

Oracle® Application Integration Architecture

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

Release 11.3

E50337-01

December 2013

Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management, Release 11.3

E50337-01

Copyright © 2011, 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xv
Audience	xv
Downloading Oracle Documentation	xv
Documentation Accessibility	xv
Related Documents	xv
1 Overview of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM	
Overview of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM	1-1
About Leveraging Third-Party Applications	1-2
About the Test Orchestration Process	1-3
Process Integrations and Business Flows for Individual Integration Packs	1-3
About the Process Integrations	1-5
About Product Lifecycle Management	1-5
About Order Lifecycle Management	1-6
About Customer Management	1-6
About Order Fallout Management	1-7
Part I Understanding the Process Integrations	
2 Understanding the Process Integration for Product Lifecycle Management	
Overview of the Process Integration for Product Lifecycle Management	2-1
About the Synchronize Product and Price Business Flow	2-1
About the Query Product Classes Business Flow	2-1
3 Understanding the Synchronize Product and Price Business Flow	
Overview of the Synchronize Product and Price Business Flow	3-1
Synchronizing Billing Products and Billing Discounts in Real Time	3-1
Synchronizing Updates to Billing Products and Billing Discounts in Real Time	3-2
Synchronizing Billing Products and Billing Discounts in Batches	3-3
Synchronizing Updates to Billing Products and Billing Discounts in Batches	3-4
About Simple and Customizable Products	3-5
Synchronization of Billing Products with Pricing Details	3-6
Product Attributes	3-8

Effective Start and End Dates.....	3-8
Synchronization of Billing Discounts.....	3-8
Usage Charges on Products.....	3-10
About Price Lists and Rate Plans	3-11
Working with Price Lists and Rate Plans at Design Time.....	3-12
Associating Rate Plans in BRM with Siebel CRM Price Lists.....	3-12
Offering a Product in Multiple Currencies	3-14
Managing Pricing in Rate Plans and Price Lists	3-14
Changing the Price of a Product.....	3-15
Changing the Price List of a Product	3-15
Changing a Product from Multiple Price Lists to the Default Price List	3-16
About BRM Balance Groups.....	3-18
Understanding Product Bundling.....	3-18
Basic Entity Mappings.....	3-18
Defining Products and Discounts in BRM	3-19
Using Fixed Amounts versus Scaled Amounts in BRM.....	3-19
Physical Goods	3-20
Sales Catalogs	3-21
Recommendations for Product Definition in Siebel CRM	3-21
Recommendation for Discounts.....	3-22
Discounts Defined in Billing Systems	3-22
Discounts Defined in Siebel CRM Systems	3-22
Defining Overrides on the Product Definition	3-22
About Service Bundles	3-23
About Billing Service Types for Service Bundles	3-24
About Simple Service Bundles	3-24
Flagging Subscription Products as Simple Service Bundles.....	3-25
Example of Using a Service Bundle or a Simple Service Bundle	3-25
Assumptions and Constraints for Working with Simple Service Bundles	3-26
About Marketing Bundles.....	3-27
Credit Limits	3-28
About One-Time Charges for Service Activation and Changes to Promotions and Service Bundles 3-28	
Charging for Service Activation and Changes to Promotions and Services	3-28
Supporting Friends and Family	3-29
Enabling Friends and Family	3-30
Product Definition Methodology for Friends and Family: Example	3-31
About Time-Based Offerings.....	3-34
Setting Up Time-Based Offerings.....	3-34
Managing Expired Time-Based Offerings.....	3-36
Assumptions and Constraints for the Synchronize Product and Price Business Flow	3-36
4 Implementing the Synchronize Product and Price Business Flow	
Synchronize Product and Price Business Flow Overview	4-1
Product Synchronization Integration Flow	4-1
Billing Discount Synchronization Integration Flow	4-4
BRM Interfaces	4-5

Siebel CRM Interfaces.....	4-6
Industry Oracle AIA Components.....	4-6
Integration Services.....	4-7
SyncProductBRMCommsReqABCImpl.....	4-7
SyncDiscountBRMCommsReqABCImpl.....	4-7
SyncItemCompositionListSiebelCommsProvABCImpl.....	4-7
ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.....	4-8
5 Understanding the Query Product Classes Business Flow	
About Organizing Products.....	5-1
About the Query Product Classes Business Flow.....	5-2
Assumptions and Constraints for the Query Product Classes Business Flow.....	5-2
6 Implementing the Query Product Classes Business Flow	
Overview of the Query Product Classes Business Flow.....	6-1
Overview of Implementing the Query Product Classes Business Flow.....	6-3
Logical Data Model in Siebel CRM.....	6-5
Updating the Attribute Valueset.....	6-5
Supporting for Effectivity During Design-Time.....	6-6
Deciding on Effectivity and Cartridge Deployment.....	6-7
Query Product Classes Integration Flow.....	6-7
Siebel CRM Interfaces.....	6-9
Industry Oracle AIA Components.....	6-10
Integration Services.....	6-10
QueryProdClassAndAttributesSCEReqCommsABCImpl.....	6-11
QueryClassificationListSiebelCommsProvABCImpl.....	6-11
QuerySpecificationListSiebelCommsProvABCImpl.....	6-11
QuerySpecificationValueSetListSiebelCommsProvABCImpl.....	6-11
7 Understanding the Process Integration for Order Lifecycle Management	
Order Lifecycle Management Overview.....	7-1
Order Lifecycle Management Business Flows.....	7-3
Typical Topology.....	7-4
Order Capture Overview.....	7-5
About the Deliver Customer Order Subflow.....	7-6
About the Qualify Customer Order Subflow.....	7-8
Product Definition and Mapping Design Considerations.....	7-9
Data Requirements.....	7-11
8 Understanding the Process Sales Order Fulfillment Business Flow	
Process Sales Order Fulfillment Business Flow Overview.....	8-1
About Sales Orders.....	8-1
About Supplemental Orders.....	8-2
About Follow-On Orders.....	8-3
About Modify Orders.....	8-3

About Future-Dated Orders	8-3
Supporting Multiple Price Lists on Orders	8-4
Specifying Different Price Lists on New Orders	8-4
Changing Price Lists on Supplemental Orders	8-5
Changing Price Lists on Modify Orders	8-5
Supporting Order Priorities	8-6
Assumptions and Constraints for the Process Sales Order Fulfillment Business Flow	8-7

9 Implementing the Process Sales Order Fulfillment Business Flow

Process Sales Order Fulfillment Business Flow Overview	9-1
Submitting Orders from Siebel CRM to OSM Integration Flow	9-1
Defining Transaction Boundaries and Recovery Details	9-3
Supporting Order Priorities	9-3
Supporting Multiple Price Lists	9-5
Siebel CRM Interfaces	9-5
Industry Oracle AIA Components	9-6
Integration Services	9-6
ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer	9-6
ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl	9-6
ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer	9-7

10 Understanding the Synchronize Fulfillment Order Billing Account Business Flow

Overview of the Synchronize Fulfillment Order Billing Account Business Flow	10-1
Assumptions and Constraints for the Synchronize Fulfillment Order Billing Account Business Flow	10-7

11 Implementing the Synchronize Fulfillment Order Billing Account Business Flow

Overview of the Synchronize Fulfillment Order Billing Account Business Flow	11-1
Interfacing Orders to Create Customer Data in BRM Integration Flow	11-1
Defining Transaction Boundaries and Recovery Details	11-3
BRM Interfaces	11-4
Industry Oracle AIA Components	11-5
Integration Services	11-5
ProcessFulfillmentOrderBillingAccountListOSMCFSCCommsJMSConsumer	11-6
CommsProcessFulfillmentOrderBillingAccountListEBF	11-6
CommsProcessBillingAccountListEBF	11-7
ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSProducer	11-7

12 Understanding the Bill Fulfillment Order Business Flow

Bill Fulfillment Order Overview	12-1
About Interfacing Orders to BRM	12-2
Creating or Updating Service Instances	12-2
Communicating Pricing Information	12-2
Communicating One-time and Penalty Charges	12-2

Communicating Pricing or Discount Overrides.....	12-3
Communicating Price List Information.....	12-4
Communicating Service Identifiers.....	12-4
Communicating Siebel CRM Promotion Information.....	12-4
Rolling Back Transactions.....	12-5
Supporting Balance Groups.....	12-5
Disabling Service-Level Balance Groups.....	12-5
Working with Service-Level Balance Groups Enabled.....	12-6
How BRM Tracks and Bills New Services in Service-Level Balance Groups.....	12-6
How BRM Tracks Services in Nested Service Bundles in Service-Level Balance Groups.....	12-8
How BRM Tracks Service Bundles and Products Purchased on Change Orders in Service-Level Balance Groups.....	12-9
How BRM Tracks Account-Level Products in the Default Account-Level Balance Group.....	12-9
About Transferring Services.....	12-10
Working with Service-Level Balance Groups Disabled.....	12-11
Supporting Product Bundling.....	12-11
Example of Mapping for Bundled Products.....	12-12
Synchronizing Simple Service Bundles.....	12-13
Changing Purchased Simple Service Bundles.....	12-13
Simple Service Bundle Cross-References Entries.....	12-14
Supporting Single-Phase and Two-Phase Billing.....	12-14
Choosing Between Single-Phase and Two-Phase Billing.....	12-14
Single-Phase Billing.....	12-14
Two-Phase Billing.....	12-15
Using Single-Phase Billing or Two-Phase Billing.....	12-15
INITIATE BILLING Mode.....	12-15
FULFILL BILLING Mode.....	12-17
Assumptions and Constraints for Two-Phase Billing.....	12-18
Supporting Revisions.....	12-18
Assumptions and Constraints for Revisions.....	12-20
Supporting Time-Based Offerings on Orders.....	12-20
Supporting Time-Based Offerings on New Orders.....	12-20
Supporting Time-Based Offerings on Change Orders.....	12-21
Supporting Friends and Family Lists.....	12-21
Using Change Orders and Special Rating Products.....	12-22
Modifying Friends and Family List.....	12-22
Assumptions and Constraints for the Bill Fulfillment Order Business Flow.....	12-23

13 Implementing the Bill Fulfillment Order Business Flow

Bill Fulfillment Order Business Flow Overview.....	13-1
Interfacing Orders to Create Transaction Data in BRM Integration Flow.....	13-1
Defining Transaction Boundaries and Recovery Details.....	13-2
BRM Interfaces.....	13-3
Industry Oracle AIA Components.....	13-4
Integration Services.....	13-4

ProcessFulfillmentOrderBillingOSMCFSCCommsJMConsumer	13-5
ProcessFulfillmentOrderBillingBRMCommsProvABCImpl	13-5
ProcessFulfillmentOrderBillingBRMCommsAddSubProcess	13-7
ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess	13-8
ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess	13-9
ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess	13-11
ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess	13-12
ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer	13-13

14 Understanding the Provision Order and Update Fulfillment Order Business Flows

Overview of Order Provisioning	14-1
About Creating Provisioning Orders	14-1
About Updating Fulfillment Orders	14-2
Assumptions and Constraints for the Provision Order and Update Fulfillment Order Business Flows	14-2

15 Implementing the Provision Order and Update Fulfillment Order Business Flows

Provision Order and Update Fulfillment Order Business Flows Overview	15-1
OSM Fulfillment to OSM Provisioning Integration Flow	15-1
Defining Transaction Boundaries and Recovery Details	15-3
Industry Oracle AIA Components	15-3
Integration Services	15-4
ProcessProvisioningOrderOSMCFSCCommsJMConsumer	15-4
ProcessProvisioningOrderOSMPROVCommsJMSProducer	15-4
ProcessFulfillmentOrderUpdateOSMPROVCommsJMConsumer	15-5
ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer	15-5

16 Understanding the Update Sales Order Business Flow

Update Sales Order Overview	16-1
Timing Updates to Sales Order Data	16-1
Updating the Sales Order Status	16-2
About Creating or Updating Installed Assets in Siebel CRM	16-3

17 Implementing the Update Sales Order Business Flow

Overview of the Update Sales Order Business Flow	17-1
Updating Statuses from OSM to Siebel CRM Integration Flow	17-1
Defining Transaction Boundaries and Recovery Details	17-2
Siebel CRM Interfaces	17-3
Industry Oracle AIA Components	17-3
Integration Services	17-4
UpdateSalesOrderOSMCFSCCommsJMConsumer	17-4
UpdateSalesOrderSiebelCommsProvABCImpl	17-4

18 Understanding the Process Integration for Customer Management	
Customer Management Process Integration Overview	18-1
Assumptions and Constraints for the Process Integration for Customer Management	18-1
Data Requirements	18-2
About the Synchronize Customer Account Business Flow	18-2
About the Create/Sync Account Integration Flow	18-3
About the Update Customer Account Integration Flow	18-6
Account Status Synchronization Methodology	18-8
About the Synchronize Customer Special Rating Profile Business Flow	18-9
19 Implementing the Synchronize Customer Account Business Flow	
Overview of the Synchronize Customer Account Business Flow	19-1
Create/Sync Customer Account Integration Flow	19-1
Update Customer Account Integration Flow	19-1
BRM Interfaces	19-3
Siebel CRM Interfaces	19-4
Industry Oracle AIA Components	19-5
Integration Services	19-5
ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer	19-6
CommunicationsCustomerPartyEBSV2Resequencer	19-6
CommsProcessFulfillmentOrderBillingAccountListEBF	19-6
CommsProcessBillingAccountListEBF	19-7
SyncCustomerSiebelEventAggregator	19-7
SyncAccountSiebelAggregatorAdapter	19-8
SyncContactSiebelAggregatorAdapter	19-8
SyncAddressSiebelAggregatorAdapter	19-9
SyncBPSiebelAggregatorAdapter	19-9
SyncAcctSiebelAggrEventConsumer	19-9
SyncAccountSiebelReqABCImpl	19-9
CustomerPartyEBSV2	19-9
QueryCustomerPartyListSiebelProvABCImplV2	19-10
SyncCustomerPartyListBRMCommsProvABCImpl	19-10
SyncCustomerPartyListBRM_01CommsJMSConsumer	19-11
20 Implementing the Synchronize Customer Special Rating Profile Business Flow	
Overview of the Synchronize Customer Special Rating Profile Business Flow	20-1
Synchronize Friends and Family List Updates to BRM Integration Flow	20-1
Defining Transaction Boundaries and Recovery Details	20-2
BRM Interfaces	20-3
Siebel CRM Interfaces	20-3
Industry Oracle AIA Components	20-3
Integration Services	20-4
ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer	20-4
ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl	20-4
ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl	20-4

21	Understanding the Process Integration for Order Fallout Management	
	Overview of the Process Integration for Order Fallout Management	21-1
	About Order Fallout Detection	21-2
	About Order Fallout Notification	21-3
	About Order Correction	21-5
	How Oracle AIA Error Handling Framework Captures Faults	21-7
	Order Fallout Management Process Integration Business Flows	21-11
	Create Trouble Ticket from Oracle AIA Business Flow	21-11
	Assumptions and Constraints for the Create Trouble Ticket from Oracle AIA Business Flow	21-13
	Create and Manage Trouble Ticket from OSM Business Flow	21-13
	Assumptions and Constraints for the Create and Manage Trouble Ticket from OSM Business Flow	21-14
	Extending Fault Messages to Capture Order Fallout Information	21-14
	Exception Handling	21-18
22	Implementing the Create Trouble Ticket from Oracle AIA Business Flow	
	Create Trouble Ticket from Oracle AIA Business Flow Overview	22-1
	Creating a Trouble Ticket in Siebel CRM Integration Flow	22-1
	Defining Transaction Boundaries and Recovery Details	22-3
	Exception Handling	22-3
	Siebel CRM Interfaces	22-4
	Industry Oracle AIA Components	22-4
	Integration Services	22-4
	CreateTroubleTicketSiebelCommsProvABCServiceImpl.....	22-5
	AIAOrderFalloutJMSBridgeService	22-5
	AIACOMOrderFalloutNotificationJMSConsumer	22-5
	CreateTroubleTicketAIACommsReqImpl.....	22-5
	AIAOrderFalloutErrorHandlerExtension - Java Class	22-5
	Business Flow Fallout-Enabled Services	22-6
23	Implementing the Create and Manage Trouble Ticket from OSM Business Flow	
	Overview of the Create and Manage Trouble Tickets from OSM Business Flow	23-1
	Order Failure Notification to OSM Integration Flow	23-1
	Creating a Trouble Ticket in Siebel CRM from OSM Integration Flow	23-3
	Defining Transaction Boundaries and Recovery Details	23-5
	Updating a Trouble Ticket in Siebel CRM from OSM Integration Flow	23-6
	Defining Transaction Boundaries and Recovery Details	23-7
	Siebel CRM Interfaces	23-8
	Industry Oracle AIA Components	23-8
	Integration Services	23-9
	UpdateTroubleTicketSiebelCommsProvABCServiceImpl	23-9
	CreateOrderFalloutNotificationOSMCFSCommsProvImpl.....	23-9
	CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer	23-10
	CreateOrderFalloutNotificationOSMCFSCommsJMSProducer	23-10

CreateTroubleTicketOSMCFSCommsJMConsumer.....	23-10
CreateTroubleTicketRespOSMCFSCommsJMProducer.....	23-10
UpdateTroubleTicketOSMCFSCommsJMConsumer.....	23-10
CreateFaultNotificationLFCommsJMConsumer.....	23-10
Business Flow Fallout-Enabled Services.....	23-11

Part II Configuring the Process Integrations

24 Configuring the Process Integration for Product Lifecycle Management

Prerequisites for Integrated Product Lifecycle Management.....	24-1
Configuring BRM for Integrated Product Lifecycle Management.....	24-2
Configuring Siebel CRM for Integrated Product Lifecycle Management.....	24-3
Working with DVMs for Product Lifecycle Management.....	24-4
Working with Cross-References for Product Lifecycle Management.....	24-5
Handling Error Notifications for Product Lifecycle Management.....	24-5
Describing Delivered Error Notification Roles and Users.....	24-6
Configuring Properties for Product Lifecycle Management.....	24-6
Configuring Properties for the Query Product Classes Business Flow.....	24-13

25 Configuring the Process Integration for Order Lifecycle Management

Configuring BRM for Integrated Order Lifecycle Management.....	25-1
Configuring Siebel CRM for Integrated Order Lifecycle Management.....	25-1
Working with Domain Value Maps for Order Lifecycle Management.....	25-2
Working with the PRICELIST DVM.....	25-3
Updating the PRICELIST DVM.....	25-3
Migrating PRICELIST DVM Between Environments.....	25-4
Working with Cross-References for Order Lifecycle Management.....	25-4
Handling Error Notifications for Order Lifecycle Management.....	25-6
Describing Delivered Error Notification Roles and Users.....	25-7
Configuring the Process Integration for Order Lifecycle Management.....	25-7
Setting AIA Configuration Properties.....	25-7
Setting the BRM Version Number for Backward Price List Compatibility.....	25-17

26 Configuring the Process Integration for Customer Management

Configuring BRM for Integrated Customer Management.....	26-1
Configuring Siebel CRM for Integrated Customer Management.....	26-1
Working with Domain Value Maps for Customer Management.....	26-2
Working with Cross-References for Customer Management.....	26-3
Handling Error Notifications.....	26-5
Describing Delivered Error Notification Roles and Users.....	26-5
Order Fallout Management.....	26-6
Configuring the Process Integration for Customer Management.....	26-6

27 Configuring the Process Integration for Order Fallout Management

Configuring Oracle AIA for Order Fallout Management.....	27-1
--	------

Configuring Siebel CRM for Integrated Order Fallout Management	27-1
Working with Domain Value Maps for Order Fallout Management	27-2
Working with Cross-References for Order Fallout Management	27-3
Handling Error Notifications for Order Fallout Management	27-3
Describing Delivered Error Notification Roles and Users	27-3
Using Error Type to Control Response to Order Fallout	27-4
Configuring the Process Integration for Order Fallout Services	27-6

Part III Appendixes

A Cross-References for the Process Integration for Product Lifecycle Management

Cross-References for the Process Integration for Product Lifecycle Management	A-1
Cross-Reference Values	A-1
Product Synchronization Flow	A-2
Example of Simple Product Synchronization	A-4
Example of Complex Product Synchronization	A-6
Discount Synchronization Flow	A-8
Example of Discount Synchronization	A-9

B Communications Orders Dictionary

Order Header Component Attributes	B-1
Order Line Component Attributes	B-9

C Mapping Billing Dates

How Dates are Set in BRM	C-1
--------------------------------	-----

D Supporting MACD Actions and Attribute Changes

MACD Line Actions Supported	D-1
MACD Line Actions Supported for Service Bundle Components	D-2
MACD Line Actions Supported for Account-Level Products	D-2
Communicating Product Attribute Changes to BRM	D-3
Communicating Changes Made to Attributes of Marketing Bundles	D-3
Communicating Changes Made to Attributes of Service Bundles	D-4
Communicating Changes Made to Attributes of Service Bundle Components	D-4
Communicating Changes Made to Attributes of Service-Level Billing Subscription Products	D-4
Communicating Changes Made to Attributes of Service-Level Billing Discount Products	D-5
Communicating Changes Made to Attributes of Service-Level Billing Item Products	D-5
Communicating Changes Made to Attributes of Service-Level Special Rating Products	D-5
Communicating Changes Made to Attributes of Account-Level Products	D-6
Communicating Changes Made to Attributes of Account-Level Billing Subscription Products	D-6
Communicating Changes Made to Attributes of Account-Level Billing Discounts	D-6

E Examples of Changing the Paying Parent on Subordinate Accounts

About the Examples	E-1
Examples when Service-Level Balance Groups Are Enabled	E-2
Changing the Paying Parent for a Subordinate Account with Separate Billing Profiles.....	E-2
Changing the Paying Parent for a Subordinate Account with a Single Billing Profile.....	E-3
Changing the Paying Parent for Multiple Subordinate Accounts	E-4
Examples when Service-Level Balance Groups Are Disabled	E-6
Changing the Paying Parent for a Subordinate Account	E-6
Changing the Paying Parent for Multiple Subordinate Accounts with Separate Billing Profiles..	E-7
Problems When Integrating Separate Billing Profiles on the Same Account.....	E-8

F Configuring Multiple BRM Instances for Communications Integrations

Understanding System Codes in Oracle AIA	F-1
Configuring Multiple BRM Instances - General Steps	F-2
Creating a New Consumer for Product Synchronization	F-7
Creating a New Consumer for Discount Synchronization	F-15
Creating a New Consumer for Customer Updates.....	F-23
Creating a Data Source and Connection Factory	F-32
Creating Logical Instances in Oracle AIA	F-33
Creating Service Bundles in Siebel CRM	F-34
Merging Logical BRM Instances into a Single BRM Instance	F-35

G Changing the BRM Instance

Changing the BRM Instance	G-1
---------------------------------	-----

H Expectations from an Order Management System for Billing Integration

I Using the Oracle Mediator Resequencer Feature

Queues and Flows Enabled for Sequencing	I-1
Resolving Errors in Flows with Resequencer.....	I-3

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

Populating Sender Context Information in the EBM Header	J-1
Populating the Enriched Fault Message with Business Faults.....	J-2
Populating the Enriched Fault Message in Services without EBMs	J-5

K Composite Application Validation System Changes

Configuration Properties for CAVS Enablement in 11.1	K-1
Configuration Properties for CAVS Enablement in 11.2	K-1
Requestor ABCS	K-2
Provider ABCS.....	K-2

L Reintroducing Enterprise Business Services

Reintroducing Enterprise Business Services	L-1
--	-----

M Understanding Multischema Migration

About Multischema Migration	M-1
Mapping BRM POIDs to AIA Cross-References	M-2
Synchronizing Cross-References for Migrated Accounts	M-2
Configuring Oracle AIA and BRM to Work Together in Multischema Environments	M-4
Configuring Oracle AIA to Work with BRM in Multischema Environments	M-4
Configuring BRM to Work with Oracle AIA in Multischema Environments	M-5
Handling Errors.....	M-5
Assumptions and Constraints for Multischema Migration	M-5
Integration Services.....	M-6
About the SyncMultiSchemaChangeBRMAQ Consumer.....	M-6
About the SyncMultiSchemaChangeBRMRequester Service.....	M-6
About the MultiSchemaUpdate.sql Process.....	M-6

Preface

This document describes how to implement and use the Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management.

Audience

This document is intended for customer service representatives, billing and pricing administrators, and other individuals who are responsible for configuring, managing and maintaining Oracle AIA Communications Pre-Built Integrations.

Downloading Oracle Documentation

Product documentation is located on Oracle Technology Network:

<http://docs.oracle.com>

Additional documentation is available from the Oracle software delivery Web site:

<http://edelivery.oracle.com>

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.

Related Documents

My Oracle Support Information Centers provide the most recent information about the following:

- Product guides
- Alerts

- Troubleshooting details
- FAQs
- Patches
- Community links

To see the Information Center for the Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management, see My Oracle Support note 1392638.2 at:

<http://support.oracle.com/epmos/faces/DocumentDisplay?id=1392638.2>

For more information about Oracle AIA concepts, see *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*.

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

Overview of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM

This chapter provides an overview of the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration). It describes the integration architecture, the process integrations, and the pre-built integration options.

Overview of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM

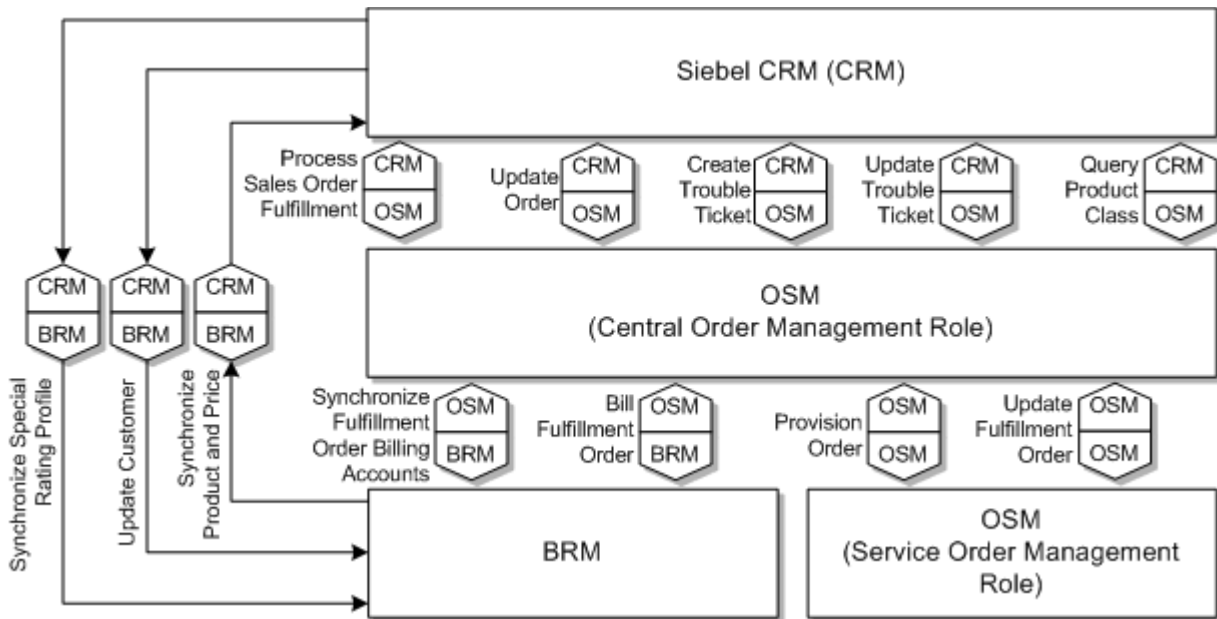
The integration provides integration and deployment accelerators that build on industry best practices and a comprehensive integration methodology. The integration automates business flows for Business Support Systems (BSS) concept to launch and BSS order to activate processes across Siebel CRM, OSM, and BRM.

When you install the integration, you choose from the following Pre-Built Integration options:

- Oracle Communications Order to Cash Siebel CRM, OSM, and BRM Pre-Built Integration option
- Oracle Communications Order to Cash Siebel CRM and OSM Pre-Built Integration option (assumes integration with a billing system other than BRM)
- Oracle Communications Order to Cash Siebel CRM and BRM Pre-Built Integration option (assumes integration with a central order management system other than OSM)

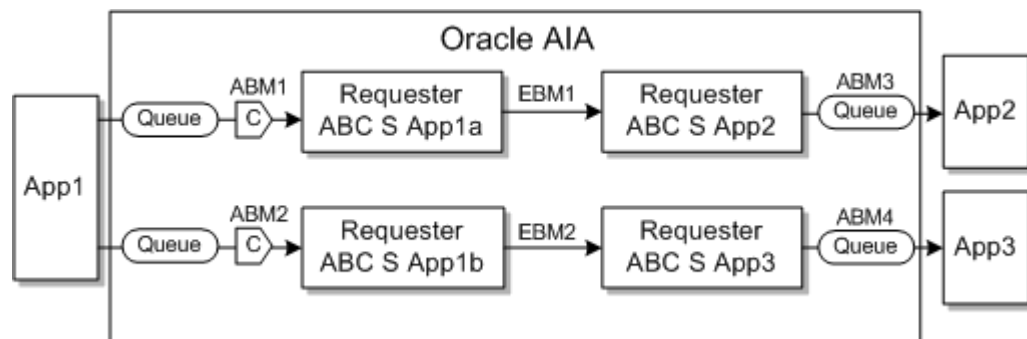
[Figure 1-1](#) illustrates how the components of the pre-built integration options enable the Order to Cash business flows.

Figure 1–1 Order to Cash Functional Overview



The components of the Pre-Built Integration options enable the business flows using a service-oriented architecture (SOA) that translates a request from an application like Siebel CRM, OSM, or BRM into an enterprise business message (EBM) payload which is translated into an application business message (ABM) specific to a second application. [Figure 1–2](#) illustrates this integration architecture.

Figure 1–2 Integration Patterns Applicable to Oracle Comms Order to Cash



Siebel CRM, OSM, and BRM participate as providers or requesters in the Order to Cash processes. Each of the integration options package the integration artifacts between the Siebel CRM, OSM, or BRM and Oracle AIA up to and including the application business connector services (ABCS) for the Siebel CRM, OSM, or BRM.

The integration is built on top of the Oracle AIA Foundation Pack. You can extend the delivered process integrations and build new ones by leveraging the pre-defined enterprise business objects specifically tailored for the communications industry contained within the Oracle AIA Foundation Pack Extension for Communications.

About Leveraging Third-Party Applications

Deploying the Siebel CRM, OSM, and BRM Pre-Built Integration option provides accelerated integration between all three applications, but you can also leverage third

party applications based on the overall integration architecture. You can deploy the Siebel CRM and OSM Pre-Built Integration option if your deployment does not include BRM or the Siebel CRM and BRM Pre-Built Integration option if your deployment does not include OSM.

To leverage a third party application within the context of the integration architecture, you must construct specific ABCs which conform to the design specified and which allow your third-party applications to fulfill the roles and responsibilities specified by the design.

Caution: This guide provides an overview of the design and implementation instructions for the process integrations available for Siebel CRM, OSM, and BRM. However, if your deployment has only one or two of the three applications, your systems and connectors must mimic what is outlined in this guide to achieve the same functionality.

About the Test Orchestration Process

If you have deployed the Siebel CRM and BRM Pre-Built Integration option and use your own order management system, a Test Orchestration Process (TOP) is shipped to sanity test the ready-to-use order flow. You must replace the TOP with your order management system.

See the discussion of replacing the test order orchestration with your order management system in Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations for more information.

Process Integrations and Business Flows for Individual Integration Packs

This section describes how the Order to Cash business flows are supported by each Pre-Built Integration option, and shows the process integration to which each business flow belongs. The Siebel CRM, OSM, and BRM Pre-Built Integration option supports the business flows listed in [Table 1-1](#) and [Table 1-2](#) in addition to those listed in [Table 1-3](#).

[Table 1-1](#) lists the business flows that the Siebel CRM and OSM Pre-Built Integration option supports.

Table 1-1 Business Flows Supported by the Siebel CRM and OSM Pre-Built Integration Option

Process Integration	Business Flow
Product Lifecycle Management	<ul style="list-style-type: none"> ▪ Query Product Classes: Oracle Communications Design Studio sends a query to Siebel CRM and Siebel CRM responds to Design Studio. See "Understanding the Query Product Classes Business Flow" for more information.

Table 1–1 (Cont.) Business Flows Supported by the Siebel CRM and OSM Pre-Built Integration Option

Process Integration	Business Flow
Order Management	<ul style="list-style-type: none"> ■ Process Sales Order Fulfillment: Siebel CRM sends a request to OSM in the central order management role (OSM COM). See "Understanding the Process Sales Order Fulfillment Business Flow" for more information. ■ Update Sales Order: OSM COM sends an update request to Siebel CRM. See "Understanding the Update Sales Order Business Flow" for more information. ■ Provision Order: OSM COM sends provisioning requests to OSM in the Service Order Management role (OSM SOM). ■ Update Fulfillment Order: OSM SOM sends order updates to OSM COM.
Order Fallout Management	<ul style="list-style-type: none"> ■ Create and Update Trouble Tickets: OSM COM sends a request to create or update a trouble ticket to Siebel CRM. See "Understanding the Process Integration for Order Fallout Management" for more information.

Table 1–2 describes lists the business flows that the Siebel CRM and BRM Pre-Built Integration option supports.

Table 1–2 Business Flows Supported by the Siebel CRM and BRM Pre-Built Integration Option

Process Integration	Business Flow
Product Lifecycle Management	<ul style="list-style-type: none"> ■ Synchronize Product and Price: BRM sends a synchronization request to Siebel CRM. See "Understanding the Synchronize Product and Price Business Flow" for more information.
Customer Management	<ul style="list-style-type: none"> ■ Synchronize Customer Account: Siebel CRM sends customer account updates to BRM. See "Understanding the Process Integration for Customer Management" for more information. ■ Synchronize Customer Special Rating Profile: Siebel CRM sends special rating profile updates to BRM. See "Understanding the Process Integration for Customer Management" for more information.
Order Fallout Management	<ul style="list-style-type: none"> ■ Create Trouble Tickets: BRM sends a request to create a trouble ticket to Siebel CRM. See "Understanding the Process Integration for Order Fallout Management" for more information.

Table 1–3 lists the business flows that the Siebel CRM, OSM, and BRM Pre-Built Integration option supports in addition to those listed in Table 1–1 and Table 1–2.

Table 1–3 Additional Business Flows Supported by the Siebel CRM, OSM, and BRM Pre-Built Integration Option

Process Integration	Business Flow
Order Management	<ul style="list-style-type: none"> ■ Synchronize Fulfillment Order Billing Account: OSM sends a request to enrich an order with customer account details from Siebel CRM and synchronize the account details in BRM. See "Understanding the Synchronize Fulfillment Order Billing Account Business Flow" for more information. ■ Bill Fulfillment Order: OSM sends order data to BRM for fulfillment. See "Understanding the Bill Fulfillment Order Business Flow" for more information.

About the Process Integrations

The integration provides the following process integrations:

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

About Product Lifecycle Management

The process integration for product lifecycle management lets you:

- Create and update products and discounts in BRM and use the integration to synchronize the products and discounts in Siebel CRM
- Create and update product classes in Siebel CRM and use the integration to import them into Design Studio. Design Studio maps the product classes as product specifications.

Note: Deployments using Oracle Product Hub have a different process for importing product classes. See *Oracle Application Integration Architecture Oracle Product Master Data Management Integration Implementation Guide* for more information.

[Table 1–4](#) lists the business flows for product lifecycle management and the Pre-Built Integration options that enable them.

Table 1–4 Product Lifecycle Management Business Flows

Business Flow	Pre-Built Integration Options Enabling the Flow
Synchronize Product and Price See " Understanding the Synchronize Product and Price Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM and BRM ■ Siebel CRM, OSM, and BRM
Query Product Classes See " Understanding the Query Product Classes Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM

About Order Lifecycle Management

The process integration for order lifecycle management lets you submit orders from Siebel CRM to OSM for order fulfillment in BRM.

OSM uses the services provided by this integration to enable the following business flows:

- **Synchronize Fulfillment Order Billing Account:** OSM decomposes orders to create customer data in BRM.
- **Bill Fulfillment Order:** OSM decomposes orders to create transaction data in BRM.
- **Synchronize Provisioning Order:** OSM in the Central Order Management (COM) role decomposes and sends orders to OSM in the Service Order Management (SOM) role for provisioning.
- **Update Fulfillment Order:** OSM in the SOM role sends provisioning status updates to OSM in the COM role.
- **Update Sales Order:** OSM sends order updates to Siebel CRM.

Table 1–5 lists the business flows for order lifecycle management and the Pre-Built Integration options that enable them.

Table 1–5 Order Management Business Flows

Business Flow	Pre-Built Integration Options Enabling the Flow
Process Sales Order Fulfillment See " Understanding the Process Sales Order Fulfillment Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM
Update Sales Order See " Understanding the Update Sales Order Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM
Synchronize Fulfillment Order Billing Account See " Understanding the Synchronize Fulfillment Order Billing Account Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM, OSM, and BRM
Bill Fulfillment Order See " Understanding the Bill Fulfillment Order Business Flow ".	<ul style="list-style-type: none"> ■ Siebel CRM, OSM, and BRM
Provision Order See " Understanding the Provision Order and Update Fulfillment Order Business Flows ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM
Update Fulfillment Order See " Understanding the Provision Order and Update Fulfillment Order Business Flows ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM

About Customer Management

The process integration for customer management synchronizes customer information from Siebel CRM to BRM. You define customer accounts in Siebel CRM and the integration synchronizes these accounts to BRM as part of the order fulfillment process. After synchronizing an account to BRM, the process integration continues to synchronize any changes to the account from Siebel CRM to BRM.

[Table 1–6](#) lists the business flows for customer management and the Pre-Built Integration options that enable them.

Table 1–6 Customer Management Business Flows

Business Flow	Pre-Built Integration Options Enabling the Flow
Synchronize Customer Account See " Understanding the Process Integration for Customer Management ".	<ul style="list-style-type: none"> ■ Siebel CRM and BRM ■ Siebel CRM, OSM, and BRM
Synchronize Customer Special Rating Profile See " Understanding the Process Integration for Customer Management ".	<ul style="list-style-type: none"> ■ Siebel CRM and BRM ■ Siebel CRM, OSM, and BRM

About Order Fallout Management

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

[Table 1–7](#) lists the business flows for order fallout management and the Pre-Built Integration options that enable them.

Table 1–7 Order Fallout Management Business Flows

Business Flow	Pre-Built Integration Options Enabling the Flow
Create and Update Trouble Ticket from OSM See " Understanding the Process Integration for Order Fallout Management ".	<ul style="list-style-type: none"> ■ Siebel CRM and OSM ■ Siebel CRM, OSM, and BRM
Create Trouble Ticket from Oracle AIA See " Understanding the Process Integration for Order Fallout Management ".	<ul style="list-style-type: none"> ■ Siebel CRM and BRM ■ Siebel CRM, OSM, and BRM

Part I

Understanding the Process Integrations

Part I contains the following chapters:

- Understanding the Process Integration for Product Lifecycle Management
- Understanding the Synchronize Product and Price Business Flow
- Implementing the Synchronize Product and Price Business Flow
- Understanding the Query Product Classes Business Flow
- Implementing the Query Product Classes Business Flow
- Understanding the Process Integration for Order Lifecycle Management
- Understanding the Process Sales Order Fulfillment Business Flow
- Implementing the Process Sales Order Fulfillment Business Flow
- Understanding the Synchronize Fulfillment Order Billing Account Business Flow
- Implementing the Synchronize Fulfillment Order Billing Account Business Flow
- Understanding the Bill Fulfillment Order Business Flow
- Implementing the Bill Fulfillment Order Business Flow
- Understanding the Provision Order and Update Fulfillment Order Business Flows
- Understanding the Update Sales Order Business Flow
- Implementing the Provision Order and Update Fulfillment Order Business Flows
- Implementing the Update Sales Order Business Flow
- Understanding the Process Integration for Customer Management
- Implementing the Synchronize Customer Account Business Flow
- Implementing the Synchronize Customer Special Rating Profile Business Flow
- Understanding the Process Integration for Order Fallout Management
- Implementing the Create Trouble Ticket from Oracle AIA Business Flow
- Implementing the Create and Manage Trouble Ticket from OSM Business Flow

Understanding the Process Integration for Product Lifecycle Management

This chapter describes the process integration for Product Lifecycle Management (PLM).

Overview of the Process Integration for Product Lifecycle Management

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

For this process integration, you use BRM to create and update billing products and billing discounts and synchronize them to Siebel CRM for enrichment.

The process integration for PLM delivers the following business flows:

- Synchronize Product and Price
- Query Product Classes

About the Synchronize Product and Price Business Flow

The Synchronize Product and Price business flow lets you create new billing products and billing discounts in BRM and synchronize them to Siebel CRM. It also lets you update billing products and billing discounts in BRM and resynchronize them to Siebel CRM.

This business flow is enabled using the Siebel CRM and BRM Pre-Built Integration option or the Siebel CRM, OSM, and BRM Pre-Built Integration option.

See "[Understanding the Synchronize Product and Price Business Flow](#)" for more information about the Synchronize Product and Price business flow.

About the Query Product Classes Business Flow

The Query Product Classes business flow lets you create new product classes in Siebel CRM and query them from Oracle Communications Design Studio to create product specifications. The query process includes product classes, associated attributes and valuesets. The business flow also lets you update product classes in Siebel CRM and query the updates from Design Studio to update the product specifications.

This business flow is enabled using the Siebel CRM and Oracle Communications Order and Service Management (OSM) Pre-Built Integration option or the Siebel CRM, OSM, and BRM Pre-Built Integration option.

If you are using Oracle Product Hub as your product master, this flow can be enabled using the Pre-Built Integrations for Oracle Product Master Data Management in addition to those listed above. See *Oracle Application Integration Architecture Oracle Product Master Data Management Integration Implementation Guide* for more information about the integration between Product Hub and Design Studio.

See "[Understanding the Query Product Classes Business Flow](#)" for more information about the Query Product Classes business flow.

Understanding the Synchronize Product and Price Business Flow

This chapter provides an overview of the Synchronize Product and Price business flow and describes the concepts from Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM) that are related to the business flow. It also lists the assumptions and constraints for the business flow.

Overview of the Synchronize Product and Price Business Flow

This section describes process of synchronizing billing products and billing discounts in real-time and batches and synchronizing the updates to these billing products and billing discounts. as part of billing product and discount synchronization, synchronization of billing products with pricing details, and synchronization of billing discounts.

Synchronizing Billing Products and Billing Discounts in Real Time

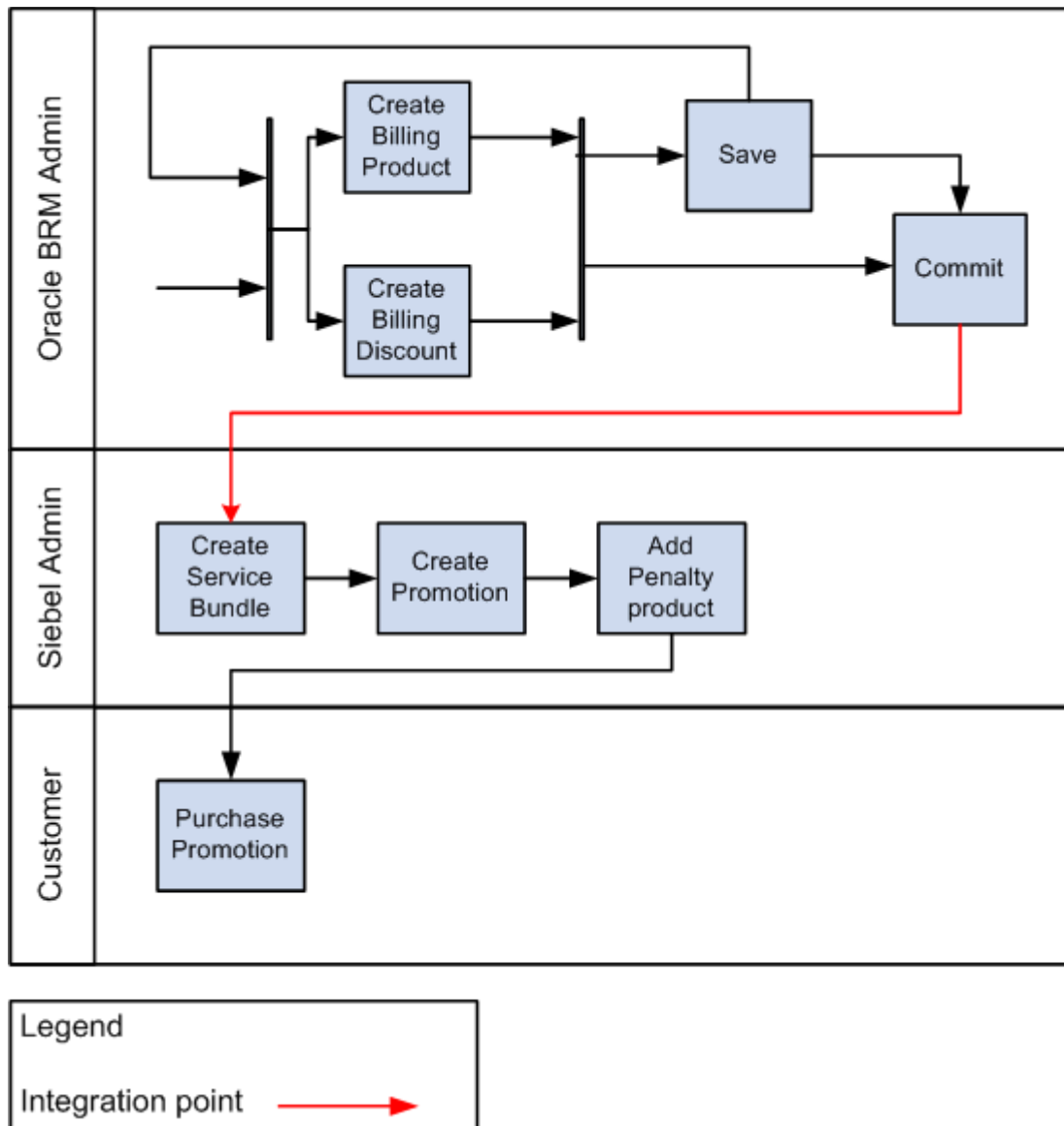
In this flow, the BRM product administrator creates billing products and billing discounts in Pricing Center. The BRM product administrator can either commit single billing products or discounts to the BRM database, or save sets of billing products and discounts to a file and commit the entire file to the BRM database.

When products and discounts are committed to the BRM database the Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle Communications Order and Service Management, and BRM (the integration) instantaneously synchronizes them to Siebel CRM. The Siebel CRM product administrator uses the synchronized billing products to create service bundles or promotions. The Siebel CRM product administrator can also add charges and penalties to the promotion.

See "[Understanding Product Bundling](#)" for more information about service bundles and promotions.

[Figure 3–1](#) shows how billing products and billing discounts are created in BRM, synchronized to Siebel CRM in real time, and bundled in Siebel CRM for customers to purchase in promotions.

Figure 3–1 Synchronizing Billing Products and Billing Discounts in Real Time



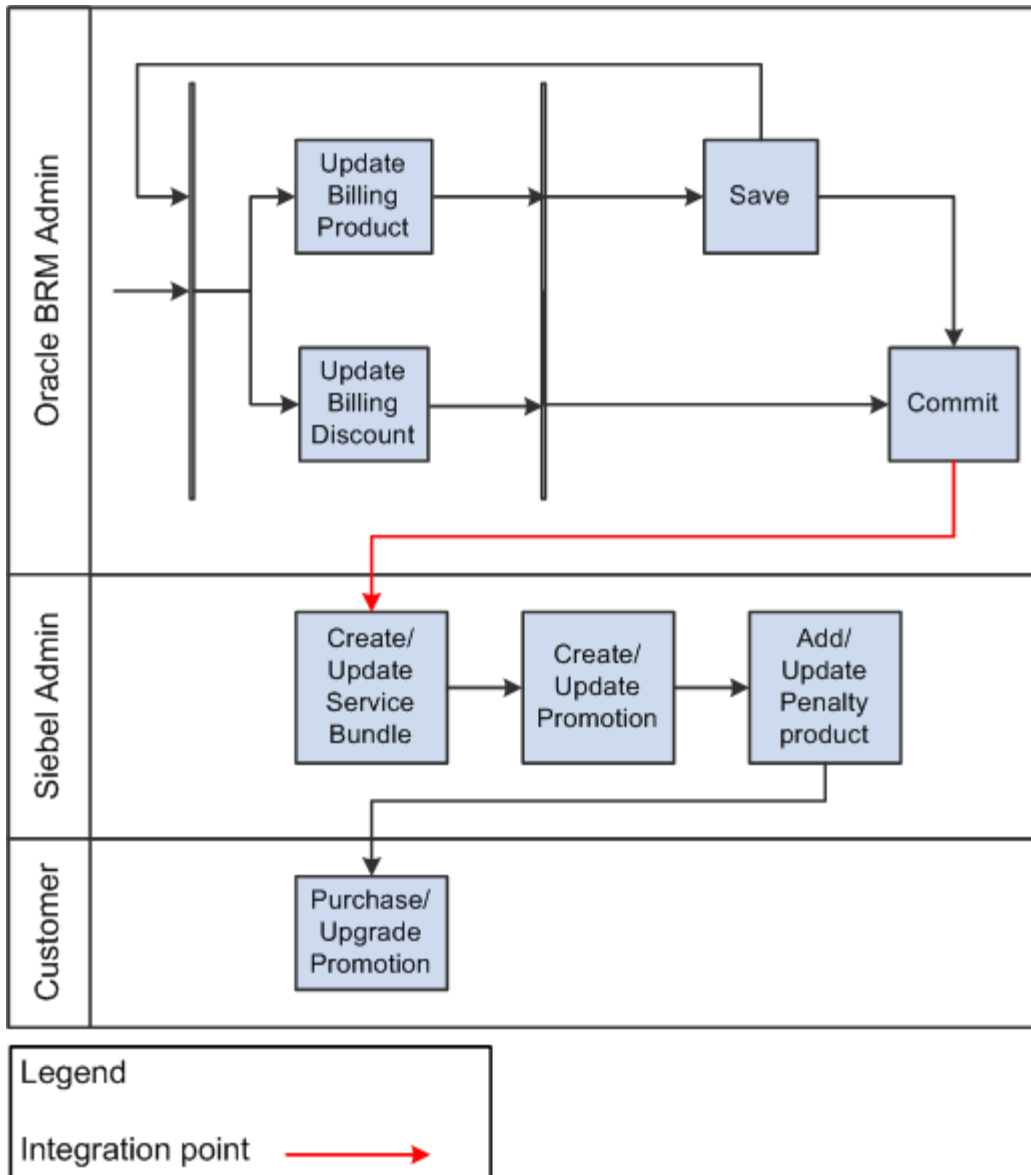
Synchronizing Updates to Billing Products and Billing Discounts in Real Time

In this flow, the BRM product administrator updates the attributes of billing products and billing discounts in Pricing Center and commit them to the BRM database singly or as sets in a file.

When the updated billing products or discounts are committed to the BRM database, the integration 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 can make any necessary changes to promotions and bundles.

Figure 3–2 shows how billing products and billing discount are updated in BRM, synchronized to Siebel CRM in real time, and updated in Siebel CRM bundles and promotions for customers to upgrade their promotions.

Figure 3–2 Synchronizing Updates to Billing Products and Billing Discounts in Real Time



Synchronizing Billing Products and Billing Discounts in Batches

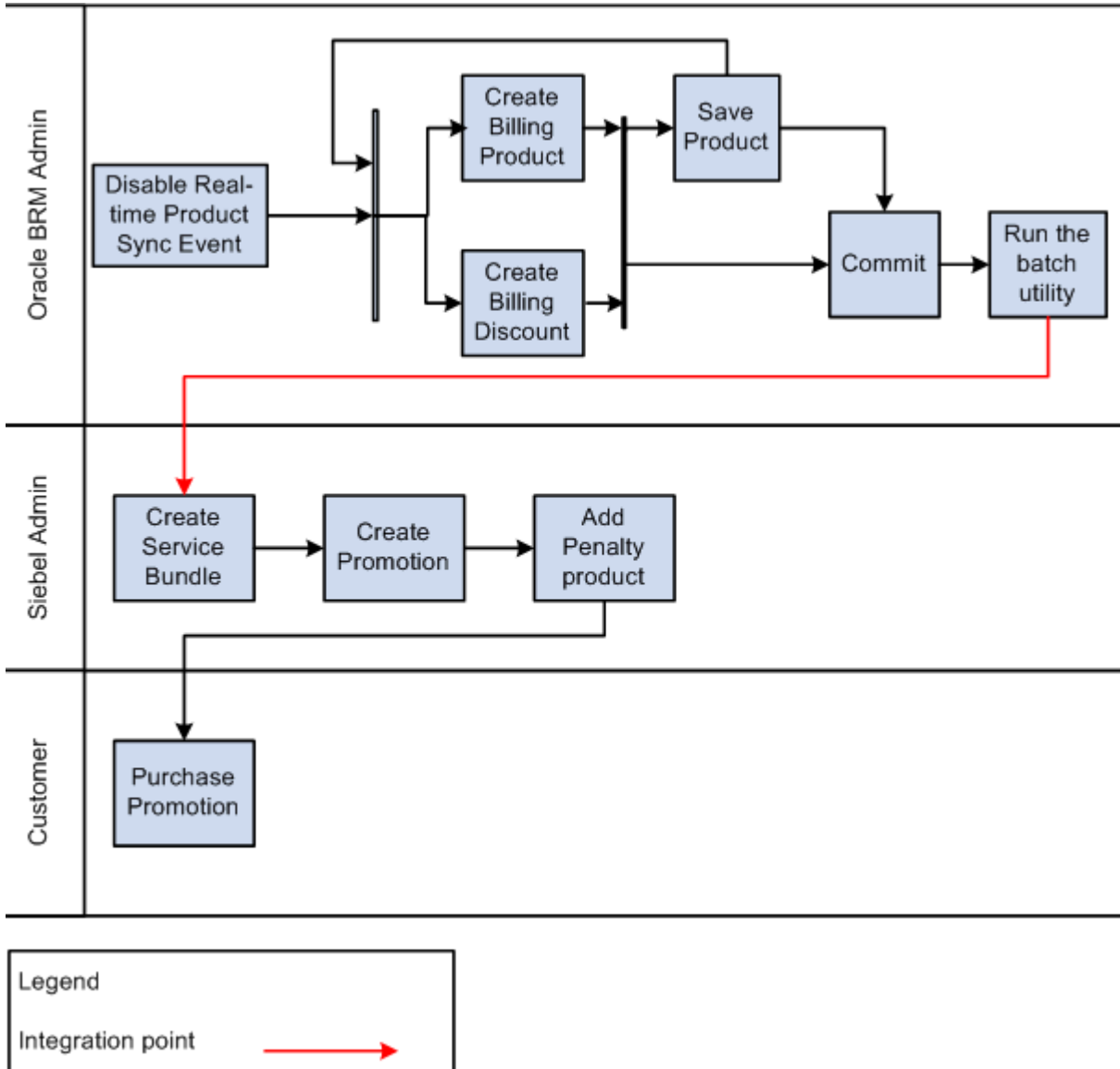
In this flow, the BRM product administrator disables the event for real-time product synchronization, and then creates billing products and billing discounts in Pricing Center. The products can be saved in sets to a file. The BRM product administrator runs a batch utility to store the products singly or from the file in the BRM database and synchronize the products with Siebel CRM.

The Siebel CRM product administrator uses the 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.

See "[Understanding Product Bundling](#)" for more information about service bundles and promotions.

Figure 3–3 shows how billing products and billing discounts are created in BRM, synchronized to Siebel CRM in batches, and bundled in Siebel CRM for customers to purchase in promotions.

Figure 3–3 Synchronizing Billing Products and Billing Discounts in Batches



See *Oracle Communications Billing and Revenue Management Synchronization Queue Manager* for information about how to disable the event for real-time product synchronization.

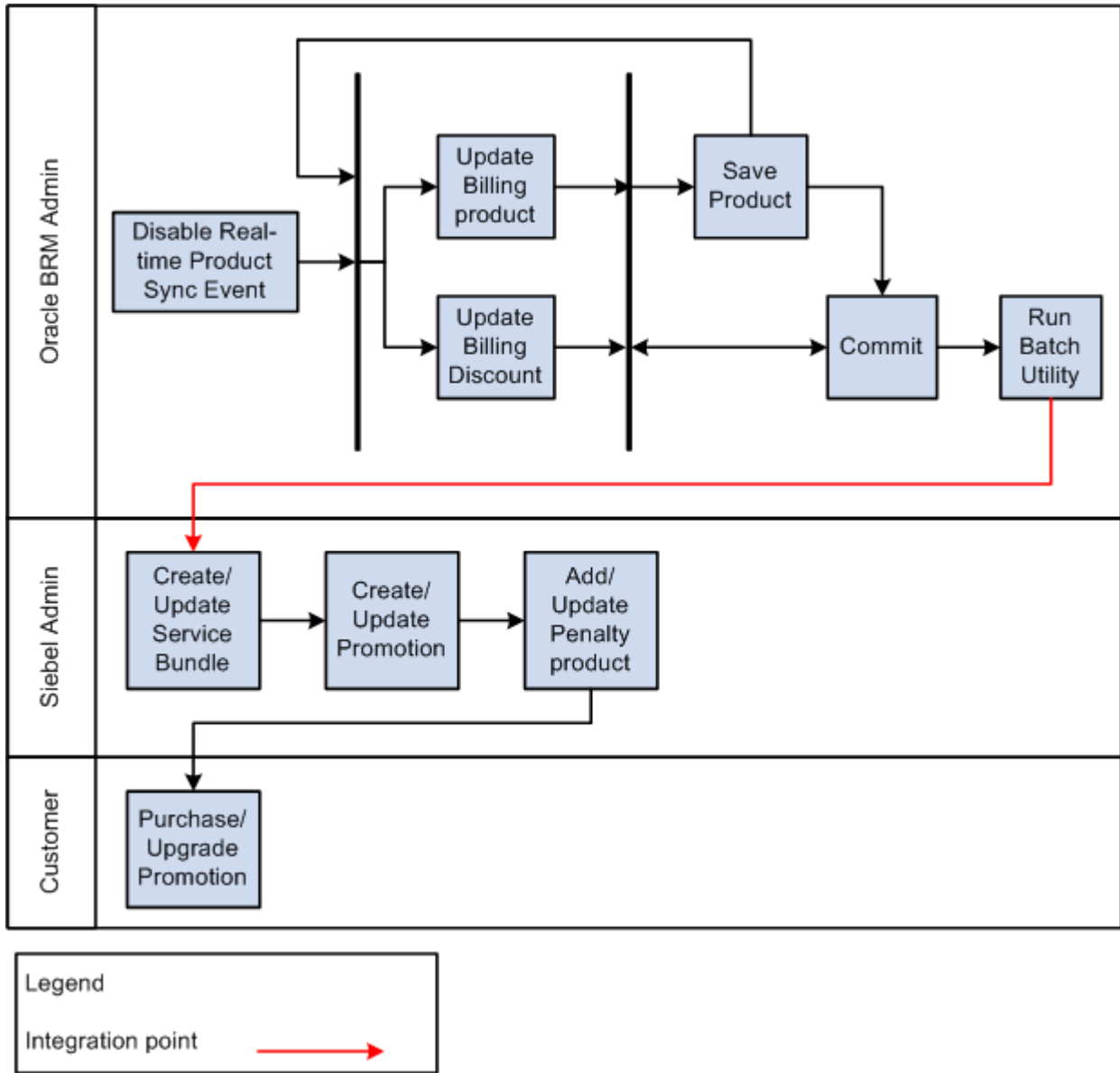
Synchronizing Updates to Billing Products and Billing Discounts in Batches

In this flow, the BRM product administrator disables the event for real-time product synchronization. The BRM product administrator can update the attributes of billing products and billing discounts in Pricing Center. The products can be saved singly or as sets in a file. The BRM product administrator runs a batch utility to store the updates singly or from the file in the BRM database and synchronize them with Siebel CRM.

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 can make any necessary changes to the promotions or bundles.

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

Figure 3-4 Synchronizing Updates to Billing Products and Billing Discounts in Batches



See *Oracle Communications Billing and Revenue Management Synchronization Queue Manager* for information about how to disable the event for real-time product synchronization.

About Simple and Customizable Products

When products are created in BRM, they are associated with billable events that determine how much and how often to charge customers. Each product created in

BRM is associated with at least one billable event. Products that are associated with a single event are synchronized to Siebel CRM as simple products and products that are associated with multiple events are synchronized as customizable products.

Table 3-1 gives an example of how products are synchronized to Siebel CRM. Because the Internet product in BRM has multiple events, it is synchronized as a customizable product in Siebel CRM.

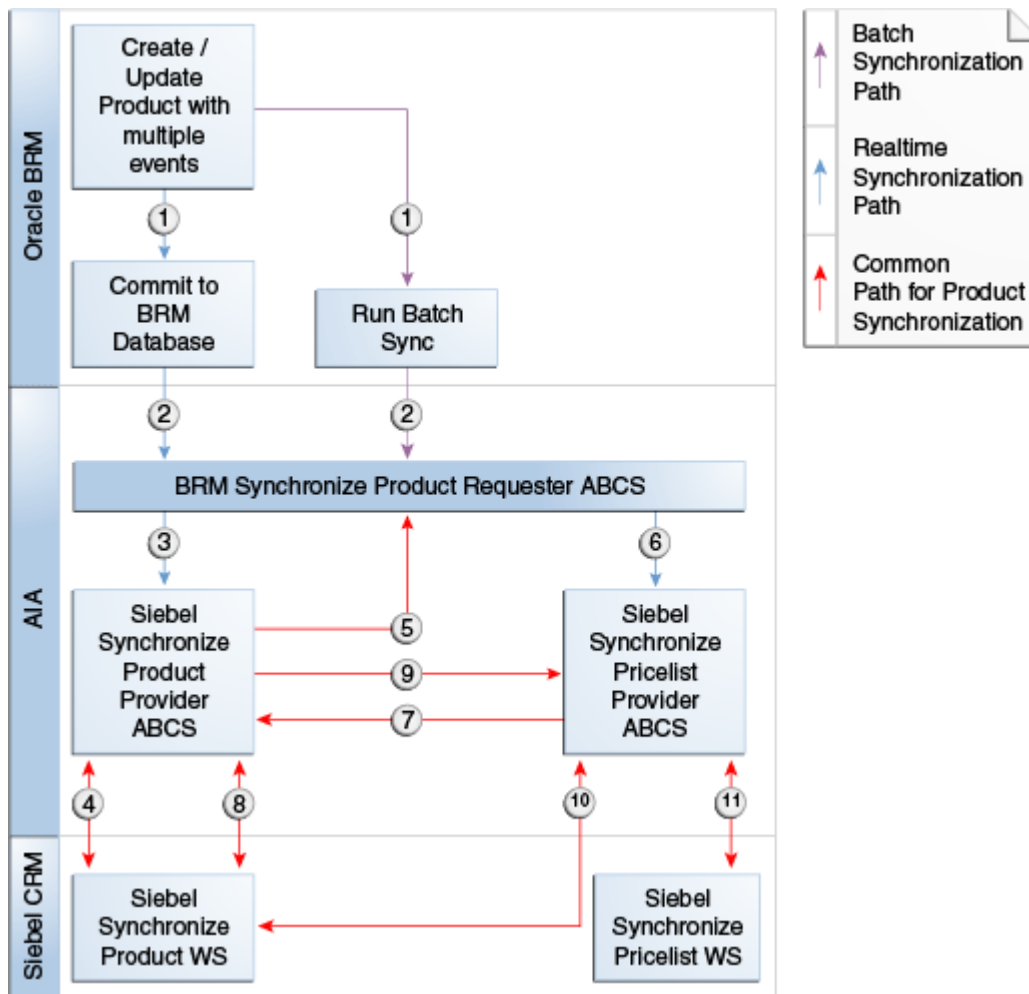
Table 3-1 Example of Synchronizing Products to Siebel CRM

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

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. A BRM user creates single-event and multi-event billing products in Pricing Center and commits them to the BRM database.
2. Real-time synchronization is invoked automatically or the product administrator invokes a batch utility to synchronize the products. The integration raises a ProductABM business event in BRM with the complete definition of the products.
3. The AQConsumer connector service, which is subscribed to this business event, extracts the product-related details from the ProductABM and passes the message to the BRM requester service.

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

7. The Siebel Synchronize Pricelist provider service transforms the standardized pricelist definition (PriceListEBM) to the Siebel-specific definition of the pricing. If there are multiple events associated with the pricing then simple products are created in the target CRM for each event. The prices related to the events are assigned to the corresponding simple products. To create simple products, the connector service transforms the events into a standardized representation of the items (ItemCompositionListEBM).

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

8. The Siebel Synchronize Product provider service transforms the standardized product definition (ItemCompositionListEBM) to a Siebel-specific definition of the product. It invokes the Siebel application web services to create the simple products for each event 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 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**, you must manually set the billable flag in Siebel CRM.

See *Siebel Communications Guide* for more information about setting the billable flag in Siebel.

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 and all other attributes go into the product lines.

Effective Start and End Dates

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

When the effective start date and effective end date are unspecified or the product has infinite effectivity, the BRM Enterprise Application Integration (EAI) `infranet.eai.xml_zero_epoch_as_null` parameter must be set to **TRUE**. Setting this parameter ensures that BRM publishes a null value for the effective start date and the effective end date.

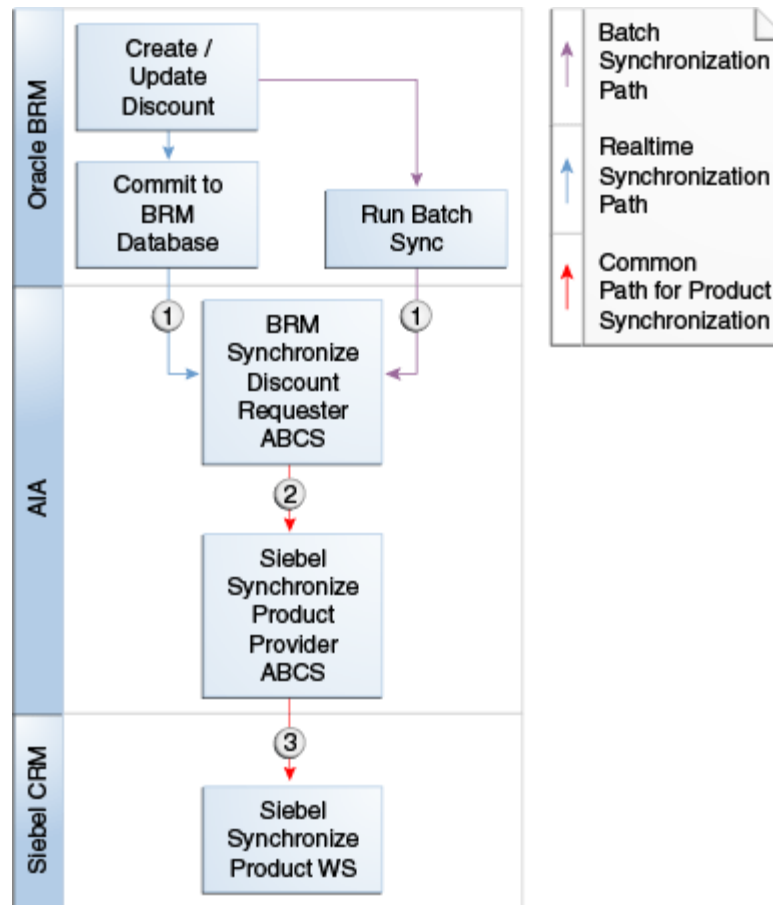
Caution: This is a mandatory step as part of the post installation setup activity.

See *Oracle Communications Billing and Revenue Management Developer's Guide* for more information defining infinite start and end date values.

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. The product administrator creates billing discounts in Pricing Center and commits them to the BRM database.
2. Real-time synchronization with Siebel CRM is invoked automatically or the product administrator invokes a batch utility to synchronize the products. The integration raises a DiscountABM business event in BRM with the complete definition of the discounts.
3. The connector service (BRM Synchronize Discount requester) that is subscribed to this business event extracts DiscountABM all the discount-related details and transforms them into a standardized representation of the discount (ItemCompositionListEBM).

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

4. The Siebel Synchronize Product provider service transforms the standardized discount definition (ItemCompositionListEBM) to a Siebel-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 BRM. The status of the web services call (Success or Fail) is returned back to the caller (Siebel Synchronize Product Provider service).

Usage Charges on Products

If a billing product has only one event, then the billing product is synchronized with Siebel CRM as a simple product with no list price.

Table 3–2 shows an example of a product in BRM, Wireless Usage, with only one event, Delayed Telco GSM Session. Wireless Usage is synchronized as a simple product in Siebel CRM.

Table 3–2 Example of a Synchronizing a Billing Product with a Single Event

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

If a billing product in BRM has two events and one of them is a usage charge event, then the billing product is synchronized with Siebel CRM as a simple product. The usage charge 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.

Table 3–3 shows an example of a product in BRM, Call Forwarding, with two events, Monthly Cycle Forward and Delayed Telco GSM Session, a usage charge event. Call Forwarding is synchronized as a simple product and the list price is the price of the Monthly Cycle Forward event.

Table 3–3 Billing Product with Two Events Example

Product in BRM	Simple Product in Siebel CRM
Call Forwarding: - Monthly Cycle Forward Event - \$3.00 - Delayed Telco GSM Session Event - \$0.40	Call Forwarding - \$3.0

If a billing product in BRM has more than two events and one event is a usage charge event then the billing product is synchronized with Siebel CRM as a customizable product. The usage charge event is not synchronized with Siebel CRM. The list price of the customizable product in Siebel CRM is set to charge on another event of the billing product.

Table 3–4 shows an example of a product in BRM, Internet, with three events, Product Purchase Fee, Monthly Cycle Forward Fee, and Delayed Telco GSM Session, a usage charge event. Internet is synchronized as a customizable product and the list price is the price of the Monthly Cycle Forward Fee event.

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

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

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.

See ["About One-Time Charges for Service Activation and Changes to Promotions and Service Bundles"](#) for more information about handling cancel fees (as a result of service, promotion cancellation/upgrade/downgrade).

See ["Working with DVMs for Product Lifecycle Management"](#) for more information about DVMs.

About Price Lists and Rate Plans

In Siebel CRM, a price list is a set of standard prices for products and services. You can use multiple price lists to offer separate prices for the same product and you can specify a default price list. The price list specifies a price, the currency for that price, and the frequency with which the price is charged.

For example, you can use separate price lists to charge business customers US\$30 a month for internet service and to charge residential customers US\$50 a month for the same service. In this example, the residential price list specifies that the price is 30, the currency is US dollars, and the frequency is monthly; the business price list specifies that the price is 50, the currency is US dollars, and the frequency is monthly.

