

# Oracle® Application Integration Architecture

## Oracle Communications Order to Cash Integration Pack Implementation Guide



Release 12.3.2

F92298-01

June 2024

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered within a solid red square.

ORACLE®

F92298-01

Copyright © 2011, 2024, Oracle and/or its affiliates.

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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

|                             |     |
|-----------------------------|-----|
| Audience                    | xix |
| Documentation Accessibility | xix |
| Diversity and Inclusion     | xix |

## 1 Overview of the Oracle Communications Order to Cash Integration Pack

---

|  |      |
|--|------|
| About Oracle AIA and Oracle Communications Solutions Integrations    | 1-1  |
| Overview of the Oracle Communications Order to Cash Integration Pack | 1-1  |
| About Using Third-Party Applications                                 | 1-3  |
| About the Test Orchestration Process                                 | 1-3  |
| About Oracle AIA in Product-Based Solutions                          | 1-3  |
| About RODOD  | 1-3  |
| About RSDOD  | 1-3  |
| About Integrated RODOD and RSDOD Deployments                         | 1-4  |
| About the Process Integrations                                       | 1-5  |
| About Product Lifecycle Management                                   | 1-5  |
| Pricing Center and Pricing Design Center Terminology                 | 1-6  |
| About Order Lifecycle Management                                     | 1-7  |
| About Customer Management  | 1-7  |
| About Order Fallout Management                                       | 1-8  |
| Example Order to Cash Business Scenarios                             | 1-8  |
| Promotion Groups and Large Order Scenario                            | 1-9  |
| Design Time  | 1-9  |
| Order Time   | 1-9  |
| More Information About Large Orders and Promotion Groups             | 1-10 |
| Family Share Plan and Payment on Order Scenario                      | 1-10 |
| Design Time  | 1-10 |
| Order Time   | 1-11 |
| More Information About Family Share Plans and Payments on Orders     | 1-11 |
| Corporate Account Hierarchy Scenario and Split Billing Scenario      | 1-12 |
| Design Time  | 1-12 |
| Order Time   | 1-12 |
| More Information About Corporate Hierarchies and Split Billing       | 1-13 |

|   |      |
|---|------|
| Legal Owner and Split Billing Scenario                | 1-13 |
| Design Time   | 1-14 |
| Order Time  | 1-14 |
| More Information About Legal Owners and Split Billing | 1-14 |
| Directory Placeholders Used in This Guide             | 1-15 |

## Part I Understanding the Process Integrations

---

### 2 Understanding the Process Integration for Product Lifecycle Management

---

|  |     |
|--|-----|
| About 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

---

|   |      |
|---|------|
| About the Synchronize Product and Price Business Flow         | 3-1  |
| About Synchronizing Pricing Components                        | 3-1  |
| About Synchronizing Updates to Pricing Components             | 3-2  |
| About the Pricing Component Synchronization Integration Flows | 3-3  |
| About the Product Synchronization Integration Flow            | 3-3  |
| About Unspecified Effective Start and End Dates               | 3-6  |
| About the Discount Synchronization Integration Flow           | 3-6  |
| About the Chargeshare Synchronization Integration Flow        | 3-8  |
| About Synchronizing Product Usage Charges                     | 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 Price Lists                | 3-12 |
| Rate Plan Example: Multiple Lines of Business and Regions     | 3-13 |
| Offering a Product in Multiple Currencies                     | 3-14 |
| Rate Plan Example: Multiple Currencies                        | 3-14 |
| Managing Pricing in Rate Plans and Price Lists                | 3-15 |
| Changing the Price of a Product                               | 3-15 |
| Changing the Price List of a Product                          | 3-16 |
| Changing from Multiple Price Lists to the Default Price List  | 3-17 |
| About BRM Balance Groups                                      | 3-19 |
| Understanding Product Bundling                                | 3-19 |
| Basic Entity Mappings   | 3-19 |
| Defining Products and Discounts in BRM                        | 3-20 |
| Using Fixed Amounts or Scaled Amounts in BRM                  | 3-21 |
| About Creating Physical Goods                                 | 3-22 |
| About Adding Products to Siebel CRM Sales Catalogs            | 3-22 |

|   |      |
|---|------|
| Recommendations for Product Definition in Siebel CRM                            | 3-22 |
| Recommendation for Discounts  | 3-23 |
| Discounts Defined in BRM  | 3-23 |
| Discounts Defined in Siebel CRM   | 3-23 |
| Defining Overrides on the Product Definition                                    | 3-23 |
| About Service Bundles   | 3-24 |
| About Billing Service Types for Service Bundles                                 | 3-25 |
| About Simple Service Bundles  | 3-25 |
| Flagging Subscription Products as Simple Service Bundles                        | 3-26 |
| Example of Using a Service Bundle or a Simple Service Bundle                    | 3-26 |
| Assumptions and Constraints for Working with Simple Service Bundles             | 3-27 |
| About Service Grouping  | 3-28 |
| Enabling Service Grouping at Design Time  | 3-28 |
| About Promotion Groups  | 3-29 |
| Creating Promotion Groups   | 3-29 |
| About Family Share Plans  | 3-30 |
| Creating Family Share Plans   | 3-31 |
| About Marketing Bundles   | 3-33 |
| About Credit Limits   | 3-34 |
| About One-Time Charges for Activating and Changing Services                     | 3-34 |
| Charging for Service Activation and Changes to Promotions and Services          | 3-34 |
| About Friends and Family Lists  | 3-35 |
| Enabling Friends and Family Lists   | 3-36 |
| Product Definition Methodology for Friends and Family Lists: Example            | 3-36 |
| About Time-Based Offerings  | 3-40 |
| Setting Up Time-Based Offerings   | 3-40 |
| Managing Expired Time-Based Offerings   | 3-42 |
| Assumptions and Constraints for the Synchronize Product and Price Business Flow | 3-42 |

## 4 Implementing the Synchronize Product and Price Business Flow

---

|  |     |
|--|-----|
| About the Synchronize Product and Price Business Flow  | 4-1 |
| About the Product Synchronization Integration Flow     | 4-1 |
| About the Discount Synchronization Integration Flow    | 4-3 |
| About the Chargeshare Synchronization Integration Flow | 4-5 |
| BRM Interfaces   | 4-6 |
| Siebel CRM Interfaces                                  | 4-6 |
| Industry Oracle AIA Components                         | 4-6 |
| Integration Services                                   | 4-7 |
| SyncProductBRMCommsReqABCImpl                          | 4-7 |
| SyncDiscountBRMCommsReqABCImpl                         | 4-8 |
| SyncSponsorshipBRMCommsReqABCImpl                      | 4-8 |

|   |     |
|---|-----|
| CommunicationsItemCompositionEBSV1                      | 4-8 |
| SyncItemCompositionListSiebelCommsJMSConsumer           | 4-8 |
| SyncItemCompositionListSiebelCommsProvABCImpl           | 4-8 |
| CommunicationsPriceListEBSV2                            | 4-9 |
| ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer | 4-9 |
| ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl | 4-9 |

## 5 Understanding the Query Product Classes Business Flow

---

|  |     |
|--|-----|
| About the Query Product Classes Business Flow    | 5-1 |
| About the Query Product Classes Integration Flow | 5-4 |
| About the Logical Data Model in Siebel CRM       | 5-5 |
| About Updating the Attribute Valueset            | 5-6 |
| Updating Product Classes and Effectivity         | 5-6 |

## 6 Implementing the Query Product Classes Business Flow

---

|  |     |
|--|-----|
| About the Query Product Classes Business Flow        | 6-1 |
| About the Query Product Classes Integration Flow     | 6-1 |
| Siebel CRM Interfaces                                | 6-3 |
| Industry Oracle AIA Components                       | 6-3 |
| Integration Services                                 | 6-4 |
| QueryProdClassAndAttributesSCEReqCommsABCImpl        | 6-4 |
| QueryClassificationListSiebelCommsProvABCImpl        | 6-5 |
| QuerySpecificationListSiebelCommsProvABCImpl         | 6-5 |
| QuerySpecificationValueSetListSiebelCommsProvABCImpl | 6-5 |

## 7 Understanding the Process Integration for Order Lifecycle Management

---

|  |      |
|--|------|
| About the Process Integration for Order Lifecycle Management | 7-1  |
| About Order Lifecycle Attributes                             | 7-4  |
| About the OrderSubject Attribute                             | 7-4  |
| About the ServiceFamily Attribute                            | 7-6  |
| About the TechnicalInventoryId Attribute                     | 7-6  |
| About the Order Lifecycle Management Business Flows          | 7-7  |
| About Order Capture  | 7-8  |
| About the Deliver Customer Order Subflow                     | 7-9  |
| About the Qualify Customer Order Subflow                     | 7-11 |
| Product Definition and Mapping Design Considerations         | 7-12 |
| About Defining Products                                      | 7-12 |
| About Mapping Orders to Fulfillment Functions                | 7-13 |

## 8 Understanding the Process Sales Order Fulfillment Business Flow

---

|   |      |
|---|------|
| About the Process Sales Order Fulfillment Business Flow                           | 8-1  |
| About Sales Orders  | 8-1  |
| About Change Orders   | 8-2  |
| About Supplemental Orders   | 8-2  |
| About Follow-On Orders  | 8-3  |
| About Modify Orders   | 8-3  |
| About Future-Dated Orders   | 8-3  |
| Supporting Large Orders   | 8-4  |
| Configuring Extra Large Order Support   | 8-6  |
| About Accepting Payments on Orders  | 8-6  |
| Accepting Payments on Change Orders   | 8-6  |
| Supporting Multiple Price Lists on Orders   | 8-7  |
| Specifying Different Price Lists on New Orders                                    | 8-7  |
| Changing Price Lists on Supplemental Orders                                       | 8-7  |
| Changing Price Lists on Modify Orders   | 8-8  |
| Supporting Promotion Groups on Orders   | 8-9  |
| Using Change Orders with Promotion Groups   | 8-10 |
| Supporting Family Share Plans on Orders   | 8-11 |
| Using Change Orders with Family Share Plans                                       | 8-12 |
| Supporting Order Priority   | 8-12 |
| About Order Priority in Siebel CRM  | 8-12 |
| Supporting Split Billing on Orders  | 8-13 |
| Supporting Corporate Account Hierarchies on Orders                                | 8-14 |
| Supporting Legal Owners on Orders   | 8-14 |
| Supporting Legal Owners and Split Billing On Orders                               | 8-15 |
| Assumptions and Constraints for the Process Sales Order Fulfillment Business Flow | 8-15 |

## 9 Implementing the Process Sales Order Fulfillment Business Flow

---

|   |     |
|---|-----|
| About the Process Sales Order Fulfillment Business Flow             | 9-1 |
| About the Submitting Orders from Siebel CRM to OSM Integration Flow | 9-1 |
| Defining Transaction Boundaries and Recovery Details                | 9-3 |
| Supporting Order Priority   | 9-4 |
| Supporting Price Lists  | 9-5 |
| Siebel CRM Interfaces   | 9-5 |
| Industry Oracle AIA Components                                      | 9-5 |
| Integration Services  | 9-6 |
| ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer                  | 9-6 |

|  |     |
|--|-----|
| ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl  | 9-6 |
| ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer | 9-7 |

## 10 Understanding the Synchronize Fulfillment Order Billing Account Business Flow

---

|   |      |
|---|------|
| About the Synchronize Fulfillment Order Billing Account Business Flow                           | 10-1 |
| About the Interfacing Orders to Create Customer Data Integration Flow                           | 10-2 |
| About Creating Account, Billing, and Corporate Hierarchies in BRM                               | 10-3 |
| About the Payment on Order Integration Flow   | 10-3 |
| Assumptions and Constraints for the Synchronize Fulfillment Order Billing Account Business Flow | 10-3 |

## 11 Implementing the Synchronize Fulfillment Order Billing Account Business Flow

---

|   |       |
|---|-------|
| About the Synchronize Fulfillment Order Billing Account Business Flow   | 11-1  |
| About the Interfacing Orders to Create Customer Data Integration Flow   | 11-1  |
| About the Payment on Order Integration Flow                             | 11-6  |
| Defining Transaction Boundaries and Recovery Details                    | 11-7  |
| BRM Interfaces  | 11-9  |
| Siebel CRM Interfaces   | 11-10 |
| Industry Oracle AIA Components  | 11-10 |
| Integration Services  | 11-10 |
| ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSSConsumer        | 11-11 |
| ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSSConsumer        | 11-11 |
| CommsProcessFulfillmentOrderBillingAccountListEBF                       | 11-11 |
| QueryReceivedPaymentListSiebelCommsProvABCImpl                          | 11-12 |
| CreateReceivedPaymentListBRMCommsProvABCImpl                            | 11-12 |
| ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer | 11-13 |

## 12 Understanding the Bill Fulfillment Order Business Flow

---

|  |      |
|--|------|
| Overview of the Bill Fulfillment Order Business Flow | 12-1 |
| About Interfacing Orders to BRM                      | 12-1 |
| Creating and Updating Service Instances              | 12-1 |
| Sending Pricing Information                          | 12-2 |
| Applying One-time and Penalty Charges                | 12-2 |
| Applying Pricing or Discount Overrides               | 12-2 |
| Sending Price List Information                       | 12-3 |
| Using Service Identifiers                            | 12-3 |
| Communicating Promotion Information                  | 12-3 |



|   |       |
|---|-------|
| Rolling Back Transactions   | 12-4  |
| Supporting Balance Groups   | 12-4  |
| Disabling Service-Level Balance Groups  | 12-4  |
| Working with Service-Level Balance Groups Enabled                             | 12-5  |
| About Tracking and Billing Services with Service-Level Balance Groups Enabled | 12-5  |
| About Transferring Services in Siebel CRM                                     | 12-9  |
| Working with Service-Level Balance Groups Disabled                            | 12-10 |
| Supporting Product Bundling   | 12-11 |
| Examples of Mapping for Bundled Products                                      | 12-11 |
| Synchronizing Simple Service Bundles  | 12-13 |
| Changing Purchased Simple Service Bundles                                     | 12-13 |
| Simple Service Bundle Cross-References Entries                                | 12-14 |
| Synchronizing Promotion Groups  | 12-14 |
| Synchronizing Family Share Plans  | 12-15 |
| Synchronizing Service Grouping  | 12-16 |
| Supporting Single-Phase and Two-Phase Billing                                 | 12-16 |
| Choosing Between Single-Phase and Two-Phase Billing                           | 12-17 |
| Single-Phase Billing  | 12-17 |
| Two-Phase Billing   | 12-17 |
| Using Single-Phase Billing or Two-Phase Billing                               | 12-18 |
| INITIATE BILLING Mode   | 12-18 |
| FULFILL BILLING Mode  | 12-20 |
| Assumptions and Constraints for Two-Phase Billing                             | 12-20 |
| Supporting Revisions  | 12-21 |
| Assumptions and Constraints for Revisions                                     | 12-22 |
| Supporting Time-Based Offerings on Orders                                     | 12-23 |
| Supporting Time-Based Offerings on New Orders                                 | 12-23 |
| Supporting Time-Based Offerings on Change Orders                              | 12-23 |
| Supporting Friends and Family Lists   | 12-24 |
| Using Change Orders with Special Rating Products                              | 12-24 |
| Supporting Split Billing  | 12-25 |
| Assumptions and Constraints for the Bill Fulfillment Order Business Flow      | 12-25 |

## 13 Implementing the Bill Fulfillment Order Business Flow

---

|   |      |
|---|------|
| About the Bill Fulfillment Order Business Flow                                  | 13-1 |
| About the 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 |
| ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer                            | 13-5 |

|   |       |
|---|-------|
| ProcessFulfillmentOrderBillingBRMCommsProvABCImpl             | 13-5  |
| ProcessFulfillmentOrderBillingBRMCommsAddSubProcess           | 13-6  |
| ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess       | 13-7  |
| ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess | 13-8  |
| ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess        | 13-10 |
| ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess        | 13-11 |
| ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess  | 13-12 |
| ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer | 13-12 |

## 14 Understanding the Provision Order and Update Fulfillment Order Business Flows

---

|  |      |
|--|------|
| About Order Provisioning                         | 14-1 |
| About the Provision Order Business Flow          | 14-1 |
| About the Update Fulfillment Order Business Flow | 14-1 |

## 15 Implementing the Provision Order and Update Fulfillment Order Business Flows

---

|   |      |
|---|------|
| About the Provision Order and Update Fulfillment Order Business Flows | 15-1 |
| About the OSM Fulfillment to OSM Provisioning Integration Flow        | 15-1 |
| Defining Transaction Boundaries and Recovery Details                  | 15-2 |
| Industry Oracle AIA Components  | 15-3 |
| Integration Services  | 15-4 |
| ProcessProvisioningOrderOSMCFSCCommsJMSProducer                       | 15-4 |
| ProcessProvisioningOrderOSMPROVCommsJMSProducer                       | 15-4 |
| ProcessFulfillmentOrderUpdateOSMPROVCommsJMSProducer                  | 15-4 |
| ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer                  | 15-5 |

## 16 Understanding the Update Sales Order Business Flow

---

|  |      |
|--|------|
| About the Update Sales Order Business Flow | 16-1 |
| About Updating Sales Order Data            | 16-1 |
| About Updating Sales Order Status          | 16-1 |

## 17 Implementing the Update Sales Order Business Flow

---

|   |      |
|---|------|
| About the Update Sales Order Business Flow                          | 17-1 |
| About the 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 |
| UpdateSalesOrderOSMCFSCommsJMSConsumer  | 17-4 |
| UpdateSalesOrderSiebelCommsJMSConsumer  | 17-4 |
| UpdateSalesOrderSiebelCommsProvABCSImpl | 17-4 |

## 18 Understanding the Process Integration for Customer Management

---

|   |       |
|---|-------|
| About the Process Integration for Customer Management               | 18-1  |
| About the Synchronize Customer Account Business Flow                | 18-1  |
| About the Create/Sync Customer Account Integration Flow             | 18-2  |
| About the Update Customer Account Integration Flow                  | 18-4  |
| About Account Status Synchronization                                | 18-5  |
| About the Synchronize Customer Special Rating Profile Business Flow | 18-6  |
| Data Requirements   | 18-6  |
| About Synchronizing Accounts to Multiple BRM Instances              | 18-7  |
| About Account and Billing Hierarchies                               | 18-7  |
| About Account Hierarchies   | 18-7  |
| About Billing Hierarchies   | 18-7  |
| About Corporate Account Hierarchies                                 | 18-8  |
| Enabling and Disabling Corporate Account Hierarchy Synchronization  | 18-8  |
| Creating Corporate Account Hierarchies                              | 18-8  |
| Synchronizing Corporate Account Hierarchies to BRM                  | 18-9  |
| Synchronizing Hierarchies for New Corporate Accounts                | 18-9  |
| Synchronizing Hierarchies for Existing Corporate Accounts           | 18-10 |
| Updating Corporate Account Hierarchies                              | 18-11 |
| Adding Accounts to Existing Corporate Account Hierarchies           | 18-11 |
| Removing Accounts from Corporate Account Hierarchies                | 18-12 |
| Moving Accounts to Different Corporate Account Hierarchies          | 18-13 |
| About Legal Hierarchies   | 18-13 |
| Enabling and Disabling Legal Hierarchy Synchronization              | 18-13 |
| Creating and Synchronizing Legal Hierarchies                        | 18-14 |
| Example of Legal Hierarchies  | 18-14 |
| Updating Legal Hierarchies  | 18-15 |
| Adding New Members to Legal Hierarchies                             | 18-15 |
| Changing the Legal Owner of a Child Account                         | 18-15 |
| Removing the Legal Owner from a Child Account                       | 18-16 |

## 19 Implementing the Synchronize Customer Account Business Flow

---

|   |      |
|---|------|
| About the Synchronize Customer Account Business Flow    | 19-1 |
| About the Create/Sync Customer Account Integration Flow | 19-1 |
| About the Update Customer Account Integration Flow      | 19-1 |

|   |       |
|---|-------|
| BRM Interfaces                                    | 19-4  |
| Siebel CRM Interfaces                             | 19-4  |
| Industry Oracle AIA Components                    | 19-5  |
| Integration Services                              | 19-6  |
| CommsProcessBillingAccountListEBF                 | 19-6  |
| CommunicationsCustomerPartyEBSV2Resequencer       | 19-7  |
| SyncCustomerPartyListBRMCommsProvABCImpl          | 19-7  |
| SyncCustomerPartyListBRMCommsJMSProducer          | 19-9  |
| SyncCustomerPartyListBRM_01CommsJMSSConsumer      | 19-9  |
| CustomerPartyEBSV2                                | 19-9  |
| ProcessCollectionSharingBRMCommsProvABCImpl       | 19-10 |
| ProcessAccountHierarchyListSiebelCommsProvABCImpl | 19-10 |
| QueryCustomerPartyListSiebelProvABCImplV2         | 19-11 |
| SyncCustomerSiebelEventAggregator                 | 19-11 |
| SyncAccountSiebelAggregatorAdapter                | 19-12 |
| SyncContactSiebelAggregatorAdapter                | 19-12 |
| SyncAddressSiebelAggregatorAdapter                | 19-12 |
| SyncBPSiebelAggregatorAdapter                     | 19-12 |
| SyncAcctSiebelAggrEventConsumer                   | 19-12 |
| SyncAccountSiebelReqABCImpl                       | 19-13 |

## 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  | 20-2 |
| Siebel CRM Interfaces   | 20-3 |
| Industry Oracle AIA Components  | 20-3 |
| Integration Services  | 20-3 |
| ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSSConsumer        | 20-4 |
| ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl          | 20-4 |
| ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl            | 20-4 |

## 21 Understanding the Process Integration for Order Fallout Management

---

|   |      |
|---|------|
| About the Process Integration for Order Fallout Management        | 21-1 |
| About the Order Fallout Management Business Flows                 | 21-1 |
| About the Create Trouble Ticket from Oracle AIA Business Flow     | 21-1 |
| About the Create and Manage Trouble Ticket from OSM Business Flow | 21-2 |
| About Order Fallout Management for Different Error Types          | 21-4 |

|   |       |
|---|-------|
| About Order Fallout Management for System Errors              | 21-4  |
| Reactivating Retired Consumer Services                        | 21-5  |
| Disabling Consumer Service Retirement                         | 21-6  |
| About Order Fallout Management for Business Errors            | 21-6  |
| About Order Fallout Management for Service Errors             | 21-6  |
| About Order Fallout Detection                                 | 21-7  |
| About Order Fallout Notification                              | 21-8  |
| About Order Correction  | 21-9  |
| How Oracle AIA Error Handling Framework Captures Faults       | 21-11 |
| Extending Fault Messages to Capture Order Fallout Information | 21-15 |
| 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         | 22-2 |
| Exception Handling   | 22-3 |
| Siebel CRM Interfaces  | 22-3 |
| Industry Oracle AIA Components                               | 22-3 |
| Integration Services   | 22-4 |
| CreateTroubleTicketSiebelCommsProvABCServiceImpl             | 22-4 |
| AIAOrderFalloutJMSBridgeService                              | 22-5 |
| AIACOMOrderFalloutNotificationJMSConsumer                    | 22-5 |
| CreateTroubleTicketAIACommsReqImpl                           | 22-5 |
| AIAOrderFalloutErrorHandlerExtension - Java Class            | 22-5 |
| Business Flow Fallout-Enabled Services                       | 22-5 |

## 23 Implementing the Create and Manage Trouble Ticket from OSM Business Flow

---

|   |      |
|---|------|
| About the Create and Manage Trouble Tickets from OSM Business Flow          | 23-1 |
| About the Order Failure Notification to OSM Integration Flow                | 23-1 |
| About the Creating a Trouble Ticket in Siebel CRM from OSM Integration Flow | 23-2 |
| Defining Transaction Boundaries and Recovery Details                        | 23-3 |
| Updating a Trouble Ticket in Siebel CRM from OSM Integration Flow           | 23-4 |
| Defining Transaction Boundaries and Recovery Details                        | 23-5 |
| Siebel CRM Interfaces   | 23-6 |
| Industry Oracle AIA Components  | 23-6 |
| Integration Services  | 23-7 |
| AIAOrderFalloutJMSBridgeService   | 23-7 |
| CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer                        | 23-7 |

|  |      |
|--|------|
| CreateOrderFalloutNotificationOSMCFSCommsProvImpl    | 23-7 |
| CreateOrderFalloutNotificationOSMCFSCommsJMSProducer | 23-8 |
| CreateTroubleTicketOSMCFSCommsJMSConsumer            | 23-8 |
| CreateTroubleTicketSiebelCommsProvABCImpl            | 23-8 |
| CreateTroubleTicketRespOSMCFSCommsJMSProducer        | 23-8 |
| UpdateTroubleTicketOSMCFSCommsJMSConsumer            | 23-9 |
| UpdateTroubleTicketSiebelCommsProvABCImpl            | 23-9 |
| CreateFaultNotificationLFCommsJMSConsumer            | 23-9 |
| Business Flow Fallout-Enabled Services               | 23-9 |

## Part II Configuring the Process Integrations

---

### 24 Configuring the Process Integration for Product Lifecycle Management

---

|  |       |
|--|-------|
| About Configuring the Process Integration for Product Lifecycle Management   | 24-1  |
| Configuring Siebel CRM for Integrated Product Lifecycle Management           | 24-1  |
| Configuring BRM for Integrated Product Lifecycle Management                  | 24-2  |
| Working with DVMs for Product Lifecycle Management                           | 24-4  |
| Working with Cross-References for Product Lifecycle Management               | 24-4  |
| Handling Error Notifications for Product Lifecycle Management                | 24-5  |
| About Error Notification Roles and Users                                     | 24-5  |
| Configuring Properties for Product Lifecycle Management                      | 24-5  |
| Configuration Properties for the Synchronize Product and Price Business Flow | 24-6  |
| Configuration Properties for BRM Services                                    | 24-6  |
| Configuration Properties for Siebel CRM Services                             | 24-8  |
| Configuration Properties for the Query Product Classes Business Flow         | 24-13 |

### 25 Configuring the Process Integration for Order Lifecycle Management

---

|  |       |
|--|-------|
| About Configuring the Process Integration for Order Lifecycle Management | 25-1  |
| Working with Domain Value Maps for Order Lifecycle Management            | 25-1  |
| Working with the PRICELIST DVM   | 25-3  |
| Updating the PRICELIST DVM   | 25-3  |
| Migrating PRICELIST DVM Between Environments                             | 25-3  |
| Working with Cross-References for Order Lifecycle Management             | 25-4  |
| Handling Error Notifications for Order Lifecycle Management              | 25-5  |
| About Error Notification Roles and Users                                 | 25-6  |
| Configuring the Process Integration for Order Lifecycle Management       | 25-6  |
| Setting Oracle AIA Configuration Properties                              | 25-7  |
| System-Level Configuration Properties                                    | 25-7  |
| Configuration Properties for Siebel CRM Services                         | 25-7  |
| Configuration Properties for BRM Services                                | 25-10 |

## 26 Configuring the Process Integration for Customer Management

---

|   |      |
|---|------|
| About Configuring the Process Integration for Customer Management | 26-1 |
| Working with Domain Value Maps for Customer Management            | 26-1 |
| Working with Cross-References for Customer Management             | 26-2 |
| Handling Error Notifications                                      | 26-3 |
| About Error Notification Roles and Users                          | 26-4 |
| Order Fallout Management  | 26-4 |
| Configuring the Process Integration for Customer Management       | 26-5 |
| System-Level Configuration Properties                             | 26-5 |
| Configuration Properties for BRM Services                         | 26-5 |
| Configuration Properties for Siebel CRM Services                  | 26-9 |

## 27 Configuring the Process Integration for Order Fallout Management

---

|  |      |
|--|------|
| About Configuring the Process Integration for Order Fallout Management | 27-1 |
| 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 |
| About Error Notification Roles and Users                               | 27-4 |
| 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-5 |
| Example of Complex Product Synchronization                                    | A-6 |
| Discount Synchronization Flow   | A-8 |
| Example of Discount Synchronization   | A-9 |

|          |  |     |
|----------|--|-----|
| <b>B</b> | <b>Communications Orders Dictionary</b>  |     |
|          | Order Header Component Attributes  | B-1 |
|          | Order Line Component Attributes  | B-9 |
| <b>C</b> | <b>Mapping Billing Dates</b>   |     |
|          | How Dates are Set in BRM   | C-1 |
| <b>D</b> | <b>Supporting MACD Actions and Attribute Changes</b>                           |     |
|          | 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-3 |
|          | Communicating Product Attribute Changes to BRM                                 | D-3 |
|          | Communicating Marketing Bundle Attribute Changes                               | D-4 |
|          | Communicating Service Bundle Attribute Changes                                 | D-4 |
|          | Communicating Service Bundle Component Attribute Changes                       | D-4 |
|          | Communicating Service-Level Billing Subscription Product Attribute Changes     | D-4 |
|          | Communicating Service-Level Billing Discount Product Attribute Changes         | D-5 |
|          | Communicating Service-Level Billing Item Product Attribute Changes             | D-5 |
|          | Communicating Service-Level Special Rating Product Attribute Changes           | D-5 |
|          | Communicating Account-Level Product Attribute Changes                          | D-6 |
|          | Communicating Account-Level Billing Subscription Product Attribute Changes     | D-6 |
|          | Communicating Account-Level Billing Discount Attribute Changes                 | D-6 |
|          | Communicating Account-Level Billing Item Product Attribute Changes             | D-7 |
| <b>E</b> | <b>Examples of Changing the Paying Account for Child Accounts</b>              |     |
|          | About the Examples   | E-1 |
|          | Examples when Service-Level Balance Groups Are Enabled                         | E-2 |
|          | Changing the Paying Account for a Child Account with Separate Billing Profiles | E-2 |
|          | Alternative: Single Billing Profile  | E-3 |
|          | Alternative: Multiple Parents  | E-3 |
|          | Changing the Paying Account for a Child Account with a Single Billing Profile  | E-4 |
|          | Alternative: Multiple Billing Profiles   | E-5 |
|          | Changing the Paying Account for Multiple Child Accounts                        | E-5 |
|          | Alternative: Single Billing Profile  | E-6 |
|          | Changing Multiple Paying Accounts  | E-7 |
|          | Alternative: Separate Billing Profiles   | E-8 |
|          | Alternative: Self Paying and Paying Parent                                     | E-8 |
|          | Examples when Service-Level Balance Groups Are Disabled                        | E-9 |
|          | Changing the Paying Parent for a Child Account                                 | E-9 |



|   |      |
|---|------|
| Changing the Paying Parent for Multiple Child Accounts with Separate Billing Profiles | E-10 |
| Problems When Integrating Separate Billing Profiles on the Same Account               | E-11 |

## 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-14 |
| Creating a New Consumer for Customer Updates             | F-22 |
| Creating a Data Source and Connection Factory            | F-30 |
| Creating Logical Instances in Oracle AIA                 | F-33 |
| Creating Service Bundles in Siebel CRM                   | F-33 |
| 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 Reintroducing Enterprise Business Services

---

|  |     |
|--|-----|
| Reintroducing Enterprise Business Services | K-1 |
|--|-----|

## L Understanding Multischema Migration

---

|   |     |
|---|-----|
| About Multischema Migration   | L-1 |
| Mapping BRM POIDs to AIA Cross-References                                   | L-1 |
| Synchronizing Cross-References for Migrated Accounts                        | L-2 |
| Configuring Oracle AIA and BRM to Work Together in Multischema Environments | L-3 |

|   |     |
|---|-----|
| Configuring Oracle AIA to Work with BRM in Multischema Environments | L-3 |
| Configuring BRM to Work with Oracle AIA in Multischema Environments | L-4 |
| Handling Errors   | L-5 |
| Assumptions and Constraints for Multischema Migration               | L-5 |
| Integration Services  | L-5 |
| About the SyncMultiSchemaChangeBRMAQ Consumer                       | L-5 |
| About the SyncMultiSchemaChangeBRMRequester Service                 | L-5 |
| About the MultiSchemaUpdate.sql Process                             | L-6 |

# Preface

This document describes how to implement and use the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack.

## Audience

This document is intended for customer service representatives, billing and pricing administrators, and other individuals who are responsible for configuring, managing and maintaining the Order to Cash integration pack.

