queue.

3. AIACOMOrderFalloutNotificationJMSConsumer picks up the messages stamped with the JMS Correlation ID AIACOM_ORDERFALLOUT_CFS from the AIA_

ORDERFALLOUT_JMSQ and invokes
CreateOrderFalloutNotificationOSMCFSCommsProvImpl.

4. CreateOrderFalloutNotificationOSMCFSCommsProvImpl parses the fault message, transforms it to OrderFalloutNotification message, and invokes the CreateOrderFalloutNotificationOSMCFSCommsJMSProducer service.

5. CreateOrderFalloutNotificationOSMCFSCommsJMSProducer pushes the message to the AIA_FALLOUT_JMSQ store and forward (SAF) queue. Oracle OSM CFS picks up this message, marks the order as failed, and initiates the request to create a trouble ticket.

23.3 Creating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow

This integration flow uses these interfaces:

- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCSImpl
- CreateTroubleTicketOSMCFSCommsJMSProducer

Figure 23–2 describes the creation of trouble tickets in Siebel CRM from Oracle OSM.

Figure 23–2   Creating a Trouble Ticket in Siebel CRM from Oracle OSM
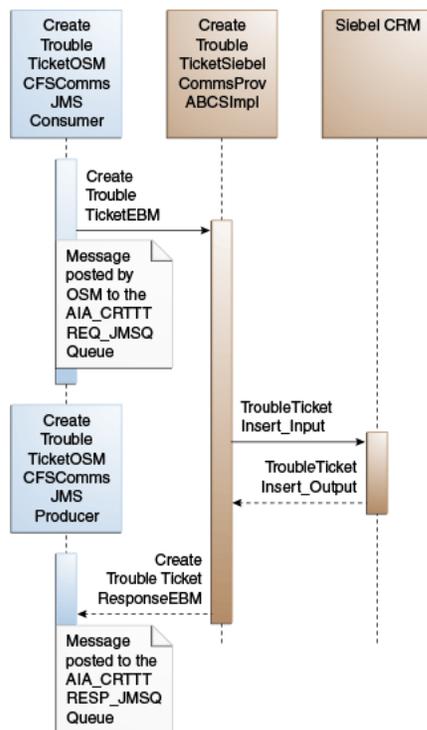
When this process initiates, the following events occur:

1. The OSM Fulfillment system produces the CreateTroubleTicketEBM in the AIA_CRTTTREQ_JMSQ SAF queue.

2. CreateTroubleTicketOSMCFSCommsJMSConsumer picks up the message from the queue and routes the message to the CreateTroubleTicketSiebelCommsProvABCSImpl service.

3. The CreateTroubleTicketSiebelCommsProvABCSImpl service invokes the Siebel web service (SWITroubleTicketIO.wsdl: SWITroubleTicketInsert) synchronously, and the response trouble ticket ID is received in the form of a SWITroubleTicketInsert_Output message.

4. This process invokes the CreateTroubleTicketSiebelCommsProvABCSImpl with the CreateTroubleTicketEBM and transforms the CreateTroubleTicketEBM to TroubleTicketInsert_Input ABM.

5. The CreateTroubleTicketSiebelCommsProvABCSImpl service invokes the Siebel web service (ServiceRequest.wsdl) synchronously, and the response trouble ticket ID is received in the form of a TroubleTicketInsert_Output message. This ABM is transformed to CreateTroubleTicketResponseEBM, and the message is routed to the CreateTroubleTicketRespOSMCFSCommsJMSProducer service, which pushes the message to the AIA_CRTTTRESP_JMSQ SAF queue.

6. Oracle OSM CFS picks up the message and stores the TroubleTicketID for reference.

23.3.1 Defining Transaction Boundaries and Recovery Details

For the Order Failure Notification to Oracle OSM and Creating a Trouble Ticket in Siebel CRM from Oracle OSM flows there are three transaction boundaries. Table 23–1 describes the transactions involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer
- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCSImpl
- CreateTroubleTicketRespOSMCFSCommsJMSProducer

Table 23–1   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID and publishes to AIA_ORDERFALLOUT_JMSQ.	Message enqueued in AIA_ORDERFALLOUT_JMSQ.	Rollback JMS message to AIA_ERROR_TOPIC.	Resubmit from AIA_ERROR_TOPIC.
CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_CFS and invokes CreateOrderFalloutNotificationOSMCFSCommsProvImpl, which parses fault message and invokes CreateOrderFalloutNotificationOSMCFSCommsJMSProducer.	Message is submitted to Oracle OSM through AIA_FALLOUT_JMSQ.	Rollback the message to AIA_ORDERFALLOUT_JMSQ.	Resubmit from AIA_ORDERFALLOUT_JMSQ.
CreateTroubleTicketOSMCFSCommsJMSConsumer picks up message and routes it to CreateTroubleTicketSiebelCommsProvABCSImpl, which invokes Siebel web service. Response trouble ticket ID is received. Response message is created, which invokes CreateTroubleTicketSiebelCommsProvABCSImpl, which routes message to CreateTroubleTicketRespOSMCFSCommsJMSProducer.	AIA cross-references created. Message goes to queue AIA_CRTTTRESP_JMSQ.	Rollback the message to AIA_CRTTTREQ_JMSQ.	Resubmit from AIA_CRTTTREQ_JMSQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

23.4 Updating a Trouble Ticket in Siebel CRM from Oracle OSM Integration Flow

This integration flow uses the following interfaces:

- UpdateTroubleTicketOSMCFSCommsJMSConsumer

- UpdateTroubleTicketSiebelCommsProvABCSImpl

Figure 23–3 describes the Trouble Ticket Update flow from Oracle OSM to Siebel CRM.

Figure 23–3   Updating a Trouble Ticket in Siebel CRM from Oracle OSM
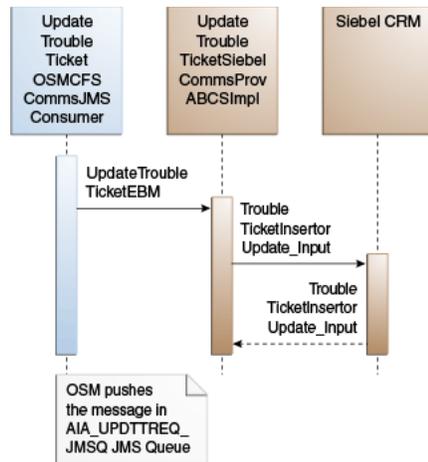
When this process initiates, the following events occur:

1. The Oracle OSM fulfillment system produces the UpdateTroubleTicketEBM in the AIA_UPDTTREQ_JMSQ SAF queue.

2. UpdateTroubleTicketOSMCFSCommsJMSConsumer picks up the message from the queue and invokes the UpdateTroubleTicketSiebelCommsProvABCSImpl service.

3. The UpdateTroubleTicketSiebelCommsProvABCSImpl service. transforms the EBM to TroubleTicketInsertorUpdate_Input ABM, and the Siebel web service is invoked to update the trouble ticket.

23.4.1 Defining Transaction Boundaries and Recovery Details

For this flow there is one transaction boundary. Table 23–2 describes the transaction involved, the database operations, and what actions to take in case of an error.

For more information about system errors and business errors, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

The following services are involved:

■   UpdateTroubleTicketOSMCFSCommsJMSConsumer

■   UpdateTroubleTicketSiebelCommsProvABCSIm

Table 23–2   Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
UpdateTroubleTicketOSMCFSCommsJMSConsumer picks up the message from the queue and invokes UpdateTroubleTicketSiebelCommsProvABCSImpl, which invokes the Siebel web service to update the trouble ticket.	AIA cross-references updated. Trouble Ticket updated in Siebel.	Message goes back to the originating queue AIA_UPDTTREQ_JMSQ.	Resubmit from AIA_UPDTTREQ_JMSQ.

For more information about rollback procedures, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages.

23.5 Siebel CRM Interfaces

The Create and Manage Trouble Ticket from Oracle OSM business flow uses this Siebel CRM interface:

■ SWI Trouble Ticket Service

This inbound web service is invoked by the Siebel ABCS to create or update a trouble ticket in Siebel CRM. If the request is for creating a new trouble ticket, then a new trouble ticket is created and the trouble ticket number is returned. If the request is to update a specific trouble ticket, typically to close the trouble ticket, then the trouble ticket is updated.

For more information about web services, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

23.6 Industry Oracle AIA Components

The Create and Manage Trouble Ticket from Oracle OSM business flow uses these industry components:

■ TroubleTicketEBO

■ CreateTroubleTicketEBM

■ CreateTroubleTicketResponseEBM

■ UpdateTroubleTicketEBM

The industry enterprise business object (EBO) and EBM XSD files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1/

The industry EBS WSDL files are located here: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/TroubleTicket/V1/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and remain intact after a patch or an upgrade, so long as the extensibility guidelines are followed.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with AIA Design Patterns," AIA Assets Extensibility Patterns, Extending Existing Schemas in AIA.

23.7 Integration Services

These services are delivered with this integration:

■ CreateTroubleTicketSiebelCommsProvABCSImpl

■ UpdateTroubleTicketSiebelCommsProvABCSImpl

■ AIAOrderFalloutJMSBridgeService

- AIACOMOrderFalloutNotificationJMSConsumer

- CreateTroubleTicketAIACommsReqImpl

- CreateOrderFalloutNotificationOSMCFSCommsProvImpl

- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer

- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

- CreateTroubleTicketOSMCFSCommsJMSConsumer

- CreateTroubleTicketRespOSMCFSCommsJMSProducer

- UpdateTroubleTicketOSMCFSCommsJMSConsumer

- CreateFaultNotificationLFCommsJMSConsumer

Some of these services have been enabled to use Session Pool Manager.

For more information about Session Pool Manager, see *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide*, "Session Pool Manager."

For more information, see Chapter 27, "Configuring the Process Integration for Order Fallout Management."

23.7.1 UpdateTroubleTicketSiebelCommsProvABCSImpl

The UpdateTroubleTicketSiebelCommsProvABCSImpl is a service that acts as the provider for Siebel CRM Update Trouble Ticket functionality. This service does not return any response.

The UpdateTroubleTicketSiebelCommsProvABCSImpl service. transforms the EBM to TroubleTicketInsertorUpdate_Input ABM, and the Siebel web service is invoked to update the trouble ticket.

Updating open trouble tickets to *Closed* or adding additional failed order lines to an existing open trouble ticket are the scenarios in which an update to trouble ticket request originates from Oracle OSM CFS.

23.7.2 CreateOrderFalloutNotificationOSMCFSCommsProvImpl

The CreateOrderFalloutNotificationOSMCFSCommsProvImpl service is a BPEL process that picks up the Fault message from CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer. It parses the Fault message and then constructs the OrderFalloutNotification message.

Next, the CreateOrderFalloutNotificationOSMCFSCommsJMSProducer service is invoked to enqueue the order fallout notification message in the AIA_FALLOUT_JMSQ SAF queue for Oracle OSM CFS. This action consumes and triggers a fallout event for the particular order.

23.7.3 CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer

The CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer service is a Mediator service that picks up the Fault message from the AIA_ORDERFALLOUT_JMSQ. It passes the Fault message to the CreateOrderFalloutNotificationOSMCFSCommsProvImpl process based on the JMS Correlation ID.

This service acts as a consumer, listening to the messages produced in AIA_ORDERFALLOUT_JMSQ.

23.7.4 CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

The CreateOrderFalloutNotificationOSMCFSCommsJMSProducer is a BPEL process that enqueues the OrderFalloutNotification message to the AIA_FALLOUT_JMSQ SAF queue. Oracle OSM then picks the message from this queue and triggers a fallout event in Oracle OSM. The CreateOrderFalloutNotificationOSMCFSCommsProvImpl service invokes this service.

23.7.5 CreateTroubleTicketOSMCFSCommsJMSConsumer

The CreateTroubleTicketOSMCFSCommsJMSConsumer is a Mediator service that picks up the CreateTroubleTicketEBM message from the AIA_CRTTTREQ_JMSQ SAF queue. It routes the message to the CreateTroubleTicketSiebelCommsProvABCSImpl service. This service acts as a consumer, listening to the messages produced in the AIA_CRTTTREQ_JMSQ SAF queue.

23.7.6 CreateTroubleTicketRespOSMCFSCommsJMSProducer

The CreateTroubleTicketRespOSMCFSCommsJMSProducer is a BPEL process that enqueues the CreateTroubleTicketResponseEBM message to the AIA_CRTTTRESP_JMSQ SAF queue. Oracle OSM then picks up the message from this queue and then updates the order task with the created trouble ticket ID.

23.7.7 UpdateTroubleTicketOSMCFSCommsJMSConsumer

The UpdateTroubleTicketOSMCFSCommsJMSConsumer is a Mediator service that picks up the UpdateTroubleTicketEBM message from the AIA_UPDTTREQ_JMSQ SAF queue and invokes the UpdateTroubleTicketSiebelCommsProvABCSImpl service. This service acts as a consumer, listening to the messages produced in the AIA_UPDTTREQ_JMSQ SAF queue.

23.7.8 CreateFaultNotificationLFCommsJMSConsumer

The CreateFaultNotificationLFCommsJMSConsumer is a Mediator service that picks up the Fault message from the AIA_LFERROR_JMSQ queue. It routes this message to the BPEL service AIAAsyncErrorHandlingBPELProcess, which queues the message in the AIA_ERROR_TOPIC.

Line Fulfillment (provisioning) systems that want to notify the central fulfillment system about an error in processing the order creates an enriched fault message (fault message with order details). This enriched fault message is queued to the AIA_LFERROR_JMSQ to get the fault processed by the order fallout framework. This fault message is processed by the Order Fallout Management framework, and Oracle OSM CFS is notified about the errors in the Line Fulfillment system for a particular order.

This service acts as a consumer, listening to the messages produced in the AIA_LFERROR_JMSQ.

23.8 Business Flow Fallout-Enabled Services

The following Create and Manage Trouble Ticket from Oracle OSM business flow services are fallout-enabled:

- UpdateSalesOrderSiebelCommsProvABCSImpl
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

- ProcessFOBillingAccountListRespOSMCFSCommsJMSProducer
- ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer
- TestOrderOrchestrationEBF
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer_RS

  default.ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl.ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl_1_0
- OSM.ABCS.ConsumeCustomerCFO_JMSAdapter
- OSM.ABCS.ConsumeCustomerCFO_JMSAdapter_RS
- OSM.ABCS.ConsumeBillingCFO_JMSAdapter
- OSM.ABCS.ConsumeBillingCFO_JMSAdapter_RS
- OSM.ABCS.ConsumeUpdateFulfillmentOrder_JMSAdapter_RS
- OSM.ABCS.ConsumeUpdateFulfillmentOrder_JMSAdapter
- OSM.ABCS.BillingResponseConsumer
- OSM.ABCS.BillingResponseConsumer_RS
- OSM.ABCS.CustomerResponseConsumer
- OSM.ABCS.CustomerResponseConsumer_RS
- OSM.ABCS.OrderOrchestrationConsumer
- OSM.ABCS.OrderOrchestrationConsumer_RS

  default.TestOrderOrchestrationEBF.TestOrderOrchestrationEBF_1_0
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsSequencer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer
- OSMPROV
- OSM.ABCS.ConsumeProvUpdate_RS
- OSM.ABCS.ConsumeProvUpdate
- OSM.ABCS.Consume_ProcessProvOrder
- OSM.ABCS.Consume_ProcessProvOrder_RS

Part II

Implementing the Delivered Integrations

Part II contains the following chapters:

24

Configuring the Process Integration for Product Lifecycle Management

This chapter provides a list of prerequisites and discusses how to configure the process integration for product lifecycle management (PLM). This includes setting up Oracle Communications Billing and Revenue Management (Oracle BRM) and configuring Siebel Customer Relationship Management (Siebel CRM) to integrate with Oracle Application Integration Architecture (Oracle AIA) for communications. It discusses how to work with domain value maps (DVMs) and cross-references, how to handling error notifications and how to configure properties located in the AIAConfigurationProperties.xml file.

This chapter includes the following sections:

- Section 24.1, "Prerequisites"
- Section 24.2, "Setting Up Oracle BRM"
- Section 24.3, "Configuring Siebel CRM to Integrate with Oracle AIA for Communications"
- Section 24.4, "Working with DVMs"
- Section 24.5, "Working with Cross-References"
- Section 24.6, "Handling Error Notifications"
- Section 24.7, "Configuring Properties in the AIAConfigurationProperties File"

24.1 Prerequisites

These are the prerequisites for the process integration for product management:

1. Oracle BRM must be set up before you can create billing products.

2. The following pricing objects and data must be created in the Oracle BRM database:

   - Services.
   - Events.
   - Resources.
   - Currency exchange rates.
   - G/L IDs.
   - Tax codes and tax suppliers.
   - Rateable Usage Metrics (RUMs).

3. You must define billing products in Oracle BRM and associate them with billing events and billing rate plans.

   Oracle BRM triggers an event that is used to synchronize the defined billing products to Siebel CRM. The synchronization in this step is based on functional events available in Oracle BRM to identify changes (additions, deletions, modifications) that triggers the integration flow to propagate those billing product changes and make the corresponding changes to Siebel CRM billing products.

24.2 Setting Up Oracle BRM

This section discusses how to set up Oracle BRM.

To set up Oracle BRM

1. Create services and events.

   New services must be added before a pricelist is created. Oracle BRM includes internet access and email services by default. A list of events must be configured to track each service. If new services are created, new events must be created to track the services.

2. Create resources.

   Each product is associated with rate plans. Resources must be created to supplement the rate plans. These include both the currency, such as USD, and the noncurrency-related resources, such as minutes.

3. Create General Ledger (GL) IDs.

   GL IDs are used to collect general ledger information from the Oracle BRM database and export it to your accounting application. Decide how to track the revenue for each type of rate, and create the appropriate GL IDs.

4. Define tax codes and tax suppliers. (Optional)

   To calculate taxes using Taxware, you must define tax codes and tax suppliers.

5. Define RUMs for events.

   RUMs are used to identify the event attributes that define rates for each event. RUM definitions are stored in the Oracle BRM database.

6. Map event types to RUMs.

   Each event must be associated with a list of RUMs. When products are created, a rate plan structure is associated with every RUM that is linked for the event.

7. Map event types to services.

   When a product is created, a set of services and events that must be rated are selected. The events are related to the service. Not all event types are valid for all services. A mapping must be defined between the event types and the services. Creating the mapping prevents you from selecting an event that is not applicable for a given service.

8. Define zones.

   For real-time rating, zones are created as single values to represent groups of values. The representative value is used in a rate plan selector.

9. Define impact categories.

For real time rating, impact categories are used to specify that particular groups of balance impacts within rates must be used. If the plan is to use attribute value grouping during rating, then some impact categories must be created.

10. Define pipeline data.

If pipeline rating is used, several types of data and pricing components must be created.

11. Set up pricing for friends and family functionality.

For more information about setting up pricing for friends and family, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "Setting Up Pricing and Rating," Working with extended rating attributes.

12. Install, configure, and run Synchronization Queue Data Manager (DM).

This DM enables you to synchronize changes in the Oracle BRM database with external applications. For example, when a product is created or modified, Synchronization Queue DM sends the data to a database queue. The data in the queue can then be retrieved by an external application. You can use the Synchronization Queue DM to synchronize data in real time, and you can use it with the pin_export_price utility to export data as a batch.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "Service Integration Components," Synchronization Queue Data Manager.

13. Set the Oracle BRM EAI parameter.

To ensure that immediate effective start dates and end dates with infinite effectivity are communicated as set as a null date value in Siebel CRM, (instead of 31-Dec-1969/01-Jan-1970), the BRM EAI parameter infranet.eai.xml_zero_epoch_as_null must be set to *True*. This setting is required for the flow to work correctly.

For more information about the behavior of effective dates based on the BRM EAI parameter, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*.

24.3 Configuring Siebel CRM to Integrate with Oracle AIA for Communications

To integrate Siebel CRM with Oracle AIA for Communications, you must:

1. Install ACR 474.

For information about how to install ACR 474, see the *Siebel Maintenance Release Guide*.

2. Set the process property UTCCanonical to *Y* in Siebel for some Siebel CRM interfaces.

For more information about which Siebel CRM interfaces require you to enable the UTCCanonical process property, see instructions for ACR 474 and ACR 508 in the *Siebel Maintenance Release Guide*.

3. Configure important data elements, including:

a. Set up a Siebel price list.

The price list is required for the product synchronization integration flow. Create a price list in Siebel CRM, and then update the AIAConfigurationProperties.xml file with the Siebel ROWID of the price list.

For more information, see the property Siebel.PriceList.ID, listed in Table 24–5.

b. Set up a Siebel organization.

Identify the organization in Siebel CRM and update the AIAConfigurationProperties.xml file.

For more information, see the property Siebel.BusinessUnit, listed in Table 24–5.

c. Set up a Siebel workspace.

Identify that workspace in Siebel CRM and update the AIAConfigurationProperties.xml file.

For more information, see the property Siebel.Product.WorkspaceName, listed in Table 24–5.

d. Set up friends and family products.

For more information about friends and family products, see Support for Friends and Family in Section 3.3.12, "Supporting Friends and Family."

e. Make workflow changes to use penalty products synchronized from Oracle BRM.

This can be done only after you run the product synchronization integration flow.

ISS Promotion Disconnect Process must be modified to use the product synchronized from Oracle BRM.

For more information about ISS Promotion Disconnect Process, see the *Siebel Order Management Guide Addendum for Communications*, "Workflows for Employee Asset-Based Ordering."

Onetime charge products must be included in the Siebel Catalog. If not, you do not see the onetime charge recommended products pick list.

After products are synchronized from Oracle BRM to Siebel CRM, and after onetime charge products have been added to a Siebel Catalog, you must associate onetime charges with Modify, Add, Change, Delete (MACD) order types.

For more information about Related Product functionality in Siebel, see the *Siebel Order Management Guide Addendum for Communications*, "Employee Asset-Based Ordering".

Define simple Special Rating products and set the composition type to *Partial*.

f. Set up service bundles:

Set Billing Type to *Service Bundle* and set Billing Service Type to the same string as the billing service bundle on the component products (that have been synchronized from Oracle BRM).

g. Set up promotions, bundling service bundles, account-level products, and discounts.

h. Add service bundles and promotions to the price list used by the product synchronization integration flow.

For more information about service bundles, see Section 3.3, "Understanding the Product Bundling Methodology."

24.4 Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle service-oriented architecture (SOA) Suite. They enable you to equate lookup codes and other static values across applications, for example, *FOOT* and *FT* or *US* and *USA*.

DVMs are static in nature, though administrators can add maps as needed. Transactional business processes never update DVMs; they only read from them. DVMs are stored in XML files and cached in memory at run time.

DVM types are seeded for the oracle Communications Order to Cash flows, and administrators can extend the list of mapped values by adding more maps. The DVM data should be synchronized with what the participating applications use. This synchronization should occur before any initial loads are run or any incremental transactional flows are initiated.

Table 24–1 lists the DVMs for the process integration for product management.

Table 24–1   DVMs

DVM	Description
PRICECHARGETYPE	Price Charge Type (common values are *One-Time* or *Recurring*.)
PRICECHARGETYPEUOM	Price Charge Type Unit Of Measure (common values are *Per Day* or *Per Month*.)
PRICETYPE_EVENT	Price Type Event (common values are *Purchase* or *Cancel*.)
PRODUCTTYPECODE	Product Type Code (common values are *Item* or *Subscription*.)
ITEM_BILLINGTYPECODE	Maps Billing Type from Oracle BRM to Siebel CRM
RESOURCE	Non-Monetary resources (*Free Minutes*, *Text Messages*, and so on).
CURRENCY_CODE	Currency codes.

For more information about DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with Message Transformations," Working with DVMs and Cross-References.

24.5 Working with Cross-References

Cross-references map and connect the records within the application network and they enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

Table 24–2 list the product management cross-references.

For more information about product management cross-references, see Appendix A, "PLM - Cross-References for the Process Integration for Product Management."

Table 24–2   Product Management Cross-References

Cross-reference Table Name	Column Names Column Values --		--	Description
ITEM_ITEMID	COMMON	SEBL_01	BRM_01	Cross references the Oracle BRM (Portal) ProductID and the Siebel CRM ProductID.
	auto generated GUID	ProductID of Siebel Product ABM.	POID of BRM Product ABM.	
PRICELINE_ID	COMMON	SEBL_01	BRM_01	Cross references the Oracle BRM (Portal) Product ID to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ITEMID.
	auto generated GUID.	Siebel PriceListItemID for the main product.	POID of BRM Product ABM.	
PRICELINETYPE_ID	COMMON	SEBL_01	BRM_01	Cross references Oracle BRM (Portal) Product's Events to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ITEMID.
	auto generated GUID.	Siebel PriceListItemID for the event product.	POID of BRM Product ABM + Event Name.	
SIEBELPRODUCTEVENTXREF	ITEM_ID_ COMMON	LINEPRICETYPECODE	--	Cross references Oracle BRM (Portal) Product's Event that is associated with the main product in Siebel CRM.
	From ITEM_ ID.COMMON	PRICELINETYPE _ ID.COMMON	--	

24.6 Handling Error Notifications

Based on the roles defined for the services, email notifications are sent if a service ends due to an error. No Oracle Application Integration Architecture (Oracle AIA)-specific errors are caused by the process integration for product management services.

For more information about the errors caused by Oracle BRM or Siebel CRM, see that product's documentation.

24.6.1 Describing Delivered Error Notification Roles and Users

The following roles and users are delivered as default values for issuing error notifications for the process integration for product management.

Actor roles and users:

- **Role:** *AIAIntegrationAdmin*

- **User:** *AIAIntegrationAdminUser*

  The default password set for all users is *welcome1*.

For more information about setting up error notifications using these values, see *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

24.7 Configuring Properties in the AIAConfigurationProperties File

Configure these properties in the AIAConfigurationProperties.xml file. It is located here: $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. Entries in the AIAConfigurationProperties.xml file are case sensitive.

For more information about requirements for working with AIAConfigurationProperties.xml, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows", How to Set Up AIA Workstation.

24.7.1 Configuring Properties for the Product Lifecycle Management Feature

Configure these properties in the AIAConfigurationProperties.xml file. Entries in the AIAConfigurationProperties.xml file are case sensitive.

Table 24–3 shows the settings for the SyncProductBRMCommsReqABCSImpl service property.

Table 24–3   SyncProductBRMCommsReqABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in the business service repository (BSR). This is used only if the request message does not contain the system instance ID.
ABCSExtension.PreXFormABMtoSyncItemCompositionListEBM	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into-point. If set to true, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokeItemCompositionEBS	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into-point.If set to true, then the Extension process is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
ABCSExtension.PreXFormABMto PriceListListEBM	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into-point. If set to true, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExension.PreInvokePriceListEBS	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into-point. If set to true, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
CallBackAddress	http://$<http.host name>:$<http.port>/soa-infra/services/default/SyncProductBRMCommsReqABCSImpl/SyncProductBRMCommsReqABCSImpl	This property is used to set the ReplyTo element in the EBM Header. The provider ABCS would use this WSAddress, if present, when sending the response.

Table 24–3 (Cont.) SyncProductBRMCommsReqABCSImpl

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList	true/false. Default = true	This property indicates if EBS call needs to be bypassed. If true, it uses the 4 properties below to identify the service it should invoke.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.Address	http:// $<http.host name>:$<http.port>/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCSImpl/SyncItemCompositionListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.PortType	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSImpl/V1}SyncItemCompositionListSiebelCommsProvABCSImpl	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSImpl/V1}SyncItemCompositionListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList	true/false. Default = true	This property indicates if EBS call needs to be bypassed. If true, it uses the 4 properties below to identify the service it should invoke.
EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.Address	http://$<http.host name>:$<http.port>/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.PortType	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.

Table 24–4 shows the settings for the SyncDiscountBRMCommsReqABCSImpl service property.

Table 24–4   SyncDiscountBRMCommsReqABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
ABCSExtension.PreXFormAB MtoEBM	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into-point. If set to true, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokeEBS	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into-point. If set to true, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
EBSOverride.Communications ItemCompositionEBSV1.SyncIt emCompositionList	true/false. Default = true	This property indicates if EBS call needs to be bypassed. If true, it uses the 4 properties below to identify the service it should invoke.
EBSOverride.Communications ItemCompositionEBSV1.SyncIt emCompositionList.Address	http:// $<http.host name>:$<http.port>/soa-infra /services/default/SyncItemC ompositionListSiebelCommsPr ovABCSImpl/SyncItemComp ositionListSiebelCommsProvA BCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.Communications ItemCompositionEBSV1.SyncIt emCompositionList.PortType	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/SyncItemCompositionList SiebelCommsProvABCSImpl/ V1}SyncItemCompositionListS iebelCommsProvABCSImpl	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications ItemCompositionEBSV1.SyncIt emCompositionList.ServiceNa me	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/SyncItemCompositionList SiebelCommsProvABCSImpl/ V1}SyncItemCompositionListS iebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.

Table 24–5 shows the settings for the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service property.

Table 24–5   ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.SiebelProductService.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Siebel Product import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
Routing.SiebelProductService.RouteToCAVS	true/false. Default = false	If true, it invokes the actual target system whose end point is indicated by the service-level property Routing.SiebelProductService.SEBL_01.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Routing.SiebelPriceListService.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Siebel PriceList web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
Routing.SiebelPriceListService.RouteToCAVS	true/false. Default = false	If true, it invokes the actual target system whose end point is indicated by the service-level property Routing.SiebelPriceListService.SEBL_01.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Siebel.SEBL_01.BusinessUnit	No default value.	All the products created belong to this business unit in the Siebel system. The value for this property should be the ID of the business unit in the Siebel system. This value must be set before product sync is run.
Siebel.SEBL_01.Product.Workspace Name	Demo Workspace	Name of the workspace to be used by Siebel. Create a workspace and update this file with that workspace name.
Siebel.Product.Workspace ReleaseFlag	Y/N. Default = Y	Indicates whether the workspace must be released after the product is synchronized.
Siebel.Product.WorkspaceReuseFlag	Y/N. Default = Y	Indicates whether the workspace must be reused for product to be synced.
Siebel.SEBL_01.PriceList.ID	No default value.	All the products created by this sync belongs to this price list in the Siebel system. The value for this property should be the ID of the price list in the Siebel system. This value must be set before product sync is run.
Siebel.PriceList.Currency	USD	Currency code of the price list mentioned in the preceding property. If the currency of the prices in PriceListEBM does not match this currency, price in Siebel is be set to 0 (zero). This value must be set before the product sync is run.

Table 24–5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
ABCSExtension.PreXFormEBMto ABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormEBMt oABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the EBM to ABM transformation).
ABCSExtension.PreInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service. (PostInvoke Application).
ABCSExtension.PreXFormPriceLi stListEBMtoItemCompositionEB M	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to EBM transformation.)
ABCSExtension.PreInvokeItemC ompositionEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PreXFormPriceLi stListEBMtoProductABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PreInvokeProdu ctABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service. (PostInvoke Application).
EBSOverride.CommunicationsIte mCompositionEBSV1.SyncItemC ompositionList	true/false. Default = true	This property indicates if EBS call needs to be bypassed. If true, it uses the 4 properties below to identify the service it should invoke.
EBSOverride.CommunicationsIte mCompositionEBSV1.SyncItemC ompositionList.Address	http:// $<http.host name>:$<http.port>/soa-infra /services/default/SyncItemCo mpositionListSiebelCommsPro vABCSImpl/SyncItemCompos itionListSiebelCommsProvAB CSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.CommunicationsIte mCompositionEBSV1.SyncItemC ompositionList.PortType	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/SyncItemCompositionList SiebelCommsProvABCSImpl/ V1}SyncItemCompositionListSi ebelCommsProvABCSImpl	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsIte mCompositionEBSV1.SyncItemC ompositionList.ServiceName	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/SyncItemCompositionList SiebelCommsProvABCSImpl/ V1}SyncItemCompositionListSi ebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsPr iceListResponseEBS.SyncPriceLis tListResponse	true/false. Default = true	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCo mmsProvABCSImpl does not invoke CommunicationsPriceListResponseEBS.

Table 24–5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.Address	http://$<http.host name>:$<http.port>/soa-infra /services/default/ProductOptimizedSyncPriceListListSiebel CommsProvABCSImpl/ProductOptimizedSyncPriceListListS iebelCommsProvABCSImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCo mmsProvABCSImpl does not invoke CommunicationsPriceListResponseEBS.
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.PortType	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/ProductOptimizedSyncPri ceListListSiebelCommsProvAB CSImpl/V1}ProductOptimized SyncPriceListListSiebelComms ProvABCSImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCo mmsProvABCSImpl does not invoke CommunicationsPriceListResponseEBS.
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.ServiceName	{http://xmlns.oracle.com/AB CSImpl/Siebel/Industry/Com ms/ProductOptimizedSyncPri ceListListSiebelCommsProvAB CSImpl/V1}ProductOptimized SyncPriceListListSiebelComms ProvABCSImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCo mmsProvABCSImpl does not invoke CommunicationsPriceListResponseEBS.

Table 24–6 shows the settings for the SyncItemCompostionListSiebelCommsProvABCSImpl service property.

Table 24–6 SyncItemCompositionListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.SiebelProductService.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_ enu/start.swe?SWEExtSourc e=SecureWebService&S WEExtCmd=Execute& WSSOAP=1	Siebel Product Import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
Routing.SiebelProductService.RouteToCAVS	true/false. Default = false	If true, it invokes the actual target system whose end point is indicated by the service-level property Routing.SiebelProductService.SEBL_ 01.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Siebel.SEBL_01.BusinessUnit	No default value.	All the products created belong to this business unit in the Siebel system. The value for this property should be the ID of the business unit in the Siebel system. This value must be set before Product Sync is run.
Siebel.Product.WorkspaceName	Demo Workspace	Name of the workspace to be used by Siebel. Create a workspace and update this file with that workspace name.
Siebel.Product.WorkspaceReleaseFlag	Y/N. Default = N	Indicates whether the workspace must be released after the product is synchronized.