You can use multiple price lists to offer different prices in different market segments (such as consumer or business customers, as in the previous example), different currencies, different sales channels (such as products purchased online or at a store), or different geographic locations.

Siebel CRM price lists map to rate plans in BRM. You create the price lists in Siebel CRM and set up the mapping between price lists and rate plans in the PRICELIST domain value map (DVM) before creating products in BRM. See ["Configuring Siebel CRM for Integrated Product Lifecycle Management"](#) for more information.

While creating products in BRM, you define rate plans to specify what to charge for the products. You associate the rate plans with corresponding price lists configured in Siebel CRM so that the integration can determine where Siebel CRM tracks charges.

Note: BRM also has a price list entity, but this is different from the Siebel CRM price list. When this document refers to price lists, it is referring to the Siebel CRM entity. For more information about BRM price lists, see *BRM Setting Up Pricing and Rating*.

Note: Integration of multiple price lists is supported only with BRM version 7.5 and later. For earlier versions, you must use a single default price list.

Working with Price Lists and Rate Plans at Design Time

At design time, you create products in BRM and define the rates to charge for those products in rate plans.

You can define rates in BRM according to the following rate plan structures:

- **Single rate plan:** charges according to one rate. The integration automatically associates the single rate plan structure with the default Siebel CRM price list.
- **Rate plan selector:** charges according to different rates depending on event data. You must associate each rate plan in a rate plan selector with a separate Siebel CRM price list.
- **Custom event analysis:** charges according to different rates depending on event data based on custom attributes. Using custom event analysis is similar to using a rate plan selector. You must associate each rate plan that uses custom event analysis with a Siebel CRM price list and you must modify BRM policy opcodes to define your custom rating criteria. See the discussion of using custom event analysis in the Pricing Center Help for more information about custom event analysis.

After you have created products in BRM and synchronized them to Siebel CRM, you can manage product pricing as described in "[Managing Pricing in Rate Plans and Price Lists](#)".

See *BRM Setting Up Pricing and Rating* for more information about rate plans, rate plan selectors, and custom event analysis.

Associating Rate Plans in BRM with Siebel CRM Price Lists

You associate rate plans in BRM with Siebel CRM price lists in Pricing Center using a rate plan selector.

To associate a rate plan with a price list using a rate plan selector:

1. In Pricing Center, follow the steps for defining rate plan selectors described in the Pricing Center Help.
2. When setting up columns for your rate plan selector, create a column called **EVENT.PIN_FLD_USAGE_TYPE**.
3. Add a row for each rate plan and corresponding price list that you intend to use.
4. In the **EVENT.PIN_FLD_USAGE_TYPE** column:
 - To associate a rate plan with a specific price list, enter the name of the price list exactly as it appears in the **PRICELIST DVM**.

If you enter a name that does not appear in the PRICELIST DVM, an error occurs when you synchronize the products to Siebel CRM.

- To associate a rate plan with the default price list, enter *. The integration maps * to the default price list. See [Table 3–7](#) for an example.
5. In the Rate Plan column, enter the name of the rate plans that correspond to the price lists that you entered in the EVENT.PIN_FLD_USAGE_TYPE column.
 6. Finish defining the rate plan selector as described in the Pricing Center Help.

Example Rate Plan Structures

In this example:

- Two products have been synchronized from BRM to Siebel CRM: **Broadband** and **GSM**.
- A default price list has been set up in Siebel CRM, and entered into the **AIAConfigurationProperties.xml** file and the PRICELIST DVM, as described in "[Configuring Siebel CRM for Integrated Product Lifecycle Management](#)".
- Four additional price lists have been set up in Siebel CRM and entered into the PRICELIST DVM: ConsumerPL, BusinessPL, NewYorkPL, and CaliforniaPL.
- Five rate plans have been set up in Pricing Center: ConsumerRP, BusinessRP, NewYorkRP, CaliforniaRP, and StatesRP.

[Table 3–6](#) shows the rate plan structure for the Broadband product. For this product, the product administrator uses two price lists to offer different prices for consumer and business customers.

Table 3–6 Example Rate Plan Structure for Broadband Product

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	12/31/2013	US\$40
BusinessRP	BusinessPL	1	01/01/2013	12/31/2013	US\$30

[Table 3–7](#) shows the rate plan structure for the GSM product. For this product, the pricing administrator uses the NewYorkPL and CaliforniaPL price lists to offer different prices for customers in New York and California and the default price list for customers in all other states. To make the integration use the default price list, the product administrator enters * for the price list associated with the StatesRP rate plan.

Table 3–7 Example Rate Plan Structure for GSM Product

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
NewYorkRP	NewYorkPL	1	01/01/2013	12/31/2013	US\$45
CaliforniaRP	CaliforniaPL	1	01/01/2013	12/31/2013	US\$40
StatesRP	*	1	01/01/2013	12/31/2013	US\$35

In Siebel CRM, the Broadband product is mapped to price list line items under the ConsumerPL and BusinessPL price lists and the GSM product is mapped to price list line items under the NewYorkPL, CaliforniaPL, and default price lists.

Offering a Product in Multiple Currencies

To offer a product in multiple currencies:

1. In Siebel CRM, create a separate price list for each currency, as described in ["Configuring Siebel CRM for Integrated Product Lifecycle Management"](#), and enter them in the PRICELIST DVM.
2. In Pricing Center, create rate plans that use the same currencies as the price lists in the PRICELIST DVM.
3. Define a rate plan selector for your product, associating the rate plans in the rate plan selector with the Siebel CRM price lists that use the corresponding currency. You must ensure that the currency in the rate plans matches the currency in the associated price lists. Currency matching is not validated by Siebel CRM or BRM.
4. Finish defining the rate plan selector and product as described in the Pricing Center Help.
5. Commit the product to the BRM database, which synchronizes the product to Siebel CRM.

Example of Offering a Product in Multiple Currencies

To offer a product called Broadband in Canadian dollars and U.S. dollars, the BRM product administrator uses a separate rate plan associated with a separate price list for each currency while creating the product.

In this example:

- A default price list has been set up in Siebel CRM and entered into the **AIAConfigurationProperties.xml** file and the PRICELIST DVM.
- Two additional price lists have been set up in Siebel CRM and entered into the PRICELIST DVM: CanadaPL and USAPL. The currency for the CanadaPL price list is Canadian dollars (CDN\$) and the currency for the USAPL price list is U.S. dollars (US\$).
- Two rate plans have been set up in Pricing Center: CanadaRP and USARP. The currency for the CanadaRP rate plan is Canadian dollars and the currency for the USARP rate plan is U.S. dollars.

The product administrator uses the rate plan structure shown in [Table 3–8](#) when creating the product.

Table 3–8 Offering the Broadband Product in Multiple Currencies

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
CanadaRP	CanadaPL	1	01/01/2013	12/31/2013	CDN\$30
USARP	USAPL	1	01/01/2013	12/31/2014	US\$35

When the product administrator commits the Broadband product to the BRM database to synchronize it to Siebel CRM, the Broadband product is mapped to price list line items under the CanadaPL and USAPL price lists.

Managing Pricing in Rate Plans and Price Lists

After you have synchronized your products from BRM to Siebel CRM, you can manage the prices in the rate plans in BRM and resynchronize the products to Siebel CRM to update the price lists. You can manage prices by:

- Changing the price of a product by updating the existing rate plan
- Changing the price list associated with a product's rate plan
- Changing a product from using multiple price lists to using the single default price list

Changing the Price of a Product

Use the following methods to change the price of a product in the rate plan:

- Change the price in the existing rate plan tier by changing the balance impact. See the discussion of defining balance impacts in the Pricing Center Help for more information.
- Add a new rate plan tier with the new price and adjust the effective dates of the old tier. See the discussions of defining single rate plans and defining valid time periods in the Pricing Center Help for more information.

Examples of Changing the Price of a Product

To change the price of the Broadband product, the BRM product administrator uses Pricing Center to edit the rate plan structure shown in [Table 3-6](#).

[Table 3-9](#) shows how the product administrator changes the price by changing the balance impact of the monthly cycle forward fee in the existing rate plan tier to US\$35.

Table 3-9 Changing the Balance Impact for the Broadband Product

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	12/31/2013	US\$35
BusinessRP	BusinessPL	1	01/01/2013	12/31/2014	US\$30

[Table 3-10](#) shows how a product administrator changes the price by adding a new tier with a monthly cycle forward fee of US\$35 to the ConsumerRP rate plan and adjusting effectivity dates of the old tier.

Table 3-10 Adding a Rate Tier to the Existing Rate Plan for the Broadband Product

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	01/31/2013	US\$40
ConsumerRP	ConsumerPL	2	01/31/2013	12/31/2014	US\$35
BusinessRP	BusinessPL	1	01/01/2013	12/31/2014	US\$30

When the product administrator has made the changes and committed the Broadband product to the BRM database, the Broadband product is resynchronized to Siebel CRM and the corresponding price list line items are updated.

Changing the Price List of a Product

To change the price list of a product:

1. In Pricing Center, set the duration end date to the current day for the rate plan that is associated with the old price list. See the discussion of defining the duration of a rate in the Pricing Center Help for more information.
2. Add a new row to the rate plan selector.

3. In the EVENT.PIN_FLD_USAGE_TYPE column, enter the name of the new price list for the rate plan exactly as it appears in the PRICELIST DVM.
If you enter a name that does not appear in the PRICELIST DVM, an error occurs during product synchronization.
4. Finish defining the new row for the rate plan selector as described in the Pricing Center Help.
5. Commit the product to the BRM database, which resynchronizes the product to Siebel CRM.

Example of Changing the Price List of a Product

To change the price list of the Broadband product with the rate plan structure shown in Table 3–6, the BRM pricing administrator uses Pricing Center to edit the rate plan structure. As shown in Table 3–11, the pricing administrator does the following:

- Changes the end dates for the ConsumerPL price list to the current date.
- Adds a new row to the rate plan selector for the ConsumerRP rate plan and new ConsumerPlusPL price list (which has already been created in Siebel CRM and entered in the PRICELIST DVM).

Table 3–11 Changing the Price List of the Broadband Product

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	01/31/2013	\$40
ConsumerRP	ConsumerPlusPL	1	01/31/2013	12/31/2014	\$40
BusinessRP	BusinessPL	1	01/01/2013	12/31/2014	\$30

When the product administrator makes the changes and commits the product to the BRM database, the product is resynchronized to Siebel CRM, the ConsumerPL price list line items are updated, and the Broadband product is mapped to new price list line items under the ConsumerPlusPL price list.

Changing a Product from Multiple Price Lists to the Default Price List

To change a product in BRM that has already been synchronized to Siebel CRM from using multiple price lists to using the default price list:

- If any of the rate plans in the rate plan selector is associated with the default price list (uses * in the EVENT.PIN_FLD_USAGE_TYPE column):
 1. In Pricing Center, set the duration end dates to the current date for all of the rate plans for the product associated with non-default price lists. Leave the rate plan associated with the default price list as is.

See the discussion of defining the duration of a rate in the Pricing Center Help for more information about setting the duration end date.
 2. Commit the product to the BRM database so that the product is resynchronized to Siebel CRM.
- If none of the rate plans in the rate plan selector are associated with the default price list:
 1. In Pricing Center, set the duration end dates to the current day for all of the rate plans associated with the product.

See the discussion of defining the duration of a rate in the Pricing Center Help for more information about setting the duration end date.

2. Commit the product to the BRM database, which resynchronizes the product to Siebel CRM.
3. Under the Rate Plan Structure column for the product, select **Single Rate Plan**.
4. Commit the product to the BRM database, which resynchronizes the product to Siebel CRM. The integration automatically associates the single rate plan structure with the default Siebel CRM price list.

Examples of Changing a Product from Multiple Price Lists to the Default Price List

To change the GSM product with the rate plan structure shown in [Table 3-7](#) from using multiple price lists to using the default price list, the BRM product administrator uses Pricing Center to edit the rate plan selector. The product administrator does the following:

- Changes the end dates of the NewYorkRP and CaliforniaRP rate plans to the current date. See [Table 3-12](#).

Table 3-12 *Setting the End Date for Rate Plans for the GSM Product*

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
NewYorkRP	NewYorkPL	1	01/01/2013	01/31/2013	\$45
CaliforniaRP	CaliforniaPL	1	01/01/2013	01/31/2013	\$40
StatesRP	*	1	01/01/2013	12/31/2013	\$35

- Commits the product to the BRM database to resynchronize the product to Siebel CRM and update the effectivity dates for the price list line items.

Setting the end date for the rate plans not associated with the default price list means that the integration uses only the default price list and StatesRP rate plan for that product.

To change the Broadband product with the rate plan structure shown in [Table 3-6](#) from using multiple price lists to using the default price list, the BRM product administrator uses Pricing Center to edit the rate plan selector. The product administrator does the following:

- Changes the end dates of the ConsumerRP and Business RP rate plans to the current date. See [Table 3-13](#).

Table 3-13 *Setting the End Date for Rate Plans for the Broadband Product*

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	01/31/2013	\$40
BusinessRP	BusinessPL	1	01/01/2013	01/31/2013	\$30

- Commits the product to the BRM database to resynchronize the product to Siebel CRM and update the effectivity dates for the price list line items.
- Selects **Single Rate Plan** under the Rate Plan Structure column for the Broadband product.

- Commits the product to the BRM database to resynchronize the product to Siebel CRM.

Changing the rate plan structure to **Single Rate Plan** means that no price list is associated with the rate plan in BRM. The integration automatically associates this rate plan structure with the default Siebel CRM price list and maps the Broadband product to price list line items under the default price list.

About BRM Balance Groups

A balance group is an object in the BRM database used for tracking the balance that your customers owe for their services. Because service-level balance groups are defined in plans in BRM, and plans are not synchronized to Siebel CRM, the integration does not provide design-time support for balance groups. The integration supports service-level balance groups and account-level balance groups at runtime when submitting Siebel CRM orders to BRM. You must enable or disable service-level balance groups for your entire system.

See "[Supporting Balance Groups](#)" for more information about balance groups and instructions for enabling or disabling service-level balance groups.

Understanding Product Bundling

This section describes the methodology for using service bundles and marketing bundles with billing products synchronized from BRM to Siebel CRM.

Basic Entity Mappings

[Table 3–14](#) shows the mapping between BRM and Siebel CRM entities.

Table 3–14 Mapping Between BRM and Siebel CRM Entities

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 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 BRM, then a customizable product is created in Siebel CRM.
Product Event Binding	Simple product (automatically created)	Each recurring and nonrecurring 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)	The Price list line in Siebel is mapped to information in Rate Plan, Rate Tier, and Balance Impact in BRM.
Deal	Service bundle (manually created)	If existing 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 3–14 (Cont.) Mapping Between BRM and Siebel CRM Entities

BRM Entities	Siebel CRM Entities	Description
Plan	Promotion /marketing bundle (manually created)	If existing 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 a 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 BRM purchased products.

Defining Products and Discounts in BRM

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

- In Siebel CRM, the maximum length for product names is 100 characters, while in BRM the maximum length is 255 characters. Ensure that your complete product name synchronizes from BRM to Siebel CRM by using product names that are 100 characters or less.
- Because usage events are not synchronized when they are included as a part of multi-event product in 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.
- Because the discount value of the 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 BRM must be set to **Y** for all charges that can be discounted when orders are interfaced to billing.
- The integration does not convert time zones when synchronizing BRM products and discounts to Siebel CRM.

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

- If the EAI `infranet.eai.date_pattern` property is *not* set, BRM publishes datetime information in the BRM local server time zone. This is the default behavior.
- If the EAI `infranet.eai.date_pattern` property is set, BRM publishes the datetime information in UTC/GMT time zone.

See *Oracle Communications Billing and Revenue Management Developer's Guide* for more information about setting this property.

Using Fixed Amounts versus Scaled Amounts in BRM

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

From the user interface perspective, in the pricing center application of 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 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, 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 BRM. Therefore, it is recommended that you not use fixed amounts for either one-time or recurring charges in BRM for implementations where the intent is to use the Siebel price override functionality.

See "About Real-Time Rate Plan" in *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating* for more information about using fixed and scaled amount fields.

Physical Goods

You can use either of the following approaches:

- Create physical goods as billing products in BRM at the account or service level. These are synchronized to Siebel CRM and can be added to the product hierarchy when creating bundles and promotions.
- Define physical goods in Enterprise Resource Planning (ERP). In this case, you are responsible for synchronizing them between ERP and BRM. The product synchronization process, which is supported by the process integration, is used to synchronize the product from BRM to Siebel CRM. If the service or marketing bundle contains one or more physical goods, then those products are passed to BRM when the order is interfaced to billing.

Sales Catalogs

After all of the 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 customizable products are added to the catalog then the components are automatically added.

Table 3–15 shows examples of products that would be included in the Siebel Catalog.

Table 3–15 Example of Products Included in the Siebel Catalog

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.

Recommendations for Product Definition in Siebel CRM

These are the recommendations for defining products:

- 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 BRM results in a price increase. Oracle recommends that only scaled charges be defined for the billing products of type item and subscription with one-time or recurring charges in BRM.
See "[Using Fixed Amounts versus Scaled Amounts in BRM](#)" for more information.
- The Product Management integration maintains cross-reference information between BRM billing products and Siebel CRM products. If you delete a billing product in 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. Instead of deleting the product, inactivate it by specifying an end date.
- If products updated in 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.
- You can nest billing products within each other in Siebel CRM. Though there is no limit the levels of nesting, any product nested more than two levels below a service bundle is purchased at the account level. See "[About Service Bundles](#)" for more information about service bundles and their components.

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 BRM. These can be account-level or service-level discounts. Because you can associate general ledger IDs (GLIDs) with them in BRM, you can account for them in the general ledger in separate accounts if needed.

These discounts are defined in 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. If these discounts are applied on the order at run time, there will be a difference between the start 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 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.

About Service Bundles

Service bundles are groups of related products that are sold as a package in Siebel CRM. You create service bundles in Siebel CRM to group the following types of product:

- **Billing products:** BRM products synchronized to Siebel CRM as simple or customizable products.
- **Billing discounts:** BRM discounts synchronized to Siebel CRM as simple products. Discounts included in a service bundle apply only to the products within the service bundle.
- **Non-billing products:** Products you create in Siebel CRM that are not synchronized from BRM.
- **Non-service-bundle customizable products:** Customizable products that you create in Siebel CRM to group service bundles and products (including account-level products and non-billing products) for re-use in promotions.

Service bundles must include at least one subscription-based billing product or discount. If a product bundle does not include at least one, model it as a non-service-bundle customizable product.

You can also include other service bundles in a service bundle. These are nested service bundles. There is no limit to the levels of nested service bundles.

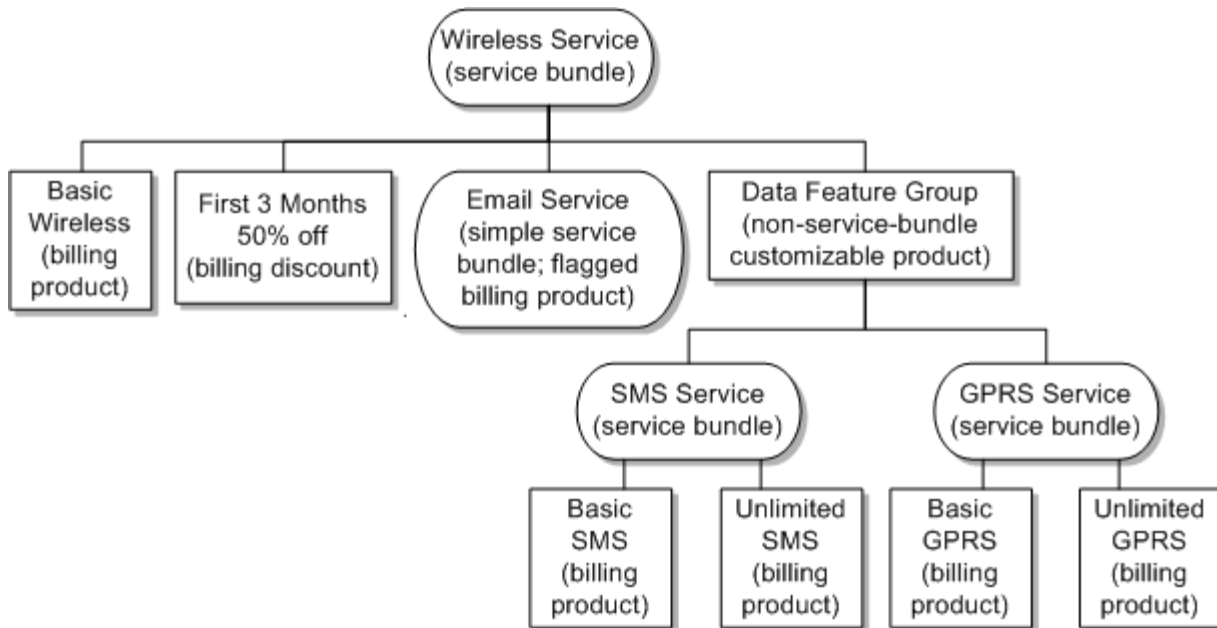
Account-level products, such as monthly charges for a hard copy of a bill, are charged to the account. Do not include these product in service bundles unless you nest them more than two levels below a service bundle.

You can nest billing products and discounts within another billing product or discount, but the integration synchronizes billing products or discounts nested more than two levels below a service bundle at the account level when they are purchased on a Siebel CRM order. See "[Supporting Product Bundling](#)" for more information about how the integration synchronizes the information on orders.

To create a service bundle in Siebel CRM, you manually create a customizable product with the billing type set to **Service Bundle** and choose which products to include in the service bundle.

You can flag subscription billing products synchronized from BRM as simple service bundles. See "[About Simple Service Bundles](#)" for more information.

[Figure 3-7](#) shows an example of the hierarchy in Siebel CRM for a service bundle that contains billing products, a billing discount, a simple service bundle, and a non-service-bundle customizable product.

Figure 3-7 Example of Service Bundle Hierarchy

When multiple instances of BRM are connected to the same Siebel CRM instance, all products included in a service bundle must come from the same BRM instance. Siebel CRM does not store the target billing instance details. See ["Configuring Multiple BRM Instances for Communications Integrations"](#) for more information about connecting multiple BRM instances.

See *Siebel Communications Guide* for more information about service bundles in Siebel CRM.

About Billing Service Types for Service Bundles

The billing service type is a field in BRM that you set when you create products and discounts to indicate which type of service they apply to. For example, a product charging for text messaging might have the service type `/service/telco/gsm/sms`.

Siebel CRM automatically assigns to service bundles the billing service type of their component products. Do not change the billing service type assigned to the service bundle.

For Oracle AIA to successfully send orders for service bundles to billing, you must only include products or discounts synchronized from BRM that have the same billing service type in a single service bundle.

Nested service bundles do not need the same billing service type as the service bundle that contains them (their parent service bundle), but all component billing products and discounts of a nested service bundle must have the same billing service type.

Because non-billing products and non-service-bundle customizable products are created in Siebel CRM, they do not have a billing service type.

About Simple Service Bundles

A simple service bundle is a subscription product synchronized from BRM that you flag in Siebel CRM. When you submit an order for a subscription product flagged as a simple service bundle, the integration treats the product as a service bundle in BRM. See ["Synchronizing Simple Service Bundles"](#) for more information about how the

integration synchronizes orders containing simple service bundles.

In Siebel CRM, you can model a simple service bundle by itself, nest it within another service bundle, or nest it within a non-service-bundle customizable product.

Flagging Subscription Products as Simple Service Bundles

To flag a subscription product synchronized from BRM as a simple service bundle in Siebel CRM, you set the Service Instance flag to **Y** in Siebel CRM for that product.

Note: You set the Service Instance flag manually. The integration does not set, update, or overwrite the flag when products are created or synchronized.

For more information about configuring simple service bundles with the Service Instance flag, see *Siebel Communications Guide*.

Example of Using a Service Bundle or a Simple Service Bundle

Table 3–16 shows how you can model the same products in Siebel CRM using service bundles or simple service bundles.

The following acronyms are used in the table:

- **CP:** Customizable product
- **SBC:** Service bundle component product synchronized from BRM
- **SB:** Service bundle manually created in Siebel CRM, billing type set to **Service Bundle**
- **SSB:** Simple service bundle made of a subscription product synchronized from BRM with the Service Instance flag set to **Y**

Table 3–16 Modelling Using Service Bundles or 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 (SBC)	1.1	---- Internet - Activation (SBC)
1.2	---- Internet - Activation (SBC) Note: The internet product is mapped to multiple events in BRM.	--	--
2	CP: Internet Service (SB)	2	CP: Internet Service (SB)
2.1	---- Dynamic Class	2.1	---- Dynamic Class
Only 1 of these 3 is selected	<i>Basic High Speed Internet MCF (SBC)</i> <i>Premium High Speed Internet MCF (SBC)</i> <i>Elite High Speed Internet MCF (SBC)</i>	Only 1 of these 3 is selected	<i>Basic High Speed Internet MCF (SBC)</i> <i>Premium High Speed Internet MCF (SBC)</i> <i>Elite High Speed Internet MCF (SBC)</i>
2.2	---- Internet Secure Firewall (SBC)	2.2	---- Internet Secure Firewall (SBC)
2.3	---- CP: High Speed Internet Features (NSB-CP) Note: The NSB-CP is optional; without it the four-feature SBs have the Internet Service SB as the parent.	2.3	---- CP: High Speed Internet Features (NSB-CP) Note: The NSB-CP is optional; without it the four-feature SBs have the Internet Service SB as the parent.

Table 3–16 (Cont.) Modelling Using Service Bundles or Simple Service Bundles

Hierarchy	Service Bundle	Hierarchy	Simple Service Bundle
2.3.1	----- CP: Internet email (SB)	2.3.1	----- Internet email (SSB)
2.3.1.1	----- Internet email (SBC)	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 (SBC)	2.3.4	----- CP: Internet Media (SB)
2.3.3	----- CP: Internet Conference Chat (SB)	2.3.4.1	----- Internet Content on Demand (SBC)
2.3.3.1	----- Internet Conference Chat (SBC)	2.3.4.2	----- Internet Video on Demand (SBC)
2.3.4	----- CP: Internet Media (SB)	2.3.4.3	----- High Speed Internet First Month-Free Discount (SBC)
2.3.4.1	----- Internet Content on Demand (SBC)	--	--
2.3.4.2	----- Internet Video on Demand (SBC)	--	--
2.3.4.3	----- High Speed Internet First Month-Free Discount (SBC)	--	--

Assumptions and Constraints for Working with Simple Service Bundles

The assumptions and constraints for working with simple service bundles are as follows:

- Simple service bundles can only ever have one billing product. They cannot include service-level billing discounts. To combine multiple products and discounts, you must use a regular service bundle.
- Only products of type **Subscription** can become simple service bundles.
- You cannot apply special rating, such as friends and family rates, to simple service bundles.
- You cannot bundle additional billing products and discounts, special rating products, or other service bundles within a simple service bundle.
- You cannot include existing products that have pending quotes, orders, or assets in Siebel CRM or are referenced by BRM in simple service bundles. Including such products would impact existing asset cross-references.
- You can neither convert simple service bundles into regular service bundles, nor convert regular service bundles into simple service bundles because of possible effects on the processing of change orders for existing assets. If your product bundling requirements change, you must create a different product in BRM, synchronize it to Siebel CRM, and bundle it differently.
- A product that is flagged as a simple service bundle cannot be included in a regular service bundle. A product that is already in a regular service bundle cannot be flagged as a simple service bundle. If your product bundling requirements change, you must create a new product in BRM, synchronize it to Siebel CRM, and bundle it differently.
- If you disconnect a simple service bundle, the integration disconnects both the service instance and the purchased product instance in BRM. You cannot change from one simple service bundle to another while retaining the same service instance.

- You must provide the service ID for both regular and simple service bundle lines for the integration to successfully interface purchases to BRM.

About 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 3–17](#) is an example of a marketing bundle for a wireless promotion with SMS.

Table 3–17 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
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 non-service-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.

See "[Product Definition Methodology for Friends and Family: Example](#)" for more information on 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.

See "[Understanding the Bill Fulfillment Order Business Flow](#)" for more information about Price and Discount overrides.

See *Siebel Pricing Administration Guide* for more information about Promotion definition.

Credit Limits

Because credit limits are typically defined at the billing-plan level in BRM, and such plans are not synchronized, you can optionally define the default credit limits for each separate service type. The integration does not support overrides of credit limits while bundling products or capturing orders.

About One-Time Charges for Service Activation and Changes to Promotions and Service Bundles

You can charge your customers for the following actions using one-time charges:

- Activating their services
- Canceling, upgrading, or downgrading a promotion
- Suspending, resuming, moving, or disconnecting a service bundle. These are called Move, Add, Change, and Disconnect (MACD) actions.

When charging for changes to a promotion, you can define proration plans in Siebel CRM to prorate the charge.

When charging for MACD actions on a service bundle, the integration uses related products in Siebel CRM instead of BRM-internal event mappings. Using Siebel CRM instead of BRM lets you see the charges on the order.

See the discussion of employee asset-based ordering in *Siebel Order Management Guide Addendum for Communications* for more information about setting up service charges using related products in Siebel CRM.

When you submit an order to cancel, upgrade, or downgrade a promotion, or suspend, resume, move, or disconnect a service bundle, Siebel CRM automatically adds the charge product with the appropriate charge amount to the order.

Charging for Service Activation and Changes to Promotions and Services

To charge your customers for canceling, upgrading, or downgrading a promotion:

1. In BRM, define penalty charges as **Item** products with a one-time charge.
2. Commit the products to the BRM database, which synchronizes them to Siebel CRM.
3. In Siebel CRM, modify the promotion disconnect workflow process (ISS Promotion Disconnect Process) to use the penalty charge products synchronized from BRM.

See "Workflows for Employee Asset-Based Ordering" in *Siebel Order Management Guide Addendum for Communications* for more information about ISS Promotion Disconnect Process.

To charge your customers for MACD actions for service bundles:

1. In BRM, define the charges as **Item** products for every service type that you enable MACD charges for.
2. Commit the products to the BRM database, which synchronizes them to Siebel CRM.
3. In Siebel CRM, associate the charge products for the MACD actions to the service bundles as related products. See the discussion of setting up service charges in *Siebel Order Management Guide Addendum for Communications* for more information.

To charge your customers for service activation:

1. In BRM, define an **Item** product with a one-time charge.
2. Commit the product to the BRM database, which synchronizes it to Siebel CRM.
3. In Siebel CRM, set the Track as Asset flag for the charge product to **Y**.

Supporting Friends and Family

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

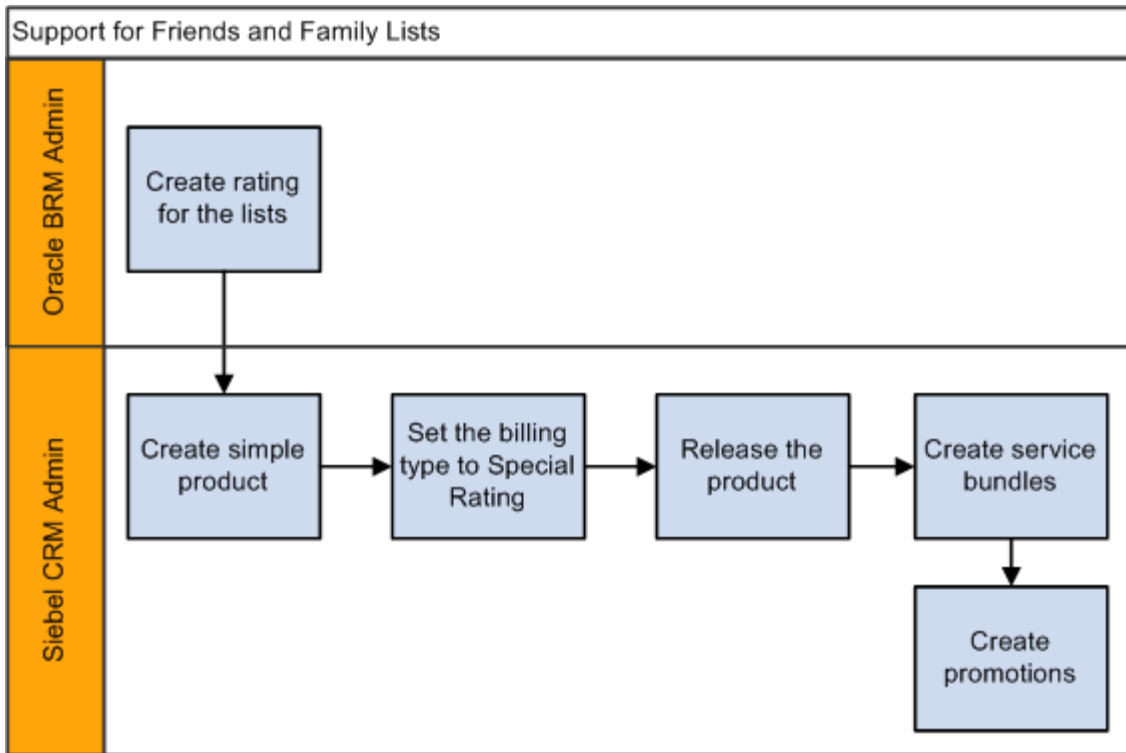
Special rating products and special rating profile lists in Siebel CRM are used to associate friends and family lists to services. Discounted rating for friends and family lists is defined in BRM.

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.

See ["Supporting Friends and Family Lists"](#) and ["Implementing the Synchronize Customer Special Rating Profile Business Flow"](#) for more information about how the lists are created and associated with the list product during run time.

[Figure 3–8](#) shows the business process task flow for friends and family.

Figure 3–8 Friends and Family Lists Business Process Flow



Enabling Friends and Family

To enable friends and family:

You must perform the following in 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 BRM Provisioning Tag Framework to support the Friends and Family feature.

See "Working with Extended Rating Attributes" and "About rating based on Friends and Family ERA" in Oracle *BRM Documentation* for more information.

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 BRM (while defining the discounted pricing for the lists).
2. Set the billing type of the product to **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.
- See "Profiles in Siebel Communications" in *Siebel Communications Guide* for more information on friends and family plans.

Product Definition Methodology for Friends and Family: Example

Table 3–18 and Table 3–19 are examples of the product definition methodology.

BRM Definition

Table 3–18 BRM Definition

Products in BRM	Service Type
Basic Wireless 550 ----- Monthly Cycle Forward Event ----- Delayed Telco GSM Event	/service/telco/gsm/telephony
Premium Wireless 800 ----- Monthly Cycle Forward Event ----- Delayed Telco GSM Event	/service/telco/gsm/telephony
Unlimited Wireless Voice ----- Monthly Cycle Forward Event ----- Delayed Telco GSM Event	/service/telco/gsm/telephony
Wireless Add On Line ----- Monthly Cycle Forward Event ----- Delayed Telco GSM Event ----- Product Purchase Fee Event	/service/telco/gsm/telephony
Wireless Voice Activation ----- Product Purchase Fee Event	/service/telco/gsm/telephony
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 BRM for rating phone numbers on the Special Rating lists. Use the label *Friends and Family*.

Siebel CRM Representation

Table 3–19 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

Service Bundles (SB)

Table 3–20 contains some examples of the service bundles that include special rating products.

Table 3–20 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	--
----- Wireless Voice Service Feature	--

Table 3–20 (Cont.) Service Bundles (SB)

Service Bundles	Comments
----- 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.

Promotions

Here are some examples of the promotion definition:

Table 3–21 Promotions

Promotion Component
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 3–21 (Cont.) Promotions

Promotion Component	
Nation 800 Minutes	<i>Different Promotion Variant created from the same bundle.</i>
----- Wireless Plan	
----- Wireless Service	
----- Premier Wireless 800	
----- Wireless Voice Activation	
----- Friends	<i>Friends and Family lists added to the Wireless Service</i>
----- 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	

About Time-Based Offerings

Time-based offerings let you use a Siebel CRM product class to set validity periods for products and discounts synchronized from BRM. When bundling products, you use the attributes of the product class to define the duration and its unit of measure. For example, you could offer a 50% discount on monthly cycle forward fees for the first three months by defining the duration as **3** and the unit of measure as **months** when bundling the discount product.

For more information about time-based offerings, see *Siebel Order Management Guide Addendum for Communications*.

For information about how the integration handles time-based offerings purchased on sales orders, see "[Supporting Time-Based Offerings on Orders](#)".

Setting Up Time-Based Offerings

To set up time-based offerings:

1. In Siebel CRM, create the following attribute definitions, using the values listed in [Table 3–22](#):
 - **Duration**: An integer used to calculate how long the time-based offering is valid.
 - **DurationUnitOfMeasure**: The unit used to measure the duration.
 - **DurationValidityStart**: The date that the duration should start, used to calculate the Service End Date. Possible values are:

- **Now:** The duration of validity starts at the due date. Siebel CRM calculates the service end date based on the due date plus the Duration value and the DurationUnitOfMeasure value.
- **Original Start:** The duration of validity starts at the service start date. Siebel CRM calculates the service end date based on the service start date and the Duration value and the DurationUnitOfMeasure value.
- **Original End:** The duration of validity starts at the original end date. This value is used in change orders. Siebel CRM calculates the service end date based on the original end date plus the Duration value and the DurationUnitOfMeasure value.

Table 3–22 Product Class Attributes for Time-Based Offerings

Attribute Name	Data Type	Domain Type	Value	Sequence	ValueSet	Attribute Type
Duration	Number	Enumerated	1	1	--	int
			2	2		
			3	3		
				
			31	31		
DurationUnitOfMeasure	Text	Enumerated	Days	1	UnitOfMeasure	enum
			Months	2		
			Years	3		
DurationValidityStart	Text	Enumerated	Now	1	ValidityStart	enum
			Original Start	2		
			Original End	3		

Note: To retrieve the values of the validity attributes in OSM, you must use the same attribute names listed in [Table 3–22](#).

2. Create a new product class for time-based offerings that uses the new attributes.
3. Release the product class for time-based offerings, which lets you use it with any charge or discount type product.
4. Change the simple products that you want to make time-based offerings to customizable products so that you can associate them with a product class.
5. Set the product type for the products and discounts that you want to make time-based offerings to **Time Based Offer**.
6. Associate the product class for time-based offerings with the products and discounts that you want to make time-based offerings and provide values for the validity attributes.
7. Create promotions and service bundles using the time-based offerings in the same way as you use regular products and discounts.

Note: Oracle recommends that you use product class inheritance to pass validity attributes along the product class hierarchy for time-based offerings so that the product class for time-based offerings does not conflict with product classes that support other Oracle AIA features.

For more information about creating attribute definitions and product classes, and associating the product classes with products, see the discussion of creating products with attributes in *Siebel Product Administration Guide*.

Managing Expired Time-Based Offerings

To manage expired time-based offerings:

1. In Siebel CRM, schedule a daily recurring job to execute the workflow that inactivates time-based offering assets whose end date has passed (SWI Asset Status Update Workflow) to ensure that change orders for services that include time-based offering products are successfully processed. For more information about this workflow, see *Siebel Order Management Guide Addendum for Communications*
2. In BRM, periodically run the **pin_cycle_fees -cancel** and **pin_discount_cleanup** utilities to ensure that purchased products and discounts reflect the correct status after passing the end date. For more information about these utilities, see *BRM Configuring and Running Billing*.
3. In BRM, create custom scripts to inactivate service instances that correspond to simple service bundles in Siebel CRM. These scripts are required because when the end date passes for time-based offerings for subscription products that are marked as simple service bundles in Siebel CRM, the Siebel CRM asset and corresponding purchased product instance in BRM change to inactive, but the corresponding service instance in BRM remains active.

Assumptions and Constraints for the Synchronize Product and Price Business Flow

- BRM deals and plans are not synchronized from BRM to Siebel CRM. The service bundles and promotions are manually defined in Siebel CRM.
- Credit limits are not synchronized from BRM to Siebel CRM.
- Sharing groups are not synchronized from BRM to Siebel CRM.
- Multiple brands defined within a single instance of BRM are not supported by the integration.
- The synchronization of billing products and billing discounts is one-way. Billing products created or updated in Siebel CRM are not synchronized back to BRM. BRM is the product master.
- The integration supports a single default Siebel CRM price list and optional additional Siebel CRM price lists. You must specify the default price list in the **AIAConfigurationProperties.xml** file and any additional price lists in the PRICELIST (DVM).

See "[Configuring Siebel CRM for Integrated Product Lifecycle Management](#)" for more information about creating and configuring price lists.

- Siebel CRM supports only one currency in each price list. To use multiple currencies, you must set up a separate price list for each currency. See ["Offering a Product in Multiple Currencies"](#) for more information.

- 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.

See the Siebel CRM product documentation for more information about business units.

See ["Configuring the Process Integration for Product Lifecycle Management"](#) for more information about configuration properties.

- 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.

See the Siebel CRM product documentation for more information about workspaces.

See ["Configuring the Process Integration for Product Lifecycle Management"](#) for more information about configuration properties.

- The integration sets the price to \$0 in the default Siebel CRM price list for products in BRM with multiple rate plans that are not explicitly mapped to Siebel CRM price lists.

See ["Understanding Product Bundling"](#) for more information.

- Oracle recommends 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.

See ["Defining Overrides on the Product Definition"](#) for more information about pricing commit type.

For 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 BRM and synchronized as products to Siebel CRM, they can only be used in the bundling of products if they are nested no more than two levels below a service bundle. Also, products, bundles, or promotions that 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.

- The lists associated with the Special Rating products (such as Friends and Family) are defined in Siebel CRM. The Siebel CRM pricing administrator must share the names of the Special Rating products list with the product administrator BRM pricing administrator outside of AIA so that the BRM product administrator can create the labels for the corresponding list names in BRM. BRM uses labels to identify the friends and family type lists. The labels are used to associate special pricing models in BRM Pricing.

- When a billing product is deleted in 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 BRM that is synchronized with Siebel CRM, then the cross-reference data for that billing product is not deleted. This must be purged manually. Oracle recommends that you do not delete products in BRM but instead inactivate the product in BRM by setting the product end date.
- The billable events that are associated with billing products in BRM must be included in the PRICETYPE_EVENT DVM. 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.
- BRM is the master for usage pricing. When billing products with only one usage event are synchronized from 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 BRM or applied when the order is interfaced to billing.
- Service bundles must have the same billing service type as their component products, except nested service bundles. Nested service bundles can have a different billing service type than their parent service bundle. Purchasing service bundles with a different billing service type than their component products (other than nested service bundles) can result in BRM grouping the billed charges under the wrong bill unit.
- The product synchronization sets the asset-trackable flag to **Y** for BRM products of type Subscription and **N** for products of type Item or System.
- 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.
- 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.
- By default 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 multi-play promotions where products are billing fulfilled in different billing systems).

Implementing the Synchronize Product and Price Business Flow

This chapter describes the Synchronize Product and Price business flow and explains how the Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM) Pre-built Integration option (the integration) implements the business flow using BRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Synchronize Product and Price Business Flow Overview

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

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

Product Synchronization Integration Flow

The product synchronization integration flow enables you to create new products in 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 BRM. The updates are then synchronized in Siebel CRM.

When products are created in 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.

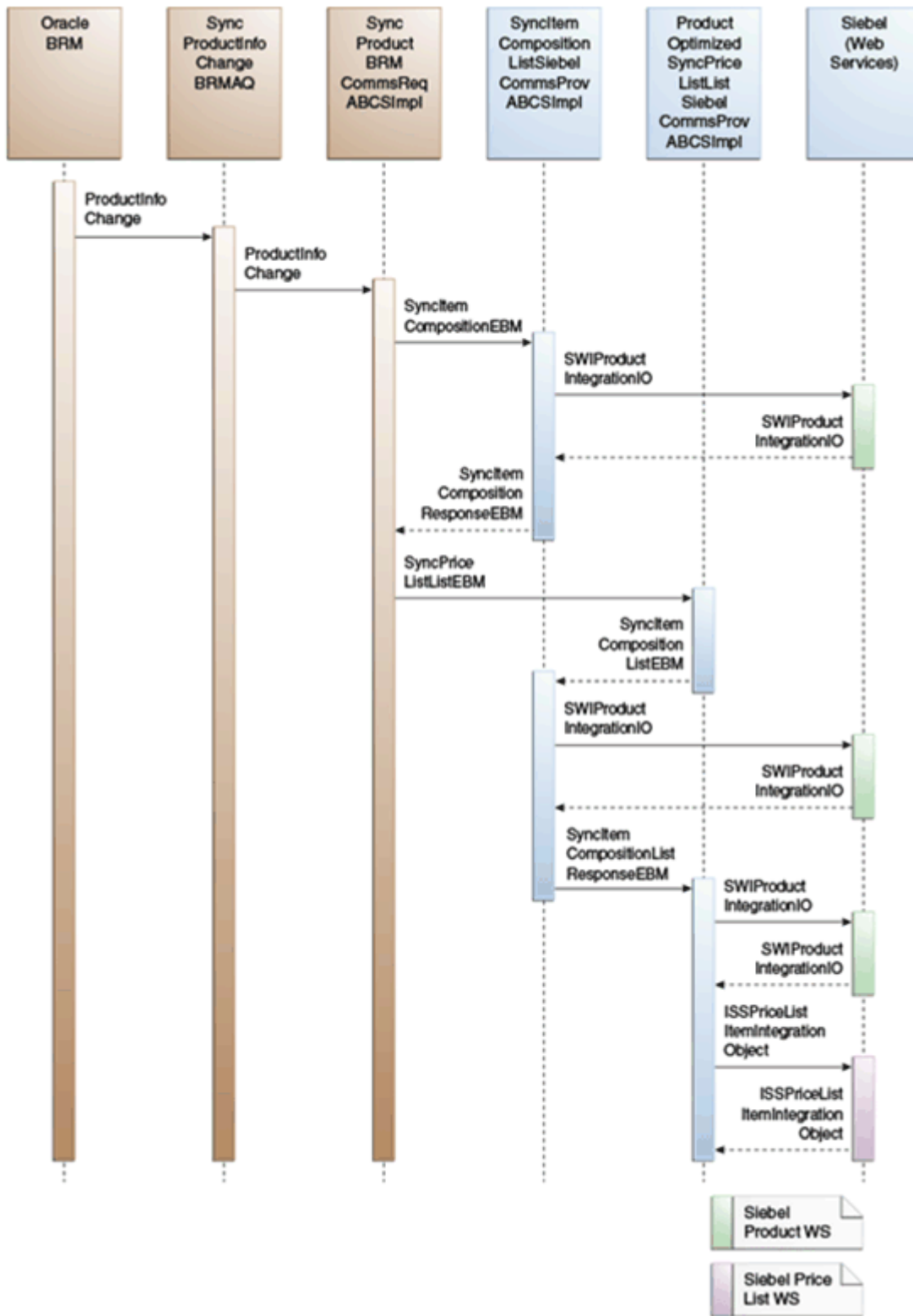
After synchronizing the product data from BRM to Siebel CRM, Siebel CRM returns a message containing both the product and the price data. The integration separates the product data from the price list data, and then synchronizes the price data in a separate process.

This integration flow delivers these services:

- SyncProductBRMCommsReqABCImpl with operation SyncProduct
- SyncItemCompositionListSiebelCommsProvABCImpl with operation SyncItemCompositionList
- ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl 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 Pricing Center, create or edit a product.

2. Commit the product to the BRM database so that the new or changed product drops into the BRM product queue.
3. Dequeue the BRM product queue. The adapter SyncProductInfoChangeBRMAQ polls the BRM product queue.
It is dequeued whenever it sees a message in the queue and invokes the SyncProductBRMCommsReqABCImpl with the operation SyncProduct.
4. The SyncProductBRMCommsReqABCImpl first transforms the BRM product message into an ItemCompositionEBM and routes the message to the SyncItemCompositionListSiebelCommsProvABCImpl
5. The SyncItemCompositionListSiebelCommsProvABCImpl transforms the ItemCompositionEBM into the Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIProductImportUpsert. The Siebel CRM web service completes the request and returns a response message. The SyncItemCompositionListSiebelCommsProvABCImpl then transforms the Siebel CRM response message to an ItemCompositionResponseEBM and sends it back to the SyncProductBRMCommsReqABCImpl.
6. The SyncProductBRMCommsReqABCImpl transforms the BRM product message into the PriceListListEBM and routes the message to the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.
7. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl 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 SyncItemCompositionListSiebelCommsProvABCImpl.
8. The SyncItemCompositionListSiebelCommsProvABCImpl transforms the ItemCompositionEBM to the Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIProductImportUpsert. The Siebel CRM web service completes the request and returns a response message. SyncItemCompositionListSiebelCommsProvABCImpl then transforms the Siebel CRM response message to an ItemCompositionResponseEBM and returns it to the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.
9. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms the PriceListEBM to a Siebel CRM price list message and then calls the Siebel CRM price list web service on operation Price_spcList_spcItem_spcInsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms the PriceListListEBM to a Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIProductImportUpsert. The Siebel CRM web service completes the request and returns a response message. SWIProductImportUpsert then transforms the Siebel CRM response message to a PriceList ListResponseEBM.

Billing Discount Synchronization Integration Flow

The billing discount synchronization integration flow enables you to create billing discounts as billing products in 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 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

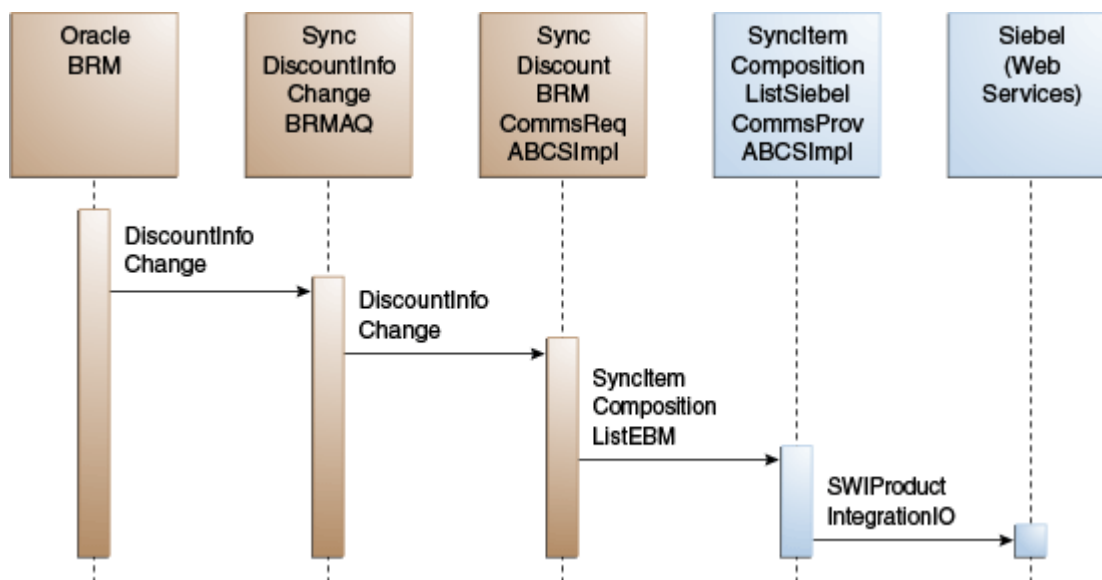
billing discount detail information in the description of the billing discount when creating billing discounts

This integration flow delivers these services:

- SyncDiscountBRMCommsReqABCSEImpl with operation SyncDiscount
- SyncItemCompositionListSiebelCommsProvABCSEImpl 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 Pricing Center, create or edit a discount.
2. Commit the discount to the BRM database so that the new or changed discount drops into the BRM product queue.
3. Dequeue the BRM discount queue. The adapter SyncDiscountInfoChangeBRMAQ polls the BRM discount queue. It is dequeued whenever it sees a message in the queue and invokes the SyncDiscountBRMCommsReqABCSEImpl with the operation SyncDiscount.
4. The SyncDiscountBRMCommsReqABCSEImpl first transforms the BRM discount message into the ItemCompositionEBM and routes the message to the SyncItemCompositionListSiebelCommsProvABCSEImpl.
5. The SyncItemCompositionListSiebelCommsProvABCSEImpl transforms the ItemCompositionEBM into the Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIProductImportUpsert. The Siebel CRM web service completes the request and returns a response message to the SyncItemCompositionListSiebelCommsProvABCSEImpl.

BRM Interfaces

The Synchronize Product and Price business flow uses the following BRM interfaces:

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

Siebel CRM Interfaces

The Synchronize Product and Price business flow uses the following Siebel CRM interfaces:

- SWIISPriceListItemIO: 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 CRM database.

See the discussion of Siebel CRM primary web services in *Siebel CRM Web Services Reference* for more information.

Industry Oracle AIA Components

The Synchronize Product and Price business flow uses the following communications industry Oracle 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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link

- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Synchronize Product and Price business flow:

- SyncProductBRMCommsReqABCImpl
- SyncDiscountBRMCommsReqABCImpl
- SyncItemCompositionListSiebelCommsProvABCImpl
- ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

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

See *Oracle Application Integration Architecture Pre-Built Integrations Utilities Guide* for more information about Session Pool Manager.

SyncProductBRMCommsReqABCImpl

The SyncProductBRMCommsReqABCImpl 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. A 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 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 BRM message. It fetches the PriceList name from a configuration parameter.

The configuration parameter is located in the **AIACConfigurationProperties.xml** file.

SyncDiscountBRMCommsReqABCImpl

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

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

SyncItemCompositionListSiebelCommsProvABCImpl

The SyncItemCompositionListSiebelCommsProvABCImpl process accepts the SyncItemCompositionListEBM. It transforms SyncItemCompositionListEBM into the

Siebel CRM product application business message (ABM). It then invokes the Siebel CRM Product web service to create products and product structures in Siebel.

This service is SPM enabled.

See *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide* for more information about Session Pool Manager.

ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl 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 CRM price list message and then calls the Siebel CRM price list web service on operation Price_spcList_spcItem_spcInsertOrUpdate. The

ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms the PriceListListEBM into a Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIProductImportUpsert. The Siebel CRM web service completes the request and returns a response message.

SWIProductImportUpsert then transforms the Siebel CRM response message to a PriceList ListResponseEBM.

This service is SPM enabled.

See *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide* for more information about Session Pool Manager.

Understanding the Query Product Classes Business Flow

This chapter explains concepts necessary for understanding the Query Product Classes business flow and defines the assumptions and constraints for the business flow. It describes how Siebel customer relationship management (Siebel CRM) organizes products and how Oracle Communications Order and Service Management (OSM) uses this organization in the business flow.

The Query Product Classes business flow is enabled using the following Pre-Built Integration options for the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and Oracle Communications Billing and Revenue Management (BRM) (the integration):

- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

If you are using Oracle Product Hub for Communications to manage your products, the Query Product Classes business flow is enabled using the following pre-built integration options in addition to those listed above:

- Oracle Product Master Data Management Integration Base Pack
- Oracle Product Master Data Management Integration Option for Design Studio

See *Oracle Application Integration Architecture Oracle Product Master Data Management Integration Implementation Guide* for more information about how the Query Product Classes business flow is implemented using Product Hub.

About Organizing Products

You organize products in meaningful ways, such as by service, size, or bandwidth, using product classes in Siebel CRM. You can create product classes manually in Siebel CRM or you can create them by importing item catalog categories from Product Hub.

When creating product classes, you associate attributes with them in Siebel CRM. Attributes are defined by sets of values. For example, the value set for a bandwidth attribute might include 2 megabits per second (Mbps), 3 Mbps, and 5 Mbps.

The Query Product Classes business flow lets you reuse the organization of your products from Siebel CRM in your order management system.

See *Siebel Product Administration Guide* for more information about product classes and attributes.

About the Query Product Classes Business Flow

The Query Product Classes business flow lets you reuse the organization of products from Siebel CRM in OSM.

You can query product classes and their attributes from Siebel CRM to create and update product specifications in Oracle Communications Design Studio for use in OSM. OSM uses product specifications for order orchestration.

See *Oracle Communications Order and Service Management Concepts* for more information about product classes and specifications in OSM and Design Studio.

Assumptions and Constraints for the Query Product Classes Business Flow

The assumptions and constraints for the Query Product Classes business flow are as follows:

- Any updates made to product specifications in Design Studio will not automatically be synchronized back to Siebel CRM. Any updates made to product classes in Siebel CRM will not automatically be synchronized to Design Studio. These changes must be resynchronized by a new query.
- Design Studio maintains the mapping between product specifications and product classes.

Implementing the Query Product Classes Business Flow

This chapter describes the Query Product Classes business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services. It also describes how the integration supports effectivity during design time.

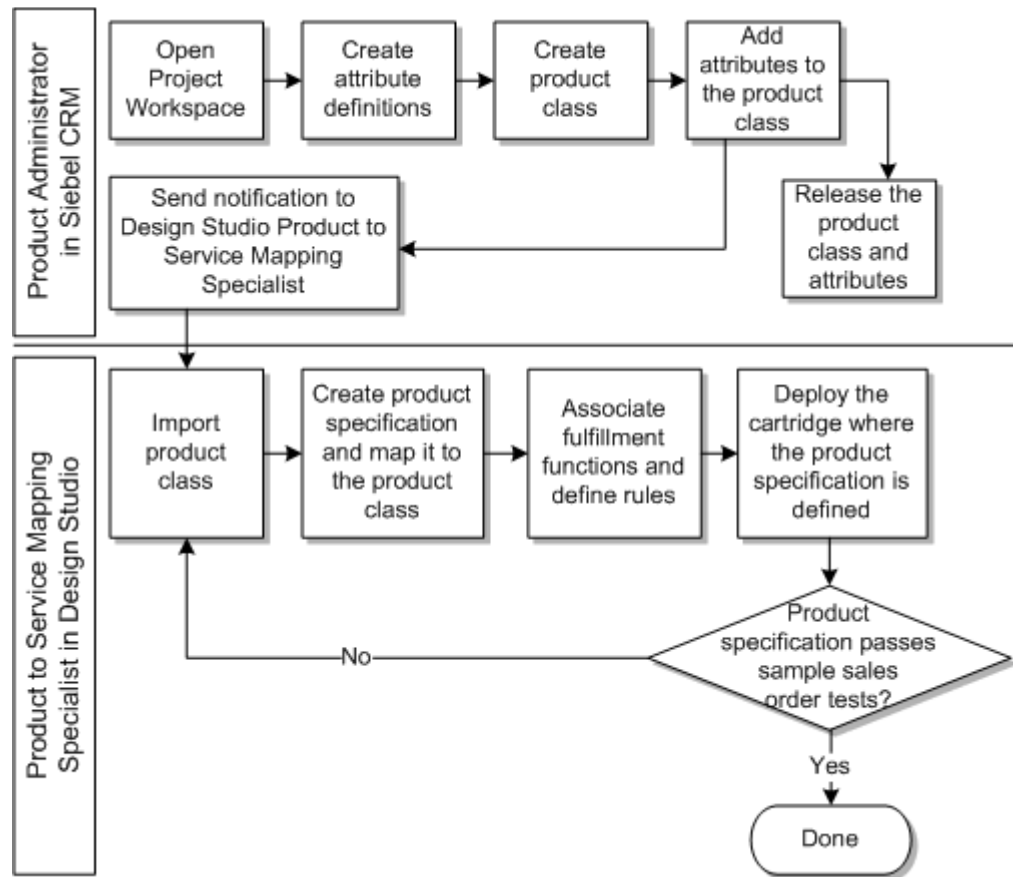
For information about the implementation of the Query Product Classes business flow using Oracle Product Hub to manage products, see *Oracle Application Integration Architecture Oracle Product Master Data Management Integration Implementation Guide*.

Overview of the Query Product Classes Business Flow

In this flow, a Siebel CRM user defines product classes and transaction attributes which an Oracle Communications Design Studio user queries and imports into a cartridge. The Design Studio user then maps the product class to a product specification in the cartridge and uses the product specification to associate decomposition rules, fulfillment functions, and their dependencies. After the design-time setup, the Design Studio user deploys the cartridge to OSM. The cartridge defines various fulfillment topologies to process order lines during order processing.

[Figure 6-1](#) illustrates the flow.

Figure 6–1 Query Product Classes Business Process Flow



The Query Product Classes business flow progresses as follows:

1. In a Siebel CRM Workspace project, a Siebel CRM user does the following:
 - a. Creates a new product class and transaction attributes and associates attribute definitions to them.
 - b. Updates the product class by adding or deleting transaction attributes.
 - c. Updates the attribute definitions (value sets) by adding or removing values from them.

See *Siebel Product Administration Guide* for more information about creating classes, attributes, and attribute definitions in Siebel.

2. The Siebel CRM user sends notification to a Oracle Communications Design Studio user in one of the following ways:
 - Using the notify menu function from the Product Class UI in Siebel CRM. The notification uses fixed templates to communicate the product class details. One or more classes can be sent on a single notification from Siebel CRM.
See the discussion of configuring the email notification for the product class in *Siebel Communications Guide* for more information about setting up the notification in Siebel CRM.
 - Using 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, the notification, must include the product class name so that the Design Studio user can use it to query in Design Studio. The template in the notify menu function automatically adds the product class name.

3. The Design Studio user does the following:
 - a. Queries or imports the product class and the transaction attribute details from Siebel CRM into a cartridge using the product class name.

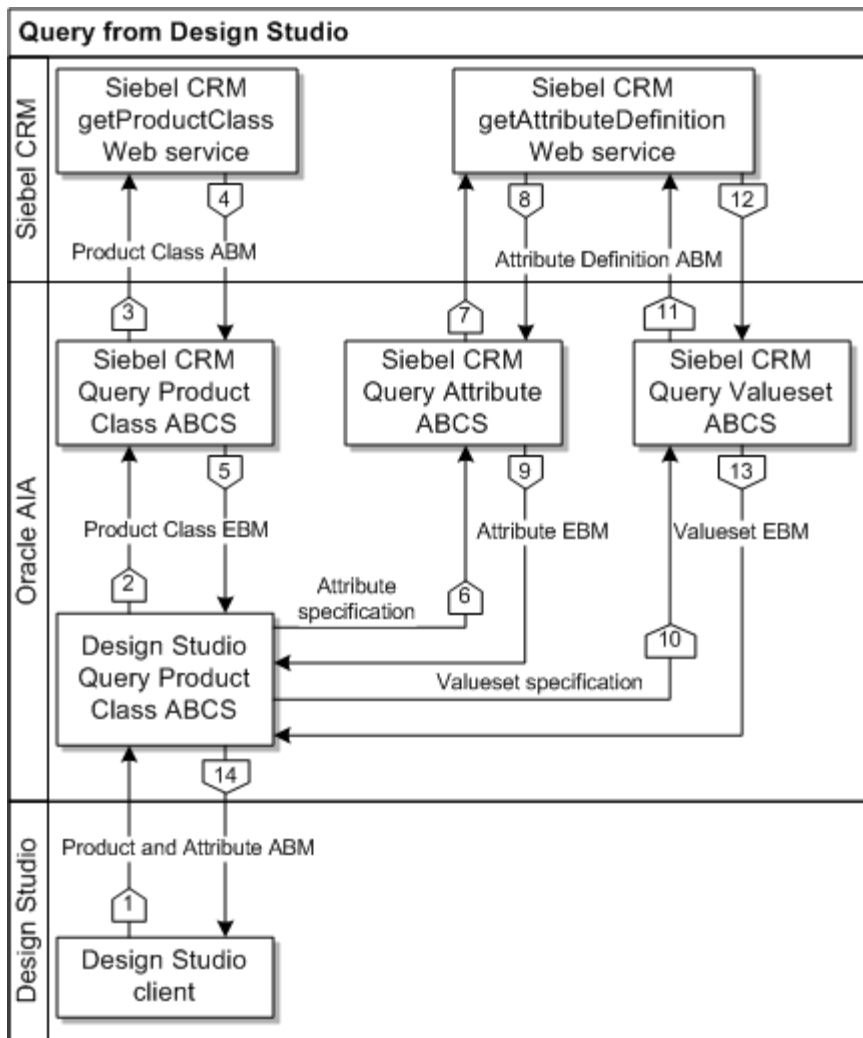
Design Studio stores the transaction attributes and associated valuesets in the data dictionary.
 - b. Maps the product class to an existing or newly-created product specification.
 - c. Defines fulfillment functions and their dependencies and associates them to the product specification.
 - d. Defines validation and decomposition rules.
 - e. Deploys the cartridge to OSM.
4. If the product specification causes errors in test orders, the Design Studio user does the following:
 - a. Creates and configures a new product specification or imports the product class again
 - b. Redeploys the cartridge.

For more information about Design Studio, including product specifications and cartridges, see *Oracle Communications Design Studio Concepts*.

Overview of Implementing the Query Product Classes Business Flow

[Figure 6–2](#) illustrates how the integration implements the Query Product Classes business flow.

Figure 6–2 Querying Product Class and Associated Attributes from Design Studio



When this process initiates, the following events occur:

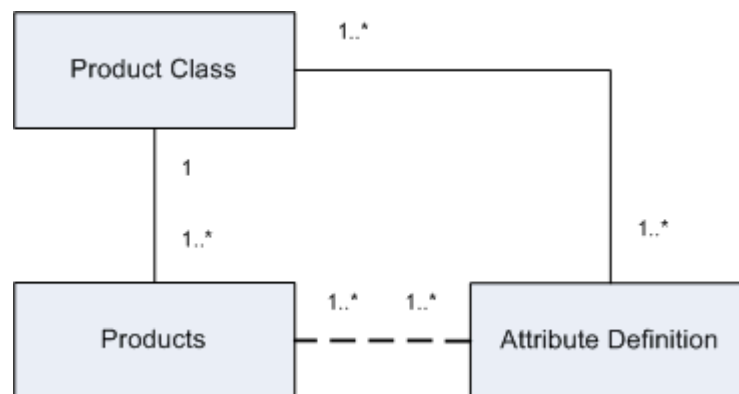
1. The product-to-service mapping specialist must include all the necessary product class names in the query dialogue in Design Studio and invoke the query operation. The Design Studio client calls the Design Studio Query Product Class Application Business Connector Service (ABCS) and provides one or more Class details as input. Design Studio must be configured to provide the input in the Product Class EBO structure that has only the Product Class names.
2. The Design Studio Query Product Class ABCS provides the Product Class enterprise business message (EBM) and routes it to the Siebel Query Product Class ABCS.
3. The Siebel Query Product Class ABCS transforms the Product Class EBM, extracts the product class name, and invokes the Siebel getProductClass Web service.
4. The Siebel getProductClass Web service returns the complete product class information for all input product classes to the Siebel Query Product Class ABCS as a Product Class ABM.
5. The Siebel Query Product Class ABCS transforms the Product Class ABM into the Product Class EBM and routes it to the Design Studio Query Product Class ABCS.

6. The Design Studio Query Product Class ABCS identifies that the Product Class EBM has references to one or more attribute definition entities. 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 and provides all attribute IDs that are associated with all the product classes. It routes a request to the Siebel CRM Query Attribute ABCS.
7. The Siebel Query Attribute ABCS invokes the `getAttributeDefinition` Web service.
8. The `getAttributeDefinition` Web service returns the complete attribute information for all attribute IDs to the Siebel CRM Query Attribute ABCS.
9. The Siebel Query Attribute ABCS performs the transformation, filters the necessary fields, constructs the Attribute EBM and routes it back to the Design Studio Query Product Class ABCS.
10. The Design Studio Query Product Class ABCS identifies that the attributes have references to the valueset entity. To query all of the valuesets associated with all the attributes it aggregates all of the attributes that are associated with all the valuesets and routes a request to the Siebel CRM Query Valueset ABCS as a Specification Valueset EBM.
11. The Siebel Query Valueset ABCS invokes the `getAttributeDefinition` Web service.
12. The `getAttributeDefinition` Web service returns the complete attribute information to the ABCS as an Attribute Definition ABM.
13. The Siebel Query Valueset ABCS performs the transformation, filters the necessary fields, constructs the Valueset EBM, and routes it back to the Design Studio Query Product Class ABCS.
14. The Design Studio Query Product Class ABCS composes the Product Class EBM, the Attribute EBM, and the Valueset EBM and returns it to the Design Studio client application as a Product and Attribute ABM.

Logical Data Model in Siebel CRM

Figure 6–3 illustrates the logical data model.

Figure 6–3 Logical Data Model of Product Class Structure in Siebel CRM



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 Design Studio, which updates the corresponding valueset metadata in the data dictionary.

Supporting for Effectivity During Design-Time

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 Design Studio must be updated and the cartridge must be redeployed to the environment.

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

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 Design 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.

Query Product Classes Integration Flow

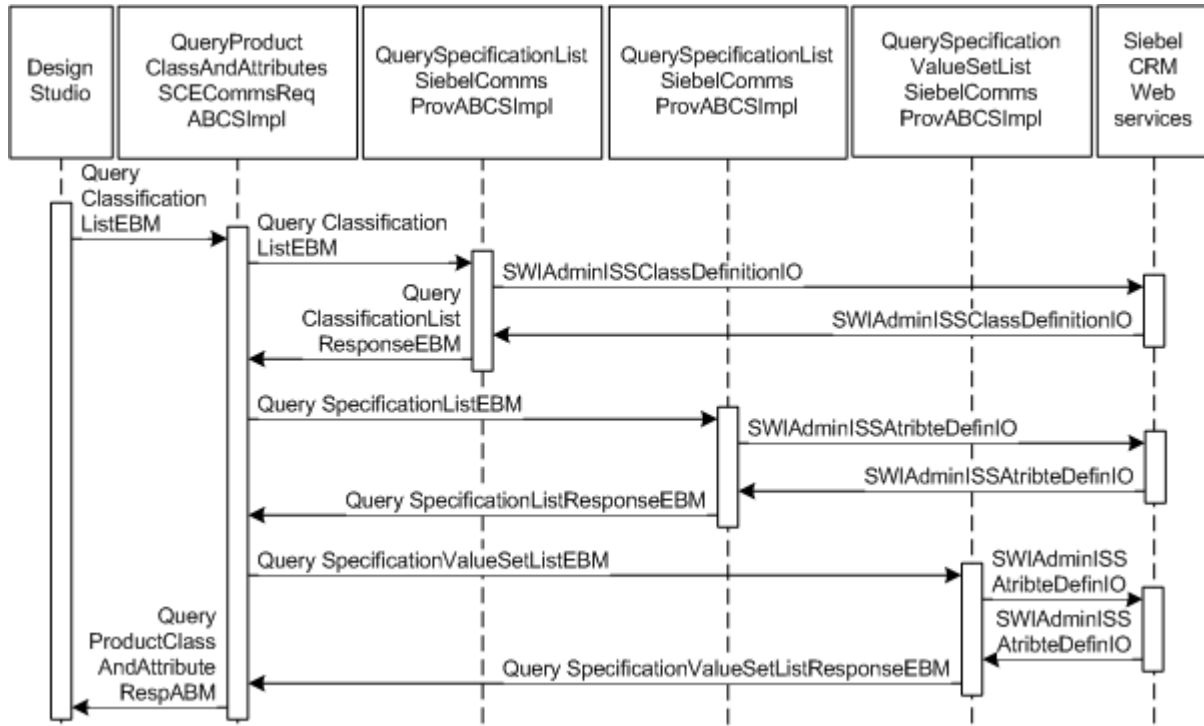
This integration flow uses the following interfaces:

- QueryProductClassAndAttributesSCECommsReqABCImpl

- QueryClassificationListSiebelCommsProvABCImpl
- QuerySpecificationListSiebelCommsProvABCImpl
- QuerySpecificationValueSetListSiebelCommsProvABCImpl

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. A Design Studio user (a product-to-service mapping specialist) triggers the Query Product Class and Attribute definitions integration flow using the Design Studio client, which provides the Class Code and the Oracle Fusion Middleware (FMW) URL. In the case of an update, Design Studio also provides the Class Codes for all of the subclasses.
2. Design Studio invokes the `QueryProdClassAndAttributesSCEReqCommsABCImpl` with the `QueryClassificationListEBM`, which contains the Product Class codes.
3. `QueryProdClassAndAttributesSCEReqCommsABCImpl` passes through the `QueryClassificationListEBM` to the `QueryClassificationListSiebelCommsProvABCImpl`.
4. `QueryClassificationListSiebelCommsProvABCImpl` 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

QueryClassificationListSiebelCommsProvABCImpl transforms the Siebel GetProductClassResponseABM into the QueryClassificationListRespEBM.