## 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 that have purchased support 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.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

# 1

## Overview of the Oracle Communications Order to Cash Integration Pack

This chapter provides an overview of the Oracle Communications Order to Cash pre-built integration (the integration). It describes integration architecture and topologies, the process integrations, and the pre-built integration options.

### About Oracle AIA and Oracle Communications Solutions Integrations

Oracle Application Integration Architecture (Oracle AIA) is an Oracle Communications Solutions Integrations framework that provides pre-built integrations using standard integration patterns, business processes, orchestration logic, and common objects and services to connect Oracle applications.

Oracle Communications Solutions Integrations is a set of integration frameworks, technologies, and tools that lets you design and build integrations that connect Oracle applications to support end-to-end business processes for communications service providers across operations support systems and business support systems.

### Overview of the Oracle Communications Order to Cash Integration Pack

Oracle AIA provides integration and deployment accelerators based on industry best practices and a comprehensive integration methodology.

The integration automates business flows across Siebel CRM, Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM).

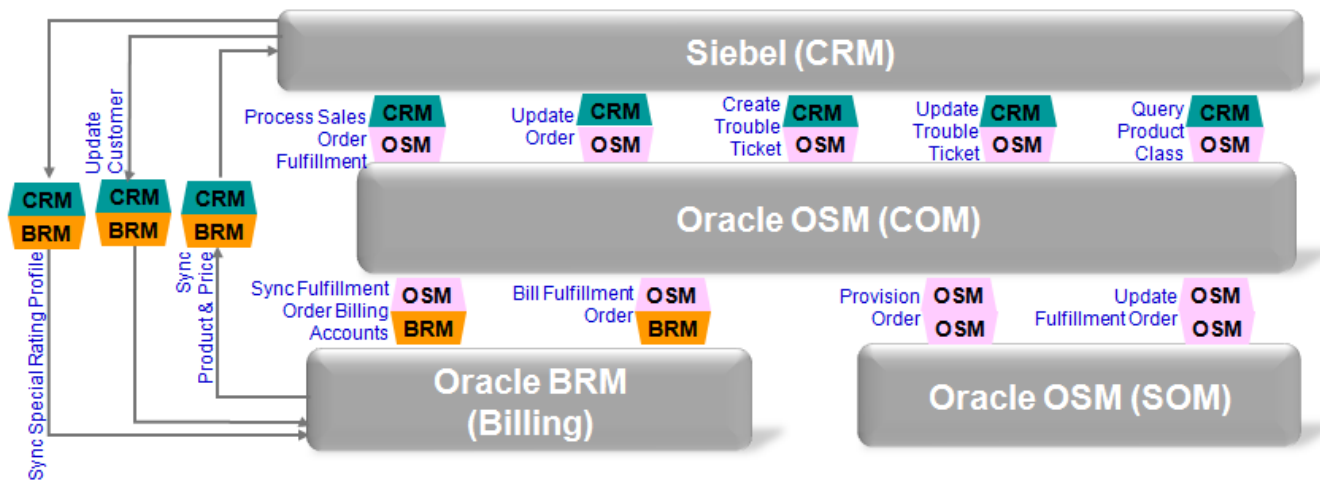
A business flow is a high-level process coordinated by the integration that accomplishes an organizational goal, such as provisioning an order or updating a customer's account. Business flows are implemented by integration flows. An integration flow is a run-time processes that represents the journey of a message from one application to another, including intermediary stops and transformation on the way.

When you install the integration, you choose to deploy any of the following Oracle Communications pre-built integration options, based on your system topology:

- Order to Cash for BRM
- Order to Cash for OSM
- Order to Cash for Siebel CRM

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

**Figure 1-1 Order to Cash Functional Overview**

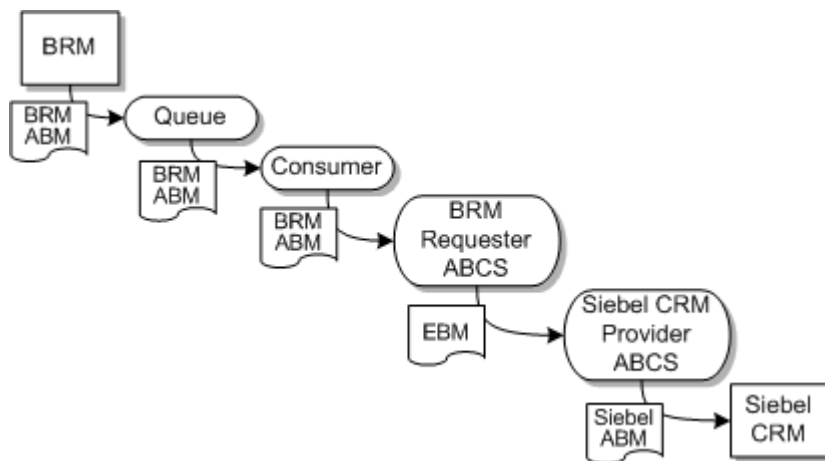


The integration flows implement the business flows using the following service-oriented architecture components:

- Application business messages (ABMs): Messages specific to a requester or provider application such as Siebel CRM, OSM, or BRM.
- Enterprise business messages (EBMs): Generic messages used within the integration.
- Applications business connector services (ABCSs): Services that receive messages, transform them, and pass them on. ABCSs can receive and send messages to and from other ABCSs, queue consumer or producer services, and application web services.

Figure 1-2 shows an example of an integration flow. Not all integration flows include queues and some include queues for the requester and provider application.

**Figure 1-2 Example Integration Flow**



You can extend the ready-to-use integration using the pre-defined communications industry enterprise business objects (EBOs) delivered with Oracle AIA.

## About Using Third-Party Applications

Deploying the OSM, BRM, and Siebel CRM pre-built integration options provides accelerated integration between all participating applications, but you can also use third party applications for billing, orchestration, or account and asset management.

To use a third party application with the Order to Cash pre-built integration, you must extend the integration by creating services which allow your third-party applications to fulfill the roles and responsibilities specified by the design described in this guide. To achieve the functionality described in this guide, you must develop your system and connections to mimic what is described for the ready-to-use integration.

## About the Test Orchestration Process

If you have deployed the BRM and Siebel CRM pre-built integration options and use your own order management system, you can optionally install the test orchestration process to sanity test the ready-to-use order flow. You must replace the test orchestration process 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.

## About Oracle AIA in Product-Based Solutions

Oracle AIA and the Order to Cash integration are central to Oracle's product-based order management and service fulfillment solutions. These solutions are:

- Oracle Communications Rapid Offer Design and Order Delivery Solution (RODOD)
- Oracle Communications Rapid Service Design and Order Delivery Solution (RSDOD)

### About RODOD

RODOD is a TM Forum certified solution that provides rapid offer design and order lifecycle management. RODOD includes the following aspects:

- Offer design: Using Oracle Product Hub, you design and maintain products and offers in a central location to use across RODOD applications. Using Design Studio, you use the products defined in Product Hub to design the workflows for order fulfillment, decomposition, orchestration, and change management, which can be reused and adapted to new offers.
- Order delivery: Using Siebel CRM, you capture and validate orders from a variety of channels. Siebel CRM passes customer orders to OSM, which delivers orders by implementing the fulfillment workflows designed in Design Studio.

Oracle AIA integrates the RODOD components with the Order to Cash integration.

### About RSDOD

RSDOD provides rapid service design and order lifecycle management with an automated flow of service requests through provisioning systems. RSDOD includes the following aspects:

- Service design: Using Design Studio, you design and deploy standardized provisioning, inventory, and activation configurations to use across the RSDOD applications and any

legacy applications. Commercial products and services are decoupled from their underlying network technology, letting you reuse or modify existing configurations for new services and domains.

- Order delivery: OSM provides advanced automated order lifecycle management that includes technical service and resource qualification and real-time visibility into order lifecycle status. During order delivery, OSM communicates with Oracle Communications Unified Inventory Management (UIM), which designs and assigns services, and Oracle Communications ASAP, which activates the services on the network.

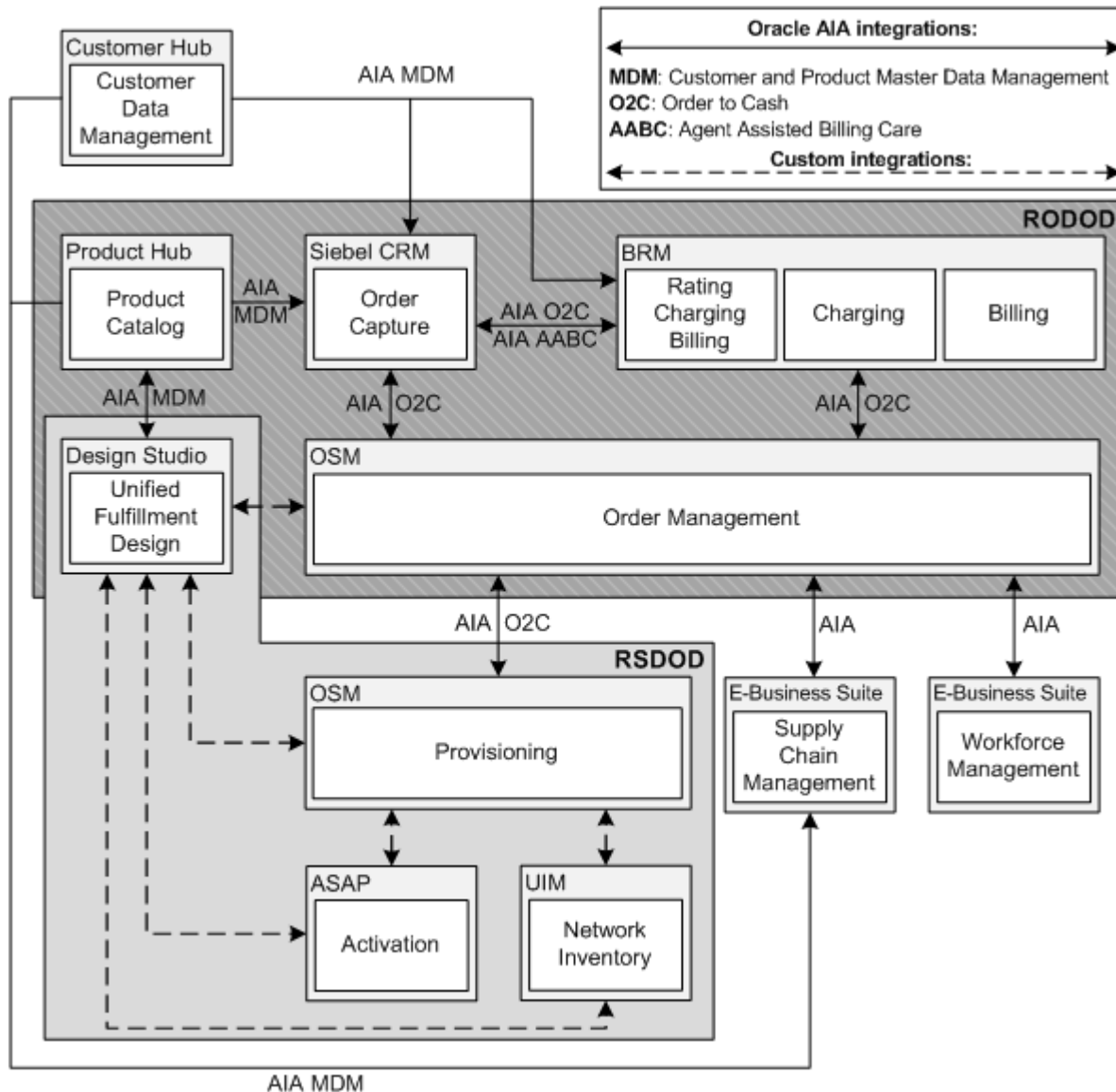
You configure the RSDOD components to interact directly through custom integrations.

## About Integrated RODOD and RSDOD Deployments

Used together, RSDOD and RODOD provide an end-to-end integrated solution that lets you efficiently design products and services and fulfill and deliver them across multiple channels and multiple network domains. The Order to Cash integration plays a key role in integrating the RODOD and RSDOD solutions and extending the solutions to include other applications.

[Figure 1-3](#) shows how RODOD and RSDOD fit together and overlap in an integrated deployment, and how Oracle AIA integrates the component applications.

Figure 1-3 RODOD and RSDOD in an Oracle AIA Integrated Deployment



## 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, discounts, and chargeshares in BRM and use the integration to synchronize them to 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 AIA Oracle Product Master Data Management Integration Implementation Guide* for more information.

Table 1-1 lists the business flows for product lifecycle management and the pre-built integration options that enable them.

**Table 1-1 Product Lifecycle Management Business Flows**

| Business Flow  | Pre-Built Integration Options Required for the Flow                           |
|--|---|
| Synchronize Product and Price: Oracle AIA synchronizes products, discounts, chargeshares, and their pricing information from BRM to Siebel CRM<br>See " <a href="#">Understanding the Synchronize Product and Price Business Flow</a> ". | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• BRM</li> </ul> |
| Query Product Classes: Import product classes from Siebel CRM to Design Studio<br>See " <a href="#">Understanding the Query Product Classes Business Flow</a> ".   | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• OSM</li> </ul> |

## Pricing Center and Pricing Design Center Terminology

You can use either the Pricing Center or Pricing Design Center (PDC) component of BRM to create the products and discounts that Oracle AIA synchronizes to Siebel CRM with the Order to Cash integration.

Because Pricing Center and PDC are components of BRM, the diagrams and flow descriptions in this guide use BRM to stand in for Pricing Center and PDC.

The PDC term for a component sometimes differs from the Pricing Center term for the same component. This guide uses the Pricing Center term unless otherwise specified. Table 1-2 lists the differences relevant to this guide.

**Table 1-2 Mapping Pricing Center to PDC Terminology**

| Pricing Center Term | PDC Term          |
|---------------------|-------------------|
| chargeshare model   | chargeshare       |
| chargeshare         | chargeshare offer |
| deal                | bundle            |
| discount            | discount offer    |
| plan                | package           |
| product             | charge offer      |

**Table 1-2 (Cont.) Mapping Pricing Center to PDC Terminology**

| Pricing Center Term | PDC Term        |
|---------------------|-----------------|
| rate plan           | charge          |
| rate plan selector  | charge selector |
| subscription group  | service group   |

For more information about PDC, including further terminology differences between PDC and Pricing Center, see *Oracle Communications Pricing Design Center User's Guide*.

## About Order Lifecycle Management

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

[Table 1-3](#) describes the business flows for order lifecycle management and the pre-built integration options that enable them.

**Table 1-3 Order Management Business Flows**

| Business Flow  | Pre-Built Integration Options Required for the Flow  |
|--|--|
| Process Sales Order Fulfillment: Siebel CRM submits orders to OSM<br>See " <a href="#">Understanding the Process Sales Order Fulfillment Business Flow</a> ".  | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• OSM</li> </ul>                |
| Update Sales Order: OSM sends order updates to Siebel CRM<br>See " <a href="#">Understanding the Update Sales Order Business Flow</a> ".   | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• OSM</li> </ul>                |
| Synchronize Fulfillment Order Billing Account: OSM decomposes orders to create customer data in BRM<br>See " <a href="#">Understanding the Synchronize Fulfillment Order Billing Account Business Flow</a> ".  | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• OSM</li> <li>• BRM</li> </ul> |
| Bill Fulfillment Order: OSM decomposes orders to create transaction data in BRM<br>See " <a href="#">Understanding the Bill Fulfillment Order Business Flow</a> ".   | <ul style="list-style-type: none"> <li>• OSM</li> <li>• BRM</li> </ul>                       |
| Provision Order: OSM in the Central Order Management (COM) role decomposes and sends orders to OSM in the Service Order Management (SOM) role for provisioning<br>See " <a href="#">Understanding the Provision Order and Update Fulfillment Order Business Flows</a> ". | <ul style="list-style-type: none"> <li>• OSM</li> </ul>                                      |
| Update Fulfillment Order: OSM in the SOM role sends provisioning status updates to OSM in the COM role<br>See " <a href="#">Understanding the Provision Order and Update Fulfillment Order Business Flows</a> ".   | <ul style="list-style-type: none"> <li>• OSM</li> </ul>                                      |

## 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 1-4](#) lists the business flows for customer management and the pre-built integration options that enable them.

**Table 1-4 Customer Management Business Flows**

| Business Flow   | Pre-Built Integration Options Required for the Flow                           |
|---|---|
| Synchronize Customer Account: Using order data, query Siebel CRM for customer data to create accounts in BRM.<br>See " <a href="#">About the Synchronize Customer Account Business Flow</a> ".  | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• BRM</li> </ul> |
| Synchronize Customer Special Rating Profile: Using order data, query Siebel CRM for friends and family list information to create special rating profiles in BRM.<br>See " <a href="#">About the Synchronize Customer Special Rating Profile Business Flow</a> ". | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• BRM</li> </ul> |

## 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 1-5](#) lists the business flows for order fallout management and the pre-built integration options that enable them.

**Table 1-5 Order Fallout Management Business Flows**

| Business Flow   | Pre-Built Integration Options Enabling the Flow                               |
|---|---|
| Create and Manage Trouble Ticket from OSM<br>See " <a href="#">About the Create and Manage Trouble Ticket from OSM Business Flow</a> ". | <ul style="list-style-type: none"> <li>• Siebel CRM</li> <li>• OSM</li> </ul> |
| Create Trouble Ticket from Oracle AIA<br>See " <a href="#">About the Create Trouble Ticket from Oracle AIA Business Flow</a> ".         | <ul style="list-style-type: none"> <li>• Siebel CRM</li> </ul>                |

## Example Order to Cash Business Scenarios

This section presents example business scenarios that illustrate how some of the features of the Oracle AIA Order to Cash integration work together to meet business needs. The scenarios provide a high-level end-to-end perspective, from design time, to order capture, to billing.

The features described in this section are explained in greater detail throughout this document. Each scenario includes references to specific sections where you can learn more about the concepts discussed.

The scenarios use the fictional example companies described in [Table 1-6](#).

**Table 1-6 Example Companies**

| Company         | Description   |
|-----------------|---|
| TruGreen        | A telecommunications service provider using an Oracle RODOD-based solution to offer wireless, VoIP, and broadband services to corporations and consumers. |
| Century Systems | A large software corporation with offices in London, Paris, and San Francisco.  |

## Promotion Groups and Large Order Scenario

This scenario illustrates how Oracle AIA supports large orders and promotion groups.

In this scenario, TruGreen offers a Corporate Optima offer for their business market. This offer provides VoIP and wireless services for businesses with sites in multiple cities. The offer includes free calls within a site, a 25% discount for calls between sites, and a pool of 20,000 free minutes shared between sites. External calls cost 10 cents a minute.

### Design Time

TruGreen models the free calls, discount, and pool of free minutes using BRM discounts, Siebel CRM special rating products, and Siebel CRM promotion groups. At design time, the TruGreen product administrator does the following:

1. Creates the 25% discount and the free minutes as discount products in BRM and synchronizes them to Siebel CRM through Oracle AIA.
2. Creates a special rating product in Siebel CRM to support free calls within a site.
3. Creates promotion group membership products in Siebel CRM for promotion group owners and members.
4. Creates a promotion group definition in Siebel CRM with the discount products and special rating product as rewards and the group membership products a owner and members.

### Order Time

Century Systems signs a contract with TruGreen to set up a network that includes each of Century Systems' three sites. To provide VoIP and wireless services for employees, Century Systems subscribes to TruGreen's Corporate Optima offer.

A TruGreen sales representative does the following:

1. Sets up the network with a network order in Siebel CRM. The order includes line items for:
  - The network
  - A network node for each Century Systems office
  - Network connections for each node
  - A subscription to the Corporate Optima offer for each node
2. Creates a bulk request in Siebel CRM that includes:
  - 100 accounts for each site
  - VoIP and wireless services for each account

- The promotion group with the corporate-level wireless service as the owner and the employee-level wireless services as members
- Three special rating lists, one for each of Century System's sites, associated with the promotion group's special rating reward product

The bulk request results in a large order with more than 1000 order lines.

Siebel CRM sends the large order to OSM through Oracle AIA. To process the large order efficiently, Oracle AIA splits the order into small segments, transforms it into a format that OSM understands, and then recombines the segments and sends the order to OSM.

OSM manages fulfillment of the order, including sending billing information to BRM through Oracle AIA. To support the promotion group, Oracle AIA automatically creates three sharing groups in BRM; one for the 25% discount product, one for the 20,000 free minutes discount product, and one for the special rating profile product. For each sharing group, the corporate account's wireless service is the owner and the employee accounts' wireless services are members.

After the order is fulfilled, the TruGreen representative uses Siebel CRM to update the London, Paris, and San Francisco special rating lists with the phone numbers for each respective Century Systems site. The employees can make calls that take advantage of the discounts provided by the promotion group on the Corporate Optima offer.

## More Information About Large Orders and Promotion Groups

See the following sections for more information about large orders and promotion groups:

- See "[Supporting Large Orders](#)" for information about how Oracle AIA supports large orders.
- See "[About the Submitting Orders from Siebel CRM to OSM Integration Flow](#)" for information about the integration flow that handles large orders.
- See "[Supporting Promotion Groups on Orders](#)" for information about including promotion groups on sales orders.
- See "[Synchronizing Promotion Groups](#)" for information about how Oracle AIA uses Siebel CRM promotion groups to create sharing groups in BRM.
- See "[Configuring the Process Integration for Order Lifecycle Management](#)" for information about configuration properties and cross references for large orders and promotion groups.

## Family Share Plan and Payment on Order Scenario

This scenario illustrates how Oracle AIA supports family share plans and payments on orders.

In this scenario, TruGreen offers a Family Wireless offer for their consumer market. This offer includes a primary wireless phone line for \$50 monthly, a second line for \$45 monthly, and up to three more add-on lines for \$30 monthly. As part of the offer, these lines all share up to 250MB of data.

## Design Time

TruGreen models the primary, secondary, and add-on wireless lines using Siebel CRM service bundles that include products synchronized from BRM, and the shared data using a BRM discount. The service bundles and discount are included in bundled product promotions.

At design time, the TruGreen product administrator does the following:

1. Creates the components of wireless lines, including voice, data, and text service, as products in BRM and synchronizes them to Siebel CRM.
2. Creates the shared data as a discount in BRM and synchronizes it to Siebel CRM.
3. In Siebel CRM, using the wireless line component products synchronized from BRM, creates a service bundle to represent the primary line, a service bundle to represent the secondary line, and a service bundle to represent the add-on lines.
4. Creates a new Community Member attribute to indicate that a service bundle is a member of the family share plan.
5. Creates a new Community Offer attribute to indicate that a discount is a shared reward in a family share plan.
6. Associates the attributes with the product classes of the service bundles representing wireless lines and the discount representing the shared data.
7. Creates a product promotion that includes the primary, secondary, and add-on wireless lines, and the shared data discount.
8. Sets the Community Member attribute on the wireless line service bundles to indicate that they are members in a family share plan.
9. Sets the Community Offer attribute on the shared data discount to indicate that it is a reward in a family share plan.

## Order Time

A new customer, Denise, signs up for TruGreen's Family Wireless offer, and wants to purchase two new phones at the same time.

A TruGreen customer service representative (CSR) does the following:

1. Creates accounts for Denise and her two daughters, Michelle and Jessica, setting the parent on the daughters' accounts to Denise's account.
2. Creates an order for the Family Wireless offer. Denise wants to pay for all three wireless lines using a different billing profile for herself and each daughter.
3. Records two new credit card payments for the phones on the order, one using the billing profile for Michelle's service, and one using the billing profile for Jessica's service.
4. Submits the order.

Siebel CRM sends the order to OSM through Oracle AIA. OSM manages fulfillment of the order, including sending billing and payment information to BRM through Oracle AIA. Oracle AIA records the payments received on the order in BRM.

Because the Community attributes are set on the order lines, Oracle AIA creates a sharing group in BRM. The shared data is the reward, Denise's wireless service is the owner, and Michelle and Jessica's wireless services are members.

After the order is fulfilled and they receive their new phones, Denise, Michelle, and Jessica can all use their wireless lines and the shared data.

## More Information About Family Share Plans and Payments on Orders

See the following sections for more information about family share plans and payments made at order time:

- See "[About Family Share Plans](#)" for information about setting up family share plans at design time.

- See "[Supporting Family Share Plans on Orders](#)" for more information about placing orders that include family share plans.
- See "[Synchronizing Family Share Plans](#)" for information about how Oracle AIA synchronizes order lines that represent family share plans to BRM as sharing groups.
- See "[About Accepting Payments on Orders](#)" for more information about accepting payments at order time.

## Corporate Account Hierarchy Scenario and Split Billing Scenario

This scenario illustrates how Oracle AIA supports corporate account hierarchies and split billing.

In this scenario, Century Systems subscribes to TruGreen's Corporate Double-Play offer, which provides VoIP and wireless service for employees. The offer includes reduced rates for the wireless service compared to the consumer rates. Century Systems pays for employees' VoIP service, and employees can choose to sign up and pay for the reduced rate wireless service.

### Design Time

At design time, the TruGreen administrator does the following:

1. In BRM, creates the components of VoIP and wireless lines as products and synchronizes them to Siebel CRM.
2. In Siebel CRM, using the VoIP and wireless line component products synchronized from BRM, creates service bundles to represent the lines.
3. In the AIA configuration properties file, sets the value of the Oracle AIA corporate hierarchy system property to BUSINESS.

### Order Time

To support the corporate account hierarchy, the TruGreen administrator does the following:

1. Creates a hierarchy of accounts for Century Systems in Siebel CRM, including a main account for the entire corporation, and billing and service accounts for all departments and employees.
2. Submits orders for the various employees. While processing orders and synchronizing accounts to BRM, Oracle AIA also synchronizes the corporate account hierarchy.

Andrew, a new employee in the sales department, decides to sign up for the VoIP and wireless services. Century Systems will pay for Andrew's VoIP service and Andrew will pay for his wireless service.

To add Andrew to the corporate hierarchy and split the bills, a TruGreen CSR does the following:

1. Creates an account for Andrew with the account type of BUSINESS and sets the parent for Andrew's account to the account for the sales department.
2. From the main Century Systems account, submits an order for the VoIP and wireless services. The CSR uses the following accounts:
  - Andrew's account as the service account for both services
  - Andrew's account as the billing account for the wireless service
  - The sales department's account as the billing account for the VoIP service

When the CSR submits the order, Siebel CRM sends the order to OSM through Oracle AIA. OSM manages fulfillment of the order, including sending billing information to BRM through Oracle AIA.

Because the account types on the order match the value of the Oracle AIA corporate hierarchy system property, Oracle AIA gets the entire linear hierarchy for Andrew's account from Siebel CRM and uses this hierarchy to do the following:

- Create a hierarchical account group for Andrew's account in BRM. This group includes the sales department as the parent.
- Update the existing hierarchical account group for the sales department in BRM. This group includes Andrew as a child.

To let the sales department pay for Andrew's VoIP service, Oracle AIA creates a billing hierarchy for Andrew's account in BRM. The hierarchy associates the VoIP service with the sales department's account and billing profile and the wireless service with Andrew's account and billing profile. When BRM generates the bills for Andrew's account, it uses the billing hierarchy to assign the bill for the VoIP service to the sales department and the bill for the wireless service to Andrew.

## More Information About Corporate Hierarchies and Split Billing

See the following sections for more information about corporate account hierarchies and split billing:

- See "[About Corporate Account Hierarchies](#)" for information about corporate account hierarchies, including conceptual information, how to enable and disable the synchronization of corporate account hierarchies, how to create corporate account hierarchies in Siebel CRM, and how Oracle AIA synchronizes them to BRM as hierarchical account groups.
- See "[Supporting Corporate Account Hierarchies on Orders](#)" for information about placing orders to support corporate account hierarchies.
- See [Table 26-9](#) for information about the Oracle AIA configuration property for enabling the synchronization of corporate account hierarchies.
- See "[About Account and Billing Hierarchies](#)" for conceptual information about billing hierarchies.
- See "[Supporting Split Billing on Orders](#)" for information about placing orders to support split billing.
- See "[Supporting Split Billing](#)" for information about how Oracle AIA creates billing hierarchies in BRM.

## Legal Owner and Split Billing Scenario

This scenario illustrates how Oracle AIA supports legal owners and split billing.

In this scenario, TruGreen offers a double-play promotion for consumers that includes broadband and wireless service. Helen, a 17-year-old university student, orders the promotion. Her mother, Lisa, agrees to pay for the broadband service if Helen pays for the wireless service.

In Helen's country, it is illegal for TruGreen to take collections actions against minors. Because Helen is a minor, Lisa is the legal owner of Helen's services.



## Design Time

At design time, a TruGreen administrator does the following:

1. In BRM, creates the components of the broadband and the wireless line as products and synchronizes them to Siebel CRM.
2. In Siebel CRM, using the broadband and wireless line component products synchronized from BRM, creates service bundles to represent the broadband and wireless services.
3. In the AIA configuration properties file, sets the value of the Oracle AIA legal group system property to TRUE.

## Order Time

To split the bills, a TruGreen CSR places two orders: one for the broadband service, and one for the wireless service. The CSR uses Lisa's account as the billing account for the broadband service and Helen's account as the billing account for the wireless service. To make Lisa the legal owner of the services, the CSR uses Lisa's account as the owner account for both services.

Table 1-7 shows the accounts that the CSR uses for each service on the order.

**Table 1-7 Example Orders for Legal Owner and Split Billing**

| Order | Service   | Service Account | Billing Account | Billing Profile | Owner Account |
|-------|-----------|-----------------|-----------------|-----------------|---------------|
| 123   | Broadband | Helen           | Lisa            | Lisa-H          | Lisa          |
| 456   | Wireless  | Helen           | Helen           | Helen           | Lisa          |

The CSR submits both orders and Siebel CRM sends them to OSM through Oracle AIA. OSM manages fulfillment of the order, including sending billing information to BRM through Oracle AIA.

On the first order, to let Lisa pay for Helen's broadband service, Oracle AIA creates a billing hierarchy for Helen's account in BRM. The hierarchy associates the broadband service with Lisa's account and billing profile.

On the second order, to let Helen pay for her own wireless service, Oracle AIA updates the billing hierarchy in Helen's account. The hierarchy associates the wireless service with Helen's account and billing profile. When BRM generates the bills for Helen's account, it uses the billing hierarchy to assign the bill for the broadband service to Lisa and the bill for the wireless service to Helen.

To support legal ownership, Oracle AIA creates a collections sharing group in BRM that uses Lisa's **/billinfo** as the parent and Helen's **/billinfo** as the child. If Helen fails to pay the bill for her wireless service, BRM uses the data from the collections sharing group to take collections actions against Lisa rather than her daughter.

When Helen reaches the age of majority, the CSR can submit a change order to change the owner account of Helen's wireless service. Oracle AIA removes Helen's **/billinfo** from the Lisa's collections sharing group in BRM.

## More Information About Legal Owners and Split Billing

See the following sections for more information about legal owners and split billing:

- See "[About Legal Hierarchies](#)" for conceptual information about legal owners in legal hierarchies.
- See "[Supporting Legal Owners on Orders](#)" for information about placing orders to support legal owners.
- See [Table 26-4](#) for information about the Oracle AIA configuration property for enabling legal owners.
- See "[About Account and Billing Hierarchies](#)" for conceptual information about billing hierarchies.
- See "[Supporting Split Billing on Orders](#)" for information about placing orders to support split billing.
- See "[Supporting Split Billing](#)" for information about how Oracle AIA creates billing hierarchies in BRM.
- See "[Synchronizing Promotion Groups](#)" for information about how Oracle AIA uses Siebel CRM promotion groups to create sharing groups in BRM.

## Directory Placeholders Used in This Guide

The placeholders in [Table 1-8](#) are used in this guide.

**Table 1-8 Placeholders Used in This Guide**

| Placeholder        | Directory Description   |
|--------------------|---|
| <i>Oracle_home</i> | The base directory into which the Oracle AIA software was installed. This is typically the Oracle base directory.<br><br>This directory includes the <b>comms_home</b> directory, which the <b>commsenv</b> script sets as the COMMS_HOME environment variable. |

# 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](#)
- [Implementing the Provision Order and Update Fulfillment Order Business Flows](#)
- [Understanding the Update Sales Order Business Flow](#)
- [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](#)

# 2

## Understanding the Process Integration for Product Lifecycle Management

This chapter describes the process integration for product lifecycle management.