Table 24–6 (Cont.) SyncItemCompositionListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Siebel.Product.WorkspaceReuseFlag	Y/N. Default = Y	Indicates whether the workspace must be reused for product to be synced.
ABCSExtension.PreXFormEBMtoABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormABMtoEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation).
ABCSExtension.PreInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).
EBSOverride.CommunicationsItemCompositionEBSV1.CommunicationsItemCompositionResponseEBSV1.SyncItemCompositionListResponse	true/false. Default = true	This property indicates if Response EBS call needs to be bypassed. If true, the ABCS first checks if the ReplyTo element is set in the EBM header. If ReplyTo is present, it uses that info. If ReplyTo is not set, it uses the 4 properties below to identify the service it should invoke to send the response.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.Address	http:// $<http.host name>:$<http.port>/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCSImpl/SyncItemCompositionListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.PortType	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSImpl/V1}SyncItemCompositionListSiebelCommsProvABCSImpl	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSImpl/V1}SyncItemCompositionListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.

24.7.2 Configuring Properties for the Query Product Class Feature

Configure these properties in the AIAConfigurationProperties.xml file. Entries in the AIAConfigurationProperties.xml file are case sensitive.

Table 24–7 shows the settings for the QueryProductClassAndAttributesSCECommsReqABCSImpl service property.

Table 24–7   QueryProductClassAndAttributesSCECommsReqABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SCE_01	SCE instance code. This is used only if the request message does not contain the target system ID.
Routing.CommunicationsClassificationEBSV1.QueryClassificationList.RouteToCAVS	true/false. Default = false	Controls whether CommunicationsClassificationEBSV1 routes messages to the verification system or to the Provider ABCS implementation.
Routing.CommunicationsSpecificationEBSV1.QuerySpecificationList.RouteToCAVS	true/false. Default = false	Controls whether CommunicationsSpecificationEBSV1 routes messages to the verification system or to the Provider ABCS implementation.
Routing.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationList.RouteToCAVS	true/false. Default = false	Controls whether CommunicationsSpecificationValueSetEBSV1 routes messages to the verification system or to the Provider ABCS implementation.
ABCSExtension.PreInvokeCommunicationsClassificationEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application.)
ABCSExtension.PostInvokeCommunicationsClassificationEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).
ABCSExtension.PreInvokeEBSQueryClassificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostXFormQueryClassificationListResponseEBMtoProductClassAndAttributesResponseABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PreXFormQueryClassificationListResponseEBMtoQuerySpecificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to EBM transformation.
ABCSExtension.PreInvokeCommunicationsSpecificationEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostInvokeCommunicationsSpecificationEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).
ABCSExtension.PreXformQueryClassificationListResponseEBMtoQuerySpecificationValueSetListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to EBM transformation.
ABCSExtension.PreInvokeCommunicationsSpecificationValueSetEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostInvokeCommunicationsSpecificationValueSetEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Table 24–7 (Cont.) QueryProductClassAndAttributesSCECommsReqABCSImpl

Property Name	Value/Default Values	Description
EBSOverride.Communications ClassificationEBSV1.QueryClassificationList.Address	http://$<http.host name>:$<http.port>/soa-infra/services/default/QueryClassificationListSiebelCommsProvABCSImpl/QueryClassificationListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.Communications ClassificationEBSV1.QueryClassificationList.PortType	QueryClassificationListSiebelCommsProvABCSImplService	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications ClassificationEBSV1.QueryClassificationList.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/QueryClassificationListSiebelCommsProvABCSImpl/V1}QueryClassificationListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications SpecificationEBSV1.QuerySpecificationList.Address	http://$<http.host name>:$<http.port>/soa-infra/services/default/QuerySpecificationListSiebelCommsProvABCSImpl/QuerySpecificationListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.Communications SpecificationEBSV1.QuerySpecificationList.PortType	QuerySpecificationListSiebelCommsProvABCSImplService	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications SpecificationEBSV1.QuerySpecificationList.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/QuerySpecificationListSiebelCommsProvABCSImpl/V1}QuerySpecificationListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications SpecificationValueSetEBSV1.QuerySpecificationValueSetList.Address	http://$<http.host name>:$<http.port>/soa-infra/services/default/QuerySpecificationValueSetListSiebelCommsProvABCSImpl/QuerySpecificationValueSetListSiebelCommsProvABCSImpl	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
EBSOverride.Communications SpecificationValueSetEBSV1.QuerySpecificationValueSetList.PortType	QuerySpecificationValueSetListSiebelCommsProvABCSImplService	PortType of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications SpecificationValueSetEBSV1.QuerySpecificationValueSetList.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/QuerySpecificationValueSetListSiebelCommsProvABCSImpl/V1}QuerySpecificationValueSetListSiebelCommsProvABCSImpl	ServiceName of the webservice that needs to be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.

Table 24–8 shows the settings for the QueryClassificationListSiebelCommsProvABCSImpl service property.

Table 24–8   QueryClassificationListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.ProductClassQuery.RouteToCAVS	true/false. Default = false	Controls whether ProductClassQuery routes messages to the verification system or to the Provider ABCS implementation.
Routing.ProductClassQuery.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Siebel ProductClassQuery import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
ABCSExtension.PreXFormEBMtoABMClassificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormABMtoEBMClassificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABSListOfSwiAdminIssClassDefinitionABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeABSListOfSwiAdminIssClassDefinitionABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Table 24–9 shows the settings for the QuerySpecificationListSiebelCommsProvABCSImpl service property.

Table 24–9   QuerySpecificationListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.QueryProductClassAttributes.RouteToCAVS	true/false. Default = false	Controls whether QueryProductClassAttributes routes messages to the verification system or to the Provider ABCS implementation.
Routing.QueryProductClassAttributes.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Siebel QueryProductClassAttributes import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
ABCSExtension.PreXFormEBMtoABMSpecificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)

Table 24–9   (Cont.) QuerySpecificationListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
ABCSExtension.PostXFormAB MtoEBMSpecificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABS AttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeAB SAttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Table 24–10 shows the settings for the QuerySpecificationValueSetListSiebelCommsProvABCSImpl service property.

Table 24–10   QuerySpecificationValueSetListSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.QueryProductClassAt tributes.RouteToCAVS	true/false. Default = false	Controls whether QueryProductClassAttributes routes messages to the verification system or to the Provider ABCS implementation.
Routing.QueryProductClassAt tributes.SEBL_01.EndpointURI	http://$<http.host name>:$<http.port>/eai_ enu/start.swe?SWEExtSourc e=SecureWebService&S WEExtCmd=Execute& WSSOAP=1	Siebel QueryProductClassAttributes import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
ABCSExtension.PreXFormEB MtoABMSpecificationValueSet ListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormAB MtoEBMSpecificationValueSet ListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABS AttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeAB SAttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

25

Configuring the Process Integration for Order Lifecycle Management

This chapter discusses how to configure the process integration for order lifecycle management (OLM). This includes setting up Oracle Communications Billing and Revenue Management (Oracle BRM) and configuring Siebel Customer Relationship Management (Siebel CRM) to integrate with Oracle Application Integration Architecture (Oracle AIA) for communications. It discusses how to work with domain value maps (DVMs) and cross-references, and handling error notifications.

This chapter includes the following sections:

- Section 25.1, "Setting Up Oracle BRM"
- Section 25.2, "Configuring Siebel CRM to Integrate with Oracle AIA for Communications"
- Section 25.3, "Working with DVMs"
- Section 25.4, "Working with Cross-References"
- Section 25.5, "Handling Error Notifications"
- Section 25.6, "Configuring the Process Integration for Order Lifecycle Management"

25.1 Setting Up Oracle BRM

This section describes how to set up Oracle BRM.

To set up Oracle BRM:

- Install and configure the Oracle BRM JCA adapter.

    For more information about how to configure the Oracle BRM JCA adapter, see the *JCA Resource Adapter Guide*, "Deploying and Configuring the Oracle BRM JCA Resource Adapter."

25.2 Configuring Siebel CRM to Integrate with Oracle AIA for Communications

To integrate Siebel CRM with Oracle AIA for Communications, you must:

1. Install ACR 474.

    For information about how to install ACR 474, see the *Siebel Maintenance Release Guide*.

2. Set the process property UTCCanonical to *Y* in Siebel for some Siebel CRM interfaces.

   For more information about which Siebel CRM interfaces require you to enable the UTCCanonical process property, see instructions for ACR 474 and ACR 508 in the *Siebel Maintenance Release Guide*.

3. Perform the following Oracle Advanced Queuing (AQ) configurations:

   - For the order flow, configure the SISOMBillingSubmitOrderWebService Siebel outbound workflow to enqueue the Siebel messages in AIA_SALESORDERJMSQUEUE.

     For this service, in Siebel, you must set the process property UTCCanonical to *Y*.

   - For updating the order information from your central fulfillment system (CFS) to Siebel CRM, enable the SWIOrderUpsert Siebel inbound web service.

     For this service, in Siebel, you must set the process property UTCCanonical to *Y*.

   - For the Special Rating List Sync Flow, configure the SWISpecialRatingList Siebel outbound workflow to enqueue the Siebel messages in AIA_SPECIALRATINGJMSQ.

For more information about the web services, see the *Siebel Order Management Guide Addendum for Communications*, "Web Services Reference."

For more information about Siebel side configuration, see *Transports and Interfaces: Siebel Enterprise Application Integration v8.1, Process of Configuring JMS Messaging Between Siebel Business Applications and Oracle SOA Suite*.

For more information about the corresponding Oracle AIA side configuration, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

25.3 Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite and enable you to equate lookup codes and other static values across applications, for example, *FOOT* and *FT* or *US* and *USA*.

DVMs are static in nature, though administrators can add maps as required. Transactional business processes never update DVMs-they only read from them. They are stored in XML files and cached in memory at run time.

DVM types are seeded for the order management flows, and administrators can extend the list of mapped values by adding more maps.

> **Caution:** The DVM names in the following table have an underscore. If you open the file in FTP mode, the underscore is replaced with *95*.

Table 25–1 lists the DVMs for the process integration for order management.

Table 25–1   DVMs

DVM	Description
SALESORDER_DYNAMICPRICEIND	Dynamic Pricing Indicator

Table 25–1 (Cont.) DVMs

DVM	Description
SALESORDER_FULFILLCOMPOSITIONTYPE	Fulfillment Composition Type Code
SALESORDER_FULFILLMENTMODECODE	Fulfillment Mode Code
SALESORDER_LINEFULFILLMENTMODECODE	Line Fulfillment Mode Code
SALESORDER_NETWORKINDICATOR	Network Indicator
SALESORDER_PARTIALFULFILLALLOWEDIND	Partial Fulfillment Mode Indicator
SALESORDER_PRIORITY	Priority
SALESORDER_PROCESSINGTYPECODE	Processing Type Code
SALESORDER_PRODUCTTYPECODE	Product Type Code
SALESORDER_REVISIONPERMISIBLECODE	Revision Permissible Code
SALESORDER_SERVICEINDICATOR.	Service Indicator
SALESORDER_STRTBILLSERVICEUSAGE	Start Billing Service Usage
SALESORDER_STATUS	Status
SALESORDER_TYPECODE	Type Code
STATE	State
PROVINCE	Province
ADDRESS_COUNTRYID	Country Code
CUSTOMERPARTY_TYPECODE	Account Type Code
ITEM_BILLINGTYPECODE	Examples of values are *Subscription, Discount, Item, Special Rating*, and so on. Billing Type Code.
SALESORDER_CHANGEDIND	Order Changed Indicator. Values are *True* or *False*. Used to validate the OrderChangedIndicator attribute. For example, The order management system can set this attribute to *True* if, as part of fulfillment, the order changes significantly such that Siebel CRM must make a copy of the customer order to preserve the customer intent before updating the working version of the order.
SALESORDER_ACTIONCODE	Sales Order Line Action Code
SALESORDER_REVISIONPERMISIBLECODE	Revision Permissible Code
SALESORDER_LINESTATUS	Order Line Status
DISCOUNT_METHODCODE	Discount Method Code
CURRENCY_CODE	Currency Code
PRICE_TYPE.	Price Type

For more information about DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with Message Transformations," Working with DVMs and Cross-References.

25.4 Working with Cross-References

Cross-references map and connect the records within the application network, and they enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

Table 25–2 lists the order management cross-references:

Table 25–2   Order Management Cross-References

Cross-reference Table Name	Column Names Column Values --		--	Description
SALESORDER_ID	COMMON	SEBL_01	BRM_01	Siebel Sales Order ID is cross-referenced.
	SalesOrderIdentification	Id	--	
SALESORDER_LINEID	COMMON	SEBL_01	BRM_01	Order Item ID from Siebel is mapped to SalesOrderLine Identification in EBM
	SalesOrderLine/Identification	OrderItem/Id	--	
INSTALLEDPRODUCT_ID	COMMON	SEBL_01	BRM_01	Siebel Asset Integration ID is mapped to Product/Service/Discount POID of Oracle BRM
	InstalledProductIdentification	AssetIntegrationId	PRODUCT/SERVICE/DISCOUNT OBJ	
ITEM_ITEMID	COMMON	SEBL_01	BRM_01	Siebel Product ID is mapped to PRODUCT/DISCOUNT OBJ of the Oracle BRM.
	ItemIdentification	ProductId	PRODUCT/DISCOUNT POID	
CUSTOMERPARTY_ACCOUNTID	COMMON	SEBL_01	BRM_01	Siebel Customer ID is mapped to Oracle BRM Account POID
	CustomerPartyAccountIdentification	AccountId	Account POID	
CUSTOMERPARTY_CONTACTID	COMMON	SEBL_01	BRM_01	Siebel Contact ID is mapped to Oracle BRM Contact POID
	CustomerPartyAccountContactIdentification	ContactId	Contact POID	
CUSTOMERPARTY_DEFAULTBALANCEGROUPID	COMMON	SEBL_01	BRM_01	Default balance group POID is mapped to common ID of account.
	CustomerPartyAccountContactIdentification	--	Balance Group POID	
CUSTOMERPARTY_PAYPROFILEID	COMMON	SEBL_01	BRM_01	Bill Profile ID from Siebel is mapped to Pay info POID of the Oracle BRM.
	PaymentProfileIdentification	BillingProfileId	Pay Info POID	

Table 25–2   (Cont.) Order Management Cross-References

Cross-reference Table Name	Column Names Column Values --		--	Description
CUSTOMERPARTY_ BILLPROFILEID	COMMON	SEBL_01	BRM_01	Bill Profile ID from Siebel is mapped to Bill info POID of the Oracle BRM.
	BillingProfileIdenti fication	BillingProfileId	Bill Info POID	
CUSTOMERPARTY_ ADDRESSID	COMMON	SEBL_01	BRM_01	Siebel Address ID is mapped to Oracle BRM Contact POID.
	CustomerPartyAcc ountAddressIdenti fication	AddressId	Address POID	

25.5 Handling Error Notifications

Based on the roles defined for the services, email notifications are sent if a service ends due to an error.

Order Fallout Management can generate trouble tickets for failed orders.

For more information about order fallout, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

Table 25–3 lists the error messages that are issued when order billing integration is called in billing initiation mode.

Table 25–3   Error Messages

Error Code	Error Text	Description
AIA_ERR_AIACOMOMPI_ 0001	Date Validation Failed: Either a Purchase Date/Cycle Start Date/ Usage Start Date should be set to the future.	In Billing Initiation mode, the ProcessFulfillmentOrderBillingBRMComm s AddSubProcess ends in an error when at least one billing date (purchase, cycle start, usage start date) is not set to the future for lines with products of type *Subscription* or *Discount*.
AIA_ERR_AIACOMOMPI_ 0002	Date Validation Failed: Purchase Date should be set to the future.	In Billing Initiation mode, the ProcessFulfillmentOrderBillingBRMComm s AddSubProcess ends in an error when the purchase date is not set to the future for lines with products of type *Item*.
AIA_ERR_AIACOMOMPI_ 0003	Purchased promotion instance does not exist for a promotion that was previously purchased. A data upgrade script was not run.	ProcessFulfillmentOrderBillingBRMComm sProvABCSImpl ends in an error if a change order is processed for data that was created using Oracle AIA for Communications 2.0/2.0.1 and the custom upgrade script was not run to create the necessary cross-reference and purchased promotion instances in BRM.
AIA_ERR_AIACOMOMPI_ 0004	Promotion referenced on Sales Order &OrderNum, Line &LineNum for &Product has not been interfaced to billing. The promotion must be interfaced to billing, before interfacing the order line that references it.	ProcessFulfillmentOrderBillingBRMComm sProvABCSImpl ends in an error if service bundle/account-level product with promotion reference is sent to billing before the corresponding promotion line.

For more information about the errors caused by Siebel CRM or Oracle BRM, see the documentation for the product.

For more information about Oracle AIA error handling, see *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

25.5.1 Describing Delivered Error Notification Roles and Users

The following roles and users are delivered as default values for issuing error notifications for the process integration for customer management.

Actor roles and users:

- **Role:** *AIAIntegrationAdmin*

- **User:** *AIAIntegrationAdminUser*

  The default password set for all users is *welcome1*.

For more information about Oracle AIA error handling, see *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

25.6 Configuring the Process Integration for Order Lifecycle Management

Configure these properties in the AIAConfigurationProperties.xml file. The file is located in $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. Entries in the AIAConfigurationProperties.xml file are case-sensitive.

For more information about requirements for working with AIAConfigurationProperties.xml, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows", How to Set Up AIA Workstation.

Table 25–4 shows the settings for the UpdateSalesOrderSiebelCommsProvABCSImpl service name.

Table 25–4   UpdateSalesOrderSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	URL for Siebel Instance web service for "Order spcLine spcItem spcUpdate spc- _ spcComplex" web service.
Routing.SWI_spcOrder_ spcUpsert.RouteToCAVS	true/false. Default = false.	Controls whether UpdateSalesOrderSiebelCommsProvABC SImpl routes messages to the CAVS or to the Siebel system.

Table 25–4   (Cont.) UpdateSalesOrderSiebelCommsProvABCSImpl

Property Name	Value/Default Values	Description
Routing.SWI_spcOrder_ spcUpsert.SEBL_01.EndpointURI	Target Endpoint URL for the Siebel upsert web service. example: http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1Status OpenFixedClosed	Target Endpoint URL for the Siebel upsert web service.
ABCSExtension.PreXformEBMtoABM	true/false. Default = false.	Whether there is any extension in the ABCS before transformation of EBM to ABM.
ABCSExtension.PreInvokeABS	true/false. Default = false.	Indicates whether there is any extension in the ABCS before invoking application business service.

Table 25–5 shows the settings for the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl service name.

Table 25–5   ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Default Siebel CRM system instance code (defined in BSR). This is used only if the Siebel Order message does not contain the EnterpriseServerName, for example, SEBL_01.
ABCSExtension.PreXformABMtoEBM	true/false. Default = false	Whether there is any extension in the ABCS before transformation of EBM to ABM.
ABCSExtension.PostXformABMtoEBM	true/false. Default = false	Whether there is any extension in the ABCS after transformation of EBM to ABM.
ABCSExtension.PreInvokeEBS	true/false. Default = false	Indicates whether there is any extension in the ABCS before invoking application business service.
ABCSExtension.PostInvokeEBS	true/false. Default = false	Indicates whether there is any extension in the ABCS after invoking application business service.

Table 25–5 (Cont.) ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

Property Name	Value/Default Values	Description
EBSOverride.Communications SalesOrderEBSV2.ProcessSales OrderFulfillment.Address	Address of web service that must be invoked. Example (Default): `http://<soa_server>:<soa port>/soa-infra/services/def ault/ProcessSalesOrderFulfil lmentOSMCFSCommsJMSProducer/ client`	This property is used to dynamically invoke any web service from this service. This holds the address of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.Communications SalesOrderEBSV2.ProcessSales OrderFulfillment.PortType	PortType of the web service that must be invoked. Example (Default): ProcessSalesOrderFulfillmentOS MCFSCommsJMSProducer	PortType of the web service that must be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.
EBSOverride.Communications SalesOrderEBSV2.ProcessSales OrderFulfillment.ServiceName	ServiceName of the web service that must be invoked. Example (Default): `http://xmlns.oracle.com/Proc essSalesOrderFulfillmentOSMC FSCommsJMSProducer}ProcessSa lesOrderFulfillmentOSMCFSCom msJMSProducer`	Service Name of the web service that must be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned above.

Table 25–6 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess service name.

Table 25–6 ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
BRM_ 01.FutureTimeThresholdForBil lingDates	8640	This property is used for future date validation in Billing Initiation. It is set to a default value of 8640 hours (360 days). This property is billing-instance-specific and must be set for any instance that the order must be sent for billing integration. For more information about how this property is used, see Section 12.4.2, "Using the Single Phase versus the Two Phase Billing Pattern."
ABCSExtension.PreInvokeEB M	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEB M	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PostprocessA ddPCM_OP_ SUBSSCRIPTION_ PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_ OP_SUBSCRIPTION_PURCHASE_DEAL.

Table 25–6   (Cont.) ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

Property Name	Value/Default Values	Description
ABCSExtension.PostProcessAddPCM_OP_ SUBSCRIPTION_ PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL.
ABCSExtension.PreProcessAddPCM_OP_CUST_MODIFY_ CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_MODIFY_CUSTOMER.
ABCSExtension.PostProcessAddPCM_OP_CUST_ MODIFY_CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_MODIFY_CUSTOMER.

Table 25–7 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess service name.

Table 25–7   ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PreProcessPCM_ OP_SUBSCRIPTION_ CANCEL_ DISCOUNTABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_ DISCOUNT.
ABCSExtention.PostprocessPCM _OP_SUBSCRIPTION_ CANCEL_ DISCOUNTABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_ DISCOUNT.
ABCSExtension.PreProcessPCM_ OP_SUBSCRIPTION_ CANCEL_ PRODUCTABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_ PRODUCT.
ABCSExtention.PostProcessPCM _OP_SUBSCRIPTION_ CANCEL_ PRODUCTABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_ PRODUCT.
ABCSExtension.PreProcessPCM_ OP_CUST_SET_ STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_SET_STATUS.

Table 25–7   (Cont.) ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

Property Name	Value/Default Values	Description
ABCSExtension.PostProcessPCM _OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_SET_STATUS.
ABCSExtension.PreProcessPCM_ OP_SUBSCRIPTION_ PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_ PURCHASE_DEAL.
ABCSExtension.PostProcessPCM _OP_SUBSCRIPTION_ PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_ PURCHASE_DEAL.

Table 25–8 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess service name.

Table 25–8   ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM
ABCSExtension.PreProcessMoveAddPCM_OP_ SUBSCRIPTION_PURCHASE_ DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_ OP_SUBSCRIPTION_PURCHASE_DEAL.
ABCSExtension.PostProcessMoveAddPCM_OP_ SUBSCRIPTION_PURCHASE_ DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_ OP_SUBSCRIPTION_PURCHASE_DEAL.

Table 25–9 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl service name.

Table 25–9   ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
Routing.BRMSUBSCRIPTIONService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMSUBSCRIPTIONService.BRM_01.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"

Table 25–9 (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

Property Name	Value/Default Values	Description
Routing.BRMSUBSCRIPTIONS ervice.BRM_02.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM2	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMSUBSCRIPTIONS ervice.BRM_03.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM3	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMCUSTService.Rou teToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMCUSTService.BR M_01.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMCUSTService.BR M_02.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM2	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMCUSTService.BR M_03.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM3	End point for Oracle BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBALService.Rout eToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMBALService_ ptt.BRM_01.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM	End point for Oracle BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBALService_ ptt.BRM_02.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM2	End point for Oracle BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBALService_ ptt.BRM_03.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM3	End point for Oracle BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMARService.Route ToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMARService.BRM_ 01.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM	End point for Oracle BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBASEService.Rou teToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMBASEService.BR M_01.EndpointURI	End point for Oracle BRM Adapter. Example: eis/BRM	End point for Oracle BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.

Table 25–9   (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

Property Name	Value/Default Values	Description
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PreProcessPCM_OP_CUST_CREATE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_CREATE_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_CREATE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_CREATE_PROFILE.
ABCSExtension.PreProcessPCM_OP_CUST_MODIFY_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_MODIFY_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_MODIFY_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_MODIFY_PROFILE.
ABCSExtension.PreProcessPCM_OP_CUST_DELETE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_DELETE_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_DELETE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_DELETE_PROFILE.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_SET_BUNDLEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_BUNDLEABM.
ABCSExtension.PostProcessPCM_OP_SUBSCRIPTION_SET_BUNDLEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_BUNDLEABM.
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.Address	Address of the web service that must be invoked. Example (Default): `{http://sdc60031sems.us.oracle.com:8042/soa-infra/services/default/ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer/client`	This property is used to dynamically invoke any web service from this service. This holds the Address of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.PortType	PortType of the web service that must be invoked. Example (Default): ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	This property is used to dynamically invoke any web service from this service. This holds the PortType of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.ServiceName	ServiceName of the web service that must be invoked. Example (Default): `{http://xmlns.oracle.com/ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer}ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer`	This property is used to dynamically invoke any web service from this service. This holds the ServiceName of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.

ABCSExtensionTable 25–10 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess service name.

Table 25–10   ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used if the request message does not contain the target information.
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PreProcessResumePCM_OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_SET_STATUS for resume scenario.
ABCSExtension.PostProcessResumePCM_OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_SET_STATUS for resume scenario.
ABCSExtension.PreProcessSuspendPCM_OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_SET_STATUS for suspend scenario.
ABCSExtension.PostProcessSuspendPCM_OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_SET_STATUS for suspend scenario.
ABCSExtension.PreProcessResumePCM_OP_SUBSCRIPTION_PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL for resume scenario.
ABCSExtension.PostProcessResumePCM_OP_SUBSCRIPTION_PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL for resume scenario.
ABCSExtension.PreProcessSuspendPCM_OP_SUBSCRIPTION_PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL for suspend scenario.
ABCSExtension.PostProcessSuspendPCM_OP_SUBSCRIPTION_PURCHASE_DEALABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL for suspend scenario.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS.

Table 25–10   (Cont.) ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

Property Name	Value/Default Values	Description
ABCSExtension.PostProcessPCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS.
ABCSExtension.PostProcessPCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS.

Table 25–11 shows the settings for the ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess service name.

Table 25–11   ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used if the request message does not contain the target information.
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PreProcessUpdate1PCM_OP_SEARCHABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the first BRM opcode call PCM_OP_SEARCH
ABCSExtension.PostProcessUpdate1PCM_OP_SEARCHABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the first BRM opcode call PCM_OP_SEARCH.
ABCSExtension.PreProcessUpdate2PCM_OP_SEARCHABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the second BRM opcode call PCM_OP_SEARCH.
ABCSExtension.PostProcessUpdate2PCM_OP_SEARCHABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the second BRM opcode call PCM_OP_SEARCH.
ABCSExtension.PreProcessUpdate1PCM_OP_CUST_MODIFY_CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the first BRM opcode call PCM_OP_CUST_MODIFY_CUSTOMER.
ABCSExtension.PostProcessUpdate1PCM_OP_CUST_MODIFY_CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the first BRM opcode call PCM_OP_CUST_MODIFY_CUSTOMER.
ABCSExtension.PreProcessUpdate2PCM_OP_CUST_MODIFY_CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the second BRM opcode call PCM_OP_CUST_MODIFY_CUSTOMER.

Table 25–11   (Cont.) ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

Property Name	Value/Default Values	Description
ABCSExtension.PostProcessUpdate2PCM_OP_CUST_MODIFY_CUSTOMERABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the second BRM opcode call PCM_OP_CUST_MODIFY_CUSTOMER.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTIONABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the BRM opcode call PCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTION.
ABCSExtension.PostProcessPCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTIONABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the BRM opcode call PCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTION.
ABCSExtension.PreProcessPCM_OP_CUST_UPDATE_SERVICESABM	true/false. Default = false	To indicate whether the ABCS has any extension before making the BRM opcode call PCM_OP_CUST_UPDATE_SERVICE.
ABCSExtension.PostProcessPCM_OP_CUST_UPDATE_SERVICESABM	true/false. Default = false	To indicate whether the ABCS has any extension after making the BRM opcode call PCM_OP_CUST_UPDATE_SERVICE.

Table 25–12 shows the settings for the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl service name.

Table 25–12   ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Default Siebel CRM system instance code (defined in BSR). This is used only if the Siebel ABM does not contain the EnterpriseServerName, for example, SEBL_01.
ABCSExtension.ABCSExtension.PreXformABMtoEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transforming ABM to EBM.
ABCSExtension.PreInvokeEBS	true/false. Default = false	To indicate whether the ABCS has any extension before making call to EBS.

Table 25–12   (Cont.) ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.PortType	PortType of the web service that must be invoked. Example (Default): ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImplService	This property is used to dynamically invoke any web service from this service. This holds the PortType of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.ServiceName	ServiceName of the web service that must be invoked. Example (Default): `{http://xmlns.oracle.com/ABCSImpl/BRM/Industry/Comms/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl/V1} ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImplService`	This property is used to dynamically invoke any web service from this service. This holds the ServiceName of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.Address	Address of the web service that must be invoked. Example (Default): `http://sdc60008sems.us.oracle.com:8090/soa-infra/services/default/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl`	This property is used to dynamically invoke any web service from this service. This holds the Address of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.

Table 25–13 shows the settings for the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl service name.

Table 25–13   ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
ABCSExtension.PreXformEBMtoABM	true/false. Default = false	To indicate whether the ABCS has any extension before transforming ABM to EBM.
ABCSExtension.PreInvokeABS	true/false. Default = false	To indicate whether the ABCS has any extension before making call to ABS.
Routing.BRMCUSTService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMCUSTService.BRM_01.EndpointURI	End point for Oracle BRM adapter. For example, eis/BRM	End point for Oracle BRM adapter. To Invoke CAVS update the above property to "true" and this property with corresponding CAVS URL

26

Configuring the Process Integration for Customer Management

This chapter describes how to configure the process integration for order lifecycle management (OLM). This includes setting up Oracle Communications Billing and Revenue Management (Oracle BRM) and configuring Siebel Customer Relationship Management (Siebel CRM) to integrate with Oracle Application Integration Architecture (Oracle AIA) for communications. It discusses how to work with domain value maps (DVMs) and cross-references, and handling error notifications.

This chapter includes the following sections:

- Section 26.1, "Setting Up Oracle BRM"
- Section 26.2, "Configuring Siebel CRM to Integrate with Oracle AIA for Communications"
- Section 26.3, "Working with DVMs"
- Section 26.4, "Working with Cross-References"
- Section 26.5, "Handling Error Notifications"
- Section 26.6, "Configuring the Process Integration for Customer Management"

26.1 Setting Up Oracle BRM

This section describes how to set up Oracle BRM.

To set up Oracle BRM:

1. You must add a phone number validation format to Oracle BRM so that the nonformatted phone numbers coming from Siebel CRM are not rejected by Oracle BRM. The format you must add is: ###-###-####.

   For more information about phone number formats, see *Oracle Communications Billing and Revenue Management Concepts*, "Using BRM with Oracle Application Integration Architecture", Validating Customer Contact Information.

2. Configure the Oracle BRM adapter.

   For more information about how to configure the Oracle BRM JCA adapter, see the *JCA Resource Adapter Guide*, "Deploying and Configuring the Oracle BRM JCA Resource Adapter."

26.2 Configuring Siebel CRM to Integrate with Oracle AIA for Communications

To integrate Siebel CRM with Oracle AIA for communications, you must:

1. Install ACR 474.

   For information about how to install ACR 474, see the *Siebel Maintenance Release Guide*.

2. Set the process property UTCCanonical to *Y* in Siebel for some Siebel CRM interfaces.

   For more information about which Siebel CRM interfaces require you to enable the UTCCanonical process property, see instructions for ACR 474 and ACR 508 in the *Siebel Maintenance Release Guide*.

26.3 Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle service-oriented architecture (SOA) Suite that enable you to equate lookup codes and other static values across applications, for example, *FOOT* and *FT* or *US* and *USA*.

DVMs are static in nature, though administrators can add maps as required. Transactional business processes never update DVMs-they only read from them. They are stored in XML files and cached in memory at run time.

DVM types are seeded for the customer management flows, and administrators can extend the list of mapped values by adding more maps.

Table 26–1 lists the DVMs for the process integration for customer management.

Table 26–1   DVMs

DVM	Columns	Description
CUSTOMERPARTY_ ACCOUNTTYPECODE.dvm	SEBL_ 01,COMMON,BRM_01	Used to get the type of the account, such as *Business* or *Customer*.
PROVINCE.dvm	SEBL_ 01,COMMON,BRM_01	Province name.
STATE.dvm	SEBL_ 01,COMMON,BRM_01	State name.
ADDRESS_COUNTRYID.dvm	SEBL_ 01,COMMON,BRM_01	Country codes.
ADDRESS_COUNTRYSUBDIVID.dvm	SEBL_ 01,COMMON,BRM_01	State codes.
CONTACT_SALUTATION.dvm	SEBL_ 01,COMMON,BRM_01	Salutation (such Mr., Mrs.). In Oracle BRM, Salutation is not a language-independent code. If Oracle BRM requires salutations in a language other than English, then you must update the DVM with the appropriate Oracle BRM values.
CURRENCY_CODE.dvm	SEBL_ 01,COMMON,BRM_01	Currency codes.
CUSTOMERPARTY_BILLPROFILE_ BILLTYPECODE.dvm	SEBL_ 01,COMMON,BRM_01	Bill type (*summary* and *detailed*).
CUSTOMERPARTY_BILLPROFILE_ FREQUENCYCODE.dvm	SEBL_ 01,COMMON,BRM_01	Billing frequency (*monthly, yearly, quarterly,* and so on.)