6. The QueryClassificationListRespEBM goes all the way back to the QueryProdClassAndAttributesSCEReqCommsABCImpl.
7. Next, the QueryProdClassAndAttributesSCEReqCommsABCImpl 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 QueryProdClassAndAttributesSCEReqCommsABCImpl invokes the QuerySpecificationListSiebelCommsProvABCImpl with the QuerySpecificationListEBM.
9. The QuerySpecificationListSiebelCommsProvABCImpl transforms the QuerySpecificationListEBM into the Siebel Attribute definitions IO (ABM) along with the Workspace Name from the AIAConfigurationProperties.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 QuerySpecificationListSiebelCommsProvABCImpl transforms the response into the QuerySpecificationListRespEBM.
11. The QueryProdClassAndAttributesSCEReqCommsABCImpl then invokes the QuerySpecificationValueSetListSiebelCommsProvABCImpl with the QuerySpecificationValueSetListEBM. The QuerySpecificationValueSetListSiebelCommsProvABCImpl transforms the QuerySpecificationValueSetListEBM into the Siebel Attribute definitions IO (ABM) along with the Workspace Name from the AIAConfigurationProperties.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 QuerySpecificationValueSetListSiebelCommsProvABCImpl transforms the response into the QuerySpecificationValueSetListRespEBM.
13. The QuerySpecificationListRespEBM and QuerySpecificationValueSetListRespEBM go all the way back to the QueryProdClassAndAttributesSCEReqCommsABCImpl. The QueryProdClassAndAttributesSCEReqCommsABCImpl merges the QuerySpecificationListRespEBM, QuerySpecificationValueSetListRespEBM and the QueryClassificationListRespEBM and maps them to the QueryProdClassAndAttributesRespABM.
14. The QueryProdClassAndAttributesSCEReqCommsABCImpl replies to Design Studio 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.

Siebel CRM Interfaces

The Query Product Classes business flow uses the following 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.

See the discussion of Siebel CRM primary web services in *Siebel CRM Web Services Reference* for more information.

Industry Oracle AIA Components

The Query Product Classes business flow uses the following communications industry-specific Oracle AIA 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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The Query Product Classes business flow uses the following integration services:

- QueryProdClassAndAttributesSCEReqCommsABCImpl with operation QueryProdClassAndAttributes
- QueryClassificationListSiebelCommsProvABCImpl with operation QueryProductClass
- QuerySpecificationListSiebelCommsProvABCImpl with operation QuerySpecificationList

- QuerySpecificationValueSetListSiebelCommsProvABCImpl with operation QuerySpecificationValueSetList

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

See *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide* for more information about Session Pool Manager.

QueryProdClassAndAttributesSCEReqCommsABCImpl

The QueryProdClassAndAttributesSCEReqCommsABCImpl is a synchronous BPEL process and is the Design Studio 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 requestor ABCS. This service has one operation: QueryProdClassAndAttributes. This accepts a QueryClassificationListEBM as a request and returns QueryProdClassAndAttributesRespABM as a response.

QueryClassificationListSiebelCommsProvABCImpl

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.

QuerySpecificationListSiebelCommsProvABCImpl

The QuerySpecificationListSiebelCommsProvABCImpl 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.

QuerySpecificationValueSetListSiebelCommsProvABCImpl

The QuerySpecificationValueSetListSiebelCommsProvABCImpl 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.

Understanding the Process Integration for Order Lifecycle Management

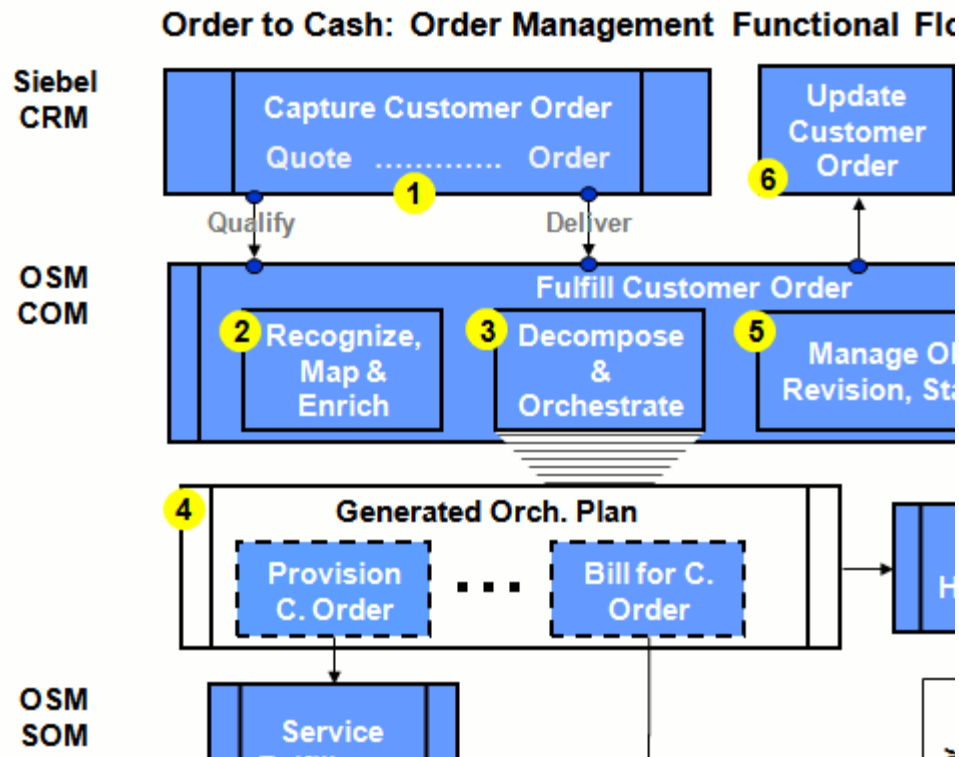
This chapter describes the process integration for Order Lifecycle Management (OLM) and discusses a typical topology and order capture flow. It also describes the Qualify Customer Order and Deliver Customer Order subflows and design considerations for product definition and mapping.

Order Lifecycle Management Overview

The process integration for OLM extends from the time a quote or order is created to the time when the goods and services are delivered and billed. The Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) works with participating applications to accomplish this process for customer relationship management, order management, billing, and 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



The functional flow for OLM is as follows:

1. **Capture Customer Order:** A customer order is captured and, if necessary, validated in Siebel CRM. When the order capture is complete and the order is validated, the system submits it to OSM in the central order management (COM) role for delivery. In Figure 7-1, the two arrows from Capture Customer Order to Fulfill Customer Order show the Qualify Customer Order and Deliver Customer Order subflows.
2. **Recognize, Map, and Enrich:** OSM recognizes customer orders (both Qualify and Deliver request types) as Oracle Application Integration Architecture (Oracle AIA) customer orders, maps them to fulfillment patterns, and enriches them with fulfillment metadata.
3. **Decompose and Orchestrate:** OSM decomposes and orchestrates the customer orders, dividing the order into suborders called order components. Order components have cross-order components, cross-order lines, and cross-order dependencies that reflect the specific demands of the communications service provider.
4. **Generate Orchestration Plan:** The outcome of decomposition and orchestration 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 OSM Order to Activate PIP cartridge product provides ready-to-use automatic integration with AIA web services. When BRM) pre-built integration option is in use, the integration forwards the billing related requests (Sync Customer, Initiate and Fulfill Billing) generated in OSM to BRM. The Sync Customer Account integration flow also uses the Siebel CRM pre-built integration option to get customer account details.

5. **Manage OLM Events:** OSM manages OLM events. For cancel and revision requests, OSM generates and executes compensation plans to match a change. OLM manages order data and status updates, and order fallout.
6. **Update Customer Order:** Throughout the fulfillment process, OSM maps fulfillment function responses to common statuses, which are then aggregated into order line statuses and order header status values. OSM updates Siebel CRM with relevant customer status and milestone values when order lines reach their point-of-no-return (PONR) to prevent the submission of new revisions. OSM also updates Siebel CRM with any enrichment to order lines that occurs during fulfillment.
7. **Create/Update Trouble Tickets:** OSM detects, reports, and resolves order fulfillment fallout incidents such as system, validation, and fulfillment errors. AIA also reports any integration errors to OSM. OSM then creates trouble tickets in Siebel CRM for error notification, reporting, and management.

Note: See ["Understanding the Process Integration for Order Fallout Management"](#) for more information about managing order fallout in OSM and creating trouble tickets in Siebel CRM.

OSM delivers pre-built cartridges for use with the integration and provides an Oracle AIA Emulator, which you can use to emulate an order. See *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture* for more information about how to install and deploy the delivered cartridges and the emulator.

Note: This guide focuses on the automated integration points among Siebel CRM, OSM COM, OSM Service Order Management (SOM), and billing. This guide does not cover process details within OSM SOM, for example, service design, assign, and activation.

Order Lifecycle Management Business Flows

The process integration for OLM enables the following business flows:

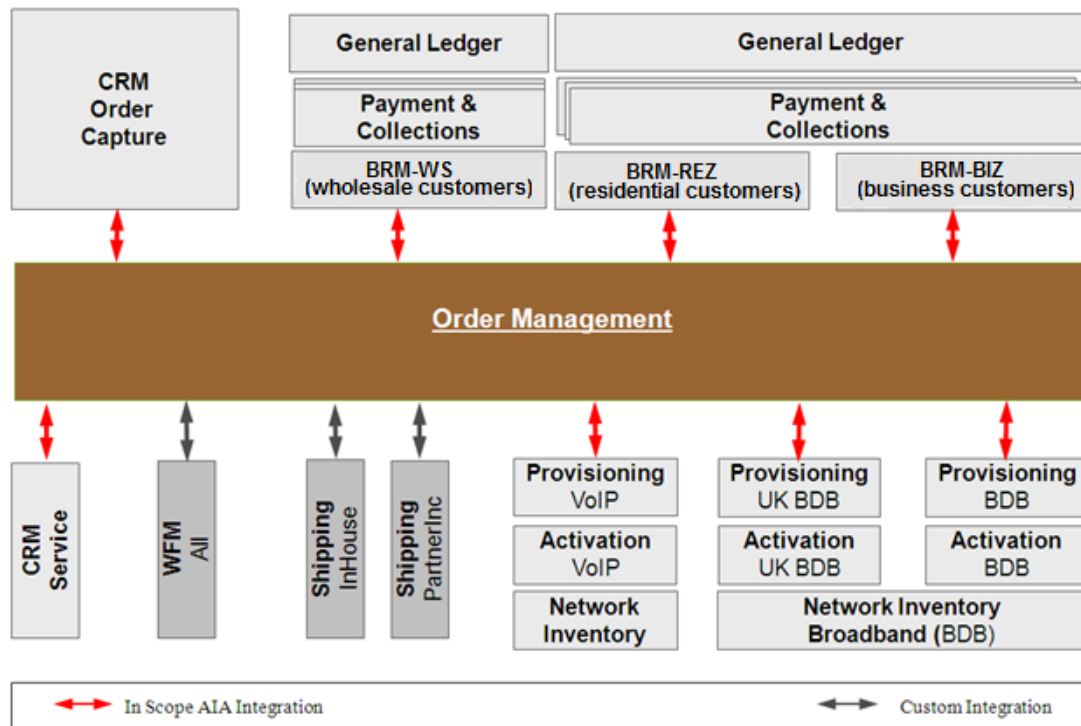
- **Process Sales Order Fulfillment:**
 - Enabled using either the Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option or the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option.
 - Used when submitting orders from Siebel CRM to OSM for order fulfillment processing.
 - See ["Understanding the Process Sales Order Fulfillment Business Flow"](#) for more information.
- **Synchronize Fulfillment Order Billing Account:**
 - Enabled using the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option.
 - Used when interfacing orders to create customer data in BRM.
 - See ["Understanding the Synchronize Fulfillment Order Billing Account Business Flow"](#) for more information.
- **Bill Fulfillment Order:**

- Enabled using the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option.
- Used when interfacing orders to create transaction data in BRM.
- See "[Understanding the Bill Fulfillment Order Business Flow](#)" for more information.
- **Provision Order and Update Fulfillment Order:**
 - Enabled using either the Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option or the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
 - Used when provisioning orders in OSM in the SOM role, and updating orders and statuses in OSM in the COM through explicit order updates from OSM in the SOM role.
 - See "[Understanding the Provision Order and Update Fulfillment Order Business Flows](#)" for more information.
- **Update Sales Order:**
 - Enabled using either the Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option or the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
 - Used when sending order updates from OSM in the COM role to Siebel CRM.
 - See "[Understanding the Update Sales Order Business Flow](#)" for more information.

Typical Topology

[Figure 7-2](#) illustrates a typical deployment topology for the integration. An order management system is at the center of the integration deployment

Figure 7-2 Typical Oracle Communications Order to Cash Deployment Topology



The figure shows order captures systems passing orders to the order management system that is central to the integration deployment. The order management system then decomposes the order into suborders called order components, each of which targets a particular fulfillment provider.

The topology in the figure uses the following fulfillment providers:

- Three billing providers based on customer segment: wholesale, residential, and business.
- Three provisioning stacks based on service family and geography: VoIP, UK Broadband, and Broadband.
- Two shipping providers, one for in-house products and another for partner supplier products.
- One workforce management provider
- One customer relationship management (CRM) service provider for trouble ticketing

Order Capture Overview

Figure 7-3 shows a typical order capture flow. The flow can vary by service provider for many reasons, including service family, customer segment, and line of business.

Order-based system interactions between different business support systems (BSS) and operational support systems (OSS) generally require order decomposition and orchestration to go through the order management layer. The process integration for OLM features integration points for the following systems interactions:

- **Qualify Customer Order:** validates the availability of a service design and the capacity to fulfill the customer order

- Deliver Customer Order: fulfills the products and services purchased by the customer or fulfills actions on existing customer assets.

Figure 7–3 Typical Order Capture Flow

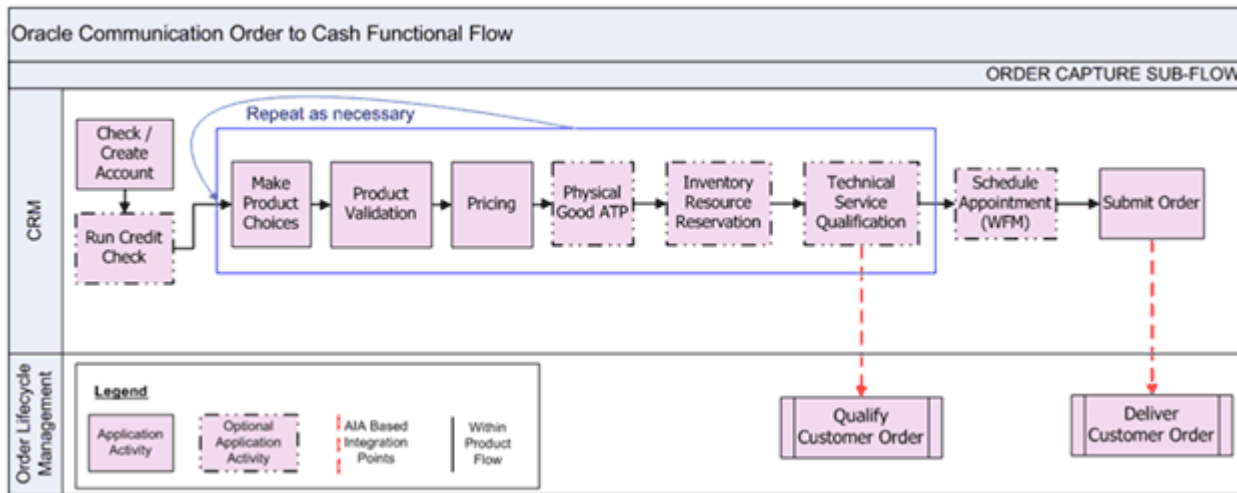


Figure 7–3 shows the typical order capture flow. The CRM swim lane shows the typical order activities for the CRM system. The OLM swim lane shows the Qualify Customer Order and Deliver Customer Order subflows connected to the CRM activities.

A typical order capture flow is as follows:

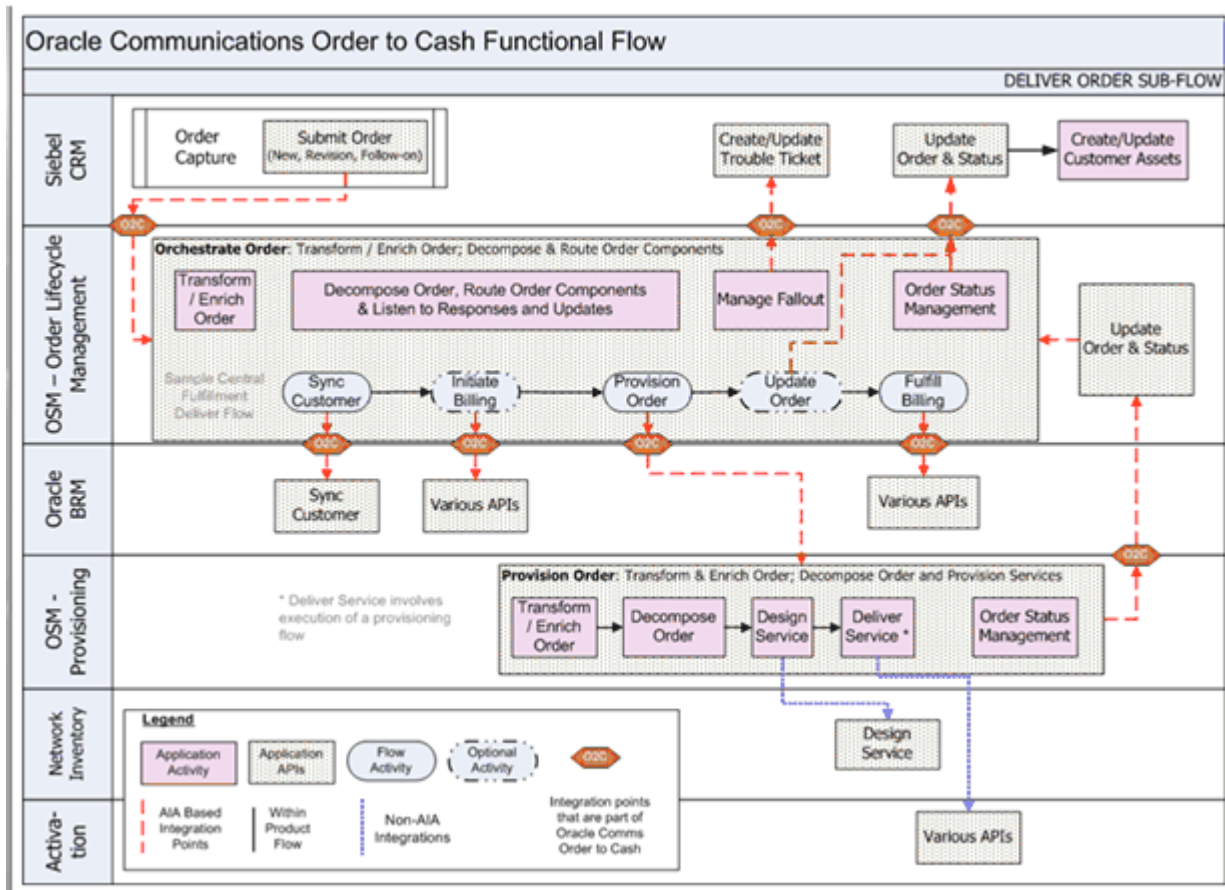
1. New customer accounts are created or existing customer accounts are updated. Customer information might also be captured earlier, for example, when an opportunity or quote is created or updated during order capture.
2. (Optional) A credit check is run on the new customer.
3. The customer makes product choices and the CRM system validates the products.
4. The CSP prices the selected products and product options.
5. (Optional) When physical goods are chosen, the CRM system checks the availability to purchase (ATP).
6. (Optional) For some services, such as phone numbers, the CRM system reserves the resource.
7. (Optional) The orders passes through technical service qualification
8. (Optional) An appointment with an engineer is scheduled for the customer.
9. The order is submitted and the delivery process starts.

About the Deliver Customer Order Subflow

Figure 7–4 shows six swim lanes, one for each of the following applications: Siebel CRM, OSM, BRM, OSM 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 integration points. See the legend in Figure 7–4

for other details.

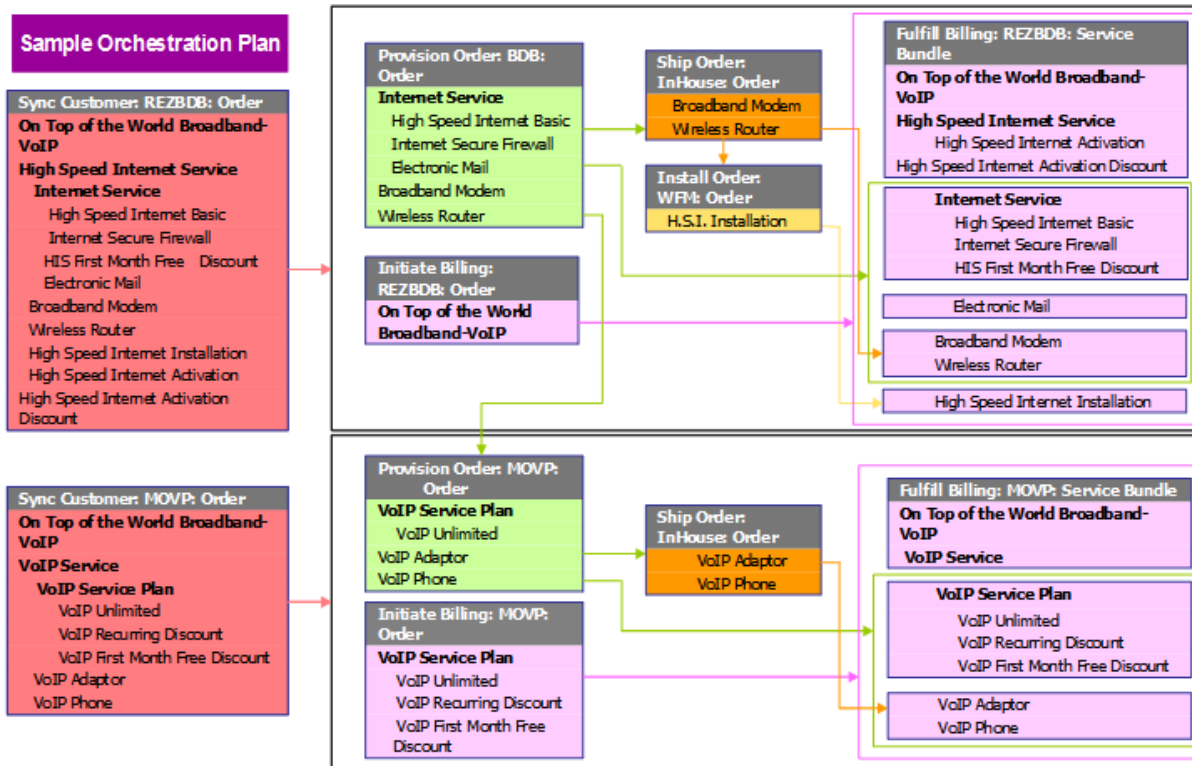
Figure 7-4 Deliver Customer Order Subflow



This flow starts with a new order, an order revision, future-dated order, or a follow-on order submitted from Siebel CRM to OSM. OSM performs the following activities:

1. Transforms and enriches the order by mapping order lines to fulfillment flows and enriching them with fulfillment metadata and other relevant data.
2. Decomposes and routes the order by dividing the order into suborders, which are called order components. Order components can have cross-order components, cross-order lines, and cross-order dependencies. The outcome of decomposition is an order orchestration plan that is executed at the computed fulfillment start time to meet the requested delivery date. The produced fulfillment flow orchestrates fulfillment requests using preconfigured fulfillment functions, such as synchronizing the customer into BRM, initiating and fulfilling billing, provisioning the order, shipping the order, and installing the order. The OSM decompose and route order function also generates compensation plans that are associated with revision orders. Figure 7-4 illustrates a simple flow; orchestration plans are typically more complex, as shown in Figure 7-5.

Figure 7-5 Complex Deliver Customer Order Subflow



3. Manages order fallout by creating trouble tickets in Siebel CRM.

The integration provides for detection, reporting, and resolution of order fulfillment fallout conditions such as validation, and fulfillment errors using Siebel CRM trouble tickets. System errors (such as an unreachable system) are handled differently.

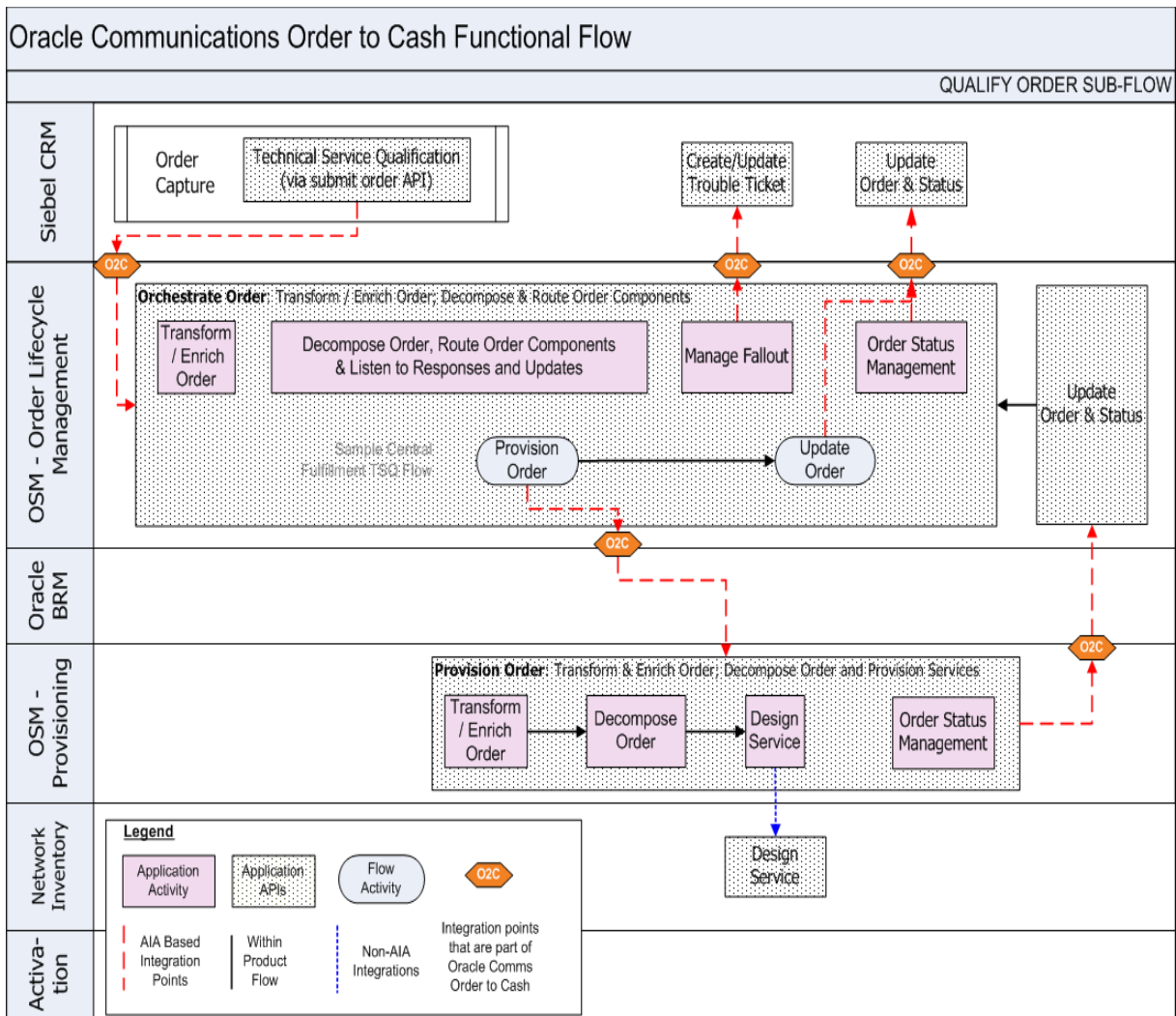
See "[Using Error Type to Control Response to Order Fallout](#)" for more information.

4. Manages order status by mapping fulfillment function responses to common statuses which are aggregated into order line statuses and order header status values. OSM updates Siebel CRM with relevant customer status and milestone values. It also updates Siebel CRM when order lines reach their point of no return to prevent the submission of new revisions.

About the Qualify Customer Order Subflow

Figure 7-6 shows six swim lanes, one for each of the following applications: Siebel CRM, OSM, BRM, OSM 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 Quality Customer Order Subflow

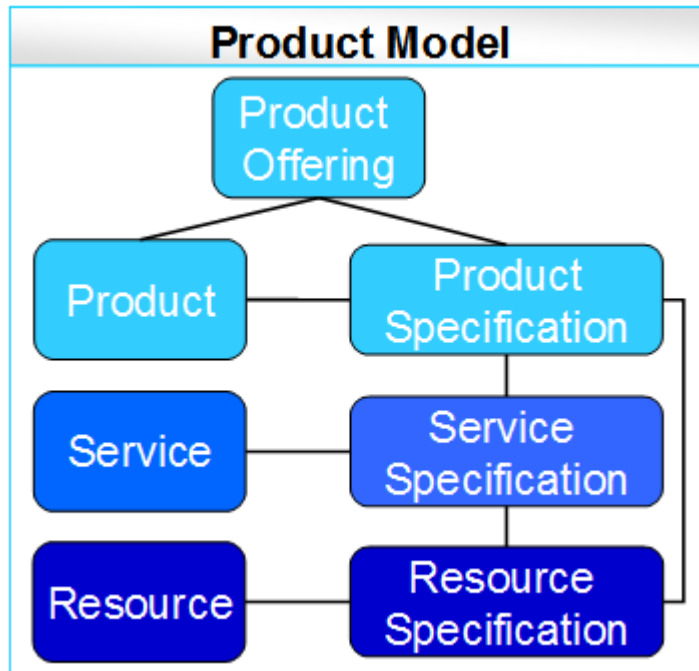


This flow starts with a request to qualify the technical validity of a customer order submitted from Siebel CRM to OSM. OSM performs the same four functions detailed for the Deliver customer order with one key distinction: the metadata used and the fulfillment flow produced is for qualifying the customer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

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.

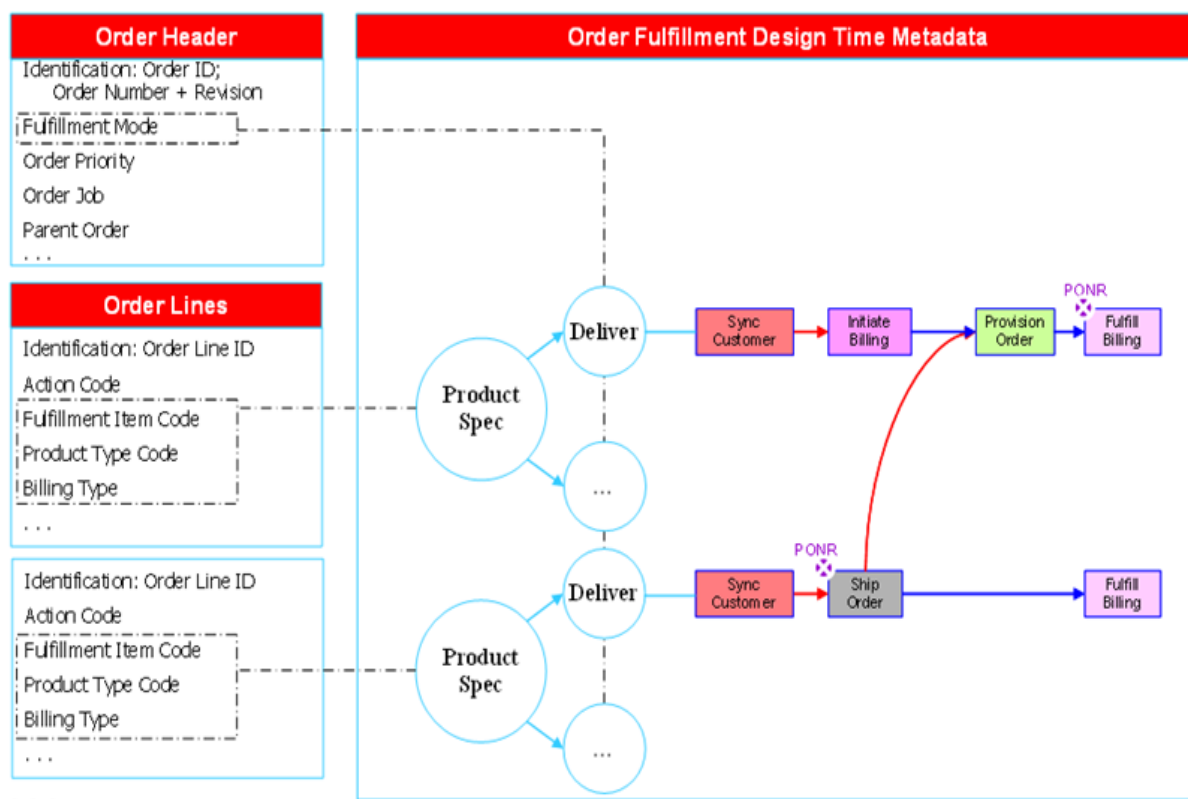
A key element of the Oracle methodology is a one-to-one mapping of every order line to a product specification. This approach is key to achieving fast time-to-market and low-cost operations. The Oracle solution facilitates this mapping by associating product offerings with a product class in Siebel CRM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

Order management systems act 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 order management 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.

Data Requirements

The data requirements for Siebel CRM orders for the process integration for OLM are as follows:

- An order must be of type Sales Order.

- Any price list specified on an order must match one created in Siebel CRM and configured in the PRICELIST domain value map (DVM). The default price list must also be configured in the **AIAConfigurationProperties.xml** file.
- If a price list is specified in the order header, any order lines that do not specify a price list will use the price list in the order header. If no price list is specified in the order header, each order line must specify a price list, with the exception of order lines for discounts synchronized from BRM as simple products in Siebel CRM. Price list information is not sent for billing discounts.
- 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.

See "[Assumptions and Constraints for the Bill Fulfillment Order Business Flow](#)" for more information.

- 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.
- The following enterprise business object (EBO) attributes are mandatory for integration with OSM:
 - Order header: Order ID, Order Number, Revision, Fulfillment Mode, Order Type
 - Order line: Line ID, Base Line ID, Action Code, Product Name, Product Type

Tip: The Sales Order EBO includes a vast set of attributes that are sufficient for most fulfillment systems, and it is extensible.

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.

The Process Sales Order Fulfillment business flow is enabled by either of the following Pre-Built Integration options:

- Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM) and Oracle Communications Order and Service Management (OSM) Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM, OSM, and Oracle Communications Billing and Revenue Management (BRM) Pre-Built Integration option

Process Sales Order Fulfillment Business Flow Overview

The Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM (the integration) uses the Process Sales Order Fulfillment business flow when a Siebel CRM user submits an order to OSM to process before submitting to bill fulfillment in BRM.

When a Siebel CRM user submits an order, Siebel CRM drops the order application business message (ABM) into a Java Message Service (JMS) queue and gives the control back to the Siebel CRM user, making the submit order event an asynchronous process. A JMS Consumer that listens to the JMS queue dequeues the message and invokes the Siebel CRM Application Business Connector Service (ABCS).

About Sales Orders

Sales orders are orders that purchase products and services for customers. Siebel CRM submits orders and the integration sends the orders to OSM. Orders from Siebel CRM are composed of an order header and order lines. The order header includes attributes applicable to the customer and to all order lines. Order lines apply to particular products or services and are composed of a subject and an action.

The order line subjects can include but are not limited to simple and customizable products, discounts (modeled as simple products), service bundles, promotions, and pricing event products (used with multi-event billing products). When order line items are fulfilled and provisioned, they are called **assets** in Siebel CRM.

Siebel CRM supports the following order line actions:

- **Add:** adds a new asset

- **Move-Add:** used when transferring an existing asset from one address to another to add the asset at the target location
- **Move-Delete:** used when transferring an existing asset from one address to another to delete the asset from the source location
- **Delete:** disconnects/cancels an existing asset
- **Update:** updates an attribute on an existing asset or product or service that has yet to be fulfilled
- **Suspend:** changes the status of an existing asset to **Suspended**
- **Resume:** changes the status of an existing asset from **Suspended** to **Active**

You can revise orders in Siebel CRM several times before submitting them. Siebel CRM tracks these revisions and each revision replaces any previous revisions. Siebel CRM internal revisions are not considered OSM revision orders because they are not submitted for fulfillment.

When you submit an order, the integration uses the attributes on the order to populate cross-reference tables and pass the fulfillment information to OSM and BRM.

When OSM receives the Siebel CRM order information from the integration, OSM determines the point of no return as set for the order items. An order past the point of no return is not yet complete, but you can no longer revise it.

For more information about the point of no return, see *OSM Concepts*, and for information about setting a point of no return, see *OSM Cartridge Guide for Oracle Application Integration Architecture*.

Siebel CRM and OSM use different terms to refer to the different types of order. This chapter uses the Siebel CRM term. [Table 8–1](#) defines and maps the terms from Siebel CRM to OSM. For more details about the Siebel CRM orders, see "[About Supplemental Orders](#)", "[About Follow-On Orders](#)", and "[About Future-Dated Orders](#)".

Table 8–1 Order Term Mapping

Siebel CRM Term	OSM Term	Description
Open order	In-flight order	An order that has been submitted to fulfillment but is not yet complete
Supplemental order	Revision order	A changed version of an in-flight/open order
Follow-on order	Follow-on order	A changed version of an in-flight/open order that has passed the point of no return. Fulfillment of the follow-on order waits until the fulfillment of the order item on which the follow-on order depends is complete.
Modify order	--	An order to modify the attributes of assets on a completed order. There is no direct correlate in OSM; such orders are treated as new orders.
Future-dated order	Future-dated order	An order scheduled to start at a future date

In Siebel CRM, **change order** is the category of orders that make changes to previous orders. This category includes supplemental orders, follow-on orders, and modify orders.

About Supplemental Orders

Supplemental orders are revised versions of open orders that have been submitted for fulfillment but have not yet passed the point of no return.

Siebel CRM allows only one pending supplemental order for each open order.

When you submit a supplemental order from Siebel CRM, OSM does the following:

1. Suspends the fulfillment flows associated with the revised order
2. Computes the changes for each order line
3. Creates a compensation plan for fulfillment activities that have occurred and that are affected by the revision. The compensation plan is merged with the fulfillment plan for the OSM revision order, and the revision fulfillment does not begin until completion or another revision is submitted.

See *Siebel Order Management Guide* for information about revising an order in Siebel CRM.

About Follow-On Orders

Follow-on orders are revised versions of open orders that have passed the point of no return but are not yet complete. Siebel CRM simulates the future completion of the open order to set up a dependency between the fulfillment of the open order and the processing of the follow-on order. When OSM receives a follow-on order that depends on an open order, it manages the dependency and does not process the follow-on order until the fulfillment of the order item on which the follow-on order depends is complete.

To ensure that the integration correctly updates Siebel CRM assets, do the following before creating a follow-on order:

- Check that you have submitted the base order, establishing correct order dependency in OSM. If you submit the follow-on order before submitting the base order on which it depends, OSM processes the follow-on order as a base order.
- Check that the base order is past the point of no return.
- Discard any pending supplemental orders that you have not yet submitted for the open order.

You can submit supplemental orders and additional follow-on orders to revise follow-on orders.

About Modify Orders

Modify orders are revised versions of complete orders. They modify installed Siebel CRM assets using the base order for those assets (the Siebel CRM documentation also calls them asset-based orders). You can submit modify orders only for orders that have been fulfilled and provisioned. OSM treats modify orders from Siebel CRM as new orders that modify the data created by the base order.

You can submit supplemental orders and follow-on orders to revise modify orders.

About Future-Dated Orders

A future-dated order is an order scheduled to start at a future date. Future-dated orders are created in Siebel CRM with the Due Date attribute set to a future date and submitted immediately to OSM. OSM manages the date that fulfillment starts. When you create a future-dated order, Siebel CRM simulates the future state of the asset.

See *OSM Concepts* for more information about OSM future-dated orders and *OSM Cartridge Guide for Oracle Application Integration Architecture* for more information about handling current, past, future, and requested but not provided delivery date-time values.

To avoid complex future asset states, Oracle recommends that you do not create multiple future-dated orders for the same asset and that you limit future-dated orders to one per customer. If you must create multiple future-dated orders for the same asset, follow these guidelines:

- Ensure that new future-dated orders do not invalidate previously-submitted future-dated orders.
- Create the orders in chronological order.
- When the requested delivery date for an order line is earlier than a future-dated order that you created previously, revise the previous order to ensure that it is based on the future state of the asset determined by the new future-dated order.

Supporting Multiple Price Lists on Orders

The integration supports using multiple price lists for products on a single order.

When you create an order in Siebel CRM, the price list assigned to the customer's account is automatically assigned to the order header. You can specify a different price list for the order header and for the individual order lines.

See *Siebel Pricing Administration Guide* for information about assigning a price list to a customer's account.

You create price lists in Siebel CRM, add the default price list to the **AIAConfigurationProperties.xml** file and add the default and any additional price lists to the PRICELIST domain value map (DVM) before creating products in BRM. See ["Working with Price Lists and Rate Plans at Design Time"](#) for more information about creating price lists at design time. OSM uses the price list information sent on a Siebel CRM order to initiate and fulfill billing in BRM using the correct rate plan.

Specifying Different Price Lists on New Orders

When you create a new order in Siebel CRM, the order can use the default price list for the order header, or you can specify a different one. You can also specify different price lists for the individual order lines. If you submit the order without specifying a price list for an order line, OSM populates the empty order line with the price list specified for the order header.

Orders in Siebel CRM must have at least a default price list in the order header. Oracle recommends that you extend Siebel to enforce this requirement.

When you submit an order that includes a customizable product, such as a service bundle, a marketing bundle, or a non-service-bundle customizable product, Siebel CRM automatically assigns the price list for the customizable product to all order lines for components of the customizable product in the sales order ABM. You cannot change the price list for the order lines for components of the customizable product.

[Table 8–2](#) shows an example of the order lines for a new Siebel CRM order. The table shows only the attributes relevant to this example.

Table 8–2 Example of Specifying Price Lists on a New Order

Line Number	Product	Action	Price List
1	Internet Access	Add	-
2	VoIP Service Bundle	Add	Premium Consumer Price List
2.1	VoIP Access	Add	-
2.2	VoIP Voicemail	Add	-

When you submit the order in [Table 8-2](#), Siebel CRM populates the price list for VoIP Access and VoIP Voicemail products with **Premium Consumer Price List**. When AIA passes the order to OSM, OSM populates the Internet Access product with the price list specified for the order header and sends the order through the integration to BRM for billing.

Changing Price Lists on Supplemental Orders

As part of a supplemental order for a new order in Siebel CRM, you can change price lists for existing lines that use the Add action.

- **Change price list for order header:** When you change the price list for the order header, OSM populates new and existing order lines without a specified price list with the new price list for the order header.
- **Change price list for order line:** When you change the price list for the order line, OSM updates the line item with the new price list.
- **Remove price list for order line:** When you remove the price list on the order line, leaving it empty, OSM populates the empty field with the price list specified for the order header.

[Table 8-3](#) shows an example of a revision of the order shown in [Table 8-2](#).

Table 8-3 Example of Changing Price Lists on a Supplemental Order

Line Number	Product	Action	Price List
1	Internet Access	Add	Premium Consumer Price List
2	VoIP Service Bundle	Add	-
2.1	VoIP Access	Add	-
2.2	VoIP Voicemail	Add	-

When you submit the order in [Table 8-3](#), OSM changes the price list for the Internet Access product to **Premium Consumer Price List** and changes the price list for the VoIP service bundle and its components to the price list specified for the order header. OSM sends the revised order through the integration to BRM for billing.

Changing Price Lists on Modify Orders

You can change price lists for installed assets as part of a modify order in Siebel CRM and as part of a supplemental order for a modify order as follows:

- **Change price list for order header:** When you change the price list for the order header for an existing asset and use the Add action to include new order lines without specifying a price list, OSM populates the price list for the order lines with the new price list for the order header.

If existing assets use a price list originally populated from the order header, OSM does not repopulate these when the price list for the order header is changed. You must change the price list for existing assets manually at each order line.

- **Change or remove price list for order line:** Because Siebel CRM does not send prior price list information to OSM, changes to price lists on order lines are ignored. To change or remove price lists for order lines on a modify order, you must first manually override the price for line items as follows:
 1. In Siebel CRM, enter the price in the **Manual Price Override** field for the line item. See the discussion of entering a manual discount for an individual line item in *Siebel Order Management Guide* for more information.

The line action changes to Update.

2. Assign the new price list to the order line or remove the price list, leaving it empty.
3. If you are changing the price list for a service bundle, marketing bundle, or non-service-bundle customizable product, repeat step 1 for each component of the bundle.
4. Submit the order.

If you leave the price list empty, OSM populates the empty field with the price list specified for the order header.

Table 8–4 shows an example of the line items for a change order to change the price lists of the assets installed by the order in Table 8–3.

Table 8–4 Example of Changing Price Lists of Installed Assets on a Modify Order

Line Number	Product	Action	Price List
1	Internet Access	Update	Consumer Price List
2	VoIP Service Bundle	Update	Premium Consumer Price List
2.1	VoIP Access	Update	-
2.2	VoIP Voicemail	Update	-

When you create the order in Table 8–4, you must manually override the price of each line item so that the line action changes to Update. When you submit the order, Siebel CRM populates the price list for the VoIP Access and VoIP Voicemail products with **Premium Consumer Price List**. When AIA passes the order to OSM, OSM updates the price lists for the installed Internet Access and VoIP Service Bundle assets. OSM sends the order through the integration to BRM for billing.

Note: Siebel CRM does not track price lists for assets. When you update the price list of an order line on a change order, Siebel CRM only sends the new price list value. If you are using an order management system other than OSM, it must recognize that the Update action for a line with a non-empty price list attribute value means that the price list attribute has changed.

Supporting Order Priorities

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

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

The submission process for orders is the same for new orders, revision orders, and follow-on orders. The CSR selects a priority when submitting an order.

Siebel CRM provides and maps the priority values as shown in Table 8–5.

Table 8–5 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. You can extend Siebel CRM to support priority values other than those described in [Table 8–5](#).

Assumptions and Constraints for the Process Sales Order Fulfillment Business Flow

The assumptions and constraints for the Process Sales Order Fulfillment business flow are as follows:

- Siebel CRM implements service points as assets and you typically uploaded them into Siebel CRM from external sources. You should manage service points in a common place and share them 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.sdbv
 - 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 AIA cross-reference exists for this attribute.
- When you create a change order, leave it pending in Siebel CRM, and submit it at a later date, Siebel CRM ensures that the change order data is up to date with the actual data from the installed assets. Any customization of Siebel CRM or integration with a different CRM system must also ensure that pending orders are up to date before submitting them.
- If you submit a follow-on order before submitting the base order on which it depends, OSM processes this follow-on order as a base order. Submit base orders first to establish the follow-on order dependency in OSM.
- Mixing future-dated, follow-on, and revision orders requires a well-trained CSR because some scenarios could produce unintended results.
- Siebel CRM can capture revisions to order Due Date in Siebel CRM (Requested Delivery Date in Oracle AIA) and submit them to Oracle OSM.
- Revising the Requested Delivery Date for an order only affects OSM if the base order did not start fulfillment by the time OSM received the revision.

Implementing the Process Sales Order Fulfillment Business Flow

This chapter describes the Process Sales Order Fulfillment business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services. It also describes how the integration supports order priorities and multiple price lists.

Process Sales Order Fulfillment Business Flow Overview

The following Pre-Built Integration options enable the Process Sales Order Fulfillment business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option

The Oracle Communications Order to Cash for Siebel CRM and BRM Pre-Built Integration option includes a Test Orchestration Process (TOP) to test the ready-to-use order flow. You must replace the TOP with your own order management system. See *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations* for more information.

The Process Sales Order Fulfillment business flow supports the following integration flow:

- Submitting Orders from Siebel CRM to OSM

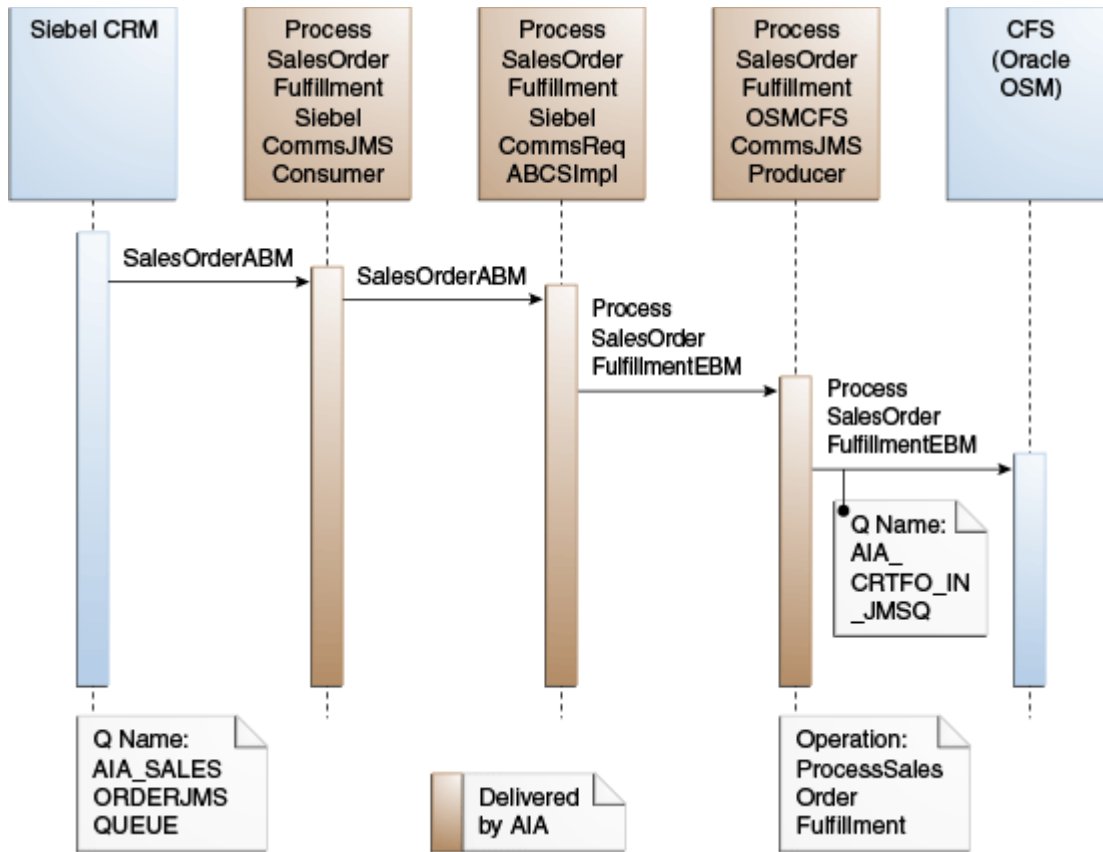
Submitting Orders from Siebel CRM to OSM Integration Flow

This integration flow uses the following services:

- ProcessSalesOrderFulfillmentSiebelCommsJMConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

[Figure 9-1](#) illustrates an order submitted from Siebel CRM to OSM using AIA.

Figure 9–1 Siebel CRM to OSM CFS Sequence Diagram



When a new order process is initiated, the following events occur:

1. A Siebel CRM user submits a new order.
2. Siebel CRM creates an application business message (ABM) called SalesOrderABM with all the sales order details and enqueues the ABM in the AIA_SALESORDERJMSQUEUE queue.
3. The ProcessSalesOrderFulfillmentSiebelCommsJMSSConsumer service dequeues the ABM and passes it on to the ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl service.
4. The ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl service transforms the ABM into an enterprise business message (EBM) called ProcessSalesOrderFulfillmentEBM and routes the EBM to the ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer service.

The Siebel CRM order structure does not support multiple charge types for a single order line, but the order enterprise business object (EBO) structure does. For this reason, the order lines referencing a complex product of billing type *Subscription* and its component products of billing type *Event* (multi-event billing products) are transformed into a single EBO order line referencing a product with multiple charge types.

5. The ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer service wraps the EBM into OSM CreateOrder message format and enqueues the CreateOrder message into the AIA_CRTFO_IN_JMSQ.

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

- JMSPriority
 - URI -- /osm/wsapi
 - _wls_mimehdrContent_Type -- text/xml; charset=utf-8
6. The store and forward mechanism forwards the CreateOrder message from the AIA WebLogic server to the OSM WebLogic server.

OSM receives the CreateOrder message and decomposes the order, creating EBMs to send for fulfillment and provisioning.

See "[Understanding the Process Integration for Order Lifecycle Management](#)" for more information on how OSM processes the order.

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 CRM 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.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information on system and business errors.

The following services are involved:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer

Table 9–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer passes the Siebel CRM message to ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl, which invokes transformation logic to convert that Siebel CRM message into an EBM. The EBM is then routed to ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer.	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 CRM.

See "[Configuring Oracle AIA Processes for Error Handling and Trace Logging](#)", [Configuring Fault Policies to Not Issue Rollback Messages in Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack](#) for more information about rollback procedures.

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 CRM queue and to make manual changes to seeded priority values.

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

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

See the Siebel CRM product documentation for Lists of Values for more information.

These steps describe how the integration handles order priorities:

1. When the order is submitted, Siebel CRM 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 ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl. This process does a lookup on the SALESORDER_PRIORITY domain value map (DVM) and populates the ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/FulfillmentPriorityCode.
3. The ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer 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 OSM CFS.
5. OSM CFS and OSM Provisioning honor the priority through internal mechanisms. Higher priority orders are fulfilled and provisioned first, followed by lower priority orders.
6. 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. OSM and Oracle AIA, unlike Siebel CRM, follow the same values for JMS priorities.

Supporting Multiple Price Lists

The Process Sales Order Fulfillment business flow supports multiple price lists on orders as follows:

1. A Siebel CRM user submits an order with separate price lists for the order lines. Siebel CRM creates an application business message (ABM) containing the order information, including the price lists specified for the order header and order lines. While creating the ABM, Siebel CRM automatically populates empty price list order lines for the components of customizable products with the price list from the customizable product order line.
2. Siebel CRM drops the ABM into the `AIA_SALESORDERJMSQUEUE` queue.
3. The `ProcessSalesOrderFulfillmentSiebelCommsJMSProducer` service receives the ABM and passes it to the `ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl` service.
4. The `ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl` service looks up the Siebel CRM row ID in the `PRICELIST` domain value map (DVM) and populates the `ProcessSalesOrderFulfillmentEBM` message with the corresponding price list information.
5. The `ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl` service passes the `ProcessSalesOrderFulfillmentEBM` message to the `ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer` service.
6. The `ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer` service wraps the `ProcessSalesOrderFulfillmentEBM` message in `CreateOrder` message format and drops it into the `AIA_CRTFO_IN_JMSQ` queue.
7. OSM receives the `CreateOrder` message and transforms it into a `ProcessFulfillmentOrderBillingEBM` message. During this transformation, OSM populates empty price list order lines with the price list from the order header.
8. OSM passes the `ProcessFulfillmentOrderBillingEBM` message on for billing. See ["Bill Fulfillment Order Business Flow Overview"](#) for information about how this message is used.

Siebel CRM Interfaces

The Process Sales Order Fulfillment business flow uses the following Siebel CRM interface:

- `SISOMBillingSubmitOrderWebService`

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

See *Siebel Order Management Guide Addendum for Communications* for more information about web services.

Industry Oracle AIA Components

The Process Sales Order Fulfillment business flow 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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Process Sales Order Fulfillment business flow:

- ProcessSalesOrderFulfillmentSiebelCommsJMConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

ProcessSalesOrderFulfillmentSiebelCommsJMConsumer

The ProcessSalesOrderFulfillmentSiebelCommsJMConsumer service is implemented as a Mediator process.

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

ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl

The ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl 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 CRM application. This service is the Siebel CRM ABCS implementation, which converts the Siebel CRM 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, organization ID, and product or discount 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 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, address ID, balance group ID, and asset ID between Siebel CRM values and generated common values.

This service also looks up the PRICELIST DVM to populate the price list common IDs for the Siebel CRM price list row IDs in the sales order EBM.

See [Table 25-2, "Order Lifecycle Management Cross-References"](#) for more details about the cross-references created by this service.

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 OSM) and OSM's CreateOrder envelope.

This service has one operation: ProcessSalesOrderFulfillment. It takes the ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducerRequestMessage as input.

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.

The Synchronize Fulfillment Order Billing Account business flow is enabled by the following Pre-Built Integration option of the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration):

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

Overview of the Synchronize Fulfillment Order Billing Account Business Flow

The Synchronize Fulfillment Order Billing Account business flow lets you create only the customer data necessary for order fulfillment in BRM without overburdening BRM with all of the customer information from Siebel CRM.

This business flow provides the following service:

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

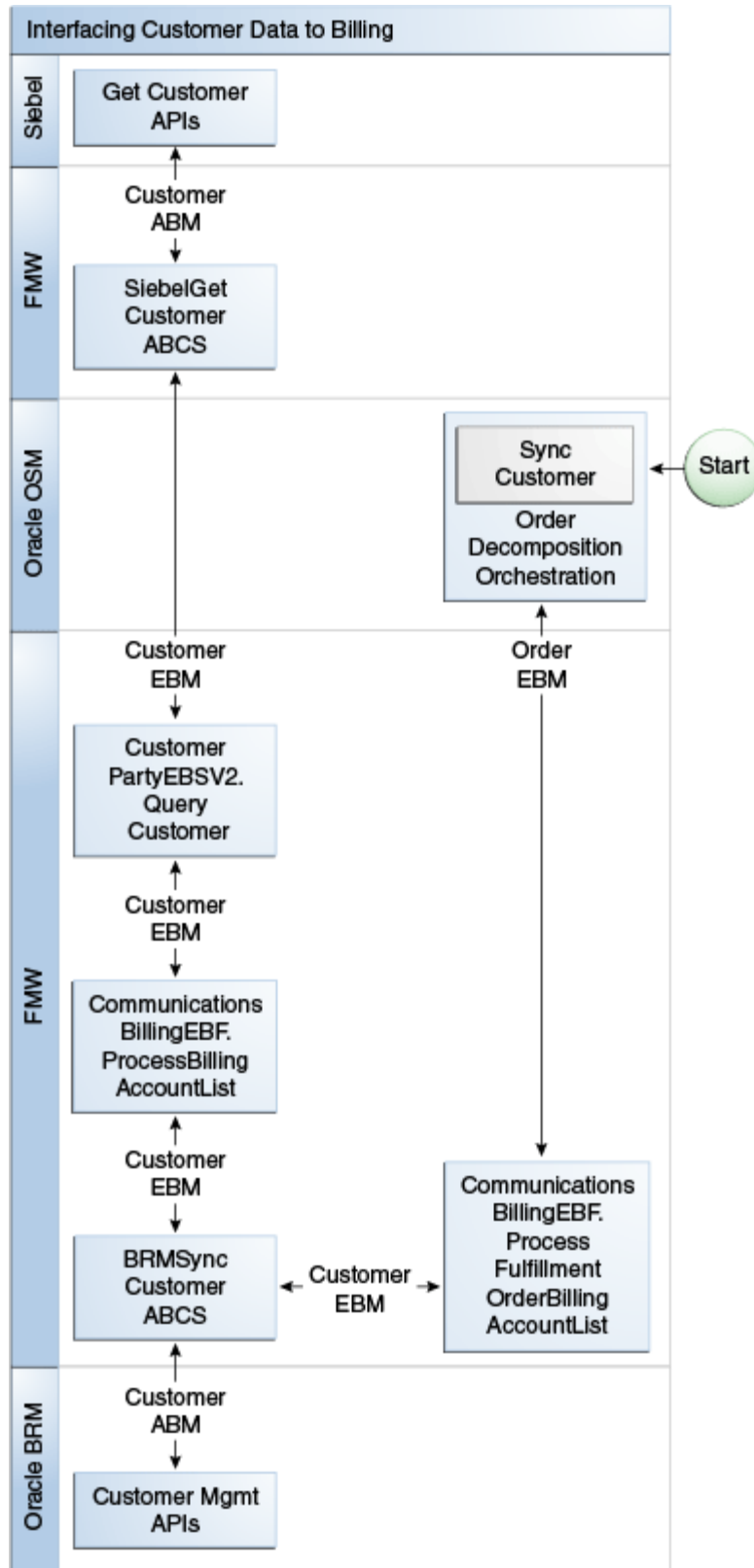
This service takes an order as input and collates order data and then calls other enterprise billing services (from the process integration for Customer Management) to create accounts and their components (such as billing preferences and payment methods) in a BRM instance. This service can be invoked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

See "[Expectations from an Order Management System for Billing Integration](#)" for more information about calling this service from an order management system other than OSM.

See "[Understanding the Process Integration for Customer Management](#)" for more information about the Customer Management process integration.

[Figure 10-1](#) shows how the integration interfaces orders to create customer data in BRM.

Figure 10–1 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 order lines for customer data collation as follows:

- 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, /billinfo objects, and /payinfo objects in BRM.

Customer creation that occurs in BRM as part of order fulfillment using the InterfaceOrderToCustomerEBF service cannot be undone. The service does not support inactivating or deleting accounts, /billinfos objects, or /payinfo objects in BRM.

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

When you call the service again with different customer data than before, the service detects the delta and creates just the account, /billinfo objects, and /payinfo objects that do not exist in 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 BRM. This creates a paying hierarchy in billing.

A paying hierarchy, when created, cannot be undone by the cancelling 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.

See "[About the Create/Sync Account Integration Flow](#)" for more information about what Siebel CRM account information is sent to BRM.

If you are using an order management system other than OSM, [Table 10-1](#) summarizes what is expected for the action on the line. OSM and OSM AIA cartridges conform to these expectations.

Table 10–1 Actions on Order Line Expectations Summary

Original Action on Order Line	Type of Order (Revision or New)	Type of Revision	Action Set by Order Management System on Compensation Order Line	Comments
ADD	New	Not applicable	ADD	None
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	<p>The order management system creates customer data in BRM if it does not already exist.</p> <p>If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM.</p> <p>The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.</p> <p>Prior value fields are used in flagging and determining that a paying hierarchy change has occurred.</p>
ADD	Revision	Cancellation. Manifests as a missing line on the revision.	DELETE	<p>This action is ignored.</p> <p>If the original ADD line added a new account, /billinfo object, and /payinfo object, and the revision cancels the request for the new purchase, the account, /billinfo, and /payinfo are not inactivated or deleted.</p> <p>If the original ADD line created a paying hierarchy and the revision cancels the request for the new purchase, then the paying hierarchy stays in place.</p>
UPDATE	New	Not applicable	UPDATE	Expects prior value fields to be populated.

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

Original Action on Order Line	Type of Order (Revision or New)	Type of Revision	Action Set by Order Management System on Compensation Order Line	Comments
UPDATE	Revision	No changes to service account, billing account, or billing profile	NONE	No changes.
UPDATE	Revision	Changes to service account, billing account, or billing profile	UPDATE	The order management system creates customer data in BRM if it does not already exist. If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM. 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 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. If a hierarchy has been updated, this reverts that update.
MOVE-ADD	New, 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	Type of Order (Revision or New)	Type of Revision	Action Set by Order Management System on Compensation Order Line	Comments
MOVE-ADD	Revision	No changes to service account, billing account, or billing profile	NONE	No changes.
MOVE-ADD	Revision	Changes to billing account or billing profile	MOVE-ADD	The order management system creates customer data in BRM if it does not already exist. If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM. 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 "-" (The line is canceled).	MOVE-ADD	If the original MOVE-ADD line created a new account and billing profile in 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. If a hierarchy has been updated, this reverts that update.

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.

See the discussion of billing management in *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide* for more information.

Assumptions and Constraints for the Synchronize Fulfillment Order Billing Account Business Flow

See "[Assumptions and Constraints for the Bill Fulfillment Order Business Flow](#)" for information on this business flow.

Implementing the Synchronize Fulfillment Order Billing Account Business Flow

This chapter describes the Synchronize Fulfillment Order Billing Account business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using BRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Overview of the Synchronize Fulfillment Order Billing Account Business Flow

The following Pre-Built Integration option enables the Synchronize Fulfillment Order Billing Account business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

The Synchronize Fulfillment Order Billing Account business flow supports the following integration flow:

- Interfacing Orders to Create Customer Data in BRM

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

See "[Understanding the Process Integration for Customer Management](#)" for more information about the Create/Sync Customer Account integration flow.

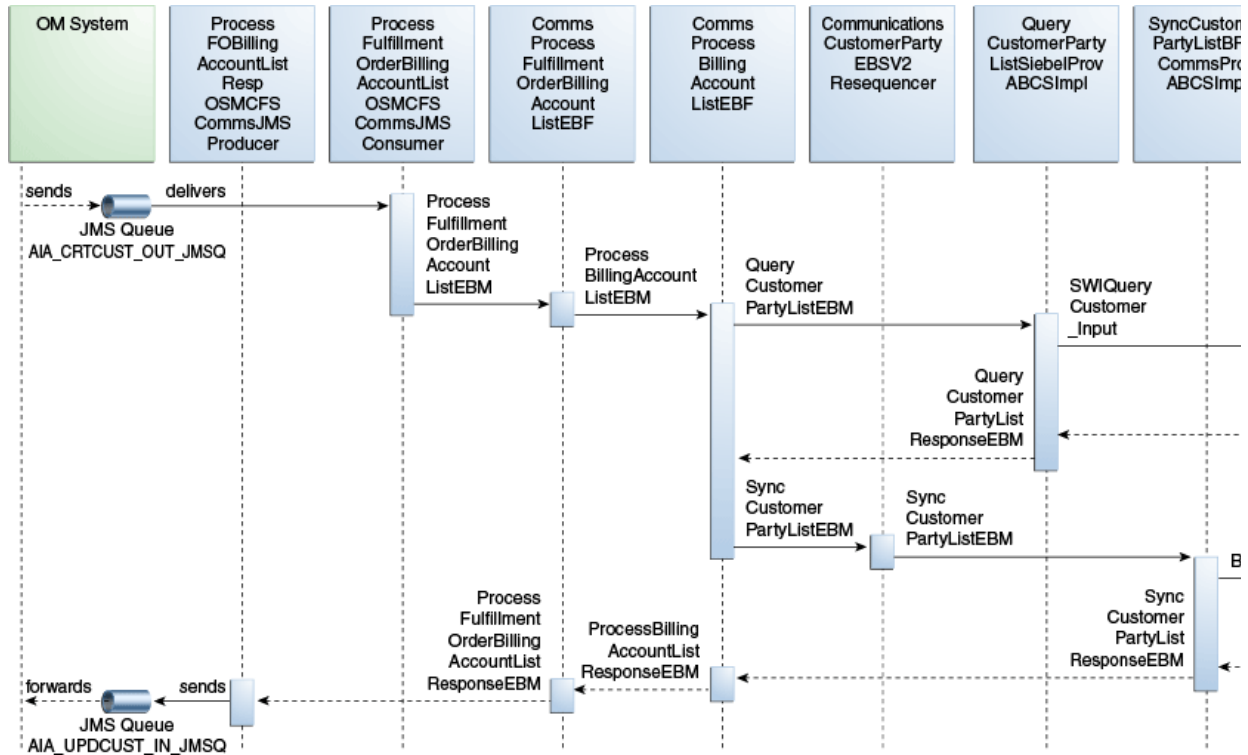
Interfacing Orders to Create Customer Data in BRM Integration Flow

This integration flow uses the following services:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSProducer
- From the process integration for Customer Management (see "[Integration Services](#)" for more information):
 - CommunicationsCustomerPartyEBSV2Resequencer (from the Create/Sync Customer Account integration flow)

- QueryCustomerPartyListSiebelProvABCSImplV2 (from the Create/Sync Customer Account integration flow)
- SyncCustomerPartyListBRMCommsProvABCSImpl

Figure 11-1 Interfacing Orders to Create Customer Data in BRM Sequence Diagram



This flow progresses as follows:

1. OSM drops the message into the AIA_CRTCUST_OUT_JMSQ JMS queue.
2. The ProcessFulfillmentOrderBillingAccountListOSMCFSCOMMSJMSConsumer services picks up the message and routes it to the CommsProcessFulfillmentOrderBillingAccountListEBF service.
3. The CommsProcessFulfillmentOrderBillingAccountListEBF service extracts the relevant customer data to create a ProcessBillingAccountListEBM message and routes it to the CommsProcessBillingAccountListEBF service.
This leverages the Create/Sync Customer Account integration flow.
4. The CommsProcessBillingAccountListEBF service 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 BRM
5. The CommsProcessBillingAccountListEBF service routes the message to the QueryCustomerPartyListSiebelProvABCSImplV2 service (a core service implemented in the MDM Pre-Built Integration).
6. The QueryCustomerPartyListSiebelProvABCSImplV2 service prepares the application business message (ABM), which is required to invoke the Siebel CRM SWI_Customer_Party_Service service.

This query service invokes the Siebel CRM database, fetches the account details, and replies to the QueryCustomerPartyListSiebelProvABCSImplV2 service with a response ABM.