### About the Process Integration for Product Lifecycle Management

The process integration for product lifecycle management lets you synchronize products, discounts, and chargeshares (pricing components) from Oracle Communications Billing and Revenue Management (BRM) to Siebel CRM. You synchronize pricing components in real time or in batches.

You create and update pricing components in either Pricing Design Center or Pricing Center, which are product design components of BRM. When you commit the pricing components to the database, the process integration for product lifecycle management synchronizes them to Siebel CRM for enrichment.

This process integration includes 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 pricing components in BRM and synchronize them to Siebel CRM. It also lets you update pricing components in BRM and resynchronize them to Siebel CRM.

This business flow is enabled by the following pre-built integration options:

- Order to Cash for BRM
- Order to Cash for Siebel CRM

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 conceptual model Product entities. 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 conceptual model Product entities.

This business flow is enabled using the following pre-built integration options:

- Order to Cash for Siebel CRM
- Order to Cash for Oracle Communications Order and Service Management (OSM)

If you are using Oracle Product Hub as your product master, this flow can also include the Product Master Data Management pre-built integration. 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.

# 3

## Understanding the Synchronize Product and Price Business Flow

This chapter describes the Synchronize Product and Price business flow and explains the concepts from 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.

### About the Synchronize Product and Price Business Flow

This section describes how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) synchronizes products, discounts, and chargeshares (pricing components) in real-time and batches and synchronizes updates to these pricing components.

You can use either Pricing Center or Pricing Design Center (PDC), which are both components of BRM, to create the pricing components.

This chapter uses BRM to indicate both Pricing Center and PDC. It uses the terminology for BRM and Pricing Center, which may differ from the terminology for PDC. See "[Pricing Center and Pricing Design Center Terminology](#)" for information about differences between Pricing Center and PDC terminology.

### About Synchronizing Pricing Components

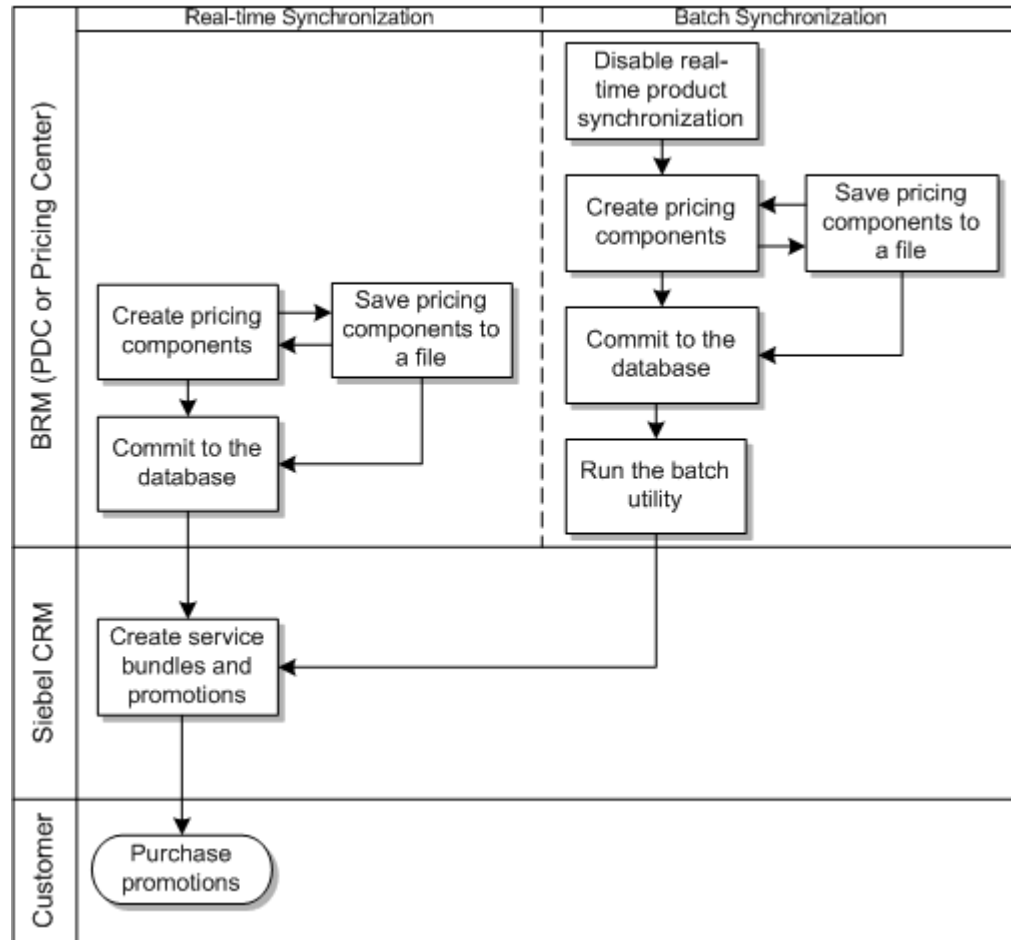
In this flow, the BRM product administrator creates pricing components and either commits single pricing components to the BRM database, or saves sets of pricing components to a file and commits the entire file to the BRM database at once. If the event for real-time product synchronization is disabled, the BRM administrator runs a batch utility to store the pricing components in the BRM database.

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

When the BRM product administrator stores the pricing components in the BRM database, either in real time or by using the batch utility, the integration synchronizes them to Siebel CRM. The Siebel CRM product administrator enriches the synchronized pricing components by defining product bundling. See "[Understanding Product Bundling](#)" for more information.

[Figure 3-1](#) shows how pricing components are created in BRM, synchronized to Siebel CRM in real time or in batches, and bundled in Siebel CRM for customers to purchase.

**Figure 3-1 Synchronizing Pricing Components**



## About Synchronizing Updates to Pricing Components

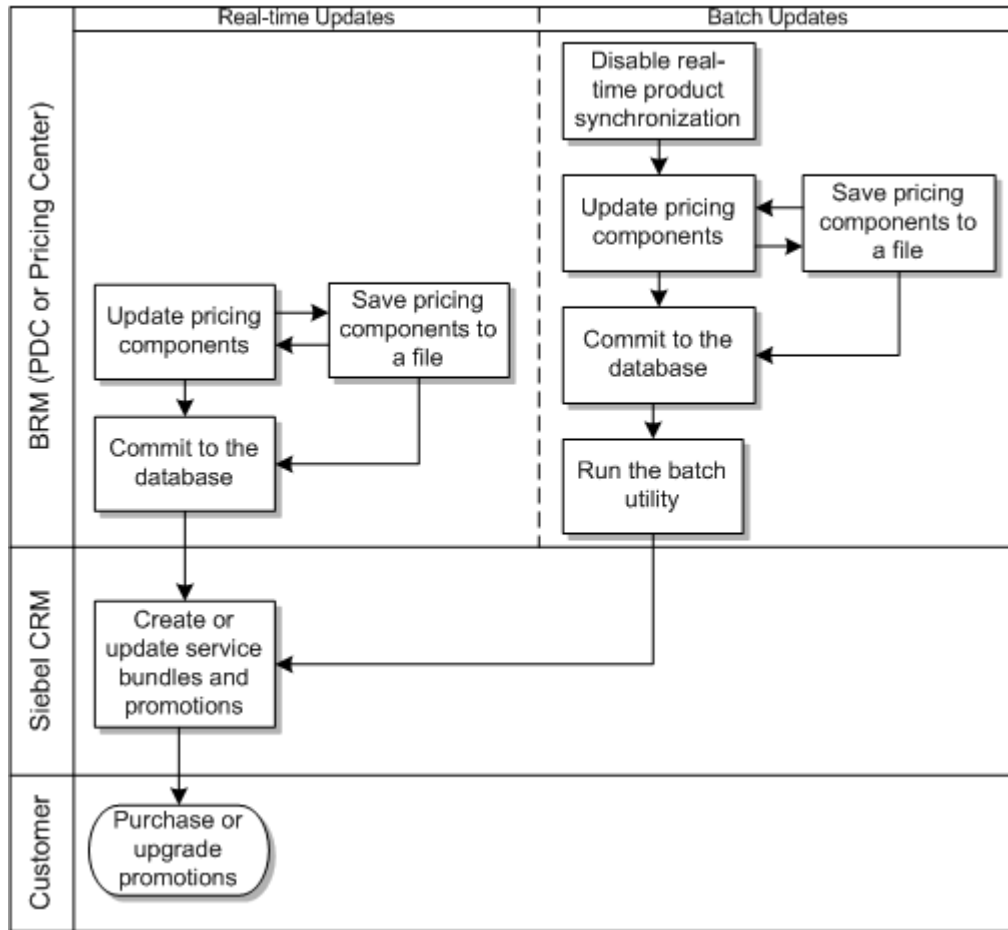
In this flow, the BRM product administrator updates the attributes of pricing components and commits them to the BRM database singly or as sets in a file. If the event for real-time product synchronization is disabled, the BRM administrator runs a batch utility to store the updated pricing components in the BRM database.

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

When the BRM product administrator commits the updated pricing components to the BRM database, either in real time or by running the batch utility, the integration synchronizes the updates to Siebel CRM. The service bundles and the promotions in Siebel CRM are automatically updated to use the latest version of the pricing components. The Siebel CRM product administrator can make any necessary changes to promotions and bundles before customers can purchase or upgrade a promotion.

Figure 3-2 shows how pricing components are updated in BRM, synchronized to Siebel CRM in real time or in batches, and updated in Siebel CRM bundles and promotions.

Figure 3-2 Synchronizing Pricing Component Updates



## About the Pricing Component Synchronization Integration Flows

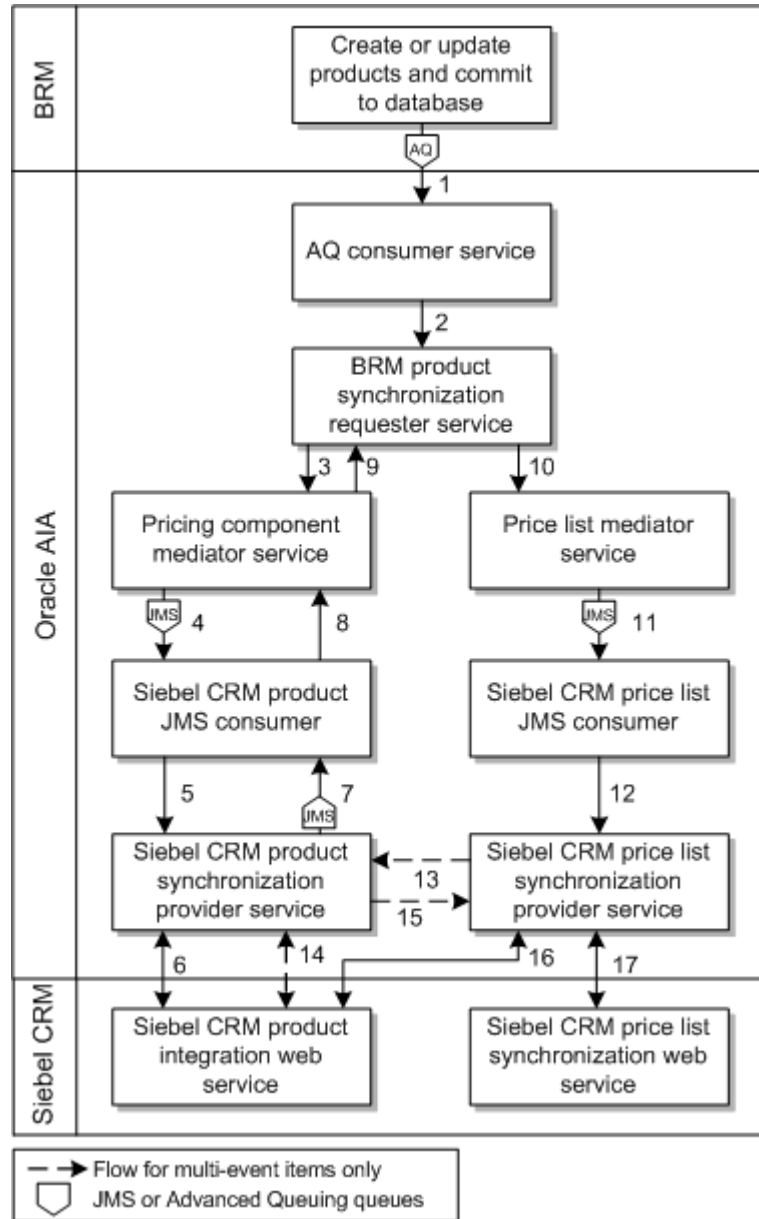
This section describes the integration flows for synchronizing pricing components. It also describes how event charges for products are synchronized from the BRM database to Siebel CRM.

### About the Product Synchronization Integration Flow

Figure 3-3 shows the synchronization of products with pricing details when using Siebel CRM for product enrichment and bundling.



Figure 3-3 Synchronizing Products and Pricing Details When Using Siebel CRM



The integration synchronizes products and pricing details from BRM to Siebel CRM as follows:

1. A BRM user creates or updates products in PDC or Pricing Center and commits them to the BRM database. If real-time product synchronization is disabled, the BRM user runs the batch synchronization utility.  
A BRM business event is generated as an ABM containing the complete definition of the products.
2. The AQ consumer that is subscribed to the BRM business event extracts the product definitions from the product ABM and sends the message to the BRM product synchronization requester service.

3. The BRM product synchronization requester service transforms the BRM-specific product definitions from the ABM into standardized product definitions in an EBM and sends the EBM to the pricing component mediator service.
4. The mediator service checks adds the EBM to the Siebel CRM queue.
5. The JMS consumer subscribed to the Siebel CRM queue picks up the EBM and sends it to the Siebel CRM product synchronization provider service.
6. The Siebel CRM product synchronization provider service transforms the standardized product definitions from the EBM into Siebel CRM-specific product definitions in an ABM and invokes the Siebel CRM web service to create the products.  
The web service returns a success or failure message to the Siebel CRM product synchronization provider service.
7. The Siebel CRM product synchronization provider service transforms the status message into a standardized response EBM and adds it to the Siebel CRM product JMS queue.
8. The Siebel CRM JMS consumer subscribed to the queue picks up the EBM and sends it to the product response mediator service.
9. The product response mediator service sends the EBM to the BRM product synchronization requester service.
10. The BRM product synchronization requester service extracts the pricing information from the product ABM and transforms it into a standardized price list EBM.  
The BRM product synchronization requester service sends the price list EBM to the price list mediator service.
11. The price list mediator service adds the EBM to the Siebel CRM queue.
12. The JMS consumer subscribed to the Siebel CRM queue picks up the EBM and sends it to the Siebel CRM price list synchronization provider service.
13. For price list items with multiple events:
  - a. The Siebel CRM price list synchronization provider service transforms the standardized price list EBM into a standardized product EBM containing definitions of simple products for each event.
  - b. The Siebel CRM price list synchronization provider service sends the product EBM to the Siebel CRM product synchronization provider service.
  - c. The Siebel CRM product synchronization provider service transforms the simple product definitions from the EBM into Siebel CRM-specific product definitions in an ABM and invokes the Siebel CRM web service to create the products.  
The web service returns a success or failure message to the Siebel CRM product synchronization provider service.
  - d. The Siebel CRM product synchronization provider service transforms the response message into a response EBM and sends it to the Siebel CRM price list synchronization provider service.
14. The Siebel CRM price list synchronization provider service invokes the Siebel CRM web service to update the products with pricing definitions.  
The web service returns a success or failure message to the Siebel CRM price list synchronization provider service.
15. The Siebel CRM price list synchronization provider service transforms the standardized price list EBM into a Siebel CRM-specific price list definition in an ABM and invokes the Siebel CRM price list synchronization web service to update the price list with pricing information.

The web service returns a success or failure message to the Siebel CRM price list synchronization provider service.

Product synchronization automatically sets the billable flag in Siebel CRM for all products of billing type Subscription. This 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 as described in *Siebel Communications Guide*.

The integration sends the following product attributes to Siebel CRM:

- Product name
- Product type
- Purchase level
- Description
- Billable events
- Rate plan
- Effective start date
- Effective end date

Rate plan details (charges) go into the price list line and all other attributes go into the product lines.

## About Unspecified Effective Start and End Dates

For the effective start date and effective end date to be unspecified (making the product always effective) 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:**

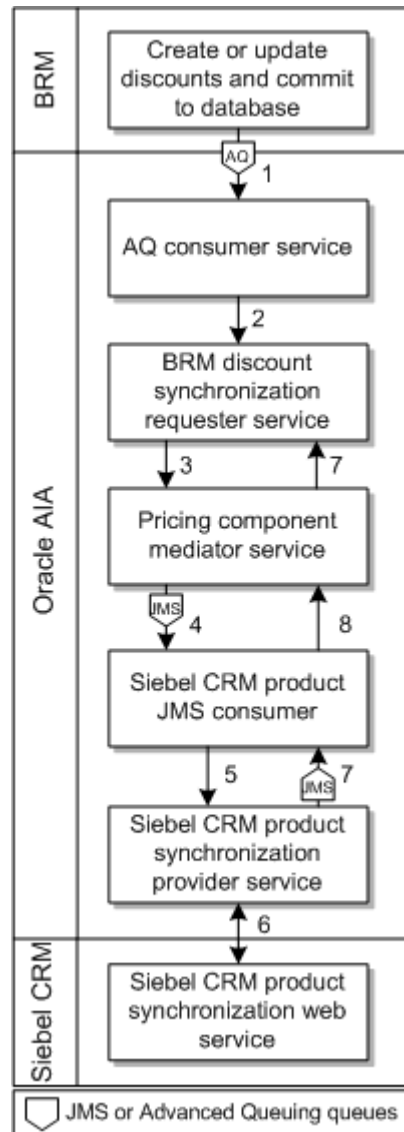
This is a mandatory step as part of the post installation setup activity.

See *BRM Developer's Guide* for more information defining infinite start and end date values.

## About the Discount Synchronization Integration Flow

[Figure 3-4](#) shows the synchronization of discounts when using Siebel CRM for product enrichment and bundling.

Figure 3-4 Synchronizing Discounts When Using Siebel CRM



The integration synchronizes discounts from BRM to Siebel CRM as follows:

1. A BRM user creates discounts in PDC or Pricing Center and commits them to the BRM database. If real-time product synchronization is disabled, the BRM user runs the batch synchronization utility.  
A business event is generated in BRM with a discount ABM containing the complete definition of the discounts.
2. The AQ consumer that is subscribed to the BRM business event extracts the discount definitions from the discount ABM and sends the message to the BRM discount synchronization requester service.
3. The BRM discount synchronization requester service transforms the BRM-specific discount definitions from the ABM into standardized discount definitions in a discount EBM and sends the EBM to the pricing component mediator service.
4. The mediator service adds the EBM to the Siebel CRM queue.

5. The JMS consumer subscribed to the Siebel CRM queue picks up the EBM and sends it to the Siebel CRM product synchronization provider service.
6. The Siebel CRM product synchronization provider service transforms the standardized discount definitions from the EBM into Siebel CRM-specific product definitions in an ABM and invokes the Siebel CRM web service to create the discounts.  
  
The web service returns a success or failure message to the Siebel CRM product synchronization provider service.
7. The Siebel CRM product synchronization provider service transforms the status message into a standardized response EBM and adds it to the Siebel CRM product JMS queue.
8. The Siebel CRM JMS consumer subscribed to the queue picks up the EBM and sends it to the discount response mediator service.
9. The discount response mediator service sends the EBM to the BRM discount synchronization requester service.

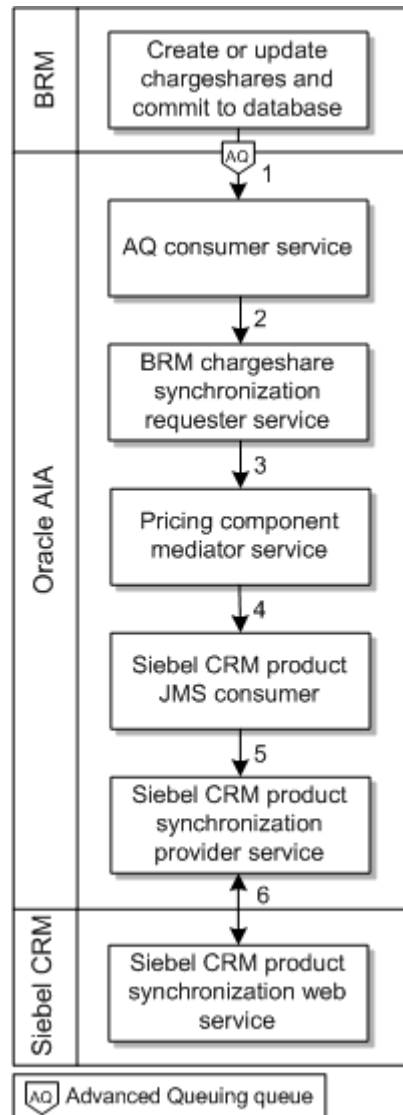
## About the Chargeshare Synchronization Integration Flow

A chargeshare is an object in the BRM database used to let one customer pay for the charges of another customer. You create chargeshares and chargeshare models (called chargeshare offers and chargeshares in PDC) and synchronize the chargeshares to Siebel CRM as simple products. You use these Sponsorship products while modeling promotion groups in Siebel CRM.

See "[About Promotion Groups](#)" for more information about modeling promotion groups in Siebel CRM; see the discussion of setting up charge sharing in *BRM Configuring Pipeline Rating and Discounting* for more information about creating chargeshares.

[Figure 3-5](#) shows the process of synchronizing chargeshares when using Siebel CRM for product enrichment and bundling.

Figure 3-5 Synchronizing Chargeshares When Using Siebel CRM



The integration synchronizes chargeshares from BRM to Siebel CRM as follows:

1. A BRM user creates or edits chargeshares and commits them to the BRM database. If real-time product synchronization is disabled, the BRM user also runs the batch synchronization utility.  
The integration raises a business event in BRM with a chargeshare ABM containing the complete definition of the chargeshares.
2. The AQ consumer that is subscribed to the BRM business event extracts the chargeshare definitions from the chargeshare ABM and sends the message to the BRM chargeshare synchronization requester service.
3. The BRM chargeshare synchronization requester service transforms the BRM-specific chargeshare definitions from the ABM into standardized chargeshare definitions in a product EBM and sends the EBM to the pricing component mediator service.
4. The discount mediator service adds the EBM to the Siebel CRM queue.

5. The JMS consumer subscribed to the Siebel CRM queue picks up the EBM and sends it to the Siebel CRM product synchronization provider service.
6. The Siebel CRM product synchronization provider service transforms the standardized chargeshare definitions from the EBM into Siebel CRM-specific product definitions in a product ABM. Then it calls the Siebel CRM product synchronization web service, which creates a simple product in Siebel CRM for each chargeshare.

The web service returns a status message (Success or Fail) to the Siebel CRM product synchronization provider service.

Chargeshare synchronization automatically sets the Billable flag for all chargeshares (Siebel CRM simple products of billing type Sponsorship).

## About Synchronizing Product Usage Charges

When you create products in BRM, you associated them with billable events that determine how much and how often to charge customers. You associate each product with at least one billable event.

Products associated with a single event are synchronized to Siebel CRM as simple products and products associated with multiple events are synchronized as customizable products.

Products are synchronized from BRM to Siebel CRM as follows:

- If a product has only one event, then the product is synchronized with Siebel CRM as a simple product with no list price.
- If a product has two events and one of them is a usage charge event, then the product is synchronized to Siebel CRM as a simple product. The usage charge event is not synchronized to Siebel CRM. The list price of the simple product in Siebel CRM is set to charge on the other event of the product.
- If a product in BRM has more than two events and one event is a usage charge event then the 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 product.

[Table 3-1](#) shows examples of how products in BRM are synchronized to Siebel CRM.

**Table 3-1 Example of BRM Products Synchronized to Siebel CRM**

| Product in BRM   | Product in Siebel CRM  |
|--|--|
| Wireless Usage (single usage charge event) <ul style="list-style-type: none"> <li>• Delayed Telco GSM Session Even - \$0.40</li> </ul>   | Wireless Usage (simple product with no list price)   |
| Call Forwarding (two events, one usage charge) <ul style="list-style-type: none"> <li>• Monthly Cycle Forward Event - \$3.00</li> <li>• Delayed Telco GSM Session Event - \$.040</li> </ul>  | Call Forwarding - \$3.00 (simple product with list price of cycle forward event)   |
| Internet (multiple events, one usage charge) <ul style="list-style-type: none"> <li>• Product Purchase Fee Event - \$10.00</li> <li>• Monthly Cycle Forward Event - \$20.00</li> <li>• Delayed Telco GSM Session Event - \$0.40</li> </ul> | Internet - \$20.00 (customizable product) <ul style="list-style-type: none"> <li>• Product Purchase Fee Event - \$10.00</li> </ul> |

Oracle AIA maps BRM events as shown in [Table 3-2](#). Only events listed in the PRICETYPE\_EVENT domain value map (DVM) are synchronized. You can add more events to this DVM as required by your solution.

**Table 3-2 Oracle AIA Event Mapping**

| 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            |

See "[About One-Time Charges for Activating and Changing Services](#)" 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

A price list is a set of standard prices for products and services. It specifies the price, currency, and frequency with which a price is charged. You can create multiple price lists to charge different customers different prices for the same product.

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.

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.

Price lists map to rate plans in BRM (called charges in PDC). You define rate plans when creating products to specify what to charge for the products. BRM also has a price list entity, but this is different from the Siebel CRM price list. When this guide refers to price lists, it is referring to the Siebel CRM entity.

In Siebel CRM:

- You create price lists before product synchronization, specifying a default price list and other price lists to offer separate prices for the same product.
- You map price lists to rate plans in the Oracle AIA PRICELIST domain value map (DVM). See "[Configuring Siebel CRM for Integrated Product Lifecycle Management](#)" for more information about setting up this mapping.
- You select which price list applies to which account.

### 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 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 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 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 Price Lists

You associate rate plans in BRM with price lists by using a rate plan selector in Pricing Center or a charge selector in PDC.

To associate a rate plan with a price list, perform one of the following procedures:

- [Using Pricing Center to Associate Rate Plans](#)
- [Using PDC to Associate Rate Plans](#)

### Using Pricing Center to Associate Rate Plans

To associate rate plans with price lists by using Pricing Center, follow the steps for defining rate plan selectors described in the Pricing Center Help and do the following when setting up columns for your rate plan selector:

1. Create a column named **EVENT.PIN\_FLD\_USAGE\_TYPE**.
2. Add a row for each rate plan and corresponding price list that you intend to use.
3. 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.  
Enter the name 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.
  - Associate a rate plan with the default price list by entering an asterisk (\*). The integration maps the asterisk to the default price list. See [Table 3-4](#) for an example.
4. 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.

5. Finish defining the rate plan selector as described in the Pricing Center Help.

### Using PDC to Associate Rate Plans

To associate rate plans with price lists by using PDC, follow the steps for creating charge selectors described in the PDC Help and do the following when selecting attributes for charge selector rules:

1. In the Select Fields for Rules section, click the **Add** icon.  
The Select Field dialog box appears.
2. Expand **Event** and then select **Usage Type**.
3. Click **OK**.  
A usage type field is added to the table.
4. Confirm that the following columns contain the corresponding values:
  - In the Value Type column, **Single Value**.
  - In the Operator column, **Equals**.
  - In the Default column, **\***.
5. Click **Next**.
6. In the Rules section, add a rule for each rate plan and corresponding price list that you intend to use by clicking the **Add** icon.
7. Enter values in the following columns:
  - In the Rule Name column, enter a name for the charge.
  - In the USAGE\_TYPE column, do one of the following:
    - To associate a charge with a specific price list, enter the name of the price list. If you are using Siebel CRM, enter the name 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 charge with the default price list, enter **\***. The integration maps **\*** to the default price list.
8. Finish defining the charge selector as described in the PDC Help.

## Rate Plan Example: Multiple Lines of Business and Regions

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 3-3](#) 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 3-3 Rate Plan Structure for Broadband Product in Multiple Lines of Business**

| Rate Plan Name | Price List Associated with the Rate Plan | Tier | Start Date | End Date   | Monthly Cycle Forward Fee |
|----------------|--|------|------------|------------|---------------------------|
| ConsumerRP     | ConsumerPL                               | 1    | 01/01/2016 | 12/31/2016 | US\$40                    |
| BusinessRP     | BusinessPL                               | 1    | 01/01/2016 | 12/31/2016 | US\$30                    |

Table 3-4 shows the rate plan structure for the GSM product. For this product, the pricing administrator uses two 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 3-4 Rate Plan Structure for GSM Products in Multiple Regions**

| Rate Plan Name | Associated Price List | Tier | Start Date | End Date   | Monthly Cycle Forward Fee |
|----------------|-----------------------|------|------------|------------|---------------------------|
| NewYorkRP      | NewYorkPL             | 1    | 01/01/2016 | 12/31/2016 | US\$45                    |
| CaliforniaRP   | CaliforniaPL          | 1    | 01/01/2016 | 12/31/2016 | US\$40                    |
| StatesRP       | *                     | 1    | 01/01/2016 | 12/31/2016 | 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 the price lists in the PRICELIST DVM.
2. In Pricing Center, create rate plans that use the same currencies as the price lists in Siebel CRM.
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 products to the BRM database, which synchronizes the products to Siebel CRM.

## Rate Plan Example: Multiple Currencies

To offer a product named 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 3-5](#) when creating the product.

**Table 3-5 Rate Plan Structure for Broadband 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    | 01/01/2016 | 12/31/2016 | CAD\$30                   |
| USARP          | USAPL                                    | 1    | 01/01/2016 | 12/31/2016 | 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 (the charges) 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 in Pricing Center or charge in PDC
- Changing the price list associated with a product's rate plan or charge by updating the rate plan selector or charge selector
- 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:

- In Pricing Center:
  - 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.
- In PDC:
  - Change the price in the existing charge by changing the balance impact.  
See the discussion of configuring balance impacts for charges in the PDC Help for more information.

- Add a new charge with the new price and adjust the effective dates of the old charge.  
See the discussion of adding date ranges to a charge in the PDC 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 3-3](#).

[Table 3-6](#) 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 3-6 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    | 01/01/2016 | 12/31/2016 | US\$35                    |
| BusinessRP     | BusinessPL                               | 1    | 01/01/2016 | 12/31/2016 | US\$30                    |

[Table 3-7](#) 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 effective dates of the old tier.

**Table 3-7 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        | 01/01/2016        | <b>02/31/2016</b> | US\$40                    |
| <b>ConsumerRP</b> | <b>ConsumerPL</b>                        | <b>2</b> | <b>02/31/2016</b> | <b>12/31/2016</b> | <b>US\$35</b>             |
| BusinessRP        | BusinessPL                               | 1        | 01/01/2016        | 12/31/2016        | 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, do one of the following:

- If you are using Pricing Center:
  1. For the rate plan that is associated with the old price list, set the duration end date to the current date. 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 (for Siebel CRM).  
If you enter a name that does not appear in the PRICELIST DVM, an error occurs during product synchronization to Siebel CRM.
  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.
- If you are using PDC:
    1. For the charge associated with the old price list, set the end time of the effective date range to the current date.
    2. For the charge selector associated with the charge offer, add the same charge as a new charge in the charge selector rule.
    3. In the Rules section for the charge selector, add a new rule for the new price list.
    4. Commit the charge offer to the BRM database, which resynchronizes the charge offer 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 3-3](#), the BRM pricing administrator uses Pricing Center to edit the rate plan structure. As shown in [Table 3-8](#), 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 3-8 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        | 01/01/2016        | <b>02/31/2016</b> | \$40                      |
| <b>ConsumerRP</b> | <b>ConsumerPlusPL</b>                    | <b>1</b> | <b>02/31/2016</b> | <b>12/31/2016</b> | <b>\$40</b>               |
| BusinessRP        | BusinessPL                               | 1        | 01/01/2016        | 12/31/2016        | \$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 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 are 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.

### Example of Changing from Multiple Price Lists to the Default Price List

To change the GSM product with the rate plan structure shown in [Table 3-4](#) 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 3-9](#).