Table 26–1   (Cont.) DVMs

DVM	Columns	Description
CUSTOMERPARTY_PAYPROFILE_ BANKACCOUNTTYPE.dvm	SEBL_ 01,COMMON,BRM_01	Bank account type (*checking, savings*, and so on).
CUSTOMERPARTY_PAY PROFILE CREDIT_CARDTYPE.dvm	SEBL_01,COMMON	Credit Card type *(Visa, Mastercard*, and so on.)
CUSTOMERPARTY_PAYPROFILE_ DELIVERYPREF.dvm	COMMON,BRM_01	Bill media/delivery preference (*Email* or *Mail).*
CUSTOMERPARTY_PAYPROFILE_ PAYMETHODCODE.dvm	SEBL_ 01,COMMON,BRM_01	Payment profile payment method types (*credit card, direct debit*, and *invoice/bill me.*)
CUSTOMERPARTY_PAYPROFILE_ PAYTERMCODE.dvm	COMMON,BRM_01	Payment term codes.
CUSTOMERPARTY_ STATUSCODE.dvm	SEBL_ 01,COMMON,BRM_01	Account status codes.
PHONENUMBER_TYPE.dvm	SEBL_ 01,COMMON,BRM_01	Phone number type codes (*home, work, mobile, fax*, and so on.)

For more information about DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with Message Transformations," Working with DVMs and Cross-References.

26.4 Working with Cross-References

Cross-references map and connect the records within the application network, and they enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

Table 26–2 lists the customer management cross-references.

Table 26–2   Cross-References

Cross-Reference Table Name	--	Column Names: Column Values	--	Description
CUSTOMERPARTY_ ACCOUNTID.xref	COMMON	SEBL_01	BRM_01	Siebel Account ID is mapped one-to-one to the Oracle BRM Account ID.
	Account ID	Account ID	Account POID	
CUSTOMERPARTY_ BILLPROFILEID.xref	COMMON	SEBL_01	BRM_01	Siebel Bill Profile ID is mapped one-to-one to the Oracle BRM bill-info ID.
	Bill Profile ID	Bill Profile ID	bill-info POID	
CUSTOMERPARTY_ PAYPROFILEID.xref	COMMON	SEBL_01	BRM_01	Siebel Bill Profile ID is mapped one-to-one to the Oracle BRM pay-info ID.
	Payment Profile ID	Bill Profile ID	Pay-info POID	
CUSTOMERPARTY_ ADDRESSID.xref	COMMON	SEBL_01	BRM_01*	Oracle BRM Account ID is cross-referenced here if the address is used as the billing address (name-info[1]) on that account. Oracle BRM pay-info ID is cross-referenced if the address is used as the pay-info address on an account. The ACCOUNT and PAYINFO codes are prefixed to each ID to indicate the type of the ID.
	Address ID	Address ID	Account POID pay-info POID	

Table 26–2 (Cont.) Cross-References

Cross-Reference Table Name	--	Column Names: Column Values	--	Description
CUSTOMERPARTY_ CONTACTID.xref	COMMON	SEBL_01	BRM_01*	Oracle BRM Account ID is cross-referenced if the contact is used as the name (name-info[1]) on that account. Oracle BRM pay-info ID is cross-referenced if the contact is used as the name on the pay-info on an account.
	Contact ID	Contact ID	Account POID pay-info POID	
				The ACCOUNT and PAYINFO codes are prefixed to each ID to indicate the type of the ID.
CUSTOMERPARTY_ DEFAULTBALANCEG ROUPID.xref	COMMON*	BRM_01	--	This cross-reference maps the default balance group to the common account ID. This is populated after account creation in the CreateCustomerPartyProvider ABCSimpl service, and is referenced by the order flow during service creation.
	Account ID	Balance Group POID	--	
CUSTOMERPARTY_ PARTYID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01, SAP_01	--	Customer Party IDs
CUSTOMERPARTY_ PARTYLOCATIONID.x ref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01, SAP_01	--	Customer Party Location IDs
CUSTOMERPARTY_ CONTACTID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01, BRM_01, SAP_01	--	Customer Party contact IDs. Oracle BRM account ID is cross-referenced here if the contact is used as the name (name-info[1]) on that account. Oracle BRM pay-info ID is cross-referenced here if the contact is used as the name on the pay-info on an account.
				The ACCOUNT and PAYINFO codes are prefixed to each ID to indicate what type of ID it is.
CUSTOMERPARTY_ LOCATIONREFID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01	--	Customer Party Location Reference IDs.
CUSTOMERPARTY_ ACCOUNT_ PHONECOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01, SAP_01	--	Customer Party Account's Phone contact points.
CUSTOMERPARTY_ ACCOUNT_ FAXCOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01, SAP_01	--	Customer Party Account's Fax contact points.
CUSTOMERPARTY_ ACCOUNT_ WEBCOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01	--	Customer Party Account's Email/Web contact points.

Table 26–2   (Cont.) Cross-References

Cross-Reference Table Name	--	Column Names: Column Values	--	Description
CUSTOMERPARTY_ CONTACT_ PHONECOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01	--	Customer Party Contact's Phone contact points.
CUSTOMERPARTY_ CONTACT_ FAXCOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01	--	Customer Party Contact's Fax contact points.
CUSTOMERPARTY_ CONTACT_ EMAILCOMMID.xref	--	SEBL_ 01,COMMON,EBI Z_01,UCM_01	--	Customer Party Contact's Email/Web contact points.

26.5 Handling Error Notifications

Based on the roles defined for the services, email notifications are sent if a service ends due to an error. Table 26–3 lists the errors that are caused by the process integration for customer management services.

Table 26–3   Errors Caused by Customer Management Services

Service Name	Error Code	Possible Cause
SyncCustomerPartyListBRMCom msProvABCSImpl	AIA_ERR_AIACOMCMPI_0004	Subordinate account cannot have multiple parent accounts.
SyncCustomerPartyListBRMCom msProvABCSImpl	AIA_ERR_AIACOMCMPI_0005	Ambiguous subordinate bill profile update: Multiple distinct Pay-From-Party billing profile references are associated with a single Prior Pay-From-Party billing profile reference.
SyncCustomerPartyListBRMCom msProvABCSImpl	AIA_ERR_AIACOMCMPI_0006	None of the existing subordinate bill profiles are included in the move account request.
CommsProcessBillingAccountList EBF	AIA_ERR_AIACOMCMPI_0001	EBMHeader/Sender/ID is required.
CommsProcessBillingAccountList EBF	AIA_ERR_AIACOMCMPI_0002	EBMHeader/Target/ID is required.
CommsProcessBillingAccountList EBF	AIA_ERR_AIACOMCMPI_0003	Account sequence error: Pay-From accounts and billing profiles must appear before dependent and subordinate accounts and billing profiles.

26.5.1 Describing Delivered Error Notification Roles and Users

The following roles and users are delivered as default values for issuing error notifications for the process integration for customer management.

Actor roles and users:

- **Role:** *AIAIntegrationAdmin*

- **User:** *AIAIntegrationAdminUser*

The default password set for all users is *welcome1*.

For more information about setting up error notifications using these values, see *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

26.5.2 Order Fallout Management

When an order is submitted from Siebel CRM, the order may fail while customer-related information is being interfaced to Oracle BRM. In that case, a trouble ticket is generated by the Order Fallout flow. The trouble ticket generated is displayed in Siebel CRM. This helps the customer service representative (CSR) in getting notified about any error while processing the order without checking the instances in the Business Process Execution Language (BPEL) Console.

Whenever an error occurs during customer synchronization, it is propagated to the CommsProcessFulfillmentOrderBillingAccountListEBF. This enterprise business flow (EBF) generates an error notification in the error topic (similar to any other Oracle Application Integration Architecture (Oracle AIA) process). From the error topic, the order fallout flow is triggered only for the CommsProcessFulfillmentOrderBillingAccountListEBF (among all the processes in customer management process integration), thereby generating one trouble ticket for any error.

For more information about order fallout, see Chapter 21, "Understanding the Process Integration for Order Fallout Management."

26.6 Configuring the Process Integration for Customer Management

Configure these properties in the AIAConfigurationProperties.xml file. The file is located in $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. Entries in the AIAConfigurationProperties.xml file are case-sensitive.

For more information about requirements for working with AIAConfigurationProperties.xml, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows", How to Set Up AIA Workstation.

Table 26–4 shows the settings for the SyncCustomerPartyListBRMCommsProvABCSImpl service property.

Table 26–4   SyncCustomerPartyListBRMCommsProvABCSImpl Service Property

Property Name	Value/Default Values	Description
EnableAccountStatusSync	true/false. Default = false	This property when set to *True*, updates the status (active/inactive) of the account from Siebel CRM to Oracle BRM.
ABCSExtension.preformEBMto ABM	true/false. Default = false	This property governs whether the application business connector service (ABCS) Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled
ABCSExtension.PreInvokePCM _OP_BILL_GROUP_GET_ PARENTABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in Oracle AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property clearly suggests which extension point is enabled.
ABCSExtension.PostInvokePC M_OP_BILL_GROUP_GET_ PARENTABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokePCM _OP_SEARCHABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokePC M_OP_SEARCHABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokeABSP CM_OP_CUST_COMMIT_ CUSTOMERABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokeABS PCM_OP_CUST_COMMIT_ CUSTOMERABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.

Table 26–4   (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Service Property

Property Name	Value/Default Values	Description
ABCSExtension.PreInvokePCM_OP_CUSTCARE_MOVE_ACCTABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokePCM_OP_CUSTCARE_MOVE_ACCTABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokePCM_OP_CUST_UPDATE_CUSTOMERABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokePCM_OP_CUST_UPDATE_CUSTOMERABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokePCM_OP_CUST_DELETE_PAYINFOABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokePCM_OP_CUST_DELETE_PAYINFOABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostXFormABMtoEBM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.PortType	CommsProcessBillingAccountListEBF	PortType of the webservice that must be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned below.

Table 26–4 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Service Property

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.ServiceName	{http://xmlns.oracle.com/EnterpriseFlows/Industry/Comms/CommsProcessBillingAccountListEBF/V1}CommsProcessBillingAccountListEBF	ServiceName of the webservice that must be invoked dynamically. This value should be in consistent w.r.t to the Address property mentioned below.
EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.Address	http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/CommsProcessBillingAccountListEBF/client	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS/EBF, this property must be updated accordingly.
AccountLevelBalanceGroupName	Account Level Balance Group	This property is used to name the default balance group created in Oracle BRM when an account is created.
Default.SystemID	BRM_01	This property specifies the default target system ID to be populated in the enterprise business message (EBM) Header in case the value is not coming from the Requestor.
Routing.BRMCUSTService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the Oracle BRM Java EE Connector Architecture (JCA) adapter for the first instance of the Oracle BRM in case of multiple Oracle BRM instances. For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."
Routing.BRMCUSTService.RouteToCAVS	true/false. Default = false	This property specifies whether the end point should route to Composite Application Validation System (CAVS).
Routing.BRMCUSTService_ptt.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the BRM JCA adapter for the first instance of Oracle BRM in case of multiple Oracle BRM instances. For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."
Routing.BRMCUSTService_ptt.RouteToCAVS	true/false. Default = false	This property specifies whether the CAVS service must be invoked.
Routing.BRMCUSTCAREService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the Oracle BRM JCA adapter for the first instance of the Oracle BRM in case of multiple Oracle BRM instances for the CUSTCare opcode of Oracle BRM. For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."

Table 26–4   (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Service Property

Property Name	Value/Default Values	Description
Routing.BRMCUSTCAREService.RouteToCAVS	true/false. Default = false	This property specifies whether to route to CAVS Service.
Routing.BRMBILLService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the Oracle BRM JCA adapter for the first instance of the Oracle BRM in case of multiple Oracle BRM instances for the BillService opcode.
		For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."
Routing.BRMBILLService.RouteToCAVS	true/false. Default = false	This property specifies whether to Route to CAVS service.
Routing.BRMBASEService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the Oracle BRM JCA adapter for the first instance of the Oracle BRM in case of multiple Oracle BRM instances for the BRMBASEService.
		For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."
Routing.BRMBASEService.RouteToCAVS	true/false. Default = false	This property specifies whether the CAVS service should be invoked.
Routing.BRMTXNService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the Oracle BRM JCA adapter for the first instance of the Oracle BRM in case of multiple Oracle BRM instances for the TXNService opcode.
		For more information about multiple Oracle BRM systems, see Appendix F, "Configuring Multiple Oracle BRM Instances for Communications Integrations."
Routing.BRMTXNService.RouteToCAVS	true/false. Default = false	This property specifies whether to route to CAVS Service.

Table 26–5 shows the settings for the SyncAccountSiebelReqABCSImpl service property.

Table 26–5   SyncAccountSiebelReqABCSImpl Service Property

Property Name	Value/Default Values	Description
ABCSExtension.PreXformABMto EBMABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
ABCSExtension.PreInvokeEBSEB M	true/false. Default = false.	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to *True*, then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.
Default.SystemID	SEBL_01	This property specifies the default target system ID to be populated in the enterprise business message (EBM) header in case the value is not coming from the requestor.
Routing.SWICustomerParty.SEB L_01.EndpointURI	${participatingapplications.siebel.s erver.InternetProtocol}${participati ngapplications.siebel.server.host}${ participatingapplications.siebel.ser ver.port}/eai_ ${participatingapplications.siebel.s erver.Language}/start.swe?SWEEx tSource=SecureWebService&S WEExtCmd=Execute&WSSO AP=1	Endpoint URI of the SEBL_01 Siebel instance.
Routing.SWICustomerParty.Rout eToCAVS	true/false. Default = false.	This property specifies whether the end point should route to CAVS.
Routing.SWICustomerParty.CAV S.EndpointURI	http://${fp.server.soaserverhostna me}:${fp.server.soaserverport}/AI AValidationSystemServlet/syncres ponsesimulator	This property specifies the end point URL for the CAVS service.
Routing.CustomerPartyEBSV2.S yncCustomerPartyList.MessageP rocessingInstruction.Environmen tCode	PRODUCTION	--.
RoutingCustomerPartyEBSV2.Sy ncCustomerPartyList.CAVS.End pointURI	http://${fp.server.soaserverhostna me}:{fp.server.soaserverport}/AIA ValidationSystemServlet/asyncreq uestrecipient	This property specifies whether the end point should route to CAVS.
Routing.CustomerPartyEBSV2.S yncCustomerPartyList.RouteToC AVS	true/false. Default = false.	This property specifies whether the end point should route to CAVS.
Routing.TransformAppContextSi ebelService.RouteToCAVS	true/false. Default = false.	This property specifies whether the end point should route to CAVS.

Table 26–5   (Cont.) SyncAccountSiebelReqABCSImpl Service Property

Property Name	Value/Default Values	Description
Routing.TransformAppContextSiebelService.CAVS.EndpointURI	http://${fp.server.soaserverhostname}:{fp.server.soaserverport}/AIAValidationSystemServlet/asyncrequestrecipient	This property specifies whether the end point should route to CAVS.
Account.ProcessUpdateEventsOnly	true/false. Default = true.	Customers must set this property to *True*. This is required to optimize the flow. By setting this property to *True*, the Siebel connector does not propagate create events onwards. The out-of-the-box (OOTB) solution supports creation of customers only as part of the order flow. Setting this property to *False*, results in a less optimized flow, but OOTB behavior where customer creation occurs as part of the order flow remains the same. For more information, see the Oracle *Oracle Application Integration Architecture Functional Interoperability Configuration Guide*.
Contact.QueryAllEntities	true/false. Default = false.	--
Address.QueryAllEntities	true/false. Default = false.	--

Settings for the QueryCustomerPartyListSiebelProvABCSImplV2 service property.

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

Settings for the SyncAcctSiebelAggrEventConsumer service property.

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

27

Configuring the Process Integration for Order Fallout Management

This chapter discusses how to configure the process integration for order fallout management (OFM). This includes setting up Oracle Application Integration Architecture (Oracle AIA) and configuring Siebel Customer Relationship Management (Siebel CRM) to integrate with Oracle AIA for communications. It discusses how to work with domain value maps (DVMs) and cross-references, and handling error notifications.

This chapter includes the following sections:

- Section 27.1, "Setting Up Oracle AIA"
- Section 27.2, "Configuring Siebel CRM to Integrate with Oracle AIA for Communications"
- Section 27.3, "Working with DVMs"
- Section 27.4, "Working with Cross-References"
- Section 27.5, "Handling Error Notifications"
- Section 27.6, "Configuring Properties for Order Fallout Services"

27.1 Setting Up Oracle AIA

- The installation precedes the services that participate in the Oracle Order Fallout Framework in the AIA_ERROR_NOTIFICATIONS table.

  For more information about how to update the seeded data in the AIA_ERROR_ NOTIFICATIONS table, see Section 27.5.2, "Using Error Type to Control Response to Order Fallout."

- The SystemType for the applications configured in the AIA_SYSTEMS table must match the COMMON value of the TROUBLETICKET_AREA DVM.

27.2 Configuring Siebel CRM to Integrate with Oracle AIA for Communications

To integrate Siebel CRM with Oracle AIA for Communications, you must:

1. Install ACR 474.

   For information about how to install ACR 474, see the *Siebel Maintenance Release Guide*.

2. For the trouble ticket functionality to work correctly, the following dependencies must be manually added to Siebel Trouble Ticket Area's List of Values (LOVs):

```
Area:
Oracle OSM - OLM
Oracle OSM - Provisioning
BRM_01 (add for each Oracle BRM Instance. For example, BRM_02, BRM_03)

Sub-Area:
OSM OLM ABC
OSM Provisioning ABC
BRM ABC
```

Add additional values, if required.

For more information about adding values to a LOV, see your Siebel documentation.

27.3 Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle service-oriented architecture (SOA) Suite. They enable you to equate lookup codes and other static values across applications, for example, *FOOT* and *FT* or *US* and *USA*.

DVMs are static in nature, though administrators can add additional maps as required. Transactional business processes never update DVMs; they only read from them. DVMs are stored in XML files and cached in memory at run time.

DVM types are seeded for the order fallout flows, and administrators can extend the list of mapped values by adding more maps. The DVM data should be synchronized with what the participating applications use.

Table 27–1 lists the DVMs for the process integration for order fallout.

Table 27–1   Order Fallout Process Integration DVMs

DVM	Description
TROUBLETICKET_AREA	DVM to map the Area of the trouble ticket
	SEBL_01 column maps to the Area element in Siebel CRM.
	COMMON column points to the SystemCode column of the corresponding system in the AIA Systems page.
TROUBLETICKET_SUBAREA	DVM to map the SubArea of the trouble ticket.
	SEBL_01 column maps to the Sub-Area element in Siebel CRM.
	COMMON column points to the appropriate FailureSubSystemCode or the faulting service.
TROUBLETICKET_STATUS	DVM to map the status of the trouble ticket.
	SEBL_01 column maps to the Status element in Siebel CRM.
	COMMON column maps to the appropriate status in Oracle AIA.

Table 27–1   (Cont.) Order Fallout Process Integration DVMs

DVM	Description
TROUBLETICKET_SEVERITY	DVM to map the severity of the trouble ticket.
	SEBL_01 column maps to the Severity element in Siebel CRM.
	COMMON column maps to the appropriate severity (1-5) in Oracle AIA.
TROUBLETICKET_PRIORITY	DVM to map the recovery priority of the trouble ticket.
	SEBL_01 column maps to the Priority element in Siebel CRM.
	COMMON column maps to the appropriate priority (1-4) in Oracle AIA.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with Message Transformations", Working with DVMs and Cross-References.

27.4 Working with Cross-References

Cross-references map and connect the records within the application network, and they enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

Table 27–2 lists the order fallout cross-reference.

Table 27–2   Order Fallout Process Integration Cross-Reference

Cross-Reference Table Name	Column Names	Column Values	Description
TROUBLETICKET_ TROUBLETICKETID	COMMON	SEBL_01	The trouble ticket ID returned by the Siebel web service is cross-referenced to the BusinessComponentID of the TroubleTicket Response enterprise business message (EBM).
	CreateTroubleTicketResponseEBM/DataArea/CreateTroubleTicketResponse/Identification/BusinessComponentID stores this value. A randomly generated ID is used as the COMMON value for the trouble ticket and referenced with the Siebel value.	The row ID for the trouble ticket created in Siebel, which is returned in the ListOfSWITroubleTicketIO/TroubleTicket/Id element of the response of the web service, is cross-referenced.	The idea is to use this cross-referenced value for making any updates to this trouble ticket. So this cross-referencing is done only when the response is sought from the process CreateTroubleTicketSiebelCommsProvABCSImpl

27.5 Handling Error Notifications

Based on the roles defined for the services, email notifications are sent if a service ends due to an error.

Table 27–3 lists the localized custom errors that are caused by the order fallout management services for data insufficiency for creating a trouble ticket.

Table 27–3   Localized Custom Errors

Error Code	Message Text
AIA_ERR_AIACOMOFMPI_0001	Data Insufficient for Trouble Ticket Creation. Order Originating System Code not available.
AIA_ERR_AIACOMOFMPI_0002	Data Insufficient for Trouble Ticket Creation. Order ID not available.

For more information about the errors caused by Siebel CRM or Oracle Billing and Revenue Management (Oracle BRM), see the documentation for that product.

For more information about Oracle AIA error handling, see the *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

27.5.1 Describing Delivered Error Notification Roles and Users

The following roles and users are delivered as default values for issuing error notifications for the process integration for order fallout management.

Actor roles and users:

- **Role:** *AIAIntegrationAdmin*

- **User:** *AIAIntegrationAdminUser*

  The default password set for all users is *welcome1*.

For more information about the errors caused by Siebel CRM or Oracle BRM, see the documentation for that product.

For more information about Oracle AIA error handling, see the *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

27.5.2 Using Error Type to Control Response to Order Fallout

The ERROR_TYPE column in the AIA Error Notifications table (AIA_ERROR_NOTIFICATION) determines what happens when there is a failure during order processing.

The supported values for ERROR_TYPE are:

- AIA_EH_DEFAULT - Generates the standard Oracle AIA error notification.

- AIA_ORDERFALLOUT_CFS - Results in Oracle AIA notifying an order management system or central fulfillment system (such as Oracle Order and Service Management (Oracle OSM)) regarding the order fallout so that it can create and manage the trouble ticket. This value enables the default Order Fallout handling for the *Create and Manage Trouble Ticket for Order Fallout* business flow.

- AIA_ORDERFALLOUT_TTS - Results in Oracle AIA creating a trouble ticket for the order fallout. This value enables the default Order Fallout handling for the *Create Trouble Ticket for Order Fallout* business flow.

The value AIA_EH_DEFAULT can be combined with the value AIA_ORDERFALLOUT_CFS or AIA_ORDERFALLOUT_TTS, using a comma as the separator. For example, AIA_EH_DEFAULT,AIA_ORDERFALLOUT_CFS results in the actions associated with both the values being triggered.

> **Note:** You cannot have both AIA_ORDERFALLOUT_CFS and AIA_ORDERFALLOUT_TTS values specified for a given record.

If both the Oracle Communications Order to Cash: Siebel CRM pre-built integration option and the Oracle Communications Order to Cash: Oracle OSM pre-built integration option is installed, the seeded value for ERROR_TYPE is AIA_ORDERFALLOUT_CFS. If the Oracle Communications Order to Cash: Siebel CRM pre-built integration option is installed alone (without the Oracle Communications Order to Cash: Oracle OSM pre-built integration option) the seeded value for ERROR TYPE is AIA_ORDERFALLOUT_TTS.

Different ERROR_TYPE values can be given for different combinations of BPEL and ESB service, business process, system code, and error code. As delivered, Oracle AIA seeds these values for all order services. In cases where a service is used in multiple business processes, it is separately seeded for each business process.

In any given order service, there can be two categories of errors:

- **Business Errors**

  A business error is usually due to invalid or incomplete data on the order or missing setup in the end fulfillment system, which results in the request to process an order failing. In this case, for the order to be successfully processed, either the order must be corrected or revised and resubmitted, or the setup in the end fulfillment system must be corrected and the order resubmitted. For this type of error, Order Fallout should be triggered.

  This type of error usually happens when an order reaches either the participating or the edge application (such as Oracle BRM). The expectation is that the fault coming from the application is a BPEL error code: "{http://schemas.oracle.com/bpel/extension}bindingFault". Oracle BRM 7.4 returns a bindingFault when it sees a business error in the order.

- **All Other Errors**

  This includes system errors. System errors can arise when a certain system (such as Oracle BRM or BRM JCA Adapter) is down. The assumption is that there is actually nothing wrong with the order data itself and when system errors are addressed, the order can be resubmitted without any changes. For these types of errors, Order Fallout should not occur.

Order services are delivered seeded with two entries in the AIA_ERROR_NOTIFICATIONS table:

- Error Code - "{http://schemas.oracle.com/bpel/extension}bindingFault

  The seeded value for Error Type is either AIA_EH_DEFAULT,AIA_ORDERFALLOUT_TTS or AIA_EH_DEFAULT,AIA_ORDERFALLOUT_CFS. The expected behavior for this case is both standard AIA error notification and order fallout processing occurs.

- Error Code - null or no value

  The seeded value for Error Type is AIA_EH_DEFAULT. The expected behavior for this case is only standard Oracle AIA error notification occurs.

Table 27–4 is an example entry for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess order service.

Table 27–4   Example Entry for ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Order Service

Error Code	Service Name	Error Type	Error Extn Handler
--	ProcessFullmentOrderBillingBRMCommsAddSubProcess	AIA_EH_DEFAULT	AIACOM_OFM_EXT
{http://schemas.oracle.com/bpel/extension}bindingFault	ProcessFullmentOrderBillingBRMCommsAddSubProcess	AIA_EH_DEFAULT,AIA_ORDERFALLOUT_CFS	AIACOM_OFM_EXT

If additional error codes are also classified as business errors, you can add new entries into the AIA_ERROR_NOTIFICATIONS table with the appropriate Error Code value.

> **Note:** The Error Extn Handler value for all order service entries must be AIACOM_OFM_EXT. This is required so that the correct information is in the fallout and the standard error notification.

To update ERROR_TYPE for seed data in the Error Notifications table:

1. Open the Application Integration Architecture homepage by logging in to http://<httphost>:<soaport>/AIA.

2. Click the **Setup** link. This automatically directs you to the Setup - Error Notifications page.

3. Update the **Error Type** column with the appropriate value for each service for which you want to change the configuration:

   For example, if you want system errors to trigger order fallout, update this column on the respective records to AIA_EH_DEFAULT,AIA_ORDERFALLOUT_TTS. This indicates that if a particular service errors out, a standard Oracle AIA error notification is created and the error message is sent to Oracle AIA for fallout.

4. Click **Save** to save your changes.

5. Restart Fusion Middleware (FMW).

If you must perform a bulk update for all of the processes, you can use a SQL script to update the table ERROR_TYPE column in the AIA_ERROR_NOTIFICATIONS table with the appropriate values. See the $AIA_HOME/pips/Communications/O2C/DatabaseObjects/AIA_OFM_CreateOrderFalloutAIAErrorNotificationsData.sql for reference. After the table is updated, you must restart FMW.

For more information about Oracle AIA error handling, see the *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*, "Introduction to Oracle AIA Error Handling" and "Using Trace and Error Logs."

27.6 Configuring Properties for Order Fallout Services

Configure the properties for these services in the AIAConfigurationProperties.xml file.

- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCSImpl

If you have installed the Oracle Communications Order to Cash for Siebel CRM, Oracle OSM, and Oracle BRM integration, you must configure the properties for these services in the AIAConfigurationProperties.xml file.

- UpdateTroubleTicketSiebelCommsProvABCSImpl

- CreateOrderFalloutNotificationOSMCFSCommsProvImpl

It is located here: $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. All the property values are case-sensitive. All Boolean values are in lowercase.

For more information about requirements for working with AIAConfigurationProperties.xml, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows", How to Set Up AIA Workstation.

Table 27–5 shows the settings for the CreateTroubleTicketAIACommsReqImpl service name.

serviceName="{http://xmlns.oracle.com/ABCSImpl/AIA/Industry/Comms/CreateTroubleTicketAIACommsReqImpl/V1}CreateTroubleTicketAIACommsReqImpl

Table 27–5   CreateTroubleTicketAIACommsReqImpl Settings

Property Name	Value/Default Values	Description
Sender.Default. SystemID	COMMON	Use this only if the request message does not contain the system instance ID. This value is always COMMON because this service is triggered from Oracle AIA.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.PortType	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCSImpl/V1}CreateTroubleTicketSiebelCommsProvABCSImplService	Port Type of the provider ABCS/EBS service of the CreateTroubleTicket operation.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.Address	http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketSiebelCommsProvABCSImpl/CreateTroubleTicketSiebelCommsProvABCSImpl	Address of the provider ABCS/EBS service CreateTroubleTicket operation.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCSImpl/V1}CreateTroubleTicketSiebelCommsProvABCSImplService	Service Name of the provider ABCS/EBS service of the CreateTroubleTicket operation.
TroubleTicket.DefaultSeverity	Any number from 1 to 5. Default = 2.	If the fault message does not have any severity associated with it, the default severity is assigned to the fault message and reflects the same in the trouble ticket.
TroubleTicket.DefaultPriority	Any number from 1 to 4. Default = 2.	This service assigns the recovery priority for the trouble ticket by default to the value specified in this configuration property.

Table 27–6 shows the settings for the CreateTroubleTicketSiebelCommsProvABCSImpl service name.

{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicket
SiebelCommsProvABCSImpl/V1}CreateTroubleTicketSiebelCommsProvABCSImpl

Table 27–6   CreateTroubleTicketSiebelCommsProvABCSImpl Settings

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property. TroubleTicket.UseDefaultInstance is set to true.
ABCSExtension.PreXformEBMtoABMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).
ABCSExtension.PostXformABMtoEBMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation).
ABCSExtension.PreInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
ABCSExtension.PostInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.PortType	{http://xmlns.oracle.com/Industry/Comms/CreateTroubleTicketRespOSMCFSCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCommsJMSProducer	Port Type of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.Address	http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketRespOSMCFSCommsJMSProducer	Address of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.ServiceName	{http://xmlns.oracle.com/Industry/Comms/CreateTroubleTicketRespOSMCFSCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCommsJMSProducer	Service Name of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation
Routing.SWI_spcTrouble_spcTicket_spcService.RouteToCAVS	true/false Default: false	Indicates whether the Partner link SWI_spcTrouble_spcTicket_spcService should be routed to CAVS or the actual application.
Routing.SWI_spcTrouble_spcTicket_spcService.SEBL_01.EndpointURI	Endpoint URI of the SEBL_01 Siebel instance	Endpoint URI of the SEBL_01 Siebel instance or CAVS simulator (if RouteToCAVS is true.)

Table 27–6 (Cont.) CreateTroubleTicketSiebelCommsProvABCSImpl Settings

Property Name	Value/Default Values	Description
TroubleTicket.GenerateTroubleTicketResponse	true/false Default: false	CreateTroubleTicketSiebelCommsProvABCSImpl creates a trouble ticket response message (creates a cross-reference for the trouble ticket ID with the Siebel ID) and invokes the CommunicationsTroubleTicketResponseEBSV1 if this property is set to true or if the response code attribute is not null. Otherwise, this service acts only as a fire-and-forget flow and ignores the response.
TroubleTicket.UseDefaultInstance	true/false Default: false	If set to true, overwrites the target Siebel instance to the default instance indicated by the property Default.SystemID. Gives the user an option to create a trouble ticket in a Siebel instance different from the one where the order was placed.
TroubleTicket.SR_TYPE	Order Failure	SR_TYPE identifies that the trouble ticket is for Order Failure. Siebel web service expects this value to be Order Failure for Order Failure Trouble Tickets.

Table 27–7 shows the settings for the UpdateTroubleTicketSiebelCommsProvABCSImpl service name.

{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/UpdateTroubleTicketSiebelCommsProvABCSImpl/V1}UpdateTroubleTicketSiebelCommsProvABCSImpl

Table 27–7 UpdateTroubleTicketSiebelCommsProvABCSImpl Settings

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property TroubleTicket.UseDefaultInstance is set to true.
ABCSExtension.PreXformEBMtoABMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).
ABCSExtension.PreInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
Routing.SWI_spcTrouble_spcTicket_spcService.RouteToCAVS	true/false Default: false	Indicates whether the Partner link SWI_spcTrouble_spcTicket_spcService should be routed to CAVS or the actual application.
Routing.SWI_spcTrouble_spcTicket_spcService.SEBL_01.EndpointURI	Endpoint URI of the SEBL_01 Siebel instance	Endpoint URI of the SEBL_01 Siebel instance or CAVS simulator (if RouteToCAVS is true.)

Table 27–8 shows the settings for the CreateOrderFalloutNotificationOSMCFSCommsProvImpl service name.

{http://xmlns.oracle.com/ABCSImpl/OSMCFS/Industry/Comms/CreateOrderFallo
utNotificationOSMCFSCommsProvImpl/V1}CreateOrderFalloutNotificationOSMCFS
CommsProvImpl

Table 27–8   CreateOrderFalloutNotificationOSMCFSCommsProvImpl Settings

Property Name	Value/Default Values	Description
Default.SystemID	OSMCFS_01	Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property TroubleTicket.UseDefaultInstance is set to true.
Routing.CreateOrderFallout Notification.OSMCFS_ 01.EndpointURI	http://${http.host name}:${http.port}/orabpel/defa ult/CreateOrderFalloutNotificati onOSMCFSCommsJMSProducer /1.0	Endpoint URI of the service that sends the fallout notification to OSMCFS_01. The default value is the EndpointURI of the out-of-the-box JMS enqueuing or producing service that enqueues or produces the message in the OSM JMS Queue.
Fault.DefaultSeverity	Any number from 1 to 5 Default - 2	If the fault message has no severity associated with it, assigns the default severity to the fault message and reflects the same in the trouble ticket.