7. This response ABM is transformed to the QueryCustomerPartyListResponseEBM message and is sent back to the CommsProcessBillingAccountListEBF service.
8. The CommsProcessBillingAccountListEBF service instantiates the SyncCustomerPartyListBRMCommsProvABCSImpl service.
9. This SyncCustomerPartyListBRMCommsProvABCSImpl service calls the PCM_OP_CUST_COMMIT_CUSTOMER BRM opcode to create an account. To update an existing account, the service calls either the PCM_OP_CUST_UPDATE_CUSTOMER opcode or the PCM_OP_CUSTCARE_MOVE_ACCT opcode.
10. If an account is successfully created or updated, the SyncCustomerPartyListResponseEBM message is sent back to the CommsProcessBillingAccountListEBF service in an asynchronous delayed response mode.
11. The CommsProcessBillingAccountListEBF service then sends the ProcessBillingAccountListResponseEBM response message to the CommsProcessFulfillmentOrderBillingAccountListEBF service in an asynchronous delayed response mode.
12. The CommsProcessFulfillmentOrderBillingAccountListEBF service drops a message into the AIA_UPDCUST_IN_JMSQ store-and-forward queue where OSM is notified of the SyncCustomer status.

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.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information on system and business errors.

The following services are involved:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCSImplV2
- CommunicationsCustomerPartyEBSV2Resequencer
- SyncCustomerPartyListBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingAccountListRespOSMCFSCCommsJMSProducer

Table 11–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessFulfillmentOrderBillingAccountListOSMCFSCCommsJMSC onsumer routes the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which extracts relevant customer data and then routes message to CommsProcessBillingAccountListEBF. The message is then routed to QueryCustomerPartyListSiebelProvABCImplV2, which fetches account details and a response is sent back to CommsProcessBillingAccountListEBF that invokes CommunicationsCustomerPartyEBSV2Resequencer.	AIA cross-reference entries for some of the Siebel CRM 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 SyncCustomerPartyListBRMCommsProvABCImpl, which invokes 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. ProcessFulfillmentOrderBillingAccountListRespOSMCFSCCommsJMSProducer 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 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.

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

BRM Interfaces

The Synchronize Fulfillment Order Billing Account business flow uses the following BRM interfaces:

- PCM_OP_CUST_COMMIT_CUSTOMER
- PCM_OP_CUSTOMER_UPDATE_CUSTOMER

- PCM_OP_CUSTCARE_MOVE_AACT

See "[BRM Interfaces](#)" for information about the BRM interfaces used by the Create/Sync Account integration flow.

Industry Oracle AIA Components

The Synchronize Fulfillment Order Billing Account business flow uses the following communications industry-specific Oracle AIA components:

- FulfillmentOrderEBO
- 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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

See "[Industry Oracle AIA Components](#)" for more information about the industry Oracle AIA components used by the Create/Sync Account integration flow.

Integration Services

The following services are delivered with the Synchronize Fulfillment Order Billing Account business flow:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSProducer

See "[Integration Services](#)" for more information about the integration services delivered with the Create/Sync Account integration flow.

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.

CommsProcessFulfillmentOrderBillingAccountListEBF

The CommsProcessFulfillmentOrderBillingAccountListEBF is implemented as an asynchronous BPEL process. It performs these operations:

- Receives the ProcessFulfillmentOrderBillingAccountListEBM from OSM with the target 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, 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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about EBFs.

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 CRM web service with this payload through `QueryCustomerPartyListSiebelProvABCSImpl`.
- Receives a response `QueryCustomerPartyListResponseEBM`, constructs a `SyncCustomerPartyListEBM` message and then invokes and routes the message to `QueryCustomerPartyListSiebelProvABCSImplV2`.

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 BRM.

Understanding the Bill Fulfillment Order Business Flow

This chapter explains how the Bill Fulfillment Order business flow interfaces orders from Siebel customer relationship management (Siebel CRM) to Oracle Communications Billing and Revenue Management (BRM) through an order management system like Oracle Communications Order and Service Management (OSM).

The Bill Fulfillment Order business flow is enabled by the following Pre-Built Integration option of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, OSM, and BRM (the integration):

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

Bill Fulfillment Order Overview

In the Bill Fulfillment Order business flow, OSM uses orders submitted in Siebel CRM to create transaction data and bill customers in BRM.

You can also use an order management system other than OSM to call these services, provided it meets certain expectations. See "[Expectations from an Order Management System for Billing Integration](#)" for more information.

The integration supports purchasing the following in BRM in new orders or change orders:

- Products of type **Item** that apply to an account (for example, promotion penalty charges) or service (for example, one-time charges).
- Products of type **Subscription** that apply to an account (for example, charges for mailing a monthly paper invoice) or service (for example, wireless service).
- Discounts of type **Subscription** that apply to an account (for example, account-level discounts) or service (for example, a free minutes discount).

BRM products and discounts design-time data is synchronized to Siebel CRM by the process integration for product lifecycle management (PLM). See "[Understanding the Process Integration for Product Lifecycle Management](#)" for more information.

See "[Supporting MACD Actions and Attribute Changes](#)" for examples of supported products.

About Interfacing Orders to BRM

This section describes the actions taken by the process integration when interfacing new or change orders to BRM.

Creating or Updating Service Instances

When interfacing orders, the integration creates or updates service instances and purchased product and discount instances in BRM.

The integration supports the following actions: Add, Delete, Update, Suspend, Resume, MoveAdd, and MoveDelete.

It communicates updates to the service identifier, billing account, billing profile, and price changes on existing services.

When an old product is canceled as part of service cancellations or promotion upgrade or downgrades, whether the customer gets a refund for (billed) monthly charges or whether the refund is prorated depends on product level controls in BRM.

The integration lets you change the service identifier, billing account, and billing profile as part of a Move-Add/Move-Delete transaction. The integration does not support purchasing new products or canceling existing products as part of a Move transaction.

Transferring a service from one location to another in Siebel CRM results in lines with the action of Move-Add and Move-Delete.

See ["Supporting MACD Actions and Attribute Changes"](#), ["Examples of Changing the Paying Parent on Subordinate Accounts"](#) and ["Mapping Billing Dates"](#) for more information.

Communicating Pricing Information

When interfacing orders, the integration communicates pricing information such as price or discount overrides, discounts, and onetime and penalty charges.

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 you want the new price to go into effect immediately, then the Siebel CRM user must disconnect the product and then add it with the new price.

Communicating One-time and Penalty Charges

One-time 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 BRM as account-level charges. See ["About One-Time Charges for Service Activation and Changes to Promotions and Service Bundles"](#) for information about one-time and penalty charges in BRM.

Siebel CRM supports defining charges for Suspend, Resume, Move, and Delete actions. You can extend Siebel CRM beyond the ready-to-use support to define charges for other actions such as Update.

For example, you can charge a customer a fee for requesting a change to their phone number or billing profile. The order billing integration 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 CRM

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 CRM order) and using the charge parent line (on the order EBM) are processed by the billing interface and applied to the respective service instance.

The one-time 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, a 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.

See "[About Service Bundles](#)" for more information about service bundles and "[Supporting Balance Groups](#)" for more information about service-level balance groups.

If the application business connector service (ABCS) that transforms the Siebel CRM order application business message (ABM) to the order EBM is unable to resolve the base line that a new order or change order one-time 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.

Communicating Pricing or Discount Overrides

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 in 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 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, BRM allows a single override price. If, for example, a 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, Siebel CRM 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.

See *Siebel Order Management Guide Addendum for Communications* for more information about using the pricing commit type and dynamic discount method.

Communicating Price List Information

Orders submitted from Siebel CRM can specify a separate price list in the order header and at each order line. The integration communicates the price list ID from the creation of a sales order in Siebel CRM to orchestration and provisioning in OSM and to fulfillment in BRM. See ["Supporting Multiple Price Lists"](#) for more information about how the integration passes price list information from Siebel CRM to OSM.

The integration gets price list information from OSM in a `ProcessFulfillmentOrderBillingEBM` message and passes it to BRM as an ABM. The ABM contains price list IDs for the order header and each order line. BRM uses the price list IDs and associated rate plans to charge the appropriate amount for the products or services purchased on the order lines.

Communicating Service Identifiers

When interfacing orders, the integration communicates the service identifiers on the service bundle line in Siebel CRM to BRM. For telephony services, the service identifier is used as the phone number. For nontelephony service, it is used as the login and password.

Communicating Siebel CRM Promotion Information

To allow 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: If the purchase date from the promotion order line, it is used. If not, the request date is used. If neither is available, BRM uses the current date by default.
- 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 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.

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.

Rolling Back Transactions

The service that interfaces the order to 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.

See "[Understanding the Process Integration for Order Fallout Management](#)" for more information about order fallout.

Supporting Balance Groups

The integration supports service-level and account-level balance groups.

A balance group is an object in the BRM database used for tracking account balances and bills. When you submit a Siebel CRM order, the integration synchronizes service bundles as service instances in BRM and BRM tracks the balances for these services in balance groups. The billing profiles specified on the order in Siebel CRM are synchronized as bill units (**billinfo** objects) in BRM.

For more information about Siebel CRM orders, see "[About Sales Orders](#)" and *Siebel Order Management Guide*.

When the integration creates a customer account in BRM during the Create/Sync Customer Account integration flow, it also creates a default account-level balance group pointing to a default bill unit associated with the primary billing profile for the account.

By default, the integration enables service-level balance groups to track the balances for each service separately. You can disable service-level balance groups to track all of the services on an account together in the default account-level balance group.

The default account-level balance group is used whether you enable or disable service-level balance groups. See "[How BRM Tracks Account-Level Products in the Default Account-Level Balance Group](#)" for more information about how the default account-level balance group is used when service-level balance groups are enabled, and "[Working with Service-Level Balance Groups Disabled](#)" for more information about how the default account-level balance group is used when service-level balance groups are disabled.

Disabling Service-Level Balance Groups

To disable service-level balance groups:

1. Open the `AIA_home/ia_instances/AIA_Instance_Name/AIAMetaData/config/AIAConfigurationProperties.xml` file.

2. Search for the following element:

```
<Property name="O2C.AccountLevelBalanceGroup">False</Property>
```

3. Set the `O2C.AccountLevelBalanceGroup` property to **True**:

```
<ModuleConfiguration moduleName="BalanceGroupParameters">
  <Property name="O2C.AccountLevelBalanceGroup">True</Property>
</ModuleConfiguration>
```

Note: The `O2C.AccountLevelBalanceGroup` property is a system-level property. You enable or disable it for all accounts and services in the system.

4. Save and close the file.
5. Load the updated file to the Metadata Services (MDS) repository. See the discussion of updating MDS in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information.

If the `O2C.AccountLevelBalanceGroup` property does not exist in the properties file, service-level balance groups are disabled. You must add the property and set it to **False** if you want to enable service-level balance groups. For information about the additional steps required when adding properties to the **AIAConfigurationProperties.xml** file, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

If you enable service-level balance groups in an environment that has already processed orders, any services purchased when service-level balance groups were disabled continue to be tracked in the default account-level balance group. You cannot transfer these services to different accounts or assign them different billing profiles. To track these services in their own service-level balance groups, you must modify the services directly in BRM using opcodes.

If you disable service-level balance groups in an environment that has already processed orders, any services purchased when service-level balance groups were enabled continue to be tracked in their own service-level balance groups. You can still transfer these services to different accounts and assign them different billing profiles, but BRM tracks all new services under the account-level balance group and you cannot transfer them.

Working with Service-Level Balance Groups Enabled

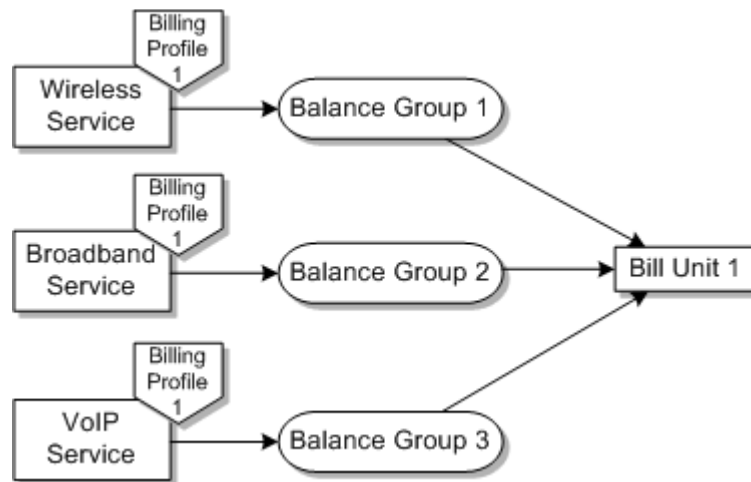
When you work with service-level balance groups enabled, BRM tracks each service under its own balance group. Tracking services in service-level balance groups lets your customers do the following:

- Track services individually
- Transfer services from one account to another

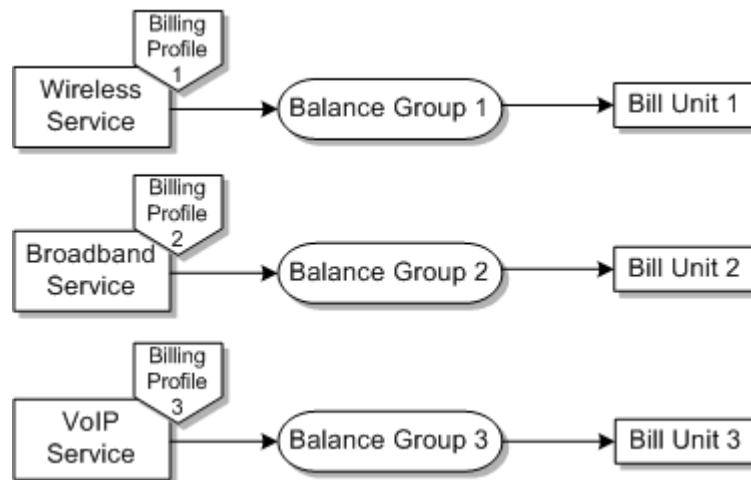
How BRM Tracks and Bills New Services in Service-Level Balance Groups

When you purchase multiple new services on one order, BRM tracks each service in a separate balance group. BRM bills the services based on which billing profile you assign each service. You can choose from the following options for billing services:

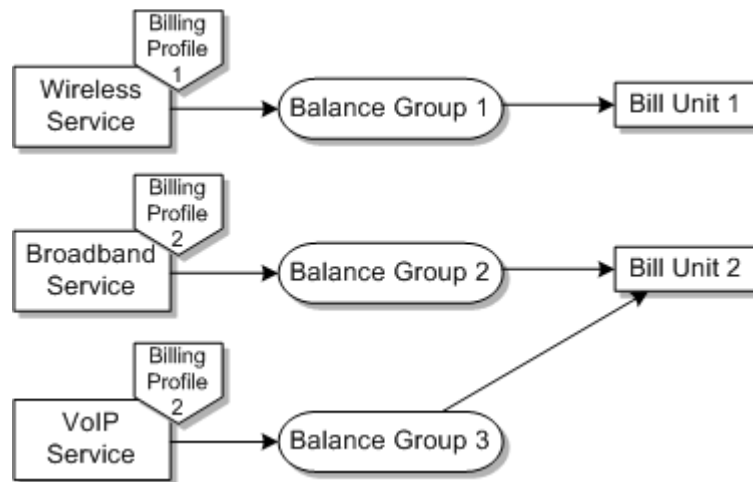
- **Billing all services together:** You assign all services the same billing profile. When you submit the order, BRM tracks each service in a separate balance group and the balance groups all point to the same bill unit. [Figure 12-1](#) illustrates this option.

Figure 12-1 Service-Level Balance Groups with a Shared Bill Unit

- Billing all services separately:** You assign each service a separate billing profile. When you submit the order, BRM tracks each service in a separate balance group and each balance group points to a separate bill unit. [Figure 12-2](#) illustrates this option.

Figure 12-2 Service-Level Balance Groups with Separate Bill Units

- Billing some services together and others separately:** You assign the same billing profile to some services and a separate billing profile to others. When you submit the order, BRM tracks each service in its own balance group. Some balance groups point to the same bill unit and others point to separate bill units. [Figure 12-3](#) illustrates this option.

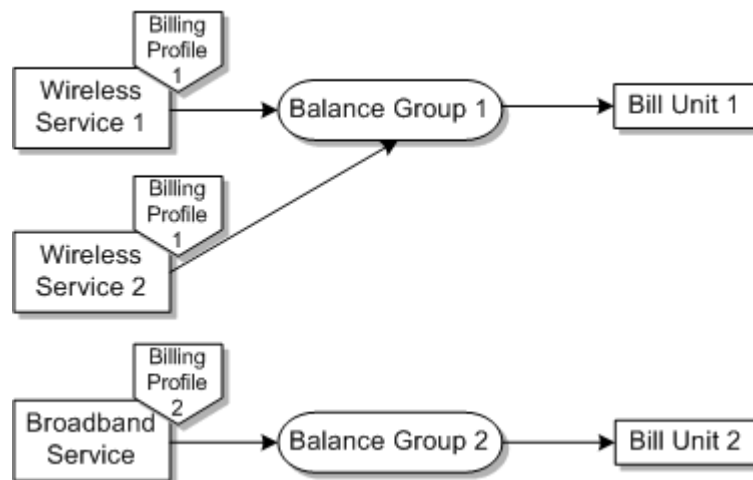
Figure 12-3 Service-Level Balance Groups with Shared and Separate Bill Units

How BRM Tracks Services in Nested Service Bundles in Service-Level Balance Groups

When you purchase service bundles containing nested service bundles (including simple service bundles), you must assign the nested service bundles the same billing profile as their parent service bundle. When you submit the order, BRM tracks the nested service bundles in the same balance group as the parent service bundle.

See ["How BRM Tracks Service Bundles and Products Purchased on Change Orders in Service-Level Balance Groups"](#) for information about how BRM tracks services in balance groups when you use a change order to add new nested services to existing service bundles (Siebel CRM installed assets).

[Figure 12-4](#) illustrates how BRM tracks nested service bundles when service-level balance groups are enabled.

Figure 12-4 Balance Groups for Nested Service Bundles

In [Figure 12-4](#), Wireless Service 2 is a service bundle nested within Wireless Service 1. Wireless Service 1 and Wireless Service 2 represent separate service instances in BRM, but BRM tracks both in the same balance group. You must assign the same billing profile to Wireless Service 1 and 2.

Because nested service bundles are tracked with their parent service bundle, you cannot transfer a nested service bundle by itself. You must transfer the parent service bundle and all of its components together.

How BRM Tracks Service Bundles and Products Purchased on Change Orders in Service-Level Balance Groups

When you use change orders to purchase additional service bundles and products, you can purchase them separately or as components of an existing service bundle.

BRM tracks the new service bundles and products as follows:

- BRM tracks each service bundle purchased separately under its own balance group. You can assign any billing profile to separate service bundles.
- BRM tracks a product purchased separately from any service bundle or nested more than two levels within a service bundle in the account-level balance group. You can assign any billing profile to the new product, but the integration overrides your choice with the primary billing profile on the account.
- BRM tracks a product purchased as an addition to an existing service bundle in the same balance group as the parent service bundle. You must assign the same billing profile as the parent service bundle to the new product.
- BRM tracks service bundles that you purchase as additions to an existing service bundle in the same balance group as the existing service bundle when the existing service bundle was purchased *after* service-level balance groups were enabled. You must assign the same billing profile as the parent service bundle to the new service bundle.

Note: If you submit an Update or MoveAdd change order for a service bundle and add a new nested service bundle on the same order, BRM tracks the new nested service bundle in a separate balance group from the parent service bundle. If you want BRM to track the new service bundle in the same balance group as its parent service bundle, you must submit a separate order to add the new nested service bundle.

- BRM tracks service bundles that you purchase as additions to an existing service bundle in a new service-level balance group when the existing service bundle was purchased *before* service-level balance groups were enabled. BRM continues to track the parent service bundle in the account-level balance group. You can assign any billing profile to the new service bundle.

For more information about service bundles and their components, see "[About Service Bundles](#)".

How BRM Tracks Account-Level Products in the Default Account-Level Balance Group

BRM automatically tracks account-level products in the default account-level balance group created in the Create/Sync Customer Account integration flow. You can assign any billing profile to account-level products, but the integration overrides your choice with the primary billing profile on the account. You cannot transfer account-level products to different accounts or different billing profiles.

About Transferring Services

The integration supports transferring services tracked in service-level balance groups. You can transfer services to a different billing profile on the same account or to a different service account.

To transfer services, submit a change order to change the service account and billing profile information on the service that you want to transfer. See the discussion of using asset-based ordering in *Siebel Order Management Guide* for information about submitting change orders.

The following restrictions apply to service transfers:

- You can only transfer services tracked in service-level balance groups. You cannot transfer any services purchased when service-level balance groups were disabled or any services purchased at the account level.
- You must transfer all services tracked in a single balance group at the same time. This restriction means that you must transfer nested service bundles along with their parent service bundle by changing the service account on both.
- If more than one BRM instance is connected to a single Siebel CRM instance, the source and target accounts for a service being transferred must be in the same BRM instance.
- You cannot add or remove nested billing products in the same order as a service transfer. You must submit one order to transfer the services and a separate order to add or remove nested billing products.

Note: If you submit an Update or MoveAdd change order to transfer a service bundle and add a new nested service bundle on the same order, BRM tracks the new nested service bundle in a separate balance group from the parent service bundle. If you want BRM to track the new service bundle in the same balance group as its parent service bundle, you must submit a separate order to add the new nested service bundle.

About Transferring Services to a Different Account

To transfer a service to a different account, submit a change order that lists the target service account for the service that you want to transfer. You can also transfer the service to a different billing profile on the same change order.

When you submit the change order, the integration transfers the service to the target account and creates a new balance group and bill unit in BRM assigned to the target account's billing profile.

About Transferring Services to a Different Billing Profile

To transfer a service to a different billing profile on the same account, submit a change order that lists the target billing profile on the service bundle order line that you want to transfer. You must also change the billing profile for any services nested within the service being transferred.

When you submit the change order, the integration creates a new balance group and bill unit in BRM assigned to the target billing profile. Because of the automatic naming conventions for balance groups, the new balance group will have the same name as the old balance group.

Working with Service-Level Balance Groups Disabled

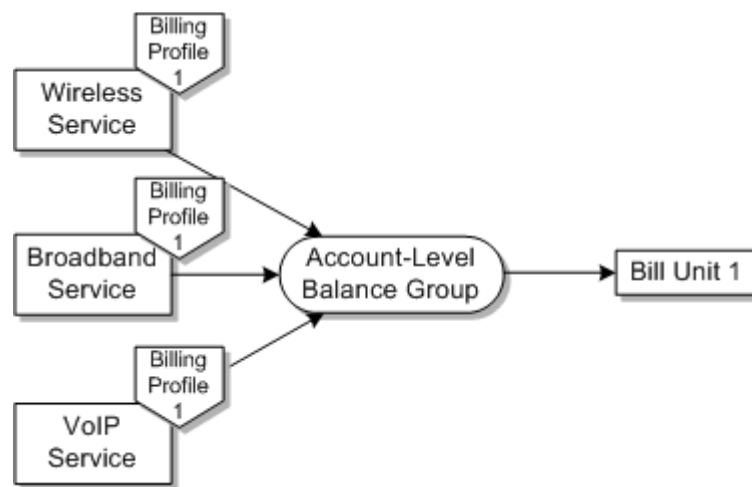
When you work with service-level balance groups disabled, BRM uses the default account-level balance group to track and pay for all of their services together.

The default account-level balance group is created at the same time as the customer account in the Create/Sync Account integration flow. When service-level balance groups are disabled, BRM tracks all services and products for an account under this default account-level balance group.

When you create subsequent orders for services (including nested service bundles and additional services purchased on change orders), you must use the same billing profile as the one selected on the first order.

Figure 12-5 illustrates how services are tracked under the account-level balance group.

Figure 12-5 Tracking Services in the Default Account-Level Balance Group



When you submit a single order for multiple products, the integration uses the billing profile of the first service on the order for all subsequent services on the same order. If an order for the services in Figure 12-5 assigned separate billing profiles to Wireless and Broadband, the result would remain the same because the billing profile for Wireless (the first service on the order) would be used for both services.

Supporting Product Bundling

When you submit an order in Siebel CRM containing bundled products, the integration synchronizes the service bundles to service instances and the component products and discounts to purchased product and discount instances in BRM.

The integration synchronizes account-level products, account-level discounts, and any product or discount nested more than two levels below a service bundle to account-level purchased product and discount instances in BRM.

Note: Because dynamic and relationship classes are not sent to BRM with the Siebel CRM order, they do not help determine a nested service bundle or nested product's parent.

See "[Understanding Product Bundling](#)" for more information about product bundling in Siebel CRM.

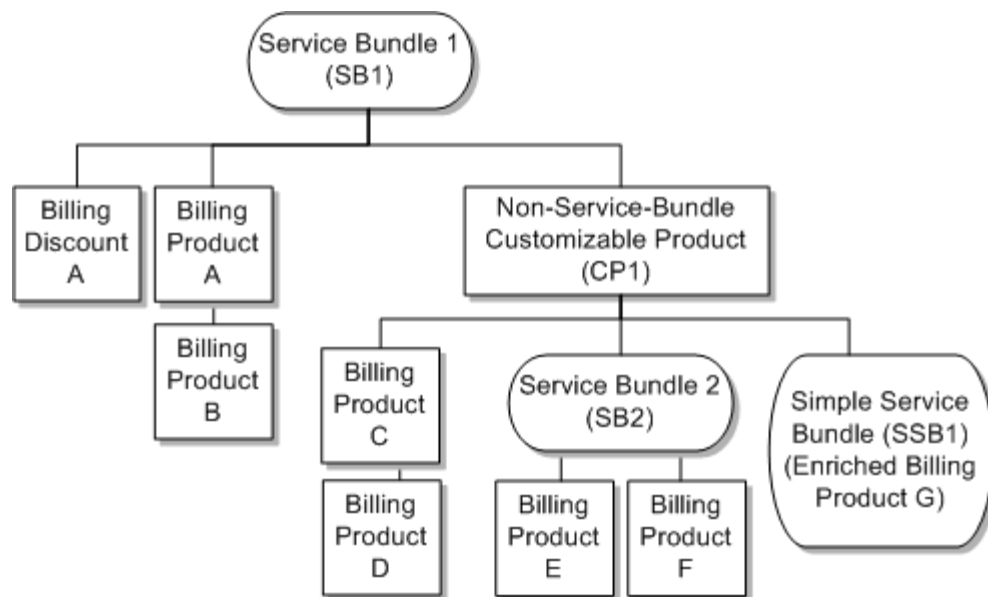
Example of Mapping for Bundled Products

This example shows how the integration maps a service bundle containing products, discounts, non-service-bundle customizable products, and nested service bundles from Siebel CRM to BRM.

Billing Products A through G and Billing Discount A are all synchronized from BRM to Siebel CRM. The service bundle hierarchy in Siebel CRM, illustrated in [Figure 12-6](#), is as follows:

- Service Bundle 1 (SB1) contains Discount A, Billing Product A, and a non-service-bundle customizable product (CP1).
- Billing Product A is modeled as the parent of Billing Product B.
- CP1 contains Billing Product C, a second service bundle (SB2), and a simple service bundle (SSB1).
- Billing Product C is modeled as the parent of Billing Product D.
- SB2 contains Billing Products E and F.
- SSB1 is the enriched form of Billing Product G, a BRM subscription product.

Figure 12-6 Example of Nested Service Bundles



When you submit an order for SB1, the integration creates the following elements in BRM:

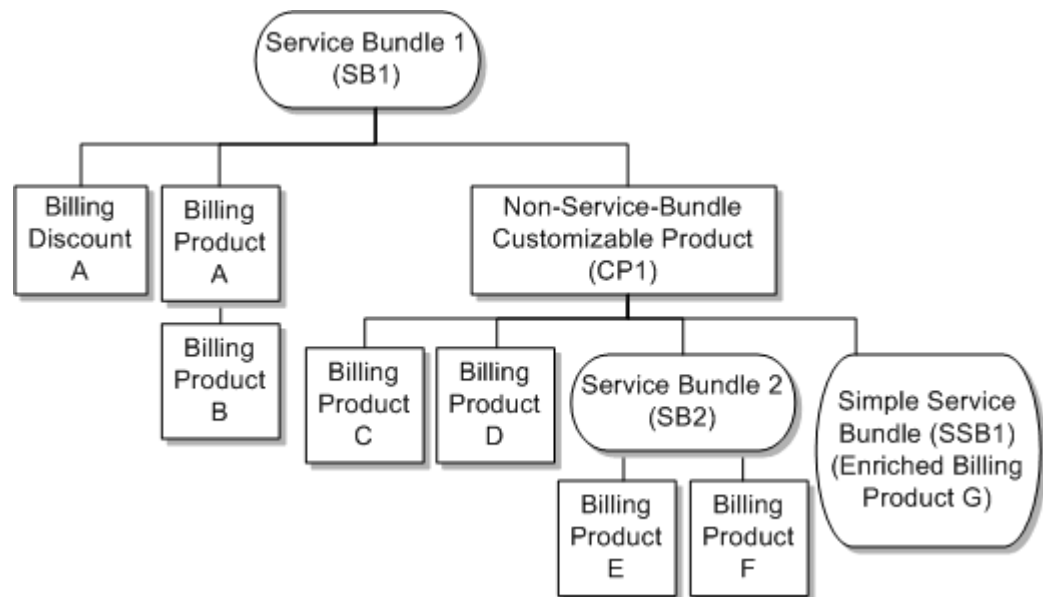
- A service instance for SB1 with purchased product instances for Billing Products A through C and a purchased discount instance for Billing Discount A
- A service instance for SB2 with purchased product instances for Billing Products E and F
- A service instance for SSB1 with purchased product instance for Billing Product G

The integration synchronizes Billing Product D to an account-level purchased product instance in BRM because it is nested more than two levels below a service bundle.

For the integration to synchronize Billing Product D to a purchased product instance under the service instance for SB1, you should model it in Siebel CRM as a sibling of

Billing Product C, as in [Figure 12-7](#).

Figure 12-7 Example of Remodeled Billing Products



You can submit orders for non-service-bundle customizable products that are not included in service bundles. However, because the integration does not create service instances in BRM for non-service-bundle customizable products, it maps billing products or discounts that are in a non-service-bundle customizable product but not included in a service bundle to account-level purchased product or discount instances in BRM. For example, on an order for CP1 alone, the integration maps Billing Products C and D to account-level purchased product instances.

Synchronizing Simple Service Bundles

When you submit an order for a simple service bundle, the integration synchronizes it as a service bundle, creating both a service instance and a purchased product instance in BRM. If the quantity on a simple service bundle line is greater than one, the quantity applies to the product instance alone.

Both single-phase billing and two-phase billing are supported for the simple service bundles.

Changing Purchased Simple Service Bundles

Suspending, resuming, or disconnecting the asset that represents a simple service bundle in Siebel CRM suspends, resumes, or cancels the service and purchased product instance in BRM. You cannot suspend, resume, or cancel the product without suspending, resuming, or canceling the service.

Transferring the asset that represents a simple service bundle in Siebel CRM using an order with MoveAdd or MoveDelete line actions adjusts the cross-references of the service and purchased product instances in BRM.

Updating the service instance attributes (for example, Service ID, billing account, billing profile) on the asset that represents a simple service bundle in Siebel CRM updates the service instance in BRM.

Updates to product attributes other than quantity changes (for example, pricing changes, promotion reference) on the asset that represents a simple service bundle in Siebel CRM updates the purchased product instance in BRM. You can also make changes to billing dates as part of two-phase billing.

If you applied a onetime charge for a line action in Siebel, the integration applies the charge to the balance group for the simple service bundle's service instance in BRM.

Simple Service Bundle Cross-References Entries

To support simple service bundles by mapping a single Siebel CRM asset to both a service instance and a purchased product instance in BRM, the integration creates a cross-reference entry in the InstalledProduct cross-reference table, as shown in [Table 12-1](#).

Table 12-1 Simple Service Bundle 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 BRM portal object ID (POID) for the service instance and BRM-B01 is the 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.

Supporting Single-Phase and Two-Phase Billing

The integration 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.

Choosing Between Single-Phase and Two-Phase Billing

Billing fulfillment scenarios lead to one of two fulfillment patterns, each of which must be supported by the order management implementation.

Single-Phase Billing

With single-phase billing, a service is interfaced to billing through billing fulfillment toward the end of the fulfillment flow, after the order is delivered and the actual delivery date is known.

You use single-phase billing in the following situation:

- When you do not have time lag or validation concerns. In this situation, interfacing to billing takes place after the service or product is made available to the customer.

The date that a product is made available can vary based on jurisdiction and whether the product 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

With two-phase billing, the integration interfaces a service to billing twice:

- **Billing initiation:** The service and purchased products are interfaced early in the fulfillment flow and before actual delivery dates are known.
- **Billing fulfillment:** Accurate billing dates are updated in billing after the order is delivered and the actual delivery date is known.

You use two-phase billing in the following situations:

- **Fulfillment latency:** when operational or deployment conditions produce a time between the time a service is made available for customer use and the time the service is interfaced into billing.

The time lag can cause errors in the usage records resulting in lost revenue. Rather than attempting to plan fulfillment of future-dated orders to meet the requested delivery date, build the fulfillment flow so that the Usage Start Date is set to the current date during billing initiation, and the Cycle Start Date is set to a distant future date. At billing fulfillment, the Cycle Start Date is then reset to match the Actual Delivery Date or Requested Delivery Date, depending on business practices and legal requirements.

- **Validation latency:** When you have inadequate controls to guarantee that orders are valid, resulting in a high rate of invalid orders, and the cost of delaying order line validation for interfacing to billing is high.

In this situation, orders must be interfaced to billing early in the fulfillment flow to ensure that the order can be interfaced successfully later. Build the fulfillment flow so 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.

Using Single-Phase Billing or Two-Phase Billing

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.

INITIATE BILLING Mode

You can design an order orchestration flow to interface the order to billing before the order is sent to provisioning. Calling the interface in INITIATE BILLING mode is optional. The billing interface is called with either of the following:

- **The whole order:** all of the lines on the order that are intended for a certain target billing system and related lines such as promotion lines.

- Order components: promotion lines, service bundle lines and all service bundle component lines, and account-level products. All component lines for a single service bundle must be sent for billing initiation and fulfillment together. Any service bundle component lines sent only for billing fulfillment are not processed.

Depending on the requirements, you can set some or all of the following dates on new purchases of products or discounts to the future (in essence they are treated as inactive when interfaced to billing):

- **Purchase Date** (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderSchedule/PurchaseDate)
- **Cycle Start Date** (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderSchedule/CycleStartDate)
- **Usage Start Date** (ProcessFulfillmentOrderBillingEBM /DataArea/ProcessFulfillmentOrderBilling/FulfillmentOrderLine/FulfillmentOrderSchedule/ServiceUsageStartDate)

For promotion lines, only the purchase date is relevant.

To rate usage as soon as the service is activated but start the cycle fees at the date that the customer requested the service when there is a fulfillment latency between service activation and billing, have your order management system set the purchase and usage start dates to current and the cycle start date to the future when calling this service. See "[General Modeling and Implementation Recommendations](#)" for more information.

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. One-time charges for actions such as Suspend, Resume, Move, and Disconnect and promotion penalties are not processed in this mode.

See "[Mapping Billing Dates](#)" for more information about how dates are set in BRM.

Handling of Revision Orders

BRM 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, but resets the cycle start and usage start date if asked by the caller.

However, when billing initiation is called to process a revision on order lines that are billing initiated, and the call resets the cycle start date when the previously set date is current, then billing initiation fails with a BRM validation error.

General Modeling and Implementation Recommendations

The interface validates that the cycle date is set to the future for products of type **Subscription** or **Discount**. For products of type **Item**, the interface validates that the purchase date is set to the future. Oracle recommends that you set the future billing date to a year ahead of the due date when calling billing initiation.

The purchase, cycle start, or usage start dates is in the future if the following is true about the billing date:

```
billing date > (Fusion Middleware current time converted to UTC + (25 or
FutureTimeThreshold hours, whichever is greater)).
```

where *FutureTimeThreshold* is the value of the *FutureTimeThresholdForBillingDates* Oracle AIA configuration property. This property has a default value of 8640 hours (360 days in hours).

If you are highly confident of the lead time required to activate the service, then you 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. You can set this property for each BRM instance.

If the *FutureTimeThresholdForBillingDates* property 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.

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 (BRM ABCS) errors.

Recommendations for Purchase Fees or Activation Charges

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 situation, 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 situation, 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.

Recommendations for 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.

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 integration supports and is required to interface an order to billing.

In this mode, the integration processes all order lines that are sent on new orders or change orders. One-time charges for actions such as Suspend, Resume, Move, and Disconnect and promotion penalties are processed in this mode.

For 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. 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.

The integration determines that an attribute has changed if prior value fields are populated. Your order management system must set the prior value fields for the following billing dates:

- **PurchaseDate:**
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
PurchaseDate
- **CycleStartDate:**
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
CycleStartDate
- **ServiceUsageStartDate:**
ProcessFulfillmentOrderBillingEBM/DataArea/ProcessFulfillmentOrderBilling/
PriorFulfillmentOrder/FulfillmentOrderLine/FulfillmentOrderSchedule/
ServiceUsageStartDate

Caution: If billing dates were set to current in billing initiation, resetting them in billing fulfillment causes a BRM error.

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 CRM) 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.
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 one-time charge associated with a MACD action (Suspend, Resume, Move, Disconnect) with the service bundle on which the action is being performed.
5. Every MoveAdd line on a Siebel CRM order has a matching MoveDelete (and vice versa). The order management system sends MoveAdd lines along with the MoveDelete lines to billing.
6. After order lines are submitted for billing fulfillment, they are assumed to have hit a hard point of no return and cannot be revised in Siebel CRM.
7. Service ID is always sent as input to the billing interface (Initiation or Fulfillment).

See "[Mapping Billing Dates](#)" for more information about how dates are set in BRM.

Supporting Revisions

To provide support for revisions after order lines are billing-initiated but not yet billing-fulfilled, the order interface to 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. Old attribute values are supplied only for delta changes.

Changes to certain attributes on revised lines result in updates to billing. These attributes are:

- On a revised promotion line: Billing Account, Purchase Date
- On a revised account-level product line: Billing Account, Bill Profile, Promotion reference, Pricing Information, Billing Dates
- On a revised service bundle line: Billing Account, Bill Profile, Promotion reference, Service ID
- On a revised service bundle component line: Pricing Information (price list must be revised at service bundle level), Billing Dates

The Pricing Information attribute includes list price, selling (or net) price, pricing commit type, dynamic discount method, discount amount, and discount percent.

For the Billing Dates attribute, 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.

See "[Supporting MACD Actions and Attribute Changes](#)" for more information about the order attributes.

Caution: Revisions to order lines for products of type *Item* can be interfaced to BRM if the billing date is not current. When it is current, the call to update BRM fails.

If an order line is successfully billing-initiated and subsequently canceled in Siebel CRM (dropped from the Siebel CRM modify order) 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, a 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 BRM), then the order management system calls billing initiation with a fulfillment mode of **NOOP**.

The Oracle AIA service that interfaces orders to BRM processes all of the lines or none of the lines. It does not do partial processing. When an order is successfully billing-initiated, when any subsequent revisions for lines on the base order are processed, the order management system must trigger compensation as described previously (using the **REDO**, **UNDO**, or **NOOP** fulfillment modes). If the order fails billing initiation (and triggers Order Fallout), a subsequent revision should be sent as is for billing initiation (DO mode).

Caution: 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	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.

Assumptions and Constraints for Revisions

1. Order lines are assumed to hit the point of no return after they have been interfaced to BRM in the Fulfill Billing mode. Revisions are only supported when 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. Because only new purchases (lines with action ADD) are processed by billing initiation, revisions are only processed 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.

Supporting Time-Based Offerings on Orders

Time-based offerings let you use a Siebel CRM product class to set validity periods for products and discounts synchronized from BRM. You purchase time-based offerings on orders in the same way as other products and discounts and the integration calculates the validity periods as described in ["Supporting Time-Based Offerings on New Orders"](#) and ["Supporting Time-Based Offerings on Change Orders"](#).

For information about creating time-based offerings and managing expired time-based offerings, see ["About Time-Based Offerings"](#).

Note: If you are using an order management system other than OSM, Oracle recommends that you configure your system not to set end dates during billing initiation. End dates are not required for billing initiation, and setting them during billing initiation avoids the requirement to manage them as part of revisions.

OSM AIA cartridges do not set end dates during billing initiation.

Supporting Time-Based Offerings on New Orders

The integration processes new orders for time-based offerings as follows:

1. When you submit the order, Siebel CRM calculates the end date based on the start date (defaulted from the due date) and the Duration, DurationUnitOfMeasure, and DurationValidityStart transaction attribute values and sends the order through the integration to OSM for fulfillment.
2. When fulfilling the order, the OSM AIA cartridges set the purchase, cycle start, and usage start dates based on service actual delivery date and recalculates the end date.
3. When the order is billing fulfilled, the integration communicates the end date for the purchased product or discount to BRM.
4. OSM sends the actual start and end dates through the integration to Siebel CRM as part of the order update message.

Supporting Time-Based Offerings on Change Orders

The integration processes orders that change the duration validity of previously-purchased time-based offerings as follows:

1. Siebel CRM recalculates the end date based on the Duration, DurationUnitOfMeasure, and DurationValidityStart transaction attribute values and sends the order through the integration to OSM for fulfillment.
2. When fulfilling the order, if the values for the validity attributes on the order are different from the prior values, the OSM AIA cartridges recalculate the end date based on the actual delivery date. The cartridges use the value for DurationValidityStart to calculate the new end date as follows:
 - **Original End:** the new value for Service End Date is the prior value for Service End Date plus the value for Duration
 - **Now:** the new value for Service End Date is the value of Actual Delivery Date Time plus the value for Duration
 - **Original Start:** the new value for Service End Date is the value of Service Start Date plus the value of Duration
3. When the order is billing fulfilled, the integration communicates the new end date for the purchased product or discount to BRM.
4. OSM sends the changed end dates through the integration to Siebel CRM as part of the order update message.

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.

See "[Supporting Friends and Family](#)" for more information about how special rating products are supported and the methodology.

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.

See "Profiles in Siebel Communications" in *Siebel Communications Guide* for more information.

When the order is interfaced to 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 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 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 BRM instance.

See ["Supporting Friends and Family"](#) and ["Configuring Multiple BRM Instances for Communications Integrations"](#) for more information.

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 BRM. This synchronization is enabled by the following integration services:

`ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer`

`ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl`

`ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl`

- 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 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 BRM.

Modifying Friends and Family List

After a service that supports special rating has been purchased and the order fulfilled and asseted, you can use the Siebel Special Rating Profile UI to make changes to the list, and then update and synchronize the list to BRM.

The flow uses the `ProcessInstalledProductSpecialRatingSetList` operation on the `ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl` service for

this purpose. The specification group on the installed product EBM is used for communicating the list entries.

See "[Implementing the Synchronize Customer Special Rating Profile Business Flow](#)" for more information.

Assumptions and Constraints for the Bill Fulfillment Order Business Flow

The assumptions and constraints for the Bill Fulfillment Order business flow are as follows:

- The integration only supports defining a single brand within a single instance of BRM.
- 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 CRM state model. The order can be revised and resubmitted for processing if it has not reached a point of no return. The solution assumes that the order line reaches the point of no return after the line has been sent for billing fulfillment.
- The integration does not support copied orders in Siebel CRM because Siebel CRM does not regenerate the asset integration ID that uniquely identifies purchases on the copied order. Instead of copying orders, Oracle recommends that you use the Siebel CRM Favorites feature.
- 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 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 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 BRM.
- No special handling exists for shippable goods. No support is available for returns or credit orders.
- If you are also using the Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care pre-built integration, order lines that must be sent to different billing systems must have different billing profiles.
- 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.

- The service account, billing account, and billing profiles are the same on all order lines for components in a service bundle.
 - When service-level balance groups are enabled, you must ensure that these fields are the same for service bundles and their components
 - When service-level balance groups are disabled, any integration logic that works on these fields looks only at the service bundle line. This constraint also applies to one-time 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 lines and applies the charge to the default account-level balance group.
- The integration does not support changing from nonpaying subordinate to self-paying account or changing from self-paying account to nonpaying subordinate. Changing accounts in this way does not produce an error but results in data that breaks the billing management integration flows.
- A subordinate account cannot have multiple paying parents. This is not supported in BRM.

Any order changing the paying parent for a subordinate account using a new purchase or a change order for an existing service 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 to the new parent so that it can successfully interface customer data to BRM.

Transactions that do not obey this assumption fail with a BRM error when an order is interfacing customer data to BRM.

- All lines within a service bundle reference products from the same billing system.

A single Siebel CRM asset can be mapped to a service instance or a purchased product or discount instance in only one billing system.
- The integration assumes that the service bundle and its component products reference the same billing service type. This assumption applies only to component products that represent BRM products of type Subscription or BRM discounts. Violation of this assumption results in a BRM error. Nested service bundles do not have to have the same service type as the root parent service bundle. See "[About Billing Service Types for Service Bundles](#)" for more information.

Implementing the Bill Fulfillment Order Business Flow

This chapter describes the Bill Fulfillment Order business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using BRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Bill Fulfillment Order Business Flow Overview

The following pre-built integration option enables the Bill Fulfillment Order business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

The Bill Fulfillment Order business flow supports the following integration flow:

- Interfacing Orders to Create Transaction Data in BRM

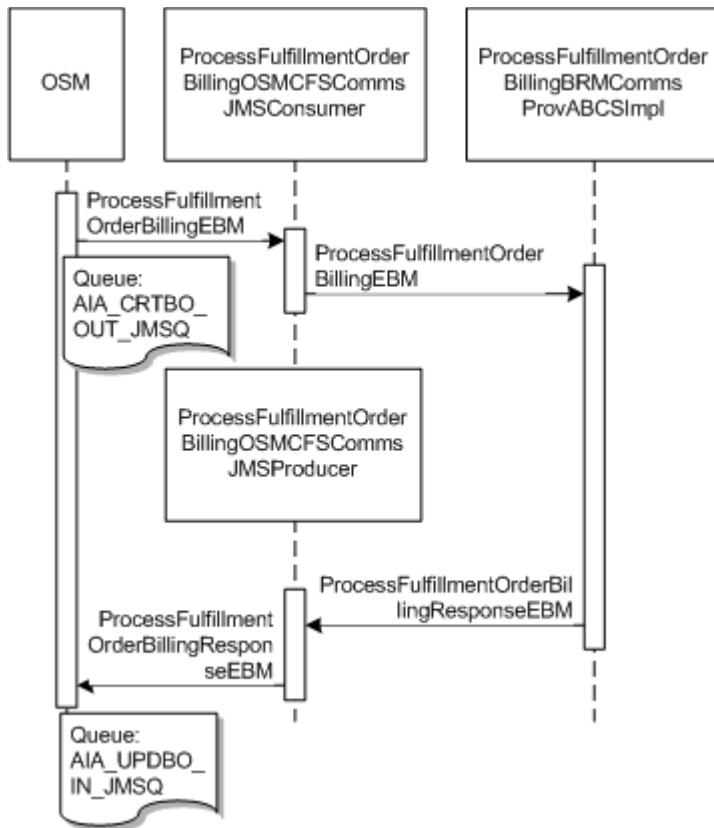
Interfacing Orders to Create Transaction Data in BRM Integration Flow

The Interfacing Orders to Create Transaction Data in BRM integration flow uses the following services:

- `ProcessFulfillmentOrderBillingOSMCFSCommsJMConsumer`
- `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl`
- `ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer`

[Figure 13-1](#) illustrates the integration components used by OSM to interface orders to create transaction data in BRM.

Figure 13–1 Bill Fulfillment Order Sequence Diagram



When this flow is initiated, the following events occur:

1. OSM converts the ProcessSalesOrderFulfillmentEBM message received in the Process Sales Order Fulfillment business flow into a ProcessFulfillmentOrderBillingEBM message and places it in a JMS queue. The store-and-forward mechanism pushes the message to the AIA_CRTBO_OUT_JMSQ messaging queue.
2. The ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer service picks up the ProcessFulfillmentOrderBillingEBM message and routes it to the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service.
3. The ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service uses the ProcessFulfillmentOrderBillingEBM message to create billing artifacts, service instances with balance group information, purchased products, purchased discounts, and so on in BRM.
4. The ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service constructs the ProcessFulfillmentOrderBillingResponseEBM message and sends it back to the ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer service.
5. The ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer service forwards the ProcessFulfillmentOrderBillingResponseEBM message in the AIA_UPDBO_IN_JMSQ queue using the store-and-forward mechanism to an OSM messaging queue on the OSM WebLogic server.

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.

See ["Using Error Type to Control Response to Order Fallout"](#) for more information about system errors and business errors.

The following services are involved:

- ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCImpl
- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer

Table 13–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
The ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer service passes the message to the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service, which creates billing artifacts and call one or more subprocesses. The response message is then routed to the ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer service.	AIA cross-references created. BRM data created. Message goes to the AIA_UPDBO_IN_JMSQ queue.	Rollback AIA cross-references. Rollback data created in BRM. Message goes back to the originating queue (AIA_CRTBO_OUT_JMSQ_ErrorQ).	Resubmit the order from the AIA_CRTBO_OUT_JMSQ_ErrorQ queue.

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

BRM Interfaces

The Bill Fulfillment Order business 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_SUBSCRIPTION_PURCHASE_DEAL
- PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT
- PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT

- PCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTION
- PCM_OP_SUBSCRIPTION_SERVICE_BALGRP_TRANSFER
- 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

See the BRM documentation for more information.

Industry Oracle AIA Components

The Bill Fulfillment Order business flow uses these industry components:

- FulfillmentOrderEBO
- ProcessFulfillmentOrderBillingEBM
- ProcessFulfillmentOrderBillingResponseEBM

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

The industry enterprise business service (EBS) WSDL files are located in: **\$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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Bill Fulfillment Order business flow:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCImpl
 - ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingOSMCFSCCommsJMSProducer

ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer

The ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer service is a Mediator process that has a JMS Adapter Service, which continuously polls the AIA_CRTBO_OUT_JMSQ queue. The ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer service dequeues the ProcessFulfillmentOrderBillingEBM message and routes it to the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service.

This service has one operation: Consume_Message.

ProcessFulfillmentOrderBillingBRMCommsProvABCImpl

The ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service 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 BRM using the custom Java EE Connector Architecture (JCA) adapter provided by 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 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 message and is responsible for transforming to the relevant BRM ABM and invoking the corresponding opcode.

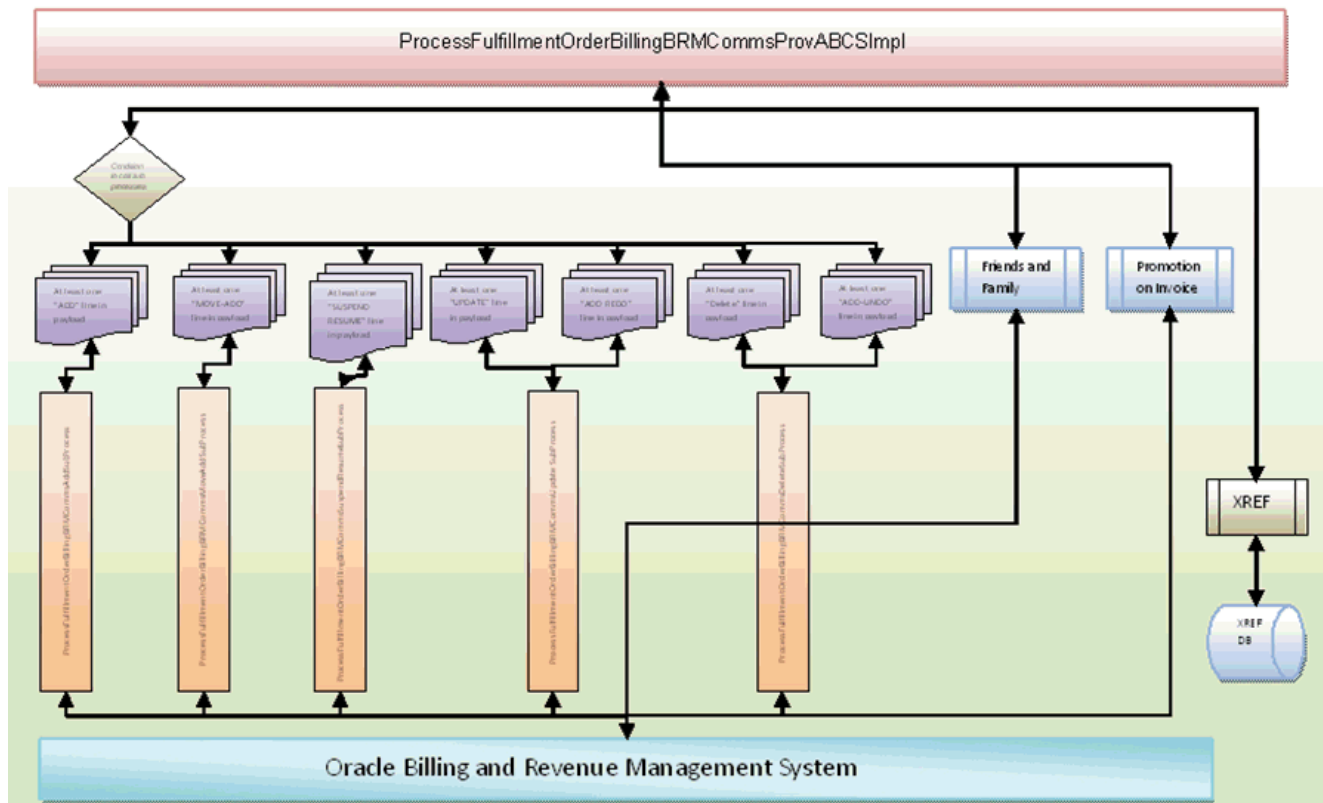
The ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service mainly performs the following activities:

- 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 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 ProcessFulfillmentOrderBillingBRMCommsProvABCImpl Process calls a 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 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 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, ProcessFulfillmentOrderBillingBRMCommsProvABCImpl 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, 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.

ProcessFulfillmentOrderBillingBRMCommsAddSubProcess

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl. 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 one-time 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 order Line, the following operations are performed in the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess:

1. The incoming payload is tunneled through two transforms. The first transform groups all the service bundles per service account and the second transform groups all the account-level purchases.
 - When the product type is service bundle, the BPEL process accumulates all of the children inside the 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 per billing profile are passed in one single PCM_OP_CUST_MODIFY_CUSTOMER opcode call.
 - 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 a BRM-specific message.
2. After the BRM opcode calls are successfully carried out, this BPEL process captures the POID (ObjectIdentifier) returned by BRM and populates the XREFPopulateData. See [Table 25–2, "Order Lifecycle Management Cross-References"](#) for more details about the cross-references populated.
3. For ITEM, the POID (ObjectIdentifier) is returned by BRM only during INITIATE BILLING mode.

This service communicates with BRM using the JCA adapter provided by BRM. The service uses the default capability of the JCA adapter to define unit transactions for every order. (Do all or none.)

This service supports two modes of billing:

- Initiate billing
- Fulfill billing

ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess

The ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl. 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 MoveAdd and the product type is a service bundle, a subscription product, an account-level product, or an account-level discount, the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess is called.

This process supports the following situations:

- Transferring a service from one billing profile or service account to another when service-level balance groups are enabled. This includes changing the paying parent for services on a subordinate account.

The `ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess` service calls the `ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess` service. See "[ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess](#)" for information about how the Update subprocess implements the service transfer.

- Changing the paying parent for services on a subordinate account when service-level balance groups are disabled. Both billing profile and billing account must be changed.

The `ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess` service calls the `ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess` service. See "[ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess](#)" for information about how the Update subprocess implements the changes to billing profile and billing account.

- Changes to ServiceID, Price Override, and Discount Override made as part of a move-add command.

To process the changes, after `ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess` is invoked, `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` invokes `ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess`.

- Simple Move-Add of the service bundles from one location to another.

There is no BRM interaction for this operation. Only the entries in the cross-reference tables are repointed.

- A Move-Add order accompanied by a one-time penalty charge.

When a one-time penalty charge is associated, the `ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess` converts the `ProcessFulfillmentOrderBillingEBM` into a BRM-specific message and calls the `PCM_OP_SUBSCRIPTION_PURCHASE_DEAL` BRM opcode.

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 `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl`. 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 a BRM-specific message and calls the PCM_OP_CUST_SET_STATUS 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 a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS 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 a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS 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 one-time penalty charge may or may not be associated.

- When a one-time penalty charge is associated with the service bundle, then depending on the action code, the one-time charge gets added in the following manner:

When the action code is Suspend, the one-time charge gets added first.

ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL BRM opcode.

After the one-time charge is added, then Operation 1 is run to suspend the service bundle.

When the action code is Resume, the one-time charge gets added after the service bundle is resumed.

Operation 1 is run to resume the service bundle.

Afterwards, the one-time charge gets added:

ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL BRM opcode.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

The ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess service is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service. This call depends on the action code present on the order line and 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, subscription product, or account-level discount, the ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess is called.

This process supports the following update situations:

- Updating billing profiles, billing accounts, or service accounts for service bundles.
 - When service-level balance groups are enabled, you can transfer a service from one billing profile, billing account, or service account to another. This includes changing the paying parent for the services on a subordinate account.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message including new billing profile, service account, and balance group information. The process then calls the PCM_OP_CUST_MODIFY_CUSTOMER opcode to update the billing profile or billing account, and the PCM_OP_SUBSCRIPTION_SERVICE_BAL_GRP_TRANSFER opcode to transfer the services to the new service account and update the balance group POIDs in the cross-reference table.

- When service-level balance groups are disabled, you can change the paying parent for the services on a subordinate account by updating both the billing profile and the billing account.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message including new billing profile and billing account. The process then calls the PCM_OP_CUST_MODIFY_CUSTOMER opcode to update the billing profile and billing account in BRM.

- Updating the service ID for a particular service bundle.

During this scenario, you 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 a BRM-specific message and calls the PCM_OP_CUST_UPDATE_SERVICES BRM opcode.

- Price override

During this scenario, you can change the PriceOverride on a product line.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO BRM opcode.

- Discount override

During this scenario, you can change the DiscountOverride on a product line.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO BRM opcode.

- TBO End Date

During this scenario, you can change the EffectiveEndDate on a product line.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_SET_PRODINFO BRM opcode in case of SUSCRPTION products and calls PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO in case of discount products.

ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

The ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is a synchronous BPEL process that is called by ProcessFulfillmentOrderBillingBRMCommsProvABCImpl. 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 a BRM-specific message and calls the PCM_OP_CUST_SET_STATUS 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 a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT BRM opcode.
- When the action code is Delete and the product type is Account-Level Subscription, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT BRM opcode.
- During these operations, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess also checks for the existence of any one-time penalty charge. If present, then the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message and calls the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL BRM opcode.

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 BRM.

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 Oracle Communications Order and Service Management in the central order management role (OSM COM).

The Provision Order and Update Fulfillment Order business flows are enabled using the following Pre-Built Integration options of the Oracle Communications Order to Cash Integration Pack for Siebel customer relationships management (Siebel CRM), OSM, and Oracle Communications Billing and Revenue Management (BRM) (the integration):

- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

Overview of Order Provisioning

Siebel CRM sends customer order fulfillment requests of both Qualify and Deliver types to OSM COM, which decomposes them into suborders called order components. OSM uses the integration to send order components that are targeted for provisioning to either OSM in the service order management (OSM SOM) role or other order management systems.

See "[Order Capture Overview](#)" for more information about customer order fulfillment request types.

When OSM SOM manages the order lifecycle management (OLM) events of the service order. For Cancel and Revision requests, OSM generates and executes compensation plans to match the change. OSM also manages order data and status updates, and fallout incidents. Throughout the fulfillment process, OSM SOM sends status and data updates to OSM COM.

About Creating Provisioning Orders

For interacting with OSM SOM, OSM COM pushes the ProcessProvisioningOrderEBM message, which includes most of the SalesOrderEBO attributes, into the AIA_CRTFO_OUT_JMSQ store-and-forward queue. Error responses come 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.

See "[Implementing the Provision Order and Update Fulfillment Order Business Flows](#)" for more information about this sequence of events.

See "[Understanding the Process Integration for Order Fallout Management](#)" for more information about order fallout.

About Updating Fulfillment Orders

This feature provides the ability to update OSM COM with OSM SOM milestones, status, and data.

Order Status Management is an integral capability of OSM COM. OSM COM provides for a configurable order status management across different fulfillment systems, including OSM SOM.

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.

Assumptions and Constraints for the Provision Order and Update Fulfillment Order Business Flows

One or more OSM Provisioning Cartridges must be deployed. They preserve the Oracle AIA interfaces.

See the OSM documentation for more information about product-specific assumptions and constraints.

Implementing the Provision Order and Update Fulfillment Order Business Flows

This chapter describes the Provision Order and Update Fulfillment Order business flows and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components and integration services.

Provision Order and Update Fulfillment Order Business Flows Overview

The following Pre-Built Integration options enable the Provision Order and Update Fulfillment Order business flows:

- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

The Provision Order and Update Fulfillment Business Order business flows support the following integration flow:

- OSM Fulfillment to OSM Provisioning integration flow: passes provision order requests to OSM in the service order management role (OSM SOM) and updates OSM in the central order management role (OSM COM) with OSM SOM milestones, status, and data.

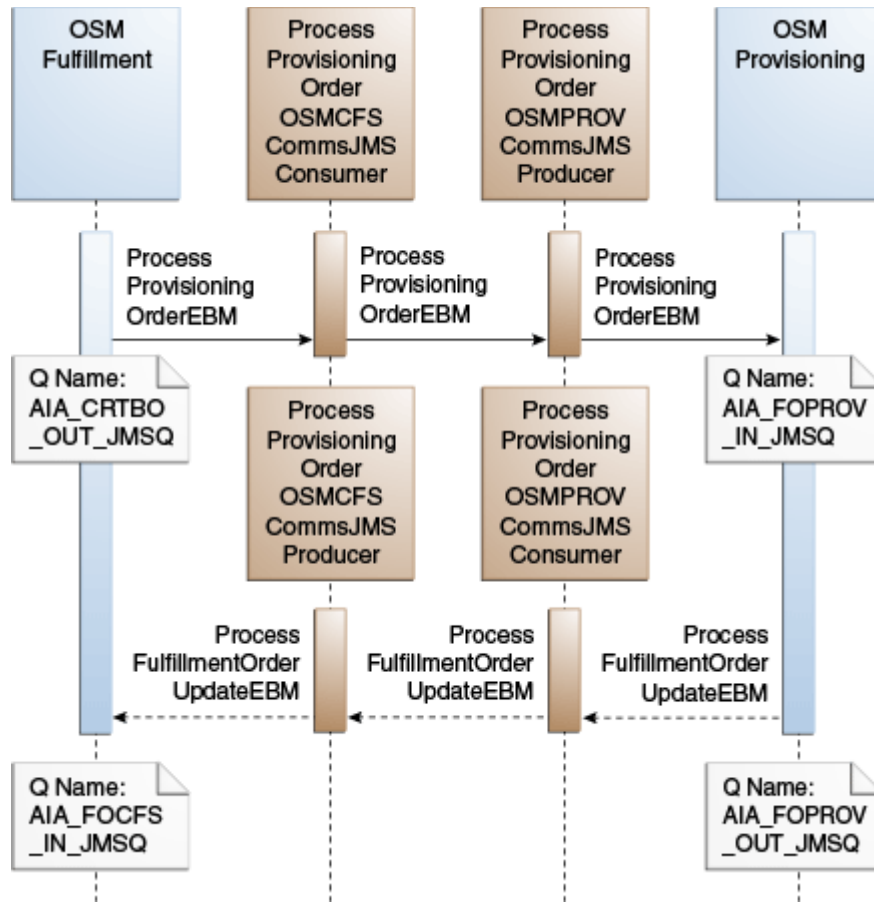
OSM Fulfillment to OSM Provisioning Integration Flow

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 OSM Provisioning and receiving updates back:

Figure 15–1 OSM Fulfillment to 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 OSM COM. The message is enqueued in the AIA_CRTFO_OUT_JMSQ queue using the store-and-forward mechanism.
2. The ProcessProvisioningOrderOSMCFSCommsJMSConsumer service 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 service.
3. The routing mentioned in the previous step produces the message into the AIA_FOPROV_IN_JMSQ queue. The store-and-forward mechanism puts the message into OSM. OSM SOM then picks up the message from the queue and processes it accordingly.
4. During provisioning, one or more update messages are enqueued by OSM SOM into OSM WebLogic and eventually moves to the AIA_FOPROV_OUT_JMSQ queue using store-and-forward. The ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer service dequeues the ProcessFulfillmentOrderUpdateEBM message and routes it to the ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer service.
5. The ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer service enqueues the ProcessFulfillmentOrderUpdateEBM message in the AIA_FOCFS_IN_JMSQ queue using store-and-forward. OSM picks up this message to update the status of the order.

See ["Understanding the Process Integration for Order Lifecycle Management"](#) for more information about the events that occur when this process initiates.

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.

See ["Using Error Type to Control Response to Order Fallout"](#) for more information about system and business errors.

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 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 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.

See ["Configuring Oracle AIA Processes for Error Handling and Trace Logging"](#), [Configuring Fault Policies to Not Issue Rollback Messages in Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack](#) for more information about rollback procedures.

Industry Oracle AIA Components

The Provision Order and Update Fulfillment Order business flows use the following communications industry-specific Oracle AIA 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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Provision Order and Update Fulfillment Order business flows:

- ProcessProvisioningOrderOSMCFSCommsJMConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMConsumer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

ProcessProvisioningOrderOSMCFSCommsJMConsumer

For interacting with OSM Provisioning, OSM Fulfillment pushes ProcessProvisioningOrderEBM message into AIA_CRTFO_OUT_JMSQ using store-and-forward.

The ProcessProvisioningOrderOSMCFSCommsJMConsumer is a Mediator process that has a JMS Adapter Service. This Mediator service continuously polls the AIA_CRTFO_OUT_JMSQ. The ProcessProvisioningOrderOSMCFSCommsJMConsumer dequeues the ProcessProvisioningOrderEBM and routes it to the ProcessProvisioningOrderOSMPROVCommsJMSProducer.

This service has one operation: Consume_Message.

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 store-and-forward. OSM Provisioning then consumes this message and processes it further.

This service has one operation: Initiate.

ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

For interacting with OSM COM, OSM SOM pushes ProcessFulfillmentOrderUpdateEBM message into AIA_FOPROV_OUT_JMSQ using store-and-forward.

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 ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer.

This service has one operation: Consume_Message.

ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer

The ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer 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 OSM queue.

This service has one operation: Initiate

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).

The Update Sales Order business flow is enabled by either of the following Pre-Built Integration options of the Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration):

- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option

Update Sales Order Overview

The integration uses the Update Sales Order business flow for the following purposes:

- Updating sales order data: OSM enriches the sales order with data coming from downstream provisioning systems. For example, when the provisioning system determines the service instance ID, the provisioning system can send the data to OSM to update the sales order.
- Update sales order status: OSM sends order and order line-level status updates to Siebel CRM. The updates keep the customer service representatives or self-service customer updated on the progress made as the order is fulfilled. OSM limits updates to those that are significant to the customer.

Timing Updates to Sales Order Data

If you are using an order management system other than OSM, ensure that your system only sends data updates after the order line reaches the point of no return, but before the Complete status value is sent to Siebel CRM. Because revisions on the order can be submitted from Siebel CRM up until the point of no return, data updates sent before the point of no return could be lost. Data updates sent for assets after the Complete status value is sent to Siebel CRM are not saved.

OSM and the OSM cartridges for Oracle Application Integration Architecture (Oracle AIA) obey these restrictions by default.

See "[About Creating or Updating Installed Assets in Siebel CRM](#)" for more information about assets.

Updating the Sales Order Status

OSM lets you configure and send order fulfillment statuses to your fulfillment systems and Siebel CRM. OSM translates the fulfillment function responses, each of which may contribute to different order line and order header status values, into common status attribute values.

Because order decomposition and fulfillment is a complex process, Oracle AIA uses the extended set of attributes listed in [Table 16–1](#) to provide comprehensive visibility. Order or order line status includes values for all of the attributes listed.

Table 16–1 Extended Set of Sales Order Status Attributes

Attribute Name	Usage
Order Header / Fulfillment Status	<p>Updates Siebel CRM on the current status of order fulfillment at a high level. The Fulfillment Status attribute tracks the order status while in fulfillment. Values can include In Progress, Complete, Canceled, Failed, and so on.</p> <p>The Fulfillment Status attribute is different from the Siebel CRM Status attribute. The Siebel CRM Status attribute tracks the order status across order capture and order fulfillment. The Complete and Canceled fulfillment status values only are reflected in the Siebel CRM Status attribute.</p> <p>The OSM cartridge implementer can configure the values for Fulfillment Status attributes.</p>
Order Header / Status Context	Provides details about the current status. OSM cartridge implementers can configure this value.
Order Line / Fulfillment Status	<p>Updates Siebel CRM and the order management system on the current status of order line fulfillment at a high level. Values can include In Progress, Complete, Canceled, Failed, and so on.</p> <p>The OSM cartridge implementer can configure the values for Fulfillment Status attributes.</p>
Order Line / Milestone	<p>The most recent fulfillment milestone reached. Values can include Shipped, Provisioned, Installed, and so on.</p> <p>The OSM cartridge implementer can configure the values for Milestone attributes.</p>
Order Line / Status Context	<p>Provides details about the current status. OSM cartridge implementers can configure this value to indicate:</p> <ul style="list-style-type: none"> ■ Required customer interaction. ■ If delivery is expected to be delayed. ■ The milestone or fulfillment function in which a failure occurred. ■ The cause of a cancellation or who canceled an order.
Order Line / Point-of-no-return	<p>Indicates if Siebel CRM should allow revisions to an order line or submission of previously created revisions to an order line.</p> <p>If a hard point of no return is established for an order line in OSM, OSM sends an update to Siebel CRM. Siebel CRM uses the point of no return to block users from revising order lines.</p>

Table 16–1 (Cont.) Extended Set of Sales Order Status Attributes

Attribute Name	Usage
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. OSM uses this date to communicate changes for specific order line dates to Siebel CRM.

Because of the increased processing complexity of using different fulfillment status values for different services, Oracle recommends that you use a set of streamlined status values across product specifications. Using the streamlined values makes the status updates easier for your customers and customer service representatives (CSRs) to understand and lets you reuse the flow.

Consider the following points to optimize the propagation of status updates:

- Not all status changes are relevant to the CSR or the customer. Do not propagate all changes to Siebel CRM.
- Not all status changes must be reflected instantly. Determine which status changes require instant propagation, such as reaching the point of no return, and which do not. Use a throttling mechanism to prevent performance and throughput problems which could result from too many status updates being sent at once.
- The Complete and Canceled status attribute values drive specific logic in Siebel CRM and must be preserved.
 - The Complete status value drives the logic to create and update Siebel CRM assets. The order management system 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.
 - The Canceled order status excludes the order from a Siebel CRM calculation of the future state of the asset when creating follow-on or future-dated orders.