**Table 3-9 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    | 01/01/2016 | <b>02/31/2016</b> | \$45                      |
| CaliforniaRP   | CaliforniaPL                             | 1    | 01/01/2016 | <b>02/31/2016</b> | \$40                      |
| StatesRP       | *  | 1    | 01/01/2016 | 12/31/2016        | \$35                      |

- Commits the product to the BRM database to resynchronize the product to Siebel CRM and update the effective 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 3-3](#) 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 3-10](#).

**Table 3-10 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    | 01/01/2016 | <b>02/31/2016</b> | \$40                      |
| BusinessRP     | BusinessPL                               | 1    | 01/01/2016 | <b>02/31/2016</b> | \$30                      |

- Commits the product to the BRM database to resynchronize the product to Siebel CRM and updates the effective 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 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 bundling pricing components synchronized from BRM to Siebel CRM.

### Basic Entity Mappings

[Table 3-11](#) shows the mapping between BRM and Siebel CRM entities.

**Table 3-11 Mapping Between BRM and Siebel CRM Entities**

| BRM Entity   | Siebel CRM Entity                               | Origin                    | Description  |
|--|---|---------------------------|--|
| Product with single event<br>(In PDC, charge offer)    | Simple product<br>(automatically created)       | Design time in BRM        | 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<br>(In PDC, charge offer) | Customizable product<br>(automatically created) | Design time in BRM        | 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<br>(automatically created)       | Design time in BRM        | In Siebel CRM, each recurring and nonrecurring event binding is synchronized as a simple product.                      |
| Discount<br>(In PDC, discount offer)                   | Simple product<br>(automatically created)       | Design time in BRM        | A discount is synchronized as a simple product regardless of the number of event bindings.                             |
| Chargeshare<br>(In PDC, chargseshare offer)            | Simple product<br>(automatically created)       | Design time in BRM        | A chargseshare is synchronized as a simple product of billing type Sponsorship.  |
| Balance impact   | Price list line<br>(automatically created)      | Design time in BRM        | Rate Plan, Rate Tier, and Balance Impact in BRM are mapped to price list lines in Siebel CRM.                          |
| Deal<br>(In PDC, bundle)                               | Service bundle (manually created)               | Design time in Siebel CRM | Deals are not synchronized from BRM. Products must be bundled manually in Siebel CRM.                                  |



**Table 3-11 (Cont.) Mapping Between BRM and Siebel CRM Entities**

| BRM Entity   | Siebel CRM Entity                                      | Origin  | Description   |
|--|--|---|---|
| Plan<br>(In PDC, package)                          | Promotion /marketing bundle (manually created)         | Design time in Siebel CRM   | Plans are not synchronized from BRM. Promotions and marketing bundles must be created manually in Siebel CRM.   |
| Service instance                                   | Service bundle asset (automatically created)           | Order time  | In Siebel CRM, 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) | Order time  | In Siebel CRM, purchasing optional and mandatory components of a service bundle results in asset components that are mapped to BRM purchased products.  |
| Sharing group                                      | Promotion group (manually created)                     | Sharing group: order time<br>Promotion group: design time in Siebel CRM | You model promotion groups at design time in Siebel CRM. For each reward product in a promotion group at order time, a separate sharing group is created in BRM.                                    |
| Subscription group<br>(In PDC, subscription group) | Service bundle with Service Grouping attribute enabled | Order time  | For each service bundle on an order with the Service Grouping attribute enabled, the integration creates a subscription group with any nested service bundles as subscription group members in BRM. |

## 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 indicate the usage so that the product or price administrator can distinguish the synchronized products in Siebel CRM.
- Because the discount value of the BRM discount objects is not synchronized to Siebel CRM, the name and description of the discount objects should indicate the general intent of the discount to be conveyed on the order.
- The discountable flag on 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 pricing components 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 *BRM Developer's Guide* for more information about setting this property.

## Using Fixed Amounts or Scaled Amounts in BRM

In BRM, the charge associated with a billable event can be one of the following:

- **Scaled amount:** Specifying the scaled amount allows price overrides and discounts to be applied on the price. In Pricing Center, 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 overrides only override the scaled amount. Price overrides applied to fixed amounts are added to the price. For example, if the fixed amount on the charge is \$5 and a price override is \$10 then the price is \$15.

When defining balance impacts in Pricing Center, you distinguish scaled and fixed amounts by entering values in the **Scaled Amount** or **Fixed Amount** field. When defining balance impacts in PDC, you distinguish scaled and fixed amounts by selecting an option in the **Per Unit** column.

When 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 product has a monthly cycle fee with a scaled amount of \$20 and a fixed amount of \$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 product has a monthly cycle fee with a scaled amount of \$20 and a fixed amount of \$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, Oracle recommends 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 the discussion of real-time rate plans in *BRM Setting Up Pricing and Rating* for more information about using fixed and scaled amount fields.

## About Creating Physical Goods

You can create physical goods directly in BRM as pricing components or you can define physical goods in Enterprise Resource Planning (ERP) and manually synchronize them to BRM. Oracle AIA synchronizes the pricing components that represent the physical goods from BRM to Siebel CRM.

## About Adding Products to Siebel CRM 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 3-12](#) shows examples of products that would be included in the Siebel Catalog.

**Table 3-12 Example of Products Included in the Siebel Catalog**

| BRM Entities  | Siebel Synchronized Entities | Siebel Catalog  |
|---|------------------------------|---|
| Product: Wireless (Yearly)<br>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)<br>Event: MCF - \$40<br>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<br>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<br>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<br>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 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 products of type item and subscription with one-time or recurring charges in BRM.

See "[Using Fixed Amounts or Scaled Amounts in BRM](#)" for more information.

- The Product Management integration maintains cross-reference information between BRM products and Siebel CRM products. If you delete a product in BRM that is synchronized with Siebel CRM, then the cross-reference data for that 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 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 BRM

You must define customizable discounts that are time-based, impact noncurrency resources, or impact multiple event types in BRM as account 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.

Discounts defined in BRM are synchronized to Siebel CRM as simple products. 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

You can define matrix or promotional discounts in Siebel CRM when you bundle the products into service bundles and promotions. If these discounts are included in an order, there will be a difference between the start or list price and the net price.

## Defining Overrides on the Product Definition

Siebel CRM products include attributes that let you determine how pricing differences between the list price and the selling price are communicated to BRM.

For Siebel CRM, use the following fields on the product definition:

- **Pricing commit type:** Indicates whether a price override or a discount override is being defined on the product. Possible values are:
  - **Committed:** Indicates that a price override has been defined on the product.
  - **Dynamic:** Indicates that a discount override has been defined on the product.
- **Dynamic discount method:** Identifies the discount type when a discount override has been defined. Possible values are:
  - **Amount**
  - **Percent**

In BRM, discount overrides can be tracked in a separate sub-bucket within the general ledger code that is tied to the product. Because the list price on the product remains the same, you can use discount overrides to apply mass price changes.

## About Service Bundles

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

- **Products:** BRM products synchronized to Siebel CRM as simple or customizable products.
- **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 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.

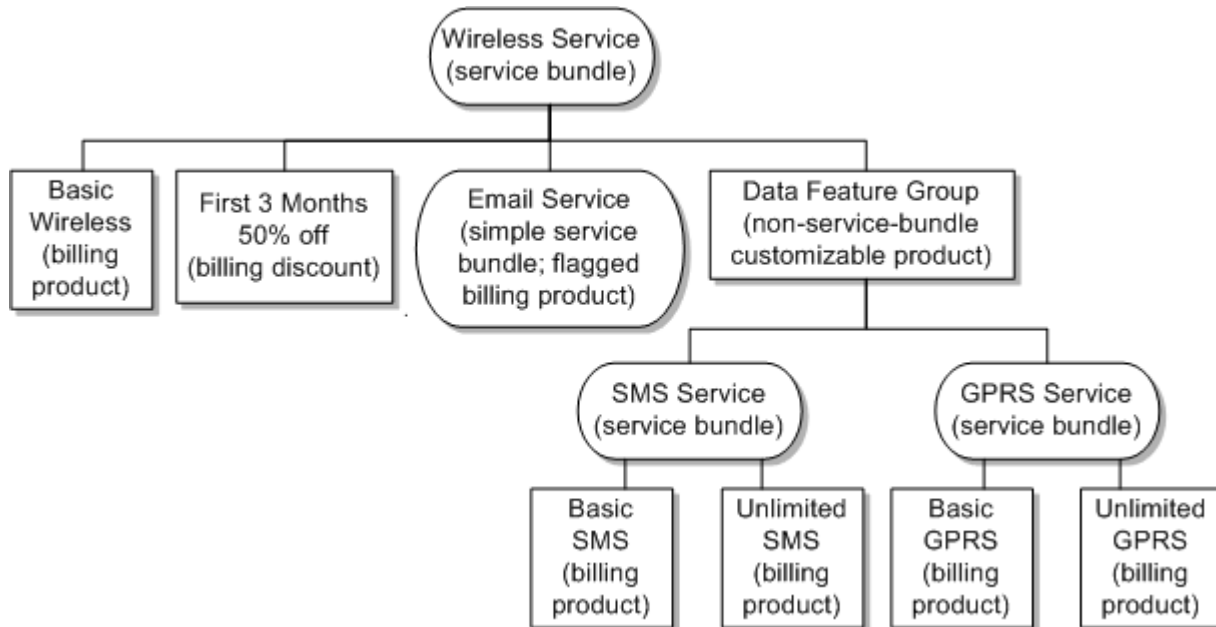
You can nest products and discounts within another product or discount, but the integration synchronizes products or discounts nested more than two levels below a service bundle at the account level when they are purchased on a Siebel CRM order. See "[Supporting Product Bundling](#)" for more information 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 products synchronized from BRM as simple service bundles. See "[About Simple Service Bundles](#)" for more information.

[Figure 3-6](#) shows an example of the hierarchy in Siebel CRM for a service bundle that contains products, a discount, a simple service bundle, and a non-service-bundle customizable product.

Figure 3-6 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 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.

You can model a simple service bundle by itself, nest it within another service bundle, or 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, you set the Service Instance attribute for that product. The integration does not set, update, or overwrite the flag when products are created or synchronized; you must set it manually.

In Siebel CRM, you set the Service Instance attribute to **Y**.

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

## Example of Using a Service Bundle or a Simple Service Bundle

Table 3-13 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 3-13 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)<br><b>Note:</b> 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 | <b>Basic High Speed Internet MCF (SBC)</b><br><b>Premium High Speed Internet MCF (SBC)</b><br><b>Elite High Speed Internet MCF (SBC)</b>                           | Only 1 of these 3 is selected | <b>Basic High Speed Internet MCF (SBC)</b><br><b>Premium High Speed Internet MCF (SBC)</b><br><b>Elite High Speed Internet MCF (SBC)</b>                           |
| 2.2                           | ---- Internet Secure Firewall (SBC)  | 2.2                           | ---- Internet Secure Firewall (SBC)  |
| 2.3                           | ---- CP: High Speed Internet Features (NSB-CP)<br><b>Note:</b> 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)<br><b>Note:</b> The NSB-CP is optional; without it the four-feature SBs have the Internet Service SB as the parent. |
| 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)  |

**Table 3-13 (Cont.) Modelling Using Service Bundles or Simple Service Bundles**

| Hierarchy | Service Bundle   | Hierarchy | Simple Service Bundle                                     |
|-----------|--|-----------|---|
| 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 product. They cannot include other products, discounts, special rating products, or service bundles. To combine multiple products and discounts, you must use a regular service bundle.
- Only products of type **Subscription** can be flagged as simple service bundles.
- You cannot apply special rating, such as friends and family rates, to simple service bundles.
- 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. Do not flag a product that is already in a regular service bundle as a simple service bundle. If your product bundling requirements change, you must create a different product in BRM, synchronize it to Siebel CRM, and bundle it differently.
- Although you can nest a simple service bundle within a regular service bundle at design time, this does not make the simple service bundle a component of the regular service bundle at order time. When processing an order containing a simple service bundle nested within a regular service bundle, the integration creates a separate service instance in BRM for each service bundle and each service is billed separately. If you want the service represented by the simple service bundle to be a component of the regular service bundle, you must create a new product in BRM to represent the service, synchronize it to Siebel CRM, and bundle it as part of the regular service bundle.
- 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 Service Grouping

Service grouping in Siebel CRM lets you group service bundles that apply to a particular device.

Using service grouping, your customers can track and bill all the services that apply to a particular device together in BRM. For example, you could group the services that apply to a wireless phone, such as voice, messaging, and data. You could then track balances and create bills for that group of services separately from other services and groups.

To support service grouping, at design time in Siebel CRM, create a Service Grouping attribute and enable it for the service bundles that you want to group. When you place an order that includes these service bundles, the integration creates subscription groups in BRM (called service groups in PDC).

### Note:

You use service grouping when creating bundled promotions for family share plans. The integration does not support service grouping in other contexts.

See the following topics for more information about service grouping:

- See "[About Family Share Plans](#)" for more information about using service grouping when creating family plans.
- See "[Enabling Service Grouping at Design Time](#)" for information about creating the Service Grouping attribute.
- See "[Synchronizing Service Grouping](#)" for information about how the integration uses the Service Grouping attribute to create subscription groups in BRM.
- See the discussion of grouping services by subscription in *BRM Managing Customers*.

## Enabling Service Grouping at Design Time

To enable service grouping, do the following at design time in Siebel CRM as described in *Siebel Product Administration Guide*:

1. Create an attribute definition using the values in [Table 3-14](#).

**Table 3-14 Attribute for Service Grouping**

| Attribute        | Data Type | Domain Type | Values  | Description   |
|------------------|-----------|-------------|---------|---|
| Service Grouping | String    | Enumerated  | Y and N | Indicates whether a service bundle and its components form a subscription group (Y) or not (N). Set the default value to <b>N</b> . |

2. Add the Service Grouping attribute to the product class for the service bundles representing the top level of grouped services.
3. Set the Service Grouping attribute to **Y** for the service bundles that you want to group, including service bundles in bundled promotions.
4. Complete other pricing activities and release the service bundles for purchase.

 **Note:**

You can disable the Service Grouping attribute on new orders, but you cannot disable the attribute on change orders. After an order is fulfilled, you cannot ungroup the services.

## About Promotion Groups

Promotion groups are groups of services in Siebel CRM that share benefits.

For example, using promotion groups, a company can pay for some of its employees' mobile phone usage and offer a discounted rate for calls between employees.

You create promotion groups in Siebel CRM. Each promotion group definition consists of the following associated components:

- An owner membership product created in Siebel CRM, associated at order time with the account or service that owns the promotion group
- A member membership product created in Siebel CRM, associated at order time with one or more services belonging to accounts other than the promotion group owner
- Discounts and chageshares created in BRM and special rating products created in Siebel CRM, added at design time as rewards to be shared by the group owner and members

## Creating Promotion Groups

Create promotion groups in Siebel CRM as described in the discussion of setting up promotion groups for administrators in *Siebel Order Management Guide Addendum for Communications*, with the following restrictions:

- In Siebel CRM, create two membership products to reuse with all of your promotion group definitions:
  - One with a fulfillment item code of **Group Owner**
  - One with a fulfillment item code of **Group Member**

Although Siebel CRM promotion groups do not distinguish between promotion group owner and member, their representations in BRM do. To maintain the distinction between owner and member in Siebel CRM, use the product name and the fulfillment item code.

You reuse the membership products with different promotion group definitions.

- When you add the membership products to promotion group definitions, use the values listed in [Table 3-15](#) for the **Min**, **Max**, and **Default** fields, where  $N$  and  $M$  are any positive integer you choose.

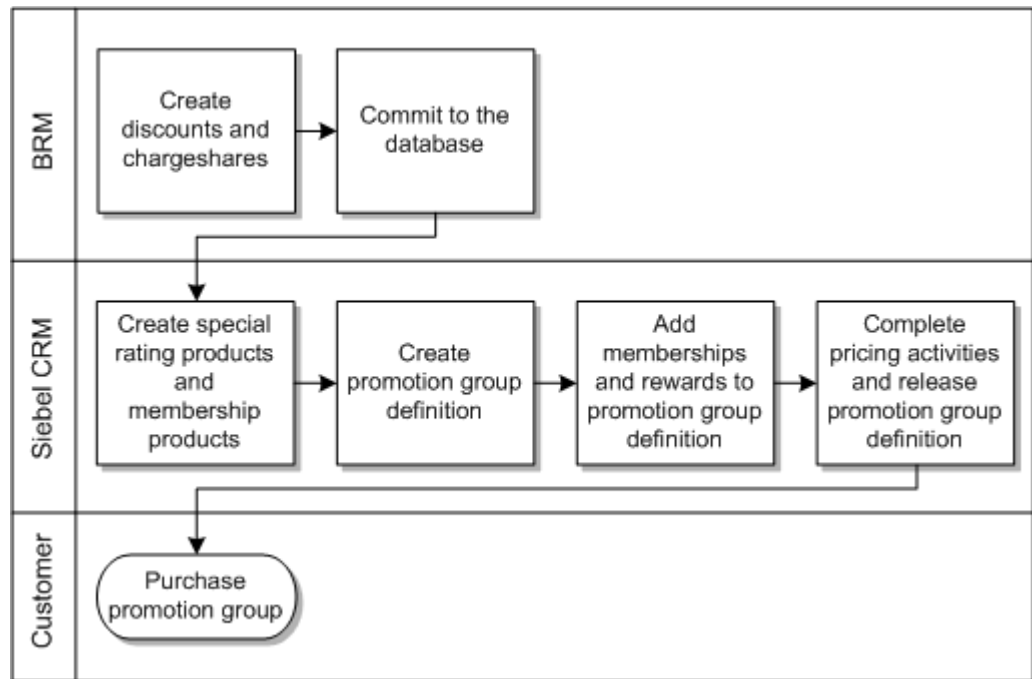
**Table 3-15 Fields for Promotion Group Memberships and Rewards**

| Product           | Min Value | Max Value | Default Value  |
|-------------------|-----------|-----------|--|
| Owner membership  | 1         | 1         | 1  |
| Member membership | 0         | $N$       | Any number between the <b>Min</b> and <b>Max</b> values. |
| Reward            | 0 to 1    | $M$       | Any number between the <b>Min</b> and <b>Max</b> values. |

- Add discounts and chargeshares created in BRM and special rating products created in Siebel CRM as reward products.  
Only products included in promotion group definitions at design time can be used as promotion group rewards at order time. For example, you cannot add a customer's existing discount as a reward at order time.
- (Optional) Add simple products representing a one-time or recurring fees as reward products to charge customers for promotion groups.

Figure 3-7 shows the high-level tasks for modelling promotion groups.

**Figure 3-7 Creating Promotion Groups**



For information about how the integration supports promotion groups at order time, see the following topics:

- [Supporting Promotion Groups on Orders](#)
- [Synchronizing Promotion Groups](#)

## About Family Share Plans

Family share plans are bundled promotions in Siebel CRM that let your self-service customers share discount products.

For example, you could offer a family share plan for a wireless phone line with add-on lines and a shared pool of data. Family share plans are similar to promotion groups but family share plans are simpler to model and order.

Family share plans do not use definitions and membership products like promotion groups. Instead, family share plans use attributes on products and service bundles to indicate resource-sharing relationships.

Use family share plans to share discounts for wireless products only. To share special rating lists or charges, or to share any rewards for broadband or VoIP products, use promotion groups.

## Creating Family Share Plans

To create family share plans:

1. Using products synchronized from BRM, create service bundles to represent the wireless lines that will be the owner and members of the family share plan. You can use a combination of service bundles, nested service bundles that include the Service Grouping attribute, and simple service bundles.

For nested service bundles that include the Service Grouping attribute, use a product class that includes the Service Grouping attribute for the top-level service bundles. Nest service bundles for voice, data, and text service within the service bundles for each line.

2. Create attribute definitions as described in *Siebel Product Administration Guide* for the attributes shown in [Table 3-16](#).

**Table 3-16 Attributes for Family Share Plans**

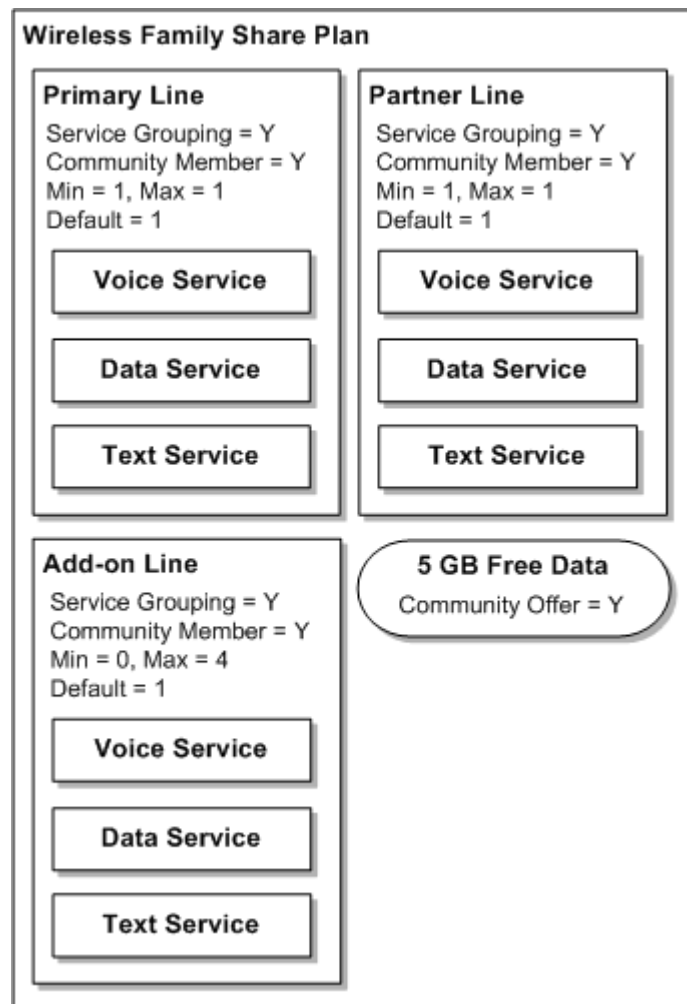
| Attribute        | Data Type | Domain Type | Values  | Description  |
|------------------|-----------|-------------|---------|--|
| Community Offer  | String    | Enumerated  | Y and N | Indicates whether a discount product is a shared reward in a family plan (Y) or not (N). Set the default value to <b>N</b> . |
| Community Member | String    | Enumerated  | Y and N | Indicates whether a service bundle is a family plan member (Y) or not (N). Set the default value to <b>N</b> .               |

3. Add the new attributes to product classes:
  - a. Add the Community Member attribute to the product class for the service bundles representing family share plan lines.
  - b. Add the Community Offer attribute to the product class for the discount products you want to offer as part of the family share plan. Use this attribute with discount products only.
4. Create a product promotion that has the bundled promotion type as described in *Siebel Pricing Administration Guide* and specify the following components:
  - a. The service bundles, simple service bundles, or nested service bundles that include the Service Grouping attribute representing the wireless lines.
  - b. The discount products representing shared resources.
5. Set the attributes related to family share plans as follows:
  - For the service bundles or simple service bundles representing the wireless lines, set the Community Member attribute to **Y**.
  - For any service bundles that you want to group, set the Service Grouping attribute to **Y**.
  - For the discount products representing shared resources, set the Community Offer attribute to **Y**.
6. Complete other pricing activities and release the product promotion for purchase.

Figure 3-8 shows an example of modeling for a family share plan. The figure shows a product promotion called Wireless Family Share Plan that contains the following:

- A service bundle representing the primary line with nested service bundles for voice, data, and text
- A service bundle representing a partner line with nested service bundles for voice, data, and text
- A service bundle representing up to four add-on lines with nested service bundles for voice, data, and text
- A discount product representing 5 GB of free data

**Figure 3-8 Example of Modeling for a Family Share Plan**



Provided you have one primary line and at least one add-on line, you can use different combinations of required or optional add-on lines in a family share plan. For example, if you wanted to charge \$50 for the primary line, \$40 for a second line, and \$10 for each subsequent line, you could use the three service bundles shown in the figure to represent each price point.

See the following topics for information about how the integration supports family share plans at order time:

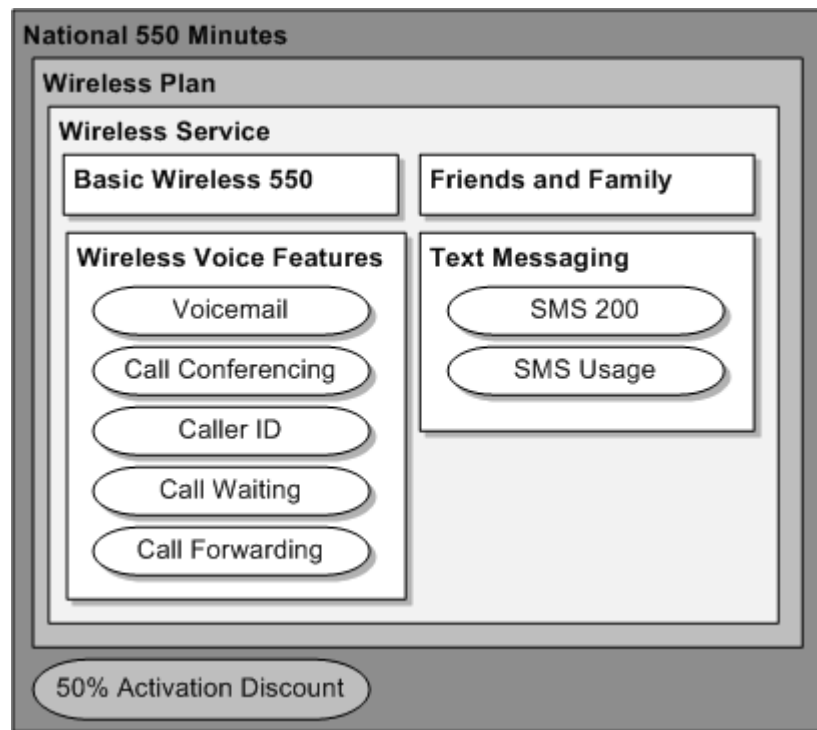
- [Supporting Family Share Plans on Orders](#)
- [Synchronizing Family Share Plans](#)

## 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.

Figure 3-9 shows an example of a marketing bundle for a wireless promotion with text messaging.

**Figure 3-9 Example of a Marketing Bundle for a Wireless Promotion**



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 in Siebel CRM, you can create promotion variants 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 Lists: 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.

## About Credit Limits

Because credit limits are typically defined at the plan level in BRM, and 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 Activating and Changing Services

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 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.

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.

## About Friends and Family Lists

Friends and family lists let you rate calls to certain phone numbers differently than others.

You define rating for friends and family lists in BRM as extended rating attributes, then define special rating products in Siebel CRM and include them in service bundles along with the usage-based subscription product to which they apply.

When a customer purchases a promotion that includes a special rating product, the customer service representative (CSR) associates special rating lists from the customer's special rating profile to the special rating products and adds phone numbers to the lists. After the order is fulfilled and completed, the customer can update the friends and family lists with new phone numbers.

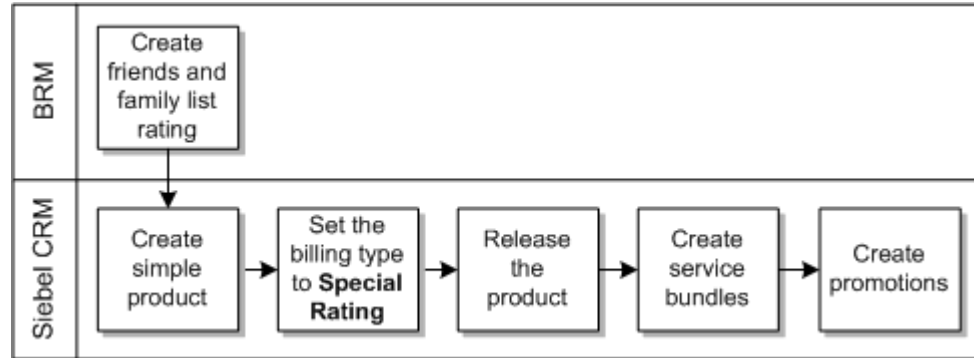
See "[Supporting Friends and Family Lists](#)" and "[Implementing the Synchronize Customer Special Rating Profile Business Flow](#)" for more information about how the lists are created and associated with the list product in BRM at run time.

You use promotion groups to let your customers share friends and family lists. For example, the TruGreen corporation described in "[Example Order to Cash Business Scenarios](#)" would have three special rating lists, one for each office. Rather than creating a special rating list for each TruGreen employee, the corporation can create and maintain one list for each location and share it with employees at that location. See "[About Promotion Groups](#)" for more information about promotion groups.

[Figure 3-10](#) shows the task flow for friends and family lists.



**Figure 3-10 Synchronizing Friends and Family Lists Business Process Task Flow**



## Enabling Friends and Family Lists

To enable friends and family lists:

1. 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 integration billing does not use the BRM Provisioning Tag Framework to support friends and family lists.

See the discussions of working with extended rating attributes and rating based on friends and family ERA in *BRM Setting Up Pricing and Rating* for more information.

2. In the Siebel CRM project workspace, 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.
3. Set the billing type of the product to **Special Rating**.
4. Leave the billing service type blank. This lets you enable friends and family lists using the same special rating product across different types of service.
5. Set the billable flag to **Y**.
6. Set the track as asset flag to **Y**.
7. 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.
8. Include the service bundle in promotions and release all the entities.

See the discussion of profiles and special rating profiles in *Siebel Communications Guide* for more information about friends and family plans in Siebel CRM.

## Product Definition Methodology for Friends and Family Lists: Example

[Table 3-17](#) and [Table 3-18](#) are examples of the product definition methodology.

## BRM Definition

**Table 3-17 BRM Definition**

| Products in BRM  | Service Type                 |
|--|------------------------------|
| Basic Wireless 550<br>----- Monthly Cycle Forward Event<br>----- Delayed Telco GSM Event                                       | /service/telco/gsm/telephony |
| Premium Wireless 800<br>----- Monthly Cycle Forward Event<br>----- Delayed Telco GSM Event                                     | /service/telco/gsm/telephony |
| Unlimited Wireless Voice<br>----- Monthly Cycle Forward Event<br>----- Delayed Telco GSM Event                                 | /service/telco/gsm/telephony |
| Wireless Add On Line<br>----- Monthly Cycle Forward Event<br>----- Delayed Telco GSM Event<br>----- Product Purchase Fee Event | /service/telco/gsm/telephony |
| Wireless Voice Activation<br>----- 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 3-18 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 |

**Table 3-18 (Cont.) Siebel CRM Representation**

| Product Name                 | Service Type                 | Billing type   | Comments         |
|------------------------------|------------------------------|----------------|------------------|
| 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 3-19 contains some examples of the service bundles that include special rating products.

**Table 3-19 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 | --   |
| ----- Wireless Voice Mail            | --   |
| ----- Wireless Call Conference       | --   |
| ----- Wireless Caller ID             | --   |
| ----- Wireless Call Waiting          | --   |
| ----- Wireless Call Forwarding       | --   |
| ----- Text Messaging                 | --   |

**Table 3-19 (Cont.) Service Bundles (SB)**

| Service Bundles                    | Comments   |
|------------------------------------|--|
| ----- 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, Oracle recommends that they be first grouped into a dynamic class and then included in the service bundle.

**Promotions**

Table 3-20 shows examples of the promotion definition.

**Table 3-20 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   |
| Nation 800 Minutes <i>Different Promotion Variant created from the same bundle.</i> |
| ----- Wireless Plan   |
| ----- Wireless Service  |

**Table 3-20 (Cont.) Promotions**

| Promotion Component   |
|---|
| ----- 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 in Siebel CRM, 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 3-21](#):
  - **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 3-21 Product Class Attributes for Time-Based Offerings**

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

 **Note:**

To retrieve the values of the validity attributes in OSM, you must use the same attribute names listed in [Table 3-21](#).

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 run 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

The Synchronize Product and Price business flow operations under the following assumptions and constraints:

- BRM deals and plans (called bundles and packages in PDC) 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 products and discounts is one-way. Products created or updated in Siebel CRM are not synchronized back to BRM.
- Because BRM requires purchase fee discounts to be purchased before the product to which they apply, and the Oracle AIA service that interfaces orders to BRM cannot ensure this sequence, Oracle recommends you use Siebel CRM discounts for discounting purchase fees on products. Based on the pricing commit type, Siebel CRM discounts are 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.