The following fields extract the localized values using the aia:getAIALocalizedString xpath function:

EBM Field Name: DataArea / CreateTroubleTicket / Description

Siebel Field Name: Description

ResourceBundle - oracle.apps.aia.core.i18n.AIAListResourceBundle

ResourceBundle Key - TROUBLETICKET_DESCRIPTION

Resource Bundle Value: **SalesOrder- {OrderNumber} # {OrderRevision}for Account {AccountName} failed at {Timestamp}**

--

EBM Field Name: EBMHeader/BusinessScope/ID

Siebel Field Name: Abstract

ResourceBundle - oracle.apps.aia.core.i18n.AIAListResourceBundle

ResourceBundle Key - TROUBLETICKET_ABSTRACT

Resource Bundle Value: **[{Timestamp}] Trouble Ticket for (Sales)Order - {OrderNumber} # {OrderRevision}**

PLM - Cross-References for the Process Integration for Product Management

This appendix describes the cross references used in the process integration for product lifecycle management (PLM) and provides information about the product synchronization flow and the discount synchronization flow.

This appendix contains the following sections:

- Section A.1, "Integration Solution Cross-References"
- Section A.2, "Product Synchronization Flow"
- Section A.3, "Discount Synchronization Flow"

A.1 Integration Solution Cross-References

Table A–1 lists the integration solution cross-references.

Table A–1   Integration Solution Cross-References

Operation	Entity	Siebel CRM ID	Oracle BRM ID
Inserts/Refers	ITEM_ITEMID	Product ID	Product ID
Inserts/Refers	PRICELINE_ID (main products only)	Price Line ID to Common ITEM_ITEMID of main product	Product ID
Inserts/Refers	PRICELINETYPE_ID (for event/special type products)	Price Line ID to Common ITEM_ITEMID	Generated Product ID for Event products (ProductIDEvent Name)
Inserts/Refers	SIEBELPRODUCTEVENTXREF	Common ITEM_ITEMID for the parent product to Common PRICELINETYPE_ID for event product	$

A.1.1 Cross-Reference Values

The following values denote the entries made to the cross-reference table and what they mean.

ITEM_ITEMID: cross references the Oracle BRM (Portal) ProductID and the Siebel CRM ProductID.

`COMMON`: auto generated GUID

`BRM_01`: POID of BRM Product ABM.

`SEBL_01`: ProductID of Siebel Product ABM.

PRICELINE _ID: cross references the Oracle BRM (Portal) Product ID to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ITEMID.

`COMMON`: auto generated GUID.

`BRM_01`: POID of BRM Product ABM.

`SEBL_01`: Siebel PriceListItemID for the main product.

`ITEM_ID_COMMON`: From ITEM_ID.COMMON.

PRICELINETYPE _ID: cross references Oracle BRM (Portal) Product's Event to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ITEMID.

`COMMON`: Auto generated GUID.

`BRM_01`: POID of BRM Product ABM + Event Name.

`SEBL_01`: Siebel PriceListItemID for the event product.

`ITEM_ID_COMMON`: From ITEM_ID.COMMON.

SIEBELPRODUCTEVENTXREF: cross references Oracle BRM (Portal) Product's Event that is associated with the main product in Siebel CRM.

`ITEM_ID_COMMON`: From ITEM_ID.COMMON

`LINEPRICETYPECODE`: PRICELINETYPE _ID.COMMON

A.2 Product Synchronization Flow

Figure A–1 illustrates the events that occur for product synchronization. Tables Table A–2, Table A–3, Table A–4, Table A–5, Table A–6, Table A–7, and Table A–8 describe the entries that are made in the XREF_DATA table for each event.

Figure A–1   Product Synchronization Flow
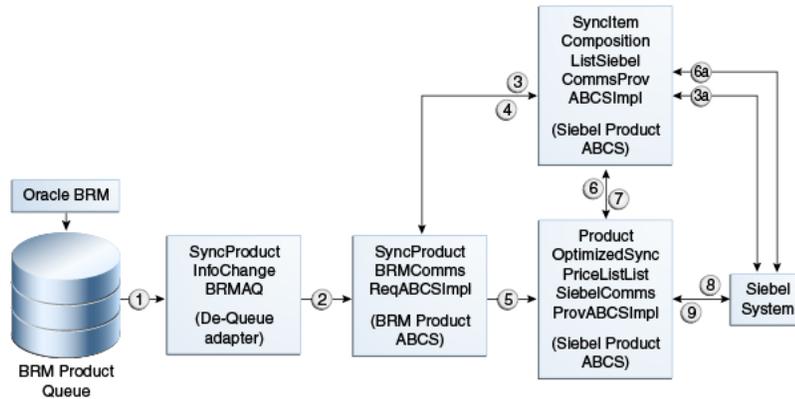
1. Before the call 3, which SyncProductBRMCommslReqABCSImpl makes to SyncItemCompositionListSiebelCommsProvABCSImpl, the entries listed in Table A–2 are made in the XREF_DATA table.

Table A–2   XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	\<POID of BRM product>
ITEM_ITEMID	COMMON GUID1

2. During the response back from Siebel CRM to SyncItemCompositionListSiebelCommsProvABCSImpl, the entry listed in Table A–3 is made in the XREF_DATA table.

Table A–3 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	<ProductID in Siebel >

3. Before the call 5 from SyncProductBRMCommslReqABCSImpl to ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl is made, the entries listed in Table A–4 are made in XREF_DATA table.

Table A–4 XREF_DATA

XREF_TABLE_NAME	VALUE
PRICELINE_ID	POID of BRM product
PRICELINE_ID	COMMON GUID2
PRICELINETYPE_ID	POID of BRM Event product
PRICELINETYPE_ID	COMMON GUID2

4. Before the call 6 from ProductOptimizedSyncPriceListListSiebelProvABCSImpl to SyncItemCompositionListSiebelCommsProvABCSImpl is made, the entries listed in Table A–5 are made in the XREF_DATA table.

Table A–5 XREF_DATA

XREF_TABLE_NAME	VALUE
SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE GUID2
SIEBELPRODUCTEVENTXREF	ITEM_ID_COMMON GUID1

5. During the response from SyncItemCompositionListSiebelCommsProvABCSImpl for the call 7, the entries listed in Table A–6 are made in the XREF_DATA table.

Table A–6 XREF_DATA

XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ITEMID	COMMON GUID2
ITEM_ITEMID	< ProductID in Siebel of Event Product >

> **Note:** For the simple product synchronization, the previous call is not made because the main product is synchronized as an *Item*.

6. Before the call 8 from ProductOptimizedSyncPriceListListSiebelProvABCSImpl to Siebel System is made, the entries listed in Table A–7 are made in the XREF_DATA table.

Table A–7 XREF_DATA

XREF_TABLE_NAME	VALUE
PRICELINE_ID	ITEM_ID_COMMON

Table A–7   (Cont.) XREF_DATA

XREF_TABLE_NAME	VALUE
PRICELINETYPE_ID (in case of multi-event product)	ITEM_ID_COMMON

7. During the response from the Siebel system, ProductOptimizedSyncPriceListListSiebelProvABCSImpl, the entries listed in Table A–8 are made in the XREF_DATA table.

Table A–8   XREF_DATA

XREF_TABLE_NAME	VALUE
PRICELINE_ID	< ProductID in Siebel for Event Product >
PRICELINETYPE_ID (in case of multi-event product)	< ProductID in Siebel for Event Product >

A.2.1 Simple Product Synchronization Example

Consider an actual scenario in which a simple product is being synchronized from Oracle Communications Billing and Revenue Management (Oracle BRM) to Siebel Customer Relationship Management (Siebel CRM).

Simple product synchronization example:

1. Create products in Oracle BRM to be synchronized with Siebel CRM, as shown in Figure A–2.

Figure A–2   Create Products in Oracle BRM

2. Verify the synced records in Siebel CRM, as shown in Figure A–3.

Figure A–3   Synced Records in Siebel CRM
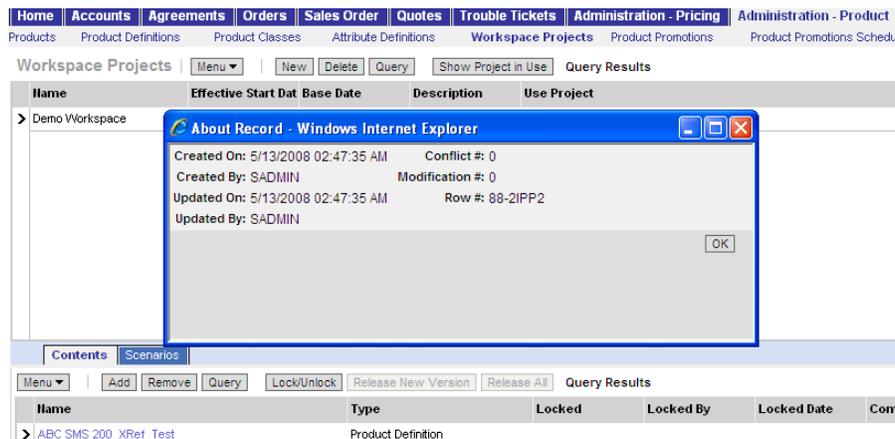
3. Verify the data entered into the XREF_DATA table is correct as shown in tables Table A–9, Table A–10, Table A–11, Table A–12, Table A–13, and Table A–14.

These tables show how data entered into the cross-reference table correspond to the points 1 through 7 explained previously.

Table A–9 Table corresponding to point 1

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ITEMID	BRM_01	<ROWNUM_1>	<BRM_PROD_01>
2	ITEM_ITEMID	COMMON	<ROWNUM_1>	<COMMON_PROD_01>

Table A–10 Table corresponding to point 2

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ITEMID	SEBL_01	<ROWNUM_1>	<CRM_PROD_01>

Table A–11 Table corresponding to point 3

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	BRM_01	<ROWNUM_2>	<BRM_PROD_01>
2	PRICELINE_ID	COMMON	<ROWNUM_2>	<COMMON_PRICE_ID1>
3	PRICELINETYPE_ID	BRM_01	<ROWNUM_3>	<BRM_PROD_01_EVENT1>
4	PRICELINETYPE_ID	COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_ID1>

Table A–12 Table corresponding to point 5

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE	<ROWNUM_4>	<COMMON_PRICETYPE_ID1>
2	SIEBELPRODUCTEVENTXREF _ID	ITEM_ID_COMMON	<ROWNUM_4>	<COMMON_PROD_01>

Table A–13 Table corresponding to point 7

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	ITEM_ID_COMMON	<ROWNUM_2>	<COMMON_PROD_01>
2	PRICELINE_ID	SEBL_01	<ROWNUM_2>	<CRM_PRICE_01>

Table A–14 Complete entry for product sync

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	BRM_01	<ROWNUM_1>	<BRM_PROD_01>

Table A–14 (Cont.) Complete entry for product sync

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
2	ITEM_ID	COMMON	<ROWNUM_1>	<COMMON_PROD_01>
3	ITEM_ID	SEBL_01	<ROWNUM_1>	<CRM_PROD_01>
4	PRICELINE_ID	BRM_01	<ROWNUM_2>	<BRM_PROD_01>
5	PRICELINE_ID	COMMON	<ROWNUM_2>	<COMMON_PRICE_ID1>
6	PRICELINETYPE_ID	BRM_01	<ROWNUM_3>	<BRM_PROD_01_EVENT1>
7	PRICELINETYPE_ID	COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_ID1>
8	SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE	<ROWNUM_4>	<COMMON_PRICETYPE_ID1>
9	SIEBELPRODUCTEVENTXREF _ID	ITEM_ID_COMMON	<ROWNUM_4>	<COMMON_PROD_01>
10	PRICELINE_ID	ITEM_ID_COMMON	<ROWNUM_2>	<COMMON_PROD_01>
11	PRICELINE_ID	SEBL_01	<ROWNUM_2>	<CRM_PRICE_01>

A.2.2 Complex Product Synchronization Example

Consider the scenario in which a complex product is being synchronized from Oracle BRM to Siebel CRM.

Complex product synchronization example:

1. Create products in Oracle BRM to be synchronized with Siebel CRM, as shown in Figure A–4.

Figure A–4 Create Products in Oracle BRM

2. Verify the synced records in Siebel CRM, as shown in Figure A–5.

Figure A–5   Synced Records in Siebel CRM

3. Verify the data entered into the XREF_DATA table is correct as shown in tables
   Table A–15, Table A–16, Table A–17, Table A–18, Table A–19, Table A–20, and
   Table A–21.

   These tables show how data entered into the cross-reference table correspond to
   the points 1 through 7 explained previously.

Table A–15   Table corresponding to point 1

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ITEMID	BRM_01	<ROWNUM_1>	<BRM_PROD_01>
2	ITEM_ITEMID	COMMON	<ROWNUM_1>	<COMMON_PROD_01>

Table A–16   Table corresponding to point 2

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ITEMID	SEBL_01	<ROWNUM_1>	<CRM_PROD_01>

Table A–17   Table corresponding to point 3

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	COMMON	<ROWNUM_2>	<BRM_PROD_01>
2	PRICELINE_ID	BRM_01	<ROWNUM_2>	<COMMON_PRICE_01>
3	PRICELINETYPE_ID	COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_01>
4	PRICELINETYPE_ID	BRM_01	<ROWNUM_3>	<BRM_PROD_01_EVENT1>
5	PRICELINETYPE_ID	BRM_01	<ROWNUM_4>	<BRM_PROD_01_EVENT2>
6	PRICELINETYPE_ID	COMMON	<ROWNUM_4>	<COMMON_PRICETYPE_02>

Table A–18   Table corresponding to point 4

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE	<ROWNUM_4>	<COMMON_PRICETYPE_01>
2	SIEBELPRODUCTEVENTXREF _ID	ITEM_ID_COMMON	<ROWNUM_4>	<COMMON_PROD_01>

Table A–19   Table corresponding to point 5

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1 ITEM_ID		COMMON	<ROWNUM_5>	<COMMON_PROD_02>
2 ITEM_ID		SEBL_01	<ROWNUM_5>	<CRM_PROD_02>

Table A–20   Table corresponding to points 6 and 7

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1 PRICELINE_ID		ITEM_ID _COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_01>
1 PRICELINE_ID		SEBL_01	<ROWNUM_3>	<CRM_ITEM_PRICE_01>
2 PRICELINETYPE_ID		ITEM_ID _COMMON	<ROWNUM_4>	<COMMON_PRICETYPE_02>
2 PRICELINETYPE_ID		SEBL_01	<ROWNUM_4>	<CRM_ITEM_PRICE_02>

Table A–21   Complete entry for product sync

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	BRM_01	<ROWNUM_1>	<BRM_PROD_01>
2	ITEM_ID	COMMON	<ROWNUM_1>	<COMMON_PROD_01>
3	ITEM_ID	SEBL_01	<ROWNUM_1>	<CRM_PROD_01>
4	PRICELINE_ID	COMMON	<ROWNUM_2>	<BRM_PROD_01>
5	PRICELINE_ID	BRM_01	<ROWNUM_2>	<COMMON_PRICE_01>
6	PRICELINETYPE_ID	COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_01>
7	PRICELINETYPE_ID	BRM_01	<ROWNUM_3>	<BRM_PROD_01_EVENT1>
8	PRICELINETYPE_ID	BRM_01	<ROWNUM_4>	<BRM_PROD_01_EVENT2>
9	PRICELINETYPE_ID	COMMON	<ROWNUM_4>	<COMMON_PRICETYPE_02>
10	SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE	<ROWNUM_4>	<COMMON_PRICETYPE_01>
11	SIEBELPRODUCTEVENTXREF _ID	ITEM_ID_COMMON	<ROWNUM_4>	<COMMON_PROD_01>
12	ITEM_ID	COMMON	<ROWNUM_5>	<COMMON_PRICETYPE_02>

Table A–21   (Cont.) Complete entry for product sync

$	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
13	ITEM_ID	SEBL_01	<ROWNUM_5>	<CRM_PROD_02>
14	PRICELINE_ID	ITEM_ID _COMMON	<ROWNUM_3>	<COMMON_PRICETYPE_01>
16	PRICELINE_ID	SEBL_01	<ROWNUM_3>	<CRM_ITEM_PRICE_01>
15	PRICELINETYPE_ID	ITEM_ID _COMMON	<ROWNUM_4>	<COMMON_PRICETYPE_02>
17	PRICELINETYPE_ID	SEBL_01	<ROWNUM_4>	<CRM_ITEM_PRICE_02>

Figure A–6 shows a high-level overview of how the mappings are maintained in the cross-reference table.

Figure A–6   Cross-Reference Table
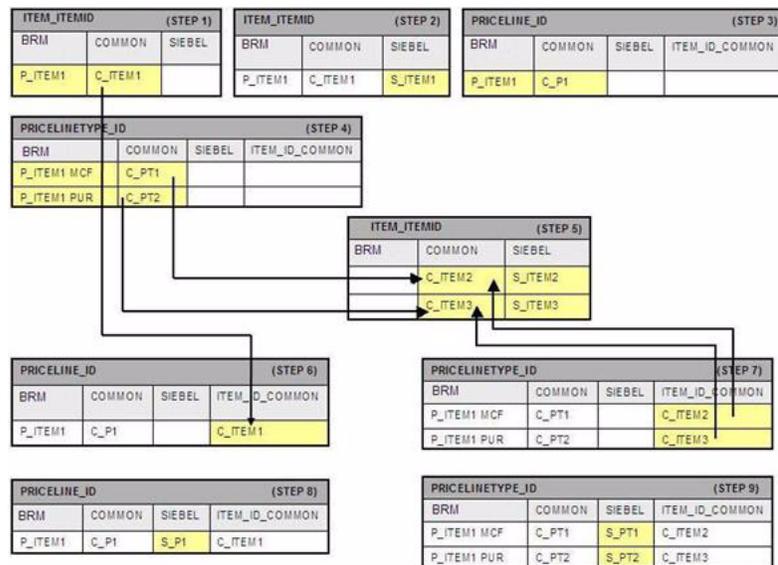
A.3 Discount Synchronization Flow

Figure A–7 illustrates the events that occur for the discount synchronization flow.

Figure A–7   Discount Synchronization Flow
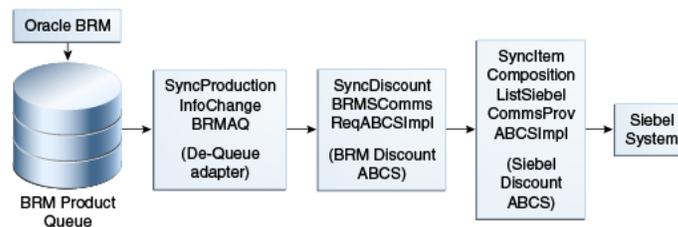
1. Before the call 3, which SyncDiscountBRMCommsReqABCSImpl makes to SyncItemCompositionListSiebelCommsprovABCSImpl, the entries listed in Table A–22 are made in the XREF_DATA table: ITEM_ITEMID, COMMON, POID of BRM product.

Table A–22 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	COMMON GUID
ITEM_ITEMID	<POID OF BRM PRODUCT>

2. During the response from the Siebel system, a SyncItemCompositionListSiebelCommsProvABCSImpl the entry listed in Table A–23 is made in the XREF_DATA table with the value: ProductID of Siebel Product.

Table A–23 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	<PRODUCTID OF SIEBEL PRODUCT>

A.3.1 Discount Synchronization Example

Consider an actual scenario in which a discount is being synchronized from Oracle BRM to Siebel CRM.

Discount synchronization example:

1. Create discounts in Oracle BRM to be synchronized with Siebel CRM, as shown in Figure A–8.

Figure A–8 Create Discounts in Oracle BRM

2. Verify the synced records in Siebel CRM, as shown in Figure A–9.

Figure A–9 Synced Records in Siebel CRM
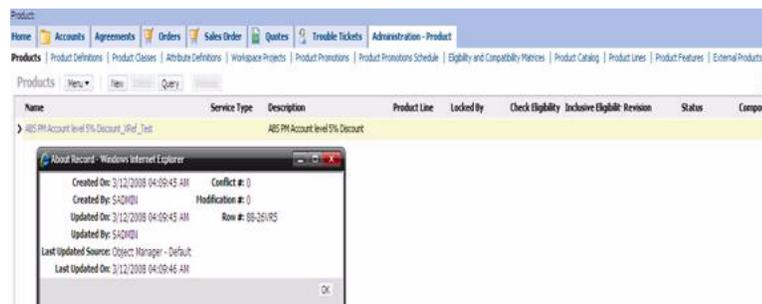
3. Verify the data entered into the XREF_DATA table is correct as shown in Figure A–10 and Figure A–11.

These tables show how data entered into the cross-reference table correspond to points 1 and 2 explained previously.

Figure A–10   Figure Corresponding to Point 1

	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE	IS_DELETED	LAST_MODIFIED	LAST_ACCESSED
1	ITEM_ID	BRM_01	2E60E990F02D11DCBFCAF1F293F06D61	0.0.0.1 /discount 600048 0	N	1205323775657	1205323775657
2	ITEM_ID	COMMON	2E60E990F02D11DCBFCAF1F293F06D61	2d313734373134383431383534303233	N	1205323775657	1205323775657

Figure A–11   Figure Corresponding to Point 2

	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE	IS_DELETED	LAST_MODIFIED	LAST_ACCESSED
1	ITEM_ID	SEBL_02	2E60E990F02D11DCBFCAF1F293F06D61	89-26YR5	N	1205323788212	1205323788212

B

OLM - Communications Orders Dictionary

This appendix provides a snapshot of the Communications Orders Dictionary at the time this appendix was created. Communications Orders include enterprise business objects (EBOs) for Sales Order, Fulfillment Order, and Provisioning Order. We refer to any of the three orders using the token <CommsOrder>.

This appendix contains the following sections:

- Section B.1, "Communications Orders - Order Header Component Attributes"
- Section B.2, "Communications Orders - Order Line Component Attributes"

To understand Table B–2 and Table B–3, you must be familiar with the terms defined in Table B–1.

Table B–1   Communications Orders Dictionary

Term	Definition
Assetable	Indicates if an attribute value is saved to the corresponding asset in Siebel CRM. An asset here refers to purchased product offering instance by a customer.
Prior Value	Indicates if, when the attribute changes, a prior value is also sent on the order message. Prior values sometimes are used to determine if a change occurred and sometimes used to roll back changes.
OM	Order Management
CRM	Siebel Customer Relationship Management

> **Caution:**   Oracle Application Integration Architecture (Oracle AIA) EBOs may present more attributes than used by one business process or application because they cross industries and are built as part of the Foundation Pack. Therefore, the listing of an attribute in a following table does not mean that the attribute is used and the corresponding feature is available. The supported features are listed in the Oracle AIA pre-built integrations and in documentation and collateral for the participating applications. Although the remarks against some attributes indicate lack of support for some attributes, they are not a complete account of uptake of these attribute across different applications.

B.1 Communications Orders - Order Header Component Attributes

Table B–2   Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: \<CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Order ID	Uniquely identifies each order.	NA	No	None	Produces a unique identifier for all orders, including revision orders. Unlike Order Number, Order ID is different for revisions of the same base order. Used by Oracle AIA for cross-reference.	SaleOrderEBO/Identification/BusinessComponentID
Order Number	Identifies an order across revisions.	NA	No	None	A revision number >1 does not necessarily mean that this is a revision order from OM Fulfillment. You can create an order in Siebel CRM and revise it several times before submitting it. If an Order Number matches an in-flight order, then the order is treated as a revision order. When an order is revised, this number stays the same. OM uses this number to identify the base order. If the same order number with the same revision is submitted, then OM rejects the revision order and places it in fallout.	\<CommsOrder>EBO/Identification/ID
Revision	A revision sequence number that, with the order number, represents the user key to an order.	NA	No	None	If an order is received with an Order Number equal to that of an in-flight order and the newly received order has a higher revision number, then OM assumes the order is a revision order and proceeds to analyze the Order Lines. If the revision number is equal or lower than that of the base order, the revision is rejected.	\<CommsOrder>EBO/Identification/Revision/Number
Success Dependency	Declares if all order lines must fulfill successfully or else the whole order fails (all or none). When the order level Success Dependency is set to All or None, it takes precedence over Order Line Success Dependency designations because it is more restrictive.	DEFAULT ALL OR NONE	No	None	$	\<CommsOrder>EBO/PartialFulfillmentAllowedIndicator

Table B–2   (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Fulfillment Mode	Qualifies the nature of fulfillment request.	Deliver, Qualify, Cancel, Initiate billing, Fulfill billing	No	None	Communications service providers (CSPs) may extend support to other modes, such as Design, Schedule and Cost. CRM can cancel an order through a revision order with no order lines or by resending the order with Fulfillment Mode = *Cancel*. OM is expected to honor the two alternatives for canceling an order, providing no order lines reaches the point of no return. When used on billing EBS, Fulfillment Mode has a different meaning. It determines the type of Billing request: *Initiate* or *Fulfill.*	<CommsOrder>EBO/FulfillmentModeCode
Customer Class	Identifies type of customer: Residential, Business, and so on	Residential Business	No	None	$	<CommsOrder>EBO/CustomerPartyReference/CustomerPartyAccountTypeCode
Organization ID	Identifies the organization/LOB generating the order. No cross-reference exists.	NA	No	None	No cross-reference. OM should use the application-specific ID if required in any of the rules; if Organization IDs are unique and synchronized across all order capture systems.	<CommsOrder>EBO/BusinessUnitReference/BusinessUnitIdentification/ID
Sales Channel	Identifies the sales channel.	NA	No	None	$	<CommsOrder>EBO/SalesChannelCode
Job ID	A string or number that uniquely identifies the job to orchestration	NA	No	None	Track orders that belong to a bulk or batch job.	<CommsOrder>EBO/ProcessingNumber
Sequence in Job	A number that identifies the order sequence within the job.	NA	No	None	$	<CommsOrder>EBO/ProcessingSequenceNumber
Job Type	Identifies the type of job. This information identifies the threshold for creating a consolidated SR for Bulk or Batch Orders. This value is optional for orders whose Job Cardinality is 1. By default, this value is HETROGENEOUS.	Heterogeneous,homogeneous, third-party homogeneous, third-party heterogeneous, correlated	No	None	$	<CommsOrder>EBO/ProcessingTypeCode

Table B–2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Job Cardinality	Indicates the total number of orders within the job.	NA	No	None	$	<CommsOrder>EBO/ProcessingQuantity
Parent Order ID	Order ID of another order that indicates the fulfillment for this order does not start before the parent order fulfillment completes.	NA	No	None	This attribute applies to explicit order-to-order dependencies and is not limited to follow-on orders. For example, in a B2B scenario, a large order can be divided into some smaller orders, with one order acting as the root order for all other orders and the remainder of the orders chained using the parent order ID attribute.	<CommsOrder>EBO/Parent<CommsOrder>Reference/<CommsOrder>Identification/BusinessComponentID
Fulfillment Priority	Indicates relevant priority of order fulfillment across orders. A lower value indicates a higher priority. Accepts values 0 to 9 in accordance with JMS Queue support.	9,7,5,3	No	None	EBM value: Siebel value 9: Urgent. Used for expedited orders. 7: High. CSP determines its use. 5: Medium. CSP determines its use. 3: Low. Recommended for job orders. Notice that Oracle Advanced Queuing (AQ) and JSM priority values have the inverse order of precedence	<CommsOrder>EBO/FulfillmentPriorityCode
Order Type	Sometimes indirectly determines sales channel to drive compensation process.	Sales Order	No	None	$	<CommsOrder>EBO/TypeCode
Requested Delivery Date Time	Overall order level due date that provides the default due date at each line level. Can be overridden at each line.	NA	Yes	None	$	<CommsOrder>EBO/RequestedDeliveryDateTime
Fulfillment Status	Reports aggregate order fulfillment status.	In Progress, Failed, Canceled, Complete	Yes	None	This is different from the Status attribute tracked within Siebel CRM.	<CommsOrder>EBO/Status/Code

Table B–2   (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: \<CommsOrder\> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Status Context	Provides details about the current status. The implementer configures this value.	NA	Yes	None	OM can use this to track the milestone causing the status change, along with context details such as *error message, cause for cancel*. One primary scenario that the Order Header / Status Context is populated: with revision orders that cancels Order Lines by dropping them from the revision and if the revision is rejected. In that case the orchestration system does not have a line on the revision order to provide fallout status and context. In such a case the header level status context is used to identify the base line the cause for the fallout.	\<CommsOrder\>EBO/Status /Description
Owner Account ID	Identifies the owner account.	NA	Yes	None	Cross-referenced.	\<CommsOrder\>EBO/Customer PartyReference/Customer PartyAccountIdentificatio n/BusinessComponentID
Owner Account Name	Identifies the Account Name. You can enter or derive this value from contact first name + last name of primary contact associated with the account.	NA	Yes	None	Required for network inventory tracking of service owner.	\<CommsOrder\>EBO/Customer PartyReference/Customer PartyAccountName
Owner Account Number	Identifies account number to customer.	NA	Yes	None	$	\<CommsOrder\>EBO/Customer PartyReference/Customer PartyAccountIdentificatio n/ID
Account Contact ID	Foreign key to contact record that holds personal and contact details of the customer/company representative who is placing the order and is the contact person for anything related to the order process.	NA	Yes	None	$	\<CommsOrder\>EBO/Customer PartyReference/Customer PartyAccountContactIden tification/BusinessCompone ntID
Account Contact Address (component)	Identifies the address used to communicate with the Contact ID.	NA	Yes	None		\<CommsOrder\>EBO/Customer PartyReference/Customer PartyAccountContactAdd ressCommunication/Addre ssCommunication/Address

Table B–2   (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset- able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: \<CommsOrder\> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Project ID	Identifies project record if the order to be delivered is part of a project that contains related orders. Foreign key reference. No cross-reference.	NA	Yes	None	No cross-reference for 2.4.	\<CommsOrder\>EBO/ProjectReference/ProjectIdentfication/ID
Fulfillment System Type	For the Get Target Fulfillment Provider utility service, determines the logical identifier for appropriate target system instance among those serving this Fulfillment System Type	NA	No	None	$	FulfillmentOrderEBO/FulfilmentSystemTypeCode
Target Instance	For the Get Target Fulfillment Provider utility service returns the logical identifier for appropriate target system instance among those serving this Fulfillment System Type.	NA	No	None	$	FulfillmentOrderEBO/FulfilmentTargetSystemID

Table B–2   (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows: \<CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Order Changed Indicator	OM sets this attribute to Yes if the order changed significantly such that CRM should make a copy of the customer order to preserve the customer intent before updating the working version of the order.	True, False	No	None	Allows Siebel to make a copy of the order if the order changes to the extent that the customer's intent is compromised.	\<CommsOrder>EBO/Order ChangedIndicator
Sales Representative ID	CRM User ID that identifies the sales representative who entered the order.	NA	No	None	No cross-reference. Use the application ID.	\<CommsOrder>EBO/Sales personPartyReference/Party Identification/ID
Owner Account Contact (multiple fields)	Identifies if the address is used to communicate with the contact ID. Includes these fields: First Name, Last Name, Phone Number, and Email.					\<CommsOrder>EBO/Custo merPartyReference/Custom erPartyAccountContact/Firs tName
						\<CommsOrder>EBO/Custo merPartyReference/Custom erPartyAccountContact/Las tName
						\<CommsOrder>EBO/Custo merPartyReference/Custom erPartyAccountContactPhon eCommunication/PhoneCo mmunication/CompleteNu mber
						\<CommsOrder>EBO/Custo merPartyReference/Custom erPartyAccountContactEmai lCommunication/EmailCo mmunication/

B.2 Communications Orders - Order Line Component Attributes

Table B–3 Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Line ID	Uniquely identifies the order line item across orders and order revisions. Automatically generated.	NA	No	None	Cross-referenced. Produces a unique identifier for all Order Lines, including revision Order Lines.	<CommsOrder>EBO/<CommsOrder>Line/Identification/BusinessComponentID
Base Line ID	References base order line revised by this order line	NA	No	None	Uses a cross-reference.	<CommsOrder>EBO/<CommsOrder>Line/Original<CommsOrder>LineReference/<CommsOrder>LineIdentification/BusinessComponentID
Asset Integration ID	Uniquely identifies an instance of a product that was or is being purchased.	NA	Yes	AIA2.0	Cross-referenced CRM populates the Asset Integration ID on all Order Lines, regardless of the Assetable state on the subject of the Order Line or whether the Order Line is for a new or existing service. A revision should never change the Asset Integration ID. When a product is dropped as part of one product hierarchy (CP or Promotion) and then added through another product hierarchy (CP or Promotion), the Asset Integration ID for the two line items are different, although for the same product.	<CommsOrder>Line/InstalledProductReference/InstalledProductIdentification/BusinessComponentID
Line Number	Identifies the line regarding its position in the line item tree.	NA	No	None	Line number establishes the parent child relationship between Order Lines of the same order, but it may vary across revisions. Therefore, do not rely on it for matching Order Lines across revisions.	<CommsOrder>EBO/<CommsOrder>Line/Identification/ID
Parent Line	References parent order line in the line items tree instantiated according to the product model definition. Points to itself if the item does not have an associated parent item.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/Parent<CommsOrder>LineIdentification/BusinessComponentID