About Creating or Updating Installed Assets in Siebel CRM

An installed asset is created when an order for a new service is fulfilled and asseted. When the order has been asseted, a CSR can use asset-based ordering to make changes to the existing services. An asset-based order (also known as change order or MACD order) references an existing installed asset and uses actions to indicate how the asset must be modified. When a change order is fulfilled, the installed asset is updated to reflect the new state.

The process integration for Order Lifecycle Management relies on the 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.

Implementing the Update Sales Order Business Flow

This chapter describes the Update Sales Order business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Overview of the Update Sales Order Business Flow

The following pre-built integration options enable the Update Sales Order business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option

The Update Sales Order business flow supports the following integration flow:

- Updating Statuses from OSM to Siebel CRM

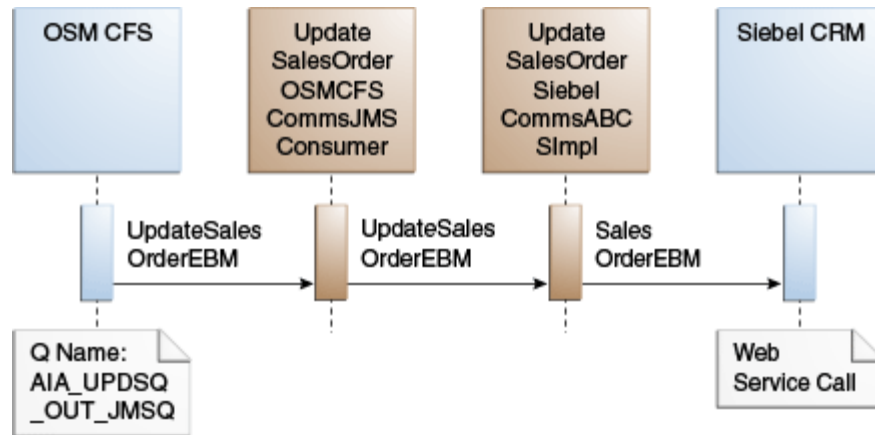
Updating Statuses from OSM to Siebel CRM Integration Flow

This integration flow uses the following services:

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCSImpl

[Figure 17-1](#) illustrates how OSM send a sales order data and status update to Siebel CRM through Oracle AIA:

Figure 17–1 Updating Statuses from OSM to Siebel CRM Sequence Diagram



When this process is initiated, the following events occur:

1. OSM creates an UpdateSalesOrderEBM message and enqueues it in the AIA_UPDSO_OUT_JMSQ queue using a store-and-forward mechanism. The UpdateSalesOrderOSMCFSCommsJMSConsumer service consumes this message and routes it to the UpdateSalesOrderSiebelCommsProvABCImpl service.

Caution: The UpdateSalesOrderOSMCFSCommsJMSConsumer service also has a sequencer. If any update to Siebel CRM causes a system or business error, further updates to the account are locked in the sequencer table until the error is fixed. If the error is a business error, the message must be removed from the sequencer table. If the error is a system error, the message must be resubmitted.

2. The UpdateSalesOrderSiebelCommsProvABCImpl service converts the UpdateSalesOrderEBM message into a Siebel CRM application business message (ABM) and invokes the Siebel CRM 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 CRM user interface to the Fulfillment Status.

In other words, the Status field in the Siebel CRM user interface represents the overarching status throughout order capture and order fulfillment. Fulfillment Status is a sub-status to a Status of Open. Fulfillment Status indicates the status of the order in OSM in the central order management role while the order is in being fulfilled.

See "[Supporting Order Priorities](#)" and "[Using the Oracle Mediator Resequencer Feature](#)" for more information about using sequencing logic to make updates to Siebel.

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 CRM causes a system or business error, further updates to the account are locked in the sequencer table until the error is fixed. If the error is a

business error, the message must be removed from the sequencer table. If the error is a system error, the message must be resubmitted.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information on system and business errors.

The following services are involved:

- UpdateSalesOrderOSMCFSCCommsJMSConsumer
- UpdateSalesOrderOSMCFSCCommsJMSConsumer_RS
- UpdateSalesOrderSiebelCommsProvABCImpl

Table 17–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
UpdateSalesOrderOSMCFSCCommsJMSConsumer consumes the message and puts it in the sequencer table defined at the Routing Service UpdateSalesOrderOSMCFSCCommsJMSConsumer_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.
UpdateSalesOrderOSMCFSCCommsJMSConsumer_RS routes the message to UpdateSalesOrderSiebelCommsProvABCImpl, 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.

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

Siebel CRM Interfaces

The Update Sales Order business flow uses the following Siebel CRM interfaces:

- SWIOrderUpsert
- SWIOrderUpsertSubProcess

These are inbound Siebel web services used to update the order information back to Siebel CRM.

See *Siebel Order Management Guide Addendum for Communications* for more information about web services.

Industry Oracle AIA Components

The Update Sales Order business flow uses the following communications industry-specific Oracle AIA components:

- SalesOrderEBO
- UpdateSalesOrderEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located in: **\$AIA_**

HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located in: **\$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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Update Sales Order business flow:

- UpdateSalesOrderOSMCFSCommsJMConsumer
- UpdateSalesOrderSiebelCommsProvABCImpl

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

See Oracle Application Integration Architecture Pre-Built Integrations Utilities Guide for more information about Session Pool Manager.

UpdateSalesOrderOSMCFSCommsJMConsumer

The UpdateSalesOrderOSMCFSCommsJMConsumer 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.

UpdateSalesOrderSiebelCommsProvABCImpl

The UpdateSalesOrderSiebelCommsProvABCImpl is a BPEL process with one operation: UpdateSalesOrder. It accepts the UpdateSalesOrderEBM as the input from the UpdateSalesOrderOSMCFSCommsJMConsumer, 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.

See *Oracle Application Integration Architecture Pre-Built Integrations Utilities Guide* for more information about Session Pool Manager.

Understanding the Process Integration for Customer Management

This chapter provides an overview of the process integration for Customer Management and describes the Synchronize Customer Account and Synchronize Customer Special Rating Profile business flows.

Customer Management Process Integration Overview

The process integration for Customer Management lets you synchronize customer information between Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM). You create customers in Siebel CRM and send their account information to BRM. You update customer data in Siebel CRM and resynchronize it to BRM through the process integration for Customer Management. This is a one-way synchronization process.

The process integration for Customer Management delivers the following business flows:

- The Synchronize Customer Account business flow enables the following integration flows:
 - The Create/Sync Customer Account integration flow interfaces customers to BRM as part of the process integration for Order Lifecycle Management.
[See "Understanding the Synchronize Fulfillment Order Billing Account Business Flow"](#) for more information.
 - The Update Customer Account integration flow updates account information (such as address, name, contact, and status) from Siebel CRM to BRM.
- The Synchronize Customer Special Rating Profile business flow synchronizes friends and family list updates from Siebel CRM to BRM.

Assumptions and Constraints for the Process Integration for Customer Management

The assumptions for the process integration for Customer Management are as follows:

1. Siebel CRM is the customer master and manages all aspects of the lifecycle from creation to updates for a customer. The process integration for Customer Management is a uni-directional flow from Siebel CRM to BRM.
2. Initial loading of customer data is not supported.
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 BRM, while the Siebel CRM service account is propagated as a non-paying sub-ordinate account in BRM.
5. Customer accounts and billing profiles are first synchronized to BRM during order processing.
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 BRM, it is current and up-to-date.
7. The Customer Account Sync process during order processing synchronizes accounts to one billing system instance (BRM) at a time. The order management system can synchronize the same customer to additional billing system instances by calling the Customer Account Sync service multiple times.

See "[Configuring Multiple BRM Instances for Communications Integrations](#)" for more information about configuring multiple billing instances.
8. The Siebel CRM account hierarchy is not synchronized to BRM. Instead, the billing account and service account relationship on a Siebel CRM order line is sent to BRM as a parent account and child account, respectively. BRM supports a single parent for a child account.

Data Requirements

The process integration for Customer Management requires the following data to successfully create customer data in 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 BRM requires the following:
 - 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.

About the 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.

See "[Understanding the Synchronize Fulfillment Order Billing Account Business Flow](#)" for more information about the Order Fulfillment 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.

About the 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 Integration Flow

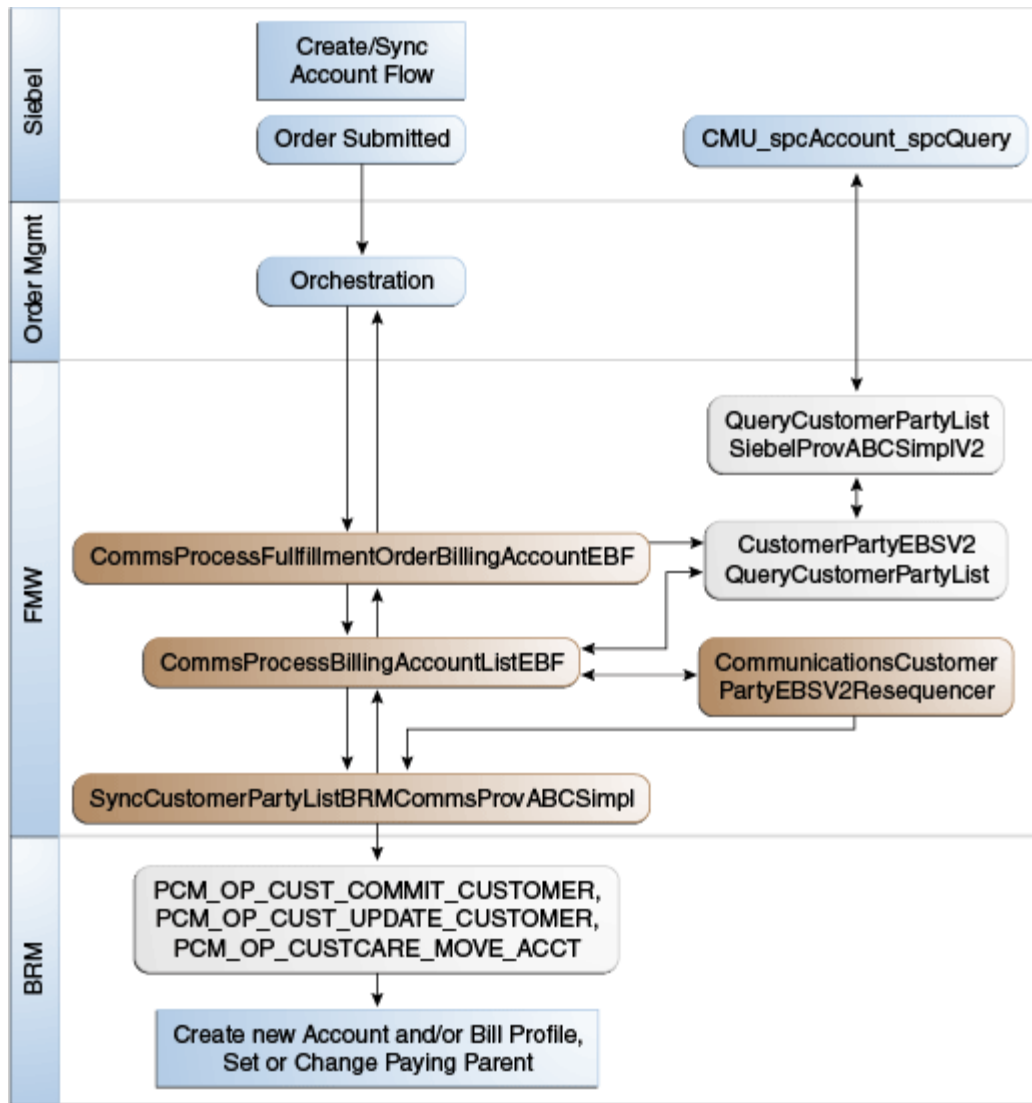


Table 18–1 provides information on Siebel CRM attributes mapped to BRM as part of the Create/Sync Account integration flow.

Table 18–1 Siebel Entities Created or Synchronized to BRM

Entity/Attributes in the Siebel CRM User Interface	Entity/Attributes in BRM Customer Center	Comments
Account	Account	--
--	Account Number	Integration sets this to the Common ID.
Account Type	Business Type	Only Siebel CRM 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.

Table 18–1 (Cont.) Siebel Entities Created or Synchronized to BRM

Entity/Attributes in the Siebel CRM User Interface	Entity/Attributes in BRM Customer Center	Comments
--	--	<p>Notes:</p> <p>Integration does not explicitly set the status when creating the customer account in BRM. BRM defaults the status to Active.</p> <p>The integration creates a two-level hierarchy in BRM with a paying parent and a subordinate service account when the billing account and the service account on the order line are different.</p> <p>See "Overview of the Synchronize Fulfillment Order Billing Account Business Flow" for more information.</p>
Contact	--	The integration only syncs the primary contact that is tied to the Account in Siebel CRM to BRM.
Mr/Mrs	Salutation	Uses the CONTACT_SALUTATION DVM.
First Name	First Name	--
Last Name	Last Name	--
Phone	Phone Number	<p>The integration maps different Siebel CRM phone number types (home, work, fax, mobile) to BRM Phone Type and Number using the PHONENUMBER_TYPE DVM.</p> <p>The phone number format should match the supported format in BRM.</p> <p>See "Using BRM with Oracle Application Integration Architecture", Validating Customer Contact Information in <i>Oracle Communications Billing and Revenue Management Concepts</i> for more information about phone number formats.</p>
Job Title	Job Title	--
Email	Email	--
Address	--	The integration only synchronizes the primary address that is tied to the Account in Siebel CRM to BRM.
Address	Address	<p>In addition to Address, fields for City, State, Postal Code, and Country are mapped.</p> <p>Uses the following DVMs: ADDRESS_COUNTRYID, ADDRESS_COUNTRYSUBDIVID, PROVINCE, STATE.</p>
Billing Profile	BillInfo	--
Name	Name	--
Frequency	Billing Frequency in Months	Uses the CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE DVM
--	Currency	<p>Integration passes account-level currency.</p> <p>Uses the CURRENCY_CODE DVM</p>

Table 18–1 (Cont.) Siebel Entities Created or Synchronized to BRM

Entity/Attributes in the Siebel CRM User Interface	Entity/Attributes in BRM Customer Center	Comments
Billing Schedule	Billing Day of Month	If the Billing Schedule is not set in and sent from Siebel CRM, then BRM defaults the Billing Day of Month. See "Setting Business Policies for Billing" in <i>Oracle Communications Billing and Revenue Management Configuring and Running Billing Guide</i> for more information about the billing schedule.
--	PayInfo	--
Payment Method	Payment Method	Only Bill Me, Credit Card or Auto-Debit is supported. Uses the CUSTOMERPARTY_PAYPROFILE_PAYMETHODECODE DVM.
Contact Last Name, First Name	Name	When the payment method is Bill Me, the Contact Name on the Siebel Billing profile is mapped to 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 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, fields for City, State, Postal Code, and Country are mapped. 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.

About the 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 process integration for Customer Management synchronizes these customer updates to BRM through the Update Customer Account integration flow.

Note: Updates are synchronized to BRM only for accounts that have been created through the order fulfillment flow.

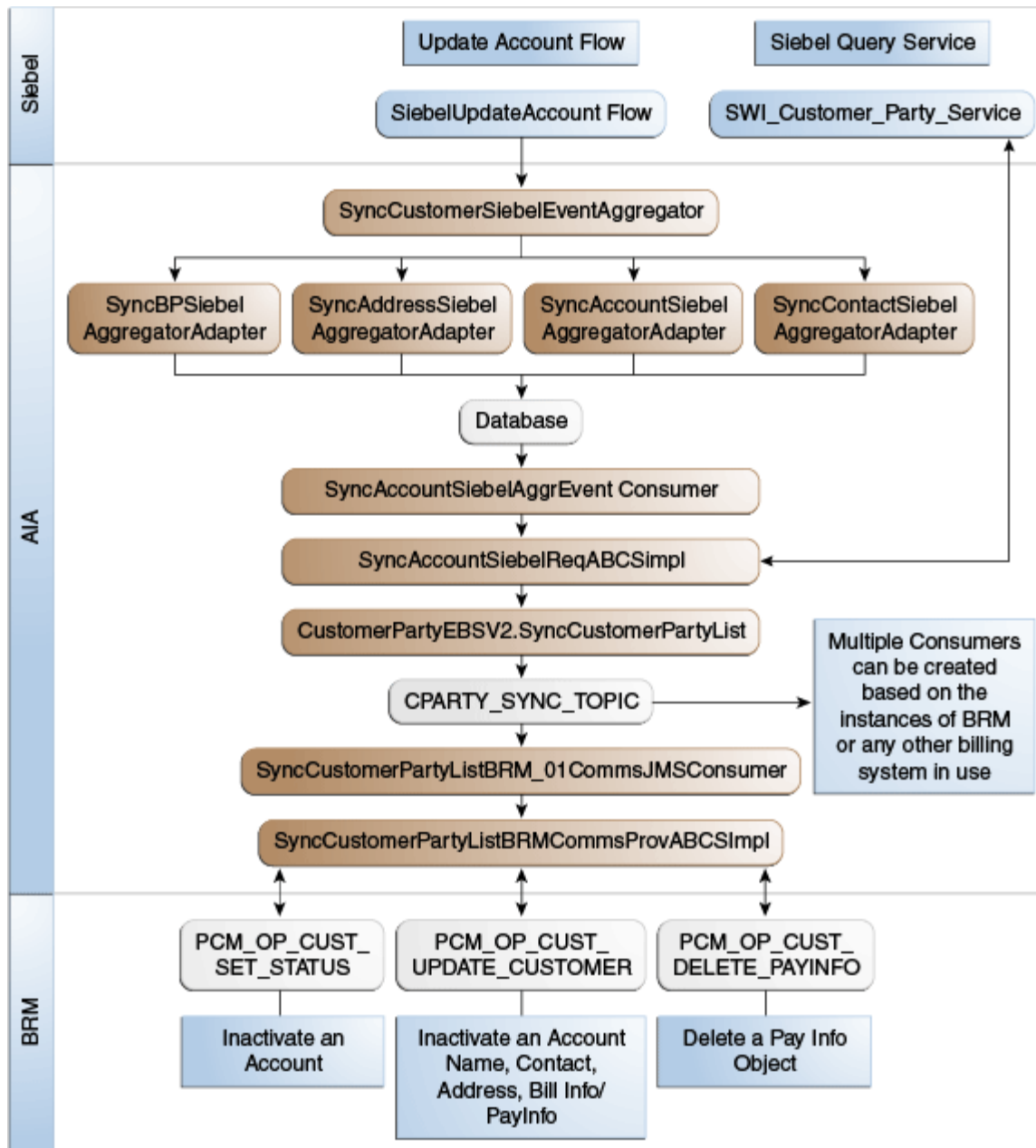
Over time customer attributes such as name, address, contact information, billing, and payment information can change. As and when customer data is changed in Siebel CRM, the process integration ensures that these changes are synchronized to BRM in real time, thereby ensuring the customer data is both consistent and current between both the applications.

A provision exists for optionally synchronizing account status updates from Siebel CRM to BRM.

See "[Account Status Synchronization Methodology](#)" for more information about the synchronization of the account status.

[Figure 18-2](#) illustrates the Update Customer Account integration flow.

Figure 18–2 Update Customer Account Integration Flow



Account Status Synchronization Methodology

The account status synchronization feature enables propagation of account status changes from Siebel CRM to BRM.

As delivered, the account status propagation to 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.

See EnableAccountStatusSync property in "[Configuring the Process Integration for Customer Management](#)" for more information about this configuration setting.

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.

See *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide* for more information about collections.

To support collections, the integration synchronizes collection actions generated by 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 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 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 BRM resulting in the cancellation of services in BRM. This is because 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 assted).

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 BRM.

About the Synchronize Customer Special Rating Profile Business Flow

Once a service that supports special rating has been purchased and the order fulfilled and assted, the customer can use the Siebel Special Rating Profile to make changes to their friends and family list. Updates are then synchronized to 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.

See "[Supporting Friends and Family Lists](#)" for more information about purchasing services that support special rating.

Implementing the Synchronize Customer Account Business Flow

This chapter describes the Synchronize Customer Account business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using Siebel CRM and BRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Overview of the Synchronize Customer Account Business Flow

The following Pre-Built Integration options enable the Synchronize Customer Account business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM and BRM Pre-Built Integration option

The Synchronize Customer Account business flow supports the following integration flows:

- Create/Sync Customer Account
- Update Customer Account

Create/Sync Customer Account Integration Flow

The Create/Sync Customer Account integration flow enables the synchronization of customer information from Siebel CRM to BRM. This flow is called during the Interfacing Orders to Create Customer Data in BRM integration flow of the Synchronize Fulfillment Order Billing Account business flow.

See "[Interfacing Orders to Create Customer Data in BRM Integration Flow](#)" for information about the sequence of events for these integration flows.

Update Customer Account Integration Flow

This flow is initiated to propagate updates to accounts in Siebel CRM to BRM.

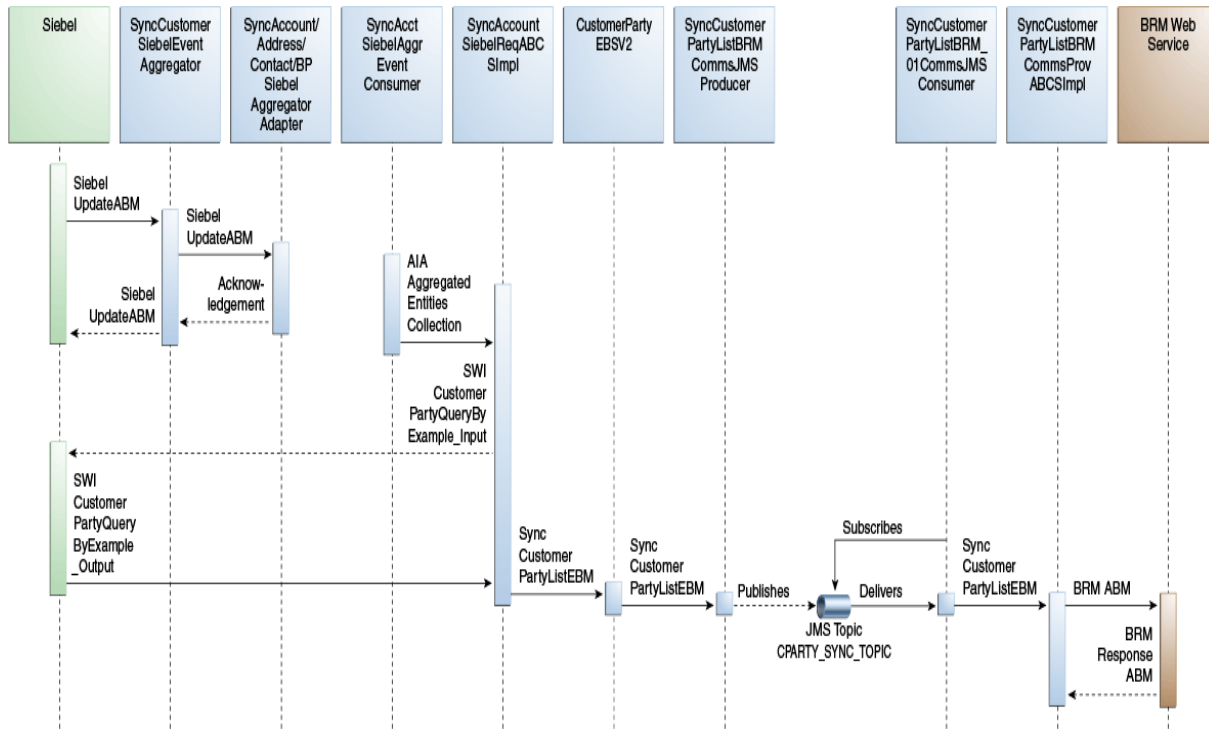
This Update Customer Account integration flow uses the following services:

- SyncCustomerSiebelEventAggregator

- SyncAcctSiebelEventAggrConsumer
- SyncAccountSiebelReqABCImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListBRM_01CommsJMSConsumer
- SyncCustomerPartyListBRMCommsProvABCImpl

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 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 BRM.

BRM Interfaces

Table 19–1 lists the BRM interfaces used by the Synchronize Customer Account business flow.

Table 19–1 RM 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 BRM.	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.	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.	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.	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 BRM.	BRM Sync Customer Provider ABCS as part of the Update Customer Account flow.

See the BRM documentation for more information.

Siebel CRM 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.

See *Siebel Order Management Guide Addendum for Communications* for more information about web services.

Table 19–3 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).

See "Workflows for Employee Asset-Based Ordering" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The Synchronize Customer Account business flow uses the following enterprise business objects (EBOs) and enterprise business messages (EBMs):

- 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/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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Synchronize Customer Account business flow:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommunicationsCustomerPartyEBSV2Resequencer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- SyncCustomerSiebelEventAggregator
- SyncAccountSiebelAggregatorAdapter
- SyncContactSiebelAggregatorAdapter

- SyncAddressSiebelAggregatorAdapter
- SyncBPSiebelAggregatorAdapter
- SyncAcctSiebelAggrEventConsumer
- SyncAccountSiebelReqABCImpl
- CustomerPartyEBSV2
- QueryCustomerPartyListSiebelProvABCImplV2
- SyncCustomerPartyListBRMCommsProvABCImpl
- SyncCustomerPartyListBRM_01CommsJMSConsumer

ProcessFulfillmentOrderBillingAccountListOSMCFSCCommsJMSConsumer

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.

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 SyncCustomerPartyListBRMCommsProvABCImpl for routing to BRM.

See "Designing and Constructing Enterprise Business Services" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* and "Understanding Enterprise Business Services," Enterprise Business Flow Processes in *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack* for more information about this EBS.

See "[Using the Oracle Mediator Resequencer Feature](#)" for more information about the Oracle Mediator Resequencer.

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.

See "Designing and Constructing Enterprise Business Flows" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* and "Understanding Enterprise Business Services," Enterprise Business Flow Processes in *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack* for more information about EBFs.

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:

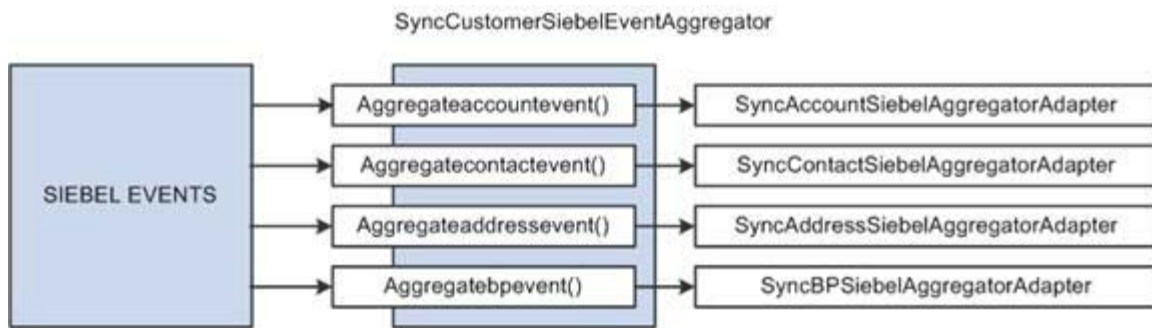
- Determining whether the customer exists and is up to date in the target billing system.
If so, optimize and do not try to create or synchronize.
- Retrieving the customer data from the appropriate Siebel CRM system using the provided IDs, if necessary.
- Optimizing, if possible, the number and size 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.

See "Designing and Constructing Enterprise Business Flows" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* and "Understanding Enterprise Business Services," Enterprise Business Flow Processes in *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack* for more information about EBFs.

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.
- **Aggregatecontactevent:**
 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.
- **Aggregateaddressevent:**
 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.

See "Describing the Event Aggregation Programming Model" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about the Event Aggregation programming model.

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.

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.

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.

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.

SyncAcctSiebelAggrEventConsumer

This service extracts the account IDs stored in the AIA_AGGREGATED_ENTITIES database table and sends them forward to the SyncAccountSiebelReqABCImpl 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.

See ["Using the Oracle Mediator Resequencer Feature"](#) for more information about the resequencer.

SyncAccountSiebelReqABCImpl

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 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 BRM systems and operates only on a single BRM system, then they can change the routing rule to route incoming requests to the SyncCustomerPartyListBRMCommsProvABCImpl_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.

CustomerPartyEBSV2

CustomerPartyEBSV2 exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object.

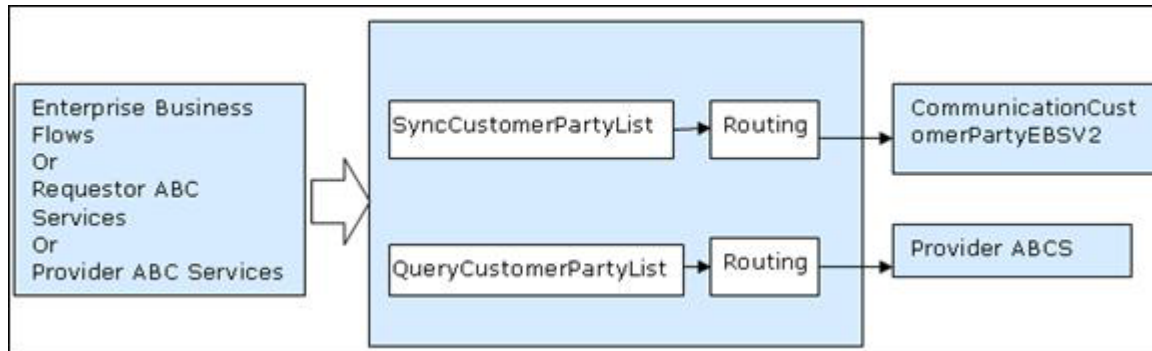
CustomerPartyEBSV2 service uses the following operations:

- 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.

See "Designing and Developing Enterprise Business Services" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* and "Understanding Enterprise Business Services," Enterprise Business Flow Processes in *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack* for more information about this EBS.

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.

SyncCustomerPartyListBRMCommsProvABCImpl

The CommsProcessBillingAccountListEBF or SyncAccountSiebelReqABCImpl service invokes SyncCustomerPartyListBRMCommsProvABCImpl. 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 BRM-specific ABM (PCM_OP_CUST_COMMIT_CUSTOMER_Inmsg).

Calls the PCM_OP_CUST_COMMIT_CUSTOMER opcode with the 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 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 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 BRM.

Transforms the SyncCustomerPartyListEBM to SyncCustomerPartyListResponseEBM.

SyncCustomerPartyListBRM_01CommsJMSSConsumer

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 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 BRM system type installation. If multiple BRM systems exist, then for each system one such consumer must be deployed.

See "[Configuring Multiple BRM Instances for Communications Integrations](#)" for more information about multiple BRM systems.

Implementing the Synchronize Customer Special Rating Profile Business Flow

This chapter describes the Synchronize Customer Special Rating Profile business flow and explains how the Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM) Pre-Built Integration option (the integration) implements the business flow using BRM interfaces, Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Overview of the Synchronize Customer Special Rating Profile Business Flow

The Synchronize Customer Special Rating Profile business flow supports the following integration flow:

- Synchronize Friends and Family List Updates to BRM

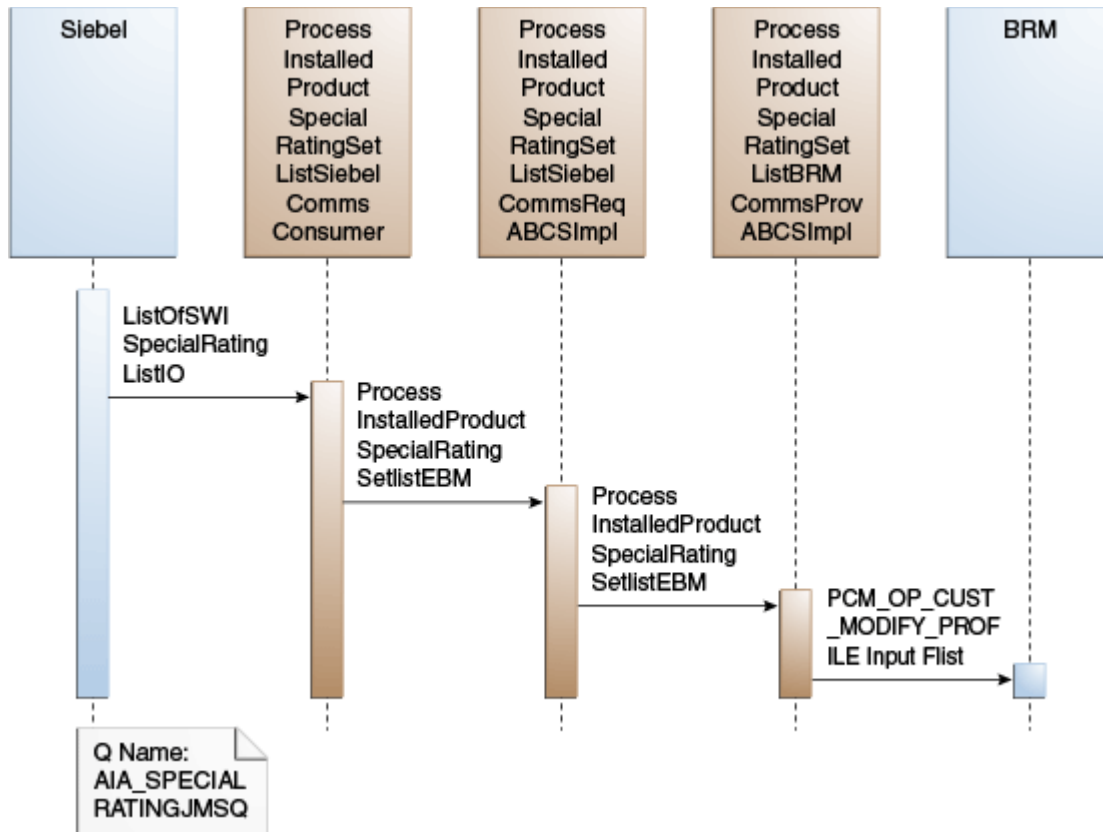
Synchronize Friends and Family List Updates to BRM Integration Flow

This integration flow uses the following services:

- `ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer`
- `ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl`
- `ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl`

[Figure 20–1](#) illustrates the sequence of events for the Synchronizing Friends and Family List Updates to BRM integration flow.

Figure 20–1 Synchronizing Friends and Family List Updates to BRM Sequence Diagram



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) queue named AIA_SPECIALRATINGJMSQ.
2. The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer service picks up the ListOfSWISpecialRatingListIO message and routes it to the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl service.
3. The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl service transforms the ListOfSWISpecialRatingListIO message into the ProcessInstalledProductSpecialRatingSetListEBM message and routes it to the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl service.
4. The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl service calls the PCM_OP_CUST_MODIFY_PROFILE BRM opcode to update the Friends and Family list information in BRM.

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.

See "Using Error Type to Control Response to Order Fallout" for more information about system and business errors.

The following services are involved:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl

Table 20–1 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer picks up message and routes it to ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl, which transforms message and routes to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl. The opcode is called to update information in BRM.	AIA cross-references updated.	Message goes back to the originating queue AIA_SPECIALRATINGJMSQ_ErrorQ.	Resubmit from AIA_SPECIALRATINGJMSQ_ErrorQ.

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

BRM Interfaces

This business flow uses the following service:

- PCM_OP_CUST_MODIFY_PROFILE

This service is used to update the special rating profile in BRM.

Siebel CRM Interfaces

This business flow 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) queue named AIA_SPECIALRATINGJMSQ.

Industry Oracle AIA Components

This business flow uses the following enterprise business message (EBM):

- ProcessInstalledProductSpecialRatingSetListEBM

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

The industry enterprise business service (EBS) WSDL files are located in: **\$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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

These services are delivered with this business flow:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMScConsumer
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl

ProcessInstalledProductSpecialRatingSetListSiebelCommsJMScConsumer

The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMScConsumer service is implemented as a Mediator process.

This consumer reads the AIA_SPECIALRATINGJMSQ queue into which Siebel CRM enqueues the SOAP-Wrapped Siebel CRM Special Rating List ABM. This consumer dequeues the messages from this queue, unwraps the message from the SOAP envelope, and routes the Siebel CRM ABM to the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl service.

ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl

The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl service is a BPEL process with one operation: ProcessInstalledProductSpecialRatingSetList.

This service accepts as input the SWISpecialRatingListIO message and converts it to the ProcessInstalledProductSpecialRatingSetListEBM message before routing it to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl service.

This service is invoked when the existing special rating (friends and family) profile in Siebel CRM for an account that exists in Siebel CRM and is synchronized to BRM is modified.

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.

ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl

The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl service is implemented as a BPEL process with a single operation: ProcessInstalledProductSpecialRatingSetList.

This service is invoked when the existing special rating (friends and family) profile in Siebel CRM for an account that exists in Siebel CRM and is synchronized to BRM is modified.

This service is the BRM ABCS implementation, which converts the `ProcessInstalledProductSpecialRatingSetList` message into the BRM ABM before invoking the `PCM_OP_CUST_MODIFY_PROFILE` BRM opcode.

Understanding the Process Integration for Order Fallout Management

This chapter provides an overview of the process integration for Order Fallout Management and discusses capturing faults, order fallout management process integration business flows, and how to extend fault messages to capture order fallout information.

Overview of the Process Integration for Order Fallout Management

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 Integration Pack 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 Application Integration Architecture (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 CRM 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.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about the Oracle AIA Error Handling Framework.

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 to 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 Communications Billing and Revenue Management (BRM) because it has bad data. In this case the sales order must be revised and resubmitted from Siebel CRM to Oracle Communications Order and Service Management (OSM) for fulfillment, and then from OSM to 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.

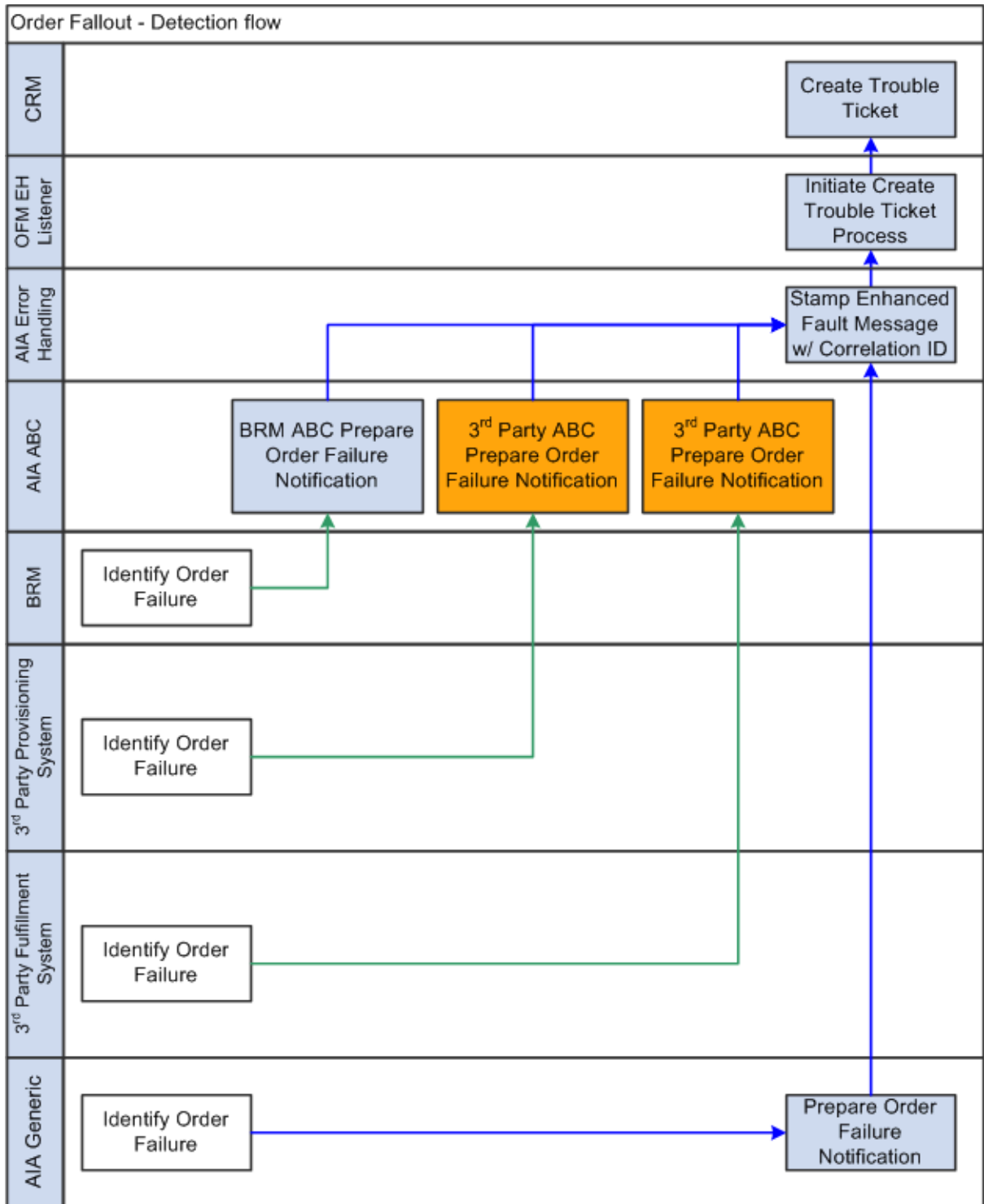
See "[Configuring the Process Integration for Order Fallout Management](#)" for more information about how to configure the process integration for order fallout management.

About 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 for Order Fallout



About 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).

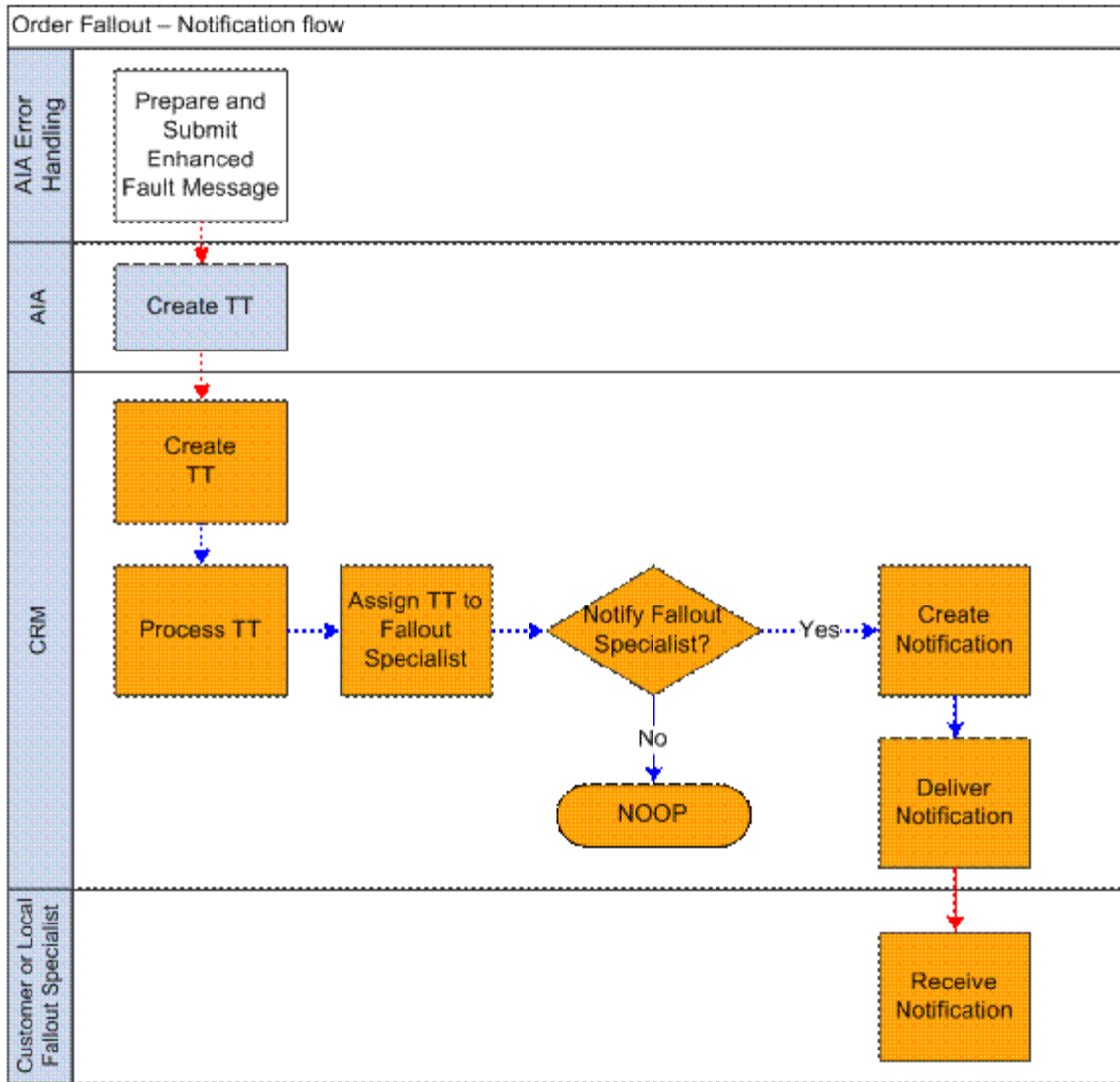
The AIA Enhanced Fault Message has some following key error and order failure information:

- Faulting Service
- Error Code
- Error Severity
- Error Text
- Time Of Failure
- Order ID
- Order Number
- Order Originating System Code
- Account ID
- Account Name

See "[Extending Fault Messages to Capture Order Fallout Information](#)" for more information about extending fault messages.

[Figure 21–2](#) illustrates the notification subprocess within the order fallout process.

Figure 21–2 Order Fallout Notification Flow



About 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 OSM in the central order management role (OSM COM).

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 OSM.

Note: Siebel CRM and 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 BRM. The order then resumes in OSM.

Figure 21–3 illustrates the Siebel CRM correction flow subprocess within the order fallout management process.

Figure 21–3 Siebel CRM Correction Flow for Order Fallou

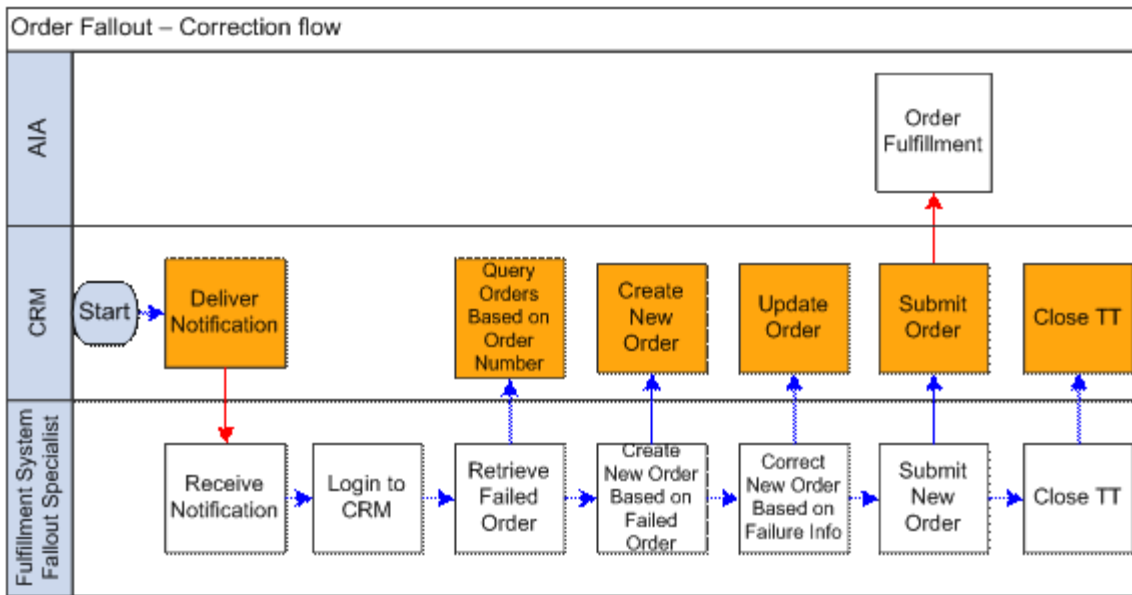
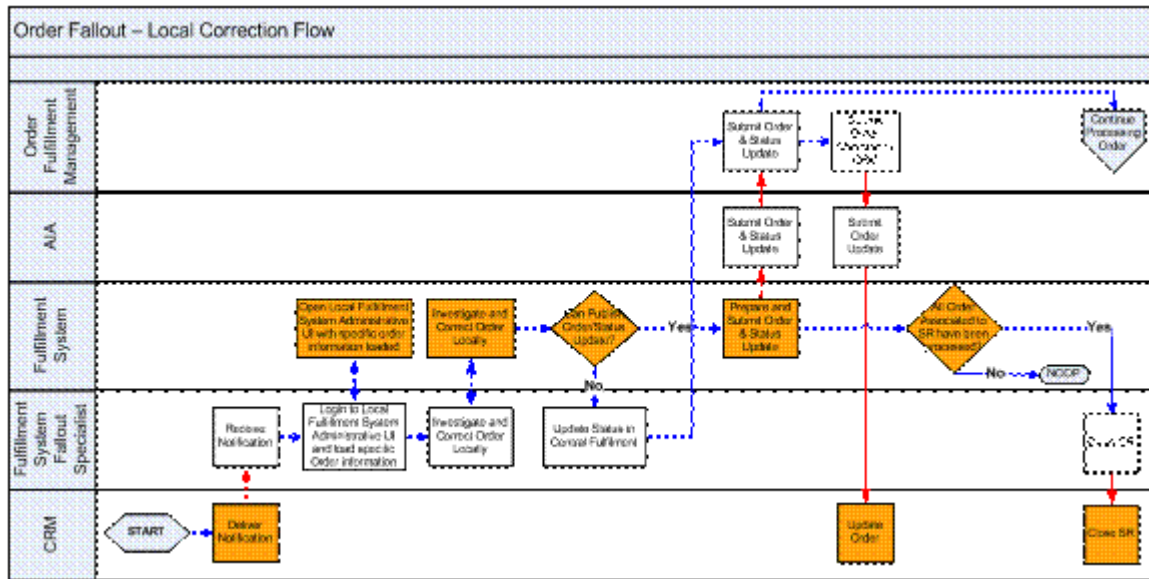


Figure 21–4 illustrates the local correction flow subprocess within the order fallout management process that must take place to undo, compensate, or otherwise fix changes that were committed locally within a fulfillment system for a failed order.

Figure 21-4 Local Correction Flow

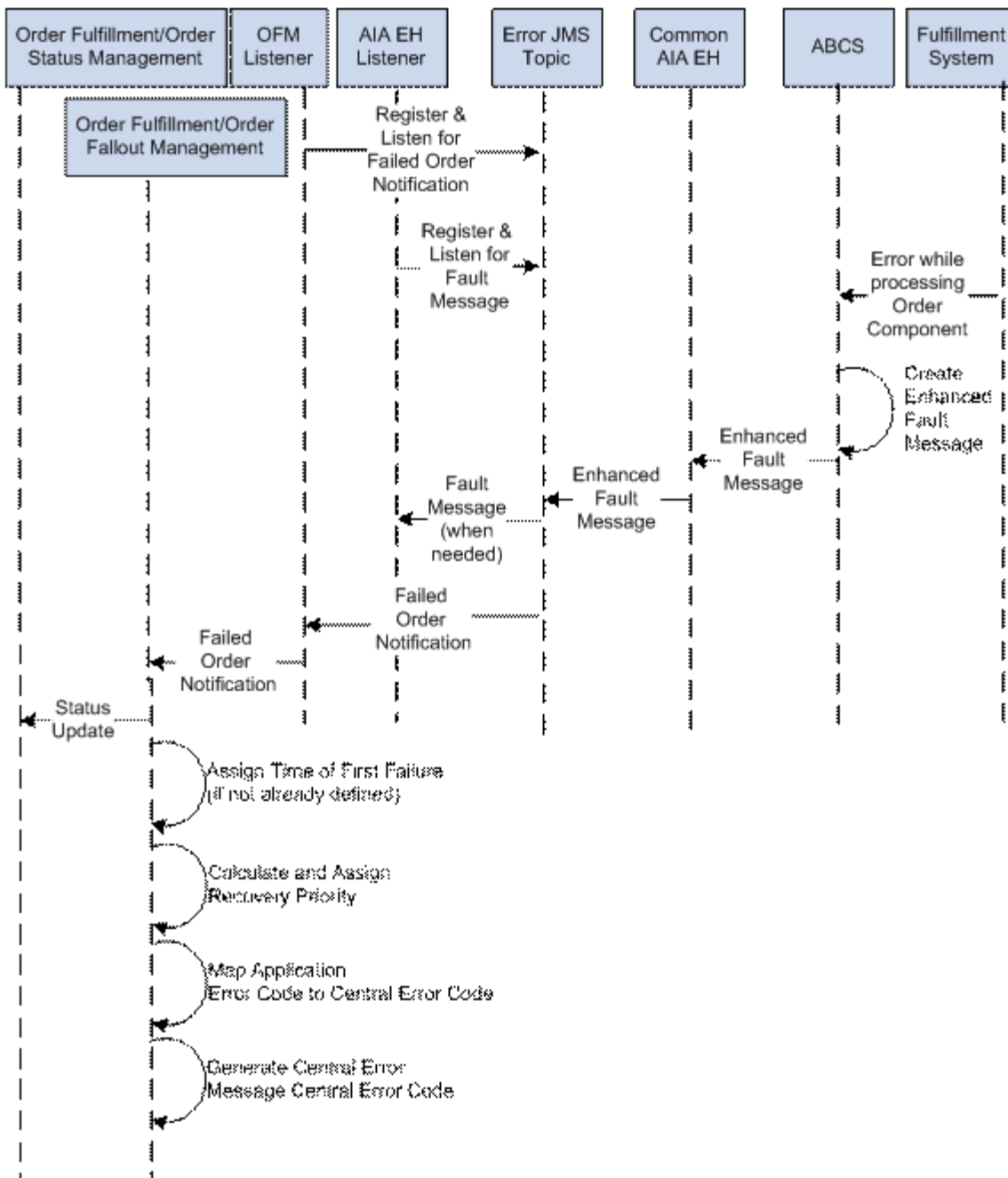


How Oracle AIA Error Handling Framework Captures Faults

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 BRM.

Figure 21–5 Capturing the Fault Sequence Diagram

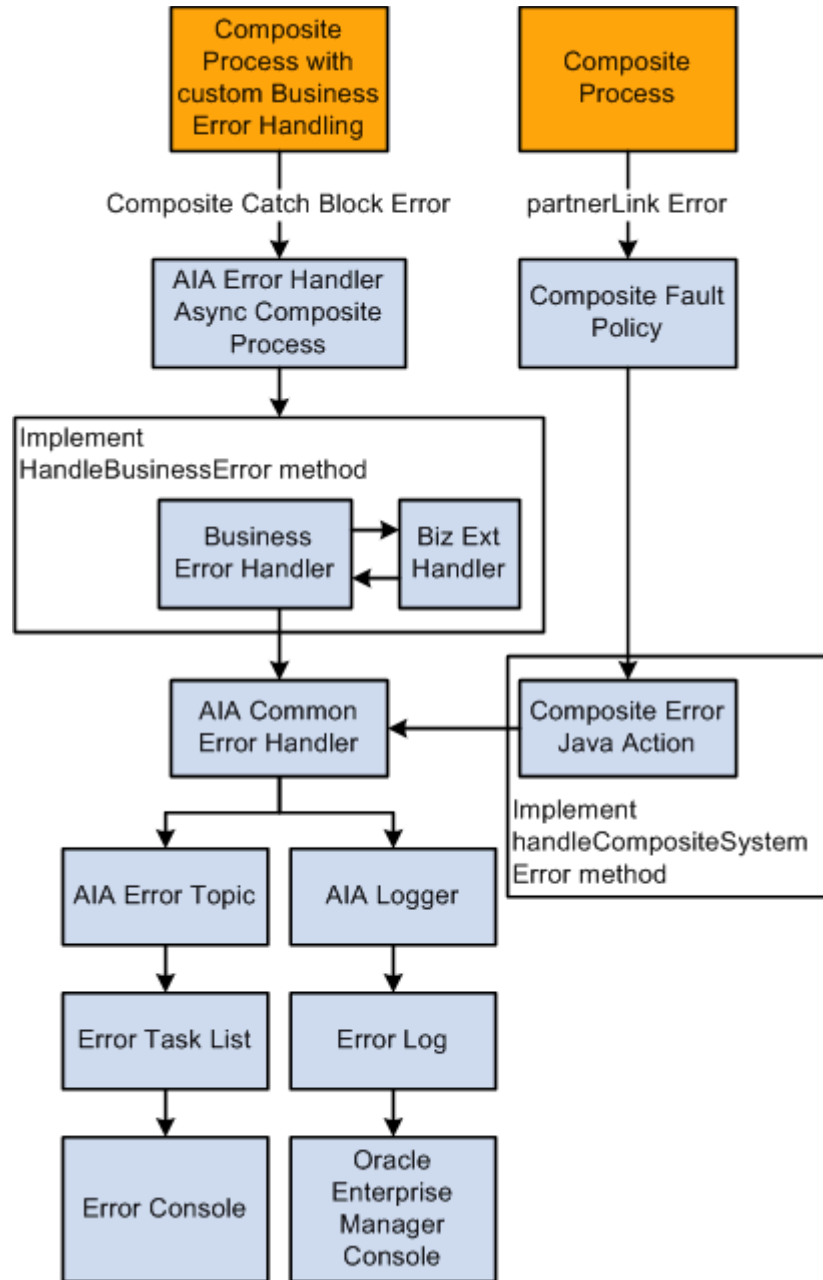


The Oracle AIA Error Handling Framework:

- 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.

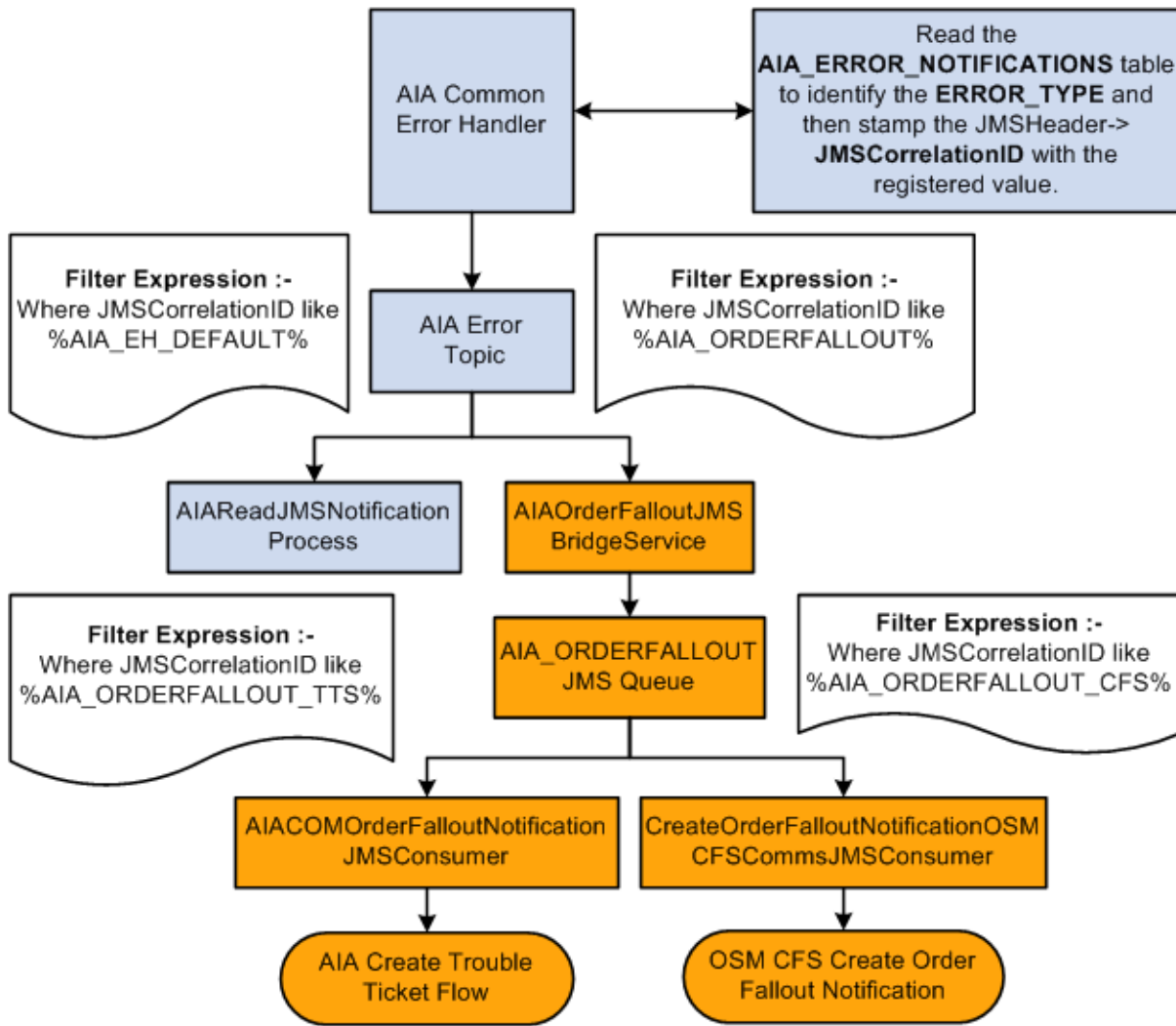
Figure 21–6 illustrates how the Oracle AIA Error Handling Framework is leveraged to submit an order failure notification to the AIA Error Topic.

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



The flow proceeds as follows:

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 OSM CFS.

See "Using Error Type to Control Response to Order Fallout" for more information on how to set up the seed data so that the trouble ticket is created from either Oracle AIA or OSM.

5. The AIACOMOrderFalloutNotificationJMSCConsumer 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 OSM and OSM initiates the Create Trouble Ticket request.

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 OSM business flows.

Create Trouble Ticket from Oracle AIA

This business flow is enabled by either the Oracle Communications Order to Cash for Siebel CRM and BRM Pre-Built Integration option or the Oracle Communications Order to Cash Siebel CRM, OSM, and 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 OSM

This business flow is enabled by either the Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option or the Oracle Communications Order to Cash Siebel CRM, OSM, and BRM 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 OSM.

The following integration flows enable this business flow:

- Order Failure Notification to OSM integration flow
- Creating a Trouble Ticket in Siebel CRM from OSM integration flow
- Updating a Trouble Ticket in Siebel CRM from OSM integration flow

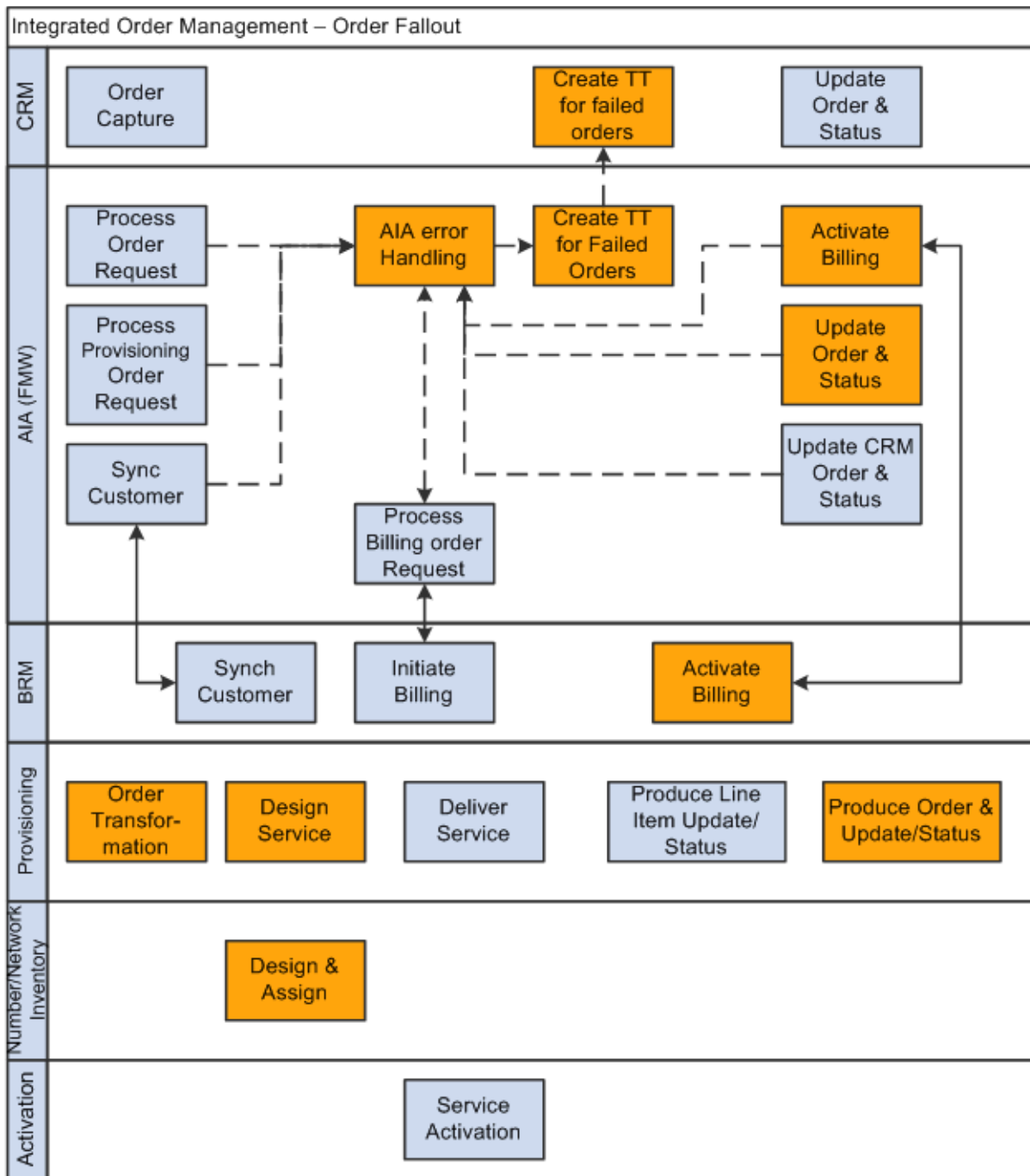
Create Trouble Ticket from Oracle AIA Business Flow

The Create Trouble Ticket from Oracle AIA business flow provides an alternative solution for order fallout management in which 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.

See "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling and Extending Fault Messages in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about extending error handling.

Assumptions and Constraints for the Create Trouble Ticket from Oracle AIA Business Flow

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.

Create and Manage Trouble Ticket from OSM Business Flow

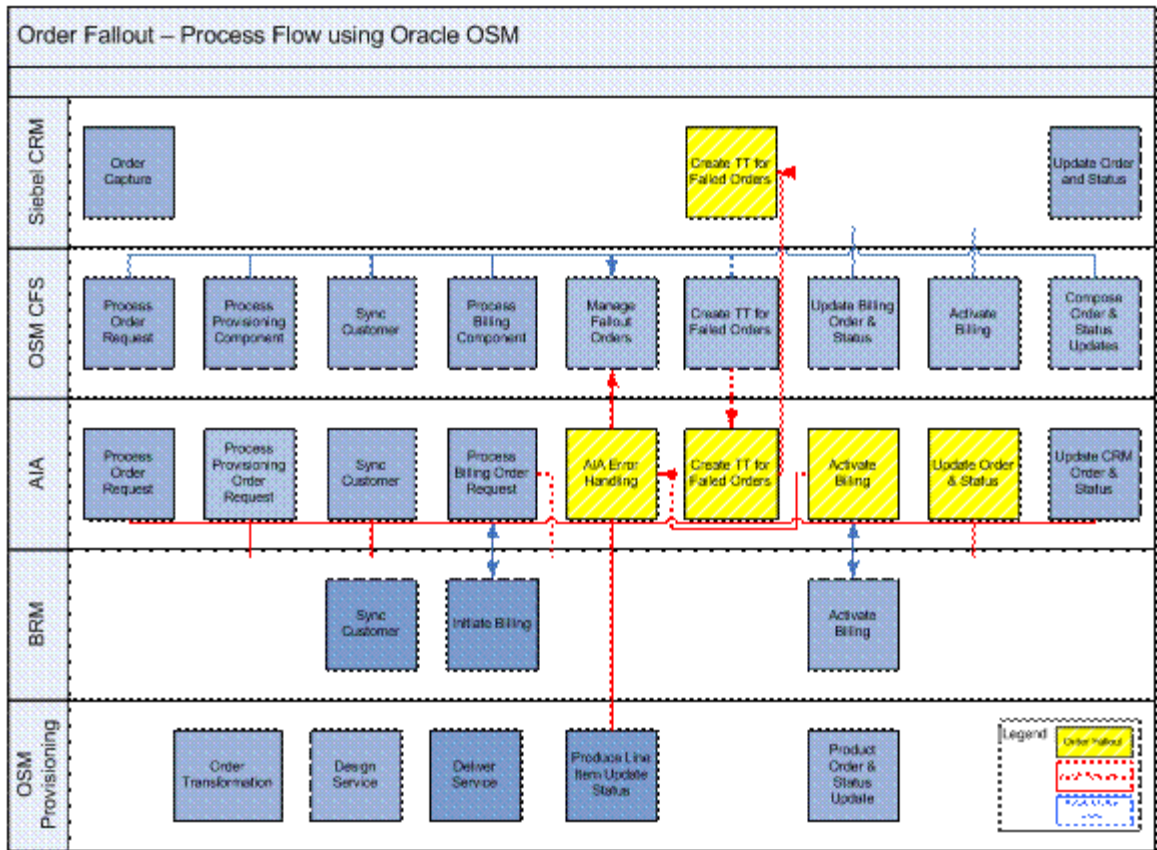
Oracle AIA or OSM can initiate the creation of trouble tickets. This is configurable. Installing either the Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option or the Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option automatically configures order fallout to occur in OSM.

With the combination of Siebel CRM and OSM:

- Trouble tickets are created in Siebel CRM from 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 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 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 OSM in Siebel CRM for the failed order:

Figure 21–9 Order Fallout Process Flow Using OSM



Assumptions and Constraints for the Create and Manage Trouble Ticket from OSM Business Flow

The assumptions and constraints for the Create and Manage Trouble Ticket from OSM business flow are as follows:

- Order fallout management functionality handles orders that fail after being submitted by Siebel CRM.
- When an order revision fails upon arrival in 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.

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.

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.

See the discussion of extending fault messages in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about extending error handling.

[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

Table 21–1 (Cont.) Order Header-Level Data

Field Name	Type	Description	Source	Optional
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
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

See "[Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant](#)" for more information about how to pass this information.

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.

See "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Extending Error Handling in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about extending error handling.

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.

Implementing the Create Trouble Ticket from Oracle AIA Business Flow

This chapter describes the Create Trouble Ticket from Oracle AIA business flow and explains how the Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM) Pre-Built Integration option (the integration) implements the business flow using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, integration services, and fallout-enabled services.

Create Trouble Ticket from Oracle AIA Business Flow 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)
- CreateTroubleTicketSiebelCommsProvABCImpl - 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.