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. 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.

- When a product is deleted in BRM, it does not publish any message. The corresponding product in Siebel CRM is not deleted or inactivated automatically. You must inactivate this product manually in Siebel CRM. If you delete a product in BRM that is synchronized to Siebel CRM, then the cross-reference data for that 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 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 and does not create a corresponding simple product that represents the event (billing type **Event**) in Siebel CRM. No error or notification is sent to indicate that an event was not found in the DVM.
- BRM is the master for usage pricing. When 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. You must not change the pricing for such products in Siebel CRM, for example, by specifying a price override or discount. Prices updated in Siebel CRM are not propagated to BRM or applied when the order is interfaced to billing.
- The product synchronization process ignores the effective start date and effective end date that are specified on the rate tier of the 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.
- The integration supports a single default 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 pricing components 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 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.
- 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.
- 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 Track as Asset flag to **Y** for BRM products of type Subscription and **N** for products of type Item or System.
- 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).

# 4

## Implementing the Synchronize Product and Price Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Synchronize Product and Price business flow.

### About the Synchronize Product and Price Business Flow

The following pre-built integration options enable the Synchronize Product and Price business flow:

- Order to Cash for Oracle Communications Billing and Revenue Management (BRM)
- Order to Cash for Siebel CRM

The Synchronize Product and Price business flow includes the following integration flows:

- Product Synchronization integration flow
- Discount Synchronization integration flow
- Chargeshare Synchronization integration flow

### About the Product Synchronization Integration Flow

The product synchronization integration flow lets you synchronize new or updated products and their prices from BRM to Siebel CRM. You use the products in quotes and orders in Siebel CRM.

The product message from Siebel CRM contains all of the required product and price data. The integration separates the product data from the price data, calls web services to create the products, receives a response from Siebel CRM, then calls web services to add the pricing data.

Products in BRM can have multiple events, each with a price. In Siebel CRM, each product has only one price. 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.

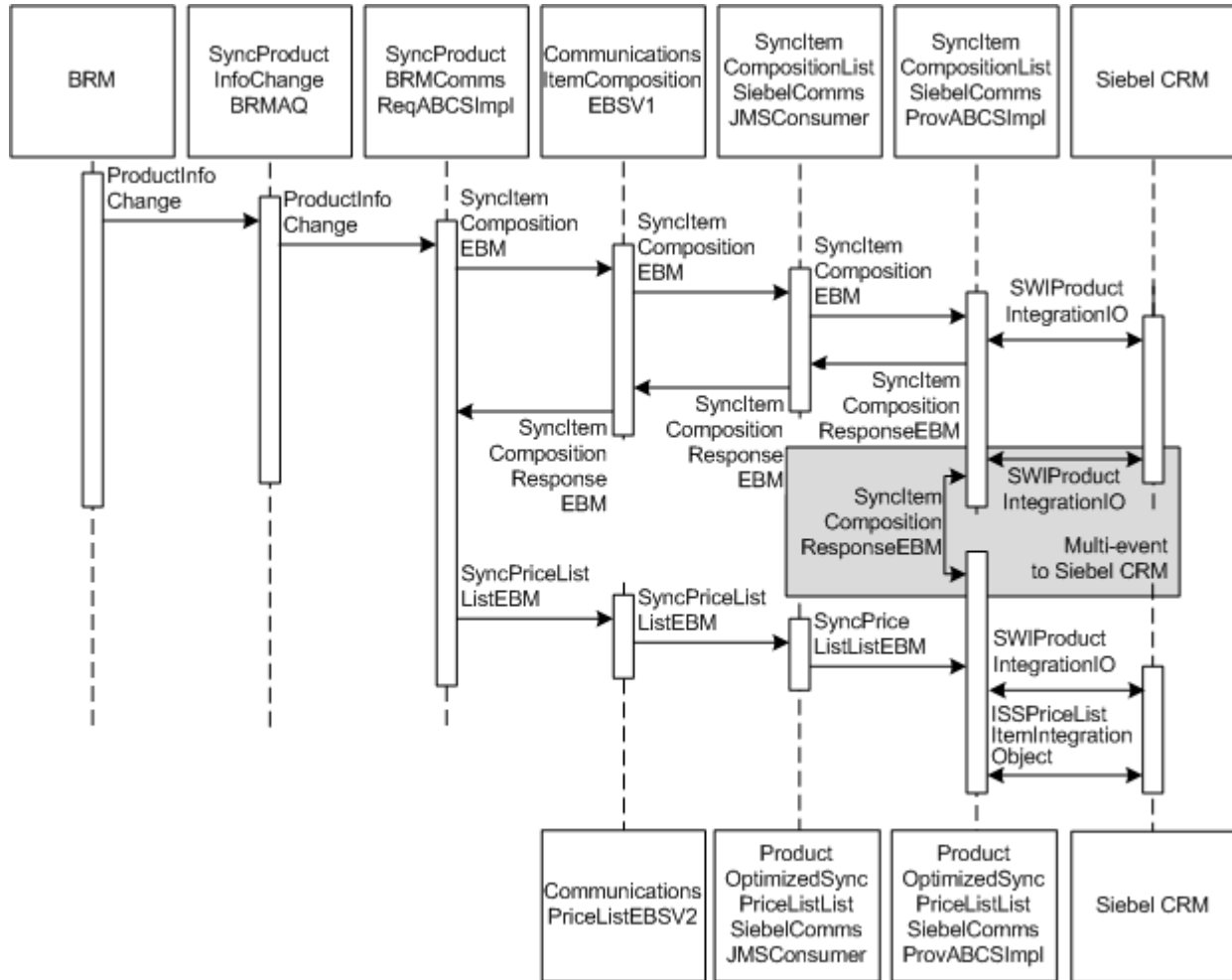
This integration flow includes the following services:

- Integration services for BRM:
  - SyncProductBRMCommsReqABCImpl
  - SyncProductInfoChangeBRMAQ
- Integration services for Siebel CRM:
  - SyncItemCompositionListSiebelCommsProvABCImpl
  - SyncItemCompositionListSiebelCommsJMSConsumer
  - ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

- ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer
- Integration services:
  - CommunicationsItemCompositionEBSV1
  - CommunicationsPriceListEBSV2

Figure 4-1 shows the sequence of integration services through which the product message passes for the Product Synchronization integration flow with Siebel CRM.

**Figure 4-1 Product Synchronization Sequence with Siebel CRM**



The integration synchronizes products as follows:

1. A BRM user creates or edits a product and commits it to the BRM database. The product is added to the BRM product queue as an ABM.
2. The SyncProductInfoChangeBRMAQ adapter, which polls the BRM product queue, picks up the ABM and invokes the SyncProductBRMCommsReqABCSImpl service.
3. The requester service transforms the BRM product message into a SyncItemCompositionListEBM message and sends it to the CommunicationsItemCompositionEBSV1 mediator service.

4. The item mediator service checks adds the request EBM to the Siebel CRM JMS queue.
5. The SyncItemCompositionListSiebelCommsJMSConsumer service picks up the request EBM and sends it to the SyncItemCompositionListSiebelCommsProvABCServiceImpl service.
6. The provider service transforms the item request EBM into the Siebel CRM product message and calls the Siebel CRM product web service.  
The web service creates the products in Siebel CRM and returns a response message to the provider service.
7. The provider service transforms the response message to a SyncItemCompositionResponseEBM and adds it to the Siebel CRM JMS queue.
8. The Siebel CRM JMS consumer service picks up the response EBM and sends it to the item mediator service.
9. The item mediator service sends the response EBM to the BRM requester service.
10. The BRM requester services transforms the item response EBM into the SyncPriceListListEBM message and sends it to the CommunicationsPriceListEBSV2 mediator service.
11. The price list mediator service adds the request EBM to the Siebel CRM JMS queue.
12. The ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer service picks up the request EBM and sends it to the ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl service.
13. For price list items with multiple events:
  - a. The price list provider service identifies the event to be associated with the main product, transforms SyncPriceListListEBM to a SyncItemCompositionListEBM message, and sends it to the SyncItemCompositionListSiebelCommsProvABCServiceImpl service.
  - b. The Siebel item requester service transforms the EBM to a Siebel CRM product ABM and calls the Siebel CRM product web service.  
The Siebel CRM web service completes the request and returns a response message to the provider service.
  - c. The provider service transforms the response to an ItemCompositionResponseEBM message and returns it to ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl.
14. The price list provider service transforms SyncPriceListListEBM to a product ABM and calls the Siebel CRM product web service to update the products with pricing information.  
The web service completes the request and returns a response message.
15. The price list provider service transforms the message to a price list ABM and calls the Siebel CRM price list web service to update the price list with the pricing information.  
The web service completes the request and returns a response message.

## About the Discount Synchronization Integration Flow

The discount synchronization integration flow lets you synchronize new or updated discounts from BRM to Siebel CRM. You use the discounts in quotes and orders in Siebel CRM.

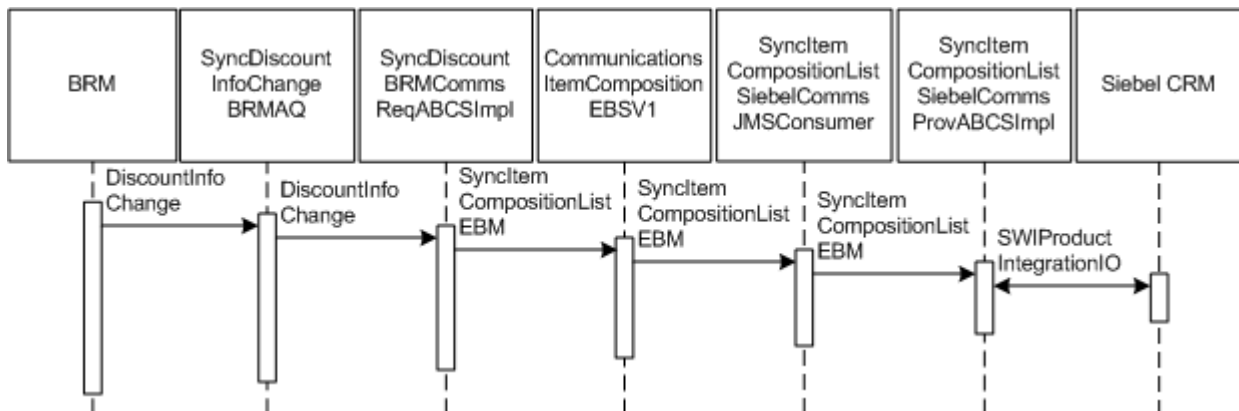
The discount synchronization integration flow synchronizes only the basic discount attributes. It does not synchronize any price information. When you create discounts in BRM, include relevant details in the description.

This integration flow includes the following services:

- Integration services for BRM:
  - SyncDiscountInfoChangeBRMAQ
  - SyncDiscountBRMCommsReqABCImpl
- Integration services for Siebel CRM:
  - SyncItemCompositionListSiebelCommsProvABCImpl
  - SyncItemCompositionListSiebelCommsJMSConsumer
- Integration services:
  - CommunicationsItemCompositionEBSV1

Figure 4-2 shows the sequence of integration services through which the product message passes for the Discount Synchronization integration flow with Siebel CRM.

**Figure 4-2 Discount Synchronization Sequence with Siebel CRM**



The integration synchronizes discounts as follows:

1. A BRM user creates or edits discounts and commits them to the BRM database. The discount definition is added to the BRM discount queue as an application business message (ABM).
2. The SyncDiscountInfoChangeBRMAQ adapter, which polls the BRM discount queue, picks up the ABM and invokes the SyncDiscountBRMCommsReqABCImpl service with the SyncDiscount operation.
3. The requester service transforms the BRM discount message into the SyncItemCompositionListEBM message and sends it to the CommunicationsItemCompositionEBSV1 mediator service.
4. The mediator service adds the item EBM to the Siebel CRM JMS queue.
5. The SyncItemCompositionListSiebelCommsJMSConsumer service picks up the message and sends it to the SyncItemCompositionListSiebelCommsProvABCImpl service.
6. The provider service transforms the SyncItemCompositionListEBM message into the Siebel CRM product message and calls the Siebel CRM product web service. The web service creates the discounts in Siebel CRM and returns a response message to the provider service.

## About the Chargeshare Synchronization Integration Flow

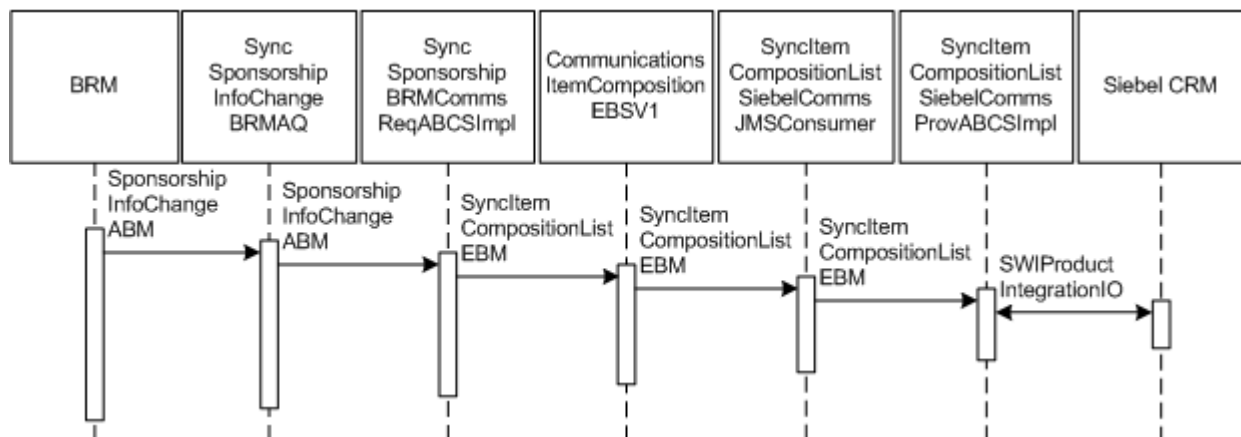
The chargeshare synchronization integration flow lets you synchronize new or updated chargeshares from BRM to Siebel CRM as simple products. In Siebel CRM, you use the chargeshares as rewards when creating promotion groups.

This integration flow includes the following services:

- Integration services for BRM:
  - SyncSponsorshipInfoChangeBRMAQ
  - SyncSponsorshipBRMCommsReqABCImpl
- Integration services for Siebel CRM:
  - SyncItemCompositionListSiebelCommsProvABCImpl
  - SyncItemCompositionListSiebelCommsJMSConsumer

Figure 4-3 shows the sequence of integration services through which the product message passes for the Chargeshare Synchronization integration flow with Siebel CRM.

**Figure 4-3 Chargeshare Synchronization Sequence with Siebel CRM**



The integration synchronizes chargeshares as follows:

1. A BRM user creates or edits chargeshares and commits them to the BRM database. The `SponsorshipInfoChangeABM` message, which contains the definition of the chargeshare, is added to the BRM chargeshare queue.
2. The `SyncSponsorshipInfoChangeBRMAQ` adapter, which polls the BRM chargeshare queue, picks up the message and calls the `SyncSponsorshipBRMCommsReqABCImpl` service with the `SyncSponsorship` operation.
3. The requester service transforms the BRM discount message into the `SyncItemCompositionListEBM` message and sends it to the `CommunicationsItemCompositionEBSV1` mediator service.
4. The mediator service checks adds the item EBM to the Siebel CRM JMS queue.
5. The `SyncItemCompositionListSiebelCommsJMSConsumer` service picks up the message and sends it to the `SyncItemCompositionListSiebelCommsProvABCImpl` service.

6. SyncItemCompositionListSiebelCommsProvABCImpl transforms the SyncItemCompositionListEBM message into the Siebel CRM product message and calls the Siebel CRM SWIProductIntegrationIO web service.

The Siebel CRM web service creates the chargeshares and returns a response message to the SyncItemCompositionListSiebelCommsProvABCImpl service.

## BRM Interfaces

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

- SyncProductInfoChangeBRMAQ: This adapter polls the BRM product queue. It dequeues whenever it sees a message in the queue and invokes SyncProductBRMCommsReqABCImpl with the operation SyncProduct.
- SyncDiscountInfoChangeBRMAQ: This adapter polls the BRM discount queue. It dequeues whenever it sees a message in the queue and invokes SyncDiscountBRMCommsReqABCImpl with the operation SyncDiscount.
- SyncSponsorshipInfoChangeBRMAQ: This adapter polls the BRM chargeshare queue. It dequeues whenever it sees a message in the queue and invokes SyncSponsorshipBRMCommsReqABCImpl with the operation SyncSponsorship.

## 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home/comms\_home/source/soainfra/apps/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](#)
- [SyncSponsorshipBRMCommsReqABCImpl](#)
- [CommunicationsItemCompositionEBSV1](#)
- [SyncItemCompositionListSiebelCommsJMSConsumer](#)
- [SyncItemCompositionListSiebelCommsProvABCImpl](#)
- [CommunicationsPriceListEBSV2](#)
- [ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer](#)
- [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

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 **`AIAConfigurationProperties.xml`** file.

## SyncDiscountBRMCommsReqABCImpl

`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 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.

## SyncSponsorshipBRMCommsReqABCImpl

`SyncSponsorshipBRMCommsReqABCImpl` is a BPEL service and it is the BRM chargeshare requestor ABCS implementation. It has the operation `SyncSponsorship`. This accepts a BRM chargeshare message as a request and does not return a response.

The BRM chargeshare message has basic chargeshare attributes and does not contain any pricing information. The BRM chargeshare message is transformed into the `SyncItemCompositionListEBM` with the basic chargeshare information.

## CommunicationsItemCompositionEBSV1

The `CommunicationsItemCompositionEBSV1` enterprise business service is a mediator service that sends the `SyncItemCompositionListEBM` message to Siebel CRM. It receives the response message from the provider service and returns it to the BRM requester service.

## SyncItemCompositionListSiebelCommsJMSSConsumer

The `SyncItemCompositionListSiebelCommsJMSSConsumer` is a JMS consumer service. It is subscribed to the `AIA_SYNCPROD_SBL_JMSQ` queue into which item EBMs are added.

This consumer picks up `SyncItemCompositionListEBM` and sends it to `SyncItemCompositionListSiebelCommsProvABCImpl`. It also picks up the response EBM and sends it to `CommunicationsItemCompositionEBSV1`.

## SyncItemCompositionListSiebelCommsProvABCImpl

The `SyncItemCompositionListSiebelCommsProvABCImpl` process accepts the `SyncItemCompositionListEBM`. It transforms the EBM into the Siebel CRM product ABM. It then invokes the Siebel CRM Product web service to create products and product structures in Siebel CRM.

This service is 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.

## CommunicationsPriceListEBSV2

The CommunicationsPriceListEBSV2 enterprise business service is a mediator service that sends the SyncPriceListListEBM message to a Siebel CRM provider service. It receives the response message from the provider service and returns it to the BRM requester service.

## ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer

ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer is a JMS consumer service. It is subscribed to the AIA\_SYNCPL\_SBL\_JMSQ queue into which price list EBMs are added.

This consumer service picks up SyncPriceListListEBM and sends it to ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl. It also picks up the response EBM and sends it to CommunicationsPriceListEBSV2.

## ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl

The ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl 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\_spclst\_spcltem\_spclInsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl transforms the PriceListListEBM into a Siebel CRM product message and then calls the Siebel CRM product web service on operation SWIPProductImportUpsert. The Siebel CRM web service completes the request and returns a response message. SWIPProductImportUpsert then transforms the Siebel CRM response message to a PriceListListResponseEBM.

This service is Session Pool Manager enabled.

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

# 5

## Understanding the Query Product Classes Business Flow

This chapter describes the Query Product Classes business flow and explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) enables the process of creating and querying product classes.

### About the Query Product Classes Business Flow

The Query Product Classes business flow lets you reuse the organization of products from Siebel CRM in Oracle Communications Order and Service Management (OSM).

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

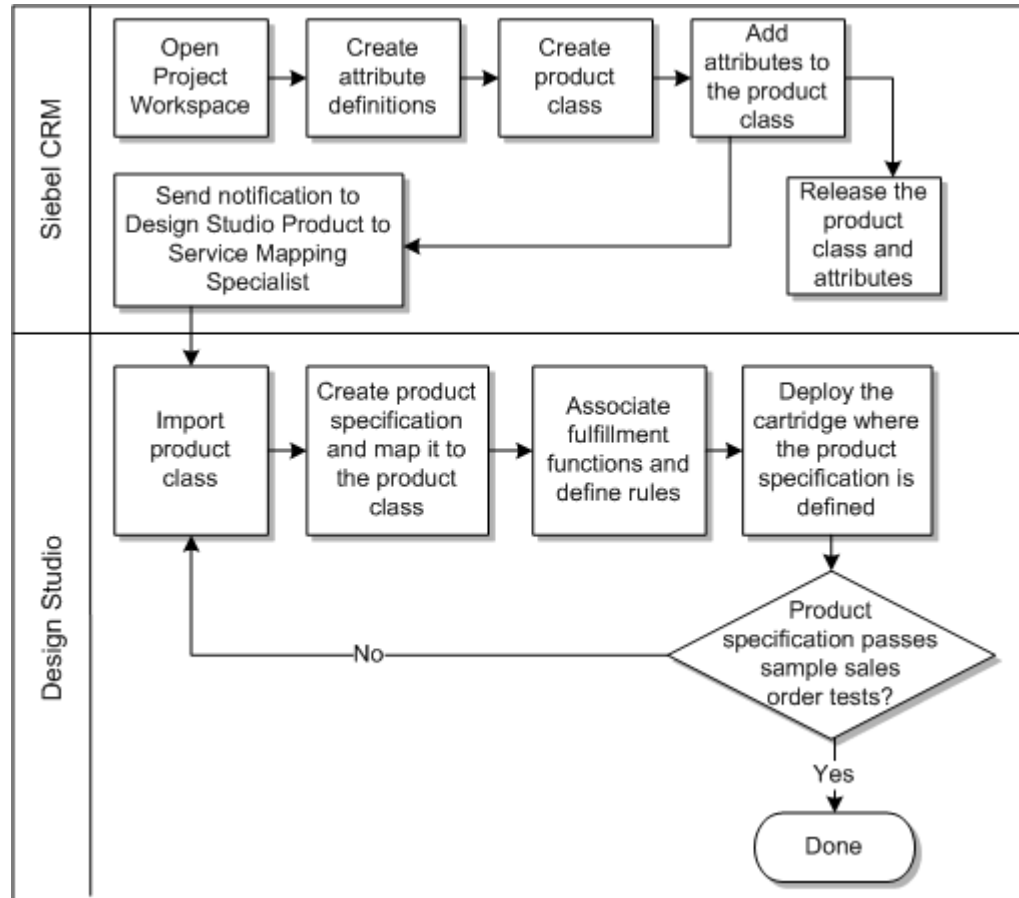
When creating product classes, you associate attributes with them in Siebel CRM. Attributes are defined by sets of values. For example, the value set for a bandwidth attribute might include 2 megabits per second (Mbps), 3 Mbps, and 5 Mbps.

You use Oracle Communications Design Studio to query the product classes and their attributes from Siebel CRM and map the product classes to product specifications in a cartridge. The product specification associate decomposition rules, fulfillment functions, and their dependencies.

After completing all design-time tasks, you deploy the cartridge to OSM for use in order orchestration.

[Figure 5-1](#) illustrates the flow.

Figure 5-1 Query Product Classes Process Flow



The integration queries product classes 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 include it in the query. 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.

 **Note:**

You can create product classes in Siebel CRM that are not used for product specifications. Oracle recommends using a naming convention to distinguish product classes mapped to product specifications from product classes used for other purposes.

For example, prefix product classes mapped to product specifications with **ProdSpec**.

Product classes are only synchronized when Design Studio queries them. Any updates you make to product specifications in Design Studio are not automatically synchronized back to Siebel CRM, and any updates you make to product classes in Siebel CRM are not automatically synchronized to Design Studio. You must resynchronize changed product classes with a new query.

Design Studio maintains the mapping between the Siebel CRM product classes and OSM product specifications.

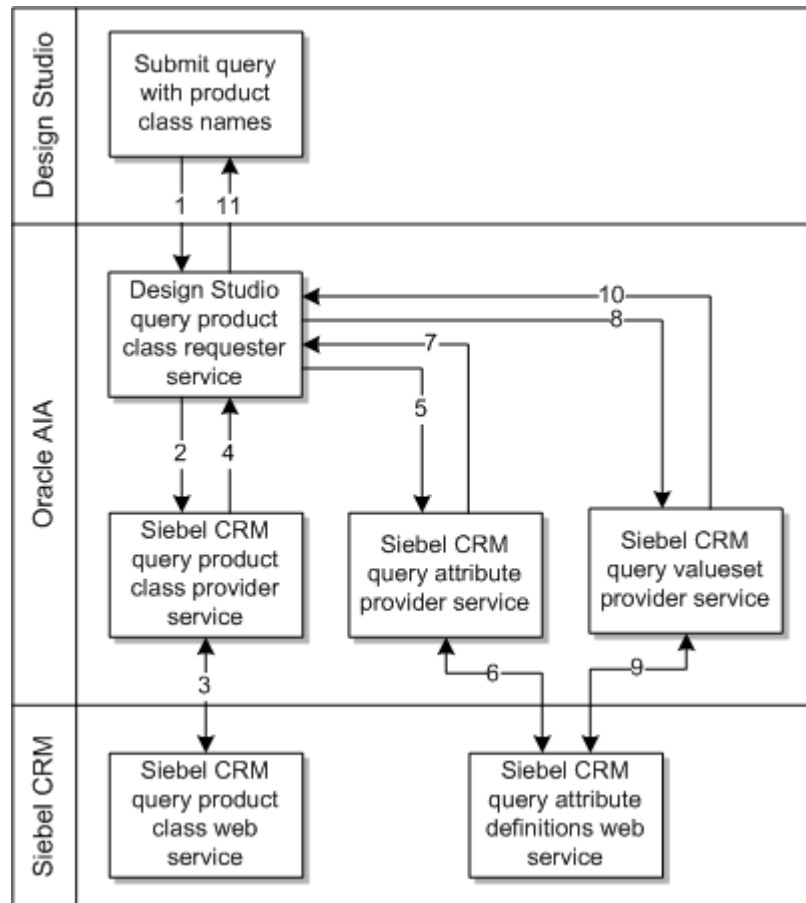
See *Oracle Communications Order and Service Management Concepts* for more information about product classes and specifications in OSM and Design Studio.

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

## About the Query Product Classes Integration Flow

Figure 5-2 shows how the integration synchronizes Siebel CRM product classes to OSM product specifications based on a query from Design Studio.

**Figure 5-2 Querying Product Classes**



The integration queries product classes and their associated attributes as follows:

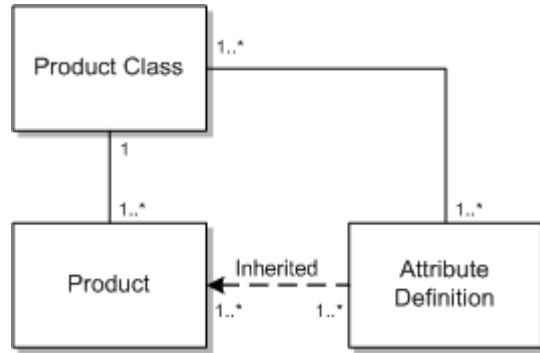
1. A Design Studio user submits a query including all the necessary product class names. Design Studio calls the Design Studio query product class requester service and provides the product class names in a Design Studio-specific application business message (ABM).
2. The Design Studio query product class requester service sends transforms the product class ABM a standardized enterprise business message (EBM) and sends it to the Siebel CRM query product class provider service.
3. The Siebel CRM query product class provider service transforms the product class EBM into a Siebel CRM-specific product class ABM, extracts the product class name, and invokes the Siebel CRM query product class web service.

The web service returns a product class ABM containing complete product class information to the Siebel CRM query product class provider service.

4. The Siebel CRM query product class provider service transforms the Siebel CRM-specific product class ABM into a standardized product class EBM and sends it to the Design Studio query product class requester service.
5. The Design Studio query product class requester service identifies that the product classes have references to attribute definition entities. To query the attribute definitions associated with the product classes, it aggregates the attributes that are associated with the product classes, provides attribute IDs into a standardized attribute EBM and sends it to the Siebel CRM query attribute provider service.
6. The Siebel CRM query attribute provider service transforms the standardized attribute EBM into a Siebel CRM-specific attribute ABM and invokes the Siebel CRM query attribute definition web service.  
  
The web service returns an attribute definition ABM containing the complete attribute information to the Siebel CRM query attribute provider service.
7. The Siebel CRM query attribute provider service transforms Siebel CRM-specific attribute ABM into a standardized attribute EBM and sends it to the Design Studio query product class requester service.
8. The Design Studio query product class requester service identifies that the attributes have references to valueset entities. To query the valuesets associated with the attributes, it aggregates the attributes that are associated with the valuesets into a standardized valueset EBM and sends it to the Siebel CRM query valueset provider service.
9. The Siebel CRM query valueset provider service transforms the standardized valueset EBM into a Siebel CRM-specific attribute definition ABM and invokes the Siebel CRM query attribute definition web service.  
  
The web service returns an attribute definition ABM containing the complete attribute information to the Siebel CRM query valueset provider service.
10. The Siebel CRM query valueset provider service transforms the Siebel CRM-specific attribute definition ABM into a standardized valueset EBM and sends it to the Design Studio query product class requester service.
11. The Design Studio query product class requester service transforms the standardized valueset, attribute, and product class EBMs into a Design Studio-specific product and attribute ABM and sends it to Design Studio.

## About the Logical Data Model in Siebel CRM

In Siebel CRM, you associate attribute definitions with product classes, then you associate product classes with products. The products inherit the attribute definitions from their product class. [Figure 5-3](#) illustrates the logical data model that results from these associations in Siebel CRM.

**Figure 5-3 Logical Data Model of Product Classes in Siebel CRM**

## About Updating the Attribute Valueset

The Siebel CRM 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.

## Updating Product Classes and Effectivity

Product classes are mapped to product specifications and fulfillment patterns in OSM at design time and any updates to the product classes can affect how OSM manages orders at run time.

When you create new product classes, update existing product classes, or deactivate existing product classes in Siebel CRM, you must:

1. Update or specify the product class effectivity.
2. In Design Studio, submit a query for the new or updated product classes.

Any of the following mappings occur:

- The product class is mapped to a new product specification for which you must define new fulfillment metadata.
  - The product class is mapped to an existing product specification.
  - The product class mapping is changed to a different product specification.
3. Update the cartridge version and redeploy the cartridge to OSM. The cartridge deployment date must match the effectivity date of the updated Siebel CRM product classes.

Design Studio bases product specification effectivity on cartridge deployment dates. You must redeploy the OSM cartridge any time the mapping between product classes and product specifications is updated.

All orders already submitted for processing in the OSM system use the previous version of the cartridge. Any subsequent orders, including revision orders, follow-on orders, change orders, and existing orders that are not yet submitted for processing, use the new version of the cartridge. If the effectivity date on the product class is not the same as the cartridge deployment date, OSM uses the cartridge deployment date for effectivity.



Oracle recommends grouping product class changes and setting effectivity for a time that is suitable for deploying a new cartridge version, such as an off-peak hour.

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.

# 6

## Implementing the Query Product Classes Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Query Product Classes business flow.

### About the Query Product Classes Business Flow

The following pre-built integration options enable the Query Product Classes business flow:

- Order to Cash for Oracle Communications Order and Service Management (OSM)
- Order to Cash for Siebel CRM

If you are using Oracle Product Hub to manage your products, the following additional pre-built integration options enable the business flow:

- 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.

The Query Product Classes business flow includes the following integration flow:

- Query Product Classes integration flow: Imports product classes and attributes from Siebel CRM into Design Studio.