Table B–3 (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Root Line	References the root order line in the line item tree instantiated according to the product model definition. Points to itself if the item is a root item itself.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/RootParent<CommsOrder>LineIdentification/BusinessComponentID
Related Line ID	BRM adaptors use to relate one-time charges to base line ID.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ChargeParentLineIdentification/BusinessComponentID
Related Asset Integration ID	Links Move-Add to Move-Delete line items	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/InstalledProductReference/PriorInstalledProductIdentification/BusinessComponentID
Depends On Line ID	Indicates order line item ID of a previous order line item that is changed by this order. Follow-on orders use this value to capture dependencies of the order line items in the follow-on order-to-order line items of original orders.	NA	No	None	Cross-referenced.	<CommsOrder>EBO/<CommsOrder>Line/Depending<CommsOrder>LineReference/<CommsOrder>LineIdentification/BusinessComponentID
Depends On Order ID	Identifies order ID of an in-flight order, which is the basis for this follow-on order line item.	NA	No	None	Cross-referenced.	<CommsOrder>EBO/<CommsOrder>Line/Depending<CommsOrder>Reference/<CommsOrder>Identification/BusinessComponentID
Promotion Line ID	References an order line that represents the promotion/marketing offer under which the order line is being purchased.	NA	No	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/Promotion<CommsOrder>LineReference/Promotion<CommsOrder>LineIdentification/Identification/BusinessComponentID
Promotion Asset Integration ID	References an asset that represents the promotion/marketing offer under which the order line is being purchased.	NA	Yes	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/Promotion<CommsOrder>LineReference/InstalledProductReference/InstalledProductIdentification/BusinessComponentID
Product ID	References product record based on which order line is instantiated. Foreign key reference.	NA	Yes	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/Identification/BusinessComponentID

Table B–3 (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Quantity	Identifies the quantity of the item requested by a customer. Default is 1.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/OrderQuantity
Action Code	Specify action required to meet customer request	None, Add, Update,Suspend Resume, Delete,Move-Add , Move-Delete	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ServiceActionCode
Deliver To Address	Address record that represents the delivery/service installation address.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/ServiceAddress/Address
Requested Delivery Date Time	When Null, the requested date for delivery of the goods or service is ASAP; otherwise, it is the specified date. This date is not guaranteed. Typically, it is a future date; if it is a past date, then the default behavior equals a Null value.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/RequestedDeliveryDateTime
Usage Start Date	Determines the date when usage events should start being rated. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to the current date.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ServiceUsageStartDate
Cycle State Date	Determines the date when cycle charges should start being billed. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to the current date according to previous patterns.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/CycleStartDate

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Purchase Date	Determines the date when one-time purchase charges should be billed. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to current date according to previously mentioned patterns.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/PurchaseDate
Service Start Date	Indicates effective start date of service.	NA	Yes	None	Initially computed by Siebel based on Due Date and then updated by Order Management based on Actual Delivery Date	<CommsOrder>EBO/<CommsOrder>Line/EffectiveTimePeriod/StartDateTime
Earliest Delivery Date	Identifies the date when the work associated to the order can start. Typically used for fulfillment actions that require customer presence such as in cases customer must be available to install service or deliver shipment	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/EarliestDeliveryDateTime
Service End Date	Indicates the effective end date of service. Applies to services with a specified duration.	NA	Yes	None	Initially computed in Siebel and then updated by Order Management. Update is sent to Siebel.	<CommsOrder>EBO/<CommsOrder>Line/EffectiveTimePeriod/EndDateTime
Actual Delivery Date Time	Determines the date when the purchased product or service is considered available to the customer by the CSP. This date may be when physical goods are shipped, delivered, or their receipt is acknowledged. For service-based products, the service is activated on this date. This date is computed in the OM Fulfillment flow according to previous patterns.	NA	Yes	None	Oracle BRM does not allow for starting any charges before the Purchase Date; therefore, the ABCS for Oracle BRM always overrides the Purchase Date if it is later than any of the Cycle or Usage start dates. OM should facilitate calculation of Order Line level Actual Delivery Date and Order Line attributes for billing Usage Start Date, Cycle Start Date, and Purchase Date.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ActualDeliveryDateTime

Table B–3 (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Expected Delivery Date Time	Indicates the due date expected by the system because of Design and Assign. The default is the Order Due Date when the order is created by CRM.	NA	No	None	Computed by OM based on preconfigured time estimates on fulfillment actions. Used by OM to communicate to CRM changes to expected delivery date of specific Order Lines.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ExpectedDeliveryDateTime
Fulfillment Status	Updates orchestration and CRM regarding the current status of order line fulfillment at a high level.	In Progress, Failed,Canceled Complete	Yes	None	Additional values can be added as an extension	<CommsOrder>EBO/<CommsOrder>Line/Status/Code
Milestone	Fulfillment passes the last reached milestone into this field.	NA	No	None	$	<CommsOrder>/<CommsOrder>Line/MilestoneCode
Status Context	Provides details about the current status of the order line. The implementer configures this value.	NA	Yes	None	OM can include the reached milestone (from the fulfillment system, the cause for the status update that is necessary because of dynamic nature of fulfillment plan) and a textual string for context per current status as follows (canonical Status / status context): Submitted / NA In Progress / <milestone>: context text Failed / <milestone>: reason text Canceled / <milestone>: reason text Complete / NA In Progress: Context Text could be used to indicate any of the following among others: o Requires customer interaction o Delivery is expected to be delayed	<CommsOrder>EBO/<CommsOrder>Line/Status/Description
Point-of-no-return	Determines if Siebel should allow order line revisions to be submitted.	Not yet, Hard	No	None	OM Fulfillment flows allow configuration of setting a hard PONR when a condition is met for a particular service. When a hard PONR is reached for an Order Line in OM, a status update is issued to reflect the same in CRM. Additional values such as SOFT can be added as an extension.	<CommsOrder>EBO/<CommsOrder>Line/RevisionPermissibleCode

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Billing Account	References an account record that represents the bill payer or the branch of a company responsible for bill payment. This value may be a customer account or an account from the account hierarchy.	NA	Yes	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/BillToPartyReference/CustomerPartyAccountIdentification/BusinessComponentID
Billing Profile	References the billing profile record that holds the customer's billing/payment preferences. This value may be associated to the customer account or to a separate billing account.	NA	Yes	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/BillToPartyReference/BillingProfileReference/BillingProfileIdentification/BusinessComponentID
Payment Profile	Identifies the Payment Profile.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/BillToPartyReference/BillingProfileReference/PaymentProfileReference/PaymentProfileIdentification/BusinessComponentID
Service Account	References an account record that represents a service user or the branch of the company where service is installed. This value may be customer account or an account from the account hierarchy.	NA	Yes	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/OwnerPartyReference/CustomerPartyAccountIdentification/BusinessComponentID
Owner Contact	Represents a contact of the customer account or service account who should be contacted during fulfillment of the line if required.	NA	Yes	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/OwnerPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID
Shipping Contact	Represents a contact of the customer account or service account who should be contacted for shipping purposes.	NA	Yes	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ShipToPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID

Table B–3 (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Node	Alphanumerically references the root order line that corresponds to access at site A of a connection. This value is relevant for network ordering only.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="Node"]/ValueText
To Node	Alphanumerically references the root order line that corresponds to access at site B of a connection. This value is relevant for network ordering only.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToNode"]/ValueText
Network ID	Unique compound product number that represents the virtual network ID. Relevant for network orders. Provided by default from the order number and cascaded to network connection items.	NA	Yes	AIA2.4	Identifies which Access and Nodes belong to the same network. This information may be of value to decomposition.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="NetworkID"]/ValueText
Port Number	Identifies the port number allocated to the access circuit connected to provide (starting) edge router during the fulfillment process.	NA	Yes	AIA2.4	For new services, port number comes back from Network Inventory through provisioning.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="PortNumber"]/ValueText
To Port Number	Identifies the port number allocated to the access circuit connected to provide (ending) edge router during the fulfillment process.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToPortNumber"]/ValueText
Service Address Prefix	Identifies the area code/NPA for the access circuits on starting or two ends of the connection.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ServiceAddressPrefix"]/ValueText

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
To Service Address Prefix	Identifies the area code/NPA for the access circuits on the end of the connection.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressPrefix"]/ValueText
Access Circuit	Provides the Common Language Location Identification (CLLI) for the access circuit on two sides or starting side of the connection.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="AccessCircuit"]/ValueText
To Access Circuit	Provides the CLLI for the access circuit on ending side of the connection.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToAccessCircuit"]/ValueText
To Service Account ID	Identifies the Service Account ID associated with the end side of a network.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAccountID"]/ValueText
From Service Address ID	Identifies the Service Address ID for the starting point of a network.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="FromServiceAddressID"]/ValueText
To Service Address ID	Identifies the Service Address ID for the ending point of a network.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressID"]/ValueText
To Service Point ID	References a dummy asset record that represents the access point to which the starting side of a network service is connected on the customer's premises.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServicePointID"]/ValueText

Table B–3 (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Service Point	References a dummy asset record that represents the access point to which this service is connected on the customer's premises. For example, NTE for PSTN, Set top box for Broadband/Cable service.	NA	Yes	AIA2.4	Expected to be mastered in network inventory and loaded in Siebel in batch.	<CommsOrder>EBO/<CommsOrder>Line/ServicePointCode
Promotion Description	Provides short description that appears on the invoice.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/Description This is Promotion Description used for display purposes on customer invoice
Service ID	Identifies the product/service instance as recognized across BSS and OSS applications. Most significantly this is the ID used to correlate rating records to customer accounts.	NA	Yes	AIA2.0	Can be populated as part of order capture process or during fulfillment, but before interface an order to billing.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ItemInstance/Identification/ID
Balance Bundle Identification	Identifies the Balance Bundle to which a service instance belongs.	NA	$	$	Not Used by Oracle AIA for Communications	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/BalanceBundleIdentification/BusinessComponentID
Line Description	Provides additional description for an order line. For example, to indicate that a charge is being applied for a penalty.	NA	No	None	Not used by Oracle AIA for Communications	<CommsOrder>EBO/<CommsOrder>Line/Description
Service Length	Indicates requested service length in Service Length Unit of Measure	NA	Yes	Yes	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ServiceTimePeriod/Duration
Service Length Unit of Measure	Indicates the service length unit of measure.	NA	Yes	Yes	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/ServiceTimePeriod/Duration

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Fulfillment Mode	Designates compensation operations for Initiate Billing. May be used in the future to provide explicit revision operations at the line level.	DO, NOOP,REDO,UNDO	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/FulfillmentModeCode
Product Name	Provides the name of the product.	NA	$	$	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/Name
Composition Type	Determines product composition granularity. PartialItem is an order line that constitutes an indivisible element of another order line. This type typically denotes a piece of a product. WholeItem is an order line that represents a self-contained subject. A WholeItem may be represented by a single line item or some PartialItem order lines. May also assume no value signified by a Null value or absence of value.	<no value> for NULL, PARTIAL ITEM, WHOLE ITEM	No	None	Consult Oracle on usage.	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/FulfillmentCompositionTypeCode
Product Type	Classifies products into Products, Discounts, Bundles, Promotion (Offer), and so on.	Product, Offer, Bundle	No	None	Used part of fulfillment to determine the order lines Subject Type, which drives the mapping to Product Specifications.	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/TypeCode
Billing Type	Classifies products for Billing into Service Bundles, Subscriptions, Items, Discounts, and Special Ratings.	Service Bundle, Subscription, Item, Discount, Special Rating	No	None	Used with Product Type.	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/ClassificationCode [listID="BillingProductTypeCode"]
Billing Service Type	Specifies the service type so that when a corresponding product is created in Billing, it is associated to the specified service.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/ClassificationCode [listID="PermittedTypeCode"]

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Service Flag	Indicates the product of a service or non-service, for example, physical goods.	TRUE, FALSE	No	None	Used with Product Type and may be used to parameterize fulfillment flows.	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/ServiceIndicator
Vendor	Identifies the vendor supplying the product when the product is supplied by a third-party.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/SupplierPartyReference/PartyIdentification/ID
Vendor Part Number	Identifies the product part number to the vendor.	NA	Yes	AIA2.4	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/ItemIdentification/SupplierItemID
Fulfillment Item Code	Uniquely identifies the mapping of an Order Line Subject to a Product Specification.	1) Null 2) A unique code that identifies the Product Spec to OM	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/ClassificationCode [listID = "FulfillmentItemCode"]
Item Class Name	Determines business classification of a product.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/PrimaryClassificationCode
Success Dependency	Declares if all order lines of a bundle or offer must fulfill successfully or else the whole bundle or offer fails (all or none).	Default, All Or None	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/FulfillmentSuccessCode
Start Billing on First Usage	When set to Yes by CRM or OSM, passes the request along to BRM. In this case, Usage Start Date, Cycle Start Date, and Purchase Date should have no effect.	True, False	No	None	Not yet supported by integration.	<CommsOrder>EBO/<CommsOrder>Line/StartBillingOnFirstServiceUsageIndicator. We have added BillingStartCode to ItemReference, if this requirement is at the item/itemReference level and not line level then BillingStartCode from ItemReference is necessary.
Smart Part Number	Automatically generated based on a predefined scheme. Mainly, drives dynamic product configuration/pricing rules in CRM. The billing system may use it to dynamically derive a price/discount value.	NA	Yes	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/AlternateObjectKey [ContextID=SmartPartNumber]

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Network Product Flag	Indicates if this is a network product, which helps determine which user-defined attributes to expect.	True, False	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/NetworkIndicator
Network Element Type	Indicates if this network product represents a node, a connection, or a network.	NA	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/ItemReference/NetworkItemTypeCode
Charge Frequency Code	Indicates charge frequency unit of measure, for example, monthly, quarterly, yearly.	NA	$	$	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/ChargeFrequencyCode
List Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/TypeCode
List Price	Identifies base price of the item.	NA	Yes	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/UnitListPrice/Amount
Sale Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/TypeCode
Sale Price	Identifies net price of the item.	NA	Yes	AIA2.0	$	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/UnitSalePrice/Amount
Pricing Commit Type	Indicates whether the pricing is Committed or Dynamic.	Common/Siebel values are true/Dynamic, false/Committed.	Yes	AIA2.4	$	<CommsOrder>/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/DynamicPricingIndicator
Dynamic Discount Method	Indicates whether the discount is of type amount or percent.	Amount, Percent	Yes	AIA2.4	$	<CommsOrder>/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/DiscountMethodCode
Discount Percent	Indicates the percent by which the list price is discounted.	NA	Yes	AIA2.4	$	<CommsOrder>/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/DiscountPercent

Table B–3   (Cont.) Communications Orders - Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath. Depends on context as follows <CommsOrder> variable for SalesOrder, FulfillmentOrder, or ProvisioningOrder
Discount Amount	Indicates the amount by which the list price is discounted	NA	Yes	AIA2.4	$	<CommsOrder>/<CommsOrder>Line/<CommsOrder>Schedule/<CommsOrder>ScheduleCharge/Charge/DiscountAmount
Member [0.N]	Represents a member of a list by their phone number.	NA	No	None	Used for capturing membership to friends and family plans.	<CommsOrder>EBO/<CommsOrder>Line/<CommsOrder>LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="SpecialRating"]/ValueText [0.N]
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA Name	<CommsOrder>/<CommsOrder>Line/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification/Name
User Defined Attributes	Indicates attribute is common across all Specification components.	Add, Update, Delete	Yes	None	UDA Action Code (Expected to change to a Service Action Code element to allow additional value NONE.)	<CommsOrder>/<CommsOrder>Line/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/@actionCode
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	has Previous LIC Value	UDA language-independent code Value	<CommsOrder>/<CommsOrder>Line/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/Value
User Defined Attributes	Indicates attribute is common across all Specification components.	String, Date, Number	Yes	None	UDA Data Type	<CommsOrder>/Prior<CommsOrder>/<CommsOrder>Line/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute"]/DataTypeCode
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA language-independent code Prior Value	<CommsOrder>/Prior<CommsOrder>/<CommsOrder>Line/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/Value

C

OLM - Mapping Billing Dates

This appendix provides information about how dates are set in Oracle Billing and Revenue Management (Oracle BRM) as part of the Bill Fulfillment Order flow.

This appendix contains the following section:

- Section C.1, "How Dates are Set in Oracle BRM"

C.1 How Dates are Set in Oracle BRM

Table C–1 provides information about how dates are set in Oracle BRM. These terms and abbreviations are used in the table:

- **ODT: Order Datetime:**

  The date that the order was placed by the customer and is captured on the order in the order capture system (Siebel CRM). Siebel CRM defaults this date, but it can be changed by the user.

- **RDDT: Requested Delivery Datetime:**

  The delivery date requested by the customer; it is captured on the order in the order capture system (Siebel CRM). It is also known as *Due Date*.

- **ADDT: Actual Delivery Datetime:**

  The actual delivery date time; it is supplied by the order management system that fulfills the order, and is updated in the order capture system (Siebel CRM).

- **Purchase Start Date:**

  The date as of which Oracle BRM applies purchase fees.

- **Cycle Start Date:**

  The date as of which Oracle BRM applies cycle fees.

- **Usage Start Date:**

  The date as of which Oracle BRM rates usage and applies usage fees.

Table C–1   Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service
Customer data creation	Oracle AIA uses order date as the effective date for customer data creation	PCM_OP_CUST_COMMIT_CUSTOMER	Pass Order Date coming from Siebel CRM.
For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	For the CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service
Single Phase Billing - Billing Fulfillment Promotion Purchase	Oracle AIA passes the Purchase Date as the Valid From date for bundle purchase (that represents purchased promotion). If Purchase Date is null, then it passes Requested Delivery Date and if that is null, it passes no date and Oracle BRM defaults current date.	PCM_OP_SUBSCRIPTION_SET_BUNDLE	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Purchase Date to Actual Delivery Datetime.
Single Phase Billing - Billing Fulfillment	If all three of the billing dates are set, then Oracle AIA uses Order Date as Effective Date, and sets respective offset (Order Date - respective billing date). Billing dates are: Purchase Date, Cycle Start Date and Usage Start Date. If any of the three billing dates are not set, then Oracle AIA passes no dates to Oracle BRM and lets Oracle BRM default the Purchase, Cycle Start and Usage Start dates. For purchase of a service bundle, this check for existence of billing dates applies to ALL products and discounts included in the service bundle.	PCM_OP_CUST_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Purchase Date, Start Cycle, and Start Usage to Actual Delivery Datetime to explicitly control setting of billing dates.
Single Phase Billing - Billing Fulfillment. Time Based Account or Service level Subscription Product/Discount Purchase	In addition to setting of billing dates as described previously, if Service End Date is passed, then Oracle AIA additionally sets the Purchase, Cycle and Usage end date offsets (difference between the respective billing date and service end date). If any of the billing dates (Purchase, Cycle, or Usage start) are not set then Oracle AIA uses the Order Date to calculate the Purchase, Cycle and Usage end date offsets (difference between the Order Date and Service End Date).	PCM_OP_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Populate Purchase, Cycle and Usage Start dates (this is required for enabling time-based offerings (TBO). Calculate the Service End Date based on TBO attributes as documented in TBO section. For more information about TBO attributes, see Section 12.6, "Supporting Time-Based Offerings."

Table C–1 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
Single Phase Billing - Billing Fulfillment. Time Based Account or service-level Subscription Product/Discount Update (of end date due to promotion upgrade or downgrade, or other pricing changes).	If Service End Date is passed (and prior value is set), then Oracle AIA uses that to reset the Purchase, Cycle and Usage end dates.	PCM_OP_ SUBSCRIPTION_SET_ PRODINFO PCM_OP_ SUBSCRIPTION_SET_ DISCOUNTINFO	Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate prior value to trigger update. For more information about TBO attributes, see Section 12.6, "Supporting Time-Based Offerings."
Single Phase Billing - Billing Fulfillment. Promotion Cancellation	If ADDT is passed, Oracle AIA uses that to set the VALID_TO date in Oracle BRM for the bundle. If ADDT is not passed then Oracle AIA uses the Requested Delivery Datetime. If Requested Delivery Datetime is not passed then Oracle AIA does not set the VALID_TO date.	PCM_OP_ SUBSCRIPTION_SET_ BUNDLE	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Actual Delivery Datetime
Single Phase Billing - Billing Fulfillment Application of Promotion Penalties or MACD One Time Charge (Suspend, Resume, Disconnect, or Move charge) Note - These are processed only in Billing Fulfillment.	If ADDT (on service bundle line) is passed, Oracle AIA sets the effective date to ADDT (from service bundle line). If ADDT (on service bundle line) is not passed, then Oracle AIA lets Oracle BRM default the purchase date (to current date).	PCM_OP_ SUBSCRIPTION_ PURCHASE_DEAL	Set Actual Delivery Datetime
Single Phase Billing - Billing Fulfillment. Suspend, Resume, or Cancellation of Service or account-level or service-level Subscription Product/Discount.	If ADDT is passed, then Oracle AIA uses that as the effective date for the operation, else it lets BRM default the date (to current date)	PCM_OP_ SUBSCRIPTION_SET_ PRODUCT_STATUS PCM_OP_ SUBSCRIPTION_SET_ DISCOUNT_STATUS PCM_OP_CUST_SET_ STATUS	Set Actual Delivery DateTime.
Two-Phase Billing - Billing Initiation. Promotion Purchase.	Oracle AIA passes Purchase Date as the Valid From date. If Purchase Date is null, then Oracle AIA passes Requested Delivery Date and if that is null, Oracle AIA passes no date and Oracle BRM defaults current date	PCM_OP_ SUBSCRIPTION_SET_ BUNDLE	Pass Order Date and Requested Delivery Date coming in from Siebel CRM. Set Purchase Date to Expected Delivery Date.
Two Phase Billing - Billing Initiation. Account-level or service-level Item Type Product Purchase.	Oracle AIA validates that Purchase Date is set to future (based on value of configuration property - FutureTimeThreshold). Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date). Billing Dates are - Purchase Date, Cycle Start Date and Usage Start Date.	PCM_OP_CUST_ MODIFY_CUSTOMER PCM_OP_ SUBSCRIPTION_ PURCHASE_DEAL	Pass Order Date coming in from Siebel CRM. Set Purchase, Cycle, and Usage Date to Future (one year out to match default threshold).

Table C–1   (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
Two Phase Billing - Billing Initiation. Account-level or service-level Subscription Type Product/Discount Purchase.	Oracle AIA validates that Cycle Start Date is set to future (based on value of configuration property - FutureTimeThresholdForBilling Dates). Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date). Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date	PCM_OP_CUST_ MODIFY_CUSTOMER PCM_OP_ SUBSCRIPTION_ PURCHASE_DEAL	Pass Order Date coming in from Siebel CRM. To support validation mode, set all three billing dates to the future (one year out to match default threshold). To support latency mode, set Purchase and Usage Start Date to Current, but set Cycle Start Date to Future (one year out to match threshold).
Two-Phase Billing - Billing Fulfillment. Promotion Purchase.	Oracle AIA uses purchase date to reset Valid From date.	PCM_OP_ SUBSCRIPTION_SET_ BUNDLE	If purchase date had been set to Expected Delivery Date in Billing Initiation, reset purchase date to Actual Delivery Date
Two Phase Billing - Billing Fulfillment. Account-level or service-level Item Type Product Purchase.	If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset. Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.	PCM_OP_ SUBSCRIPTION_SET_ PRODINFO	Reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update).
Two Phase Billing - Billing Fulfillment. Account-level or service-level Subscription Type Product/Discount Purchase.	If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset. Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.	PCM_OP_ SUBSCRIPTION_SET_ PRODINFO PCM_OP_ SUBSCRIPTION_SET_ DISCOUNTINFO	To support validation mode, reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update). To support latency mode, reset Cycle date to Actual Delivery Datetime (set prior value to trigger update).
Two Phase Billing - Billing Fulfillment. Time-Based account-level or service-level Subscription Product/Discount Purchase.	If Service End Date is passed, then Oracle AIA uses that to set the Purchase, Cycle, and Usage end dates for products/discounts purchased.	PCM_OP_ SUBSCRIPTION_SET_ PRODINFO PCM_OP_ SUBSCRIPTION_SET_ DISCOUNTINFO	Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate Purchase, Cycle, and Usage start dates. For more information about TBO attributes, see Section 12.6, "Supporting Time-Based Offerings."

D

OLM Bill Fulfillment Order - Matrix of MACD Actions Supported Per Billing Product Type

This appendix provides information about the Move, Add, Change, Disconnect (MACD) line actions that are supported by the order lifecycle management (OLM) Bill Fulfillment Order process integration for a given product type. Also discussed is how OLM Bill Fulfillment Order communicates changes to billing for attributes on change orders (action of UPDATE (for service bundle and its components)) for a given product type.

This appendix contains the following sections:

- Section D.1, "Table A"
- Section D.2, "Table B"

D.1 Table A

OLM Bill Fulfillment Order process integration supports the MACD line actions listed in Table D–1 for a given product type.

Table D–1   Table A

Product Type	Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
--	--	--	--	--	--	See Table B for a list of attributes	--
Marketing Bundle ('Promotion' in Siebel, no representation in Oracle Communications Billing and Revenue Management (Oracle BRM)	Yes	Yes	Not Applicable. Does not affect purchased bundle in Oracle BRM.	Not Applicable. Does not affect purchased bundle in Oracle BRM.	Yes	Xref updated to reflect new Siebel Customer Relationship Management (Siebel CRM) asset.	Ignored other than to determine original Oracle BRM asset.
Service Bundle	Yes	Yes	Yes	Yes	Yes	Yes. Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original Oracle BRM asset.

Table D–1 (Cont.) Table A

Product Type	Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Service Bundle Component - Billing Subscription product (applies to service)	Yes. Can communicate price/discount override as part of this action.	Yes	Not supported by either Siebel or Oracle BRM, hence ignored by Billing Integration.	Not supported by either Siebel or Oracle BRM, hence ignored by Billing Integration.	Yes	Yes. Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original Oracle BRM asset.
Service Bundle Component - Billing Discount product (applies to service)	Yes	Yes	Not supported by either Siebel or Oracle BRM, hence ignored by Billing Integration.	Not supported by either Siebel or Oracle BRM, hence ignored by Billing Integration.	Yes. Same as UPDATE with communicating changes to line attributes.	Yes	Ignored other than to determine original Oracle BRM asset.
Service Bundle Component - Billing Item product (applies to service). Example: One-Time charge	Yes. Can communicate price/discount override as part of this action.	Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.	*Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.
Billing Subscription product (applies to account)	Yes. Can communicate price/discount override as part of this action.	Yes	Yes	Yes	Yes	Ignored	Ignored
Billing Discount (applies to account)	Yes	Yes	Yes	Yes	Yes	Ignored	Ignored
Billing Item product (applies to an account). Example: Penalty charge.	Yes. Can communicate price/discount override as part of this action.	Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.	Not Applicable, because no asset or purchased product instance is created.	*Not Applicable, because no asset or purchased product instance is created.	Ignored	Ignored

Note: * If a line is billing-initiated and a revision is processed for a service-level product of type *Item*, then pricing information and billing dates can change. If a line is billing-initiated and a revision is processed for an account-level product of type *Item*, then Billing Account, Bill Profile, Promotion Reference, Pricing Information, and Billing Dates can change.

D.2 Table B

OLM Bill Fulfillment Order process integration communicates changes to billing for the attributes on change orders (action of UPDATE (for service bundle and its components)) for a given product type, as shown in Table D–2.

> **Note:** Attributes may be updated on a revision and a change order, unless comments indicate otherwise.

Table D–2   Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Marketing Bundle (Promotion in Siebel, no representation in Oracle BRM)	NA	Yes For Billing Account, see note 4	NA See note 1	NA	NA	NA	Yes See note 3	The billing interface creates purchased bundle instances under billing accounts in Oracle BRM based on promotion lines. The purchase date on promotion lines is used as the start effective date for the bundle instance. When a billing account on a promotion line is updated to a different one, the purchased bundle instance is repointed to the new billing account. Billing Profile is irrelevant for promotions. Updates to a billing account on a revision or a change order result in the bundle instance being repointed to the new billing account. Updates to a purchase date on a revision result in the start effective date on the bundle instance being reset
Service Bundle	NA See note 2	Yes	NA	NA	Yes	NA	NA	No pricing on service bundle itself. Pricing is on service bundle components. A service can be transferred from one account to another provided it is the only service in a balance group and that balance group is not an account-level default balance group. Because this release supports only account-level balance groups, service transfers fail.

Table D–2   (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Service Bundle Component - Billing Subscription product (applies to service)	NA	NA	Yes	Yes	NA	NA	Yes	When subscription product is bundled in a service bundle.
								Service Acct, Billing Acct/Billing Profile, & Service Identifier - Integration looks only at service bundle line for these attributes.
								Promotion reference changes is communicated to billing. The purchased product instance is repointed to the new bundle instance in billing.
								Billing Dates - Cannot be reset using change orders.
								Cycle Start and Usage Start dates can be reset using revisions on billing initiation, if the dates that were previously set are not current.
								In two-phase billing, when Cycle Start and Usage Start dates have been set using billing initiation, they can be reset using billing fulfillment if the dates that were previously set are not current.
								End dates can be updated by change orders (promotion upgrade/downgrades) that change duration for products/discounts.

Table D–2   (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Service Bundle Component - Billing Discount product (applies to service)	NA	NA	NA	Yes	NA	NA	Yes	When discount is bundled in a service bundle.
								Discount products are not priced.
								Service Acct, Billing Acct/Billing Profile and Service Identifier - Integration looks only at Service bundle line for these attributes.
								Promotion reference changing is communicated to billing. The purchased discount instance is repointed to the new bundle instance in billing.
								Billing Dates cannot be reset using change orders.
								Cycle Start and Usage Start dates can be reset using revisions on billing initiation if the dates that were previously set are not current.
								In two-phase billing, when cycle start and usage start dates have been set using billing initiation, they can be reset using billing fulfillment if the dates that were previously set are not current.
								End dates can be updated by change orders (promotion upgrade and downgrades) that change duration for products and discounts.
Service Bundle Component - Billing Item product (applies to service).	NA	NA	Yes	NA	NA	NA	Yes	Example: Onetime Charge.
								No purchased product instance, hence no change orders.
								Pricing information, Promotion, and Quantity can be updated only on new purchase revisions.
								Billing dates or end date - Cannot be reset using change orders.
								In two-phase billing, when billing dates have been set using billing initiation, they can be reset using billing fulfillment if the dates that were previously set are not current.

Table D–2   (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Service Bundle Component - Special Rating product (applies to service).	NA	NA	NA	NA	NA	Yes	NA	On a change order, change in list reference results in list values being updated to new values from new F&F list. As delivered, the integration does not check for changes to the Special Rating List reference on revision orders when the list product has been billing-initiated. The assumption is that for wireless services (that F&F is targeted toward) no fulfillment latency exists between provisioning and billing and therefore they do not undergo two-phase billing. For more information, see Section 3.3.12, "Supporting Friends and Family."

Table D–2 (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Billing Subscription product (applies to account)	NA See comments	Yes	Yes	Yes	NA	NA	Yes	Subscription product that is not bundled into a service bundle. Service Acct - Oracle BRM does not support transferring account-level product from one account to another. Siebel CRM disallows this change. Promotion reference changing is communicated to billing. The purchased product instance is repointed to the new bundle instance in billing. Billing Dates - Cannot be reset using change orders. Cycle Start and Usage Start dates can be reset using revisions on billing initiation if the dates that were previously set are not current. In two-phase billing, when Cycle Start and Usage Start dates have been set using billing initiation, they can be reset using billing fulfillment if the dates that were previously set are not current. End dates can be updated by change orders (promotion upgrade and downgrades) that change duration for products and discounts.

Table D–2   (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Billing Discount (applies to an account)	NA See Comments	Yes	NA	Yes	NA	NA	Yes	When discount is not bundled in a service bundle.
								Discount products are not priced.
								Service Acct - Oracle BRM does not support transferring account-level products from one account to another.
								Siebel CRM disallows this change.
								Promotion reference changing is communicated to billing. The purchased discount instance is repointed to the new bundle instance in billing.
								Billing dates - Cannot be reset using change orders.
								Cycle Start and Usage Start dates can be reset using revisions on billing initiation if the dates that were previously set are not current.
								In two-phase billing, when Cycle Start and Usage Start dates have been set using billing initiation, they can be reset using billing fulfillment if the dates that were previously set are not current.
								End dates can be updated by change orders (promotion upgrade/downgrades) that change duration for products/discounts.

Table D–2   (Cont.) Table B

Product Type	Service Acct (transfer)	Billing Account and Billing Profile	Pricing Info	Promotion Ref	Service ID	F&F List Ref	Billing Dates and End Date	Comments
Billing Item product (applies to an account).	NA	Yes	Yes	NA	NA	NA	Yes	Example: Penalty. No purchased product instance, hence no MACD.
								Billing Acct/Profile, Pricing info, Promotion reference, can be changed on revisions.
								Service Acct - Oracle BRM does not support transferring account-level products from one account to another.
								Siebel CRM disallows this change.
								Billing dates or end date - Cannot be reset using change orders.
								Cycle Start and Usage Start dates can be reset using revisions on billing initiation, if the dates that were previously set are not current.
								In two-phase billing, when Cycle Start and Usage Start dates have been set using billing initiation, can be reset using billing fulfillment if the dates that were previously set are not current.

D.2.1 Table B Notes

- Pricing information: This includes selling price, pricing commit type, dynamic discount method, discount amount, and discount percent.

- Because this release supports only account-level balance groups, and Oracle BRM disallows transferring a service pointing to the account-level balance group. This fails.

- Billing dates: This refers to purchase date, cycle start date, and usage start date.

- With Oracle BRM 7.4: Billing profile change alone, which results in the balance group being repointed to a different bill-info, is not supported. In the case of a nonpaying subordinate account, changing the paying parent (billing account *and* billing profile) is supported.

E

OLM - Examples of Changing the Paying Parent on Subordinate Accounts

This appendix describes three example scenarios for changing the paying parent on subordinate accounts.

This appendix contains the following sections:

- Section E.1, "Scenario One - Supported"
- Section E.2, "Scenario Two - Supported"
- Section E.3, "Scenario Three - Supported (with caveats)"

Table E–1 provides descriptions for the abbreviations used in the example scenarios.

Table E–1   Legend

Abbreviation	Description
S	Service
SA	Service Account
BA	Billing Account
BG	Balance Group
BP	Billing Profile
DBP	Dummy Bill Info

E.1 Scenario One - Supported

Table E–2 is the base scenario.