This business flow supports the following integration flows:

- Creating a Trouble Ticket in Siebel CRM

Creating a Trouble Ticket in Siebel CRM Integration Flow

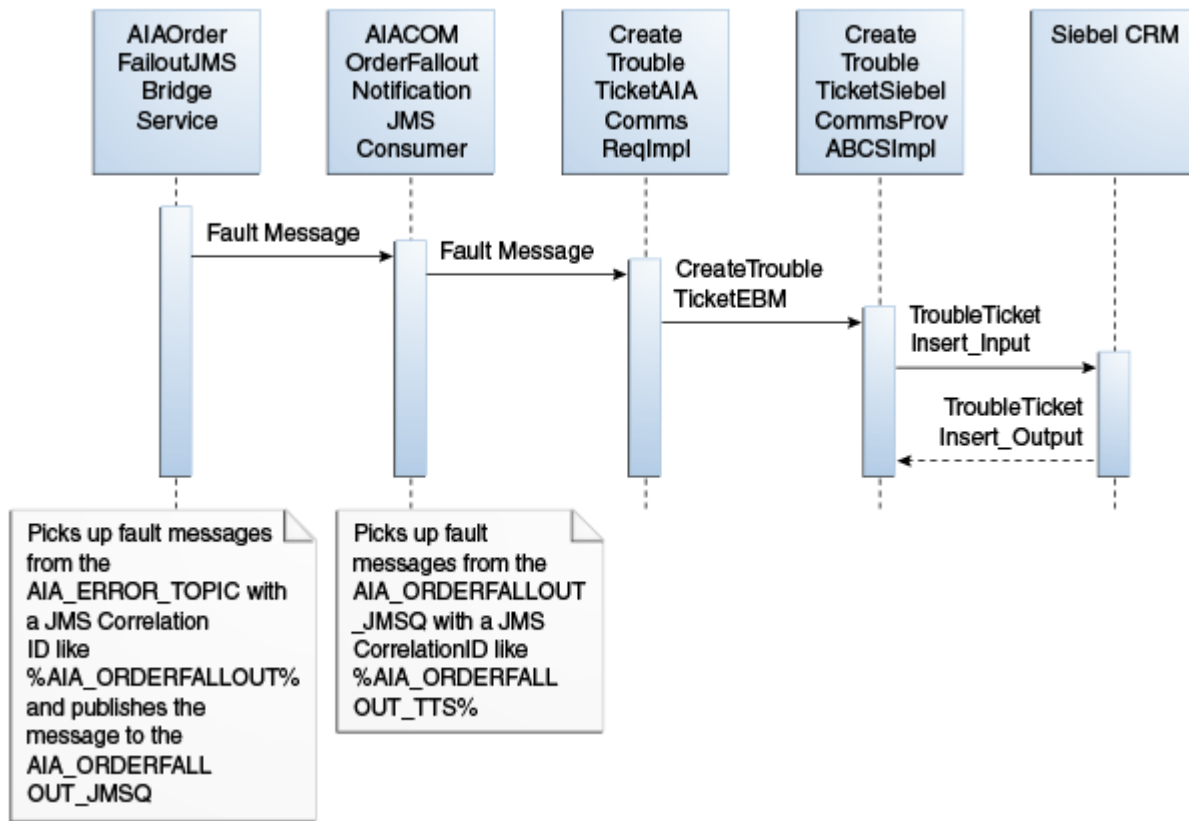
This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl

- CreateTroubleTicketSiebelCommsProvABCImpl

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 CreateTroubleTicketSiebelCommsProvABCImpl.
6. The CreateTroubleTicketSiebelCommsProvABCImpl 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.

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.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information on system and business errors.

The following services are involved:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCImpl

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 CreateTroubleTicketSiebelCommsProvABCImpl.	AIA cross-reference entries.	Rollback the message to AIA_ORDERFALLOUT_JMSQ.	Resubmit from AIA_ORDERFALLOUT_JMSQ.

See "[Configuring Oracle AIA Processes for Error Handling and Trace Logging](#)", [Configuring Fault Policies to Not Issue Rollback Messages in Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack](#) for more information about rollback procedure.

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.

Siebel CRM Interfaces

The Create Trouble Ticket from Oracle AIA business flow uses the following 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.

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The Create Trouble Ticket from Oracle AIA business flow uses the following communications industry-specific Oracle AIA components:

- TroubleTicketEBO
- CreateTroubleTicketEBM
- CreateTroubleTicketResponseEBM
- CommunicationsTroubleTicketEBSV1.wsdl

The industry EBO and EBM XML schema (XSD) files are located in: **\$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1/**

The industry EBS web service description language (WSDL) files are located in: **\$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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Create Trouble Ticket from Oracle AIA business flow:

- CreateTroubleTicketSiebelCommsProvABCSEImpl
- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- AIAOrderFalloutErrorHandlerExtension.java

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

See *Oracle Application Integration Architecture Pre-Built Integrations Utilities Guide* for more information about Session Pool Manager.

CreateTroubleTicketSiebelCommsProvABCImpl

The CreateTroubleTicketSiebelCommsProvABCImpl service 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.

CreateTroubleTicketSiebelCommsProvABCImpl 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.

See *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide* for more information about Session Pool Manager.

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 OSM.

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.

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.

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.

Business Flow Fallout-Enabled Services

The following Create Trouble Ticket from Oracle AIA business flow services are fallout-enabled:

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsProvABCImpl
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCImplV2
- SyncCustomerPartyListBRMCommsProvABCImpl

Implementing the Create and Manage Trouble Ticket from OSM Business Flow

This chapter describes the Create and Manage Trouble Ticket from OSM business flow and explains how the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration) implements the business flow using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, integration services, and fallout-enabled services.

Overview of the Create and Manage Trouble Tickets from OSM Business Flow

The following Pre-Built Integration options enable the Create and Manage Trouble Ticket from OSM business flow:

- Oracle Communications Order to Cash for Siebel CRM, OSM, and BRM Pre-Built Integration option
- Oracle Communications Order to Cash for Siebel CRM and OSM Pre-Built Integration option

The Create and Manage Trouble Ticket from OSM business flow supports the following integration flows:

- Order Failure Notification to 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 in OSM.
- Creating a Trouble Ticket in Siebel CRM from OSM:
Creates trouble tickets in Siebel CRM for individual and batch or bulk orders from OSM.
- Updating a Trouble Ticket in Siebel CRM from OSM:
Updates trouble tickets in Siebel CRM from OSM.

Order Failure Notification to 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

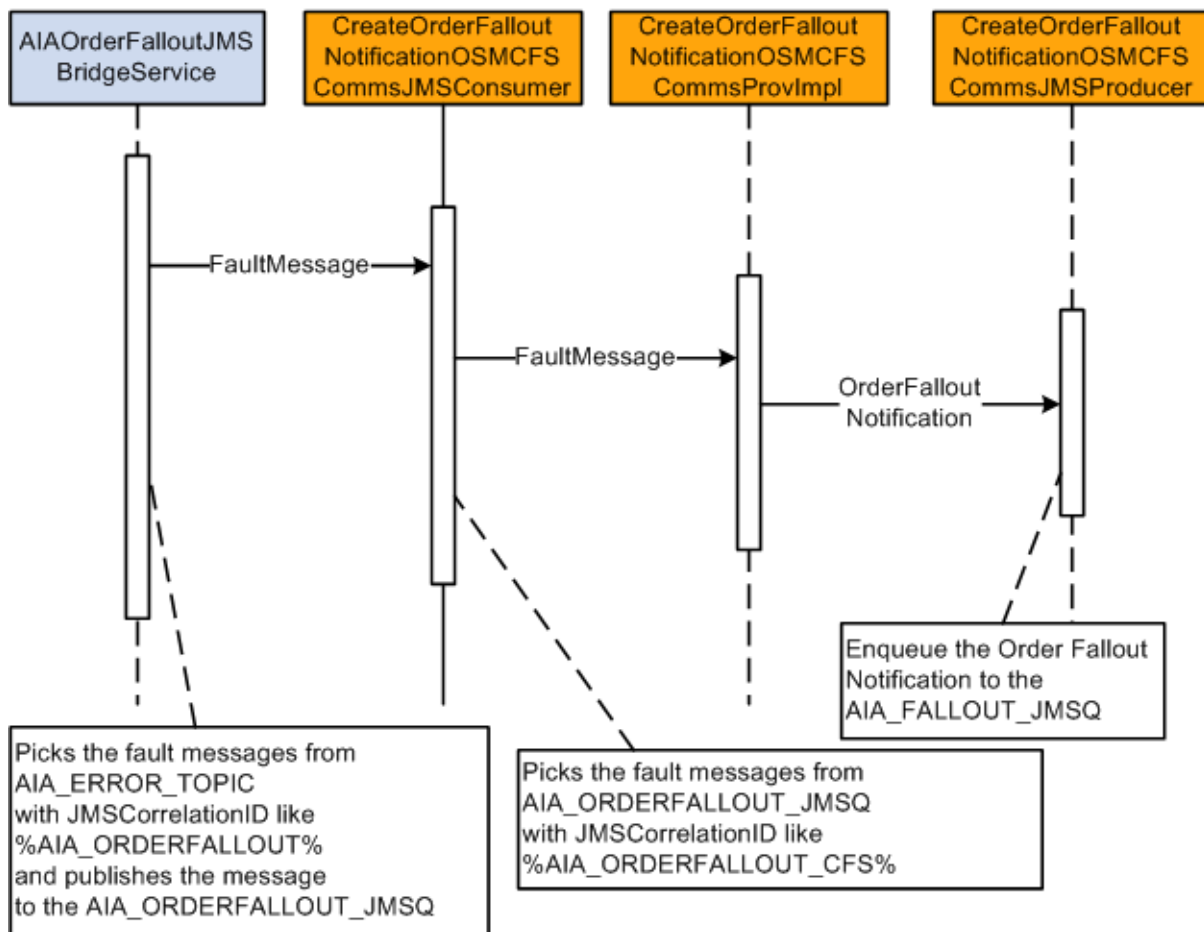
errors or faults specific to order fallout by looking at the correlation ID. These fault messages are enriched and published to the central OFM in OSM.

This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

Figure 23–1 illustrates how OSM initiates the request to create a trouble after receiving an order failure notification.

Figure 23–1 Order Failure Notification to 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. `AIACOMOrderFalloutNotificationJMSCConsumer` 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. OSM CFS picks up this message, marks the order as failed, and initiates the request to create a trouble ticket.

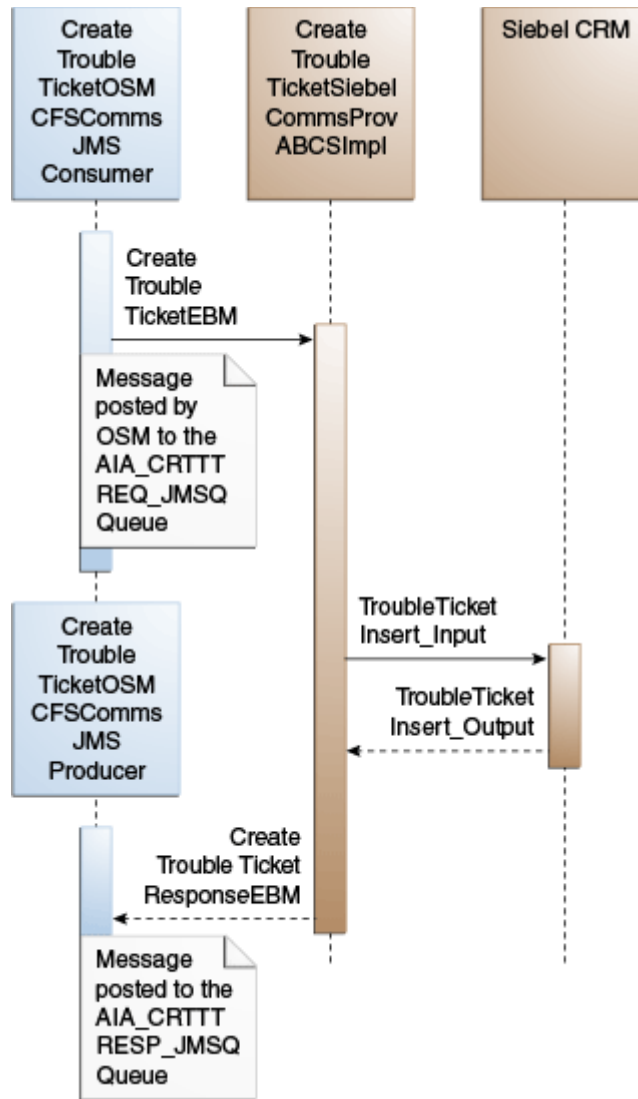
Creating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow uses these interfaces:

- `CreateTroubleTicketOSMCFSCommsJMSCConsumer`
- `CreateTroubleTicketSiebelCommsProvABCImpl`
- `CreateTroubleTicketOSMCFSCommsJMSProducer`

[Figure 23–2](#) describes the creation of trouble tickets in Siebel CRM from OSM.

Figure 23–2 Creating a Trouble Ticket in Siebel CRM from 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. CreateTroubleTicketOSMCFSCOMMSJMSSConsumer 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. OSM CFS picks up the message and stores the `TroubleTicketID` for reference.

Defining Transaction Boundaries and Recovery Details

For the Order Failure Notification to OSM and Creating a Trouble Ticket in Siebel CRM from 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.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information about system and business errors.

The following services are involved:

- `AIAOrderFalloutJMSBridgeService`
- `CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer`
- `CreateOrderFalloutNotificationOSMCFSCommsProvImpl`
- `CreateOrderFalloutNotificationOSMCFSCommsJMSProducer`
- `CreateTroubleTicketOSMCFSCommsJMSConsumer`
- `CreateTroubleTicketSiebelCommsProvABCImpl`
- `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 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 CreateTroubleTicketSiebelCommsProvABCImpl, which invokes Siebel web service. Response trouble ticket ID is received. Response message is created, which invokes CreateTroubleTicketSiebelCommsProvABCImpl, 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.

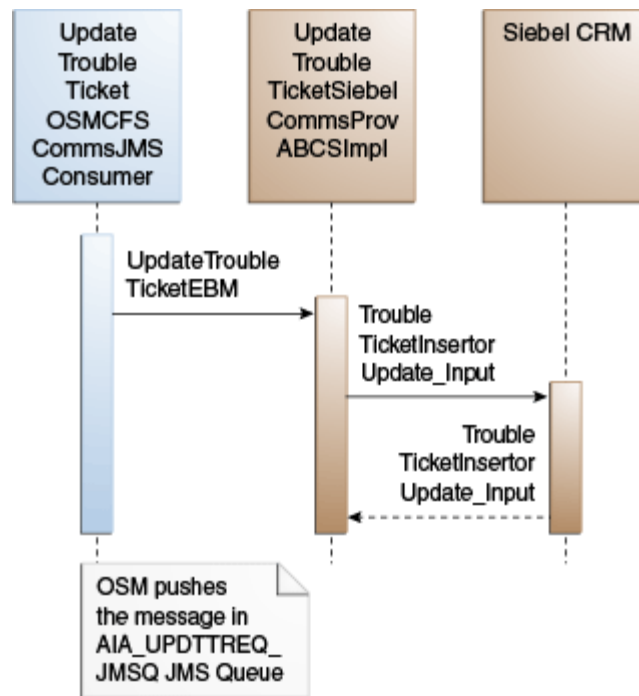
See "Configuring Oracle AIA Processes for Error Handling and Trace Logging", Configuring Fault Policies to Not Issue Rollback Messages in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about rollback procedures.

Updating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow uses the following interfaces:

- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCImpl

Figure 23–3 describes the Trouble Ticket Update flow from OSM to Siebel CRM.

Figure 23–3 Updating a Trouble Ticket in Siebel CRM from OSM

When this process initiates, the following events occur:

1. The OSM fulfillment system produces the UpdateTroubleTicketEBM in the AIA_UPDTTREQ_JMSQ SAF queue.
2. UpdateTroubleTicketOSMCFSCCommsJMSConsumer picks up the message from the queue and invokes the UpdateTroubleTicketSiebelCommsProvABCImpl service.
3. The UpdateTroubleTicketSiebelCommsProvABCImpl service transforms the EBM to TroubleTicketInsertorUpdate_Input ABM, and the Siebel web service is invoked to update the trouble ticket.

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.

See "[Using Error Type to Control Response to Order Fallout](#)" for more information about system and business errors.

The following services are involved:

- UpdateTroubleTicketOSMCFSCCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCImpl

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.

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

Siebel CRM Interfaces

The Create and Manage Trouble Ticket from OSM business flow uses the following 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.

See *Siebel Order Management Guide Addendum for Communications* for more information on web services.

Industry Oracle AIA Components

The Create and Manage Trouble Ticket from OSM business flow uses the following communications industry-specific Oracle AIA components:

- TroubleTicketEBO
- CreateTroubleTicketEBM
- CreateTroubleTicketResponseEBM
- UpdateTroubleTicketEBM

The industry enterprise business object (EBO) and EBM XSD files are located in: **\$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1/**

The industry EBS WSDL files are located in: **\$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).

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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about:

- Configuring and using OER as the Oracle SOA Repository to provide the AIA Reference doc link
- Extending existing schemas and EBOs

Integration Services

The following services are delivered with the Create and Manage Trouble Ticket from OSM business flow:

- CreateTroubleTicketSiebelCommsProvABCImpl
- UpdateTroubleTicketSiebelCommsProvABCImpl
- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer
- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketRespOSMCFSCommsJMSProducer
- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- CreateFaultNotificationLFCommsJMSConsumer

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

See *Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide* for more information about Session Pool Manager.

See "[Configuring the Process Integration for Order Fallout Management](#)" for more information.

UpdateTroubleTicketSiebelCommsProvABCImpl

The UpdateTroubleTicketSiebelCommsProvABCImpl is a service that acts as the provider for Siebel CRM Update Trouble Ticket functionality. This service does not return any response.

The UpdateTroubleTicketSiebelCommsProvABCImpl 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 OSM CFS.

CreateOrderFalloutNotificationOSMCFSCommsProvImpl

The CreateOrderFalloutNotificationOSMCFSCommsProvImpl service is a BPEL process that picks up the Fault message from

CreateOrderFalloutNotificationOSMCFSCCommsJMSConsumer. It parses the Fault message and then constructs the OrderFalloutNotification message.

Next, the CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer service is invoked to enqueue the order fallout notification message in the AIA_FALLOUT_JMSQ SAF queue for OSM CFS. This action consumes and triggers a fallout event for the particular order.

CreateOrderFalloutNotificationOSMCFSCCommsJMSConsumer

The CreateOrderFalloutNotificationOSMCFSCCommsJMSConsumer service is a Mediator service that picks up the Fault message from the AIA_ORDERFALLOUT_JMSQ. It passes the Fault message to the CreateOrderFalloutNotificationOSMCFSCCommsProvImpl process based on the JMS Correlation ID.

This service acts as a consumer, listening to the messages produced in AIA_ORDERFALLOUT_JMSQ.

CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer

The CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer is a BPEL process that enqueues the OrderFalloutNotification message to the AIA_FALLOUT_JMSQ SAF queue. OSM then picks the message from this queue and triggers a fallout event in OSM. The CreateOrderFalloutNotificationOSMCFSCCommsProvImpl service invokes this service.

CreateTroubleTicketOSMCFSCCommsJMSConsumer

The CreateTroubleTicketOSMCFSCCommsJMSConsumer is a Mediator service that picks up the CreateTroubleTicketEBM message from the AIA_CRTTTREQ_JMSQ SAF queue. It routes the message to the CreateTroubleTicketSiebelCommsProvABCImpl service. This service acts as a consumer, listening to the messages produced in the AIA_CRTTTREQ_JMSQ SAF queue.

CreateTroubleTicketRespOSMCFSCCommsJMSProducer

The CreateTroubleTicketRespOSMCFSCCommsJMSProducer is a BPEL process that enqueues the CreateTroubleTicketResponseEBM message to the AIA_CRTTTRESP_JMSQ SAF queue. OSM then picks up the message from this queue and then updates the order task with the created trouble ticket ID.

UpdateTroubleTicketOSMCFSCCommsJMSConsumer

The UpdateTroubleTicketOSMCFSCCommsJMSConsumer is a Mediator service that picks up the UpdateTroubleTicketEBM message from the AIA_UPDTTREQ_JMSQ SAF queue and invokes the UpdateTroubleTicketSiebelCommsProvABCImpl service. This service acts as a consumer, listening to the messages produced in the AIA_UPDTTREQ_JMSQ SAF queue.

CreateFaultNotificationLFCCommsJMSConsumer

The CreateFaultNotificationLFCCommsJMSConsumer 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 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.

Business Flow Fallout-Enabled Services

The following Create and Manage Trouble Ticket from OSM business flow services are fallout-enabled:

- UpdateSalesOrderSiebelCommsProvABCImpl
- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer
- ProcessFOBillingAccountListRespOSMCFSCCommsJMSProducer
- ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer
- TestOrderOrchestrationEBF
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMConsumer
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMConsumer_RS
- default.ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl.ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl_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
- ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer
- OSMPROV
- OSM.ABCS.ConsumeProvUpdate_RS

- OSM.ABCS.ConsumeProvUpdate
- OSM.ABCS.Consume_ProcessProvOrder
- OSM.ABCS.Consume_ProcessProvOrder_RS

Part II

Configuring the Process Integrations

Part II contains the following chapters:

- [Configuring the Process Integration for Product Lifecycle Management](#)
- [Configuring the Process Integration for Order Lifecycle Management](#)
- [Configuring the Process Integration for Customer Management](#)
- [Configuring the Process Integration for Order Fallout Management](#)

Configuring the Process Integration for Product Lifecycle Management

This chapter discusses prerequisites and procedures for configuring the process integration for Product Lifecycle Management (PLM) for the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration). Configuration includes:

- Configuring BRM and Siebel CRM to integrate with Oracle Application Integration Architecture (Oracle AIA)
- Working with domain value maps (DVMs) and cross-references
- Handling error notifications
- Setting Oracle AIA configuration properties

Prerequisites for Integrated Product Lifecycle Management

These are the prerequisites for the integration for product management:

1. BRM must be set up before you can create billing products.
2. The following pricing objects and data must be created in the 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 BRM and associate them with billing events and billing rate plans

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 BRM to identify changes (additions, deletions, modifications) that trigger the integration flow to propagate those billing product changes and make the corresponding changes to Siebel CRM billing products.

Configuring BRM for Integrated Product Lifecycle Management

To configure BRM for PLM:

1. **Create services and events:** new services must be added before a BRM price list is created. 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 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. (Optional) **Define tax codes and tax suppliers:** 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 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:** see *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating* for more information on setting up pricing for friends and family.
12. **Install, configure, and run Synchronization Queue Data Manager (DM):** this DM enables you to synchronize changes in the 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.

See *Oracle Communications Billing and Revenue Management Synchronization Queue Data Manager* for more information.
13. **Set the 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.

See *Oracle Communications Billing and Revenue Management Developer's Guide* for more information on effective start and end dates based on the BRM EAI parameter.

Configuring Siebel CRM for Integrated Product Lifecycle Management

To configure Siebel CRM for PLM:

1. Install ACR 474. See *Siebel Maintenance Release Guide* for information about how to install ACR 474.
2. Set the `UTCCanonical` process property to **Y** for certain Siebel CRM interfaces. The instructions for ACR 474 and ACR 508 in *Siebel Maintenance Release Guide* explain which Siebel CRM interfaces you must set the `UTCCanonical` process property for.
3. Create Siebel CRM price lists as follows before synchronizing products from BRM:
 - Create the default price list:
 - a. In Siebel CRM, define a default price list header. See *Siebel Pricing Administration Guide* for information about creating price list headers.
 - b. Update the `AIAConfigurationProperties.xml` file with the Siebel CRM row ID of the default price list. See the `Siebel.PriceList.ID` property in [Table 24–5](#).
 - c. Using SOA Composer, enter the Siebel CRM row ID for the default price list in the `PRICELIST DVM`. See "[Updating the PRICELIST DVM](#)" for more information.
 - (Optional) Create additional Siebel CRM price lists:
 - a. In Siebel CRM, define additional price list headers.

Keep a list of the names and currencies used for these price lists so that you can enter them correctly when defining rate plans in BRM at design time. See "[Working with Price Lists and Rate Plans at Design Time](#)" for more information.
 - b. Using SOA Composer, enter the Siebel CRM row IDs for the additional price lists in the `PRICELIST DVM`. See "[Updating the PRICELIST DVM](#)" for more information.

Note: Integration of multiple price lists is only supported with BRM 7.5 and later. If you are using an earlier version of BRM, a single default price list is supported, but you must change the version number as described in "[Setting the BRM Version Number for Backward Price List Compatibility](#)".

4. Set up a Siebel CRM organization and identify the organization in Siebel CRM. Update the `AIAConfigurationProperties.xml` file. See the `Siebel.BusinessUnit` property listed in [Table 24–5](#).
5. Set up a Siebel CRM workspace and identify the workspace in Siebel CRM. Update the `AIAConfigurationProperties.xml` file. See the `Siebel.Product.WorkspaceName` property listed in [Table 24–5](#).

6. Set up friends and family products. See ["Supporting Friends and Family"](#) for more information.
7. Make the following workflow changes after synchronizing the products from BRM to use penalty products synchronized from BRM:
 - Modify ISS Promotion Disconnect Process to use the product synchronized from BRM. See *Siebel Order Management Guide Addendum for Communications* for more information about ISS Promotion Disconnect Process.
 - Include one-time charge products in the Siebel CRM Catalog so that you can see the recommended pick list for one-time charges.
 - After one-time charge products have been added to a Siebel CRM Catalog, associate one-time charges with Modify, Add, Change, Delete (MACD) order types. See *Siebel Order Management Guide Addendum for Communications* for more information about setting up one-time service charges for products in Siebel CRM.
8. Define simple Special Rating products and set their composition type to **Partial**.
9. Set up service bundles and set the Billing Type to **Service Bundle** and the Billing Service Type to the same string as the billing service bundle on the component products that have been synchronized from BRM.
10. Set up promotions, bundling service bundles, account level products, and discounts.
11. Add service bundles and promotions to the price list used by the product synchronization integration flow.

Working with DVMs for Product Lifecycle Management

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 PLM.

Table 24–1 Product Lifecycle Management DVMs

DVM	Description
PRICECHARGETYPE	Price Charge Type (common values are <i>One-Time</i> or <i>Recurring</i> .)
PRICECHARGETYPEUOM	Price Charge Type Unit Of Measure (common values are <i>Per Day</i> or <i>Per Month</i> .)
PRICETYPE_EVENT	Price Type Event (common values are <i>Purchase</i> or <i>Cancel</i> .)
PRODUCTTYPECODE	Product Type Code (common values are <i>Item</i> or <i>Subscription</i> .)
ITEM_BILLINGTYPECODE	Maps Billing Type from BRM to Siebel CRM

Table 24–1 (Cont.) Product Lifecycle Management DVMs

DVM	Description
RESOURCE	Non-Monetary resources (<i>Free Minutes, Text Messages, and so on</i>).
CURRENCY_CODE	Currency codes.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about DVMs.

Working with Cross-References for Product Lifecycle Management

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.

See "[Cross-References for the Process Integration for Product Lifecycle Management](#)" for more information about product management cross-references.

Table 24–2 Product Lifecycle Management Cross-References

Cross-reference Table Name	Column Name and Value COMMON	Column Name and Value SEBL_01	Column Name and Value BRM_01	Description
ITEM_ITEMID	auto generated GUID	ProductID of Siebel CRM Product ABM	POID of BRM Product ABM	Cross references the BRM ProductID and the Siebel CRM ProductID.
PRICELINE_ID	auto generated GUID	Siebel CRM PriceListItemID for the main product	POID of BRM Product ABM	Cross references the BRM Product ID to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ ITEMID.
PRICELINETYPE_ID	auto generated GUID	Siebel CRM PriceListItemID for the event product	POID of BRM Product ABM + Event Name	Cross references BRM Product's Events to Siebel CRM PriceLineID. Also links to the COMMON of ITEM_ITEMID.
SIEBELPRODUCTEVE NTXREF	ITEM_ID_ COMMON From ITEM_ ID.COMMON	LINEPRICETYPE CODE PRICELINETYPE _ID.COMMON	--	Cross references BRM Product's Event that is associated with the main product in Siebel CRM.

Handling Error Notifications for Product Lifecycle Management

Based on the roles defined for the services, email notifications are sent if a service ends due to an error. No AIA-specific errors are caused by the process integration for product management services.

For more information about the errors caused by BRM or Siebel CRM, see the documentation for BRM and Siebel CRM.

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**.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about setting up error notifications.

Configuring Properties for Product Lifecycle Management

Configure the properties described in this section in the *AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config/AIAConfigurationProperties.xml* file.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with **AIAConfigurationProperties.xml**.

[Table 24–3](#) shows the properties for the `SyncProductBRMCommsReqABCImpl` service.

Note: Entries in the **AIAConfigurationProperties.xml** file are case sensitive

Table 24–3 *SyncProductBRMCommsReqABCImpl Properties*

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.

Table 24–3 (Cont.) SyncProductBRMCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PreXFormABMtoPriceListListEBM	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.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/SyncProductBRMCommsReqABCImpl/SyncProductBRMCommsReqABCImpl	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.
EBSOverride.CommunicationsItemCompositionList	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.CommunicationsItemCompositionList.Address	http://<http.host name>:<http.port>/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl	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.CommunicationsItemCompositionList.PortType	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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.CommunicationsItemCompositionList.ServiceName	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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.

Table 24–3 (Cont.) SyncProductBRMCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.Address	http://\$<http.host name>:\$<http.port>/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	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 properties for the SyncDiscountBRMCommsReqABCImpl service.

Table 24–4 SyncDiscountBRMCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Siebel CRM 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.PreXFormABMtoEBM	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.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.

Table 24–4 (Cont.) SyncDiscountBRMCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.Address	http:// \$<http.host name>:\$<http.port>/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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 properties for the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service.

Table 24–5 ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel CRM 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?SWExtSource=SecureWebService&SWExtCmd=Execute&WSSOAP=1	Siebel CRM 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?SWExtSource=SecureWebService&SWExtCmd=Execute&WSSOAP=1	Siebel CRM 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.

Table 24–5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCsImpl Properties

Property Name	Value/Default Values	Description
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 CRM system. The value for this property should be the ID of the business unit in the Siebel CRM 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 CRM. 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.	The products synchronized from BRM with a single rate plan with no associated price list are assigned to this price list in the Siebel CRM system. The value for this property should be the row ID of the default price list in the Siebel CRM system. This value must be set before synchronizing products.
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 CRM is set to 0 (zero). This value must be set before the product sync is run.
ABCSExtension.PreXFormEBMtoABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormEBMtoABM	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.PreXFormPriceListEBMtoItemCompositionEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to EBM transformation.)
ABCSExtension.PreInvokeItemCompositionEBS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).

Table 24–5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PreXFormPriceListEBMtoProductABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PreInvokeProductABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service. (PostInvoke Application).
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/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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.CommunicationsPriceListResponseEBS.SyncPriceListListResponse	true/false. Default = true	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS.
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.Address	http://\$<http.host name>:\$<http.port>/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS.
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.PortType	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS.
EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.ServiceName	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	This property is not used in AIA Communications PLM flow. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS.

Table 24–6 shows the properties for the SyncItemCompositionListSiebelCommsProvABCImpl service.

Table 24–6 SyncItemCompositionListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel CRM 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 CRM 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 CRM system. The value for this property should be the ID of the business unit in the Siebel CRM 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 CRM. 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.
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.

Table 24–6 (Cont.) SyncItemCompositionListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.Address	http:// \$<http.host name>:\$<http.port>/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl	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.

Configuring Properties for the Query Product Classes Business Flow

Table 24–7 shows the properties for the QueryProductClassAndAttributesSCECommsReqABCImpl service.

Table 24–7 QueryProductClassAndAttributesSCECommsReqABCImpl Properties

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).

Table 24–7 (Cont.) QueryProductClassAndAttributesSCECommsReqABCImpl Properties

Property Name	Value/Default Values	Description
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).
EBSOverride.CommunicationsClassificationEBSV1.QueryClassificationList.Address	http://<http.host name>:<http.port>/soa-infra/services/default/QueryClassificationListSiebelCommsProvABCImpl/QueryClassificationListSiebelCommsProvABCImpl	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.CommunicationsClassificationEBSV1.QueryClassificationList.PortType	QueryClassificationListSiebelCommsProvABCImplService	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.CommunicationsClassificationEBSV1.QueryClassificationList.ServiceName	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QueryClassificationListSiebelCommsProvABCImpl/V1}QueryClassificationListSiebelCommsProvABCImpl	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.CommunicationsSpecificationEBSV1.QuerySpecificationList.Address	http://<http.host name>:<http.port>/soa-infra/services/default/QuerySpecificationListSiebelCommsProvABCImpl/QuerySpecificationListSiebelCommsProvABCImpl	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.CommunicationsSpecificationEBSV1.QuerySpecificationList.PortType	QuerySpecificationListSiebelCommsProvABCImplService	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.

Table 24–7 (Cont.) QueryProductClassAndAttributesSCECommsReqABCImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsSpecificationEBSV1.QuerySpecificationList.ServiceName	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QuerySpecificationListSiebelCommsProvABCImpl/V1}QuerySpecificationListSiebelCommsProvABCImpl	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.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.Address	http://<http.host name>:<http.port>/soa-infra/services/default/QuerySpecificationValueSetListSiebelCommsProvABCImpl/QuerySpecificationValueSetListSiebelCommsProvABCImpl	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.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.PortType	QuerySpecificationValueSetListSiebelCommsProvABCImplService	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.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.ServiceName	{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QuerySpecificationValueSetListSiebelCommsProvABCImpl/V1}QuerySpecificationValueSetListSiebelCommsProvABCImpl	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 properties for the QueryClassificationListSiebelCommsProvABCImpl service.

Table 24–8 QueryClassificationListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel CRM 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_enue/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Siebel CRM 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.)

Table 24–8 (Cont.) QueryClassificationListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PostXFormABMt oEBMClassificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABSLis tOfSwiAdminIssClassDefinitionA BM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeABSLi stOfSwiAdminIssClassDefinition ABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Table 24–9 shows the properties for the QuerySpecificationListSiebelCommsProvABCImpl service.

Table 24–9 QuerySpecificationListSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel CRM system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.QueryProductClassAttri butes.RouteToCAVS	true/false. Default = false	Controls whether QueryProductClassAttributes routes messages to the verification system or to the Provider ABCS implementation.
Routing.QueryProductClassAttri butes.SEBL_01.EndpointURI	http://\$<http.host name>:\$<http.port>/eai_ enu/start.swe?SWEEExtSourc e=SecureWebService&S WEEExtCmd=Execute& WSSOAP=1	Siebel CRM 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.PreXFormEBMt oABMSpecificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormABM toEBMSpecificationListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABSA ttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeABS AttributeQueryByExample_ InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Table 24–10 shows the properties for the QuerySpecificationValueSetListSiebelCommsProvABCImpl service.

Table 24–10 QuerySpecificationValueSetListSiebelCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel CRM 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 CRM 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.PreXFormEBMtoABMSpecificationValueSetListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXFormABMtoEBMSpecificationValueSetListEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation.).
ABCSExtension.PreInvokeABSAttributeQueryByExample_InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application)
ABCSExtension.PostInvokeABSAttributeQueryByExample_InputABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).

Configuring the Process Integration for Order Lifecycle Management

This chapter discusses how to configure the process integration for order lifecycle management (OLM) for the Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) Pre-Built Integration option (the integration). Configuration includes:

- Configuring BRM and Siebel CRM to integrate with Oracle Application Integration Architecture (Oracle AIA)
- Working with domain value maps (DVMs) and cross-references
- Handling error notifications
- Setting Oracle AIA configuration properties

Configuring BRM for Integrated Order Lifecycle Management

To configure BRM for integrated OLM, verify that you have deployed and configured the BRM JCA adapter as described in *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*.

See *Oracle Communications Billing and Revenue Management JCA Resource Adapter* for more information about the BRM JCA adapter.

Configuring Siebel CRM for Integrated Order Lifecycle Management

To configure Siebel CRM to integrate with AIA for communications:

1. Install ACR 474. See *Siebel Maintenance Release Guide*.
2. Set the UTCCanonical process property to **Y** for certain Siebel CRM interfaces. The instructions for ACR 474 and ACR 508 in *Siebel Maintenance Release Guide* explain which Siebel CRM interfaces you must set the UTCCanonical process property for.
3. Perform the following Oracle Advanced Queuing (AQ) configurations:
 - For the order flow, configure the SISOMBillingSubmitOrderWebService Siebel CRM outbound workflow to enqueue the Siebel CRM messages in the AIA_SALESORDERJMSQUEUE queue.

For this service, in Siebel, you must set the process property UTCCanonical to **Y**.

- For updating the order information from your central fulfillment system to Siebel CRM, enable the SWIOrderUpsert Siebel CRM inbound Web service.
For this service, in Siebel CRM, you must set the process property UTCCanonical to **Y**.
- For the Special Rating List Sync Flow, configure the SWISpecialRatingList Siebel CRM outbound workflow to enqueue the Siebel CRM messages in the AIA_SPECIALRATINGJMSQ queue.

See *Siebel Order Management Guide Addendum for Communications*, for more information about Web services.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about the corresponding AIA-side configuration.

Working with Domain Value Maps for Order Lifecycle Management

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.

Note: 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 OLM.

Table 25–1 Order Lifecycle Management DVMs

DVM	Description
SALESORDER_DYNAMICPRICEIND	Dynamic Pricing Indicator
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	Order Priority
SALESORDER_PROCESSINGTYPECODE	Processing Type Code
ITEM_TYPE	Product/Item Type Code
SALESORDER_REVISIONPERMISSIBLECODE	Revision Permissible Code
SALESORDER_SERVICEINDICATOR.	Service Indicator
SALESORDER_STRTBILLSERVICEUSAGE	Start Billing Service Usage
SALESORDER_STATUS	Status

Table 25–1 (Cont.) Order Lifecycle Management DVMs

DVM	Description
SALESORDER_TYPECODE	Type Code
STATE	State
PROVINCE	Province
ADDRESS_COUNTRYID	Country Code
CUSTOMERPARTY_TYPECODE	Account Type Code
ITEM_BILLINGTYPECODE	Examples of values are <i>Subscription, Discount, Item, Special Rating</i> , and so on. Billing Type Code.
SALESORDER_CHANGEDIND	Order Changed Indicator. Values are <i>True</i> or <i>False</i> . Used to validate the OrderChangedIndicator attribute. For example, The order management system can set this attribute to <i>True</i> 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_REVISIONPERMISSIBLECODE	Revision Permissible Code
SALESORDER_LINESTATUS	Order Line Status
DISCOUNT_METHODCODE	Discount Method Code
CURRENCY_CODE	Currency Code
PRICE_TYPE	Price Type
PRICELIST	Maps price list Row IDs from Siebel CRM to logical common price list names for BRM and Oracle Product Hub. Used to communicate price lists from order lines for products synchronized from BRM using a rate plan selector. See " Working with the PRICELIST DVM " for details.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with DVMs.

Working with the PRICELIST DVM

You must update the PRICELIST DVM before submitting any orders so that the integration can map Siebel CRM price lists to BRM rate plans. You can migrate the PRICELIST DVM information between environments.

Updating the PRICELIST DVM

To update the PRICELIST DVM:

1. Log in to Oracle SOA Composer and open **PRICELIST.dvm**.

For information about using Oracle SOA Composer with DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

2. Replace the placeholder domain values for the first row in **PRICELIST.dvm** as follows:
 - For SEBL_01, enter the Siebel CRM price list row ID for one of the price lists you created in Siebel CRM before synchronizing the products to BRM. You can find price list row IDs using a query in Siebel CRM.

- For COMMON and BRM_01, enter the logical name of the price list that corresponds to the Siebel CRM price list.
3. Add and fill in new rows for all additional price lists that you created in Siebel CRM with the exception of the default price list that you entered into the **AIAConfigurationProperties.xml** file in "[Configuring Siebel CRM for Integrated Product Lifecycle Management](#)".
4. Save and commit the changes.

Migrating PRICELIST DVM Between Environments

To migrate the PRICELIST DVM between environments:

1. In the source environment, in Siebel CRM, open the **S_PR_LST** table.
2. Search in the **ROW_ID** column for the price list row IDs listed in the **SEBL_01** column in **PRICELIST.dvm**.
3. Copy the values in the **NAME**, **SUBTYPE_CD**, and **BU_ID** columns for each row ID to a text editor and close the table.
4. In the target environment, in Siebel CRM, open the **S_PR_LST** table.
5. Search the **ROW_ID** column for the price list row IDs listed in the source environment's **S_PR_LST** table and **PRICELIST.dvm**.
6. Enter the values copied to the text editor in the **NAME**, **SUBTYPE_CD**, and **BU_ID** columns.
7. Using Oracle SOA Composer, open **PRICELIST.dvm**.
8. In the **SEBL_01** column, add the row IDs copied to the text editor. Fill out the **COMMON** and **BRM_01** columns as described in "[Updating the PRICELIST DVM](#)".
9. Save and commit the changes in both files.

Working with Cross-References for Order Lifecycle Management

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 OLM cross-references.

Table 25–2 Order Lifecycle Management Cross-References

Cross-Reference Table Name	COMMON Value	SEBL_01 Value	BRM_01 Value	Description
SALESORDER_ID	SalesOrderIdentification	Id	--	The Siebel CRM Sales Order ID is cross-referenced.
SALESORDER_LINEID	SalesOrderLine/Identification	OrderItem/OrderId	--	The OrderItem/OrderId from Siebel CRM is mapped to the SalesOrderLine Identification in the EBM
INSTALLEDPRODUCT_ID	InstalledProductIdentification	AssetIntegrationId	PRODUCT/SERVICE/DISCOUNT OBJ	The Siebel CRM Asset Integration ID is mapped to the Product/Service/Discount OBJ of the BRM object.
ITEM_ITEMID	ItemIdentification	ProductId	PRODUCT/DISCOUNT POID	The Siebel CRM Product ID is mapped to the PRODUCT/DISCOUNT POID of the BRM object.
CUSTOMERPARTY_ACCOUNTID	CustomerPartyAccountIdentification	AccountId	Account POID	The Siebel CRM Customer ID is mapped to the BRM Account POID
CUSTOMERPARTY_CONTACTID	CustomerPartyAccountContactIdentification	ContactId	Contact POID	The Siebel CRM Contact ID is mapped to the BRM Contact POID
CUSTOMERPARTY_DEFAULTBALANCEGROUPID (Populated only when service-level balance groups are enabled)	InstalledProductIdentification (Populated from INSTALLEDPRODUCT_ID table)	AssetIntegrationID	Balance group POID	<p>The balance group POID for each service bundle or simple service bundle is mapped to the InstalledProductIdentification for the product and to the Siebel CRM AssetIntegrationID sent on the order line.</p> <p>For nested service bundles, the integration populates only the values for the parent service bundle.</p> <p>To track the parent of a nested service bundle, the integration populates the BalanceBundleIdentification field in the order EBM with the common InstalledProductIdentification value of the parent service bundle.</p>

Table 25–2 (Cont.) Order Lifecycle Management Cross-References

Cross-Reference Table Name	COMMON Value	SEBL_01 Value	BRM_01 Value	Description
CUSTOMERPARTY_DEFAULTBALANCEGROUPID (Entry for default account-level balance group)	CustomerPartyAccountContactIdentification	ServiceAccountId/BillingAccountId	Balance group POID	The default balance group POID is mapped to the common ID of the account.
CUSTOMERPARTY_PAYPROFILEID	PaymentProfileIdentification	BillingProfileId	Pay info POID	The billing profile ID from Siebel CRM is mapped to the pay info POID from BRM.
CUSTOMERPARTY_BILLPROFILEID	BillingProfileIdentification	BillingProfileId	Bill Info POID	The billing profile ID from Siebel CRM is mapped to the bill info POID from BRM.
CUSTOMERPARTY_ADDRESSID	CustomerPartyAccountAddressIdentification	AddressId	Address POID	The Siebel CRM Address ID is mapped to the BRM Contact POID.

Handling Error Notifications for Order Lifecycle Management

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.

See "[Understanding the Process Integration for Order Fallout Management](#)" for more information about order fallout.

[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 for Order Lifecycle Management

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 <i>Subscription</i> or <i>Discount</i> .

Table 25–3 (Cont.) Error Messages for Order Lifecycle Management

Error Code	Error Text	Description
AIA_ERR_AIACOMOMPI_0002	Date Validation Failed: Purchase Date should be set to the future.	In Billing Initiation mode, the ProcessFulfillmentOrderBillingBRMComm sAddSubProcess ends in an error when the purchase date is not set to the future for lines with products of type <i>Item</i> .
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 sProvABCImpl ends in an error if a change order is processed for data that was created using 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 sProvABCImpl 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 BRM, see the Siebel CRM and BRM documentation.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about AIA error handling.

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*.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about AIA error handling.

Configuring the Process Integration for Order Lifecycle Management

This section provides instructions for setting the AIA configuration properties and setting the BRM version number for backward compatibility.

Setting AIA Configuration Properties

Configure the properties described in this section in the *AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config/AIAConfigurationProperties.xml* file.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with **AIAConfigurationProperties.xml**.

Note: Entries in the **AIAConfigurationProperties.xml** file are case sensitive

Table 25–4 shows the properties for the UpdateSalesOrderSiebelCommsProvABCServiceImpl service.

Table 25–4 UpdateSalesOrderSiebelCommsProvABCServiceImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	URL for Siebel CRM Instance web service for "Order spcLine spcItem spcUpdate spc- _spcComplex" web service.
Routing.SWI_spcOrder_spcUpsert.RouteToCAVS	true/false. Default = false.	Controls whether UpdateSalesOrderSiebelCommsProvABCServiceImpl routes messages to the CAVS or to the Siebel CRM system.
Routing.SWI_spcOrder_spcUpsert.SEBL_01.EndpointURI	Target Endpoint URL for the Siebel CRM upsert web service. example: http://\${siebel.http.host}:\${siebel.http.port}/eai_enu/start.swe?SWEEExtSource=SecureWebService&SWEEExtCmd=Execute&WSSOAP=1Status OpenFixedClosed	Target Endpoint URL for the Siebel CRM 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 properties for the ProcessSalesOrderFulfillmentSiebelCommsReqABCServiceImpl service.

Table 25–5 ProcessSalesOrderFulfillmentSiebelCommsReqABCServiceImpl Properties

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 CRM 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.

Table 25–5 (Cont.) ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PostInvokeEBS	true/false. Default = false	Indicates whether there is any extension in the ABCS after invoking application business service.
EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address	Address of web service that must be invoked. Example (Default): http://<soa_server>:<soa port>/soa-infra/services/default/ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer/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.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.PortType	PortType of the web service that must be invoked. Example (Default): ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer	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.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.ServiceName	ServiceName of the web service that must be invoked. Example (Default): http://xmlns.oracle.com/ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer}ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer	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 properties for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess service.

Table 25–6 ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Properties

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.FutureTimeThresholdForBillingDates	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. See " Using Single-Phase Billing or Two-Phase Billing " for more information on how this property is used.
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.PostprocessAddPCM_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.

Table 25–6 (Cont.) ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Properties

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 properties for the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess service.

Table 25–7 ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess Properties

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.
ABCSExtension.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.
ABCSExtension.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 Properties

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 properties for the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess service.

Table 25–8 ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess Properties

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 properties for the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service.

Table 25–9 ProcessFulfillmentOrderBillingBRMCommsProvABCImpl Properties

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 BRM Adapter. Example: eis/BRM	End point for 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 Properties

Property Name	Value/Default Values	Description
Routing.BRMSUBSCRIPTIONService.BRM_02.EndpointURI	End point for BRM Adapter. Example: eis/BRM2	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMSUBSCRIPTIONService.BRM_03.EndpointURI	End point for BRM Adapter. Example: eis/BRM3	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
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 BRM Adapter. Example: eis/BRM	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMCUSTService.BRM_02.EndpointURI	End point for BRM Adapter. Example: eis/BRM2	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMCUSTService.BRM_03.EndpointURI	End point for BRM Adapter. Example: eis/BRM3	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBALService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMBALService_ptt.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for 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 BRM Adapter. Example: eis/BRM2	End point for 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 BRM Adapter. Example: eis/BRM3	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMARService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMARService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "true"
Routing.BRMBASEService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMBASEService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for 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 Properties

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/ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer/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): ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer	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/ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer}ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer	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.

ABCSExtension Table 25–10 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess service.

Table 25–10 ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess Properties

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 Properties

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 properties for the ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess service.

Table 25–11 ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess Properties

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 Properties

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 properties for the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl service.

Table 25–12 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl Properties

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 CRM 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.) ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.PortType	PortType of the web service that must be invoked. Example (Default): ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImplService	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/ABCServiceImpl/BRM/Industry/Comms/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl/V1}ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImplService	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/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl	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 properties for the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl service.

Table 25–13 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl Properties

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 BRM adapter. For example, eis/BRM	End point for BRM adapter. To Invoke CAVS update the above property to "true" and this property with corresponding CAVS URL

Setting the BRM Version Number for Backward Price List Compatibility

When sending order data to BRM, Oracle AIA checks the BRM version number in the Oracle AIA system configuration property. By default, the installation process for the integration sets the BRM version number to 7.5. Oracle AIA sends order line price list

information to BRM versions 7.5 and later. For earlier versions of BRM, Oracle AIA does not send the order line price list information.

To support the behavior for earlier versions of BRM, you must change the version number in the AIA system configuration property.

To change the BRM version number:

1. Browse to your AIA Home Page. For example:

`http://sc0000.us.oracle.com:8001/AIA`

2. In the Setup area, click **Go**.
3. Select the **Systems** tab.
4. In the Version column of the BRM row, enter your BRM version number.
5. Save your changes.

Configuring the Process Integration for Customer Management

This chapter discusses how to configure the process integration for customer management (CM) for the Oracle Communications Order to Cash for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) Pre-Built Integration option (the integration). Configuration includes:

- Configuring BRM and Siebel CRM to integrate with Oracle Application Integration Architecture (Oracle AIA)
- Working with domain value maps (DVMs) and cross-references
- Handling error notifications
- Setting Oracle AIA configuration properties

Configuring BRM for Integrated Customer Management

To configure BRM for CM:

1. Add the following phone number validation format to BRM using the Field Validation Editor:

```
###-###-####
```

This format allows nonformatted phone numbers coming from Siebel CRM in BRM.

See *Oracle Communications Billing and Revenue Management Concepts* and *Oracle Communications Billing and Revenue Management Managing Customers* for more information about validating phone number formats for integration.

2. Verify that you have configured and deployed the BRM JCA adapter as described in Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations.

See *Oracle Communications Billing and Revenue Management JCA Resource Adapter Guide* for more information about deploying and configuring the JCA Resource Adapter.

Configuring Siebel CRM for Integrated Customer Management

To configure Siebel CRM for CM:

1. Install ACR 474. See *Siebel Maintenance Release Guide* for information on how to install ACR 474.
2. Set the UTCCanonical process property to **Y** for certain Siebel CRM interfaces. The instructions for ACR 474 and ACR 508 in *Siebel Maintenance Release Guide* explain which Siebel CRM interfaces you must set the UTCCanonical process property for.

Working with Domain Value Maps for Customer Management

DVMs are a standard feature of the Oracle 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 Customer Management DVMs

DVM	Columns	Description
CUSTOMERPARTY_ACCOUNTYPECODE.dvm	SEBL_01,COMMON,BRM_01	Used to get the type of the account, such as <i>Business</i> or <i>Customer</i> .
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 BRM, Salutation is not a language-independent code. If BRM requires salutations in a language other than English, then you must update the DVM with the appropriate BRM values.
CURRENCY_CODE.dvm	SEBL_01,COMMON,BRM_01	Currency codes.
CUSTOMERPARTY_BILLPROFILE_BILLTYPECODE.dvm	SEBL_01,COMMON,BRM_01	Bill type (<i>summary</i> and <i>detailed</i>).
CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE.dvm	SEBL_01,COMMON,BRM_01	Billing frequency (<i>monthly</i> , <i>yearly</i> , <i>quarterly</i> , and so on.)
CUSTOMERPARTY_PAYPROFILE_BANKACCOUNTTYPE.dvm	SEBL_01,COMMON,BRM_01	Bank account type (<i>checking</i> , <i>savings</i> , and so on).
CUSTOMERPARTY_PAYPROFILE_CREDIT_CARDTYPE.dvm	SEBL_01,COMMON	Credit Card type (<i>Visa</i> , <i>Mastercard</i> , and so on.)
CUSTOMERPARTY_PAYPROFILE_DELIVERY_PREF.dvm	COMMON,BRM_01	Bill media/delivery preference (<i>Email</i> or <i>Mail</i>).
CUSTOMERPARTY_PAYPROFILE_PAYMENTMETHODCODE.dvm	SEBL_01,COMMON,BRM_01	Payment profile payment method types (<i>credit card</i> , <i>direct debit</i> , and <i>invoice/bill me</i> .)

Table 26–1 (Cont.) Customer Management DVMs

DVM	Columns	Description
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 (<i>home, work, mobile, fax, and so on.</i>)

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with DVMs.

Working with Cross-References for Customer Management

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 Customer Management Cross-References

Cross-Reference Table Name	Column Name and Value	Column Name and Value	Column Name and Value	Description
	COMMON	SEBL_01	BRM_01	
CUSTOMERPARTY_ ACCOUNTID.xref	Account ID	Account ID	Account POID	Siebel Account ID is mapped one-to-one to the BRM Account ID.
CUSTOMERPARTY_ BILLPROFILEID.xref	Bill Profile ID	Bill Profile ID	bill-info POID	Siebel Bill Profile ID is mapped one-to-one to the BRM bill-info ID.
CUSTOMERPARTY_ PAYPROFILEID.xref	Payment Profile ID	Bill Profile ID	Pay-info POID	Siebel Bill Profile ID is mapped one-to-one to the BRM pay-info ID.
CUSTOMERPARTY_ ADDRESSID.xref	Address ID	Address ID	Account POID pay-info POID*	BRM Account ID is cross-referenced here if the address is used as the billing address (name-info[1]) on that account. 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.

Table 26–2 (Cont.) Customer Management Cross-References

Cross-Reference Table Name	Column Name and Value COMMON	Column Name and Value SEBL_01	Column Name and Value BRM_01	Description
CUSTOMERPARTY_CONTACTID.xref	Contact ID	Contact ID	Account POID pay-info POID*	BRM Account ID is cross-referenced if the contact is used as the name (name-info[1]) on that account. BRM pay-info ID is cross-referenced 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 the type of the ID.
CUSTOMERPARTY_DEFAULTBALANCEGROUPID.xref	Account ID*	--	Balance Group POID	This cross-reference maps the default balance group to the common account ID. This is populated after account creation in the CreateCustomerPartyProviderABCSimpl service, and is referenced by the order flow during service creation.
CUSTOMERPARTY_PARTYID.xref	--	SEBL_01,COMMON,EBIZ_01,UCM_01,SAP_01	--	Customer Party IDs
CUSTOMERPARTY_PARTYLOCATIONID.xref	--	SEBL_01,COMMON,EBIZ_01,UCM_01,SAP_01	--	Customer Party Location IDs
CUSTOMERPARTY_CONTACTID.xref	--	SEBL_01,COMMON,EBIZ_01,UCM_01, BRM_01, SAP_01	--	Customer Party contact IDs. BRM account ID is cross-referenced here if the contact is used as the name (name-info[1]) on that account. 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,EBIZ_01,UCM_01	--	Customer Party Location Reference IDs.
CUSTOMERPARTY_ACCOUNT_PHONECOMMID.xref	--	SEBL_01,COMMON,EBIZ_01,UCM_01,SAP_01	--	Customer Party Account's Phone contact points.

Table 26–2 (Cont.) Customer Management Cross-References

Cross-Reference Table Name	Column Name and Value COMMON	Column Name and Value SEBL_01	Column Name and Value BRM_01	Description
CUSTOMERPARTY_ACCOUNT_FAXCOMMID.xref	--	SEBL_01,COMMON,EB IZ_01,UCM_01, SAP_01	--	Customer Party Account's Fax contact points.
CUSTOMERPARTY_ACCOUNT_WEBCOMMID.xref	--	SEBL_01,COMMON,EB IZ_01,UCM_01	--	Customer Party Account's Email/Web contact points.
CUSTOMERPARTY_CONTACT_PHONECOMMID.xref	--	SEBL_01,COMMON,EB IZ_01,UCM_01	--	Customer Party Contact's Phone contact points.
CUSTOMERPARTY_CONTACT_FAXCOMMID.xref	--	SEBL_01,COMMON,EB IZ_01,UCM_01	--	Customer Party Contact's Fax contact points.
CUSTOMERPARTY_CONTACT_EMAILCOMMID.xref	--	SEBL_01,COMMON,EB IZ_01,UCM_01	--	Customer Party Contact's Email/Web contact points.

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
SyncCustomerPartyListBRMCommsProvABCSEImpl	AIA_ERR_AIACOMCMPI_0004	Subordinate account cannot have multiple parent accounts.
SyncCustomerPartyListBRMCommsProvABCSEImpl	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.
SyncCustomerPartyListBRMCommsProvABCSEImpl	AIA_ERR_AIACOMCMPI_0006	None of the existing subordinate bill profiles are included in the move account request.
CommsProcessBillingAccountListEBF	AIA_ERR_AIACOMCMPI_0001	EBMHeader/Sender/ID is required.
CommsProcessBillingAccountListEBF	AIA_ERR_AIACOMCMPI_0002	EBMHeader/Target/ID is required.
CommsProcessBillingAccountListEBF	AIA_ERR_AIACOMCMPI_0003	Account sequence error: Pay-From accounts and billing profiles must appear before dependent and subordinate accounts and billing profiles.

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*.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about setting up error notifications using these values.

Order Fallout Management

When an order is submitted from Siebel CRM, the order may fail while customer-related information is being interfaced to 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.

See "[Understanding the Process Integration for Order Fallout Management](#)" for more information about order fallout.

Configuring the Process Integration for Customer Management

Configure the properties described in this section in the *AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config/AIAConfigurationProperties.xml* file.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with *AIAConfigurationProperties.xml*.

Note: Entries in the *AIAConfigurationProperties.xml* file are case sensitive

[Table 26–4](#) shows the properties for the *SyncCustomerPartyListBRMCommsProvABCServiceImpl* service.

Table 26–4 SyncCustomerPartyListBRMCommsProvABCSEmpl Properties

Property Name	Value/Default Values	Description
EnableAccountStatusSync	true/false. Default = false	This property when set to <i>True</i> , updates the status (active/inactive) of the account from Siebel CRM to BRM.
ABCSExtension.preformEBMtoABM	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 <i>True</i> , 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 <i>True</i> , 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.PostInvokePCM_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 <i>True</i> , 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 <i>True</i> , 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.PostInvokePCM_OP_SEARCHABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to <i>True</i> , 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.PreInvokeABSPCM_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 <i>True</i> , then the Extension process (defined in AIA ABCS Extension guidelines) is invoked. The name of the property indicates which extension point is enabled.
ABCSExtension.PostInvokeABSPCM_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 <i>True</i> , 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 Properties

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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 Properties

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 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 BRM Java EE Connector Architecture (JCA) adapter for the first instance of the BRM in case of multiple BRM instances. See "Configuring Multiple BRM Instances for Communications Integrations" for more information about multiple BRM systems.
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 BRM in case of multiple BRM instances. See "Configuring Multiple BRM Instances for Communications Integrations" for more information about multiple BRM systems.
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 BRM JCA adapter for the first instance of the BRM in case of multiple BRM instances for the CUSTCare opcode of BRM. See "Configuring Multiple BRM Instances for Communications Integrations" for more information about multiple BRM systems.
Routing.BRMCUSTCAREService.RouteToCAVS	true/false. Default = false	This property specifies whether to route to CAVS Service.

Table 26–4 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Routing.BRMBILLService.BRM_01.EndpointURI	eis/BRM	This property specifies the Connection factory to connect to the BRM JCA adapter for the first instance of the BRM in case of multiple BRM instances for the BillService opcode. See " Configuring Multiple BRM Instances for Communications Integrations " for more information about multiple BRM systems.
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 BRM JCA adapter for the first instance of the BRM in case of multiple BRM instances for the BRMBASEService. See " Configuring Multiple BRM Instances for Communications Integrations " for more information about multiple BRM systems.
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 BRM JCA adapter for the first instance of the BRM in case of multiple BRM instances for the TXNService opcode. See " Configuring Multiple BRM Instances for Communications Integrations " for more information about multiple BRM systems.
Routing.BRMTXNService.RouteToCAVS	true/false. Default = false	This property specifies whether to route to CAVS Service.

Table 26–5 shows the properties for the SyncAccountSiebelReqABCSImpl service.

Table 26–5 SyncAccountSiebelReqABCImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PreXformABMt oEBMABM	true/false. Default = false	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to <i>True</i> , 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.PreInvokeEBSE BM	true/false. Default = false.	This property governs whether the ABCS Extension is enabled at the predefined plug-into point. If set to <i>True</i> , 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. server.InternetProtocol}\${participa tingapplications.siebel.server.host} \${participatingapplications.siebel. server.port}/eai_ \${participatingapplications.siebel. server.Language}/start.swe?SWE ExtSource=SecureWebService&am p;SWEExtCmd=Execute&WS SOAP=1	Endpoint URI of the SEBL_01 Siebel instance.
Routing.SWICustomerParty.Rou teToCAVS	true/false. Default = false.	This property specifies whether the end point should route to CAVS.
Routing.SWICustomerParty.CA VS.EndpointURI	http://\${fp.server.soaserverhostn ame}:\${fp.server.soaserverport}/A IValidationSystemServlet/syncre sponsesimulator	This property specifies the end point URL for the CAVS service.
Routing.CustomerPartyEBSV2.S yncCustomerPartyList.Message ProcessingInstruction.Environment Code	PRODUCTION	--.
RoutingCustomerPartyEBSV2.Sy ncCustomerPartyList.CAVS.End pointURI	http://\${fp.server.soaserverhostn ame}:\${fp.server.soaserverport}/A IValidationSystemServlet/asyncre questrecipient	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.TransformAppContextS iebelService.RouteToCAVS	true/false. Default = false.	This property specifies whether the end point should route to CAVS.
Routing.TransformAppContextS iebelService.CAVS.EndpointURI	http://\${fp.server.soaserverhostn ame}:\${fp.server.soaserverport}/A IValidationSystemServlet/asyncre questrecipient	This property specifies whether the end point should route to CAVS.

Table 26–5 (Cont.) SyncAccountSiebelReqABCImpl Properties

Property Name	Value/Default Values	Description
Account.ProcessUpdateEventsOnly	true/false. Default = true.	<p>Customers must set this property to <i>True</i>. This is required to optimize the flow. By setting this property to <i>True</i>, 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.</p> <p>Setting this property to <i>False</i>, results in a less optimized flow, but OOTB behavior where customer creation occurs as part of the order flow remains the same.</p> <p>See <i>Oracle Application Integration Architecture Pre-Built Integrations Functional Interoperability Configuration Guide</i> for more information.</p>
Contact.QueryAllEntities	true/false. Default = false.	--
Address.QueryAllEntities	true/false. Default = false.	--

Configuring the Process Integration for Order Fallout Management

This chapter discusses how to configure the process integration for order fallout management (OFM) for the Oracle Communications Order to Cash Integration Pack for Siebel customer relationship management (Siebel CRM), Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM) (the integration). Configuration includes:

- Configuring Oracle Application Integration Architecture (Oracle AIA)
- Configuring Siebel CRM to integrate with Oracle AIA
- Working with domain value maps (DVMs) and cross-references
- Handling error notifications
- Setting Oracle AIA configuration properties

Configuring Oracle AIA for Order Fallout Management

To configure Oracle AIA for OFM:

1. If necessary, update the data seeded by the installation process in the AIA_ERROR_NOTIFICATIONS table. See ["Using Error Type to Control Response to Order Fallout"](#) for more information.
2. Ensure that the SystemType for the applications configured in the AIA_SYSTEMS table matches the COMMON value of the TROUBLETICKET_AREA DVM.

Configuring Siebel CRM for Integrated Order Fallout Management

To configure Siebel CRM for OFM:

1. Install ACR 474. See *Siebel Maintenance Release Guide* for information.
2. Add the following dependencies to Siebel Trouble Ticket Area's List of Values (LOVs) for the trouble ticket functionality:

Area:

Oracle OSM - OLM

Oracle OSM - Provisioning

BRM_01 (add for each BRM Instance. For example, BRM_02, BRM_03)

Sub-Area:

OSM OLM ABC

OSM Provisioning ABC

BRM ABC

Add additional values, if required.

See Siebel documentation for more information on adding values to a LOV.

Working with Domain Value Maps for Order Fallout Management

DVMs are a standard feature of the Oracle 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 Management DMVs

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.
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.

See "Working with Message Transformations", Working with DVMs and Cross-References in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information.

Working with Cross-References for Order Fallout Management

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 Management Cross-Reference

Cross-Reference Table Name	Column Name COMMON	Column Value SEBL_01	Description
TROUBLETICKET_TROUBLETICKETID	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 trouble ticket ID returned by the Siebel web service is cross-referenced to the BusinessComponentID of the TroubleTicket Response enterprise business message (EBM). 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 CreateTroubleTicketSiebelCommsProvABCImpl

Handling Error Notifications for Order Fallout Management

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 BRM, see the documentation for that product.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about AIA error handling.

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 BRM, see the documentation for that product.

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 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 OSM pre-built integration options are 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 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 BRM). The expectation is that the fault coming from the application is a BPEL error code:

"{http://schemas.oracle.com/bpel/extension}bindingFault". 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 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
--	ProcessFulfillmentOrderBillingBRMCommsAddSubProcess	AIA_EH_DEFAULT	AIACOM_OFM_EXT
{http://schemas.oracle.com/bpel/extension}bindingFault	ProcessFulfillmentOrderBillingBRMCommsAddSubProcess	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>:<soapport>/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.

Configuring the Process Integration for Order Fallout Services

Configure the properties described in this section in the *AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config/AIAConfigurationProperties.xml* file.

See *Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with *AIAConfigurationProperties.xml*.

Note: Entries in the *AIAConfigurationProperties.xml* file are case sensitive

Table 27–5 shows the properties for the CreateTroubleTicketAIACommsReqImpl service.

Table 27–5 CreateTroubleTicketAIACommsReqImpl Properties

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/CreateTroubleTicketSiebelCommsProvABCServiceImpl/V1}CreateTroubleTicketSiebelCommsProvABCServiceImplService	Port Type of the provider ABCS/EBS service of the CreateTroubleTicket operation.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.Address	http://\${fp.server.soaserverhost name}:\${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketSiebelCommsProvABCServiceImpl/CreateTroubleTicketSiebelCommsProvABCServiceImpl	Address of the provider ABCS/EBS service CreateTroubleTicket operation.

Table 27–5 (Cont.) CreateTroubleTicketAIACommsReqlmpl Properties

Property Name	Value/Default Values	Description
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCImpl/V1}CreateTroubleTicketSiebelCommsProvABCImplService	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 properties for the CreateTroubleTicketSiebelCommsProvABCImpl service name.

Table 27–6 CreateTroubleTicketSiebelCommsProvABCImpl Properties

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/CreateTroubleTicketRespOSMCFSCCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCCommsJMSProducer	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/CreateTroubleTicketRespOSMCFSCCommsJMSProducer	Address of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.ServiceName	{http://xmlns.oracle.com/Industry/Comms/CreateTroubleTicketRespOSMCFSCCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCCommsJMSProducer	Service Name of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation

Table 27–6 (Cont.) CreateTroubleTicketSiebelCommsProvABCImpl Properties

Property Name	Value/Default Values	Description
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.)
TroubleTicket.GenerateTroubleTicketResponse	true/false Default: false	CreateTroubleTicketSiebelCommsProvABCImpl creates a trouble ticket response message (creates a cross-reference for the trouble ticket ID with the Siebel ID) and invokes the CommunicationsTroubleTicketResponseEB SV1 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 properties for the UpdateTroubleTicketSiebelCommsProvABCImpl service name.

Table 27–7 UpdateTroubleTicketSiebelCommsProvABCImpl Properties

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 properties for the CreateOrderFalloutNotificationOSMCFSCCommsProvImpl service.

Table 27–8 CreateOrderFalloutNotificationOSMCFSCCommsProvImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	OSMCFCS_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.CreateOrderFalloutNotification.OSMCFCS_01.EndpointURI	http://\${http.host name}:\${http.port}/orabpel/default/CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer/1.0	Endpoint URI of the service that sends the fallout notification to OSMCFCS_01. The default value is the EndpointURI of the out-of-the-box JMS enqueueing 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}**

Part III

Appendixes

Part III contains the following appendixes:

- [Cross-References for the Process Integration for Product Lifecycle Management](#)
- [Communications Orders Dictionary](#)
- [Mapping Billing Dates](#)
- [Supporting MACD Actions and Attribute Changes](#)
- [Examples of Changing the Paying Parent on Subordinate Accounts](#)
- [Configuring Multiple BRM Instances for Communications Integrations](#)
- [Changing the BRM Instance](#)
- [Expectations from an Order Management System for Billing Integration](#)
- [Using the Oracle Mediator Resequencer Feature](#)
- [Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant](#)
- [Composite Application Validation System Changes](#)
- [Reintroducing Enterprise Business Services](#)
- [Understanding Multischema Migration](#)

Cross-References for the Process Integration for Product Lifecycle 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 between Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM).

Cross-References for the Process Integration for Product Lifecycle Management

Table A-1 lists the cross-references for the process integration for PLM.

Table A-1 Cross-References for Product Lifecycle Management

Operation	Entity	Siebel CRM ID	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	\$--

Cross-Reference Values

Table A-2 shows the values for the cross-reference entries for PLM.