### About the Query Product Classes Integration Flow

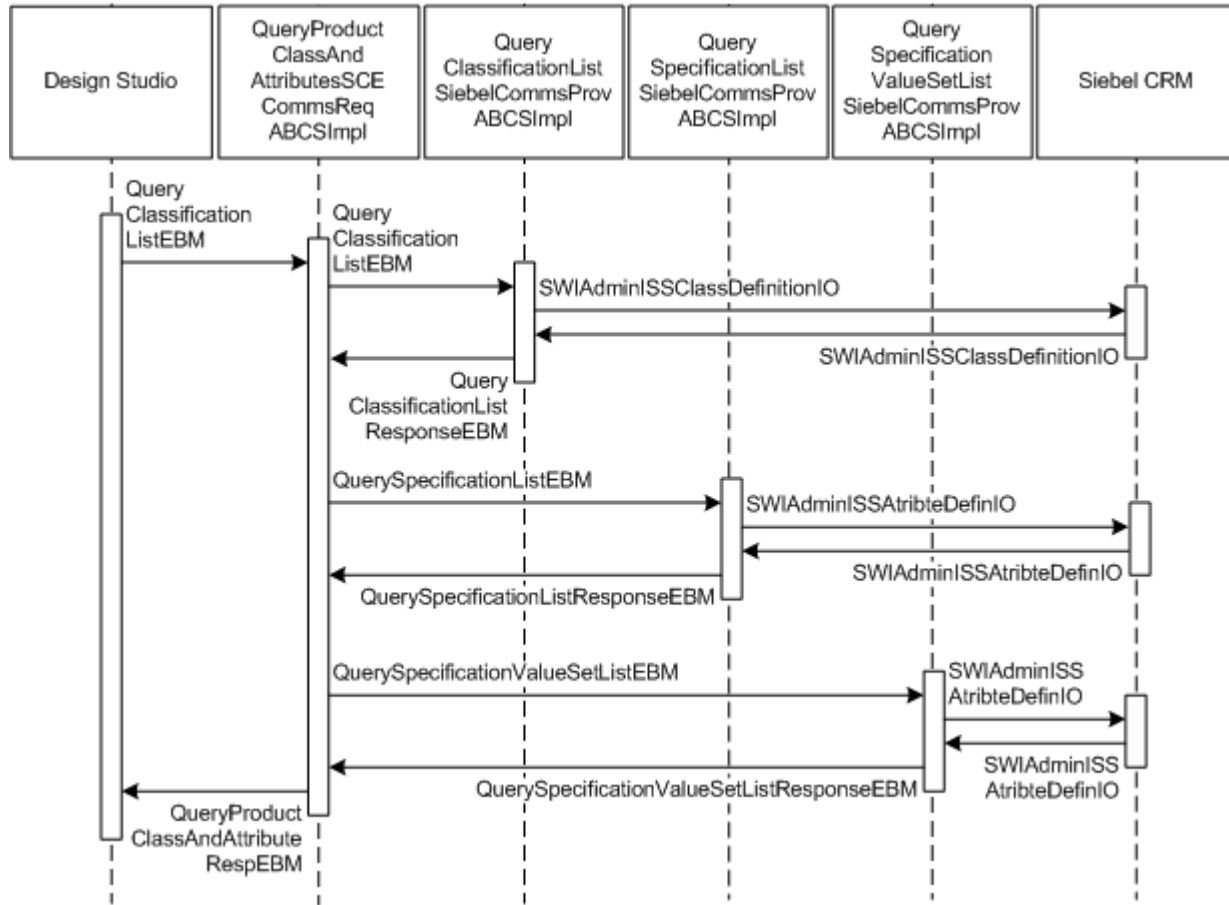
The Query Product Classes integration flow lets you import product classes and attributes from Siebel CRM into Design Studio. You submit a query for the product classes when you receive a notification from Siebel CRM that the product classes have been updated. The business flows maps the product classes and their attributes to product specifications, which OSM uses for order orchestration.

This integration flow uses the following interfaces:

- QueryProductClassAndAttributesSCECommsReqABCImpl
- QueryClassificationListSiebelCommsProvABCImpl
- QuerySpecificationListSiebelCommsProvABCImpl
- QuerySpecificationValueSetListSiebelCommsProvABCImpl

Figure 6-1 illustrates the query product classes integration flow.

Figure 6-1 Query Product Classes and Attributes Integration Flow Sequence Diagram



The integration queries product classes and attributes as follows:

1. A Design Studio user submits a query with product class names and, for updates, the product class codes for any subclasses.  
Design Studio sends the QueryClassificationListEBM message containing the product class codes to the QueryProdClassAndAttributesSCEReqCommsABCsImpl service.
2. The QueryProdClassAndAttributesSCEReqCommsABCsImpl service sends the QueryClassificationListEBM message to the QueryClassificationListSiebelCommsProvABCsImpl service.
3. The QueryClassificationListSiebelCommsProvABCsImpl service transforms the QueryClassificationListEBM message into the Siebel Class Definition IO ABM, includes the Workspace Name property from the **AIAConfigurationProperties.xml** file in the ABM, and invokes the Siebel CRM GetProductClass web service.
4. The Siebel CRM GetProductClass web service returns the product class information and associated attribute IDs.
5. The QueryClassificationListSiebelCommsProvABCsImpl service transforms the Siebel CRM GetProductClassResponseABM into the QueryClassificationListRespEBM and sends it to the QueryProdClassAndAttributesSCEReqCommsABCsImpl service.
6. The QueryProdClassAndAttributesSCEReqCommsABCsImpl service extracts the attribute IDs from the QueryClassificationListRespEBM and maps them 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.

7. The QueryProdClassAndAttributesSCEReqCommsABCServiceImpl sends the QuerySpecificationListEBM message to the QuerySpecificationListSiebelCommsProvABCServiceImpl service.
8. The QuerySpecificationListSiebelCommsProvABCServiceImpl service transforms the QuerySpecificationListEBM message into the Siebel CRM Attribute definitions IO ABM, includes the Workspace Name property from the **AIAConfigurationProperties.xml** file in the ABM, and invokes the Siebel CRM GetAttributeDefinition web service.
9. The GetAttributeDefinition web service returns the attribute information for the attribute IDs to the QuerySpecificationListSiebelCommsProvABCServiceImpl service.
10. The QuerySpecificationListSiebelCommsProvABCServiceImpl service transforms the response into the QuerySpecificationListRespEBM message and sends it to the QueryProdClassAndAttributesSCEReqCommsABCServiceImpl service.
11. The QueryProdClassAndAttributesSCEReqCommsABCServiceImpl service sends the QuerySpecificationValueSetListEBM message to the QuerySpecificationValueSetListSiebelCommsProvABCServiceImpl service.
12. The QuerySpecificationValueSetListSiebelCommsProvABCServiceImpl transforms the QuerySpecificationValueSetListEBM into the Siebel CRM attribute definitions IO ABM, includes the Workspace Name property from the **AIAConfigurationProperties.xml** file, and invokes the Siebel CRM GetAttributeDefinition web service.
13. The GetAttributeDefinition web service returns the attribute valueset information for the attribute IDs to the QuerySpecificationValueSetListSiebelCommsProvABCServiceImpl.
14. The QuerySpecificationValueSetListSiebelCommsProvABCServiceImpl service transforms the response into the QuerySpecificationValueSetListRespEBM and sends it to the QueryProdClassAndAttributesSCEReqCommsABCServiceImpl service.
15. The QueryProdClassAndAttributesSCEReqCommsABCServiceImpl service merges the QuerySpecificationListRespEBM message, QuerySpecificationValueSetListRespEBM message, and the QueryClassificationListRespEBM message and transforms them to the QueryProdClassAndAttributesRespABM message.
16. The QueryProdClassAndAttributesSCEReqCommsABCServiceImpl sends the QueryProdClassAndAttributesRespABM message to Design Studio.

## 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home/comms\_home/source/soainfra/apps/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 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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending 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 requester 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.

# 7

## Understanding the Process Integration for Order Lifecycle Management

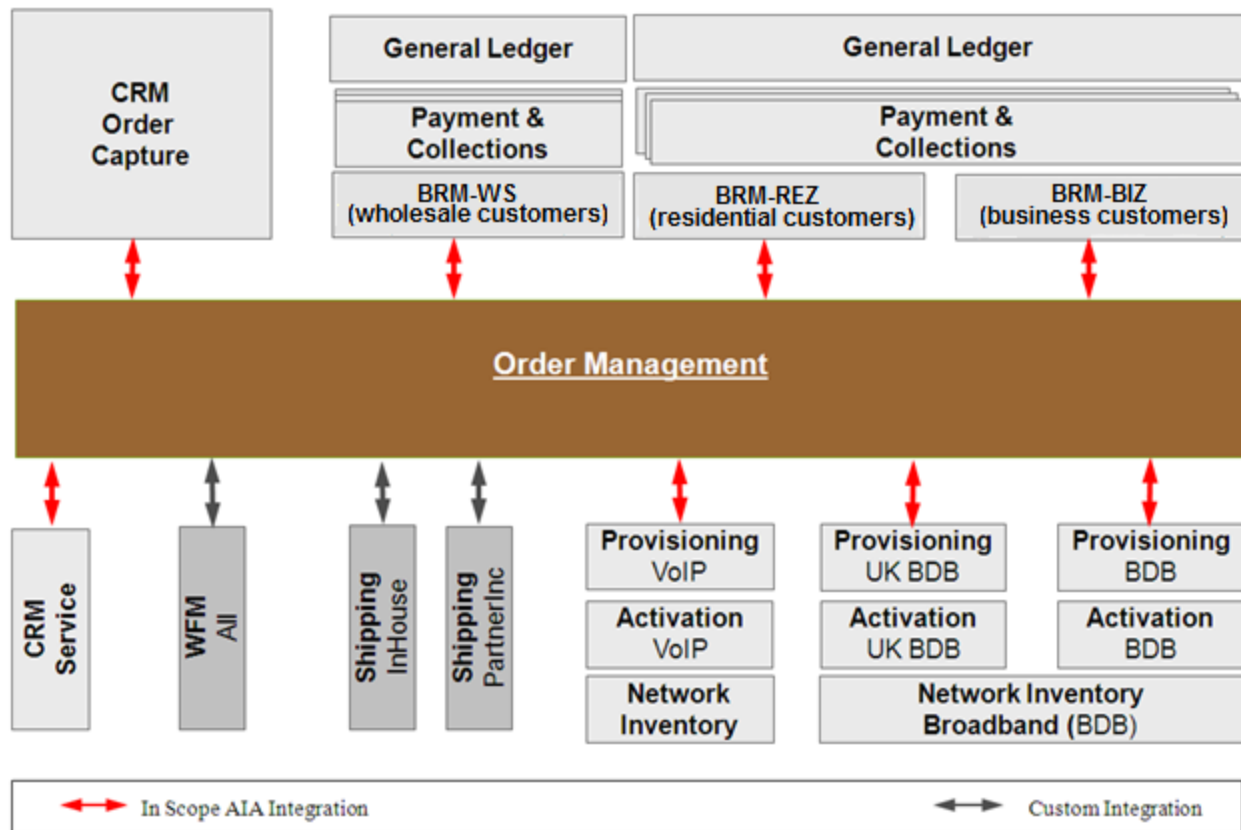
This chapter describes the process integration for order lifecycle management and discusses a typical topology and order capture flow. It also describes the Qualify Customer Order and Deliver Customer Order subflows and design considerations for product definition and mapping.

### About the Process Integration for Order Lifecycle Management

The process integration for order lifecycle management extends from the time a quote or order is created to the time when the goods and services are delivered and billed. The Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) works with Siebel CRM, Oracle Communications Order and Service Management (OSM), and Oracle Communications Billing and Revenue Management (BRM). You can integrate with other types of fulfillment system, such as supply chain management and workforce management, or with alternative billing and order management systems as an extension project at implementation time.

[Figure 7-1](#) illustrates an example deployment topology for integrated order lifecycle management overlaying the corresponding Oracle products. An order management system such as OSM in the Central Order Management role (OSM COM) is at the center of the integration deployment.

Figure 7-1 Example Oracle Communications Order to Cash Deployment Topology



In the figure, an order capture system (Siebel CRM) captures and passes orders to the order management system (OSM). The order management system then decomposes the order into order components, each of which targets a particular fulfillment provider.

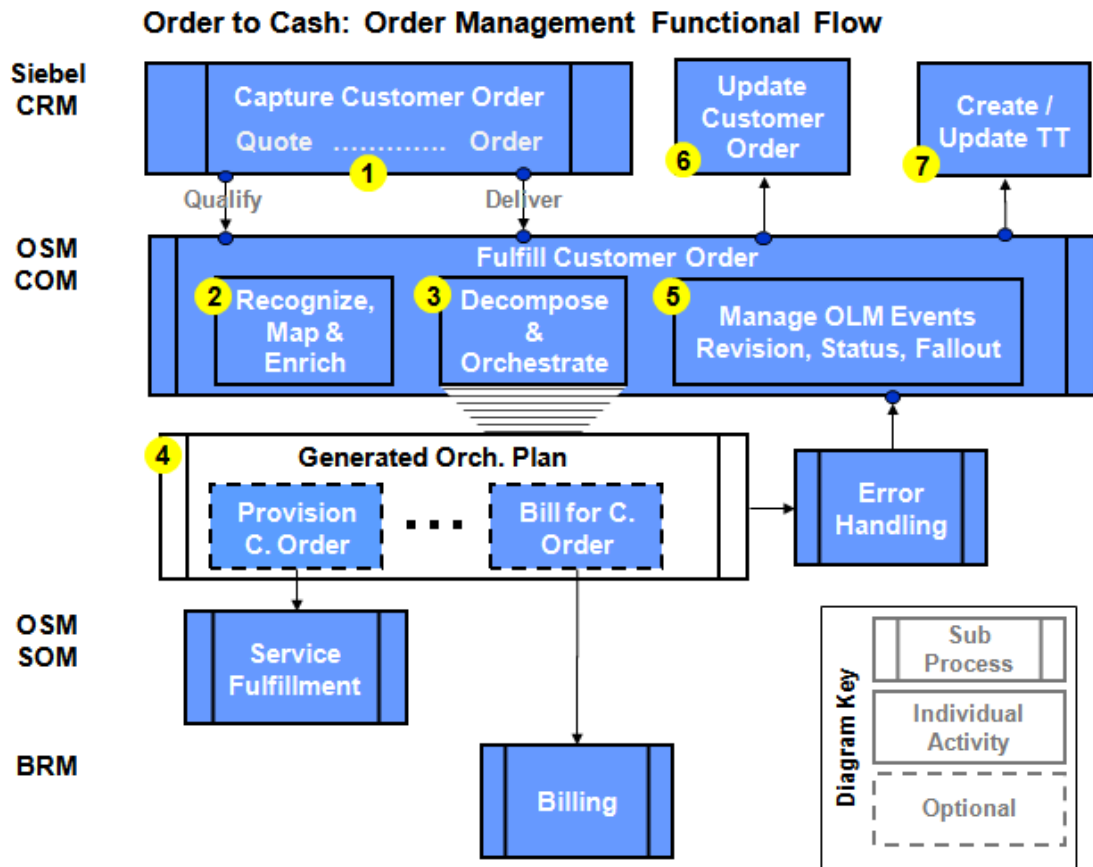
The topology in the figure uses the following fulfillment providers:

- Three billing providers, represented by BRM, based on customer segment: wholesale, residential, and business.
- Three provisioning stacks based on service family and geography: VoIP, UK Broadband, and US Broadband. The provisioning stacks include OSM in the Service Order Management role (OSM SOM), Oracle Communications ASAP, and Oracle Communications Unified Inventory Management.
- One workforce management provider
- Two shipping providers, one for in-house products and another for partner supplier products.
- One customer relationship management provider, represented by Siebel CRM, for trouble ticketing

Figure 7-2 illustrates the functional flow of an order from capture to billing and fulfillment using pieces of the typical topology.



Figure 7-2 Order Lifecycle Management Functional Flow



The functional flow for order lifecycle management is as follows:

- 1. Capture Customer Order:** A customer order is captured and validated in Siebel CRM then submitted to OSM in the central order management (COM) role for fulfillment. Siebel CRM supports technical service qualification for quotes as part of the Qualify Customer Order subflow.
- 2. Recognize, Map, and Enrich:** OSM recognizes customer orders 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 cartridge provides ready-to-use automatic integration with Oracle AIA web services. When the 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 Order Lifecycle Events:** OSM manages order lifecycle events. For cancelation and revision requests, OSM generates and executes compensation plans to match a change. The integration 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 customer status and milestone values. When order lines reach their point of no return, Siebel CRM prevents 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. Oracle 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 *OSM 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, and BRM. This guide does not cover process details within OSM, for example, service design, assign, and activation.

## About Order Lifecycle Attributes

As orders pass through their lifecycle, the integration sets or maintains the following enterprise business object (EBO) attributes to offer visibility into the order lifecycle and determine the actions of fulfillment systems:

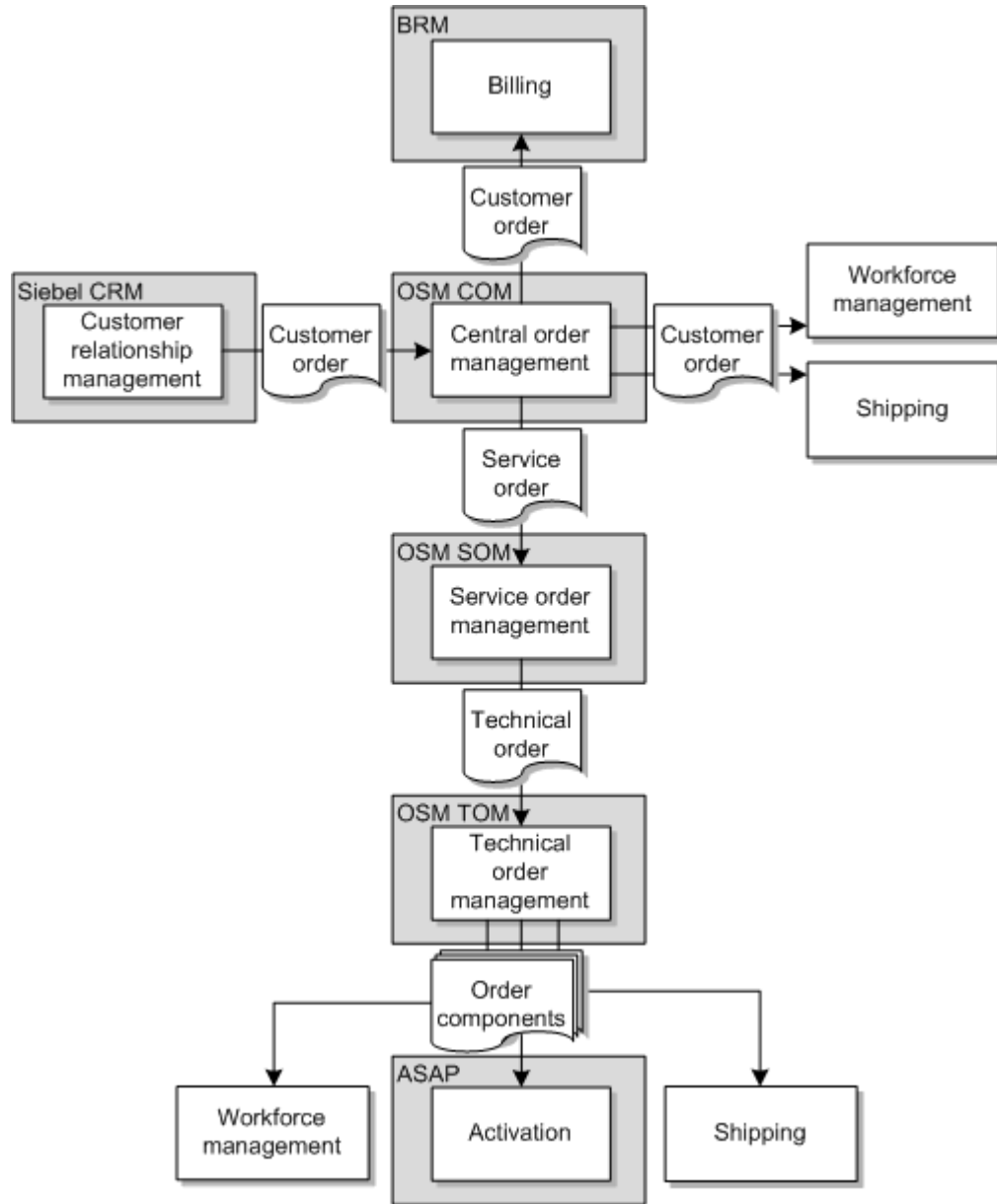
- OrderSubject
- ServiceFamily
- TechnicalInventoryId

## About the OrderSubject Attribute

The OrderSubject attribute indicates an order's subject, which reflects the type of fulfillment system to which the order is sent. As the order management system decomposes orders and order components pass through different fulfillment systems, the order type and its associated subject changes.

Figure 7-3 shows, at a high level, how the order type and subject changes as OSM in the central, service, and technical order management roles (OSM COM, OSM SOM, and OSM TOM respectively) decomposes a customer order originating in Siebel CRM and sends it to various fulfillment systems as a customer order, a service order, or a technical order.

Figure 7-3 Changing Types of Order Throughout the Order Lifecycle



Fulfillment systems determine the order subject based on the value of OrderSubject in the order EBO. The integration sets OrderSubject to CUSTOMER for orders that come from Siebel CRM. While transforming the customer order into a service order, OSM sets OrderSubject to SERVICE.

Although the figure shows OSM COM transforming the order, either OSM COM or OSM SOM can alternatively transform the order, depending on your configuration. OSM SOM uses OrderSubject to determine whether the order needs to be transformed. If OSM SOM receives an order with OrderSubject set to CUSTOMER, then OSM SOM transforms the order. If OSM

SOM receives an order with OrderSubject set to SERVICE, then OSM COM has already transformed the order and OSM SOM skips the transformation step.

See "[ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl](#)" for more information about the Oracle AIA service that sets OrderSubject to CUSTOMER and *Oracle Communications OSM Cartridge Guide for Oracle Application Integration Architecture* for more information about how the OSM O2A cartridges transform customer orders into service orders.

## About the ServiceFamily Attribute

The ServiceFamily attribute tracks the category of a service, such as broadband or mobile. The value of ServiceFamily determines the actions of various fulfillment systems. For example, if you have separate provisioning systems for VoIP and broadband services, as in [Figure 7-1](#), OSM would use the value of ServiceFamily to determine which provisioning system handles the order lines.

The OSM O2A cartridges set the value for ServiceFamily while transforming customer orders to service orders. The value is maintained on the order EBO throughout the order life cycle. The value is determined based on the domain set for the service in the OSM conceptual model, or, for certain services, calculated at run-time based on the context in which the service is used. See *Oracle Communications Order and Service Management Concepts* for more information about conceptual models in OSM.

## About the TechnicalInventoryId Attribute

The TechnicalInventoryId attribute serves as a unique identifier that correlates commercial inventory items, such as customer assets, with technical inventory items across all technical inventory systems.

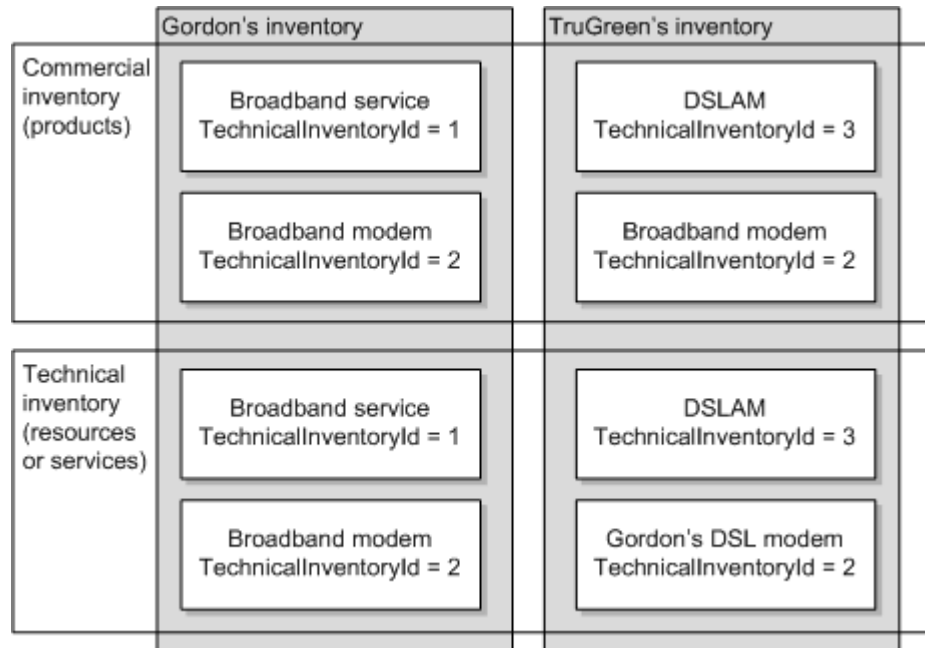
A single customer asset can correspond to multiple technical assets in multiple fulfillment systems and inventory perspectives.

The inventory perspectives include distinctions between products and resources (commercial and technical inventories), between assets you own and that your customers own (enterprise and customer inventories), between utilized and available resources, and between past and future views of inventories. An asset can appear in many of these perspectives as an order progresses through its life cycle.

The integration uses the TechnicalInventoryId attribute to correlate the customer's assets to the corresponding assets in the technical inventory.

For example, a customer named Gordon subscribes to broadband service from the TruGreen service provider, which includes a rented modem. [Figure 7-4](#) shows how some of the assets appear in the overlapping inventory perspectives, and how TechnicalInventoryId correlates them.

**Figure 7-4 Example of TechnicalInventoryId Correlating Assets**



The service and resource management system provides the technical inventory ID to the OSM O2A cartridges, which maintain the value in the order EBO throughout the order life cycle.

## About the Order Lifecycle Management Business Flows

The process integration for order lifecycle management includes the following business flows:

- **Process Sales Order Fulfillment:**
  - Enabled using the Order to Cash pre-built integration options for OSM and Siebel CRM.
  - 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 Order to Cash pre-built integration options for OSM, BRM and Siebel CRM.
  - 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 Order to Cash pre-built integration options for OSM and BRM.
  - 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 the Order to Cash pre-built integration option for OSM.

- Used when provisioning orders in OSM SOM, and updating orders and statuses in OSM COM through order updates from OSM SOM.
  - See "[Understanding the Provision Order and Update Fulfillment Order Business Flows](#)" for more information.
  - **Update Sales Order:**
    - Enabled using the Order to Cash pre-built integrations options for OSM and Siebel CRM.
    - Used when sending order updates from OSM COM to Siebel CRM.
- See "[Understanding the Update Sales Order Business Flow](#)" for more information.

## About Order Capture

Figure 7-5 shows a typical order capture flow. The flow can vary depending on, for example, service family, customer segment, or 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 order lifecycle management includes 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 7-5 Typical Order Capture Flow**

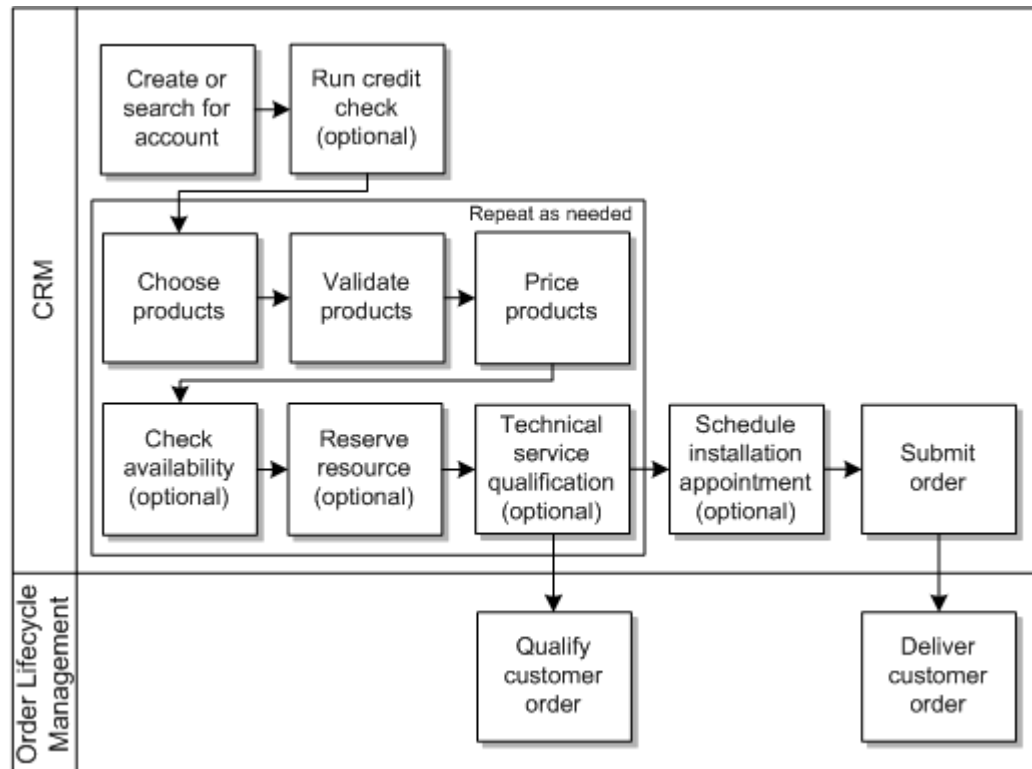


Figure 7-5 shows typical order activities for the CRM system and their integration points with the order lifecycle management activities.

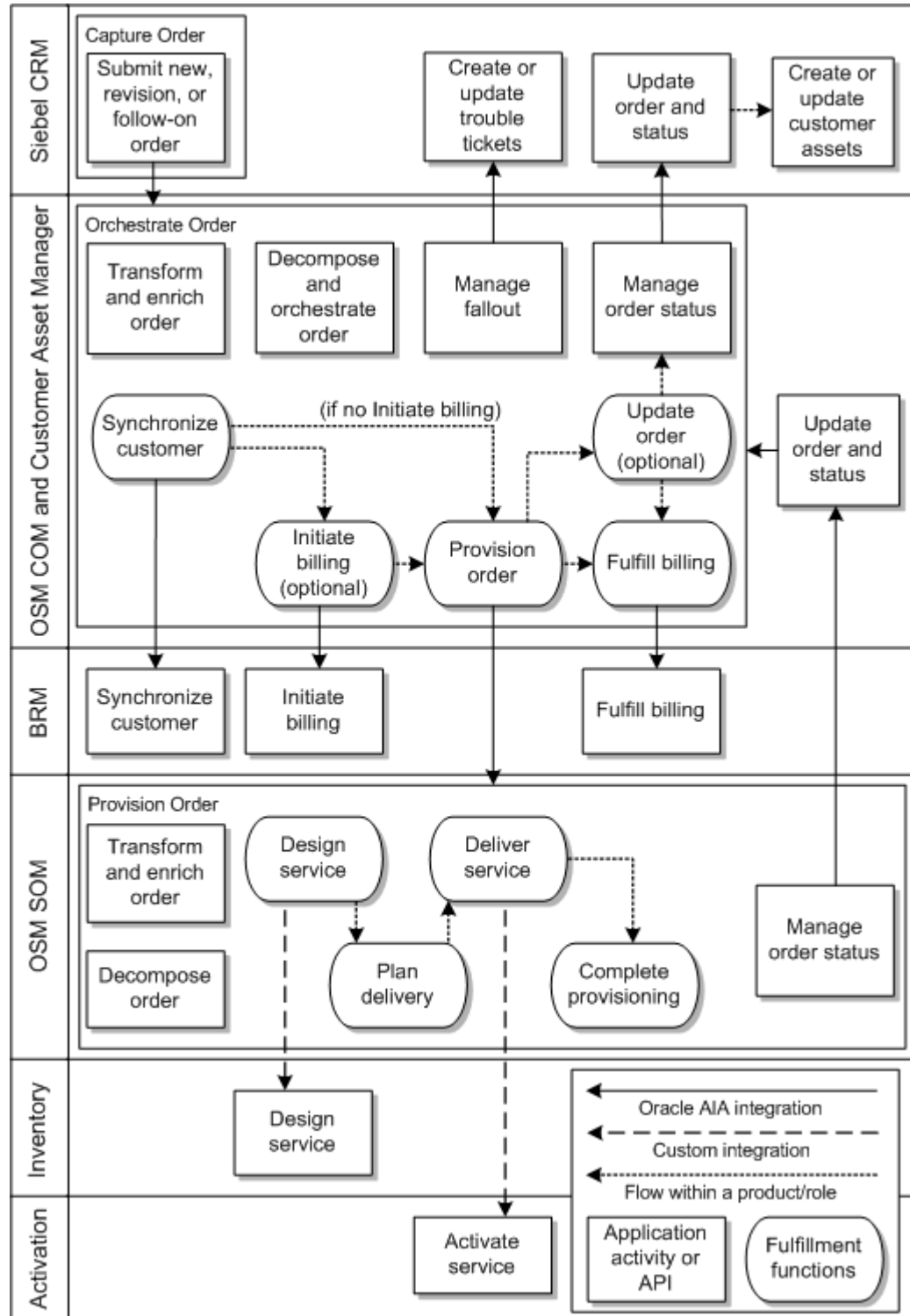
A typical order capture progresses as follows:

1. A customer service representative (CSR) creates a new customer account or searches for an existing customer account. CSRs can also capture customer information at other times, such as when creating or updating an opportunity or quote.
2. (Optional) The CSR runs a credit check on the new customer.
3. The customer chooses products and the CRM system validates the products.
4. The CSR prices the selected products and product options.
5. (Optional) The CSR checks the availability of any physical goods.
6. (Optional) The CSR reserves the resource for services such as phone numbers.
7. (Optional) The CSR subjects the order to technical service qualification and the Qualify Customer Order subflow starts.
8. (Optional) The CSR schedules an appointment with an engineer through a workforce management system.
9. The CSR submits the order and the Deliver Customer Order subflow starts.