Table E–2   Base scenario

Action	#	SA	BA	BP
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP1

Figure E–1 shows the scenario results after the order is processed to billing.

Figure E–1   Base Scenario Results
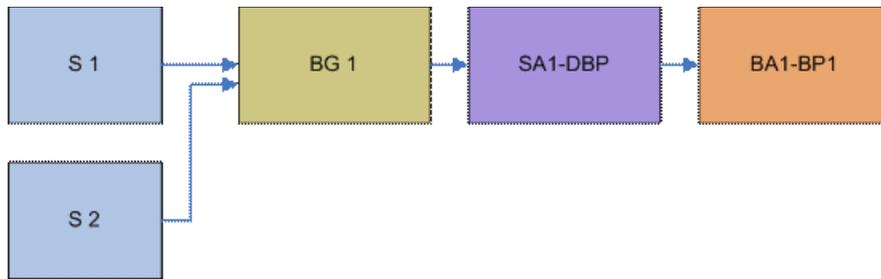
Table E–3 shows reparenting the subordinate to a different parent.

Table E–3   Reparenting subordinate to a different parent - Example 1

Action	#	SA	BA	BP
UPD	S1	SA1	BA2	BA2-BP2
UPD	S2	SA1	BA2	BA2-BP2

This works even if S2 is not included on the order. However, to keep Siebel Customer Relationship Management (Siebel CRM) assets synchronized, S2 should also be updated. Or as shown in Table E–4.

Table E–4   Reparenting subordinate to a different parent - Example 2

Action	#	SA	BA	BP
UPD	S1	SA1	BA2	BA2-BP2
UPD	S2	SA1	BA2	BA2-BP2
ADD	S3	SA1	BA2	BA2-BP2

Figure E–2 shows the results for the preceding scenario.

Figure E–2   Reparenting Subordinate to a Different Parent - Example 2 Results
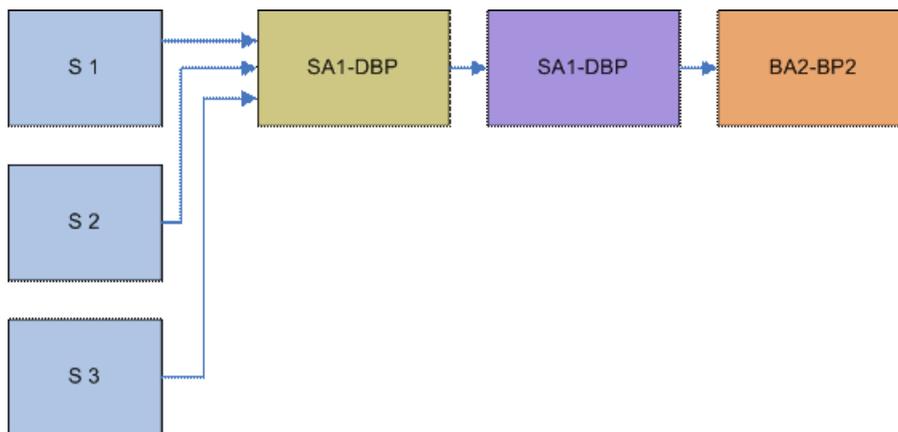
E.2 Scenario Two - Supported

Table E–5 is the base scenario:

A parent using different billing profiles to pay for their account and not each of the child accounts.

> **Note:** A parent cannot use multiple billing profile to pay for services of the same child account because the child account can have only one balance group, which can point to only one bill-info.

Table E–5   Base scenario

Action	#	SA	BA	BP
ADD	S1	BA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP2
ADD	S3	SA2	BA1	BA1-BP3

Figure E–3 shows the scenario results after the order is processed to billing.:

Figure E–3   Base Scenario Results
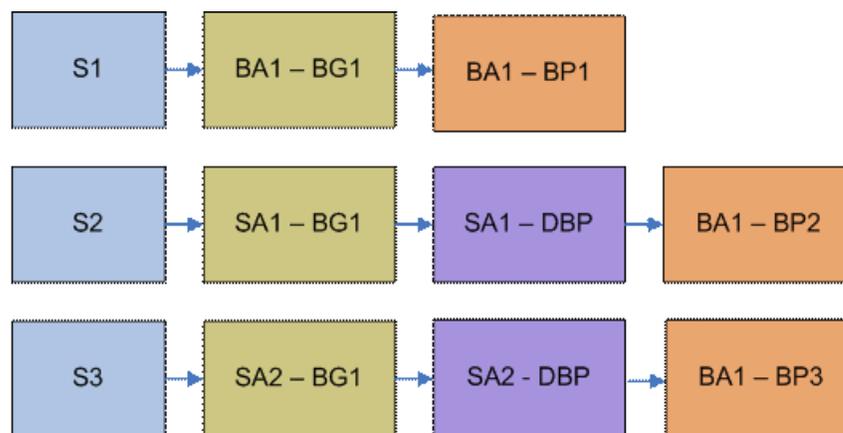
> **Note:** The account-level balance group for the parent account (BA1) references the first billing profile that is created for that account. In the previous scenario it is BP1. If the ADD line for the service purchase for the parent account (BA1) is not the first line on the order, then the account-level balance group references billing profile BP2, and the purchase of S1 fails because it is using BP1.

Table E–6 shows that the change order is processed, which reparents both the subordinate accounts to a new parent, and the same billing profile under the new parent.

Table E–6   Change order processed

Action	#	SA	BA	BP
UPDATE	S1	SA1	BA2	BA2-BP4
UPDATE	S2	SA2	BA2	BA2-BP4

Figure E–4 shows what is visible in Oracle Communications Billing and Revenue Management (Oracle BRM):

*Figure E–4   Outcome in Oracle BRM*

The repointing of the dummy bill-infos to the new billing profile is handled by the services that interface customer data to billing.

Alternates:

- A variant in which two different billing profiles under BA2 are used to pay for the different child accounts is also supported.

- A variant in which the billing profile changes but not the paying parent (such as in the base scenario, updating S3 to be paid for by BA1-BP2) is not supported. It fails in order billing integration because that involves repointing the balance group to a new dummy bill-info (pointing to BA1-BP2), which Oracle BRM does not allow.

Figure E–5 shows how S1 remains unchanged.

Figure E–5   Service 1 (S1)
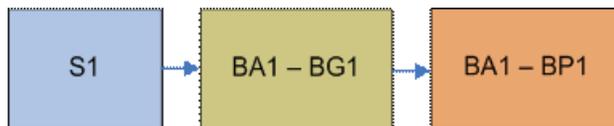
E.3 Scenario Three - Supported (with caveats)

Table E–7 is the base scenario.

Table E–7   Base scenario

Action	#	SA	BA	BP
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP2

After customer sync, the integration creates two dummy bill-infos:

- SA1-DBP1 pointing to BA1-BP1

- SA1-DBP2 pointing to BA1-BP2

The default account-level balance group points to BA1-BP1. (It points to the billing profile referenced on the first line on the order).

However, depending on which lines are sent for processing, the order billing integration behaves in one of two ways:

- *Both service bundles are sent for order billing integration at the same time (In the same call):*

  Since out-of-the-box (OOTB), the integration supports a single balance group (and therefore only a single bill-info for a self-paying account), for a given service account, Oracle AIA uses billing account and billing profile on the first service bundle purchase for all the remaining service bundles in the incoming request. The data in Oracle BRM appears the same as the example shown in Figure E–6.

Figure E–6   Results After Order is Processed to Billing
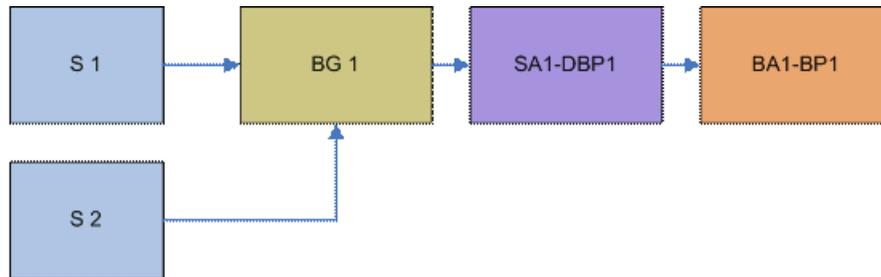
Even though this transaction is processed successfully it results in a mis-match between Siebel CRM assets and Oracle BRM in terms of the billing profile and bill-info used to pay for a given service. It is therefore recommended that if you are using OOTB functionality regarding balance group support, you ensure that orders in Siebel CRM follow the constraints (of single billing profile for a self-paying account).

- *The two service bundles are sent for order billing integration at different times (Different calls):*

  Irrespective of whether service bundle S1 is sent first or second, it is successfully processed. However order billing integration fails in attempting to re-point the default account-level balance group to SA1-DBP2 when purchasing service bundle S2.

  Resubmit the order (using a revision), as shown in Table E–8.

Table E–8   Order resubmitted

Action	#	SA	BA	BP
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	**BA1-BP1**

Figure E–7 shows what occurs after the order is processed to billing.

Figure E–7   Results After Order is Processed to Billing
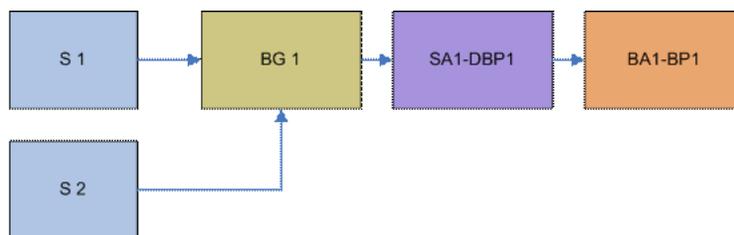
You still have the hanging subordinate bill-info SA1-DBP2 that is not used by any service. It causes a problem in the future (customer sync fails; it has validation that finds that a dummy bill profile is not being reparented). If any attempts are made to reparent SA1, because no asset is using SA1-DBP2, it cannot be repointed as part of any reparenting operation.

> **Caution:** No automated workaround is available for this issue. You must manually move SA1 to be under the new parent (thereby repointing all the subordinate dummy bill-infos) and then process the order to have assets reflect the new state.

F

Configuring Multiple Oracle BRM Instances for Communications Integrations

This appendix provides an overview of how system codes are used to identify each system instance in Oracle Application Integration Architecture (Oracle AIA) and describes how to configure additional Oracle Communications Billing and Revenue Management (Oracle BRM) instances for the process integrations in Oracle AIA for Communications.

This appendix includes the following sections:

- Section F.1, "Understanding System Codes in Oracle AIA"
- Section F.2, "Configuring Multiple Oracle BRM Instances - General Steps"
- Section F.3, "Creating a Data Source and Connection Factory"
- Section F.4, "Creating Logical Instances in Oracle AIA"
- Section F.5, "Creating Service Bundles in Siebel CRM"
- Section F.6, "Merging Logical Oracle BRM Instances into a Single Oracle BRM Instance"

F.1 Understanding System Codes in Oracle AIA

Each system instance is identified in Oracle AIA by a unique identifier, called a system code. The system codes help Oracle AIA identify the source or destination of a message.

Table F–1 describes the system codes that comes with Oracle AIA as delivered:

Table F–1   System Code Descriptions

System Code	Description
SEBL_01	The Siebel Customer Relationship Management (Siebel CRM) instance for order capture and trouble ticketing.
BRM_01	One Oracle BRM instance for order billing.
OSMCSF_01	The central fulfillment system (CFS) instance of Oracle Order and Service Management (Oracle OSM) system. This is the instance responsible for customer orders in order management.
OSMPROV_01	The provisioning system instance of Oracle OSM.

Figure F–1   Oracle AIA Topology
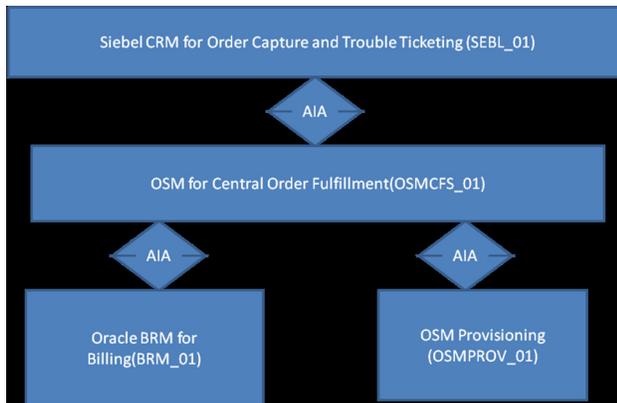
Oracle AIA uses cross-reference (xref) tables to maintain mapping of system-specific identifiers (account ID, product ID, and so on). One xref table exists per entity. In an xref table, columns are created for each system instance. System codes are used as column names.

Oracle AIA uses domain value maps (DVMs) to map values of enumeration type attributes (such as country code, state code, price type, and so on). One DVM exists for each enumeration type attribute. Columns are created for each system instance. System codes are used as column names.

System codes are also used to identify the sender and target in the enterprise business message (EBM) header for a given EBM message. Also in AIAConfigurationProperties.xml, system code values are used to name the properties that require instance-specific values such as EndPointURI (each system has a different end point URI). An example of such a property is:

```
<Property name="Routing.BRMSUBSCRIPTIONService.BRM_
01.EndpointURI">eis/BRM</Property>
```

Because Oracle OSM communicates to Oracle AIA using AIA EBMs, AIA Common IDs, and AIA DVM values, you do not require separate columns for OSMCFS_01 and OSMPROV_01 in DVMs and xrefs. Also, because Oracle AIA-OSM communications is using automatic queue synchronizations, no Oracle OSM-specific properties are in AIAConfigurationProperties.xml.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows."

Oracle OSM also recognizes the fulfillment topologies and assigns logical names to each system instance. These logical names should match the system codes configured in Oracle AIA.

F.2 Configuring Multiple Oracle BRM Instances - General Steps

Use this sample information as an overview of the process.

Assume that you have three billing instances. As shown in the previous section, the installation as delivered configures one Oracle BRM instance. To configure the second and third Oracle BRM instances, follow these steps. These steps guide you through the process to add billing instances. Repeat them for each additional Oracle BRM instance.

These abbreviations are used in this example:

- BRM_01: The first Oracle BRM instance that is installed as delivered.

- BRM_02: The second Oracle BRM instance for which the following sample configuration should be followed.

- BRM_03: The third Oracle BRM instance for which the following sample configuration should be followed.

Caution: The person performing this setup must have a working knowledge of Composite, Oracle Mediator, and JDeveloper IDE.

For more information about Composite, Oracle Mediator, and JDeveloper IDE, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

To configure a second Oracle BRM instance:

1. Modify all of the DVMs that have BRM columns to include the BRM_02 and BRM_03 columns.

   a. Copy the values from BRM_01 to BRM_02 and BRM_03 for all columns in the DVM table.

   The following is a list of the DVMs for which BRM_02 and BRM_03 column and values must be added:

   PRICE_TYPE.dvm

   ACCOUNTBALANCEADJUSTMENT_TAXTREATMENT.dvm

   PLANTIER_EFFECTIVITYMODE.dvm

   CONTACT_SALUTATION.dvm

   PRICE_OVERRIDETYPECODE.dvm

   ACCOUNTBALANCEADJUSTMENT_STATUS.dvm

   RATEDATA_DISCOUNTBRACKETCODE.dvm

   COLLECTION_ACTIONNAME.dvm

   PRICE_BILLINGPERIODCODE.dvm

   ACCOUNTBALANCEADJUSTMENT_REASON.dvm

   SALESORDER_LINESTATUS.dvm

   RATEPLAN_TAXAPPLICABILITYCODE.dvm

   COLLECTION_SUBSTATUS.dvm

   RECEIVEDPAYMENT_TYPE.dvm

   ADDRESS_COUNTRYID.dvm

   CUSTOMERPARTY_PAYPROFILE_DELIVERYPREF.dvm

   RESOURCE.dvm

   INSTALLEDPRODUCT_STATUS.dvm

   ACCOUNTBALANCEADJUSTMENT_USAGEALLOCATION_TAXTREATMENT.dvm

   MULTIPLE_DISCOUNT_PER_EVENT.dvm

   CUSTOMERPARTY_PAYPROFILE_PAYMETHODCODE.dvm

COLLECTION_PRIORITY.dvm

CUSTOMERPARTY_TYPECODE.dvm

CUSTOMERPARTY_PAYPROFILE_PAYTERMCODE.dvm

CUSTOMERPARTY_STATUSCODE.dvm

PRICETYPE_EVENT.dvm

STATE.dvm

ACCOUNTBALANCEADJUSTMENT_SUBSTATUS.dvm

SALESORDER_ACTIONCODE.dvm

PHONENUMBER_TYPE.dvm

ENTITY_TO_TARGET_APPLICATION.dvm

STOP_DISCOUNTING.dvm

PROVISIONING_TAG.dvm

PRICECHARGETYPEUOM.dvm

COLLECTION_STATUS.dvm

PLANTIER_RESTRICTIONTYPE.dvm

APPLIES_TO.dvm

PRICE_DISCOUNTTYPECODE.dvm

DAYS_OF_WEEK.dvm

CUSTOMERPARTY_PAYPROFILE_CREDIT_CARDTYPE.dvm

DISCOUNT_VALIDITY.dvm

RATEDATA_CANCELATIONPRORATIONCODE.dvm

LINEPRICE_TYPECODE.dvm

CUSTOMERPARTY_BILLPROFILE_BILLTYPECODE.dvm

RATEPLAN_TYPECODE.dvm

PRICECHARGETYPE.dvm

CUSTOMERPARTY_ACCOUNTTYPECODE.dvm

TARGET_ID.dvm

CURRENCY_CODE.dvm

RATEPLAN_UNITCODE.dvm

CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE.dvm

PRICE_IMPACTCLASSIFICATIONCODE.dvm

SALESORDER_STATUS.dvm

RATEDATA_PURCHASEPRORATIONCODE.dvm

ACCOUNTBALANCEADJUSTMENT_TYPE.dvm

CUSTOMERPARTY_PAYPROFILE_BANKACCOUNTTYPE.dvm

ADDRESS_COUNTRYSUBDIVID.dvm

GL_CODE.dvm

PRICE_MARKUPBASISCODE.dvm

ITEM_BILLING_TYPE_CODE.dvm

PLANIMPACT_UNITCODE.dvm

ITEM_BILLINGTYPECODE.dvm

PARTIAL_PURCHASE.dvm

LINEPRICE_UOMCODE.dvm

PROVINCE.dvm

TROUBLETICKET_AREA.dvm

b. Once all the columns have been added, load the DVMs to the Metadata Services (MDS) repository using the update deployment plan.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows."

2. Add BRM_02 and BRM_03 columns to the following XRef tables:

INSTALLEDPRODUCT_ID

CUSTOMERPARTY_LOCATIONREFID

CUSTOMERPARTY_CONTACT_PHONECOMMID

PaymentTermXREF

CUSTOMERPARTY_ADDRESSID

CUSTOMERPARTY_ACCOUNT_FAXCOMMID

PRICELINETYPE_ID

CUSTOMERPARTY_BILLPROFILEID

ITEMFORMULAMATERIAL_ID

PRICELINE_ID

CUSTOMERPARTY_PARTYCONTACTID

CUSTOMERPARTY_ACCOUNT_PHONECOMMID

CUSTOMERPARTY_CONTACT_FAXCOMMID

MANUFACTURINGROUTING_ID

CUSTOMERPARTY_PARTYID

SALESORDER_LINEID

CUSTOMERPARTY_ACCOUNT_COMMID

ITEMFORMULA_ID

ITEM_ITEMID

CUSTOMERPARTY_CONTACT_EMAILCOMMID

CUSTOMERPARTY_ACCOUNTID

SALESORDER_ID

ORGANIZATION_ID

CUSTOMERPARTY_CONTACTID

CUSTOMERPARTY_DEFAULTBALANCEGROUPID

CUSTOMERPARTY_PAYPROFILEID

CUSTOMERPARTY_CONTACT_COMMID

PRODUCTIONRECIPE_ID

CUSTOMERPARTY_PARTYLOCATIONID

- After adding the columns, load the Xrefs to the MDS repository using update deployment plan.

  The values are populated automatically into the new columns when the products are synchronized.

3. Add additional Oracle BRM connection factories that point to the new BRM_02 and BRM_03 instances:

   a. Go to the WebLogic Console, navigate to **Deployments, OracleBRMJCA15Adapter**.

   b. Add the following two new instances under the **Configuration, Outbound Connection Pools** tab:

   eis/BRM2

   eis/BRM3

   c. Enter the **Connection String** property value as *ip <host name/IP> <port>*. For example, *ip kappa.us.oracle.com 12345*.

   d. Enter the **Username** property value as *root.0.0.0.1*

   e. Save your changes.

   f. Update and start the adapter.

4. Modify all the BRM Provider service configurations in the AIAConfigurationProperties.xml file, which is located here:

   $AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config

   The service configuration contains the Partner link details to Oracle BRM services used in that particular BPEL Service. For example, Endpoint URI. This Endpoint URI determines which edge application to use and the location to reach the application.

   To route the messages to the appropriate Oracle BRM instance:

   a. Modify the BRM Provider service configuration for all the partner links.

   In the Communications pre-built integrations for this release the following services are used to route the messages to the Oracle BRM instance. Change the properties for these services:

   ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

   ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

   ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

   ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

   ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

   ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

   UpdateCreditAlertBRMCommsProvABCSImpl

SyncCustomerPartyListBRMCommsProvABCSImpl

QueryServiceUsageBRMCommsProvABCSImpl

QueryReceivedPaymentListBRMCommsProvABCSImpl

QueryInvoiceListBRMCommsProvABCSImpl

QueryInstalledProductListBRMCommsProvABCSImpl

QueryCustomerPartyListBRMCommsProvABCSImpl

QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl

ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

CreateReceivedPaymentBRMCommsProvABCSImpl

CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl

b. To update the AIAConfigurationProperties.xml file in the MDS repository, login to the AIA Console (http://<host>:<port>/AIA). Go to **Setup** and then select the **AIA Configuration** tab. Click **Reload**.

The following is a sample service configuration. Enter the highlighted statements:

```
<Property name="Routing.BRMBALService_ptt.BRM_
01.EndpointURI">eis/BRM</Property>
<Property name="Routing.BRMBALService_ptt.BRM_
02.EndpointURI">eis/BRM2</Property>
<Property name="Routing.BRMBALService_ptt.BRM_
03.EndpointURI">eis/BRM3</Property>

<Property name="Routing.BRMSUBSCRIPTIONService.BRM_
01.EndpointURI">eis/BRM</Property>
<Property name="Routing.BRMSUBSCRIPTIONService.BRM_
02.EndpointURI">eis/BRM2</Property>
<Property name="Routing.BRMSUBSCRIPTIONService.BRM_
03.EndpointURI">eis/BRM3</Property>

<Property name="Routing.BRMCUSTService.BRM_
01.EndpointURI">eis/BRM</Property>
<Property name="Routing.BRMCUSTService.BRM_
02.EndpointURI">eis/BRM2</Property>
<Property name="Routing.BRMCUSTService.BRM_
03.EndpointURI">eis/BRM3</Property>
```

F.2.1 Creating a New Consumer for Product Synchronization

The following steps must be followed to create a consumer for every new Oracle BRM instance. This consumer is used for product synchronization:

To create a consumer for product sync:

1. Create a new SOAProject project in Oracle JDeveloper, using the name SyncProductInfoChangeBRMAQ2.

2. Drag a new AQ Adapter component in the exposed services swimlane to display the Adapter Configuration Wizard - Welcome page, as shown in Figure F–2.

Figure F–2   Adapter Configuration Wizard - Welcome Page

3. Click **Next** to display Step 2 (Service Name).

   Go to the **Service Name** field and enter *SyncProductInfoChangeBRMAQ2*, as shown in Figure F–3.

Figure F–3   Adapter Configuration Wizard - Step 2 of 9
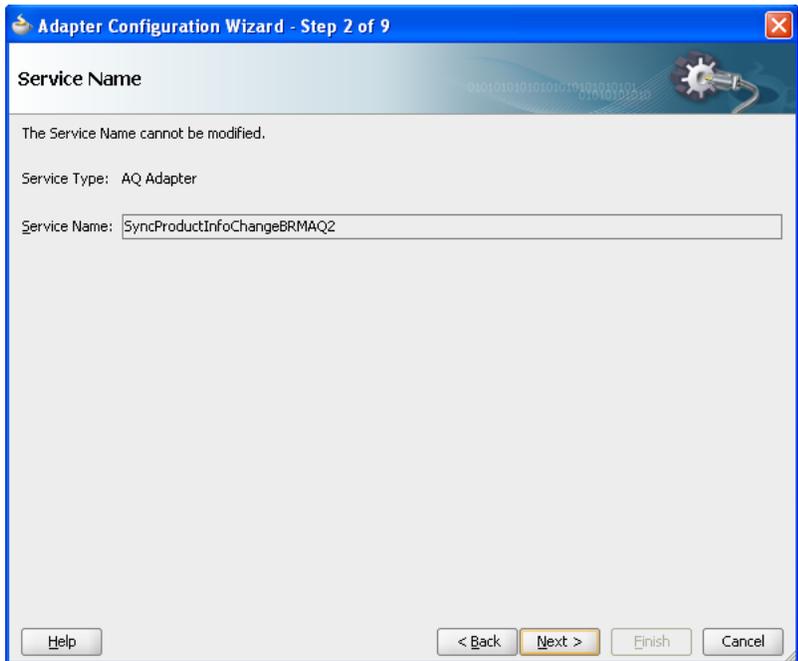
4. Click **Next** to display Step 3 (Service Connection).

   Go to **Connection** and select the Oracle BRM Database Connection. Go to the **JNDI Name** field and enter *eis/AQ/PortalEventSyncAQ2*, as shown in Figure F–4.

> **Caution:** The JNDI name is not created here by default. You must manually create the JNDI for the consumer that you are creating using the steps provided in Section F.3, "Creating a Data Source and Connection Factory."

Figure F–4   Adapter Configuration Wizard - Step 3 of 9
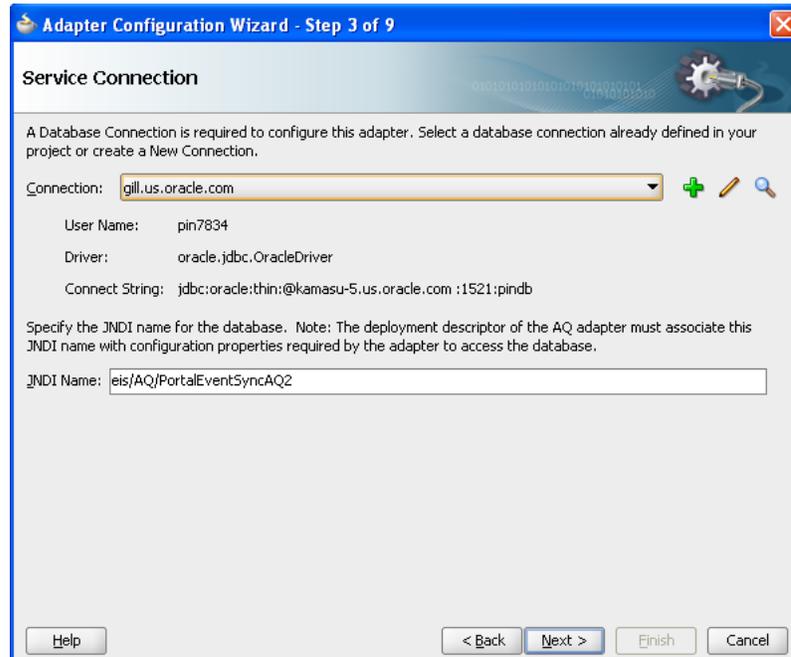
5. Click **Next** to display Step 4 (Adapter Interface).

Go to the **Interface** field and select *Define from operation and schema (specified later)*, as shown in Figure F–5.

Figure F–5 Adapter Configuration Wizard - Step 4 of 9
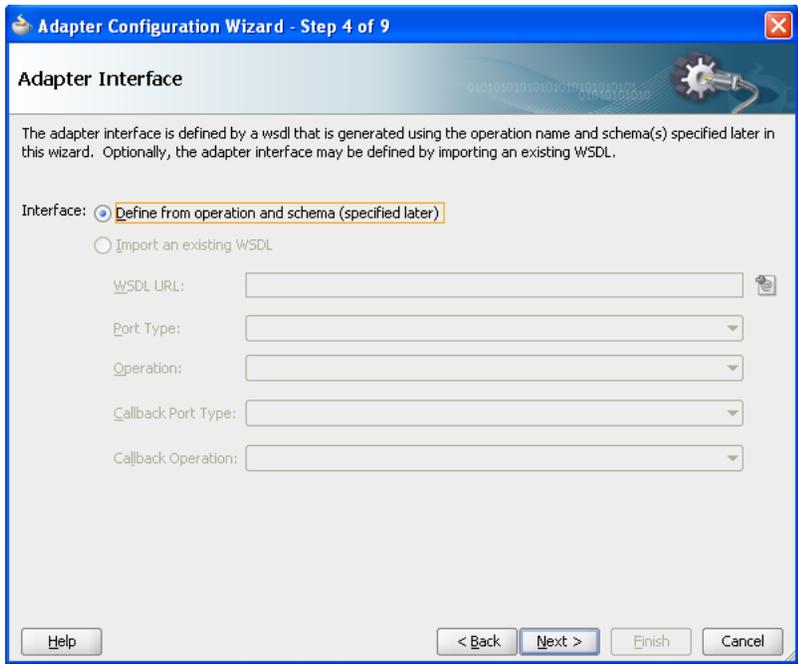
6. Click **Next** to display Step 5 (Operation).

    Go to the **Operation Name** field and enter *Dequeue*, as shown in Figure F–6.

Figure F–6 Adapter Configuration Wizard - Step 5 of 9
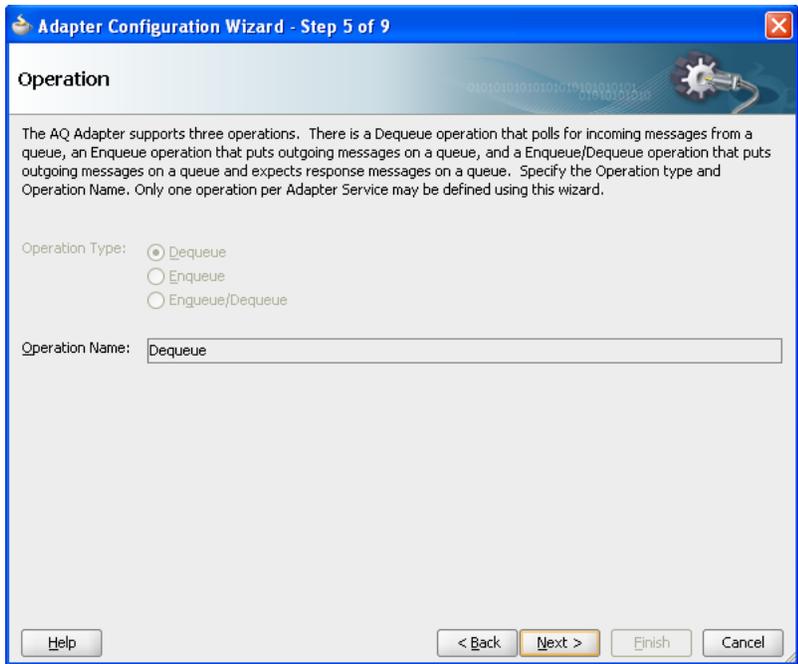
7. Click **Next** to display Step 6 (Queue Name).

    Go to the **Database Schema** field and select the additional Oracle BRM Instance's database schema, and then enter the queue name configured for this Oracle BRM instance, as shown in Figure F–7.

Figure F–7   Adapter Configuration Wizard - Step 6 of 9

8. Click **Next** to display Step 7 (Queue Parameters).

   Go to the **Correlation Id** field and enter *ProductInfoChange*, as shown in Figure F–8.

Figure F–8   Adapter Configuration Wizard - Step 7 of 9

9. Click **Next** to display Step 8 (Object Payload).

   Go to the **Business Payload** field and select *Whole Object PIN_EVENT_TY*, as shown in Figure F–9.

Figure F–9   Adapter Configuration Wizard - Step 8 of 9
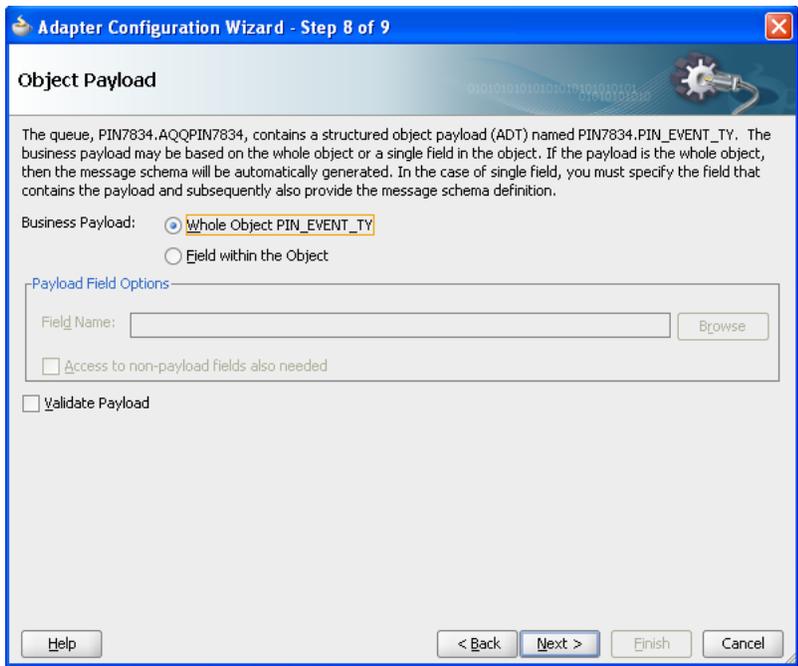
10. Click **Next** to display Step 9 (Finish), as shown in Figure F–10.