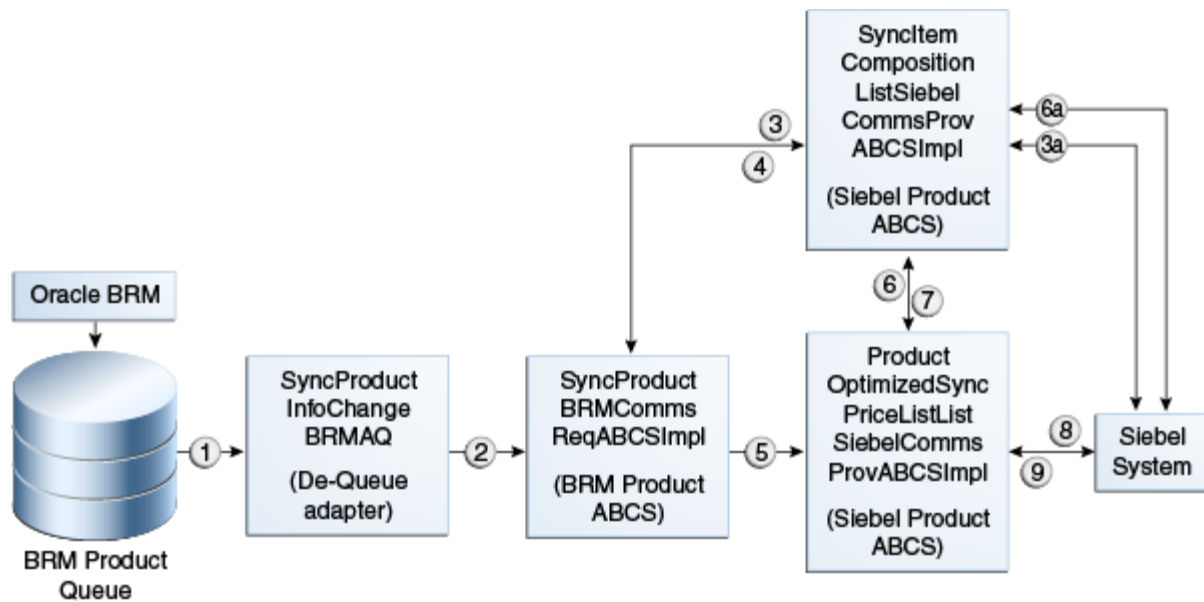
Table A-2 Values of the Cross-References for Product Lifecycle Management

Entry	Description	COMMON Value	BRM_01 Value	SEBL_01 Value	ITEM_ID_COMMON Value	LINEPRICETY PE CODE Value
ITEM_ITEMID	Cross-references the BRM ProductID and the Siebel CRM ProductID.	Auto-generated GUID	POID of BRM Product ABM	ProductID of Siebel CRM Product ABM	NA	NA
PRICELINE_ID	Cross-references the BRM Product ID and the Siebel CRM PriceLineID. Also links to the COMMON column of the ITEM_ITEMID entry.	Auto-generated GUID	POID of BRM Product ABM	Siebel PriceListItemID for the main product	From ITEM_ID.COMMON	NA
PRICELINETY PE_ID	Cross-reference BRM Product's Event and the Siebel CRM PriceLineID. Also links to the COMMON column of the ITEM_ITEMID entry.	Auto-generated GUID	POID of BRM Product ABM and Event Name	Siebel CRM PriceListItemID for the event product	From ITEM_ID.COMMON	NA
SIEBELPRODUCTEVENTX REF	Cross-references BRM Product's Event that is associated with the main product in Siebel CRM.	NA	NA	NA	From ITEM_ID.COMMON	From PRICELINETY PE_ID.COMMON

Product Synchronization Flow

Figure A-1 illustrates the events that occur for product synchronization. Tables Table A-3, Table A-4, Table A-5, Table A-6, Table A-7, Table A-8, and Table A-9 describe the entries that are made in the XREF_DATA table for each event.

Figure A-1 Product Synchronization Flow



1. Before the SyncProductBRMCommslReqABCSImpl service makes the call to the SyncItemCompositionListSiebelCommsProvABCSImpl service, the entries listed in [Table A-3](#) are made in the XREF_DATA table.

Table A-3 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 the SyncItemCompositionListSiebelCommsProvABCSImpl service, the entry listed in [Table A-4](#) is made in the XREF_DATA table.

Table A-4 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	Siebel CRM ProductID

3. Before the SyncProductBRMCommslReqABCSImpl service makes the call to the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service, the entries listed in [Table A-5](#) are made in XREF_DATA table.

Table A-5 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

- Before the ProductOptimizedSyncPriceListListSiebelProvABCImpl service calls the SyncItemCompositionListSiebelCommsProvABCImpl service, the entries listed in [Table A-6](#) are made in the XREF_DATA table.

Table A-6 XREF_DATA

XREF_TABLE_NAME	VALUE
SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE GUID2
SIEBELPRODUCTEVENTXREF	ITEM_ID_COMMON GUID1

- During the response from the SyncItemCompositionListSiebelCommsProvABCImpl service, the entries listed in [Table A-7](#) are made in the XREF_DATA table.

Table A-7 XREF_DATA

XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ITEMID	COMMON GUID2
ITEM_ITEMID	Siebel CRM ProductID of Event Product >

Note: For the simple product synchronization, the previous call is not made because the main product is synchronized as an *Item*.

- Before the ProductOptimizedSyncPriceListListSiebelProvABCImpl service calls Siebel CRM, 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	ITEM_ID_COMMON
PRICELINETYPE_ID (in case of multi-event product)	ITEM_ID_COMMON

- During the response from Siebel CRM to the ProductOptimizedSyncPriceListListSiebelProvABCImpl service, the entries listed in [Table A-9](#) are made in the XREF_DATA table.

Table A-9 XREF_DATA

XREF_TABLE_NAME	VALUE
PRICELINE_ID	Siebel CRM ProductID for Event Product
PRICELINETYPE_ID (in case of multi-event product)	Siebel CRM ProductID for Event Product

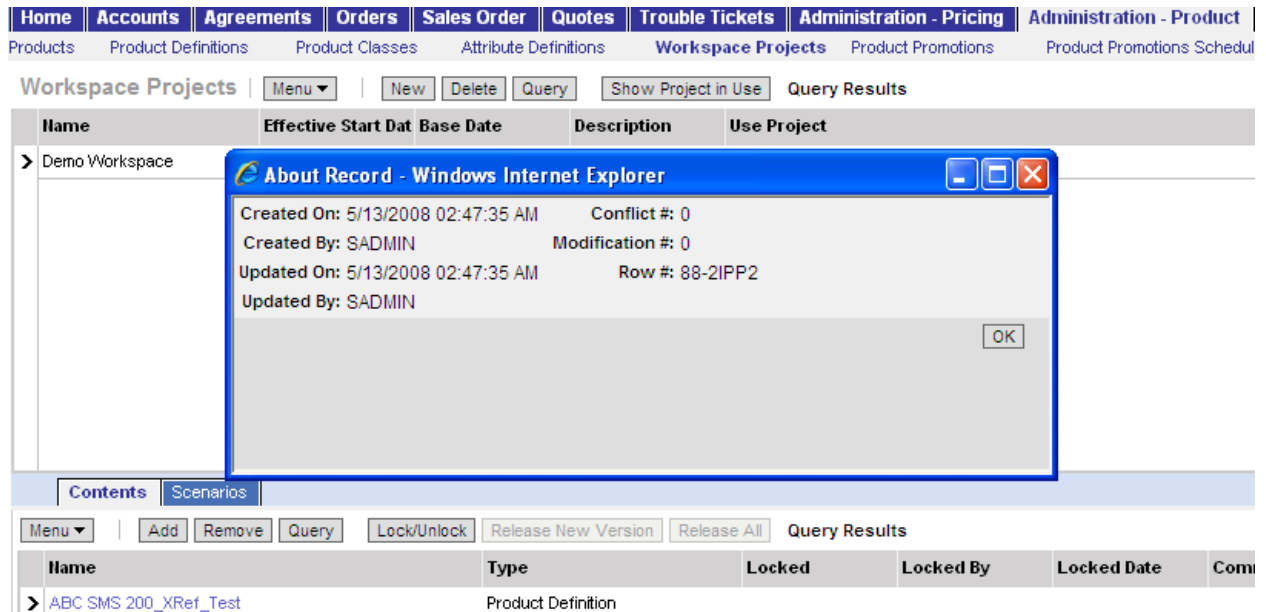
Example of Simple Product Synchronization

In the following example, a simple product is being synchronized from BRM to Siebel CRM.

- Create a simple product in BRM to be synchronized to Siebel CRM, as shown in [Figure A-2](#).

Figure A-2 Creating a Simple Product in BRM

2. Verify the synchronized records in Siebel CRM, as shown in [Figure A-3](#).

Figure A-3 Synchronized Simple Product Records in Siebel CRM

3. Verify the data entered into the XREF_DATA table is correct as shown in [table Table A-10](#).

Table A-10 Data in XREF_DATA Table for Synchronized Example Simple Product

XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
ITEM_ID	BRM_01	ROWNUM_1	BRM_PROD_01
ITEM_ID	COMMON	ROWNUM_1	COMMON_PROD_01
ITEM_ID	SEBL_01	ROWNUM_1	CRM_PROD_01
PRICELINE_ID	BRM_01	ROWNUM_2	BRM_PROD_01
PRICELINE_ID	COMMON	ROWNUM_2	COMMON_PRICE_ID1
PRICELINETYPE_ID	BRM_01	ROWNUM_3	BRM_PROD_01_EVENT1
PRICELINETYPE_ID	COMMON	ROWNUM_3	COMMON_PRICETYPE_ID1
SIEBELPRODUCTEV ENTXREF	LINEPRICETYPECODE	ROWNUM_4	COMMON_PRICETYPE_ID1
SIEBELPRODUCTEV ENTXREF_ID	ITEM_ID_COMMON	ROWNUM_4	COMMON_PROD_01
PRICELINE_ID	ITEM_ID_COMMON	ROWNUM_2	COMMON_PROD_01

Table A-10 (Cont.) Data in XREF_DATA Table for Synchronized Example Simple Product

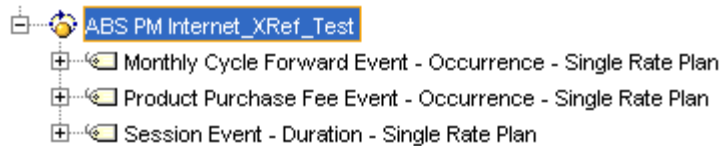
XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
PRICELINE_ID	SEBL_01	ROWNUM_2	CRM_PRICE_01

Example of Complex Product Synchronization

In the following example, a complex product is being synchronized from BRM to Siebel CRM.

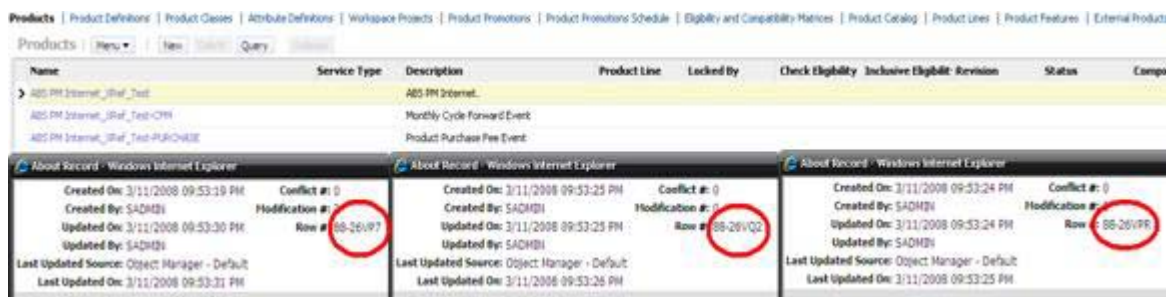
1. Create a complex product in BRM to be synchronized to Siebel CRM, as shown in [Figure A-4](#).

Figure A-4 Creating a Complex Product in BRM



2. Verify the synchronized records in Siebel CRM, as shown in [Figure A-5](#).

Figure A-5 Synchronized Complex Product Records in Siebel CRM



3. Verify the data entered into the XREF_DATA table is correct as shown in [Table A-11](#)

Table A-11 Data in XREF_DATA Table for Synchronized Example Complex Product

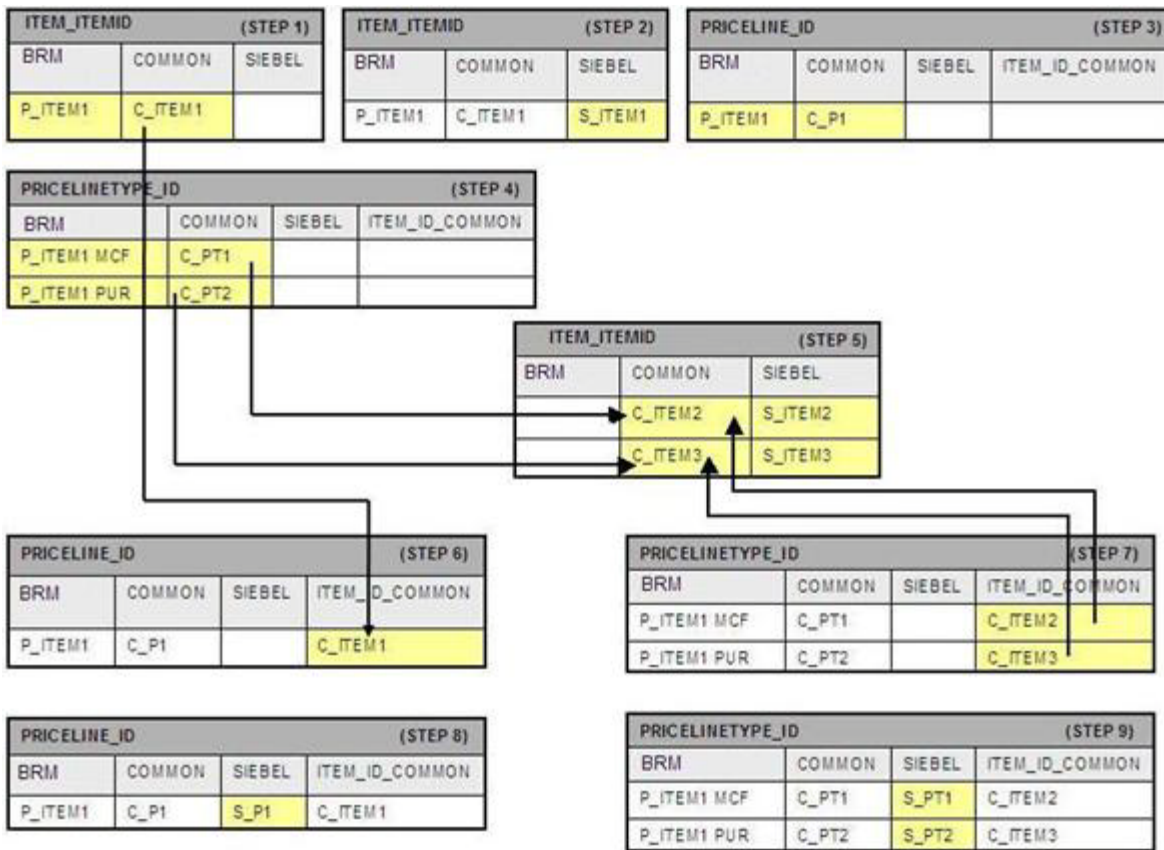
XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
ITEM_ID	BRM_01	ROWNUM_1	BRM_PROD_01
ITEM_ID	COMMON	ROWNUM_1	COMMON_PROD_01
ITEM_ID	SEBL_01	ROWNUM_1	CRM_PROD_01
PRICELINE_ID	COMMON	ROWNUM_2	BRM_PROD_01
PRICELINE_ID	BRM_01	ROWNUM_2	COMMON_PRICE_01
PRICELINETYPE_ID	COMMON	ROWNUM_3	COMMON_PRICETYP_01>
PRICELINETYPE_ID	BRM_01	ROWNUM_3	BRM_PROD_01_EVENT1

Table A-11 (Cont.) Data in XREF_DATA Table for Synchronized Example Complex

XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
PRICELINETYPE_ID	BRM_01	ROWNUM_4	BRM_PROD_01_EVENT2
PRICELINETYPE_ID	COMMON	ROWNUM_4	COMMON_PRICETYPE_02
SIEBELPRODUCTE VENTXREF	LINEPRICETYPECODE	ROWNUM_4	COMMON_PRICETYPE_01
SIEBELPRODUCTE VENTXREF_ID	ITEM_ID_COMMON	ROWNUM_4	COMMON_PROD_01
ITEM_ID	COMMON	ROWNUM_5	COMMON_PRICETYPE_02
ITEM_ID	SEBL_01	ROWNUM_5	CRM_PROD_02
PRICELINE_ID	ITEM_ID_COMMON	ROWNUM_3	COMMON_PRICETYPE_01
PRICELINE_ID	SEBL_01	ROWNUM_3	CRM_ITEM_PRICE_01
PRICELINETYPE_ID	ITEM_ID_COMMON	ROWNUM_4	COMMON_PRICETYPE_02
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.

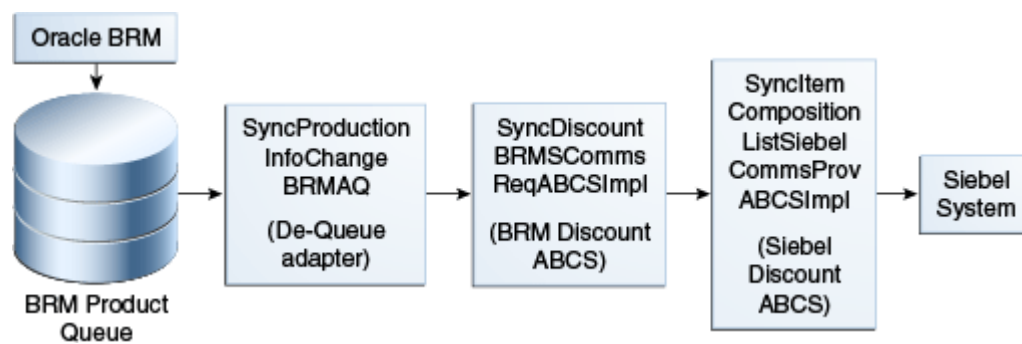
Figure A-6 Maintaining Mappings in the Cross-Reference Table



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 SyncDiscountBRMCommsReqABCSImpl service makes the call to the SyncItemCompositionListSiebelCommsprovABCSImpl service, the entries listed in Table A-12 are made in the XREF_DATA table.

Table A-12 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	COMMON GUID

Table A-12 (Cont.) XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	POID Of BRM Product

2. During the response from Siebel CRM to the SyncItemCompositionListSiebelCommsProvABCServiceImpl service, the entry listed in [Table A-13](#) is made in the XREF_DATA table.

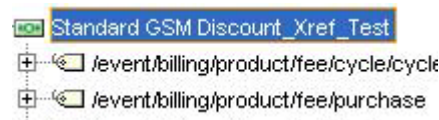
Table A-13 XREF_DATA

XREF_TABLE_NAME	VALUE
ITEM_ITEMID	Siebel CRM PRODUCTID

Example of Discount Synchronization

In this example, a discount is being synchronized from BRM to Siebel CRM.

1. Create a discount in BRM to be synchronized to Siebel CRM, as shown in [Figure A-8](#).

Figure A-8 Creating a Discount in BRM

2. Verify the synchronized records in Siebel CRM, as shown in [Figure A-9](#).

Figure A-9 Synchronized Example Discount Records in Siebel CRM

3. Verify the data entered into the XREF_DATA table is correct as shown in [Table A-14](#).

Table A-14 Data in XREF_DATA Table for Synchronized Example Discount

XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
ITEM_ID	BRM_01	ROWNUM_1	BRM_PROD_01
ITEM_ID	COMMON	ROWNUM_1	COMMON_PROD_01

Table A-14 (Cont.) Data in XREF_DATA Table for Synchronized Example Discount

XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
ITEM_ID	SEBL_01	ROWNUM_1	CRM_PROD_01

Table A-15 shows an example of the values for the cross-reference data in the ITEM_ID entry.

Table A-15 Example of Discount Cross-Reference Values

XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
ITEM_ID	BRM_01	2E60E99F02D11DCBFCA/F1F293F06D61	0.0.0.1 /discount 60048 0
ITEM_ID	COMMON	2E60E99F02D11DCBFCA/F1F293F06D61	2d313734373134383431383534303233
ITEM_ID	SEBL_01	2E60E99F02D11DCBFCA/F1F293F06D61	88-26YR5

Communications Orders Dictionary

This appendix provides a snapshot of the Communications Orders Dictionary. Communications Orders include enterprise business objects (EBOs) for sales orders, fulfillment orders, and provisioning orders.

Table B-1 defines the terms used in this appendix.

Table B-1 Communications Orders Dictionary Terminology

Term	Definition
Asset-able	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
<i>CommsOrder</i>	A variable signifying one of the following: <ul style="list-style-type: none"> ▪ Sales Order ▪ Fulfillment Order ▪ Provisioning Order

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.

Order Header Component Attributes

Table B-2 lists the attributes for order headers in communications orders.

Table B-2 Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset -able	Prior Value Available	Comments	EBO Structure XPath
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.	CommsOrderEBO/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.	CommsOrderEBO/Identification/Revision/Number

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
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	None	<i>CommsOrderEBO/PartialFulfillmentAllowedIndicator</i>
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 = <i>Cancel</i> . 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: <i>Initiate</i> or <i>Fulfill</i> .	<i>CommsOrderEBO/FulfillmentModeCode</i>
Customer Class	Identifies type of customer: Residential, Business, and so on	Residential Business	No	None	None	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountTypeCode</i>
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.	<i>CommsOrderEBO/BusinessUnitReference/BusinessUnitIdentification/ID</i>

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset -able	Prior Value Available	Comments	EBO Structure XPath
Sales Channel	Identifies the sales channel.	NA	No	None	None	<i>CommsOrderEBO/SalesChannelCode</i>
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.	<i>CommsOrderEBO/ProcessingNumber</i>
Sequence in Job	A number that identifies the order sequence within the job.	NA	No	None	None	<i>CommsOrderEBO/ProcessingSequenceNumber</i>
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	None	<i>CommsOrderEBO/ProcessingTypeCode</i>
Job Cardinality	Indicates the total number of orders within the job.	NA	No	None	None	<i>CommsOrderEBO/ProcessingQuantity</i>
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.	<i>CommsOrderEBO/ParentCommsOrderReference/CommsOrderIdentification/BusinessComponentID</i>

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
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	<i>CommsOrderEBO/FulfillmentPriorityCode</i>
Order Type	Sometimes indirectly determines sales channel to drive compensation process.	Sales Order	No	None	None	<i>CommsOrderEBO/TypeCode</i>
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	None	<i>CommsOrderEBO/RequestedDeliveryDateTime</i>
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.	<i>CommsOrderEBO/Status/Code</i>

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset -able	Prior Value Available	Comments	EBO Structure XPath
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 <i>error message, cause for cancel</i> . 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.	<i>CommsOrderEBO/Status/Description</i>
Owner Account ID	Identifies the owner account.	NA	Yes	None	Cross-referenced.	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountIdentification/BusinessComponentID</i>
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.	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountName</i>
Owner Account Number	Identifies account number to customer.	NA	Yes	None	None	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountIdentification/ID</i>

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset -able	Prior Value Available	Comments	EBO Structure XPath
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	None	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID</i>
Account Contact Address (component)	Identifies the address used to communicate with the Contact ID.	NA	Yes	None	--	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContactAddressCommunication/AddressCommunication/Address</i>
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.	<i>CommsOrderEBO/ProjectReference/ProjectIdentification/ID</i>
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	None	<i>FulfillmentOrderEBO/FulfillmentSystemTypeCode</i>
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	None	<i>FulfillmentOrderEBO/FulfillmentTargetSystemID</i>

Table B-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset -able	Prior Value Available	Comments	EBO Structure XPath
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.	<i>CommsOrderEBO/OrderChangedIndicator</i>
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.	<i>CommsOrderEBO/SalespersonPartyReference/PartyIdentification/ID</i>
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.				--	<i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContact/FirstName</i> <i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContact/LastName</i> <i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContactPhoneCommunication/PhoneCommunication/CompleteNumber</i> <i>CommsOrderEBO/CustomerPartyReference/CustomerPartyAccountContactEmailCommunication/EmailCommunication/</i>
Price List	Indicates default price list on order.	NA	No	No	If no price list value is supplied for order lines for products synchronized with rate plan selectors, the price list on the order header is used. If products on order lines were not synchronized with rate plan selectors, the default price list from the AIAConfigurationProperties.xml file is used.	<i>CommsOrderEBM/DataArea/CommsOrder/PriceListReference/PriceListIdentification/ID</i>

Order Line Component Attributes

Table B-3 lists the attributes for order lines in communications orders.

Table B-3 Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/Identification/BusinessComponentID</i>
Base Line ID	References base order line revised by this order line	NA	No	None	Uses a cross-reference.	<i>CommsOrderEBO/CommsOrderLine/OriginalCommsOrderLineReference/CommsOrderLineIdentification/BusinessComponentID</i>
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.	<i>CommsOrderLine/InstalledProductReference/InstalledProductIdentification/BusinessComponentID</i>
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.	<i>CommsOrderEBO/CommsOrderLine/Identification/ID</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/ParentCommsOrderLineIdentification/BusinessComponentID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/RootParentCommsOrderLineIdentification/BusinessComponentID</i>
Related Line ID	BRM adaptors use to relate one-time charges to base line ID.	NA	No	None	None	<i>CommsOrderEBO/CommsOrderLine/ChargeParentLineIdentification/BusinessComponentID</i>
Related Asset Integration ID	Links Move-Add to Move-Delete line items	NA	No	None	None	<i>CommsOrderEBO/CommsOrderLine/InstalledProductReference/PriorInstalledProductIdentification/BusinessComponentID</i>
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.	<i>CommsOrderEBO/CommsOrderLine/DependingCommsOrderLineReference/CommsOrderLineIdentification/BusinessComponentID</i>
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.	<i>CommsOrderEBO/CommsOrderLine/DependingCommsOrderReference/CommsOrderIdentification/BusinessComponentID</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/PromotionCommsOrderLineReference/PromotionCommsOrderLineIdentification/BusinessComponentID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/PromotionCommsOrderLineReference/InstalledProductReference/InstalledProductIdentification/BusinessComponentID</i>
Product ID	References product record based on which order line is instantiated. Foreign key reference.	NA	Yes	None	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/Identification/BusinessComponentID</i>
Quantity	Identifies the quantity of the item requested by a customer. Default is 1.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/OrderQuantity</i>
Action Code	Specify action required to meet customer request	None, Add, Update, Suspend Resume, Delete, Move-Add, Move-Delete	No	None	None	<i>CommsOrderEBO/CommsOrderLine/ServiceActionCode</i>
Deliver To Address	Address record that represents the delivery/service installation address.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/ServiceAddress/Address</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/RequestedDeliveryDateTime</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ServiceUsageStartDate</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CycleStartDate</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/PurchaseDate</i>
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	<i>CommsOrderEBO/CommsOrderLine/EffectiveTimePeriod/StartDateTime</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/EarliestDeliveryDateTime</i>
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.	<i>CommsOrderEBO/CommsOrderLine/EffectiveTimePeriod/EndDateTime</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ActualDeliveryDateTime</i>
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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ExpectedDeliveryDateTime</i>
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	<i>CommsOrderEBO/CommsOrderLine/Status/Code</i>
Milestone	Fulfillment passes the last reached milestone into this field.	NA	No	None	None	<i>CommsOrder/CommsOrderLine/MilestoneCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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: <ul style="list-style-type: none"> o Requires customer interaction o Delivery is expected to be delayed 	<i>CommsOrderEBO/CommsOrderLine/Status/Description</i>
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.	<i>CommsOrderEBO/CommsOrderLine/RevisionPermissibleCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/BillToPartyReference/CustomerPartyAccountIdentification/BusinessComponentID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/BillToPartyReference/BillingProfileReference/BillingProfileIdentification/BusinessComponentID</i>
Payment Profile	Identifies the Payment Profile.	NA	No	None	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/BillToPartyReference/BillingProfileReference/PaymentProfileReference/PaymentProfileIdentification/BusinessComponentID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/OwnerPartyReference/CustomerPartyAccountIdentification/BusinessComponentID</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/OwnerPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID</i>
Shipping Contact	Represents a contact of the customer account or service account who should be contacted for shipping purposes.	NA	Yes	None	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ShipToPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="Node"]/ValueText</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToNode"]/ValueText</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="NetworkID"]/ValueText</i>
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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="PortNumber"]/ValueText</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToPortNumber"]/ValueText</i>
Service Address Prefix	Identifies the area code/NPA for the access circuits on starting or two ends of the connection.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ServiceAddressPrefix"]/ValueText</i>
To Service Address Prefix	Identifies the area code/NPA for the access circuits on the end of the connection.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressPrefix"]/ValueText</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="AccessCircuit"]/ValueText</i>
To Access Circuit	Provides the CLLI for the access circuit on ending side of the connection.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToAccessCircuit"]/ValueText</i>
To Service Account ID	Identifies the Service Account ID associated with the end side of a network.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAccountID"]/ValueText</i>
From Service Address ID	Identifies the Service Address ID for the starting point of a network.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="FromServiceAddressID"]/ValueText</i>
To Service Address ID	Identifies the Service Address ID for the ending point of a network.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressID"]/ValueText</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServicePointID"]/ValueText</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/ServicePointCode</i>
Promotion Description	Provides short description that appears on the invoice.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/Description</i> 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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CommsOrderItemInstance/Identification/ID</i>
Balance Bundle Identification	Identifies the Balance Bundle to which a service instance belongs.	NA	NA	None	Not Used by Oracle AIA for Communications	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/BalanceBundleIdentification/BusinessComponentID</i>
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	<i>CommsOrderEBO/CommsOrderLine/Description</i>
Service Length	Indicates requested service length in Service Length Unit of Measure	NA	Yes	Yes	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ServiceTimePeriod/Duration</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
Service Length Unit of Measure	Indicates the service length unit of measure.	NA	Yes	Yes	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/ServiceTimePeriod/Duration</i>
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, REUDO, UNDO	No	None	None	<i>CommsOrderEBO/CommsOrderLine/FulfillmentModeCode</i>
Product Name	Provides the name of the product.	NA	NA	NA	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/Name</i>
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.	<i>CommsOrderEBO/CommsOrderLine/ItemReference/FulfillmentCompositionTypeCode</i>
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.	<i>CommsOrderEBO/CommsOrderLine/ItemReference/TypeCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/ItemReference/ClassificationCode [listID="BillingProductTypeCode"]</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/ClassificationCode [listID="PermittedTypeCode"]</i>
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.	<i>CommsOrderEBO/CommsOrderLine/ItemReference/ServiceIndicator</i>
Vendor	Identifies the vendor supplying the product when the product is supplied by a third-party.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/SupplierPartyReference/PartyIdentification/ID</i>
Vendor Part Number	Identifies the product part number to the vendor.	NA	Yes	AIA2.4	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/ItemIdentification/SupplierItemID</i>
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	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/ClassificationCode [listID = "FulfillmentItemCode"]</i>
Item Class Name	Determines business classification of a product.	NA	No	None	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/PrimaryClassificationCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/FulfillmentSuccessCode</i>
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.	<i>CommsOrderEBO/CommsOrderLine/StartBillingOnFirstServiceUsageIndicator</i> . We have added <i>BillingStartCode</i> to <i>ItemReference</i> , if this requirement is at the item/itemReference level and not line level then <i>BillingStartCode</i> from <i>ItemReference</i> 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	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/AlternateObjectKey [ContextID=SmartPartNumber]</i>
Network Product Flag	Indicates if this is a network product, which helps determine which user-defined attributes to expect.	True, False	No	None	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/NetworkIndicator</i>
Network Element Type	Indicates if this network product represents a node, a connection, or a network.	NA	No	None	None	<i>CommsOrderEBO/CommsOrderLine/ItemReference/NetworkItemTypeCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
Charge Frequency Code	Indicates charge frequency unit of measure, for example, monthly, quarterly, yearly.	NA	NA	NA	None	<i>CommsOrderEBO/CommsOrder>Line/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/ChargeFrequencyCode</i>
List Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/TypeCode</i>
List Price	Identifies base price of the item.	NA	Yes	None	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/UnitListPrice/Amount</i>
Sale Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/TypeCode</i>
Sale Price	Identifies net price of the item.	NA	Yes	AIA2.0	None	<i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/UnitSalePrice/Amount</i>
Pricing Commit Type	Indicates whether the pricing is Committed or Dynamic.	Common/Siebel values are true/Dynamic, false/Committed.	Yes	AIA2.4	None	<i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DynamicPricingIndicator</i>
Dynamic Discount Method	Indicates whether the discount is of type amount or percent.	Amount, Percent	Yes	AIA2.4	NA	<i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DiscountMethodCode</i>
Discount Percent	Indicates the percent by which the list price is discounted.	NA	Yes	AIA2.4	NA	<i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DiscountPercent</i>
Discount Amount	Indicates the amount by which the list price is discounted	NA	Yes	AIA2.4	NA	<i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DiscountAmount</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
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.	<i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="SpecialRating"]/ValueText [0..N]</i>
Price List	For products synchronized with a rate plan selector. Indicates which price list a purchased product should use.	NA	No	No	Service bundle components automatically receive same price list as service bundle line. If no value is supplied, the price list on the order header is used. If products on order lines were not synchronized with rate plan selectors, the default price list from the AIACConfigurationProperties.xml file is used regardless of values specified for this attribute.	<i>CommsOrderEBM/DataArea/CommsOrderLine/CommsOrderSchedule/PriceListReference/PriceListIdentification/ID</i>
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA Name	<i>CommsOrder/CommsOrderLine/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification/Name</i>
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.)	<i>CommsOrder/CommsOrderLine/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/@actionCode</i>

Table B-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset -able	Prior Value Available	Remarks	EBO Structure XPath
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	has Previous LIC Value	UDA language-independent code Value	<i>CommsOrder/CommsOrderLine/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/Value</i>
User Defined Attributes	Indicates attribute is common across all Specification components.	String, Date, Number	Yes	None	UDA Data Type	<i>CommsOrder/PriorCommsOrder/CommsOrderLine/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute"/DataTypeCode</i>
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA language-independent code Prior Value	<i>CommsOrder/PriorCommsOrder/CommsOrderLine/ItemReference/SpecificationGroup[name="ExtensibleAttributes"]/Specification[name="<OrderLine.XA.Attribute>"]/Value</i>

Mapping Billing Dates

This appendix provides information about how dates are set in Oracle Communications Billing and Revenue Management (BRM) as part of the Bill Fulfillment Order business flow.

How Dates are Set in BRM

[Table C-1](#) defines the terms used in [Table C-2](#) to explain about how dates are set in BRM.

Table C-1 *Billing Date Mapping Terminology*

Term	Abbreviation	Definition
Order Datetime	ODT	The date that the order was placed by the customer and is captured on the order in Siebel CRM. Siebel CRM defaults this date, but it can be changed by the user.
Requested Delivery Datetime	RDDT	The delivery date requested by the customer; it is captured on the order in Siebel CRM. It is also known as Due Date .
Actual Delivery Datetime	RDDT	The actual delivery date time; it is supplied by the order management system that fulfills the order, and is updated in Siebel CRM.
Purchase Start Date	-	The date as of which BRM applies purchase fees.
Cycle Start Date	-	The date as of which BRM applies cycle fees.
Usage Start Date	-	The date as of which BRM rates usage and applies usage fees.

Table C-2 Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingAccountList service	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.
CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service
Single Phase Billing - Billing Fulfillment Promotion Purchase	<p>Oracle AIA passes the Purchase Date as the Valid From date for bundle purchase (that represents purchased promotion).</p> <p>If Purchase Date is null, then it passes Requested Delivery Date and if that is null, it passes no date and BRM defaults current date.</p>	PCM_OP_SUBSCRIPTION_SET_BUNDLE	<p>Pass Order Date and Requested Delivery Date coming from Siebel CRM.</p> <p>Set Purchase Date to Actual Delivery Datetime.</p>
Single Phase Billing - Billing Fulfillment	<p>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).</p> <p>Billing dates are: Purchase Date, Cycle Start Date and Usage Start Date.</p> <p>If any of the three billing dates are not set, then Oracle AIA passes no dates to BRM and lets BRM default the Purchase, Cycle Start and Usage Start dates.</p> <p>For purchase of a service bundle, this check for existence of billing dates applies to ALL products and discounts included in the service bundle.</p>	<p>PCM_OP_CUST_MODIFY_CUSTOMER</p> <p>PCM_OP_SUBSCRIPTION_PURCHASE_DEAL</p>	<p>Pass Order Date and Requested Delivery Date coming from Siebel CRM.</p> <p>Set Purchase Date, Start Cycle, and Start Usage to Actual Delivery Datetime to explicitly control setting of billing dates.</p>

Table C-2 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
<p>Single Phase Billing - Billing Fulfillment.</p> <p>Time Based Account or Service level Subscription Product/Discount Purchase</p>	<p>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).</p> <p>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).</p>	<p>PCM_OP_MODIFY_CUSTOMER</p> <p>PCM_OP_SUBSCRIPTION_PURCHASE_DEAL</p>	<p>Populate Purchase, Cycle and Usage Start dates (this is required for enabling time-based offerings.</p> <p>Calculate the Service End Date based on TBO attributes as documented in TBO section.</p> <p>See "Supporting Time-Based Offerings on Orders" for more information.</p>
<p>Single Phase Billing - Billing Fulfillment.</p> <p>Time Based Account or service-level Subscription Product/Discount Update (of end date due to promotion upgrade or downgrade, or other pricing changes).</p>	<p>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.</p>	<p>PCM_OP_SUBSCRIPTION_SET_PRODINFO</p> <p>PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO</p>	<p>Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate prior value to trigger update.</p> <p>See "Supporting Time-Based Offerings on Orders" for more information about TBO attributes.</p>
<p>Single Phase Billing - Billing Fulfillment.</p> <p>Promotion Cancellation</p>	<p>If ADDT is passed, Oracle AIA uses that to set the VALID_TO date in BRM 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.</p>	<p>PCM_OP_SUBSCRIPTION_SET_BUNDLE</p>	<p>Pass Order Date and Requested Delivery Date coming from Siebel CRM.</p> <p>Set Actual Delivery Datetime</p>
<p>Single Phase Billing - Billing Fulfillment</p> <p>Application of Promotion Penalties or MACD One Time Charge (Suspend, Resume, Disconnect, or Move charge)</p> <p>Note - These are processed only in Billing Fulfillment.</p>	<p>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 BRM default the purchase date (to current date).</p>	<p>PCM_OP_SUBSCRIPTION_PURCHASE_DEAL</p>	<p>Set Actual Delivery Datetime</p>

Table C-2 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
<p>Single Phase Billing - Billing Fulfillment.</p> <p>Suspend, Resume, or Cancellation of Service or account-level or service-level Subscription Product/Discount.</p>	<p>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)</p>	<p>PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS</p> <p>PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS</p> <p>PCM_OP_CUST_SET_STATUS</p>	<p>Set Actual Delivery DateTime.</p>
<p>Two-Phase Billing - Billing Initiation.</p> <p>Promotion Purchase.</p>	<p>Oracle AIA passes Purchase Date as the Valid From date.</p> <p>If Purchase Date is null, then Oracle AIA passes Requested Delivery Date and if that is null, Oracle AIA passes no date and BRM defaults current date</p>	<p>PCM_OP_SUBSCRIPTION_SET_BUNDLE</p>	<p>Pass Order Date and Requested Delivery Date coming in from Siebel CRM.</p> <p>Set Purchase Date to Expected Delivery Date.</p>
<p>Two Phase Billing - Billing Initiation.</p> <p>Account-level or service-level Item Type Product Purchase.</p>	<p>Oracle AIA validates that Purchase Date is set to future (based on value of configuration property - FutureTimeThreshold).</p> <p>Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date).</p> <p>Billing Dates are - Purchase Date, Cycle Start Date and Usage Start Date.</p>	<p>PCM_OP_CUST_MODIFY_CUSTOMER</p> <p>PCM_OP_SUBSCRIPTION_PURCHASE_DEAL</p>	<p>Pass Order Date coming in from Siebel CRM.</p> <p>Set Purchase, Cycle, and Usage Date to Future (one year out to match default threshold).</p>
<p>Two Phase Billing - Billing Initiation.</p> <p>Account-level or service-level Subscription Type Product/Discount Purchase.</p>	<p>Oracle AIA validates that Cycle Start Date is set to future (based on value of configuration property - FutureTimeThresholdForBillingDates).</p> <p>Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date).</p> <p>Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date</p>	<p>PCM_OP_CUST_MODIFY_CUSTOMER</p> <p>PCM_OP_SUBSCRIPTION_PURCHASE_DEAL</p>	<p>Pass Order Date coming in from Siebel CRM.</p> <p>To support validation mode, set all three billing dates to the future (one year out to match default threshold).</p> <p>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).</p>
<p>Two-Phase Billing - Billing Fulfillment.</p> <p>Promotion Purchase.</p>	<p>Oracle AIA uses purchase date to reset Valid From date.</p>	<p>PCM_OP_SUBSCRIPTION_SET_BUNDLE</p>	<p>If purchase date had been set to Expected Delivery Date in Billing Initiation, reset purchase date to Actual Delivery Date</p>

Table C-2 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
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. See "Supporting Time-Based Offerings on Orders" for more information about TBO attributes.

D

Supporting MACD Actions and Attribute Changes

This appendix provides information about the Move, Add, Change, Disconnect (MACD) line actions that are supported by Bill Fulfillment Order business flow for orders for a given product type. It also lists which changes to product attributes the integration communicates to a billing system, such as Oracle Communications Billing and Revenue Management (BRM).

MACD Line Actions Supported

This section shows which MACD line actions are supported for each product type.

For Delete line actions, the status for the line action is **Cancelled** in Siebel CRM and **Disconnected** in BRM.

[Table D-1](#) shows which line actions are supported for marketing bundles.

Table D-1 *MACD Line Actions Supported for Marketing Bundles*

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	N/A	N/A	Yes	Yes Cross-reference tables updated to reflect new Siebel Customer Relationship Management (Siebel CRM) asset.	Ignored other than to determine original BRM asset.

[Table D-2](#) shows which line actions are supported for service bundles.

Table D-2 *MACD Line Actions Supported for Service Bundles*

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	Yes	Yes	Yes	Yes Same as Update with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

See "[Communicating Product Attribute Changes to BRM](#)" for more information about attributes.

MACD Line Actions Supported for Service Bundle Components

This section shows which MACD line actions are supported for the component products of service bundles.

The line actions are supported for billing products nested up to two levels below a service bundle. See [Figure 12–6, "Example of Nested Service Bundles"](#) for an illustration of nested billing products and nested service bundles.

[Table D–3](#) shows which line actions are supported for service-level billing subscription products.

Table D–3 MACD Line Actions Supported for Service-Level Billing Subscription Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	Yes	N/A Unsupported by Siebel CRM and BRM.	N/A Unsupported by Siebel CRM and BRM.	Yes	Yes Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

[Table D–4](#) shows which line actions are supported for service-level billing discount products.

Table D–4 MACD Line Actions Supported for Service-Level Billing Discount Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	N/A Unsupported by Siebel CRM and BRM.	N/A Unsupported by Siebel CRM and BRM.	Yes	Yes Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

[Table D–5](#) shows which line actions are supported for service-level billing item products such as one-time charges. Because no asset or purchased product instance is created, Add is the only supported action.

Table D–5 MACD Line Actions Supported for Service-Level Billing Item Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	N/A	N/A	N/A	N/A If a line is billing-initiated and a revision is processed, pricing information and billing dates can change.	N/A	N/A

MACD Line Actions Supported for Account-Level Products

This section shows which line actions are supported for account-level products.

[Table D–6](#) shows which line actions are supported for account-level billing subscription products.

Table D-6 MACD Line Actions Supported for Account-Level Billing Subscription Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	Yes	Yes	Yes	Yes	Ignored	Ignored

Table D-7 shows which line actions are supported for account-level billing discounts.

Table D-7 MACD Line Actions Supported for Account-Level Billing Discounts

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	Yes	Yes	Yes	Ignored	Ignored

Table D-8 shows which line actions are supported for account-level billing item products such as penalty charges. Because no asset or purchased product instance is created, Add is the only supported action.

Table D-8 MACD Line Actions Supported for Account-Level Billing Item Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	N/A	N/A	N/A	N/A If a line is billing-initiated and a revision is processed, billing account, billing profile, promotion reference, pricing information, and billing dates can change.	Ignored	Ignored

Communicating Product Attribute Changes to BRM

This section discusses which changes made to product attributes the integration communicates to a billing system such as BRM.

You can use supplemental orders and modify orders to update attributes as described in this section.

The information communicated for pricing includes selling price, pricing commit type, dynamic discount method, discount amount, and discount percent. The information communicated for billing dates includes purchase date, cycle start date, and usage start date.

Communicating Changes Made to Attributes of Marketing Bundles

Marketing bundles are Siebel CRM promotions, but they have no direct correlation in BRM. The integration creates purchased bundle instances under billing accounts in BRM based on promotion lines. The purchase date on promotion lines is used as the start effective date for the bundle instance.

The integration communicates the following changes to the attributes of marketing bundles to billing.

- **Billing Account:** when the billing account on a promotion line is updated on a revision or change order, the purchased bundle instance is reassigned to the new billing account.

- **Billing Dates:** updating the purchase date on a supplemental order resets the start effective date on the bundle instance.

Communicating Changes Made to Attributes of Service Bundles

The integration communicates the following changes to the attributes of service bundles to billing.

- **Service Account:**
 - When service-level balance groups are enabled, you can use change orders to transfer services to different accounts. You must also transfer all nested service bundles.
 - When service-level balance groups are disabled, the integration does not support service account changes. See ["Supporting Balance Groups"](#) for more information about service-level balance groups.
- **Billing Account and Billing Profile:**
 - When service-level balance groups are enabled, you can use change orders to update the billing account or the billing profile for the service bundle.
 - When service-level balance groups are disabled, the integration supports only changes to both billing account and billing profile at the same time as part of changing the paying parent on a subordinate account. See ["Examples when Service-Level Balance Groups Are Disabled"](#) for an example.
- **Service ID**

Communicating Changes Made to Attributes of Service Bundle Components

This section lists the changes to service bundle components that the integration communicates to billing.

For billing products and discounts that are components of service bundles, the integration uses the service account, billing account, billing profile, and service ID of the parent service bundle.

For nested service bundles, ensure that the attributes are the same as those assigned to the parent service bundle and that any changes made to the attributes of a nested service bundle match changes made to the parent service bundle.

Communicating Changes Made to Attributes of Service-Level Billing Subscription Products

The integration communicates the following changes to the attributes of service-level billing subscription products to billing.

- **Pricing Information**
- **Promotion Reference:** when the promotion reference of a service-level subscription product changes, the integration reassigns the purchased product instance in BRM to the new service instance.
- **Billing Dates:**
 - Cannot be reset using change orders
 - Cycle and usage start dates can be reset using revisions on billing initiation if the previous dates are not current

- In two-phase billing, cycle and usage start dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current
- **End Date:** can be updated by change orders that change the duration, as in the case of promotion upgrades or downgrades.

Communicating Changes Made to Attributes of Service-Level Billing Discount Products

The integration communicates the following changes to the attributes of service-level billing discount products to billing.

- **Promotion Reference:** when the promotion reference of a service-level billing discount changes, the integration repoints the purchased discount instance in BRM to the new service instance.
- **Billing Dates:**
 - Cannot be reset using change orders
 - Cycle and usage start dates can be reset using revisions on billing initiation if the previous dates are not current
 - In two-phase billing, cycle and usage start dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current
- **End Date:** can be updated by change orders that change the duration, as in the case of promotion upgrades or downgrades.

Communicating Changes Made to Attributes of Service-Level Billing Item Products

Because the integration does not create a purchased product instance in BRM for service-level billing item products like one-time charges, you cannot submit change orders for this product type.

The integration communicates the following changes to the attributes of service-level billing item products to billing:

- **Pricing Information, Promotion Reference, Quantity:** can be updated on supplemental orders for new purchases
- **Billing Dates:** in two-phase billing, billing dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current

Communicating Changes Made to Attributes of Service-Level Special Rating Products

The integration communicates the following changes to the attributes of service-level special rating products to billing:

- **Friends and Family List Reference:** change orders changing the friends and family list reference update the list values to the new values from the new friends and family list.

The integration does not check for changes to the friends and family list reference on revision orders when the list product has been billing-initiated. This is because friends and family lists are primarily used by wireless services, where there is no fulfillment latency between provisioning and billing, so two-phase billing is not used. See "[Supporting Friends and Family](#)" for more information.

Communicating Changes Made to Attributes of Account-Level Products

This section lists the changes to account-level products that the integration communicates to billing.

BRM does not support transferring account-level products or discounts from one account to another. Siebel CRM validates this.

Communicating Changes Made to Attributes of Account-Level Billing Subscription Products

A subscription product that is not bundled in a service bundle is purchased at the account level.

The integration communicates the following changes to the attributes of account-level billing subscription products to billing:

- **Billing Account and Billing Profile**
- **Pricing Information**
- **Promotion Reference:** when the promotion reference of an account-level billing subscription product changes, the integration repoints the purchased product instance in BRM to the new bundle instance.
- **Billing Dates:**
 - Cannot be reset using change orders
 - Cycle and usage start dates can be reset using revisions on billing initiation if the previous dates are not current
 - In two-phase billing, cycle and usage start dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current
- **End Date:** can be updated by change orders that change the duration, as in the case of promotion upgrades or downgrades.

Communicating Changes Made to Attributes of Account-Level Billing Discounts

A discount that is not bundled in a service bundle is purchased at the account level. Discount products are not priced.

The integration communicates the following changes to the attributes of account-level billing discounts to billing:

- **Billing Account and Billing Profile**
- **Promotion Reference:** when the promotion reference of an account-level billing discount changes, the integration repoints the purchased discount instance in BRM to the new bundle instance.
- **Billing Dates:**
 - Cannot be reset using change orders
 - Cycle and usage start dates can be reset using revisions on billing initiation if the previous dates are not current
 - In two-phase billing, cycle and usage start dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current

- **End Date:** can be updated by change orders that change the duration, as in the case of promotion upgrades or downgrades.

Communicating Changes Made to Attributes of Account-Level Billing Item Products

Because the integration does not create a purchased product instance in BRM for account-level billing item products like penalties, you cannot submit change orders for this product type.

The integration communicates the following changes to the attributes of account-level billing item products to billing:

- **Billing Account and Billing Profile, Pricing Information, Promotion Reference:** can be updated on supplemental orders.
- **Billing Dates:**
 - Cycle and usage start dates can be reset using revisions on billing initiation if the previous dates are not current
 - In two-phase billing, cycle and usage start dates that were set by billing initiation can be reset with billing fulfillment if the previous dates are not current

Examples of Changing the Paying Parent on Subordinate Accounts

This appendix gives examples of changing the paying parent on subordinate accounts. When service-level balance groups are enabled, there are different options for changing the paying parent than when service-level balance groups are disabled. The examples illustrate these options.

About the Examples

The examples show Siebel customer relationship management (Siebel CRM) orders and illustrations of the conceptual arrangement of services, billing profiles, balance groups, and bill units resulting from sending the orders to Oracle Communications Billing and Revenue Management (BRM).

The first order in each example is the base order that purchases services for a subordinate account under a paying parent account. When the service account on an order is different from the billing account, BRM creates the service account with a dummy **/billinfo** object as a subordinate account under the billing account. The integration points the dummy **/billinfo** object to the billing profile of the paying parent account.

The subsequent orders in each example are change orders that transfer the services purchased in the first order to new paying parent accounts. When the services are transferred, the integration moves the subordinate account under the new paying parent account in BRM. The integration also creates new balance groups and dummy **/billinfo** objects for the services in the subordinate account pointing to the paying parent account's chosen billing profile.

Because of the automatic naming conventions for balance groups, the new balance groups for transferred services have the same name as the old balance groups. The examples show a different number in the balance group name to illustrate that these are different balance groups. In reality the balance groups would be identical.

Because BRM does not support subordinate accounts with more than one paying parent, the change orders must transfer all services for the subordinate account at once.

[Table E-1](#) defines the abbreviations used in the examples.

Table E-1 Abbreviations Used in the Examples

Abbreviation	Description
S	Service: a Siebel CRM service bundle that represents a service instance in BRM
SA	Service account: the subordinate account

Table E-1 (Cont.) Abbreviations Used in the Examples

Abbreviation	Description
BA	Billing account: the parent account
BG	Balance group: the BRM balance group in which each service is tracked
BP	Billing profile: the billing profile on the parent account
DBP	Dummy /billinfo object: a /billinfo object created for a subordinate account that the integration points to the parent account's billing profile

Examples when Service-Level Balance Groups Are Enabled

This section gives examples of changing the paying parent on subordinate accounts when service-level balance groups are enabled.

Changing the Paying Parent for a Subordinate Account with Separate Billing Profiles

This example shows:

- A base order that purchases two services with separate billing profiles for a subordinate account
- Change orders that change the paying parent for the subordinate account and all of its services

Table E-2 shows the base order. A separate billing profile is assigned to each service.

Table E-2 Base Order for Services with Separate Billing Profiles

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP2

Figure E-1 shows the result of sending the base order to billing. The services are tracked in separate balance groups that point to separate dummy **/billinfo** objects.

Figure E-1 Result of Base Order for Services with Separate Billing Profiles



Table E-3 shows the change order to change the paying parent for the subordinate account while maintaining separate billing profiles for the services.

Table E-3 Order to Change the Paying Parent with Separate Billing Profiles

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP3
UPDATE	S2	SA1	BA2	BA2-BP4

Figure E-2 shows the result of sending the change order to billing.

Figure E-2 Result of Change Order to Change the Paying Parent with Separate Billing Profiles



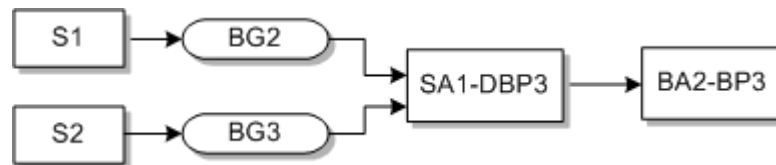
Alternatively, the change order can change the paying parent for the subordinate account and assign the same billing profile to both services. Table E-4 shows this alternative order.

Table E-4 Alternative Change Order to Change the Paying Parent with a Single Billing Profile

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP3
UPDATE	S2	SA1	BA2	BA2-BP3

Figure E-3 shows the results of sending the alternative order to billing.

Figure E-3 Result of Alternative Change Order to Change the Paying Parent with a Single Billing Profile



Changing the Paying Parent for a Subordinate Account with a Single Billing Profile

This example shows:

- A base order that purchases two services with a single billing profile for a subordinate account
- Change orders that change the paying parent for the subordinate account and all of its services

Table E-5 shows the base order. The same billing profile is assigned to both services.

Table E-5 Base Order for Services with a Single Billing Profile

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP1

Figure E-4 shows the result of sending the base order to billing. The services are tracked in separate balance groups that point to the same dummy /billinfo object.

Figure E-4 Result of Base Order for Services with a Single Billing Profile

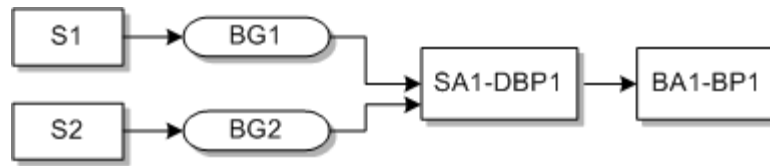


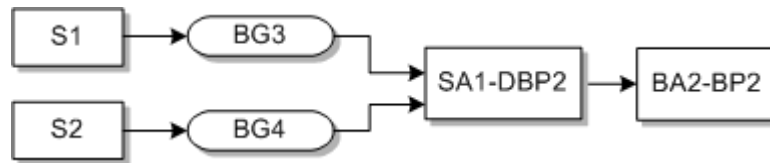
Table E-6 shows the change order to change the paying parent for the subordinate account while maintaining a single billing profile for the services.

Table E-6 Order to Change the Paying Parent with a Single Billing Profile

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP2
UPDATE	S2	SA1	BA2	BA2-BP2

Figure E-5 shows the result of sending the change order to billing.

Figure E-5 Result of Order to Change the Paying Parent with a Single Billing Profile



Alternatively, the change order can change the paying parent for the subordinate account and assign separate billing profiles to the services. Table E-7 shows this alternative order.

Table E-7 Alternative Order to Change the Paying Parent with Multiple Billing Profiles

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP2
UPDATE	S2	SA1	BA2	BA2-BP3

Figure E-6 shows the results of sending the alternative order to billing.

Figure E-6 Result of Alternative Order to Change the Paying Parent with Multiple Billing Profiles



Changing the Paying Parent for Multiple Subordinate Accounts

This example shows:

- A base order that purchases one service each for two subordinate accounts

- A change order that changes the paying parent for both subordinate accounts and all their services

Table E-8 shows the base order. A separate billing profile is assigned to each service.

Table E-8 Base Order for Services in Multiple Subordinate Accounts

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA2	BA1	BA1-BP2

Figure E-7 shows the results of sending the base order to billing. The services are tracked in separate balance groups that point to separate dummy /billinfo objects.

Figure E-7 Result of Base Order for Services in Multiple Subordinate Accounts



Table E-9 shows the change order to change the paying parent for both subordinate accounts at once while maintaining separate billing profiles.

Table E-9 Order to Change the Paying Parent for Multiple Subordinate Accounts with Multiple Billing Profiles

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP3
UPDATE	S2	SA2	BA2	BA2-BP4

Figure E-8 shows the results of sending the change order to billing.

Figure E-8 Result of Order to Change the Paying Parent for Multiple Subordinate Accounts with Multiple Billing Profiles



Alternatively, the change order can change the paying parent for the subordinate accounts and assign the same billing profile to the services. Table E-10 shows this alternative order.

Table E-10 Alternative Order to Change the Paying Parent for Multiple Subordinate Accounts with a Single Billing Profile

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP3
UPDATE	S2	SA2	BA2	BA2-BP3

Figure E-9 shows the result of sending the alternative order to billing.

Figure E-9 Result of Alternative Order to Change the Paying Parent for Multiple Subordinate Accounts with a Single Billing Profile



Examples when Service-Level Balance Groups Are Disabled

This section gives examples of changing the paying parent on subordinate accounts when service-level balance groups are disabled.

Changing the Paying Parent for a Subordinate Account

This example shows:

- A base order that purchases two services for a subordinate account
- A change order that changes the paying parent for the subordinate account and its services

Table E-11 shows the base order. The same billing profile is assigned to both services.

Table E-11 Base Order for Services in a Subordinate Account

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP1

Figure E-10 shows the result of sending the base order to billing. The services are tracked together under the default account-level balance group.

Figure E-10 Result of Base Order for Services in a Subordinate Account



The graphic shows that the services S1 and S2 are tracked together in one balance group. A dummy `/billinfo` object under the subordinate account is created for the balance group. The dummy `/billinfo` object points to the billing profile (B A1-B P1) on the parent account (B A1).

Table E-12 shows the change order to change the paying parent for the subordinate account.

Table E-12 Order to Change the Paying Parent for a Subordinate Account

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S1	SA1	BA2	BA2-BP2
UPDATE	S2	SA1	BA2	BA2-BP2

Figure E-11 shows the result of sending the change order to billing.

Figure E-11 Result of Order to Change the Paying Parent for a Subordinate Account



The graphic shows that the services purchased on the base order are tracked together in a new balance group and a new dummy **/billinfo** object which points to a new billing profile (B A2-B P2) on the new parent account (B A2) is created.

Changing the Paying Parent for Multiple Subordinate Accounts with Separate Billing Profiles

This example shows:

- A base order that purchases one service each for two subordinate accounts
- A change order that changes the paying parent for both accounts and all their services

Table E-13 shows the base order. A separate billing profile is assigned to each subordinate account.

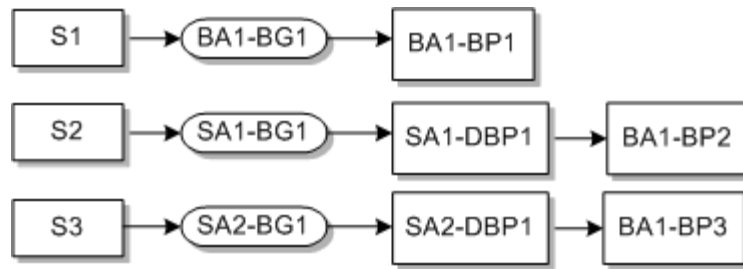
Note: When service-level balance groups are disabled, a parent account can only use multiple billing profiles to pay for services under separate subordinate accounts. A single subordinate account cannot have multiple billing profiles. See "[Problems When Integrating Separate Billing Profiles on the Same Account](#)" for an example of this restriction.

Table E-13 Base Order for Multiple Subordinate Accounts

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	BA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP2
ADD	S3	SA2	BA1	BA1-BP3

Figure E-12 shows the result of sending the base order to billing. Each service is tracked under the default account-level balance group for its account.

Figure E-12 Result of Base Order for Multiple Subordinate Accounts



Note: The account -level balance group for the parent account (BA1) references the first billing profile that is created for that account (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-14 shows the change order to change the paying parent for both subordinate accounts using one billing profile under the new parent.

Table E-14 Order to Change the Paying Parent for Multiple Subordinate Accounts

Action	Service Number	Service Account	Billing Account	Billing Profile
UPDATE	S2	SA1	BA2	BA2-BP4
UPDATE	S3	SA2	BA2	BA2-BP4

Figure E-13 shows the result of sending the change order to billing.

Figure E-13 Result of Changing the Paying Parent for Multiple Subordinate Accounts



Because S1 was purchased directly under the parent account, it was not included in the update order and remains unchanged.

Alternatively, the order could maintain separate billing profiles for the subordinate accounts.

When service-level balance groups are disabled, you cannot transfer subordinate accounts to new billing profiles on the same paying parent account.

Problems When Integrating Separate Billing Profiles on the Same Account

This example shows:

- A base order that purchases two services with separate billing profiles for one subordinate account
- A resubmitted version of the order to correct errors

Table E–15 shows the base order. Each service is assigned a separate billing profile.

Table E–15 Base Order for Services with Separate Billing Profiles

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP2

When the order is sent to billing, the integration creates two dummy **/billinfo** objects in BRM:

- DBP1 under the subordinate account (SA1) pointing the primary billing profile (BP1) on the parent account (BA1)
- DBP2 under SA1 pointing to a separate billing profile (BP2) on BA1

The default account-level balance group points to the primary billing profile (BP1) on BA1.

If both services are purchased on a single order, as in Table E–15, the integration uses billing account and billing profile on the first service purchased for all the remaining services on the order. The billing account and billing profile specified on the second service are ignored. The result of sending the order to billing is the same as in Figure E–10, "Result of Base Order for Services in a Subordinate Account".

Though the order is processed successfully, it results in a mismatch in the billing profile in Siebel CRM and the **/billinfo** object in BRM. To prevent this mismatch, ensure that orders in Siebel CRM use a single billing profile for all services purchased for one account when service-level balance groups are disabled.

If the services are purchased on two separate orders, the order for S1 succeeds but the order for S2 fails.

The integration fails to process S2 because it is attempting to point the dummy **/billinfo** object for S2 to a billing profile different than the default account-level billing profile.

Table E–16 shows the revision order to resolve the failure or data mismatch by assigning S2 to BP1.

Table E–16 Resubmitted Order

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	S1	SA1	BA1	BA1-BP1
ADD	S2	SA1	BA1	BA1-BP1

After the order is processed to billing, the result is the same as in Figure E–10, "Result of Base Order for Services in a Subordinate Account".

Though the revised order is processed successfully, the dummy **/billinfo** object for S2 (DBP2) remains in BRM under SA1. A change order to transfer SA1 to a new parent account will fail integration. BRM requires all services of a subordinate account to be paid by the same parent, so all services of a subordinate must be transferred to the new parent together. Because DBP2 is not used by any service it will not be transferred to a new parent, causing order failure.

To resolve the failure you must manually move SA1 under the new parent in BRM so that BRM can repoint all of the dummy **/billinfo** objects and resubmit the change order.

Configuring Multiple 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 (BRM) instances for the process integrations in Oracle AIA for Communications.

Understanding System Codes in Oracle AIA

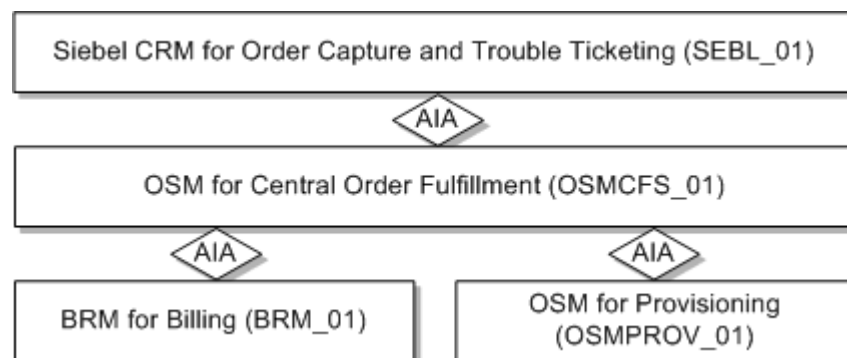
Each system instance is identified in Oracle AIA by a unique identifier, called a system code. The system codes help Oracle AIA identify the source or destination of a message.

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 BRM instance for order billing.
OSMCFS_01	The central fulfillment system (CFS) instance of Oracle Communications Order and Service Management (OSM) system. This is the instance responsible for customer orders in order management.
OSMPROV_01	The provisioning system instance of 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 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 OSM-specific properties are in **AIAConfigurationProperties.xml**.

See "Building AIA Integration Flows" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information.

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.

Configuring Multiple 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 BRM instance. To configure the second and third BRM instances, follow these steps. These steps guide you through the process to add billing instances. Repeat them for each additional BRM instance.

These abbreviations are used in this example:

- BRM_01: The first BRM instance that is installed as delivered.
- BRM_02: The second BRM instance for which the following sample configuration should be followed.
- BRM_03: The third 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.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about Composite, Oracle Mediator, and JDeveloper IDE.

To configure a second 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_OVERRIDEYPECODE.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.

See "Building AIA Integration Flows" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information.