## About the Deliver Customer Order Subflow

Figure 7-6 shows the typical application activities and interactions involved in delivering customer orders.

Figure 7-6 Deliver Customer Order Subflow



The integration delivers customer orders as follows:

1. A new, revision, or follow-on order is submitted in Siebel CRM. The integration sends the order to OSM.



2. OSM does the following:

- a. Transforms and enriches the order by mapping order lines to fulfillment flows and enriching them with fulfillment metadata and other relevant data.
- b. Decomposes the order by dividing the order into order components and composing an orchestration plan to track order dependencies.

The orchestration plan directs order fulfillment using preconfigured functions, such as synchronizing the customer into BRM, initiating and fulfilling billing, provisioning the order, shipping the order, and installing the order. Oracle AIA integrates the OSM fulfillment functions with BRM APIs, and custom integrations integrate OSM fulfillment functions with network inventory and activation system APIs.

Orchestration plans are typically more complex than the flow in [Figure 7-6](#). See the discussion of orchestration in *Oracle Communications Order and Service Management Concepts* for an examples of a more detailed orchestrations plan.

- c. OSM 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.

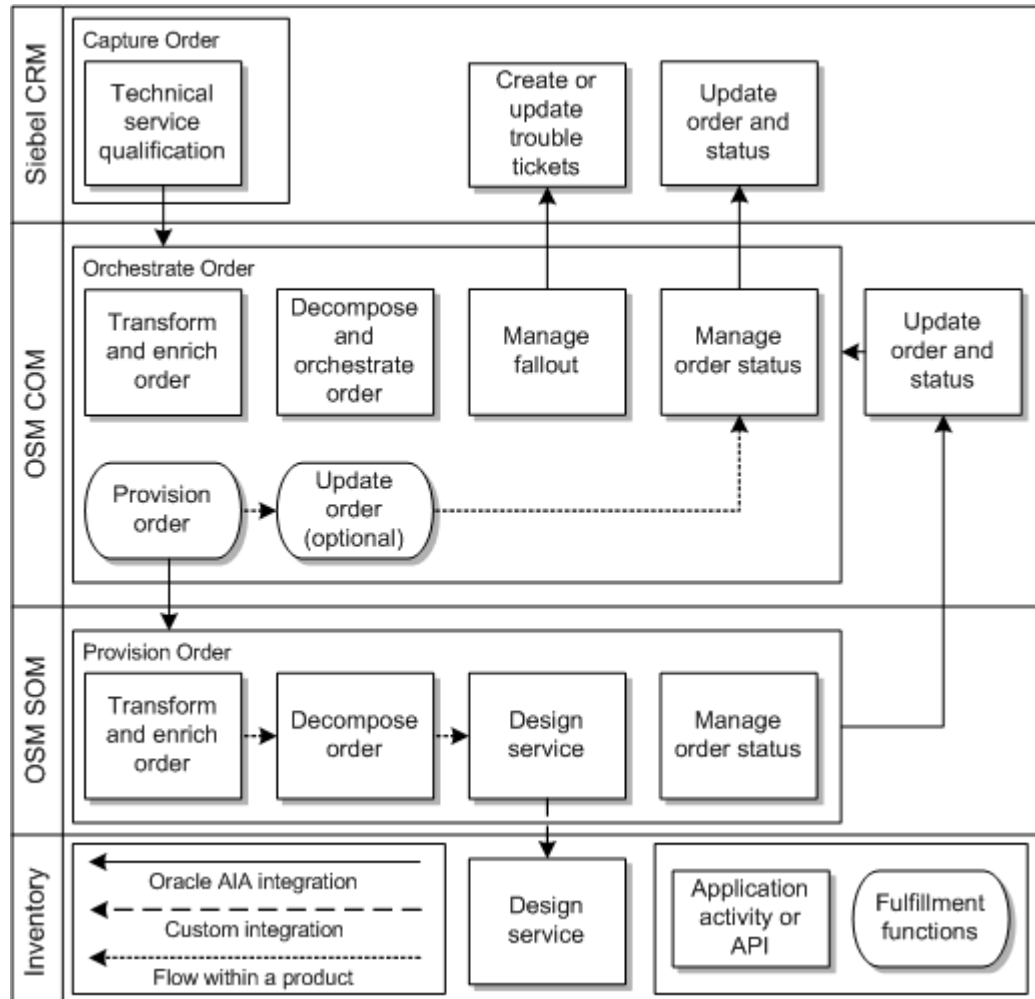
See "[Using Error Type to Control Response to Order Fallout](#)" for more information.

- d. Manages order status by mapping fulfillment function responses to common statuses. 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 7-7](#) shows the typical application activities and interactions involved in delivering customer orders.

Figure 7-7 Qualify 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 functions described in "[About the Deliver Customer Order Subflow](#)", except that the metadata and the fulfillment functions are for qualifying the customer order rather than delivering the customer order. The billing, activation, and delivery activities are not part of qualifying orders. The two subflows also produce different order and order line status updates.

## Product Definition and Mapping Design Considerations

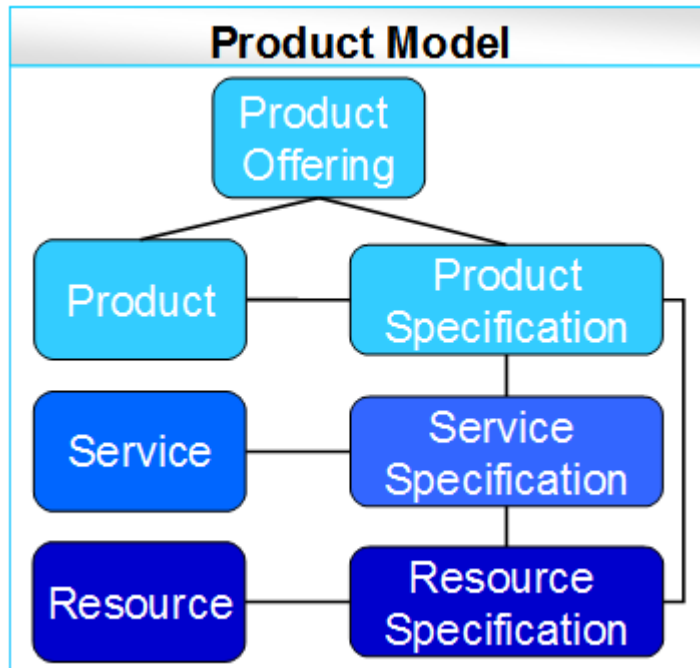
This section discusses high-level considerations for defining your products and mapping them on orders to fulfillment functions at run time.

### About Defining Products

Because the product and service definition methodology has the greatest effect on time to market and on the cost of an Order to Cash deployment, Oracle recommends a balanced approach that involves compromises between departments that result in simplified overall product life cycle and order life cycle business flow.

Figure 7-8 illustrates the basic recommended product model and aligns with TM Forum terminology and guidelines. It includes three TM Forum Information Framework entities: product, service, and resource.

**Figure 7-8 Basic Product Model**



A balanced model produces a catalog with product specifications represented by the fewest entities. Product specifications are types of products that represent unique capabilities with commercial value that are sold through product offerings

Product offerings represent tangible and intangible goods and services made available for a certain price to the market. Product offerings take one of the following forms:

- **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.

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.

## About Mapping Orders to Fulfillment Functions

Order management systems act on customer orders, which 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 or UPDATE to modify a customer's subscription to an offering. A subject is the target of the action and can represent items such as an offering, an asset, or a discount.

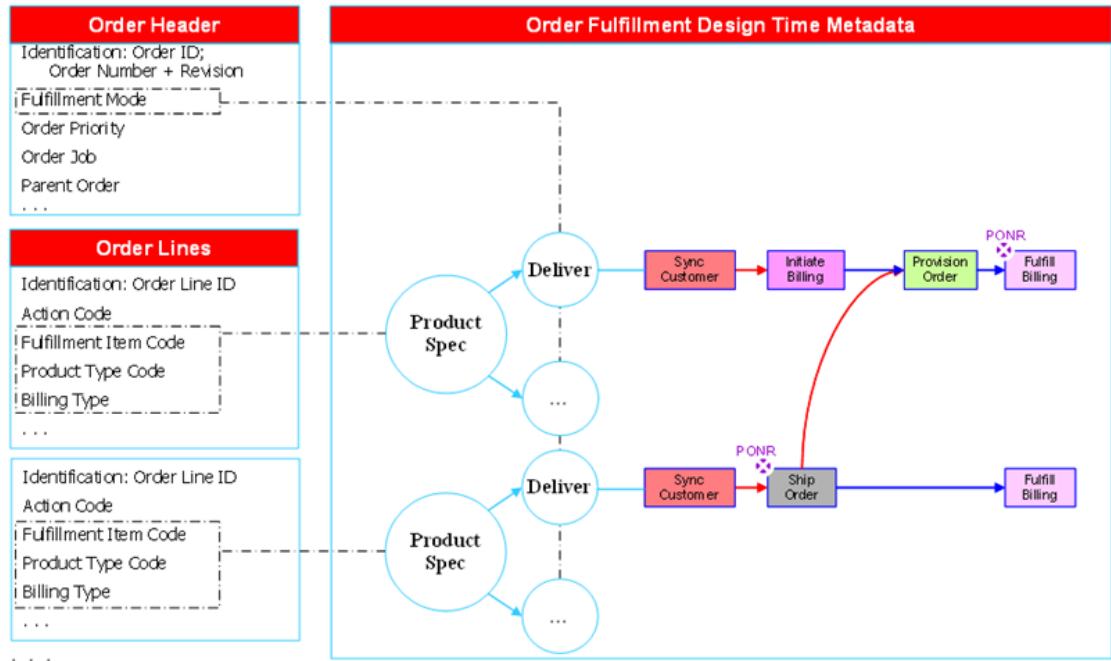
Oracle recommends mapping each order line to a separate product specification. This approach helps achieve fast time-to-market and low-cost operations. The integration implements this recommendation by associating product offerings with a Siebel CRM product

class and an OSM product specification using the Fulfillment Item Code attribute in Siebel CRM.

Mapping a customer order to a service order requires specific metadata modeled on products, product specifications, and service and resource configurations. In an Order to Cash deployment, OSM handles this mapping.

Figure 7-9 illustrates how an order management system in the integration uses the product model to map customer order lines to fulfillment functions.

**Figure 7-9 Mapping Order Lines to Fulfillment Functions**



When a customer places an order, Siebel CRM copies product offering attributes to each order line. These attributes include Fulfillment Item Code, Product Type Code, and Billing Type. The integration uses these attribute values to determine the corresponding product specification. The Fulfillment Mode order header attribute determines the fulfillment request 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.

You can synchronize product classes from Siebel CRM to product specifications or conceptual model Product entities in OSM automatically using the Query Product Classes business flow. See "[Understanding the Query Product Classes Business Flow](#)" for more information.

## Data Requirements for Integrated Order Lifecycle Management

Orders must include the following data for the integration to successfully process the order:

- An order must be of type Sales Order.
- For Siebel CRM, 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.
- The following 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

The Oracle AIA sales order enterprise business object is extensible and includes a vast set of attributes that are sufficient for most fulfillment systems.

# 8

## Understanding the Process Sales Order Fulfillment Business Flow

This chapter describes the Process Sales Order Fulfillment business flow and ordering concepts supported by the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration).

### About the Process Sales Order Fulfillment Business Flow

The business flow is initiated when a Siebel CRM customer service representative (CSR) submits an order.

Siebel CRM adds the order message to a Java Messaging Service (JMS) queue, where a JMS consumer picks it up and passes it to the Oracle AIA integration. The integration transforms the message and passes it to OSM for processing.

The integration includes a Test Orchestration Process to test the ready-to-use order flow if you are not using OSM for order management. For production systems, you must replace the test orchestration process with your own order management system. See *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations* for more information about replacing the TOP.

### 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.

Order line subjects in Siebel CRM 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**.

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. Revisions internal to Siebel CRM 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 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 in Siebel CRM.

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 orders in different states of completion. This chapter uses the Siebel CRM term.

[Table 8-1](#) defines and maps the terms from Siebel CRM to OSM.

**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. See <a href="#">"About Supplemental Orders"</a> .  |
| 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. See <a href="#">"About Follow-On Orders"</a> . |
| 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. See <a href="#">"About Future-Dated Orders"</a> .  |

## About Change Orders

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. For asset-based ordering, Siebel CRM simulates the future state of assets in future-dated orders.

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 Large Orders

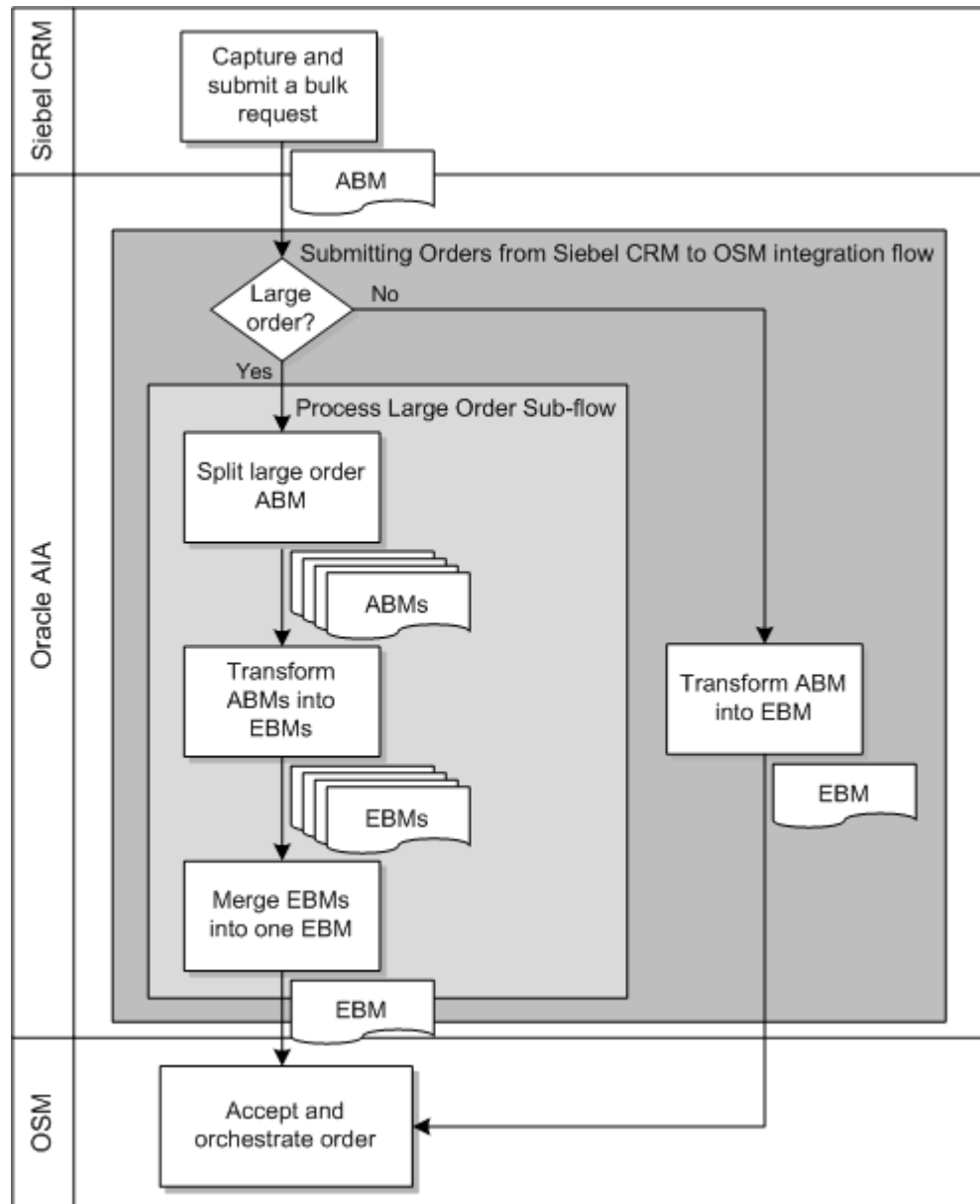
A large order is a sales order with a large number of order lines, typically several hundred order lines. Both business-to-consumer scenarios (such as bulk price changes for all customers) and business-to-business scenarios (such as corporate orders for services in multiple offices) can result in large orders. See "[Promotion Groups and Large Order Scenario](#)" for a detailed example scenario that results in a large order.

By default, Oracle AIA can process orders with up to 1000 order lines with no significant impact on order processing time. For orders with greater than 1000 order lines, you can decrease order processing time by enabling the Process Large Order sub-flow within the Submitting Orders to OSM integration flow.

To decrease processing time to transform the order message as part of the Process Large Order sub-flow, Oracle AIA splits the order ABM message into multiple smaller messages, transforms the smaller ABMs into EBMs, and then merges the smaller transformed EBMs.

[Figure 8-1](#) illustrates this process.

Figure 8-1 Processing a Large Order in an Oracle AIA Deployment



The OSM APIs, server, database, and O2A cartridges all help OSM orchestrate orders that have several thousand order lines.

The following properties in the **AIAConfigurationProperties.xml** file configure the Process Large Order sub-flow:

- The **handleLargeOrderEnabled** property specifies whether Oracle AIA uses the Process Large Order sub-flow to split large orders.
- The **numOrderLinesInLargeOrder** property specifies the number of order lines an order must have to be considered a large order.
- The **numOrderLinesInMiniABM** property specifies the number of order lines in a single small ABM.

For more information about these properties, see [Table 25-6](#).

## Configuring Extra Large Order Support

An extra large order is an order with many thousands of order lines. To support mega orders, Oracle recommends using the Process Large Order subflow in Oracle AIA, and developing large order management layer between Oracle AIA and OSM COM. This layer splits mega orders into multiple dependant sub-orders.

See the Mega Orders whitepaper available from My Oracle Support for guidelines for developing your own large order management layer.

## About Accepting Payments on Orders

The integration supports accepting payments from customers at order time.

Using Siebel CRM, you can accept the following methods of payment:

- Automatic debit
- Credit card
- Cash
- Wire transfer
- Check

You must set the payment status to **New** in Siebel CRM for payments included at order time.

You can record the payment against any billing profile that appears on the order or that has already been synchronized to BRM in a previous order. You can record multiple payments against a single billing profile or against multiple billing profiles.

For example, a new customer, Denise, sets up the following three billing profiles on her account:

- BP-D, which she will use for her own services
- BP-M, which she will use for her daughter Michelle's services
- BP-J, which she will use for her daughter Jessica's services

Denise wants to order wireless service for her daughters and pay for their phones at the same time. When placing the order, the customer service representative (CSR) assigns BP-M to Michelle's service and BP-J to Jessica's service. To record the payments for the phones, the CSR can only use BP-M or BP-J, because BP-D does not appear on the current order or any previous orders.

## Accepting Payments on Change Orders

The integration supports accepting payments on change orders in the same way as on new orders. You cannot, however, use change orders to change or reverse payments.

For example, after Denise places the order for Michelle and Jessica's wireless services and phones, she calls back to add voicemail to Michelle's wireless service and to purchase protective cases for the phones. As part of the change order for Michelle's voicemail, she can pay for the protective cases, but she cannot reverse or change the payment for either of the phones purchased in a previous order.

Because Oracle AIA does not synchronize the status for payments received on orders back to Siebel CRM, when you submit a change order for an order that originally included a payment,

you must change the payment status based on the status of the order line. For example, if the order line status is Submitted, you must change the payment status to Submitted as well.

## Supporting Multiple Price Lists on Orders

At order creation 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.

At design time in Siebel CRM, you create price lists, 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 synchronizing product from 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 from Siebel CRM 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 Promotion Groups on Orders

Your customers can share charges, discounts, and special rating profiles by purchasing promotion groups.

You create promotion groups at design time in Siebel CRM. Each promotion group definition includes an owner membership product, a member membership product, and one or more reward products. See "[About Promotion Groups](#)" for more information about creating promotion groups at design time. At order time, you associate the membership products with your customers' accounts and services.

You include a promotion group on a sales order as described in the discussion of creating new promotion group instances in *Siebel Order Management Guide Addendum for Communications*, with the following modifications:

- Create the sales order from the promotion group owner's account.
- Associate owner membership products with any of the following:
  - Service bundle
  - Simple service bundle
  - Nested service bundle
  - Account, only in the following situations:
    - \* (Optional) In promotion groups that contain only chargeshare rewards

- \* (Required) In promotion groups with a member membership product associated with a product promotion
- Do not associate the owner membership product with a product promotion or a service bundle within a product promotion.
- Associate member membership products with any of the following:
  - Service bundle
  - Simple service bundle
  - Product promotion

Table 8-5 shows some details from an example order placed from a corporate account. The order includes VoIP services for the corporate account and an employee, Andrew. It also includes a promotion group instance, promotion group memberships associated with the corporate and employee services, a recurring fee, and three shared rewards.

**Table 8-5 Example of an Order for a Promotion Group**

| Line Item              | Service Account and Billing Account | Description   |
|------------------------|-------------------------------------|---|
| VoIP Service           | Corporate                           | A service bundle  |
| VoIP Service           | Andrew                              | A service bundle  |
| VoIP Promotion Group   | Corporate                           | A promotion group instance  |
| Corporate Membership   | Corporate                           | A promotion group owner membership product, associated with corporate's VoIP Service  |
| Employee Membership    | Andrew                              | A promotion group member membership product, associated with Andrew's VoIP Service    |
| VoIP 5000 Free Minutes | Corporate                           | A discount reward product for a shared pool of free minutes                           |
| VoIP 50% Sponsorship   | Corporate                           | A sponsorship reward product for corporate to cover 50% of an employee's VoIP charges |
| VoIP Special Rating    | Corporate                           | A special rating reward product offering discounted calls between employees           |
| VoIP Promotion Fee     | Corporate                           | A recurring fee reward product charging corporate for the promotion group             |

See "[Synchronizing Promotion Groups](#)" for details about how the integration implements this order in BRM.

You can include multiple promotion groups on a single order. However, if you are using multiple instances of BRM, you cannot use a single order for multiple promotion groups that include services from different BRM instances. You must use a separate order for each promotion group that includes services from a separate BRM instance.

## Using Change Orders with Promotion Groups

You can use change orders to make the following changes to promotion groups:

- Add members by adding new order lines for member membership products associated with new services
- Delete members by disconnecting member membership products

- Change the service associated with the owner membership product to a different service by disconnecting the owner membership product and adding a new order line for the owner membership product associated with a different service owned by either the same account or a different account

 **Note:**

Although it is possible in Siebel CRM to change the service with which an existing owner membership product is associated without disconnecting the owner membership product, Oracle AIA does not support this method of changing the owner of a promotion group.

- Add new rewards of the same type (for example, you can add new discount rewards to promotion groups that already include discount rewards)
- Remove rewards
- Disconnect entire promotion groups

See the discussion of managing promotion groups and managing promotion group members in *Siebel Order Management Guide Addendum for Communications* for more information about using change orders to manage promotion groups.

## Supporting Family Share Plans on Orders

Family share plans let a group owner share resources with group members, and the integration synchronizes family share plans from Siebel CRM to sharing groups in BRM. Family share plans do not add extra order lines for memberships. Instead, the integration uses order line attributes to create sharing groups in BRM.

The accounts that participate in a family share plan must be in an account hierarchy. You create the order for the family share plan from the parent account, and purchase the add-on services in the bundled promotion for the nonpaying child accounts. Use the parent account as the billing account on all services in the bundled promotion and use the nonpaying child account as the service account for the add-on wireless lines. You can use separate billing profiles for each service.

For example, Denise purchases the Wireless Family Share Plan bundled promotion shown in [Figure 3-8](#) with add-on lines for her two children, Michelle and Jessica. Denise selects different service accounts for the services and different billing profiles of her billing account. [Table 8-6](#) shows some of the details included on the order.

**Table 8-6 Example of an Order for a Family Share Plan**

| Line Item            | Line Item Attributes                         | Service Account | Billing Account | Billing Profile |
|----------------------|--|-----------------|-----------------|-----------------|
| Wireless Family Plan | N/A  | Denise          | Denise          | Denise-Denise   |
| Primary Line         | Service Grouping = Y<br>Community Member = Y | Denise          | Denise          | Denise-Denise   |
| Add-on Line          | Service Grouping = Y<br>Community Member = Y | Michelle        | Denise          | Denise-Michelle |
| Add-on Line          | Service Grouping = Y<br>Community Member = Y | Jessica         | Denise          | Denise-Jessica  |



Table 8-6 (Cont.) Example of an Order for a Family Share Plan

| Line Item      | Line Item Attributes | Service Account | Billing Account | Billing Profile |
|----------------|----------------------|-----------------|-----------------|-----------------|
| 5 GB Free Data | Community Offer = Y  | Denise          | Denise          | Denise-Denise   |

See "[Synchronizing Family Share Plans](#)" for details about how the integration implements this order in BRM.

When placing a new or change order for a family share plan bundled promotion, you can add only the service bundles that were defined as components of the promotion at design time. You cannot add other service bundles or existing assets from the parent or child account.

See the discussions of editing a bundled promotion and bundling components for a promotion in *Siebel Order Management Guide Addendum for Communications* for more information about adding service bundles and existing assets to bundled promotions.

## Using Change Orders with Family Share Plans

You can use change orders to make the following changes to family share plans:

- Add members by adding new add-on services for other subordinate accounts
- Remove members by removing their add-on services
- Add new reward discounts
- Remove existing reward discounts
- Disconnect entire family share plans

You cannot use change orders to disable the Service Grouping attribute. After an order for a service bundle with this attribute enabled is fulfilled, you cannot ungroup the nested services.

## Supporting Order Priority

You select order fulfillment priority in Siebel CRM when submitting orders. This 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 order priority set in Siebel CRM is honored by message queues, Oracle AIA, and OSM, unless data integrity dictates a different processing sequence, such as with updates to sales orders from OSM to Siebel CRM.

## About Order Priority in Siebel CRM

By default, Siebel CRM provides the following order priority values:

- Low
- Medium
- High
- Urgent

When converting the Siebel CRM message to an EBM, the integration maps these values to integer values that are compatible with JMS priority.

JMS and the integration support priority values 0-9. You can extend Siebel CRM to support the full range of priority values by using the SWI\_ORDER\_JMS\_PRIORITY mapping, which maps string values to integers. You must set up JMS compatibility properties on the Siebel CRM queue and make manual changes to seeded priority values.

See the discussion of modifying the order priority mapping in *Siebel Order Management Guide Addendum for Communications, Employee Asset-Based Ordering* for more information about priority values in Siebel CRM.

## Supporting Split Billing on Orders

When placing orders from Siebel CRM, you can split bills for service bundles among multiple accounts for consumer and business customers.

You can split bills:

- For a nonpaying child account among multiple parent accounts and multiple billing profiles.
- For a paying child account among the child account and one or more parent accounts and multiple billing profiles.
- For multiple child accounts among multiple parent accounts on the same order.

You split bills on an order by specifying a billing account on the order line that is different than the service account.

For example, Scott subscribes to a double-play promotion that includes a wireless bundle and a broadband bundle. His father and mother, Duncan and Cathy, split the bill. Duncan pays for the wireless service and Cathy pays for the broadband service. In this situation, Scott's service account is a nonpaying child account.

The order lines for Scott's services would include the accounts shown in [Table 8-7](#).

**Table 8-7 Order Including Split Billing**

| Service          | Service Account | Billing Account | Billing Profile |
|------------------|-----------------|-----------------|-----------------|
| Broadband Bundle | Scott           | Duncan          | Duncan-Scott    |
| Wireless Bundle  | Scott           | Cathy           | Cathy-Scott     |

Alternatively, Scott could pay for his wireless service while one of his parents pays for his broadband service. In this situation, Scott's service account is a paying child account.

In the Siebel CRM account hierarchy, the service account on the order line must be subordinate to the billing account, but the service account does not need to be the immediate child of the billing account. For example, Scott's service account could be the child of Cathy's billing account, which in turn could be the child of Duncan's billing account.

When the service account for an order line is different than the billing account, the integration creates a **/billinfo** hierarchy for the service account while creating the account in BRM. See "[Supporting Split Billing](#)" for more information about the **/billinfo** hierarchy in BRM.

Either parent or child can pay for account-level products, but the same account must pay for all account-level products for the lifetime of the account. For example, if Cathy initially pays for Scott's account-level products, neither Duncan nor Scott can ever take over or pay for new account-level products for that account.

You can split the bills for service bundles and simple service bundles nested within service bundles. However, the same account must pay for the top-level service bundle and any service

bundle component that is not a nested service bundle or a simple service bundle. For example, Scott's wireless bundle includes a nested service bundle representing voicemail, another nested service bundle representing SMS, and a one-time charge for activating the line. Scott's parents want to split the bill for these components. The CSR specifies Cathy's account and billing profile for the voicemail service bundle and Duncan's account and billing profile for the SMS service bundle. The CSR also specifies Cathy's account and billing profile for the top-level wireless service bundle and the one-time activation charge.

When legal owners are enabled and you are splitting bills between a self-paying child account and the legally-responsible parent account, you must submit separate orders for the self-paid and parent-paid services. See "[Supporting Legal Owners on Orders](#)" for more information about legal owners and how AIA supports split billing for legal owners.

## Supporting Corporate Account Hierarchies on Orders

A corporate account hierarchy represents the relationship in the account management application between a corporation and its employees. You create account hierarchies in Siebel CRM and synchronize them to BRM at order time.

When placing orders for accounts in a corporate account hierarchy, the integration synchronizes the linear hierarchy of the service account and the billing account on the order. Any siblings of the accounts on the order that are not directly related to the service account or billing account are not synchronized.

Oracle recommends bundling account-level products on the order with a service. Bundling account-level products with services for corporate accounts prevents these products from being tracked in the account-level balance group, which gives you greater flexibility in transferring accounts to different hierarchies.

All accounts on the order must have the same account type. By default, the integration synchronizes the corporate hierarchy for accounts with the account type of **BUSINESS**. You can change this value in the **O2C.CorporateHierarchyAccountType** property of the **AIAConfigurationProperties.xml** file.

See "[About Corporate Account Hierarchies](#)" for more information about creating and synchronizing corporate account hierarchies and setting the type of account for which the integration synchronizes the hierarchy.

## Supporting Legal Owners on Orders

When placing orders for customers who pay their own bills but cannot legally be held responsible for the charges that they incur, you can assign a legal owner to the bill. You assign a legal owner on an order by specifying an owner account on the order line in Siebel CRM that is different than the billing account.