Click **Finish** to create the AQ Adapter service.

Figure F–10   Adapter Configuration Wizard - Step 9 of 9
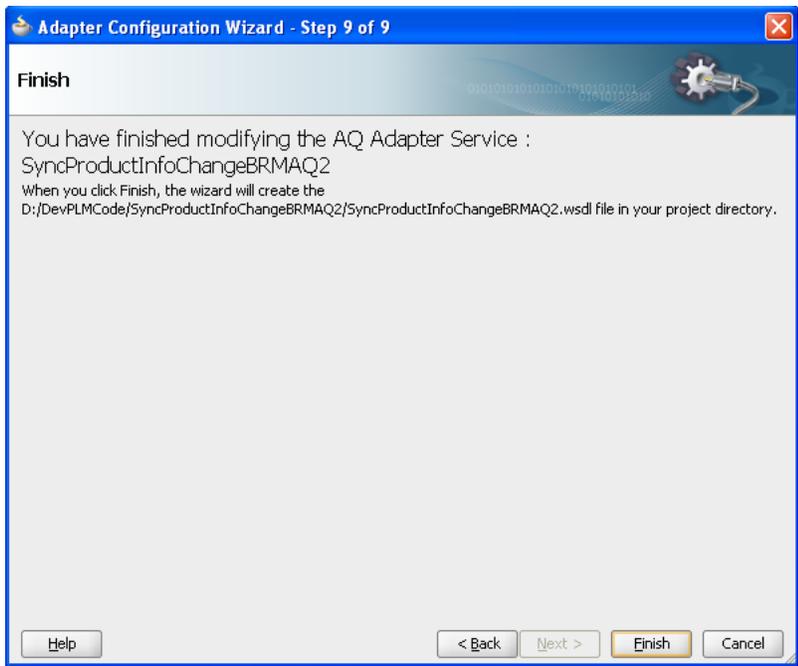
You must now: create a routing rule for the associated routing service.

- Create a routing rule for the associated routing service.
- Add a routing rule from the Adapter service created to Oracle Mediator.

- Create an External Reference for SyncProductBRMVCommsReqABCSImpl web service.

- Add a routing rule from Oracle Mediator to SyncProductBRMCommsReqABCSImpl web service for SyncProduct operation.

1. In composite.xml, provide the port and location information from concrete WSDL of the SyncProductBRMCommsReqABCSImpl web service.

```
<reference name="SyncProductBRMCommsReqABCSImpl"
ui:wsdlLocation="oramds:/apps/AIAMetaData/AIAComponents/ApplicationConnectorSer
viceLibrary/BRM/V1/RequesterABCS/SyncProductBRMCommsReqABCSImpl.wsdl">
<interface.wsdl
interface="http://xmlns.oracle.com/ABCSImpl/BRM/Industry/Comms/SyncProductBRMCo
mmsReqABCSImpl/V1#wsdl.interface(SyncProductBRMReqABCSImpl)"/>
<binding.ws
port="http://xmlns.oracle.com/ABCSImpl/BRM/Industry/Comms/SyncProductBRMCommsRe
qABCSImpl/V1#wsdl.endpoint(SyncProductBRMReqABCSImpl/SyncProductBRMReqABCSImpl)
" location="http://<host
name>:<Port>/soa-infra/services/default/SyncProductBRMCommsReqABCSImpl/SyncProd
uctBRMCommsReqABCSImpl?WSDL"/>
   </reference>
```

2. Create a mapping file PIN_EVENT_TY_To_ProductChangeInfo.xsl.

   a. Double-click Mediator to open the mplan.

   b. Add the transformation file PIN_EVENT_TY_To_ProductInfoChange.

   This file can be copied from the SyncProductInfoChangeBRMAQ/xsl folder, which is shipped with the Oracle Communications Order to Cash pre-built integration.

3. Make the following modification:

   The namespace http://xmlns.oracle.com/xdb/<BRM_CAPS_USERNAME> must be changed. The name space can be found in <USERNAME>_PIN_EVENT_ TY.xsd, which is created during the adapter creation.

   > **Caution:** Make sure that this namespace gets changed at the two places in the xsl file.

4. Deploy the Composite after the routing rule has been configured.

The same process must be followed for each additional Oracle BRM instance.

F.2.2 Creating a New Consumer for Discount Synchronization

The following steps must be followed to create a consumer for every new Oracle BRM instance. This consumer is used for discount synchronization:

To create a consumer for product sync:

1. Create a new SOAProject project in Oracle JDeveloper, using the name SyncDiscountInfoChangeBRMAQ2.

2. Drag a new AQ Adapter component in the exposed services swimlane to display the Adapter Configuration Wizard - Welcome page, as shown in Figure F–11.

Figure F–11 Adapter Configuration Wizard - Welcome Page

3. Click **Next** to display Step 2 (Service Name).

   Go to the **Service Name** field and enter *SyncDiscountInfoChangeBRMAQ2*, as shown in Figure F–12.

Figure F–12 Adapter Configuration Wizard - Step 2 of 9
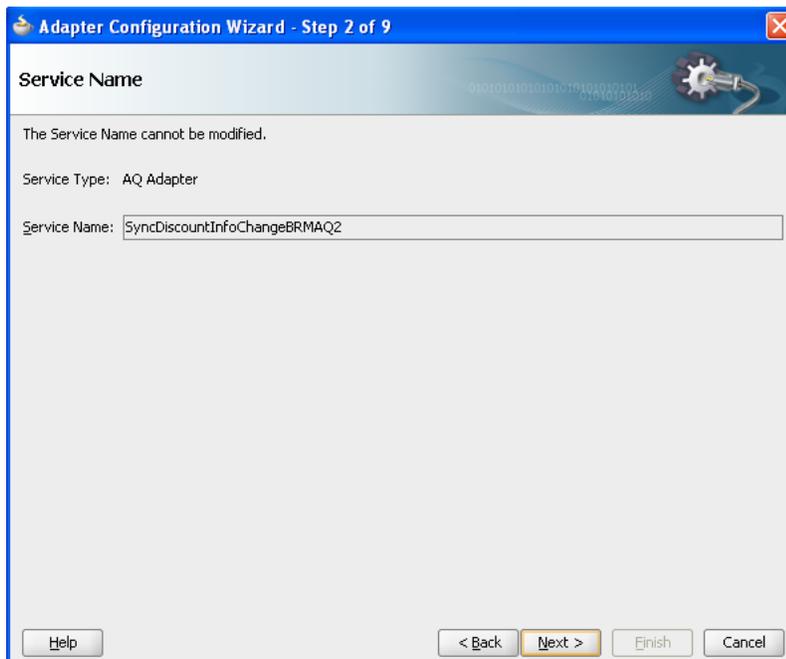
4. Click **Next** to display Step 3 (Service Connection).

   Go to **Connection** and select the Oracle BRM Database Connection. Go to the **JNDI Name** field and enter *eis/AQ/PortalEventSyncAQ2*, as shown in Figure F–13.

> **Caution:** The JNDI name is not created here by default. You must manually create the JNDI for the consumer that you are creating using the steps provided in Section F.3, "Creating a Data Source and Connection Factory."

Figure F–13   Adapter Configuration Wizard - Step 3 of 9
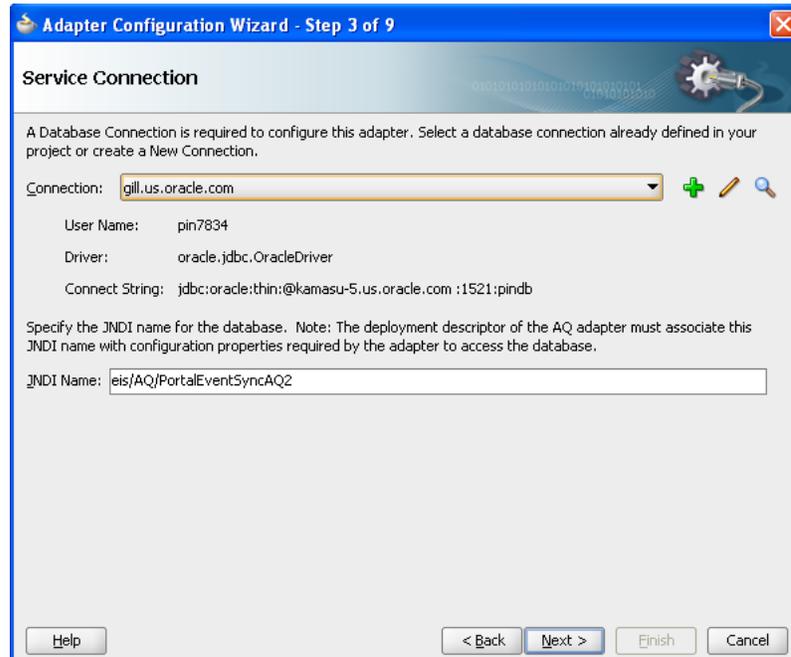
5. Click **Next** to display Step 4 (Adapter Interface).

   Go to the **Interface** field and select *Define from operation and schema (specified later)*, as shown in Figure F–14.

Figure F–14 Adapter Configuration Wizard - Step 4 of 9
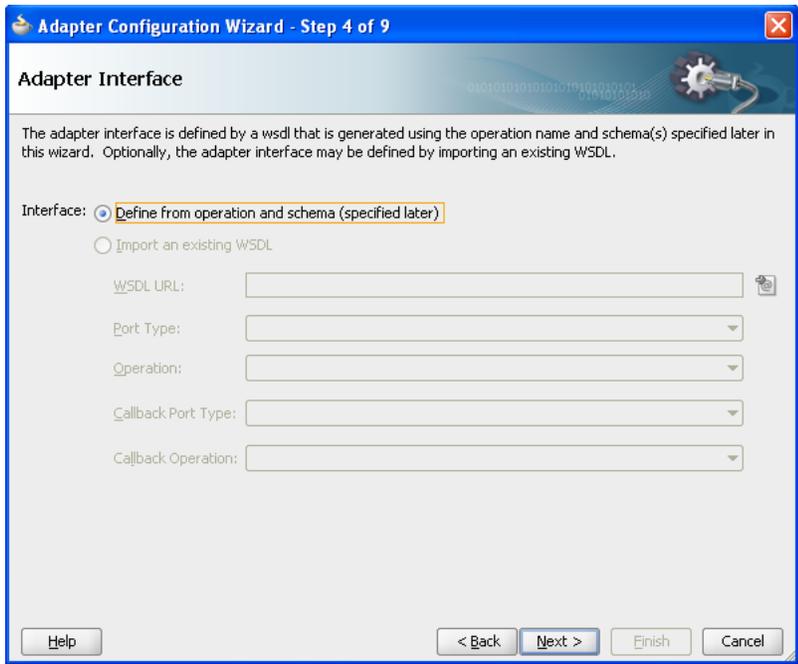
6. Click **Next** to display Step 5 (Operation).

   Go to the **Operation Name** field and enter *Dequeue*, as shown in Figure F–15.

Figure F–15 Adapter Configuration Wizard - Step 5 of 9
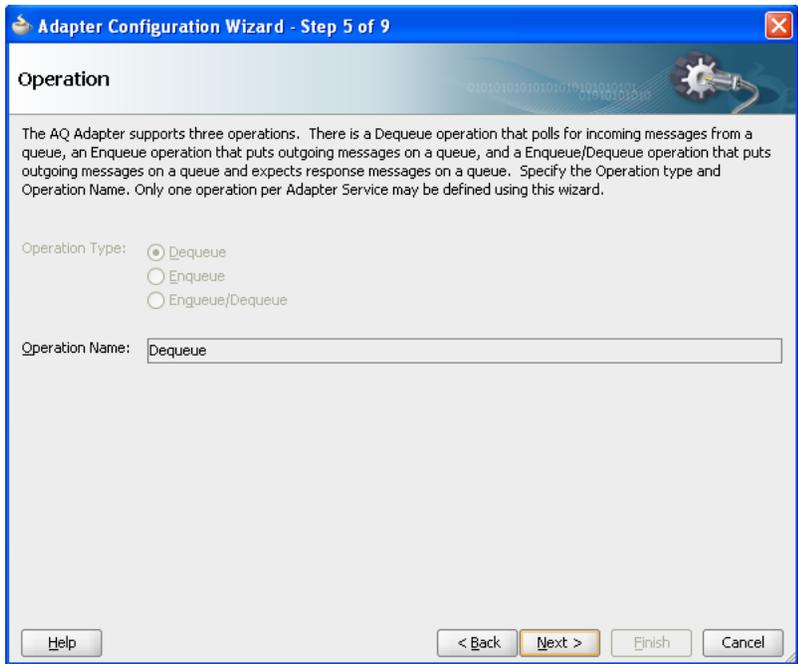
7. Click **Next** to display Step 6 (Queue Name).

   Go to the **Database Schema** field and select the additional Oracle BRM Instance's database schema, and then enter the queue name configured for this Oracle BRM instance, as shown in Figure F–16.

Figure F–16   Adapter Configuration Wizard - Step 6 of 9

8. Click **Next** to display Step 7 (Queue Parameters).

    Go to the **Correlation Id** field and enter *DiscountInfoChange*, as shown in
    Figure F–17.

Figure F–17   Adapter Configuration Wizard - Step 7 of 9

9. Click **Next** to display Step 8 (Object Payload).

    Go to the **Business Payload** field and select *Whole Object PIN_EVENT_TY*, as
    shown in Figure F–18.

Figure F–18 Adapter Configuration Wizard - Step 8 of 9
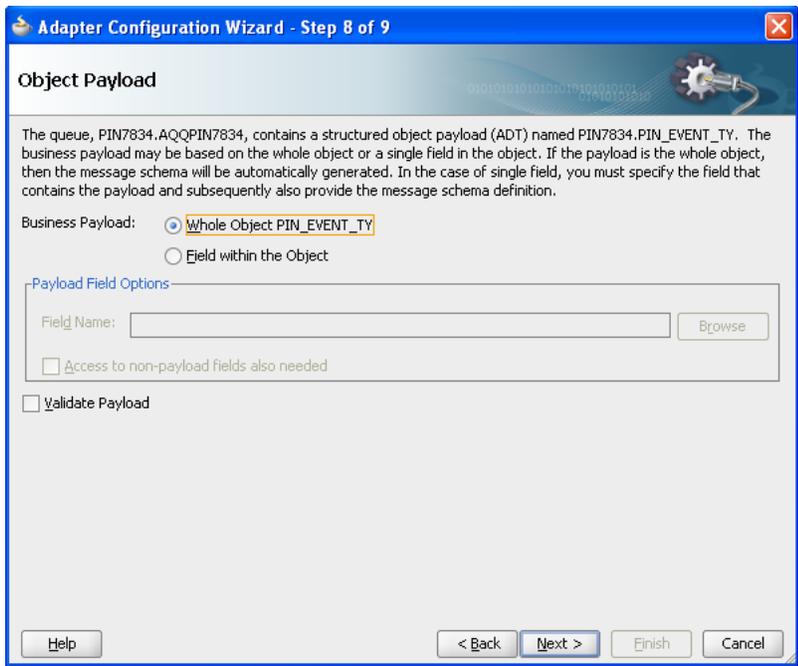
10. Click **Next** to display Step 9 (Finish), as shown in Figure F–19.

Click **Finish** to create the AQ Adapter service.

Figure F–19 Adapter Configuration Wizard - Step 9 of 9
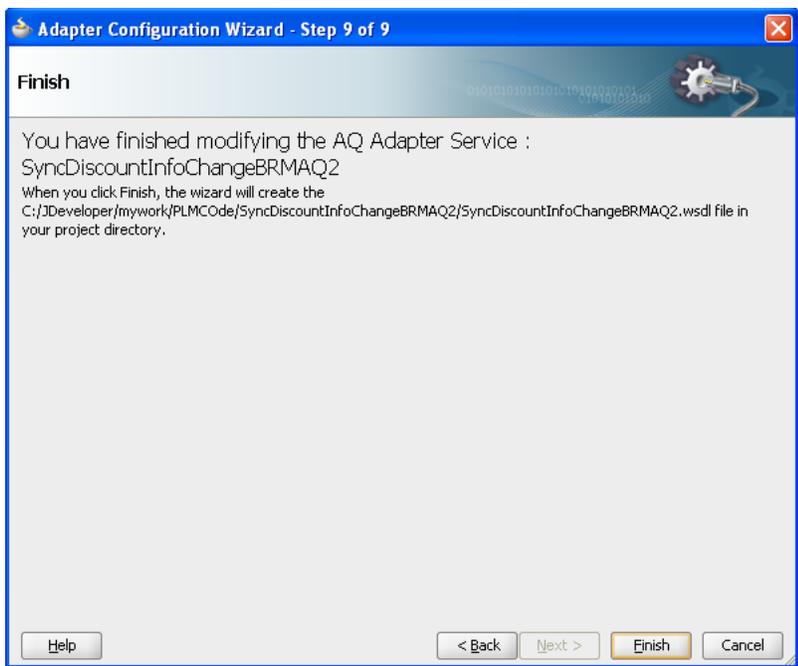
You must now: create a routing rule for the associated routing service.

- Create a routing rule for the associated routing service.
- Add a routing rule from the Adapter service created to Oracle Mediator.

- Create an External Reference for SyncDiscountBRMVCommsReqABCSImpl web service.

- Add a routing rule from Oracle Mediator to SyncDiscountBRMCommsReqABCSImpl web service for SyncDiscount operation.

1. In composite.xml, provide the port and location information from concrete WSDL of the SyncDiscountBRMCommsReqABCSImpl web service.

```
<reference name="SyncDiscountBRMCommsReqABCSImpl"
ui:wsdlLocation="oramds:/apps/AIAMetaData/AIAComponents/ApplicationConnectorSer
viceLibrary/BRM/V1/RequesterABCS/SyncDiscountBRMCommsReqABCSImpl.wsdl">
  <interface.wsdl
interface="http://xmlns.oracle.com/ABCSImpl/BRM/Industry/Comms/SyncDiscountBRMC
ommsReqABCSImpl/V1#wsdl.interface(SyncDiscountBRMReqABCSImpl)"/>
<binding.ws
port="http://xmlns.oracle.com/ABCSImpl/BRM/Industry/Comms/SyncDiscountBRMCommsR
eqABCSImpl/V1#wsdl.endpoint(SyncDiscountBRMReqABCSImpl/SyncDiscountBRMReqABCSIm
pl)" location="http://<host
name>:<port>/soa-infra/services/default/SyncDiscountBRMCommsReqABCSImpl/SyncDis
countBRMCommsReqABCSImpl?WSDL"/>
  </reference>
```

2. Create a mapping file PIN_EVENT_TY_To_DiscountChangeInfo.xsl.

   a. Double-click Mediator to open the mplan.

   b. Add the transformation file PIN_EVENT_TY_To_DiscountInfoChange.

   This file can be copied from the SyncDiscountInfoChangeBRMAQ/xsl folder, which is shipped with the Oracle Communications Order to Cash pre-built integration.

3. Make the following modification:

   The namespace http://xmlns.oracle.com/xdb/<BRM_CAPS_USERNAME> must be changed. The name space can be found in <USERNAME>_PIN_EVENT_ TY.xsd, which is created during the adapter creation.

   Caution: Make sure that this namespace gets changed at the two places in the xsl file.

   ---

4. Deploy the Composite after the routing rule has been configured.

The same process must be followed for each additional Oracle BRM instance.

F.2.3 Creating a New Consumer for Customer Updates

The following steps must be followed to create a consumer for every new Oracle BRM instance. This consumer is used for customer updates:

To create a consumer for customer updates:

1. Create a new SOAProject project in Oracle JDeveloper, using the name SyncCustomerPartyList<BRMInstanceID>CommsJMSConsumer, where the BRM Instance Id is the Id of the new Oracle BRM instance to be added. For example, SyncCustomerPartyListBRM_02CommsJMSConsumer.

2. Drag a new JMS Adapter component in the exposed services swimlane to display the Adapter Configuration Wizard - Welcome page, as shown in Figure F–20.

Figure F–20   Adapter Configuration Wizard - Welcome Page

3. Click **Next** to display Step 2 (Service Name).

   Go to the **Service Name** field and enter *SyncCustomerPartyListBRM_ 02CommsJMSConsumer*, as shown in .

Figure F–21   Adapter Configuration Wizard - Step 2 of 5
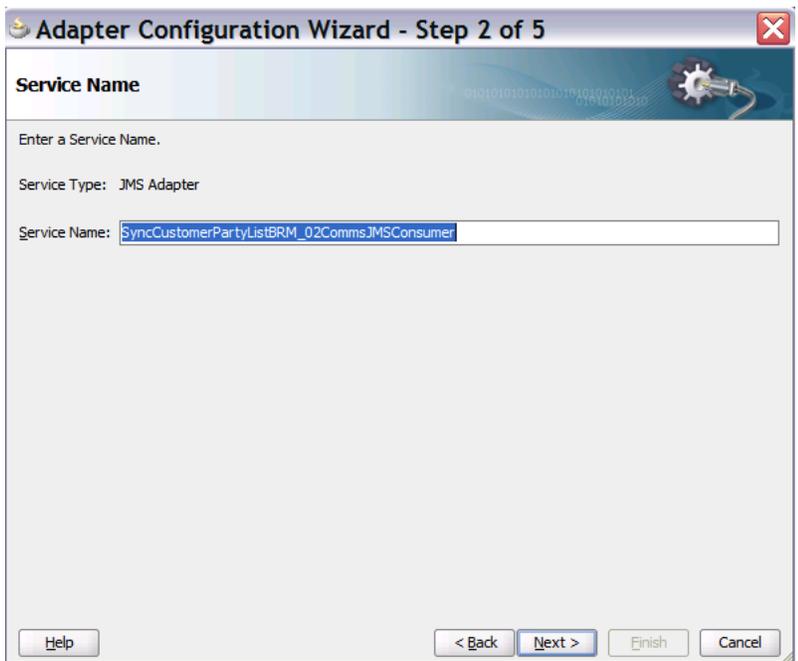
4. Click **Next** to display Step 3 (JMS Provider).

   Select **Oracle Enterprise Messaging Service (OEMS)** as the JMS Provider and then select *Oracle WebLogic JMS*, as shown in .

Figure F–22   Adapter Configuration Wizard - Step 3 of 5

5. Click **Next** to display Step 4 (Service Connection).

   Create an Application Server Connection to the Fusion Middleware (FMW) server. Go to **AppServer Connection** and select the corresponding FMW server connection from the dropdown, as shown in Figure F–23.

Figure F–23   Adapter Configuration Wizard - Step 4 of 6
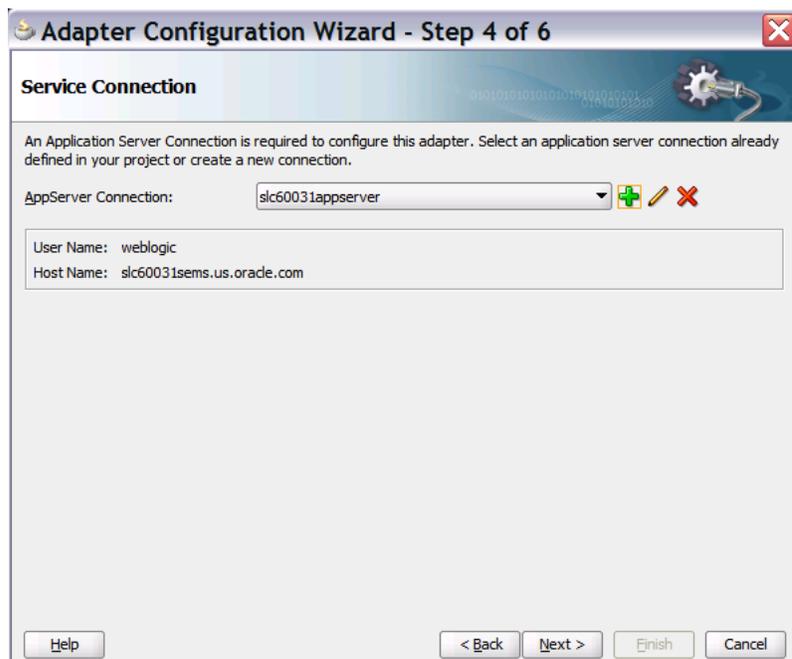
6. Click **Next** to display Step 5 (Adapter Interface).

Go to the **Interface** field and select *Define from operation and schema (specified later)*, as shown in Figure F–24.

Figure F–24 Adapter Configuration Wizard - Step 5 of 9
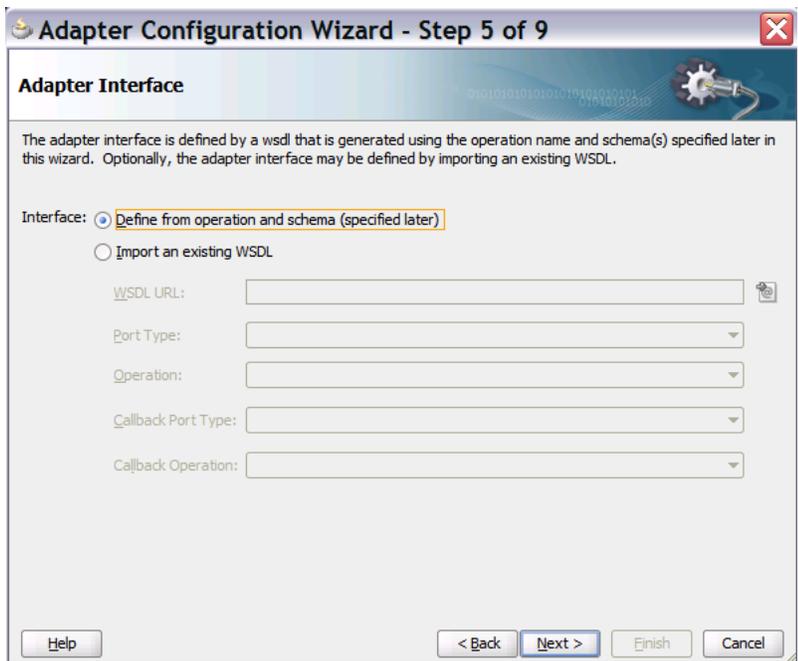
7. Click **Next** to display Step 6 (Operation).

   Go to the **Operation Type** field and select *Consume Message*. Then go to the **Operation Name** field and enter *Consume_Message*, as shown in Figure F–25.

Figure F–25 Adapter Configuration Wizard - Step 6 of 9
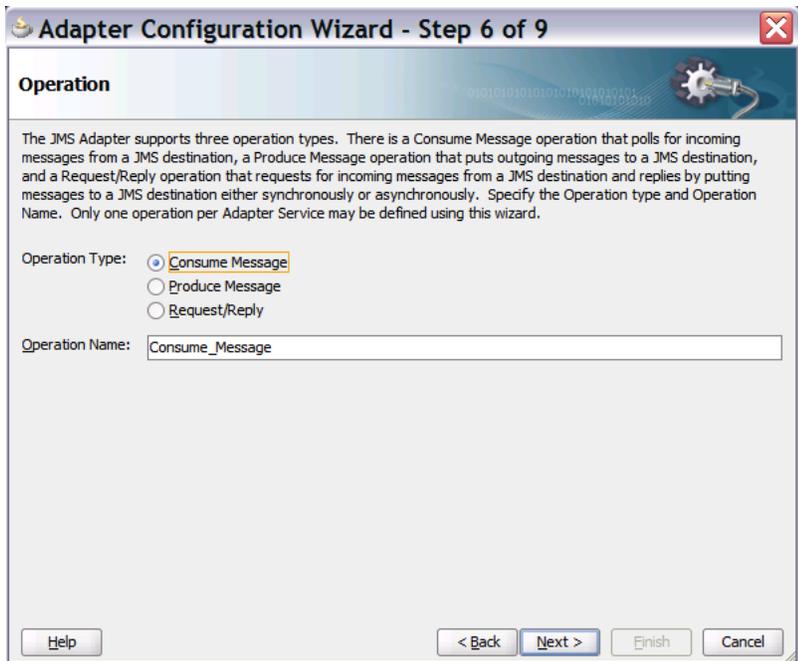
8. Click **Next** to display Step 7 (Consume Operation Parameters), as shown in Figure F–26.

Figure F–26   Adapter Configuration Wizard - Step 7 of 9

Go to the **Destination Name (Topic)** field and click **Browse** to select the required topic, as shown in Figure F–27. Click **OK** to return to Step 7.

Figure F–27   Select Destination
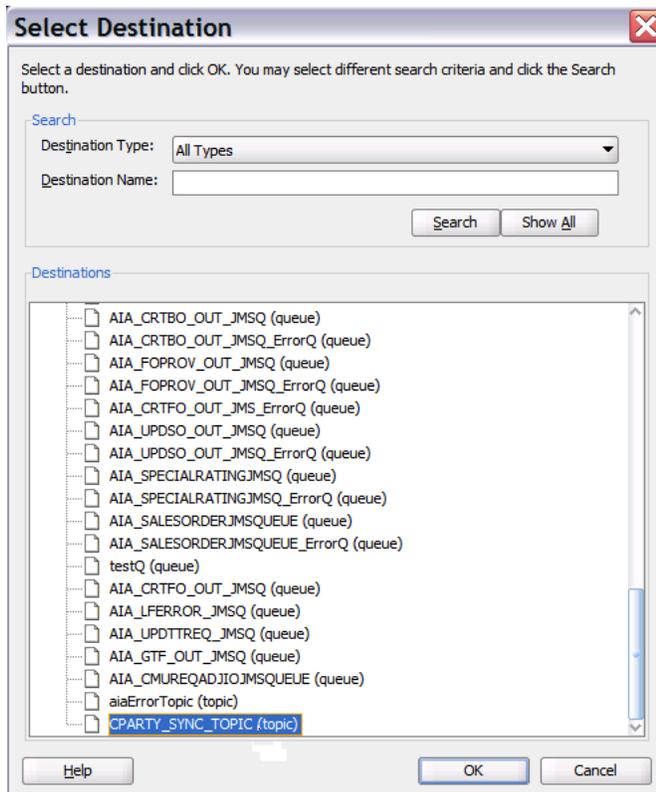
Then go to the **JNDI Name** field and enter the JNDI name for this JMS Connection.

9. Click **Next** to display Step 8 (Messages).

Go to the **URL** field and enter *oramds:/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/CustomerParty/V2/CustomerPartyEBM.xsd.* Go to the **Schema Element** field and enter *SyncCustoerPartyListEBM*, as shown in Figure F–28.

Figure F–28   Adapter Configuration Wizard - Step 8 of 9

10. Click **Next** to display Step 9 (Finish), as shown in Figure F–29.

Click **Finish** to create the JMS Adapter service.

Figure F–29   Adapter Configuration Wizard - Step 9 of 9

You must now create a routing rule for the associated routing service.

■   Create a routing rule for the associated routing service.

■   Add a routing rule against the Consumer_Message operation.

- Select the Endpoint service to be
  SyncCustomerPartyListBRMCommsProvABCSImpl.

The filter expression should be like this:

```
xref:lookupXRef('oramds:/apps/AIAMetaData/xref/CUSTOMERPARTY_
ACCOUNTID.xref','COMMON',$in.body/imp1:SyncCustomerPartyListEBM/imp1:DataArea/imp1
:SyncCustomerPartyList/imp1:CustomerPartyAccount/ns4:Identification/ns4:BusinessCo
mponentID,'BRM_01',false()) != ''
```

Along with the filter expression, a XSL must be added. Name the file
SetActionCodeandTargtID_BRM_02.xsl. The XSL should be like this:

```
<corecom:Target>
                <corecom:ID>
                    <xsl:text disable-output-escaping="no">BRM_02</xsl:text>
                </corecom:ID>
                <corecom:ApplicationTypeCode>
                    <xsl:value-of select="aia:getSystemType('BRM_02')"/>
                </corecom:ApplicationTypeCode>
            </corecom:Target>
```

Deploy the Composite after the routing rule has been configured.

F.3 Creating a Data Source and Connection Factory

This section provides the instructions for creating the data source and connection
factory.

To create the data source and connection factory:

1. Go to the WebLogic Server - Administration Console and navigate to **Services,
    DataSource, New Generic Data Source**.