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

3. 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.
4. Add additional 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.
5. Modify all the BRM Provider service configurations in the **AIAConfigurationProperties.xml** file, which is located in:

\$AIA_HOME/aia_instances/INSTANCE_NAME/AIAMetaData/config

The service configuration contains the Partner link details to 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 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 BRM instance. Change the properties for these services:
 - ProcessFulfillmentOrderBillingBRMCommsProvABCImpl
 - ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
 - ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
 - UpdateCreditAlertBRMCommsProvABCImpl
 - SyncCustomerPartyListBRMCommsProvABCImpl
 - QueryServiceUsageBRMCommsProvABCImpl
 - QueryReceivedPaymentListBRMCommsProvABCImpl
 - QueryInvoiceListBRMCommsProvABCImpl
 - QueryInstalledProductListBRMCommsProvABCImpl
 - QueryCustomerPartyListBRMCommsProvABCImpl
 - QueryAccountBalanceAdjustmentBRMCommsProvABCImpl
 - ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl
 - CreateReceivedPaymentBRMCommsProvABCImpl
 - CreateAccountBalanceAdjustmentBRMCommsProvABCImpl

- b. To update the **AIAConfigurationProperties.xml** file in the MDS repository, login to the AIA Console (<http://<host>:<port>/AIA>). Go to **Setup** and 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>
```

Creating a New Consumer for Product Synchronization

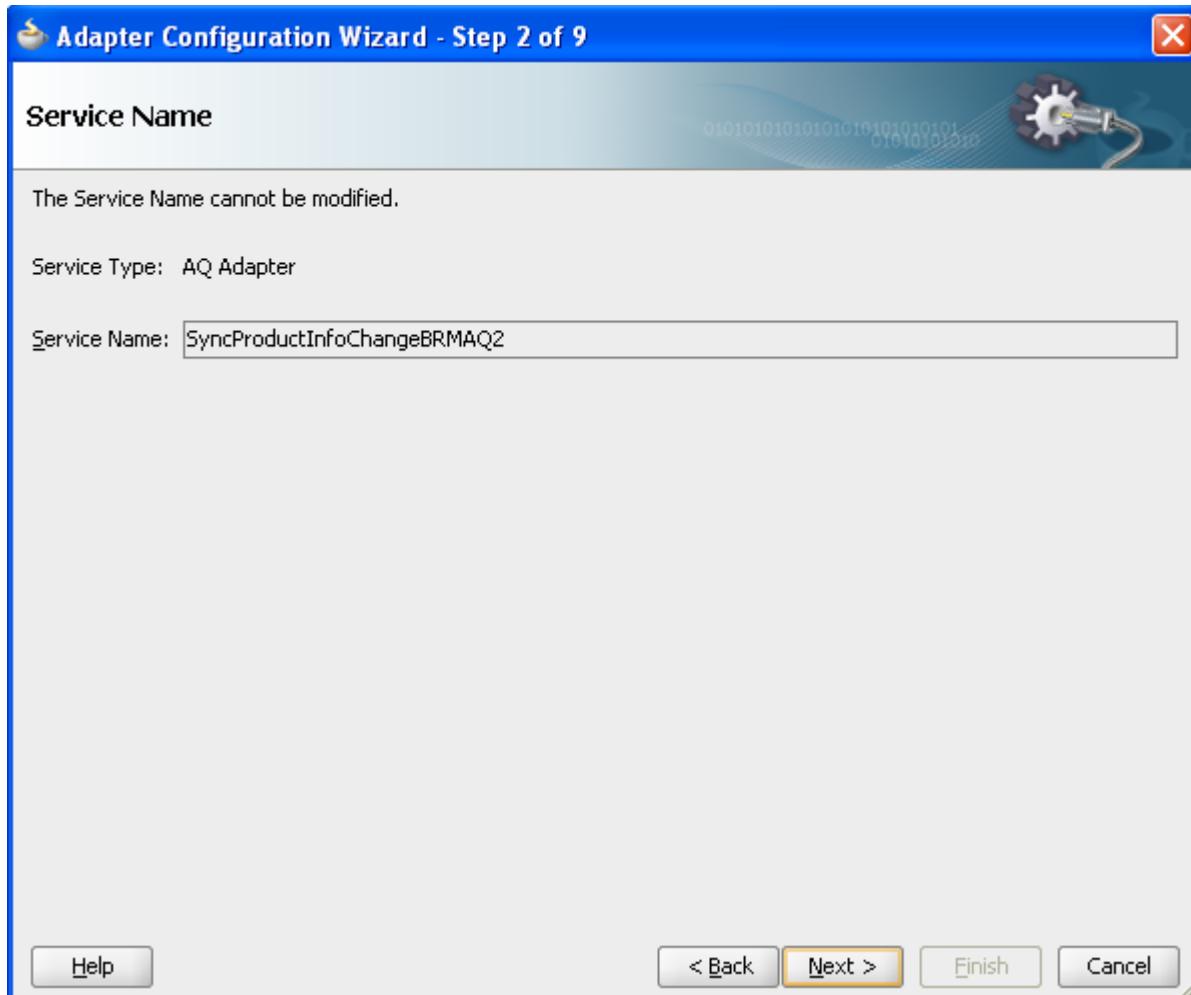
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for product synchronization:

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.
3. Click **Next** to display Step 2 (Service Name).

Go to the **Service Name** field and enter `SyncProductInfoChangeBRMAQ2`, as shown in [Figure F-2](#).

Figure F-2 Adapter Configuration Wizard - Step 2 of 9



4. Click **Next** to display Step 3 (Service Connection).

Go to **Connection** and select the BRM Database Connection. Go to the **JNDI Name** field and enter *eis/AQ/PortalEventSyncAQ2*, as shown in Figure F-3.




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 "[Creating a Data Source and Connection Factory](#)"

Figure F-3 Adapter Configuration Wizard - Step 3 of 9

Adapter Configuration Wizard - Step 3 of 9

Service Connection

A Database Connection is required to configure this adapter. Select a database connection already defined in your project or create a New Connection.

_Connection:   

User Name: pin7834

Driver: oracle.jdbc.OracleDriver

Connect String: jdbc:oracle:thin:@kamasu-5.us.oracle.com :1521:pindb

Specify the JNDI name for the database. Note: The deployment descriptor of the AQ adapter must associate this JNDI name with configuration properties required by the adapter to access the database.

JNDI Name:

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-4](#).

Figure F-4 Adapter Configuration Wizard - Step 4 of 9


Adapter Configuration Wizard - Step 4 of 9

Adapter Interface

The adapter interface is defined by a wsdl that is generated using the operation name and schema(s) specified later in this wizard. Optionally, the adapter interface may be defined by importing an existing WSDL.

Interface: Define from operation and schema (specified later)

Import an existing WSDL

WSDL URL: 

Port Type:

Operation:

Callback Port Type:

Callback Operation:

6. Click **Next** to display Step 5 (Operation).
Go to the **Operation Name** field and enter *Dequeue*, as shown in [Figure F-5](#).

Figure F-5 Adapter Configuration Wizard - Step 5 of 9

Adapter Configuration Wizard - Step 5 of 9

Operation

The AQ Adapter supports three operations. There is a Dequeue operation that polls for incoming messages from a queue, an Enqueue operation that puts outgoing messages on a queue, and an Enqueue/Dequeue operation that puts outgoing messages on a queue and expects response messages on a queue. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type:

- Dequeue
- Enqueue
- Enqueue/Dequeue

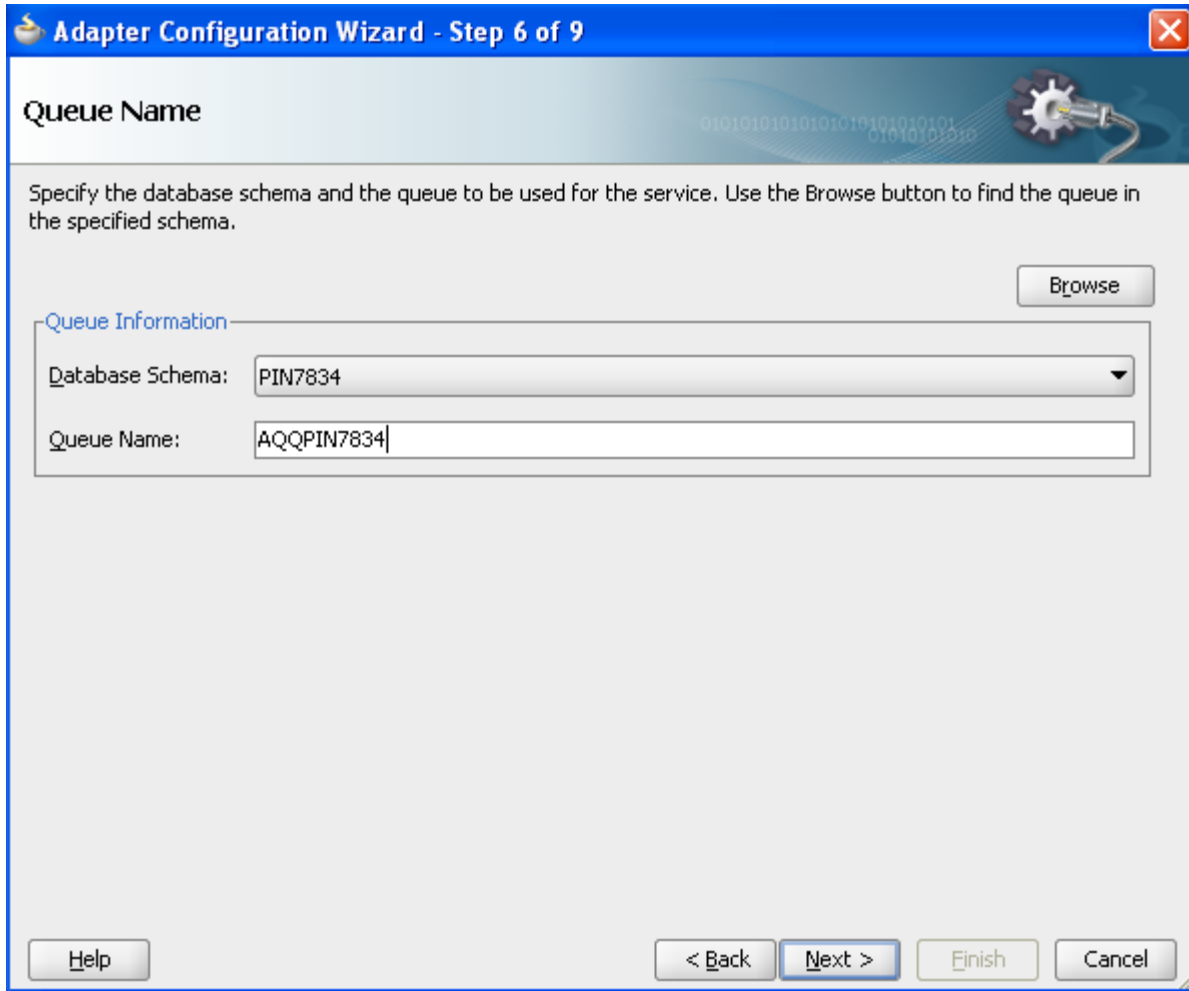
Operation Name:

Help < Back **Next >** Finish Cancel

7. Click **Next** to display Step 6 (Queue Name).

Go to the **Database Schema** field and select the additional BRM Instance's database schema, and then enter the queue name configured for this BRM instance, as shown in [Figure F-6](#).

Figure F-6 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-7](#).

Figure F-7 Adapter Configuration Wizard - Step 7 of 9

Adapter Configuration Wizard - Step 7 of 9

Queue Parameters

Specify parameters for the dequeue operation.

Correlation Id:

Dequeue Condition

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-8](#).

Figure F-8 Adapter Configuration Wizard - Step 8 of 9

Object Payload

The queue, PIN7834.AQQPIN7834, contains a structured object payload (ADT) named PIN7834.PIN_EVENT_TY. The business payload may be based on the whole object or a single field in the object. If the payload is the whole object, then the message schema will be automatically generated. In the case of single field, you must specify the field that contains the payload and subsequently also provide the message schema definition.

Business Payload: Whole Object PIN_EVENT_TY
 Field within the Object

Payload Field Options

Field Name:

Access to non-payload fields also needed

Validate Payload

Buttons: Help, < Back, Next >, Finish, Cancel

10. Click **Next** to display Step 9 (Finish).

Click **Finish** to create the AQ Adapter service.

You must now:

- 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 SyncProductBRMVCCommsReqABCSEImpl web service.
 - Add a routing rule from Oracle Mediator to SyncProductBRMCommsReqABCSEImpl web service for SyncProduct operation.
1. In **composite.xml**, provide the port and location information from concrete WSDL of the SyncProductBRMCommsReqABCSEImpl web service.

```
<reference name="SyncProductBRMCommsReqABCSEImpl"
  ui:wSDLLocation="oramds:/apps/AIAMetaData/AIAComponents/ApplicationConnectorServiceLibrary/BRM/V1/RequesterABCS/SyncProductBRMCommsReqABCSEImpl.wsdl">
<interface.wSDL
  interface="http://xmlns.oracle.com/ABCSEImpl/BRM/Industry/Comms/SyncProductBRMCommsReqABCSEImpl/V1#wSDL.interface(SyncProductBRMReqABCSEImpl)" />
<binding.ws
```

```

port="http://xmlns.oracle.com/ABCImpl/BRM/Industry/Comms/SyncProductBRMCommsRe
qABCImpl/V1#wSDL.endpoint(SyncProductBRMReqABCImpl/SyncProductBRMReqABCImpl)
"
location="http://<hostname>:<Port>/soa-infra/services/default/SyncProductBRMCom
msReqABCImpl/SyncProductBRMCommsReqABCImpl?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 BRM instance.

Creating a New Consumer for Discount Synchronization

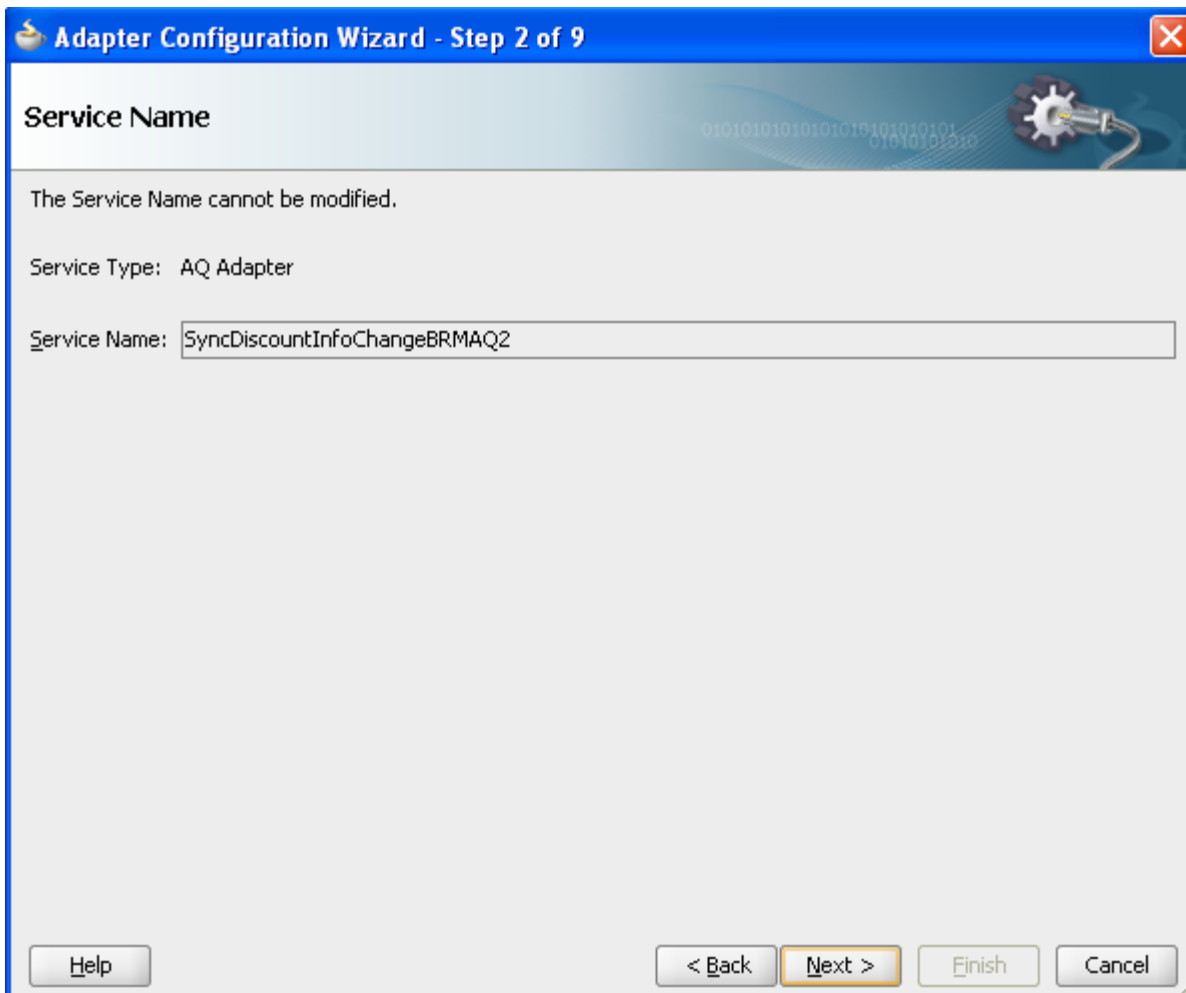
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for discount synchronization:

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.
3. Click **Next** to display Step 2 (Service Name).

Go to the **Service Name** field and enter `SyncDiscountInfoChangeBRMAQ2`, as shown in [Figure F-9](#).

Figure F-9 Adapter Configuration Wizard - Step 2 of 9



4. Click **Next** to display Step 3 (Service Connection).

Go to **Connection** and select the BRM Database Connection. Go to the **JNDI Name** field and enter *eis/AQ/PortalEventSyncAQ2*, as shown in [Figure F-10](#).




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 "[Creating a Data Source and Connection Factory](#)".

Figure F-10 Adapter Configuration Wizard - Step 3 of 9

Adapter Configuration Wizard - Step 3 of 9

Service Connection

A Database Connection is required to configure this adapter. Select a database connection already defined in your project or create a New Connection.

_Connection:   

User Name: pin7834
Driver: oracle.jdbc.OracleDriver
Connect String: jdbc:oracle:thin:@kamasu-5.us.oracle.com :1521:pindb

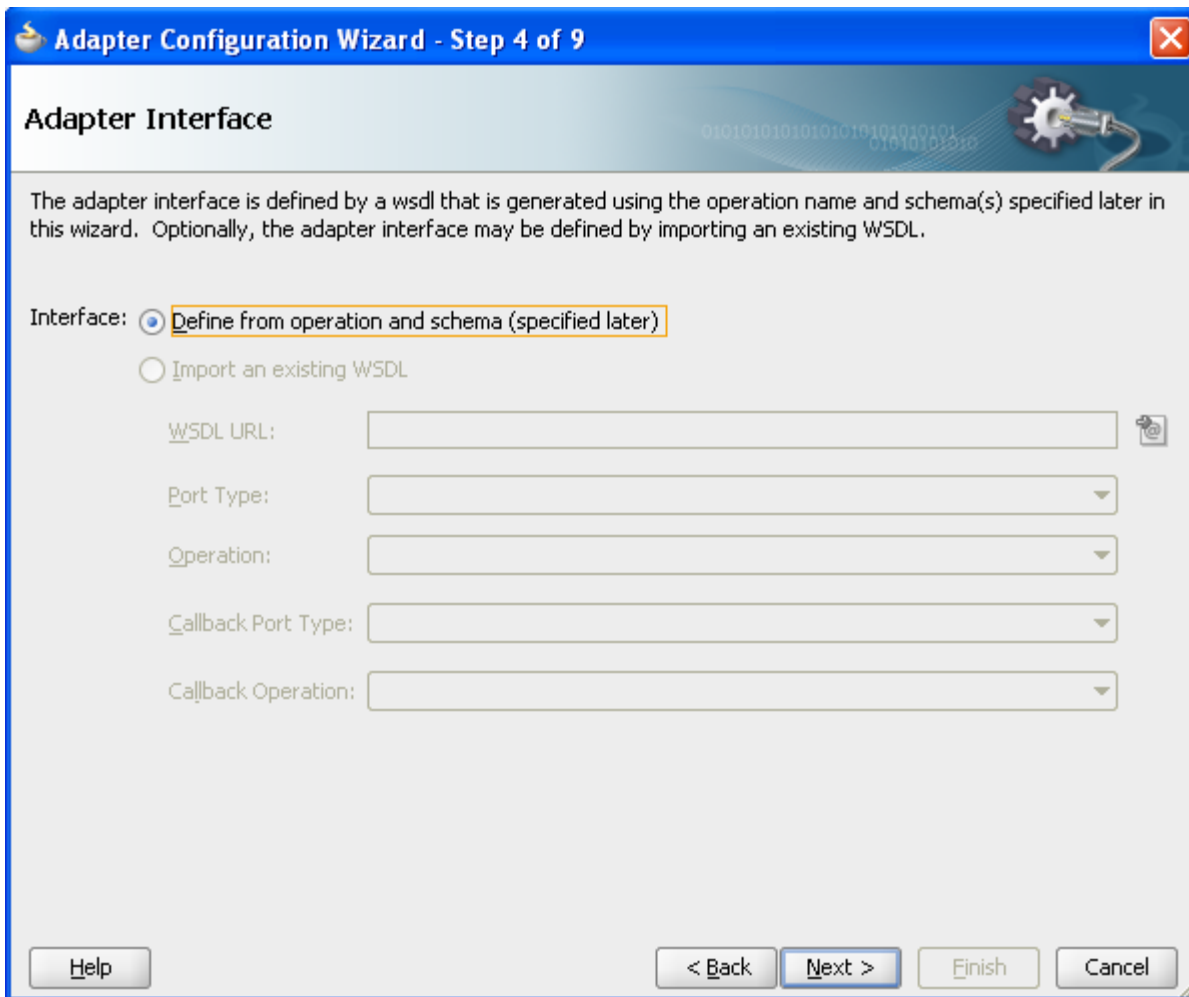
Specify the JNDI name for the database. Note: The deployment descriptor of the AQ adapter must associate this JNDI name with configuration properties required by the adapter to access the database.

JNDI Name:

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-11](#).

Figure F–11 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–12](#).

Figure F-12 Adapter Configuration Wizard - Step 5 of 9

Adapter Configuration Wizard - Step 5 of 9

Operation

The AQ Adapter supports three operations. There is a Dequeue operation that polls for incoming messages from a queue, an Enqueue operation that puts outgoing messages on a queue, and an Enqueue/Dequeue operation that puts outgoing messages on a queue and expects response messages on a queue. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type:

- Dequeue
- Enqueue
- Enqueue/Dequeue

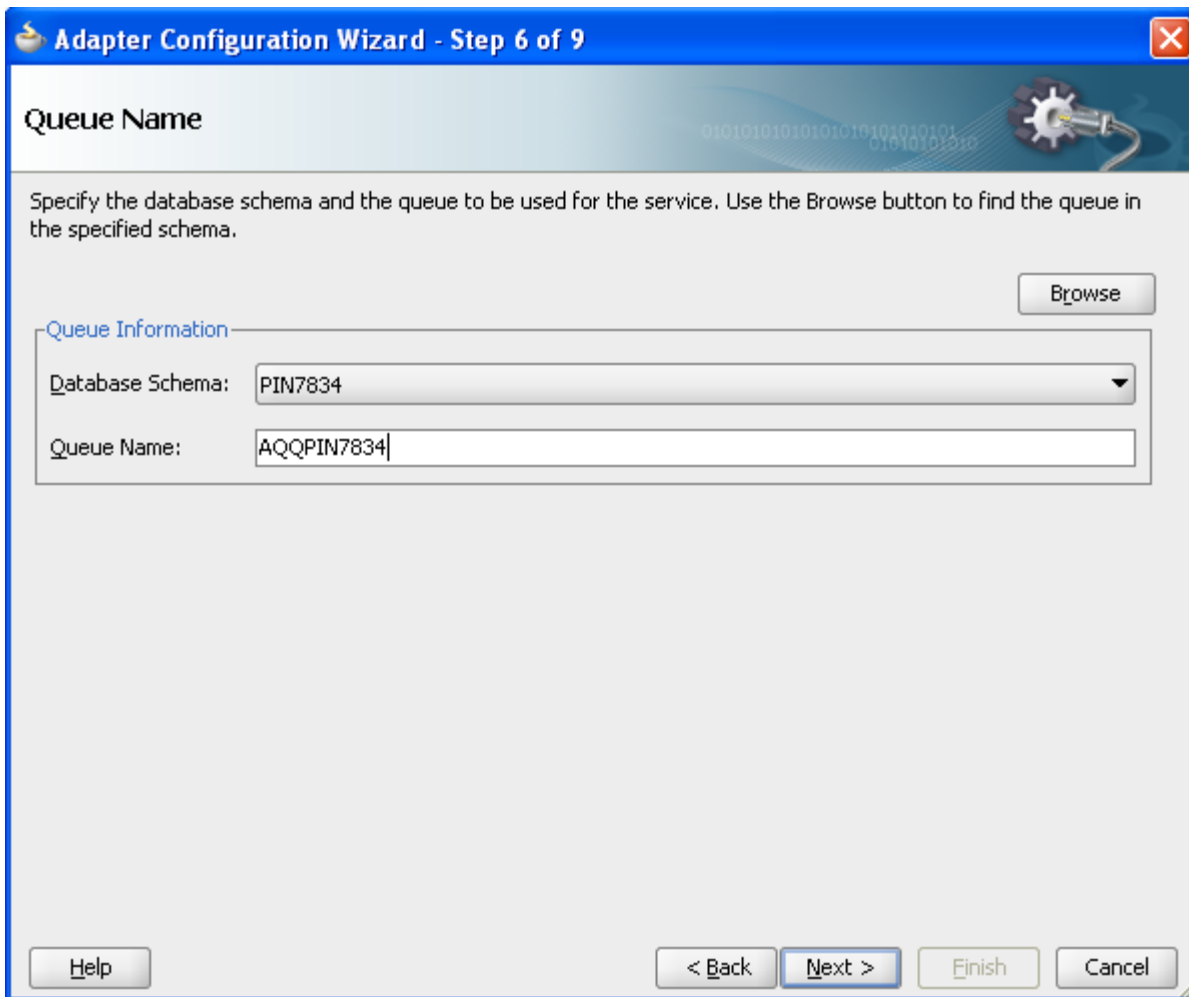
Operation Name:

Help < Back **Next >** Finish Cancel

7. Click **Next** to display Step 6 (Queue Name).

Go to the **Database Schema** field and select the additional BRM Instance's database schema, and then enter the queue name configured for this BRM instance, as shown in [Figure F-13](#).

Figure F-13 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-14.

Figure F–14 Adapter Configuration Wizard - Step 7 of 9

Adapter Configuration Wizard - Step 7 of 9

Queue Parameters

Specify parameters for the dequeue operation.

Correlation Id:

Dequeue Condition

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–15](#).

Figure F-15 Adapter Configuration Wizard - Step 8 of 9

Object Payload

The queue, PIN7834.AQQPIN7834, contains a structured object payload (ADT) named PIN7834.PIN_EVENT_TY. The business payload may be based on the whole object or a single field in the object. If the payload is the whole object, then the message schema will be automatically generated. In the case of single field, you must specify the field that contains the payload and subsequently also provide the message schema definition.

Business Payload: Whole Object PIN_EVENT_TY
 Field within the Object

Payload Field Options

Field Name:

Access to non-payload fields also needed

Validate Payload

Buttons: Help, < Back, Next >, Finish, Cancel

10. Click **Next** to display Step 9 (Finish).

Click **Finish** to create the AQ Adapter service.

You must now:

- 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 SyncDiscountBRMVCommsReqABCServiceImpl web service.
 - Add a routing rule from Oracle Mediator to SyncDiscountBRMCommsReqABCServiceImpl web service for SyncDiscount operation.
1. In **composite.xml**, provide the port and location information from concrete WSDL of the SyncDiscountBRMCommsReqABCServiceImpl web service.

```
<reference name="SyncDiscountBRMCommsReqABCServiceImpl"
  ui:wSDLLocation="oramds:/apps/AIAMetaData/AIAComponents/ApplicationConnectorServiceLibrary/BRM/V1/RequesterABCS/SyncDiscountBRMCommsReqABCServiceImpl.wsdl">
  <interface.wSDL
  interface="http://xmlns.oracle.com/ABCServiceImpl/BRM/Industry/Comms/SyncDiscountBRMCommsReqABCServiceImpl/V1#wsdl.interface(SyncDiscountBRMReqABCServiceImpl)" />
</binding.ws
```

```
port="http://xmlns.oracle.com/ABCServiceImpl/BRM/Industry/Comms/SyncDiscountBRMCommsReqABCServiceImpl/V1#wSDL.endpoint(SyncDiscountBRMReqABCServiceImpl/SyncDiscountBRMReqABCServiceImpl)" location="http://<host name>:<port>/soa-infra/services/default/SyncDiscountBRMCommsReqABCServiceImpl/SyncDiscountBRMCommsReqABCServiceImpl?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 BRM instance.

Creating a New Consumer for Customer Updates

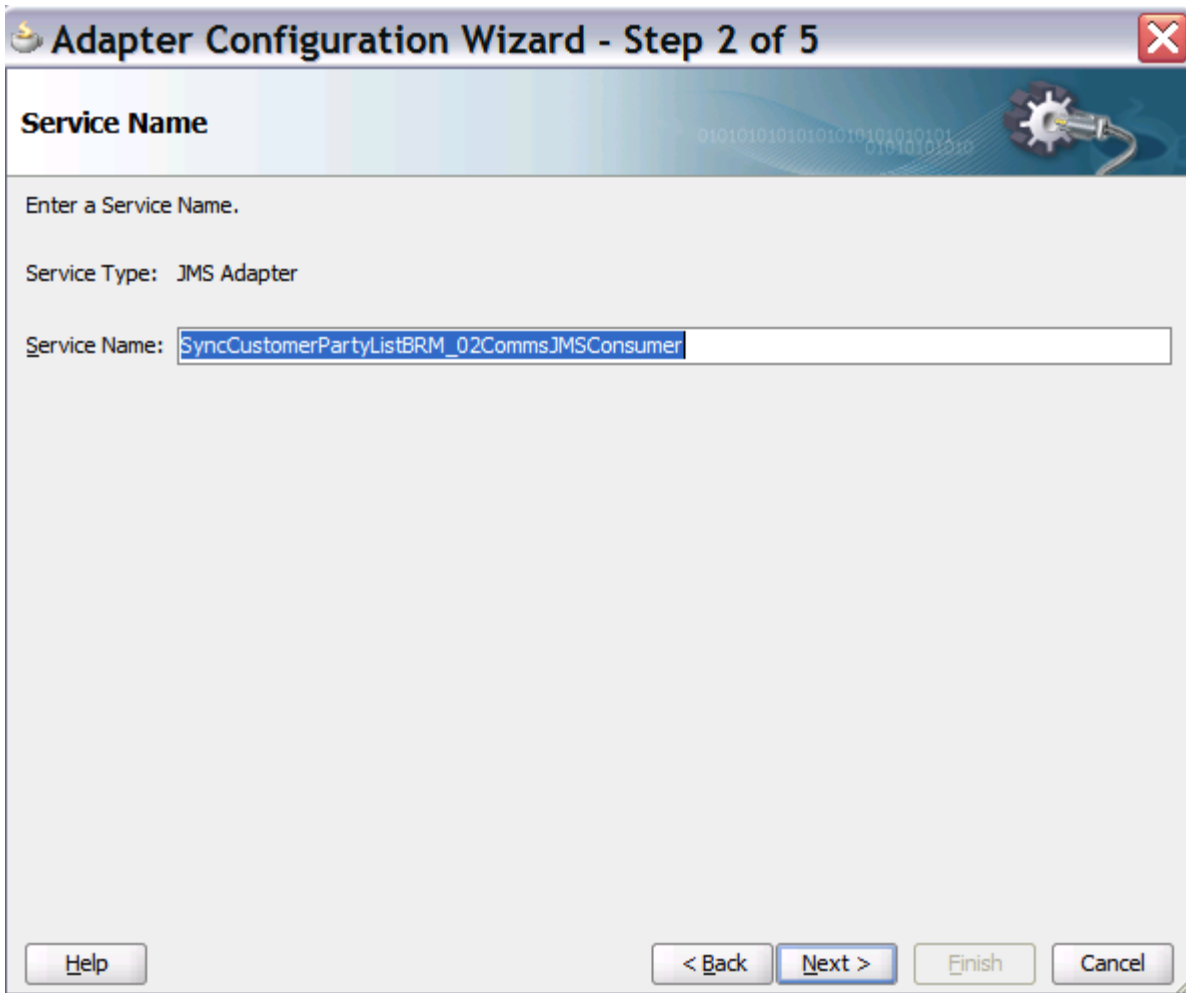
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for customer updates:

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 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.
3. Click **Next** to display Step 2 (Service Name).

Go to the **Service Name** field and enter `SyncCustomerPartyListBRM_02CommsJMSConsumer`, as shown in [Figure F-16](#).

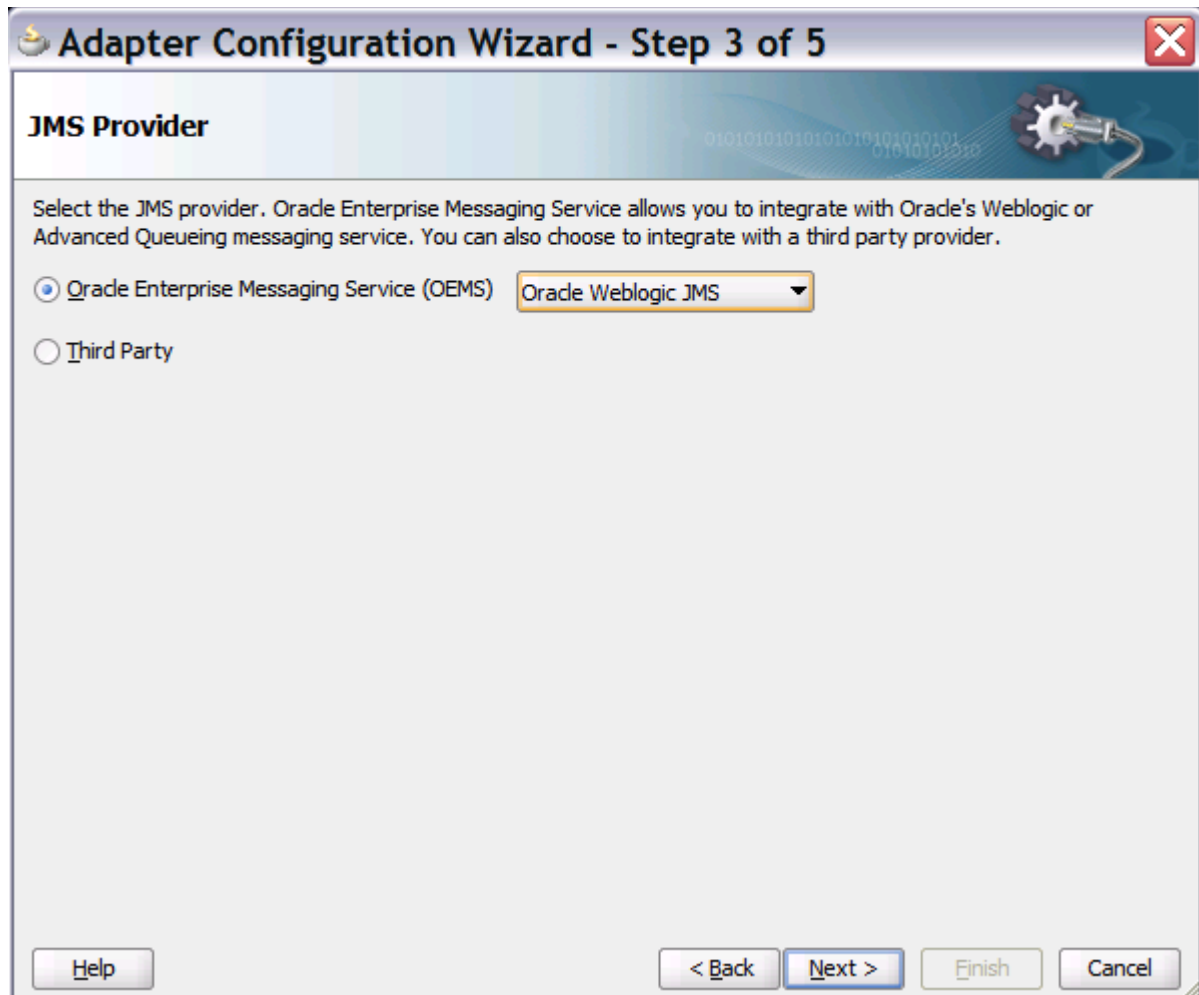
Figure F-16 Adapter Configuration Wizard - Step 2 of 5



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-17](#).

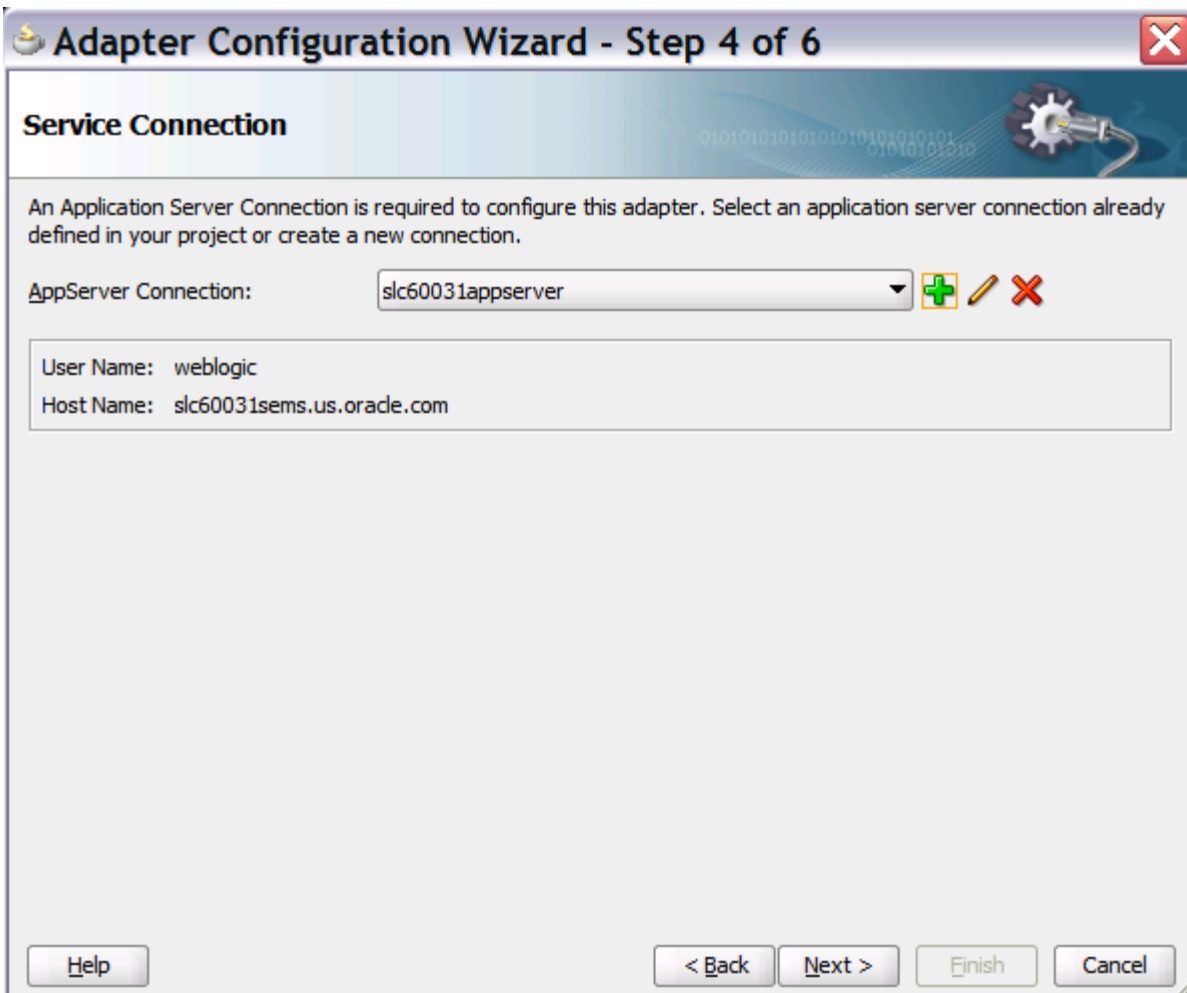
Figure F-17 Adapter Configuration Wizard - Step 3 of 5



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 drop down, as shown in [Figure F-18](#).

Figure F-18 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-19](#).

Figure F-19 Adapter Configuration Wizard - Step 5 of 9

Adapter Interface

The adapter interface is defined by a wsdl that is generated using the operation name and schema(s) specified later in this wizard. Optionally, the adapter interface may be defined by importing an existing WSDL.

Interface: Define from operation and schema (specified later)

Import an existing WSDL

WSDL URL:

Port Type:

Operation:

Callback Port Type:

Callback Operation:

Help < Back Next > Finish Cancel

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-20](#).

Figure F-20 Adapter Configuration Wizard - Step 6 of 9

Operation

The JMS Adapter supports three operation types. There is a Consume Message operation that polls for incoming messages from a JMS destination, a Produce Message operation that puts outgoing messages to a JMS destination, and a Request/Reply operation that requests for incoming messages from a JMS destination and replies by putting messages to a JMS destination either synchronously or asynchronously. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type: Consume Message
 Produce Message
 Request/Reply

Operation Name:

Help < Back Next > Finish Cancel

8. Click **Next** to display Step 7 (Consume Operation Parameters), as shown in [Figure F-21](#).

Figure F-21 Adapter Configuration Wizard - Step 7 of 9

Adapter Configuration Wizard - Step 7 of 9

Consume Operation Parameters

Enter the parameters for the Consume Message operation.

Destination Name (Topic):

Message Body Type:

Message Selector:
example 1: "country in ('US', 'UK')", example 2: "origin = 'FR'"

Use MessageListener:

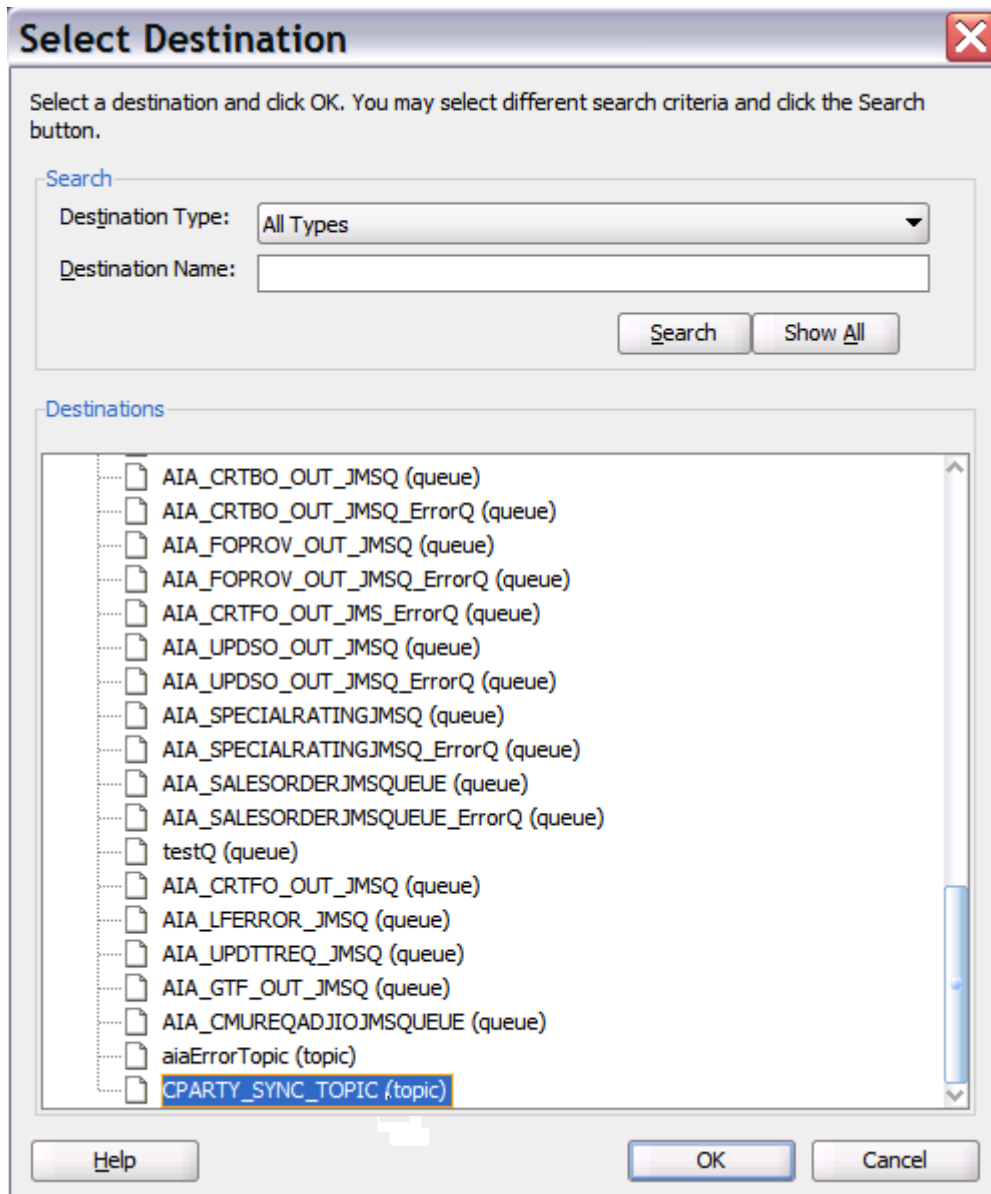
Durable Subscriber ID:

Specify the JNDI name for the JMS Connection. The deployment descriptor for the deployed instance of the JMS Adapter must associate this JNDI name with a set of configuration properties needed by the JMS Adapter to access the JMS destination at runtime.

JNDI Name:

Go to the **Destination Name (Topic)** field and click **Browse** to select the required topic, as shown in [Figure F-22](#). Click **OK** to return to Step 7.

Figure F-22 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-23.

Figure F-23 Adapter Configuration Wizard - Step 8 of 9

10. Click **Next** to display Step 9 (Finish).

Click **Finish** to create the JMS Adapter service.

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 SyncCustomerPartyListBRMCommsProvABCImpl.

The filter expression should be like this:

```
xref:lookupXRef('oramds:/apps/AIAMetaData/xref/CUSTOMERPARTY_ACCOUNTID.xref','COMMON',$in.body/impl:SyncCustomerPartyListEBM/impl:DataArea/impl:SyncCustomerPartyList/impl:CustomerPartyAccount/ns4:Identification/ns4:BusinessComponentID,'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.

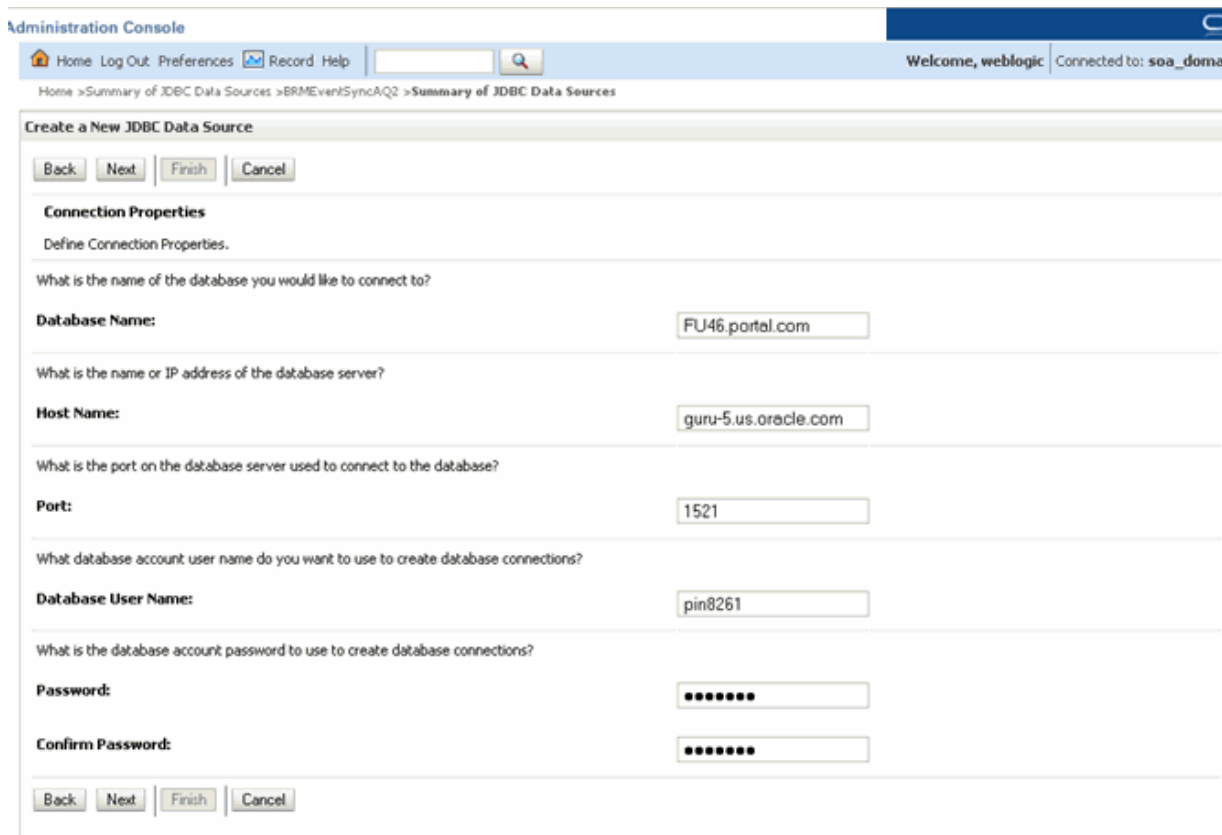
Creating a Data Source and Connection Factory

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-24](#).

Figure F-24 WebLogic Server - Administration Console - Connection Properties



Enter all the BRM connection properties and then

4. Click **Next** to display the Test Configuration page, as shown in [Figure F-25](#).

Figure F–25 WebLogic Server - Administration Console - Test Configuration

Administration Console

Home Log Out Preferences Record Help Welcome, weblogic Connected to: soa_domain

Home > Summary of JDBC Data Sources > BRMEventSyncAQ2 > Summary of JDBC Data Sources

Create a New JDBC Data Source

Test Configuration Back Next Finish Cancel

Test Database Connection

Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool?
(Note that this driver class must be in the classpath of any server to which it is deployed.)

Driver Class Name: oracle.jdbc.xa.client.OracleThinDriver

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

URL: jdbc:oracle:thin:@guru-t

What database account user name do you want to use to create database connections?

Database User Name: pin8261

What is the database account password to use to create database connections?
(Note: for secure password management, enter the password in the Password field instead of the Properties field below)

Password: [REDACTED]

Confirm Password: [REDACTED]

What are the properties to pass to the JDBC driver when creating database connections?

Properties:
user=pin8261

The set of driver properties whose values are derived at runtime from the named system property.

System Properties:

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 BRM instance.

Creating Logical Instances in Oracle AIA

Whenever the product is synchronized to Oracle AIA (through the product lifecycle management (PLM) flows, 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 Figure F-26 must be added or changed accordingly by the value given by BRM for each instance. For example, for the second BRM instance an entry must be added as shown in Figure F-26 in the AIA_SYSTEM table.

Figure F-26 Logical Instance Example

BRM_02	BRM_02	BRM Instance 01	gill.us.oracle.com	http://gill.us.oracle.com:12815	BRM	BRM
--------	--------	-----------------	--------------------	---------------------------------	-----	-----

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 BRM instance:". For example, in Figure F-26, 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 BRM instance.

Creating Service Bundles in Siebel CRM

Currently, in Typical and Reserved topologies, OSM uses the following configuration to stamp the instances. This can be changed or configured in OSM according to customer requirements so you must consult your OSM administrator before configuring the instances.

- OSMCFS_01 - 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 - OSM provisioning fulfillment for voip (Reserved Topology)
- OSMPROV_02 - OSM provisioning fulfillment for broadband US (Reserved and Typical Topology)
- OSMPROV_03 - OSM provisioning fulfillment for broadband UK (Typical Topology)
- OSMPROV_04 - 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
BRM_03 - Billing for voip
OSMCFS_01 - OSM Central Fulfillment
OSMPROV_02 - OSM provisioning fulfillment for broadband US
OSMPROV_03 - 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 - OSM Central Fulfillment
OSMPROV_01 - OSM provisioning fulfillment for voip
OSMPROV_02 - 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 - OSM Central Fulfillment
OSMPROV_04 - OSM provisioning fulfillment for voip and broadband

Merging Logical BRM Instances into a Single BRM Instance

In this case the change is related to combining multiple system instances into one.

If, for example, you start with two logical 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 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>
```

Changing the BRM Instance

This appendix provides information about how to change the Oracle Communications Billing and Revenue Management (BRM) instance after installation.

Changing the BRM Instance

Many situations occur when the BRM instance that Oracle Application Integration Architecture (Oracle AIA) points to must be changed post installation. These include:

- Moving to a new BRM server due to replacement of hardware.
- Switching from a Test instance to a Production instance

Caution: Before switching from one 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.

Oracle AIA and BRM communication happens through two adapters: inbound to Oracle AIA through Oracle Advanced Queuing (AQ) Adapter and inbound to BRM through BRM JCA Adapter. If a change occurs in the BRM instance, then the connection factories for both of these adapters must be changed.

To change the BRM instance:

1. Update connection parameters for the eis/BRM and any custom-created 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 BRM Event AQ queue name or the BRM schema name for the AQ Queue (or both) are changed, then replace occurrences of the old Event AQ queue name or the BRM schema name (or both) with the new names from <AIA_

HOME>/services/industry/Communications/BRM/AdapterServices/SyncProductInfoChangeBRMAQ.

4. Redeploy the services.

Caution: The same changes must be incorporated to any custom connection factories or datasources, or composite services.

Expectations from an Order Management System for Billing Integration

This appendix provides a summary of the general billing integration expectations from an order management system for billing integration (the Synchronize Fulfillment Order Accounts and Bill Fulfillment Order business flows).

For feature-specific expectations, see the respective flow feature sections (for example, see "[Understanding the Bill Fulfillment Order Business Flow](#)" for two-phase billing, and time-based offers).

Oracle Communications Order and Service Management (OSM) and OSM AIA Cartridges meet these documented feature specific expectations and the general expectations listed here. If you are using an order management system other than OSM, it must comply with all of these expectations.

The expectations from an order management system are as follows:

- For Account Synchronization actions:
 - The order management system can either call `CommsProcessFulfillmentOrderBillingAccountListEBF` directly to process the account synchronization message, or it can send 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.
 - To handle Oracle Communications Billing and Revenue Management (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 order management 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 order management system can optionally filter lines based on this.
- For Initiate Billing or Fulfill Billing actions:

-
- The order management system can either call `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` directly to interface the order to BRM, or it can send the Order Interface message to `AIA_CRTBO_OUT_JMSQ` SAF Queue. The Consumer (`ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer`) picks up the message and sends it to `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` to interface the order to BRM.
 - The order management 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 customizable product lines); therefore, the order management system can optionally filter them out. The target system ID must be stamped on the payload sent to the service.
 - The order management 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 order management system must interface MOVE-ADD lines with the corresponding MOVE-DELETE lines (linked using related line ID).
 - The order management system must interface the one-time charge lines tied to service bundle lines with the service bundles (linked using related line ID).
 - The order management system must interface promotion penalty charges with the promotion line (linked using related line ID).
 - For Update Sales Order actions:
 - The order management system can either call `UpdateSalesOrderSiebelCommsProvABCImpl` directly to update the sales order status in Siebel CRM, or it can send the update sales order message to `AIA_UPDSO_OUT_JMSQ` SAF Queue. The Consumer (`UpdateSalesOrderOSMCFSCCommsJMSConsumer`) picks up the message and sends it to `UpdateSalesOrderSiebelCommsProvABCImpl` to update the sales order status in Siebel CRM.
 - The order management 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.
 - See "[Understanding the Update Sales Order Business Flow](#)" for more information on how the order management system can use the extended status attributes and other guidelines.

By default, Account Synchronization, Initiate Billing, and Fulfill Billing actions do not send a response back to the caller for system or business errors (OSM and the OSM AIA Cartridges do not expect such a response).

If your order management system requires a response for business errors (or for business and system errors), see "[CommsProcessFulfillmentOrderBillingAccountListEBF](#)" and "[ProcessFulfillmentOrderBillingBRMCommsProvABCImpl](#)" for information on how to achieve this.

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.

See the discussion of resequencing in Oracle Mediator in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more information about resequencer.

Queues and Flows Enabled for Sequencing

[Table I-1](#) lists the queues and flows that are enabled for sequencing.

Note: 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	Sequencing Criteria	Comments
AIA_UPDSO_OUT_JMSQ	Update order flow from OSM to Oracle AIA for Siebel CRM.	Not set	<p>Group By: Account ID mentioned in the ObjectCrossReference section of the update message(/UpdateSalesOrderEBM/EBMHeader/Sender/ObjectCrossReference/SenderObjectIdentification/AlternateObjectKey/ID[@schemeID = 'CUSTOMERPARTY_ACCOUNTID' and @schemeAgencyID = 'COMMON'])</p> <p>Order of Processing: FIFO (First in First Out).</p> <p>Composite Name: UpdateSalesOrderOSMCFSCommsJMSConsumer.</p>	<p>Note: The consumer in the Create Trouble Ticket for Order Fallout business flow is only a sample.</p> <p>The resequencer in this flow ensures that multiple updates for the same order are processed in the right sequence.</p>
AIA_CRTCUST_OUT_JMSQ	Order flow from OSM to Oracle AIA for customer data creation in billing.	Set by OSM	<p>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.</p> <p>concat(\$in.SyncCustomerPartyListEBM/ns0:SyncCustomerPartyListEBM/ns0:DataArea/ns0:SyncCustomerPartyList/ns0:CustomerPartyAccount/corecom:Identification/corecom:ApplicationObjectKey/corecom:ID[@schemeID='AccountId'], \$in.SyncCustomerPartyListEBM/ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ID)</p> <p>Order of Processing: FIFO (First in First Out).</p> <p>Composite Name: CommunicationsCustomerPartyEBSV2Resequencer.</p>	The resequencer in this flow ensures that the solution can successfully handle processing of concurrent orders for the same customer.
--	Sync customer flow from Siebel CRM system to Oracle Customer Hub.	Not Set	<p>Group By: AccountID.</p> <p>Order of Processing: FIFO (First in First Out).</p> <p>Composite Name: SyncAcctSiebelAggrEventConsumer SyncContSiebelAggrEventConsumer.</p>	<p>Also available in the Agent Assisted Billing Care pre-built integration.</p> <p>The resequencer in this flow ensures that multiple updates for the same customer are processed in the right sequence.</p>

Table I-1 (Cont.) Queues and Flows Enabled for Sequencing

Oracle AIA Queue	Flow	JMS Priority	Sequencing Criteria	Comments
AIA_CRTFO_IN_JMSQ	Order flow from Oracle AIA to OSM	Set by ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	NA
AIA_CRTBO_OUT_JMSQ	Order flow from OSM to AIA for billing.	Set by OSM	None as delivered. Customers can use ProcessFulfillmentOrderBillingOSMCFSCommsJMConsumer to implement custom sequencing.	NA
AIA_UPDBO_IN_JMSQ	Order flow from AIA (from billing) to OSM	Set by ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	None. (Onus is on OSM.)	NA
AIA_UPDCUST_IN_JMSQ	Response of the customer creation in billing from AIA to OSM	Set by ProcessFOBillingAccountListRespOSMCFSCommsJMSProducer	None. (Onus is on OSM)	NA
AIA_CRTFO_OUT_JMSQ	Create Fulfillment Order flow from OSM to Oracle AIA for the provisioning system	Set by OSM.	None as delivered. Customer can use ProcessProvisioningOrderOSMCFSCommsJMConsumer to implement custom sequencing.	NA
AIA_FOCFS_IN_JMSQ	Update Fulfillment Order flow from Oracle AIA (from the provisioning system) to OSM)	Set by ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer	None. (Onus is on OSM)	NA
AIA_FOPROV_OUT_JMSQ	Update Fulfillment Order flow from the provisioning system to Oracle AIA (for OSM)	Set by provisioning system	None as delivered. Customer can use ProcessFulfillmentOrderUpdateOSMPROVCommsJMConsumer to implement custom sequencing.	NA
AIA_FOPROV_IN_JMSQ	Create Fulfillment Order from Oracle AIA (from OSM) to the provisioning system.	Set by ProcessProvisioningOrderOSMPROVCommsJMSProducer	None. (Onus is on OSM)	NA

Resolving Errors in Flows with Resequencer

If an error occurs in the Oracle Communications Billing and Revenue Management (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 CRM 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.

See the discussion of monitoring resequenced messages in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite* for more information on unblocking and retrying.

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.

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</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" schemeAgencyId="SEBL_01">SalesOrderNumber</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgencyId="SEBL_01">SalesOrderRevision</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgencyId="COMMON">CommonAccountID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgencyId="SEBL_01">Siebel Account
ID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTNAME" schemeAgencyId="SEBL_01">Account Name</ID>
</AlternateObjectKey>
</SenderObjectIdentification>
</ObjectCrossReference>
</Sender>
</EBMHeader>
```

Only the underlined elements are required for the SalesOrder EBM.

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}:{httpportname}/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 - FULFILLMENTSYSYSTEM_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" schemeAgencyId="SEBL_01">SalesOrderNumber</ID>
    </AlternateObjectKey>
    <AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgencyId="SEBL_01">SalesOrderRevision</ID>
    </AlternateObjectKey>
    <AlternateObjectKey>
<ID schemeID="FULFILLMENTSYSYSTEM_ID "
schemeAgencyId="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.

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).

See "[Configuring Oracle AIA Processes for Error Handling and Trace Logging](#)", [Extending Error Handling and Extending Fault Messages in Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack](#) for more information on extending error handling.

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.

Configuration Properties for CAVS Enablement in 11.1

In the 11.1 Oracle AIA CAVS implementation, every service has a number of configuration properties.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on configuration properties.

For the RouteToCAVS property, the ready-to-use 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.

To navigate to this UI:

1. Log in to the AIA Console (<http://<host>:<port>/AIA>).
2. Go to **Setup**.
3. Select the **AIA Configuration** tab.

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.

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 in: **\$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. To make your changes effective. Login to the AIA Console (<http://<host>:<port>/AIA>).
5. Go to **Setup** and select **AIA Configuration** tab.
6. Click **Reload** to reload the configuration file and make your changes effective.

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 in: **\$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 reload the configuration file and make your changes effective.

Reintroducing Enterprise Business Services

This appendix provides instructions for reintroducing enterprise business services (EBSs) into the Oracle Application Integration Architecture (Oracle AIA) deployment.

Reintroducing 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 enterprise business services:

1. Go to JDeveloper 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 in: `$AIA_HOME/aia_instances/$INSTANCE_NAME/AIAMetaData/config`.

Note: Entries in the `AIAConfigurationProperties.xml` file are case sensitive.

5. To invoke new EBS connectors you need to replace the Provider connector's name and address with the EBS name and address.

This action tells the Requestor to invoke EBS instead of the Provider application business connector service (ABCS).

6. Save and close the file.
7. To make your changes effective. Login to the AIA Console (`http://<host>:<port>/AIA`).
8. Go to **Setup**, and select the **AIA Configuration** tab.

9. Click **Reload** to reload the configuration file.

Understanding Multischema Migration

This appendix discusses how Oracle Communications Billing and Revenue Management (BRM) communicates the correct account information to Siebel customer relationship management (Siebel CRM) in a multischema environment using Oracle Application Integration Architecture (Oracle AIA) integration.

About Multischema Migration

In BRM, you can distribute your data among multiple schemas in the same database to increase scalability and support load balancing. The BRM accounts and associated objects are relocated between schemas using Account Migration Manager (AMM). AMM migrates the accounts and associated objects in batches, with each batch consisting of a list of accounts identified by their BRM Portal Objects (POID), source schema number, and destination schema number.

See *Oracle Communications Billing and Revenue Management Concepts* and *Oracle Communications Billing and Revenue Management System Administrator's Guide* for more information about multischema architecture and account migration.

In order to communicate the correct account information between BRM and Siebel CRM a common AIA identifier and an Oracle AIA cross reference table is used to map accounts between BRM and Siebel CRM. The BRM schema number is included as part of the BRM cross reference, such as 0.0.0.2 /account 11599 4.

During a BRM migration, AMM populates a MIGRATED_OBJECTS_T cross-reference table in the primary BRM schema with the batch ID, and old and new POIDs of all the objects that have been migrated successfully. The old POID represents pre-migration schema value, and new POID represent the post-migration schema value.

[Table M-1](#) shows an example of the MIGRATED_OBJECTS_T table data.

Table M-1 *MIGRATED_OBJECTS_T Table Example*

BATCH_ID	OLD_POID	NEW_POID
225313	0.0.0.2 /account 11599 4	0.0.0.3 /account 11599 4
225313	0.0.0.2 /service 14569 2	0.0.0.3 /service 14569 2
225313	0.0.0.2 /billinfo 12349 3	0.0.0.3 /billinfo 12349 3
225494	0.0.0.2 /account 10319 1	0.0.0.3 /account 10319 1
2254494	0.0.0.2 /billinfo 14569 1	0.0.0.3 /billinfo 14569 1

Mapping BRM POIDs to AIA Cross-References

When a BRM migration finishes, Oracle AIA updates the Oracle AIA cross-reference table using the following BRM object account types:

- account POID
- billinfo POID
- payinfo POID
- default account level balance group POID
- purchased product POID
- purchased discount POID
- service POID (Services objects for the account)
- purchased bundle POID
- profile POID

[Table M-2](#) shows the cross-reference mapping between the Oracle AIA cross-reference identifier and BRM cross-reference object types.

Table M-2 Cross-Reference Mapping Between Oracle AIA and BRM

AIA Cross-Reference	BRM Cross-Reference
CUSTOMERPARTY_CONTACTID	/account POID
CUSTOMERPARTY_BILLPROFILEID	/billinfo POID
CUSTOMERPARTY_PAYPROFILEID	/payinfo POID
CUSTOMERPARTY_ADDRESSID	/account POID or /payinfo POID
CUSTOMERPARTY_CONTACTID	/account POID or /payinfo POID
CUSTOMERPARTY_DEFAULTBALANCEGROUPID	/balance_group POID Note: This is the default account level.
INSTALLEDPRODUCT_ID	/purchased_product POID
INSTALLEDPRODUCT_ID	/purchased_discount POID
INSTALLEDPRODUCT_ID	/service/object POID Where <i>object</i> is the service object for the account.
INSTALLEDPRODUCT_ID	/purchased_bundle POID
INSTALLEDPRODUCT_ID	/profile/object POID Where <i>object</i> is the profile object for the account.

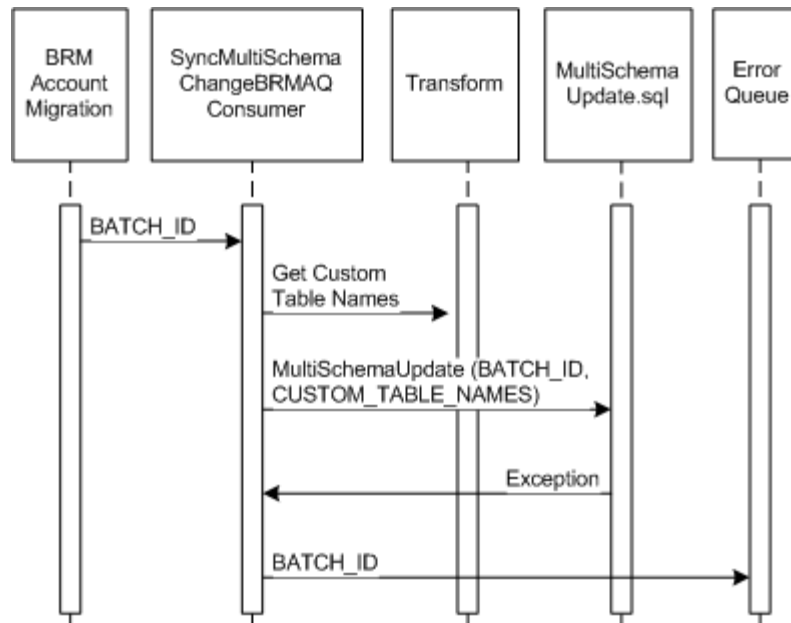
Synchronizing Cross-References for Migrated Accounts

When you migrate accounts across BRM schemas, the cross-references for the accounts are synchronized from BRM to Oracle AIA as follows:

1. When you migrate accounts across BRM schemas, AMM populates a MIGRATED_OBJECTS_T cross-reference table in the primary BRM database with the batch ID, and old and new POIDs of all the objects that have been migrated successfully.

2. When the BRM account migration is complete, an **AccountInfoChange** business event is generated and published to an Advanced Queue (AQ) database queue. The **AccountInfoChange** business event includes the batch ID and BRM object POIDs.
3. Oracle AIA retrieves the **AccountInfoChange** business event from the AQ database queue and updates the information in the Oracle AIA database by calling the SyncMultiSchemaChangeBRMAQ consumer service.
4. The SyncMultiSchemaChangeBRMAQ service reads the batch ID and BRM object POIDs from the **AccountInfoChange** business message from the AQ and passes the message to the SyncMultiSchemaChangeBRMRequester service.
5. The SyncMultiSchemaChangeBRMRequester service does the following:
 - a. Reads the **AIAConfigurationProperties.xml** file for a list of custom cross-reference tables. See "[Configuring Oracle AIA to Work with BRM in Multischema Environments](#)" for more information.
 - b. Reads the BRM_INSTANCE domain value map (DVM) to get the database link name.
 - c. Calls the MultiSchemaUpdate PLSQL procedure.
 - d. Adds a message to the error queue if an error is found. See "[Handling Errors](#)" for more information.
6. MultiSchemaUpdate PLSQL procedure updates the Oracle AIA cross-reference table by doing the following:
 - a. Retrieves the batch ID, custom table and database link names from the BRMMultiSchemaConsumer service.
 - b. Connects to MIGRATED_OBJECTS_T cross-reference table using a database link.
 - c. Locates the old POID values and updates the new POID values in the AIA cross-reference table.
 - d. Deletes the entries in MIGRATED_OBJECTS_T cross-reference table when Oracle AIA receives the data.

[Figure M-2](#) illustrates the flow when updating cross-references in a multischema migration.

Figure M-1 Updating Cross-Reference Data Sequence Diagram

Configuring Oracle AIA and BRM to Work Together in Multischema Environments

You must configure both Oracle AIA and BRM to work together in multischema environments.

Configuring Oracle AIA to Work with BRM in Multischema Environments

Before deploying the Oracle Communications Order to Cash pre-built integration, you must perform the following Oracle AIA configuration tasks:

1. Create a database link to your BRM environment so that Oracle AIA can access the BRM MIGRATED_OBJECTS_T cross-reference table by adding the following command in your Oracle AIA database (adjust to match your environment):

```

create database link brm01 connect to brm identified by brm using
' (DESCRIPTION =
  (ADDRESS =
    (PROTOCOL = TCP)
    (Host = ap6060fems.us.oracle.com)
    (Port = 1561)
  )
  (CONNECT_DATA = (SID = aiafpdev)
)';
  
```

2. Configure the cross-reference table. Oracle recommends that you split large cross-reference tables into separate tables for each BRM cross-reference object using the XREF Migration Utility. When you split the cross-reference table, the XREF Migration Utility creates custom cross-reference tables. You must add the names of these tables to the **AIAConfigurationProperties.xml** file for Oracle AIA to update the BRM cross-reference objects after migrating accounts across BRM schemas. Configure the table as follows:
 - a. Using the XREF Migration Utility, split the cross-reference table into separate tables for each BRM cross-reference object. For more information about using this utility, see the Oracle Technology Network web site at:

<http://www.oracle.com/technetwork/middleware/foundation-pack/learnmore/aiaref-524690.html>

For BRM cross-reference object types, see "[Mapping BRM POIDs to AIA Cross-References](#)".

- b. Open `$AIA_INSTANCES/config/AIAConfigurationProperties.xml` file in a text editor.
- c. Add a comma-separated list of the names of your custom cross reference tables.
- d. Save and close the file.

Configuring BRM to Work with Oracle AIA in Multischema Environments

To configure BRM to work with Oracle AIA in multischema environments, perform the following configuration tasks:

1. Configure the Account Migration Manager to populate the MIGRATED_OBJECTS_T cross-reference table as follows:
 - a. Open the `PIN_HOME/sys/amt/Infranet.properties` file in a text editor.
 - b. Locate the `publish_migrated_objects` entry and add the storable classes that Oracle AIA cross-references in comma-separated format. For example:

```
publish_migrated_objects=/billinfo,/service,/purchased_
product,/payinfo,/balance_group,/purchased_bundle
```

See the list of BRM cross-reference object types in [Table M-2, "Cross-Reference Mapping Between Oracle AIA and BRM"](#) for the storable classes to add.

- c. Save and close the file.
2. Configure the Enterprise Application Integration (EAI) Manager and the Synchronization Queue Data Manager (DM) to publish the AccountInfoChange business event to AQ.

For more information, see:

- The discussion of integrating BRM with enterprise applications and configuring event notification for EAI Manager in *Oracle Communications Billing and Revenue Management Developer's Guide*
- The discussion of specifying which business events to send to the database queue in *Oracle Communications Billing and Revenue Management Synchronization Queue Manager*

Note: The integration includes the `payloadconfig_crm_sync.xml` payload configuration file, which contains the AccountInfoChange business event.

Handling Errors

When an exception is raised the BRMMultiSchemaConsumer service will add a message into the error queue for manual intervention. See "[Handling Error Notifications for Order Fallout Management](#)" for more information.

Assumptions and Constraints for Multischema Migration

The assumptions and constraints for multischema migration are as follows

- AIA cross reference data should be updated after migration. AIA does not support the processing of orders while a migration is ongoing.
- Account migration should be planned and scheduled at a time when there is no or very little user activity on the system. Any orders flowing through the system during migration will result in order fallout and will need to be reprocessed after the AIA cross references have been synchronized.

Integration Services

The following services enable the integration of multischema migration:

- SyncMultiSchemaChangeBRMAQ Consumer
- SyncMultiSchemaChangeBRMRequesterImpl
- MultiSchemaUpdate.sql

About the SyncMultiSchemaChangeBRMAQ Consumer

The SyncMultiSchemaChangeBRMAQ consumer is a mediator-based service called by the BRM AQ after a multischema migration. It reads the BATCH_ID and BRM instance ID in the AccountInfoChange message from the AQ and passes it to the SyncMultiSchemaChangeBRMRequester service.

About the SyncMultiSchemaChangeBRMRequester Service

The SyncMultiSchemaChangeBRMRequester service is a BPEL service that accepts the AccountInfoChange message from the SyncMultiSchemaChangeBRMAQ consumer and invokes the **MultiSchemaUpdate.sql** process. It performs the following actions:

1. Reads the **AIAConfigurationProperties.xml** file for a list of custom cross reference tables. See "[Configuring Oracle AIA to Work with BRM in Multischema Environments](#)" for more information.
2. Reads the **BRM_INSTANCE.dvm** to get the database link name
3. Invokes **MultiSchemaUpdate.sql**
4. Adds a message to the error queue if an error is found. See "[Handling Error Notifications for Order Fallout Management](#)" for more information.

About the MultiSchemaUpdate.sql Process

MultiSchemaUpdate.sql is a PLSQL process which is called by the BRMMultiConsumer service after a multischema migration to update the Oracle AIA cross reference table. It performs the following actions:

1. Retrieves the BATCH_ID and CUSTOM_TABLE_NAMES
2. Connects to the BRM MIGRATED_OBJECTS_T cross-reference table using a database link.
3. Locates the old POID values and updates the new POID values in the Oracle AIA cross reference table