2. Enter the data source name as *BRMEventSyncAQ2*. Enter the JDBC name as
    jdbc/aia/BRMEventSyncAQ2.

3. Click, **Next, Next, Next** to display the Connection Properties page, as shown in
    Figure F–30.

Figure F–30   WebLogic Server - Administration Console - Connection Properties
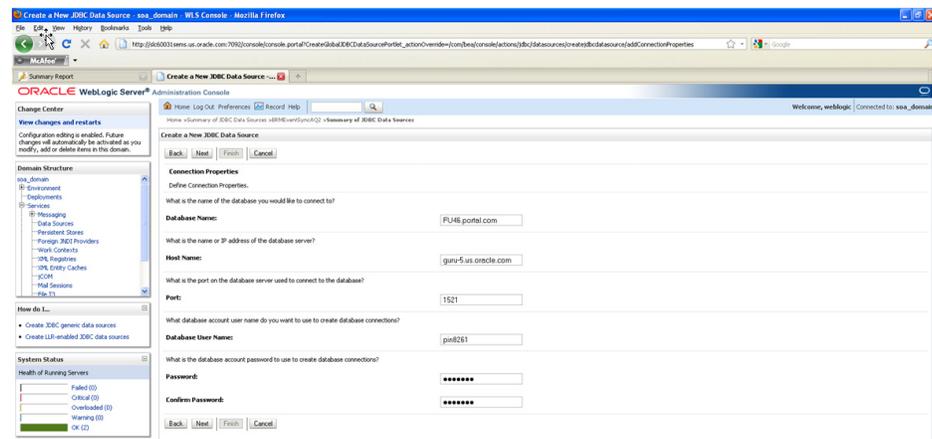
Enter all the Oracle BRM connection properties and then

4. Click **Next** to display the Test Configuration page, as shown in Figure F–31.

Figure F–31   WebLogic Server - Administration Console - Test Configuration
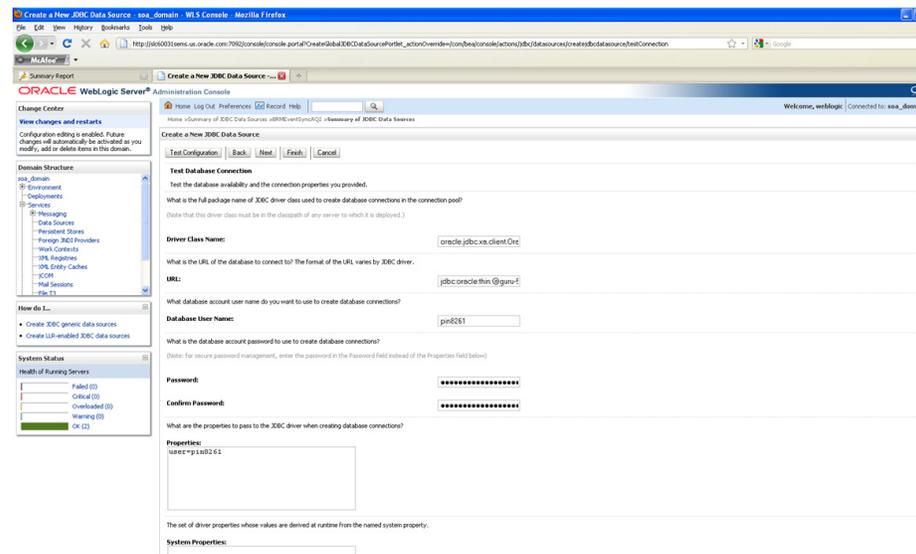
Click **Test Configuration** to verify that the provided details are correct. If the test is successful, click **Finish**.

5. Navigate to **Deployments** and click **AQAdapter**.

6. Go to the Configuration tab and expand the connection factory. Click **New** and select the connection factor.

7. Click **Next**. Provide the JNDI name as *eis/AQ/PortalEventSyncAQ2*.

8. Click **Finish**.

9. Click the newly created connection factory and go to the **XADataSourceName** field and enter *jdbc/aia/BRMEventSyncAQ2*. Click **Enter** and then click **Save**.

The same process must be followed for each additional Oracle BRM instance.

F.4 Creating Logical Instances in Oracle AIA

Whenever the product is synchronized to Oracle AIA (through the product lifecycle management (PLM) flows, Oracle BRM sends the instance ID in the payload to Oracle AIA to synchronize to Siebel CRM as follows:

```
<SyncProductReqMsg> <part xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
name="SyncProduct"> <brmproductabo:ProductInfoChange
xmlns:brmproductabo="http://www.portal.com/schemas/CRMSync"
brmproductabo:InstanceId="BRM_01">
```

Logical instances are defined in the Oracle AIA Console. The logical instance, shown in must be added or changed accordingly by the value given by Oracle BRM for each instance. For example, for the second Oracle BRM instance an entry must be added as shown in in the AIA_SYSTEM table.

Figure F–32   Logical Instance Example

> **Caution:** You must restart the server after adding entries. Otherwise, the changes are not recognized.

The logical instance name must be used in the AIAConfigurationProperties.xml, as specified in "To configure a second Oracle BRM instance:". For example, in Figure F–32, BRM_02 is the logical instance. Therefore, in the AIAConfigurationProperties.xml file, the end point configurator should be:

```
<Property name="Routing.BRMBALService_ptt.BRM_02.EndpointURI">eis/BRM2</Property>
```

The value *eis/BRM2* is the JNDI name specified when creating new consumers.

The same process must be followed for each additional Oracle BRM instance.

F.5 Creating Service Bundles in Siebel CRM

Currently, in *Typical* and *Reserved* topologies, Oracle OSM uses the following configuration to stamp the instances. This can be changed or configured in Oracle OSM according to customer requirements so you must consult your Oracle OSM administrator before configuring the instances.

- OSMCFS_01 - Oracle OSM Central Fulfillment (ALL Topologies)
- BRM_01 - Billing for broadband business (Typical Topology)
- BRM_02 - Billing for broadband residential (Typical Topology)
- BRM_03 - Billing for voip (Typical Topology)
- BRM_04 - Billing for both voip and broadband for all business and residential (Simple Topology)
- BRM_05 - Billing for voip and broadband business (Reserved Topology)
- BRM_06 - Billing for voip and broadband residential (Reserved Topology)
- OSMPROV_01 - Oracle OSM provisioning fulfillment for voip (Reserved Topology)
- OSMPROV_02 - Oracle OSM provisioning fulfillment for broadband US (Reserved and Typical Topology)
- OSMPROV_03 - Oracle OSM provisioning fulfillment for broadband UK (Typical Topology)
- OSMPROV_04 - Oracle OSM provisioning fulfillment for voip and broadband (Simple Topology)
- WFM_01 - Work force management (Typical Topology)
- SHP_01 - Shipping partnership Inc (Typical Topology)
- SHP_02 - Shipping in house (Typical Topology).

Table Format:

Table F–2   Typical Topology

Typical Topology
BRM_01 - Billing for broadband business
BRM_02 - Billing for broadband residential

Table F–2   (Cont.) Typical Topology

Typical Topology
BRM_03 - Billing for voip
OSMCFS_01 - Oracle OSM Central Fulfillment
OSMPROV_02 - Oracle OSM provisioning fulfillment for broadband US
OSMPROV_03 - Oracle OSM provisioning fulfillment for broadband UK
WFM_01 - Work force management
SHP_01 - Shipping partnership Inc
SHP_02 - Shipping in house

Table F–3   Reserved Topology

Reserved Topology
BRM_05 - Billing for voip and broadband business
BRM_06 - Billing for voip and broadband residential
OSMCFS_01 - Oracle OSM Central Fulfillment
OSMPROV_01 - Oracle OSM provisioning fulfillment for voip
OSMPROV_02 - Oracle OSM provisioning fulfillment for broadband US

Table F–4   Simple Topology

Simple Topology
BRM_04 - Billing for both voip and broadband for all business and residential
OSMCFS_01 - Oracle OSM Central Fulfillment
OSMPROV_04 - Oracle OSM provisioning fulfillment for voip and broadband

F.6 Merging Logical Oracle BRM Instances into a Single Oracle BRM Instance

In this case the change is related to combining multiple system instances into one.

If, for example, you start with two logical Oracle BRM instances and then later you decide to consolidate to a single instance. The only changes that must be made are in the AIAConfigurationProperties.xml.

In the AIAConfigurationProperties.xml file, change the End Point configuration URI to the same JNDI name.

For example, to point both these Oracle BRM instances to go to the *BRM_01* instance, just change the highlighted information:

Change this:

```
<Property name="Routing.BRMCUSTService.BRM_01.EndpointURI">eis/BRM</Property>
<Property name="Routing.BRMCUSTService.BRM_02.EndpointURI">eis/BRM2</Property>
```

To this:

```
<Property name="Routing.BRMCUSTService.BRM_01.EndpointURI">eis/BRM</Property>
<Property name="Routing.BRMCUSTService.BRM_02.EndpointURI">eis/BRM</Property>
```

G

OLM - Reconfiguring Oracle AIA for Communications

This appendix provides information about how to change the Oracle Communications Billing and Revenue Management (Oracle BRM) instance post installation.

This appendix includes the following sections:

- Section G.1, "Reconfiguring Oracle AIA for Communications Overview"
- Section G.2, "Changing the Oracle BRM Instance"

G.1 Reconfiguring Oracle AIA for Communications Overview

Many situations occur when the Oracle BRM instance that Oracle Application Integration Architecture (Oracle AIA) points to must be changed post installation. These include:

- Moving to a new Oracle BRM server due to replacement of hardware.
- Switching from a *Test* instance to a *Production* instance

> **Caution:** Before switching from one Oracle BRM instance to another, you must ensure that the new instance is a replica of the old instance. That is, all the data (such as accounts, services, products, discounts, and so on) in the old instance must also exist in the new instance, and they must also have matching IDs (POIDs). If this is not the case, failures occur in Oracle AIA. If any difference exists, then cross-reference (XREF) tables must be updated with the correct IDs before any of the flows are run.

G.2 Changing the Oracle BRM Instance

Oracle AIA and Oracle BRM communication happens through two adapters: inbound to Oracle AIA through Oracle Advanced Queuing (AQ) Adapter and inbound to Oracle BRM through Oracle BRM JCA Adapter. If a change occurs in the Oracle BRM instance, then the connection factories for both of these adapters must be changed.

To change the Oracle BRM instance:

1. Update connection parameters for the eis/BRM and any custom-created Oracle BRM connection factories for BRMJCAAdapter.

   The BRMJCAAdapter must be restarted after the changes are made.

> **Note:** The BRMJCAAdapter can be found under the Deployment section in the WebLogic console.

2. Update the Datasource PortalEventSyncAQ1DS with new database connection details.

3. If the Oracle BRM Event AQ queue name or the Oracle BRM schema name for the AQ Queue (or both) are changed, then replace occurrences of the old Event AQ queue name or the Oracle BRM schema name (or both) with the new names from <AIA_ HOME>/services/industry/Communications/BRM/AdapterServices/SyncProd uctInfoChangeBRMAQ.

4. Redeploy the services.

> **Caution:** The same changes must be incorporated to any custom connection factories or datasources, or composite services.

H

Expectations from a COM System for Billing Integration

This appendix provides a summary of the general billing integration expectations from a Central Order Management (COM) system for billing integration (Synchronize Fulfillment Order Accounts and Bill Fulfillment Order).

For feature-specific expectations, see the respective flow feature sections (for example, see Chapter 12, "OLM - Understanding the Bill Fulfillment Order Business Flow" for two-phase billing, time-based offers, and so on). *Oracle Order and Service Management (Oracle OSM) and OSM AIA Cartridges meet these documented feature specific expectations and the general expectations listed here*. If you are using a COM system other than Oracle OSM, it must comply by all these expectations.

Table H–1 lists the expectations from a COM system.

Table H–1 Expectations from a COM System

Number	Action	Comments
1	Account Synchronization *	The COM system sends the account synchronization message to AIA_CRTCUST_OUT_JMSQ Store and Forward (SAF) Queue. The Consumer (ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer) picks up the message and sends it to CommsProcessFulfillmentOrderBillingAccountListEBF for processing the account synchronization.
		Or, the COM system can call CommsProcessFulfillmentOrderBillingAccountListEBF directly to process the account synchronization message.
		To handle Oracle Billing and Revenue Management (Oracle BRM) limitations on customer hierarchy updates, all the lines on the order targeted for a given billing system must be sent at the same time. The target system ID must be stamped on the payload sent.
		The promotion line must go to every billing system in which promotion components are targeted.
		This service processes only lines with actions of ADD, UPDATE, and MOVE-ADD and ignores others. The COM system can choose to not send messages that do not have lines with these actions.
		This service processes only lines with billing type of *Service Bundle, Item, Subscription*, or *Discount*, and lines with product type of *Offer (Promotion)*. It ignores the rest. The COM system can optionally filter lines based on this.
2	Initiate Billing or Fulfill Billing *	The COM system sends the Order Interface message to AIA_CRTBO_OUT_JMSQ SAF Queue and the Consumer (ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer) picks up the message and sends it to ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl to interface the order to Oracle Billing and Revenue Management.
		Or, the COM system can call ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl directly to interface the order to Oracle BRM
		The COM system must send lines for promotions (product type is *Promotion*), account-level products (billing type is *Item, Subscription*, or *Discount*), service bundles (billing type is *Service Bundle*), or any combination of these destined for a single billing system. Service Bundle refers to the Service Bundle line and all its children. This service ignores other kinds of lines (for example, Non Service Bundle CP lines); therefore, the COM system can optionally filter them out. The target system ID must be stamped on the payload sent to the service.
		The COM system must interface the promotion lines to billing either before the first service bundle or the account-level product (including penalties) for the promotion along with it. This applies to both Initiate Billing and Fulfill Billing.
		The COM system must interface MOVE-ADD lines with the corresponding MOVE-DELETE lines (linked using related line ID).
		The COM system must interface the one-time charge lines tied to service bundle lines with the service bundles (linked using related line ID).
		The COM system must interface promotion penalty charges with the promotion line (linked using related line ID).

Table H–1 (Cont.) Expectations from a COM System

Number	Action	Comments
3	Update Sales Order	The COM system sends the update sales order message to AIA_UPDSO_OUT_JMSQ SAF Queue and the Consumer (UpdateSalesOrderOSMCFSCommsJMSConsumer) picks up the message and sends it to UpdateSalesOrderSiebelCommsProvABCSImpl to update the sales order status in Siebel CRM.
		Or, the COMS system can call UpdateSalesOrderSiebelCommsProvABCSImpl directly to update the sales order status in Siebel CRM.
		The COM system is responsible for consolidating status updates and sending only updates that are significant to Siebel CRM or the end customer. It must set a status of *Completed* for lines that complete fulfillment as this triggers auto-asset functionality in Siebel CRM. Assets are required for supporting Change Order functionality.
		For more information about how the COM system can use the extended status attributes and other guidelines, see Chapter 16, "OLM - Understanding the Update Sales Order Business Flow."

* - Out-of-the-box (OOTB) these do not send a response back to the caller for system or business errors (Oracle OSM and the OSM AIA Cartridges do not expect such a response).

If your COM system requires a response for business errors (or for business and system errors), see Section 11.5.2, "CommsProcessFulfillmentOrderBillingAccountListEBF" and Section 13.5.2, "ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl" for information about how you can achieve this.

I

Using the Oracle Mediator Resequencer Feature

The Oracle Mediator Resequencer feature is used by various integration flows to ensure that messages are processed in a particular sequence.

This appendix contains the following sections:

- Section I.1, "Queues and Flows Enabled for Sequencing"
- Section I.2, "Resolving Errors in Flows with Resequencer"

For more information about the resequencer, see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, "Resequencing in Oracle Mediator."

I.1 Queues and Flows Enabled for Sequencing

This table lists the queues and flows that are enabled for sequencing.

> **Note:** Revision Order support - Oracle OSM manages scenarios where multiple revisions for the same order are sent out of sequence. If you are using a different Order Management system it must have similar support.

Table I–1   Queues and Flows Enabled for Sequencing

Oracle AIA Queue	Flow	JMS Priority Set By	Sequencing Criteria	Comments
AIA_UPDSO_ OUT_JMSQ	Update order flow from Order Management/O SM to Oracle AIA for Siebel CRM system.	Not Set	Group By: Account ID mentioned in the ObjectCrossReference section of the update message(/UpdateSales OrderEBM/EBMHeader /Sender/ObjectCrossRe ference/SenderObjectId entification/AlternateO bjectKey/ID[@schemeI D = 'CUSTOMERPARTY_ ACCOUNTID' and @schemeAgencyID = 'COMMON']) Order of Processing: FIFO (First in First Out). Composite Name: UpdateSalesOrderOSM CFSCommsJMSConsum er.	**Note:** The consumer in the Create Trouble Ticket for Order Fallout business flow is only a sample. The resequencer in this flow ensures that multiple updates for the same order are processed in the right sequence.
AIA_CRTCUST_ OUT_JMSQ	Order flow from Order Management/O SM to Oracle AIA for customer data creation in billing.	Order Management/OSM	Group By: Account ID on the message (this is either the Billing account or the Service account on the order line that must be created in billing) and the target system identifier. concat($in.SyncCustome rPartyListEBM/ns0:Syn cCustomerPartyListEB M/ns0:DataArea/ns0:S yncCustomerPartyList/ ns0:CustomerPartyAcco unt/corecom:Identificati on/corecom:Applicatio nObjectKey/corecom:ID [@schemeID='AccountI d'], $in.SyncCustomerParty ListEBM/ns0:SyncCusto merPartyListEBM/corec om:EBMHeader/coreco m:Target/corecom:ID) Order of Processing: FIFO (First in First Out). Composite Name: CommunicationsCusto merPartyEBSV2Reseque ncer.	The resequencer in this flow ensures that the solution can successfully handle processing of concurrent orders for the same customer.

Table I–1   (Cont.) Queues and Flows Enabled for Sequencing

Oracle AIA Queue	Flow	JMS Priority Set By	Sequencing Criteria	Comments
--	Sync customer flow from Siebel CRM system to Oracle Customer Hub.	Not Set	Group By: AccountID. Order of Processing: FIFO (First in First Out). Composite Name: SyncAcctSiebelAggrEventConsumer SyncContSiebelAggrEventConsumer.	Also available in the Agent Assisted Billing Care pre-built integration. The resequencer in this flow ensures that multiple updates for the same customer are processed in the right sequence.
AIA_CRTFO_IN_JMSQ	Order flow from Oracle AIA to OSM	ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	na
AIA_CRTBO_OUT_JMSQ	Order flow from OSM to AIA for billing.	OSM)	None as delivered. Customers can use ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer to implement custom sequencing.	na
AIA_UPDBO_IN_JMSQ	Order flow from AIA (from billing) to OSM	ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	na
AIA_UPDCUST_IN_JMSQ	Response of the customer creation in billing from AIA to OSM	ProcessFOBillingAccountListRespOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	na
AIA_CRTFO_OUT_JMSQ	Create Fulfillment Order flow from OSM to Oracle AIA for the provisioning system	OSM	None as delivered. Customer can use ProcessProvisioningOrderOSMCFSCommsJMSConsumer to implement custom sequencing.	na

Table I–1   (Cont.) Queues and Flows Enabled for Sequencing

Oracle AIA Queue	Flow	JMS Priority Set By	Sequencing Criteria	Comments
AIA_FOCFS_IN_ JMSQ	Update Fulfillment Order flow from Oracle AIA (from the provisioning system) to OSM)	ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	na
AIA_FOPROV_ OUT_JMSQ	Update Fulfillment Order flow from the provisioning system to Oracle AIA (for OSM)	Provisioning System	None as delivered. Customer can use ProcessFulfillmentOrder UpdateOSMPROVCom msJMSConsumer to implement custom sequencing.	na
AIA_FOPROV_ IN_JMSQ	Create Fulfillment Order from Oracle AIA (from OSM) to the provisioning system.	ProcessProvisioning OrderOSMPROVCo mmsJMSProducer	None. (Onus is on OSM.)	na

I.2 Resolving Errors in Flows with Resequencer

If an error occurs in the Oracle BRM Customer provider, the message may be blocked in the CommunicationsCustomerPartyEBSV2Resequencer service and the error message may not propagate back to CommsProcessFulfillmentOrderBillingAccountListEBF. In these situations, fallout specialists must take corrective action on the resequencer to move the flow. If the message fails due to a system error (for example, if the target system is unavailable), then fallout specialists must retry the message from resequencer. If the message fails because of a business error, then the fallout specialist must unblock the resequencer.

An error may occur in the Siebel provider after it is consumed by UpdateSalesOrderOSMCFSCommsJMSConsumer and sent for processing. In this situation the messages are rolled back to the resequencer for this consumer and any subsequent order updates for that particular order are not processed. If this occurs, the fallout specialist must take corrective action on this resequencer to move the flow like the ones described above. If the message fails due to a system error (for example, if the target system is unavailable), then fallout specialists must retry the message from resequencer. If the message fails because of a business error, then the fallout specialist must unblock the resequencer.

For more information about retrying messages from resequencer and unblocking the resequencer, see *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*, "Monitoring Oracle Mediator Service Components and Engines", Monitoring Resequenced Messages.

J

OLM - Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant

This appendix describes the fields and attributes that must be passed to make Oracle Application Integration Architecture (Oracle AIA) processes fallout-compliant.

New services introduced, which need to participate in the order fallout notification mechanism, must be included in the AIA Error Notifications table with the appropriate Error_Type and Error_Ext_Handler.

This appendix contains the following sections:

- Section J.1, "Populating Sender Context Information in the EBM Header"
- Section J.2, "Populating the Enriched Fault Message with Business Faults"
- Section J.3, "Populating the Enriched Fault Message in Services without EBMs"

J.1 Populating Sender Context Information in the EBM Header

For all system or composite faults (binding and remote), the fault policy is initiated and publishes a notification message. By ensuring that your process has the following context information supplied, the order fallout management extension handler application programming language (API) constructs an enriched fault message.

All the enterprise business messages (EBMs) for order processing passes the following information as a sender reference in the EBMHeader. This list shows the information that you are required to pass for fallout:

- Order ID - Business Component ID of the Order - SalesOrder / Provisioning Order / Fulfillment Order / Fulfillment Billing Order
- Order Number - ID of the order - FulfillmentOrder#/ProvisioningOrder#/SalesOrder# (optional - required only if available)
- SalesOrderID - Alternate Object Key - storing the Sales Order Common ID
- Sales Order Number - Alternate Object Key - storing the Sales Order Number (Siebel CRM value)
- Sales Order Revision Number - Alternate Object Key - storing the Sales Order Number (Siebel CRM value)
- Common Account ID - Alternate Object key - storing the Common Account ID
- Account ID - Alternate Object key - storing the Siebel CRM Account ID (only for Sales Order EBM because the account information in the Xref is rolled back)

■ Account Name - Alternate Object Key - storing the Siebel CRM Account Name

Along with these fields, populate the SchemeID field indicating the name, and the SchemeAgencyID indicating the column name.

The attribute value for schemeAgencyId of SALESORDER_NUMER is considered the system code of the system from which the order was placed (Order Originating System Code)

This information should be entered in the EBM Header in the following path:

EBMHeader / Sender / ObjectCrossReference / SenderObjectIdentification /

Example J–1 is a sample EBMHeader section.

Example J–1   Sample EBMHeader Section

```
<EBMHeader>
<Sender>
<ObjectCrossReference>
 <SenderObjectIdentification>
<BusinessComponentID> OrderId</BusinenessComponentID>
            <ID> Order# (if any)</ID>
<ApplicationObjectKey>
<ID schemeID="SALESORDER_ID"schemeAgencyID="SEBL_01">SalesOrderID</ID>
</ApplicationObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_ID " schemeAgencyID="COMMON">SalesOrderCommonID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_NUMBER" schemeAgenyId="SEBL_01">SalesOrderNumber</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgenyId="SEBL_01">SalesOrderRevision</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgenyId="COMMON">CommonAccountID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgenyId="SEBL_01">Siebel Account_
ID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTNAME" schemeAgenyId="SEBL_01">Account Name</ID>
</AlternateObjectKey>
</SenderObjectIdentification>
</ObjectCrossReference>
</Sender>
</EBMHeader>
```

Only the underlined elements are required for the SalesOrder EBM.

J.2 Populating the Enriched Fault Message with Business Faults

In case non-partner link errors or business faults are in the business process execution language (BPEL) processes (where the BPEL process is creating the fault message and calling the Oracle AIA Async Error handling process), the expectation is that the ApplicationFaultData is also populated.

ApplicationFaultData is an xsd: Any field in the fault message:

Fault/FaultNotification/FaultMessage/ApplicationFaultData

The BPEL processes are expected to construct a variable of element type ApplicationFaultData defined in this xsd: http://{httphostname}:{http portname}/AIAComponents/PIPS/Communications/Schemas/OrderFailureData.xsd

The fields defined in the xsd and how they must be used are listed here.

- ApplicationFaultData / OrderFailureData / OrderID

  BusinessComponentID - SalesOrder / Provisioning Order / Fulfillment Order / Fulfillment Billing Order

  ID - SalesOrder # / Provisioning Order # / Fulfillment Order # / Fulfillment Billing Order # (If available)

  ApplicationObjectKey - If available

  AlternateObjectKey - SALESORDER_ID

  AlternateObjectKey - SALESORDER_NUMBER

  AlternateObjectKey - SALESORDER_REVISION

  AlternateObjectKey - FULFILLMENTSYSTEM_ID

  Example J–2 is a sample definition.

Example J–2 Sample Definition 1

```
<BusinessComponentID>  Order ID   </BusinessComponentID>
        <ID> Order# (if any)</ID>
        <ApplicationObjectKey>
<ID schemeID="SALESORDER_ID"schemeAgencyID="SEBL_01">SalesOrderID</ID>
        </ApplicationObjectKey>
      <AlternateObjectKey>
  <ID schemeID="SALESORDER_ID"
schemeAgencyID="COMMON">SalesOrderCommonID</ID>
        </AlternateObjectKey>
        <AlternateObjectKey>
<ID schemeID="SALESORDER_NUMBER" schemeAgenyId="SEBL_01">SalesOrderNumber</ID>
         </AlternateObjectKey>
         <AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgenyId="SEBL_01">SalesOrderRevision</ID>
        </AlternateObjectKey>
        <AlternateObjectKey>
<ID schemeID="FULFILLMENTSYSTEM_ID "
schemeAgenyId="FulfillmentSystemAppID">OrderID in the Fulfillment System</ID>
        </AlternateObjectKey>
```

- ApplicationFaultData / OrderFailureData / AccountID

  BusinessComponentID - CommonAccountID

  ID - Account Name

  ApplicationObjectKey - Siebel AccountID (required only with SalesOrder EBM)

  Example J–3 is a sample definition.

Example J–3 Sample Definition 2

```
<BusinessComponentID schemeID="CUSTOMERPARTY_ACCOUNTID"
schemeAgencyID="COMMON">AccountID</BusinessComponentID>
<ID schemeID="CUSTOMERPARTY_ACCOUNTNAME" schemeAgencyID="SEBL_01">AccountName</ID>
<ApplicationObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgencyID="SEBL_01">88-878PX</ID>
```

```
</ApplicationObjectKey>
```

■ ApplicationFaultData / OrderFailureData / ProductID

Information regarding the Product / Discount of the failed order line.

With an entire order failure, this can be mapped for the product corresponding to the first line item of the order.

Example J–4 is a sample definition.

Example J–4   Sample Definition 3
```
<BusinessComponentID schemeID="ITEM_ID" schemeAgencyID="COMMON">Item ID
</ BusinessComponentID>
<ApplicationObjectKey>
<ID schemeID="ITEM_ID" schemeAgencyID="SEBL_01">SiebelID</ ID>
<ApplicationObjectKey>
```

■ ApplicationFaultData / OrderFailureData / ProcessingNumber

Job ID - String type

■ ApplicationFaultData / OrderFailureData / ProcessingTypeCode

Common Value of the Processing Type Code

■ ApplicationFaultData / OrderFailureData / ProcessingQuantity

Processing Quantity as available in the EBM

■ ApplicationFaultData / OrderFailureData / FailureSystemCode

System where the fault occurred - 'AIA' in case the error is internal to the ABCS or BPEL.

Target System ID in case the fault is identified from the target application system

■ ApplicationFaultData / OrderFailureData / FailureSubSystemCode

The code of either the subsystem or the API, where the order has failed. This is applicable with participating applications. If the fault is within Oracle AIA, the service that faulted is assumed as the subsystem of failure

■ ApplicationFaultData / OrderFailureData / OrderLineItemFailureDataList

This is required if you are handling faults at the line-level or if the BPEL fails while it is trying to process a particular order line.

   – OrderLineItemID

Structure similar to OrderID

BusinessComponentID - SalesOrder / Provisioning Order / Fulfillment Order / Fulfillment Billing Order Line IDs (if any)

ID - SalesOrder Liner # / Provisioning Order Line # / Fulfillment Order Line # / Fulfillment Billing Order Line # (if available)

ApplicationObjectKey - If available (at the Siebel CRM end at least if the LineId is not yet cross-referenced)

AlternateObjectKey - SALESORDER_LINEID (COMMON)

Example J–5 is a sample definition.

Example J–5   Sample Definition 4

```
<BusinessComponentID>  Order Line ID   </BusinessComponentID>
<ID> Order Line# (if any)</ID>
<ApplicationObjectKey>
<ID schemeID="SALESORDER_LINEID"schemeAgencyID="SEBL_01">SalesOrderLineID</ID>
</ApplicationObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_LINEID " schemeAgencyID="COMMON">
SalesOrderLineCommonID
</ID>
 </AlternateObjectKey>
```

- – ErrorCode

  Error code associated with the failure

- – ErrorMessage

  Error message associated with the failure

- – ErrorSeverity

  Error severity associated with the failure

- – Status Context

  Status context of the order line

- – FailureSubSystemCode

  Code of the subsystem or API where the order line has failed. This is applicable with participating applications. If the fault is within Oracle AIA, the service that faulted is assumed to be the subsystem of failure.

J.3 Populating the Enriched Fault Message in Services without EBMs

In the Requestor ABCS Implementation services, populating the EBM_HEADER variable is typically the last step of this process and the chances of an error occurring (nonsystem fault error) is more likely during this last step.

For the nonpartner link faults or business faults, the application business connector service (ABCS) should follow the guidelines as stated in "Populating the Enriched Fault Message with Business Faults". The intention is to capture as many fields as possible here in this case. No common IDs can be available.

With system faults or composite faults, you can use the extension handler feature of the Oracle AIA Error Handling Framework to enrich the fault message.

As delivered, the system faults for the Siebel Requestor ABCS are handled by the Extension Handler - oracle.apps.aia.industry.comms.eh.AIAOrderFalloutErrorHandlerExtension.java to parse the Siebel order message and enrich the fault message (Fault/FaultNotification/FaultMessage/ApplicationFaultData) with the appropriate available data (OrderID and the AccountID).

For more information about extending error handling, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling and Extending Fault Messages.

K

Composite Application Validation System Changes

This appendix discusses how the Composite Application Validation System (CAVS) has changed from the Oracle Application Integration Architecture (Oracle AIA) Communications 11.1 release to the Oracle AIA Communications 11.2 release and provides details on Requestor application business connector services (ABCSs) and Provider ABCSs.

This appendix includes the following sections:

- Section K.1, "Configuration Properties for CAVS Enablement in 11.1"
- Section K.2, "Configuration Properties for CAVS Enablement in 11.2"

K.1 Configuration Properties for CAVS Enablement in 11.1

In the 11.1 Oracle AIA CAVS implementation, every service has a number of configuration properties.

For more information about these configuration properties, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

For the `RouteToCAVS` property, the out-of-the-box value is *False*. Oracle AIA provides a user interface (UI), which allows the user to toggle this property value between *True* and *False* for each service listed.

Figure K–1 is an example of the AIA Configuration UI. To navigate to this UI: log in to the AIA Console (http://<host>:<port>/AIA), go to **Setup** and then select the **AIA Configuration** tab.

Figure K–1   AIA Configuration Screen

K.2 Configuration Properties for CAVS Enablement in 11.2

CAVS enablement has been reorganized. As a result, the UI can no longer be used to toggle the value of the `RouteToCAVS` property for the Communications Order to Cash services, which are part of the 11.2 release.

The following instructions describe how to modify the configuration properties for Requestor ABCS and Provider ABCS to enable CAVS.

> **Note:** Any change in the System Configuration screen does not enable CAVS for a service. You must make changes manually in the Oracle AIA configuration file to make the service CAVS enabled.

K.2.1 Requestor ABCS

For CAVS enablement of Requestor ABCS, a single configuration property is maintained.

For example,
`EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address`

In order to enable CAVS, you must manually edit the AIAConfigurationProperties.xml file, which is located here: $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. Entries in the AIAConfigurationProperties.xml file are case sensitive.

To enable CAVS for Requestor ABCS:

1. Open the AIA configuration properties file.

2. Set the Address property to the CAVS URI for each service that you want to be CAVS enabled.

   For example, http://<soa_server>:<soa_port>/AIAValidationSystemServlet/asyncrequestrecipient

3. Save and close the file after you have set this property for all desired Requestor ABCSs.

4. Login to the AIA Console (http://<host>:<port>/AIA). Go to **Setup**, and then select the **AIA Configuration** tab. Click **Reload** to relaod the configuration file and make your changes effective.

K.2.2 Provider ABCS

For CAVS enablement of a Provider ABCS, two configuration properties are maintained. For example:

- ""Routing.SWI_spcOrder_spcUpsert.RouteToCAVS"

- ""Routing.SWI_spcOrder_spcUpsert.SEBL_01.EndpointURI"

In order to enable CAVS, you must manually edit the AIAConfigurationProperties.xml file, which is located here: $AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config. Entries in the AIAConfigurationProperties.xml file are case sensitive.

To enable CAVS for Provider ABCS:

1. Open the AIA configuration properties file.

2. Set the `RouteToCAVS` property value to *True* and set the EndpointURI property value to the actual CAVS URL for each service that you want to be CAVS enabled.

3. Save and close the file after you have set this property for all desired Provider ABCSs.

4. Login to the AIA Console (http://<host>:<port>/AIA). Go to **Setup**, and then select the **AIA Configuration** tab. Click **Reload** to relaod the configuration file and make your changes effective.

L

Reintroducing Enterprise Business Services

Enterprise business services (EBSs) are used to help route to multiple Providers. If you are using one source and one target system for your integration flows then EBSs are unnecessary. However, if you must dynamically identify a Provider system during runtime (content-based routing) then you should reintroduce EBSs.

> **Note:** With the deployment of the Fusion Middleware Foundation Pack, web service definition language (WSDL) files are provided for all EBSs.

To reintroduce EBSs:

1. Go to JDevloper and create a new composite for the EBS with an Oracle Mediator service. Use the EBS WSDL provided by Fusion Middleware Foundation Pack.

2. Create routing rules in Oracle Mediator to route to appropriate Provider connectors.

3. Save your changes.

4. Open the AIAConfigurationProperties.xml file, which is located here: *$AIA_ HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config*.

   Entries in the AIAConfigurationProperties.xml file are case sensitive.

5. For connectors that are to invoke the new EBS instead of directly invoking the Provider, replace the Provider connector's name and address with the name and address of the newly created EBS.

   This action tells the Requestor to invoke the EBS instead of directly invoking the Provider application business connector service (ABCS).

6. Save and close the file.

7. Login to the AIA Console (*http://<host>:<port>/AIA*). Go to **Setup**, and then select the **AIA Configuration** tab. Click **Reload** to relaod the configuration file and make your changes effective.