Because each child account can have only one legal owner in BRM, you must ensure that the owner account for all order lines in one order in Siebel CRM is the same. When the integration detects that the owner or customer account and billing account on an order line are different, it creates a collections sharing group in BRM to represent the legal hierarchy.

You enable legal owners using the **O2C.LegalGroup** property in the **AIAConfigurationProperties.xml** file. By default, legal owners are disabled. See "[Enabling and Disabling Legal Hierarchy Synchronization](#)" for more information about enabling legal owners.

For more information about setting up legal hierarchies and synchronizing legal owners to BRM, see "[About Legal Hierarchies](#)".

## Supporting Legal Owners and Split Billing On Orders

When placing orders in Siebel CRM, you can split bills for service bundles among multiple accounts as described in "[Supporting Split Billing on Orders](#)".

To split bills between an owner account and a child account when legal owners are enabled, submit separate orders for the services for which the owner pays and the service for which the child pays. Submitting separate orders ensures that the legal group is created properly for the services for which the child pays.

For example, Helen is a minor who wants to pay for her own wireless service. Her mother, Lisa, agrees to pay for Helen's broadband service. Lisa also agrees to be the legal owner of Helen's services. To order the services, you submit two orders including the account details shown in [Table 8-8](#).

**Table 8-8 Example Order Line with Legal Hierarchy**

| Order | Service   | Service Account | Billing Account | Billing Profile | Owner Account |
|-------|-----------|-----------------|-----------------|-----------------|---------------|
| 123   | Broadband | Helen           | Lisa            | LisaBP-Helen    | Lisa          |
| 456   | Wireless  | Helen           | Helen           | HelenBP         | Lisa          |

Because the billing account and owner account are the same on order 123, the integration does not create a collections sharing group until it processes order 456. When processing order 456, the integration creates a collections sharing group in BRM using the HelenBP billing profile from the order line and the default billing profile for the owner account.

See "[About Legal Hierarchies](#)" for information about how the integration creates collections sharing groups in BRM.

## 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 upload 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.
  - The determination of service point in Siebel CRM is replicated in Service and Resource Inventory (for example, the same result is achieved).
  - The service point attribute value is unique and common across Siebel and Service and Resource Inventory, such that Service and Resource Inventory can use the value directly.
  - The service point attribute value is a cross-reference that is understood by Service and Resource Inventory; no Oracle AIA cross-reference exists for this attribute.
- 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.

# 9

## Implementing the Process Sales Order Fulfillment Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Process Sales Order Fulfillment business flow. It also describes how the integration supports order priorities and multiple price lists.

### About the Process Sales Order Fulfillment Business Flow

The following pre-built integration options enable the Process Sales Order Fulfillment business flow:

- Order to Cash for Oracle Communications Order and Service Management (OSM)
- Order to Cash for Siebel CRM

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

- Submitting Orders from Siebel CRM to OSM

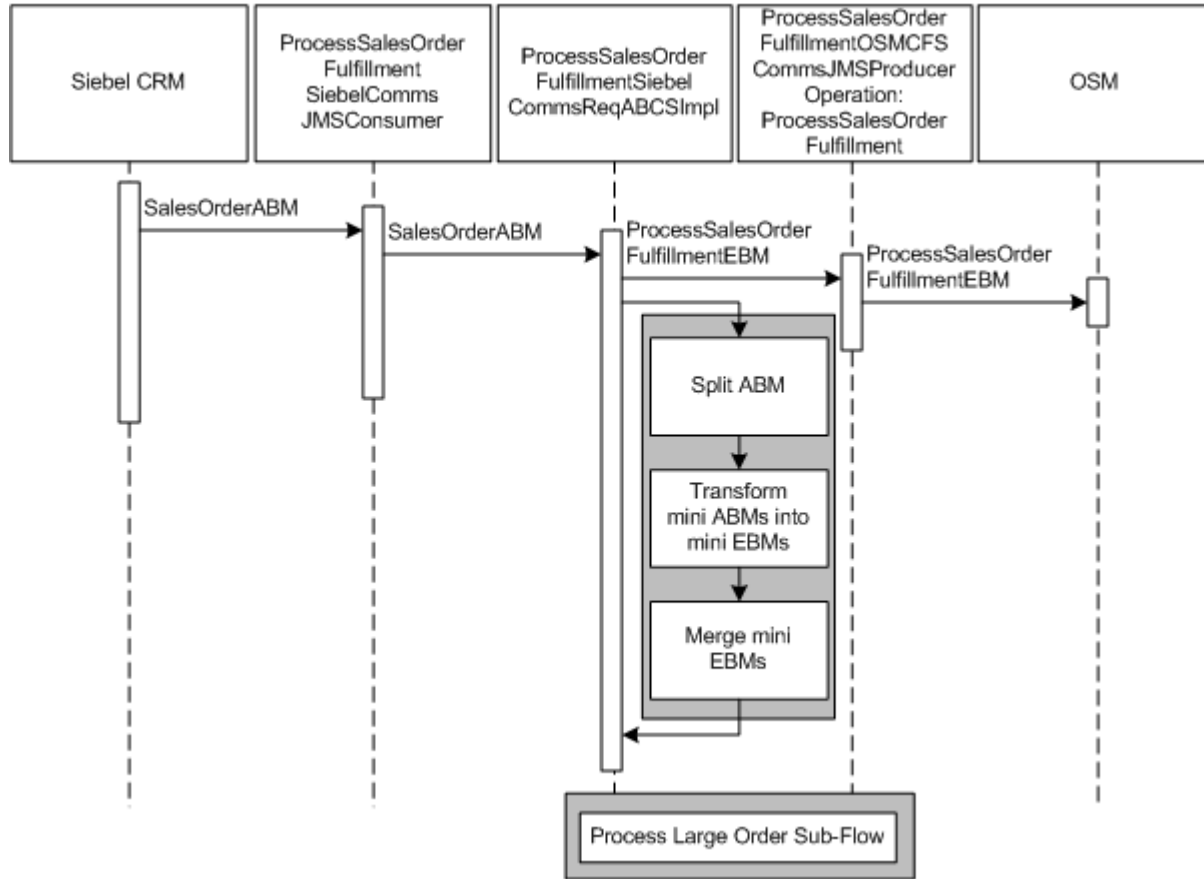
### About the Submitting Orders from Siebel CRM to OSM Integration Flow

This integration flow uses the following services:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

[Figure 9-1](#) shows the sequence of integration services for the Submitting Orders from Siebel CRM to OSM integration flow.

Figure 9-1 Submitting Orders from Siebel CRM to OSM Integration Flow



The integration submits orders from Siebel CRM to OSM as follows:

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 drops the ABM into the AIA\_SALESORDERJMSQUEUE queue.
3. The ProcessSalesOrderFulfillmentSiebelCommsJMConsumer service picks up the ABM and passes it on to the ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl service.
4. ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl checks the values of the handleLargeOrderEnabled and numOrderLinesInLargeOrder properties in the **AIAConfigurationProperties.xml** file and does the following:
  - If the value of handleLargeOrderEnabled is **true** and the number of order lines is less than the value of numOrderLinesInLargeOrder, or if the value of handleLargeOrderEnabled is **false**, ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl transforms the ABM into an enterprise business message (EBM) called ProcessSalesOrderFulfillmentEBM and sends it to the ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer service.
  - If the value of handleLargeOrderEnabled is **true** and the number of order lines is greater than or equal to the value of numOrderLinesInLargeOrder, ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl does the following:

- a. Splits the large SalesOrderABM into mini ABMs. Each mini ABM contains no more order lines than the value of the numOrderLinesInMiniABM configuration property, unless those order lines are part of a Siebel CRM promotion.  
  
Order lines that are part of a promotion always belong to the same mini ABM, even if this ABM has more order lines than the value of numOrderLinesInMiniABM.
- b. Transforms the mini ABMs into mini EBMs.
- c. Merges the mini EBMs into one ProcessSalesOrderFulfillmentEBM and sends it to ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer.

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. ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer 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

[Table 9-1](#) describes the transactions, the database operations, and what actions to take in case of an error for this flow.

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.

The following services are involved:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer



**Table 9-1 Transaction Boundaries and Recovery Details for Siebel CRM Flow**

| 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 ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer. | AIA cross-reference entries. | Roll back cross-reference transactions.<br>Roll back JMS message to AIA_SALESORDERJMSQUEUE_ErrorQ | Resubmit the order from either AIA_SALESORDERJMSQUEUE_ErrorQ or from Siebel CRM. |

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about rollback procedures.

## Supporting Order Priority

The integration supports order priority as follows:

1. A Siebel CRM user submits an order with a string value for priority.
2. The Siebel CRM JMS consumer consumes the message and sends it to the Siebel CRM requester ABCS. While transforming the order message into the EBM, the ABCS looks up the value of the OrderPriority field in the Siebel CRM column of the SALESORDER\_PRIORITY domain value map (DVM) and populates the ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/FulfillmentPriorityCode with the corresponding value from the common column.
3. When creating the OSM create order message, the ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer service looks up the value of the FulfillmentPriorityCode in the common column of the SALESORDER\_PRIORITY DVM and populates the priority in the JMSPriority field in the JMS headers with value from the JMS column using the following BPEL assign activity:

```
<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:JMSPriority"/>
  </copy>
</assign>
```

where:

- *jmsHeaders* is a variable of type JMSOutboundHeadersAndProperties
  - *msg\_priority* is a string variable
4. The store and forward mechanism picks up the messages according to their priority and sends them to OSM.

5. OSM honors the priority through internal mechanisms. Higher priority orders are fulfilled and provisioned first, followed by lower priority orders.

OSM maintains the priority of the orders and populates the FulfillmentPriorityCode element in all outbound messages.

Other producers need not do a DVM lookup because 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.

You can extend Siebel CRM to set JMS priority on the order message. See the discussion of modifying the order priority mapping in *Siebel Order Management Guide Addendum for Communications, Employee Asset-Based Ordering* for more information about priority values in Siebel CRM.

## Supporting Price Lists

The integration 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. When the ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl service receives the ABM, it 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.
3. When OSM receives the CreateOrder message, it transforms it into a ProcessFulfillmentOrderBillingEBM message. During this transformation, OSM populates empty price list order lines with the price list from the order header.
4. OSM passes the ProcessFulfillmentOrderBillingEBM message on for billing. See "[About the Bill Fulfillment Order Business Flow](#)" 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
- OptimizedSalesOrder

The following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/*

- Enterprise business service (EBS) WSDL files:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/*

- OptimizedSalesOrder EBO files:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/PIPS/Communications/Schemas*

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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

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

- [ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer](#)
- [ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl](#)
- [ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer](#)

### ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer

The ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer service is implemented as a Mediator process.

This consumer listens to 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

ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl is 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. It converts the Siebel CRM ABM into the sales order EBM.

Before converting the ABM into an EBM, this service checks whether it should use the Process Large Order sub-flow. The service uses the Process Large Order sub-flow if the handleLargeOrderEnabled property is set to **true** and the number of order lines in the ABM is greater than or equal to the value of the numOrderLinesInLargeOrder property. In the Process

Large Order sub-flow, `ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl` does the following:

1. Splits the ABM into mini ABMs
2. Transforms the mini ABMs into mini EBMs
3. Merges the mini EBMs into one sales order EBM

While transforming the ABM or mini ABMs, the service does the following:

- Looks up the following cross-reference values to find common IDs to appropriately populate the sales order EBM:
  - customer account ID
  - billing profile ID
  - pay profile ID
  - organization ID
  - product or discount ID
  - promotion group member ID
  - price list common ID (from the PRICELIST DVM)

If no cross-reference values exist for promotions and service bundles, the service creates ones.

- Creates the following cross-reference values between Siebel CRM values and generated common values:
  - order ID
  - order line ID
  - installed product ID
  - account ID
  - bill profile ID
  - pay profile ID
  - contact ID
  - address ID
  - balance group ID
  - promotion group member ID
  - asset ID
- Sets the `OrderSubject` attribute to `CUSTOMER`.
- Records any payment IDs on the order in the `SalesOrderReceivedPayment` section of the EBM.

See [Table 25-2](#) 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.

# 10

## Understanding the Synchronize Fulfillment Order Billing Account Business Flow

This chapter describes the Synchronize Fulfillment Order Billing Account business flow and assumptions and constraints for the business flow.

### About the Synchronize Fulfillment Order Billing Account Business Flow

The Synchronize Fulfillment Order Billing Account business flow lets you create data customer in Oracle Communications Billing and Revenue Management (BRM) from orders submitted from Siebel CRM and processed by Oracle Communications Order and Service Management (OSM).

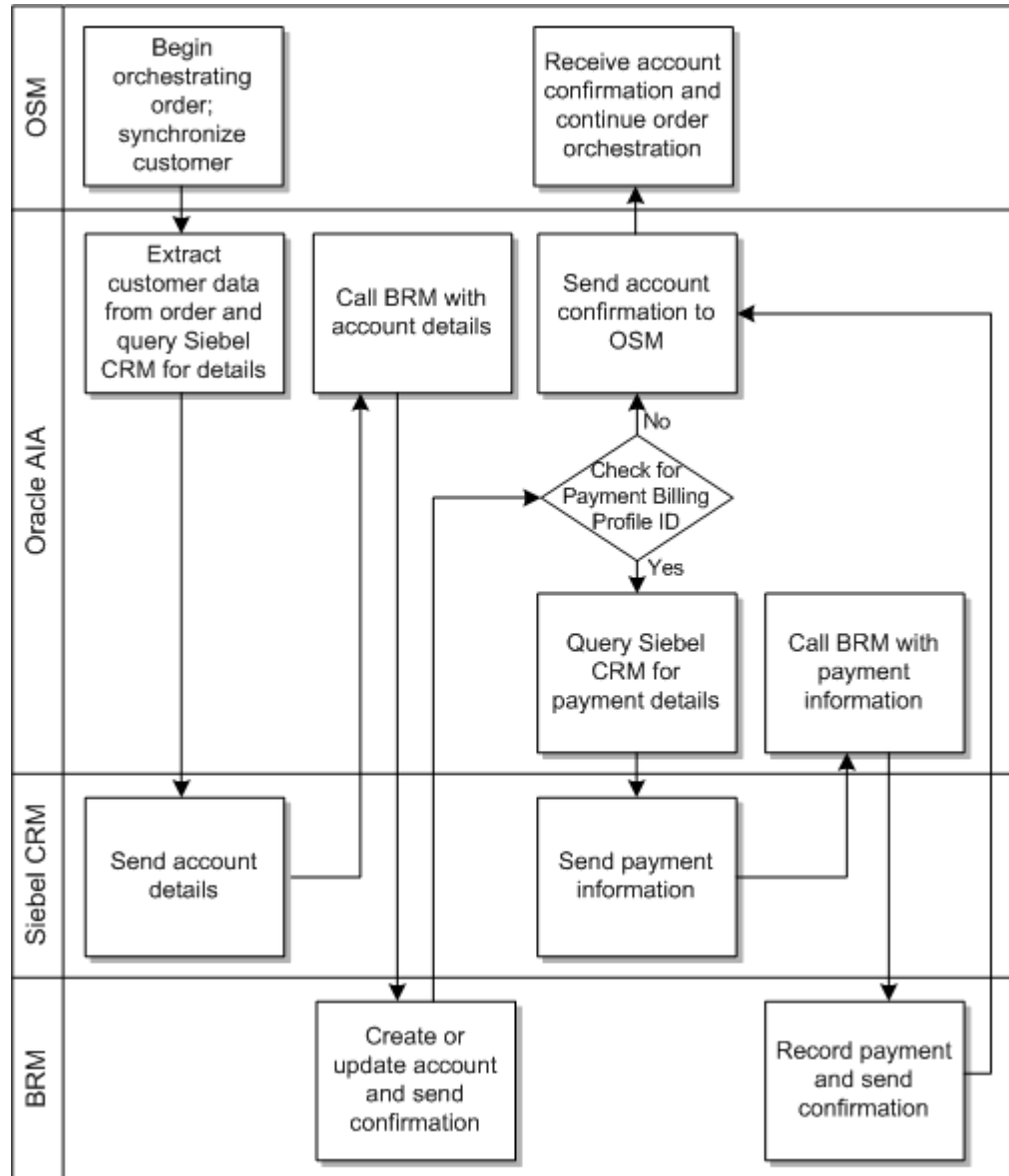
This business flow creates the customer data necessary for order fulfillment without overburdening BRM with all of the customer information available in Siebel CRM. It also lets you process payments accepted at order time.

This business flow includes the following integration flows:

- Interfacing Orders to Create Customer Data
- Payment on Order

Figure 10-1 shows the sequence of events in this business flow.

**Figure 10-1 Synchronize Fulfillment Order Billing Account Business Flow**



## About the Interfacing Orders to Create Customer Data Integration Flow

OSM starts this integration flow during order orchestration to create customer data in BRM using Siebel CRM orders. Integration flows from the process integration for customer management create accounts and their components (such as billing preferences and payment methods) in BRM.

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

This integration flow processes lines with ADD, UPDATE, and MOVE-ADD.

The lines are processed as follows:

- For lines with a billing type of Service Bundle, Item, Subscription, Discount, or Sponsorship, it uses service account, billing account, customer account, and billing profile.

If legal hierarchy synchronization is enabled, it also uses owner account from Siebel CRM. See "[About Legal Hierarchies](#)" for more information about legal hierarchy synchronization.

- For lines with product type of Promotion, it uses Billing Account.
- All other lines are ignored.

The integration flow uses the order data to create accounts, **/billinfo** objects, and **/payinfo** objects in BRM. It cannot inactivate or delete the objects that it creates.

When you call the integration flow again with different customer data than before, it creates only the accounts, **/billinfo** objects, and **/payinfo** objects that do not already exist in BRM.

See "[About the Synchronize Customer Account Business Flow](#)" for more information about what account information is sent to BRM.

## About Creating Account, Billing, and Corporate Hierarchies in BRM

When the service account on an order line is different from the billing account, the integration creates an account hierarchy by synchronizing the service account as a nonpaying child account under the billing account in BRM. It also creates a **/billinfo** hierarchy in BRM. See "[About Account and Billing Hierarchies](#)" for more information.

When the owner account on an order line from Siebel CRM is different from the billing account, the integration recognizes this as a legal hierarchy and creates a collections sharing group in BRM. See "[About Legal Hierarchies](#)" for more information.

When the type of account on the order line matches the value of the **O2C.CorporateHierarchyAccountType** property in the **AIAConfigurationProperties.xml** file, the integration synchronizes the full account hierarchy from Siebel CRM by creating hierarchical account groups in BRM. See "[About Corporate Account Hierarchies](#)" for more information.

## About the Payment on Order Integration Flow

Oracle AIA invokes this integration flow when Oracle AIA receives confirmation that BRM has created or updated a customer's account. If the order contains a payment-associated billing profile ID, the integration gets the payment information from Siebel CRM and sends it to BRM for billing. BRM records the payment and sends confirmation to Oracle AIA.

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

The Synchronize Fulfillment Order Billing Account Business Flow has the same assumptions and constraints as the Bill Fulfillment Order business flow. See "[Assumptions and Constraints for the Bill Fulfillment Order Business Flow](#)" for information.

If you use an order management system other than OSM, your order management system must meet certain expectations. [Table 10-1](#) summarizes what is expected for the action on order lines. OSM and OSM O2A 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, billing profile, or owner account. | 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>If the owner account is changed, the order management system changes the collections sharing group in BRM.</p> <p>The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.</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.  |
| UPDATE                        | Revision                        | No changes to service account, billing account, or billing profile.             | NONE   | No changes.  |

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   | Changes to service account, billing account, billing profile, or owner account.                   | 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>If the owner account is changed, the order management system changes the collections sharing group in BRM.</p> <p>The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.</p> |
| UPDATE                        | Revision   | Cancellation. Manifests as a missing line on the revision or the action changing to a "-" (NONE). | UPDATE   | <p>If the original update line created a new account and billing profile in BRM, then it cannot be undone.</p> <p>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.</p>  |
| 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.  |
| MOVE-ADD                      | Revision   | No changes to service account, billing account, or billing profile.                               | NONE   | No changes.  |

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                        | Changes to billing account or billing profile.  | MOVE-ADD   | The order management system creates customer data in BRM if it does not already exist.<br><br>If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM.<br><br>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.<br><br>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.

# Implementing the Synchronize Fulfillment Order Billing Account Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Synchronize Fulfillment Order Billing Account business flow.

## About the Synchronize Fulfillment Order Billing Account Business Flow

The following pre-built integration options enable the Synchronize Fulfillment Order Billing Account business flow:

- Order to Cash for Oracle Communications Order and Service Management (OSM)
- Order to Cash for Oracle Communications Billing and Revenue Management (BRM)
- Order to Cash for Siebel CRM

This business flow includes the following integration flows:

- Interfacing Orders to Create Customer Data.
- Payment on Order

### Note:

In this business flow, the `CommsProcessBillingAccountListEBF` enterprise business flow (EBF) communicates directly with application business connector services (ABCS).

Although direct communication between EBF and ABCS enhances performance for this pre-built integration flow, Oracle recommends that you follow the canonical model when you develop your own integration flows. In the canonical model, an EBF can invoke only an enterprise business service (EBS), which in turn invokes the ABCS. This model allows the EBF to be application agnostic.

For more information about the canonical model and developing integration flows, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

## About the Interfacing Orders to Create Customer Data Integration Flow

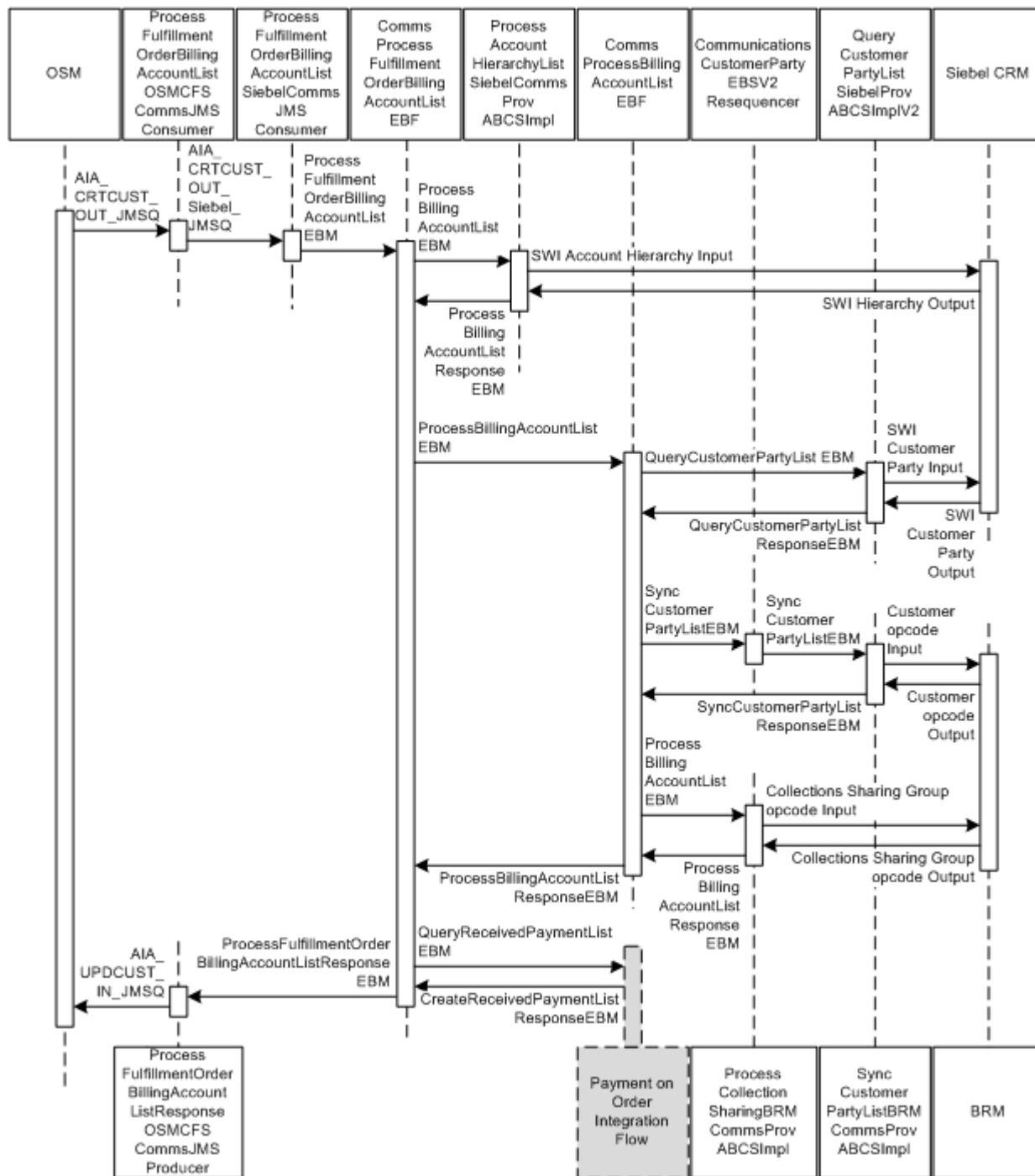
This integration flow uses the following services:

- `ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer`
- `ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer`

- CommsProcessFulfillmentOrderBillingAccountListEBF
- ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer
- From the process integration for Customer Management:
  - ProcessAccountHierarchyListSiebelCommsProvABCImpl
  - CommsProcessBillingAccountListEBF
  - QueryCustomerPartyListSiebelProvABCImplV2
  - CommunicationsCustomerPartyEBSV2Resequencer
  - SyncCustomerPartyListBRMCommsProvABCImpl
  - ProcessCollectionSharingBRMCommsProvABCImpl

Figure 11-1 shows the sequence of integration services for the Interfacing Orders to Create Customer Data integration flow.

Figure 11-1 Synchronize Fulfillment Order Billing Account Sequence



The integration interfaces orders to create accounts in BRM based on Siebel CRM data as follows:

1. OSM adds a message to the AIA\_CRTCUST\_OUT\_JMSQ JMS queue.
2. The ProcessFulfillmentOrderBillingAccountListOSMCFSCOMMSJMSConsumer service subscribed to the queue picks up the message, identifies that the order originally came from Siebel CRM based on the schemeAgencyID in the EBM, and adds the message to the AIA\_CRTCUST\_OUT\_Siebel\_JMSQ queue.

3. The `ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSSConsumer` subscribed to the queue picks up the message and sends it to the `CommsProcessFulfillmentOrderBillingAccountListEBF` service.
4. `CommsProcessFulfillmentOrderBillingAccountListEBF` transforms the `ProcessFulfillmentOrderBillingAccountListEBM` message into the `ProcessBillingAccountListEBM` message.
5. `CommsProcessFulfillmentOrderBillingAccountListEBF` checks the value of the `O2C.CorporateHierarchyAccountType` property.

If the value matches the account type on the order:

- a. `CommsProcessFulfillmentOrderBillingAccountListEBF` sends `ProcessBillingAccountListEBM` to the `ProcessAccountHierarchyListSiebelCommsProvABCSEImpl` service.
- b. `ProcessAccountHierarchyListSiebelCommsProvABCSEImpl` transforms `ProcessBillingAccountListEBM` into an application business message (ABM) and uses it to invoke the Siebel CRM SWI Account Hierarchy web service with the query operation.
- c. The query operation of SWI Account Hierarchy fetches the complete linear account hierarchy from the Siebel CRM database and sends a response ABM to `ProcessAccountHierarchyListSiebelCommsProvABCSEImpl`.
- d. `ProcessAccountHierarchyListSiebelCommsProvABCSEImpl` transforms the response ABM into the `ProcessBillingAccountListResponseEBM` message and sends it to `CommsProcessFulfillmentOrderBillingAccountListEBF`.
- e. `CommsProcessFulfillmentOrderBillingAccountListEBF` transforms `ProcessBillingAccountListResponseEBM` into the `ProcessBillingAccountListEBM` message and sends it to the `CommsProcessBillingAccountListEBF` service.

If the value of `O2C.CorporateHierarchyAccountType` does not match the account type on the order, or there is no value, `CommsProcessFulfillmentOrderBillingAccountListEBF` send `ProcessBillingAccountListEBM` to the `CommsProcessBillingAccountListEBF` service.

6. `CommsProcessBillingAccountListEBF` transforms `ProcessBillingAccountListEBM` into the `QueryCustomerPartyListEBM` message, identifies that the order originally came from Siebel CRM based on the sender ID in the EBM, and sends the message to the `QueryCustomerPartyListSiebelProvABCSEImplV2` service.
7. `QueryCustomerPartyListSiebelProvABCSEImplV2` transforms `QueryCustomerPartyListEBM` into an application business message (ABM) and uses it to invoke the Siebel CRM `SWICustomerParty` web service with the query operation.
8. The query operation of `SWICustomerParty` fetches the account details from the Siebel CRM database and sends a response ABM to `QueryCustomerPartyListSiebelProvABCSEImplV2`.
9. `QueryCustomerPartyListSiebelProvABCSEImplV2` transforms the response ABM into the `QueryCustomerPartyListResponseEBM` message and sends it to `CommsProcessBillingAccountListEBF`.
10. `CommsProcessBillingAccountListEBF` transforms `QueryCustomerPartyListResponseEBM` into the `SyncCustomerPartyListEBM` message and sends it to the `SyncCustomerPartyListBRMCommsProvABCSEImpl` service through the `CommunicationsCustomerPartyEBSV2Resequencer`.
11. `SyncCustomerPartyListBRMCommsProvABCSEImpl` invokes any of the following opcodes to create or update accounts:
  - Create accounts:

- PCM\_OP\_CUST\_COMMIT\_CUSTOMER
  - Update accounts:
    - PCM\_OP\_CUST\_UPDATE\_CUSTOMER
    - PCM\_OP\_CUSTCARE\_MOVE\_ACCT
12. The opcodes send output response messages, which SyncCustomerPartyListBRMCommsProvABCImpl transforms into the SyncCustomerPartyListResponseEBM message and sends to CommsProcessBillingAccountListEBF in an asynchronous delayed response mode.
  13. If the owner account on the order is different than the billing account:
    - a. CommsProcessBillingAccountListEBF transforms SyncCustomerPartyListResponseEBM into ProcessBillingAccountListEBM and sends it to ProcessCollectionSharingBRMCommsProvABCImpl.
    - b. ProcessCollectionSharingBRMCommsProvABCImpl checks if the **/billinfo** POID of the owner account on the order exists in the COLLECTION\_GROUP\_OWNER\_ID cross-reference table and invokes any of the following opcodes:
      - POID PCM\_OP\_COLLECTIONS\_GROUP\_CREATE: If the **/billinfo** POID does not exist in the cross-reference table, creates a collections sharing group with owner account's primary billing profile as owner and the specified billing profile of the billing account as member.
      - PCM\_OP\_COLLECTIONS\_GROUP\_ADD\_MEMBER: If the **/billinfo** POID already exists in the cross-reference table, adds the specified billing profile of the billing account to the collections sharing group of the owner account's primary billing profile.
      - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE\_MEMBER: Deletes the billing profile of the billing account from the collections sharing group of the owner account's primary billing profile.
      - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE: If the member to be deleted is the only member in the collections sharing group, deletes the entire collections sharing group.
    - c. The opcodes send output response messages, which ProcessCollectionSharingBRMCommsProvABCImpl transforms into the ProcessBillingAccountListResponseEBM message.

If the owner account on the order is the same as the billing account, CommsProcessBillingAccountListEBF transforms SyncCustomerPartyListResponseEBM into ProcessBillingAccountListResponseEBM.
  14. CommsProcessBillingAccountListEBF sends ProcessBillingAccountListResponseEBM to CommsProcessFulfillmentOrderBillingAccountListEBF in an asynchronous delayed response mode.
  15. If ProcessFulfillmentOrderBillingAccountListEBM contains a payment-related billing profile ID, CommsProcessFulfillmentOrderBillingAccountListEBF invokes the Payment on Order integration flow. See "[About the Payment on Order Integration Flow](#)".
  16. CommsProcessFulfillmentOrderBillingAccountListEBF transforms ProcessBillingAccountListResponseEBM and CreateReceivedPaymentListResponseEBM into ProcessFulfillmentOrderBillingAccountListResponseEBM messages and sends it to the ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSPProducer service.
  17. ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSPProducer drops a message into the AIA\_UPDCUST\_IN\_JMSQ queue where OSM receives updates.



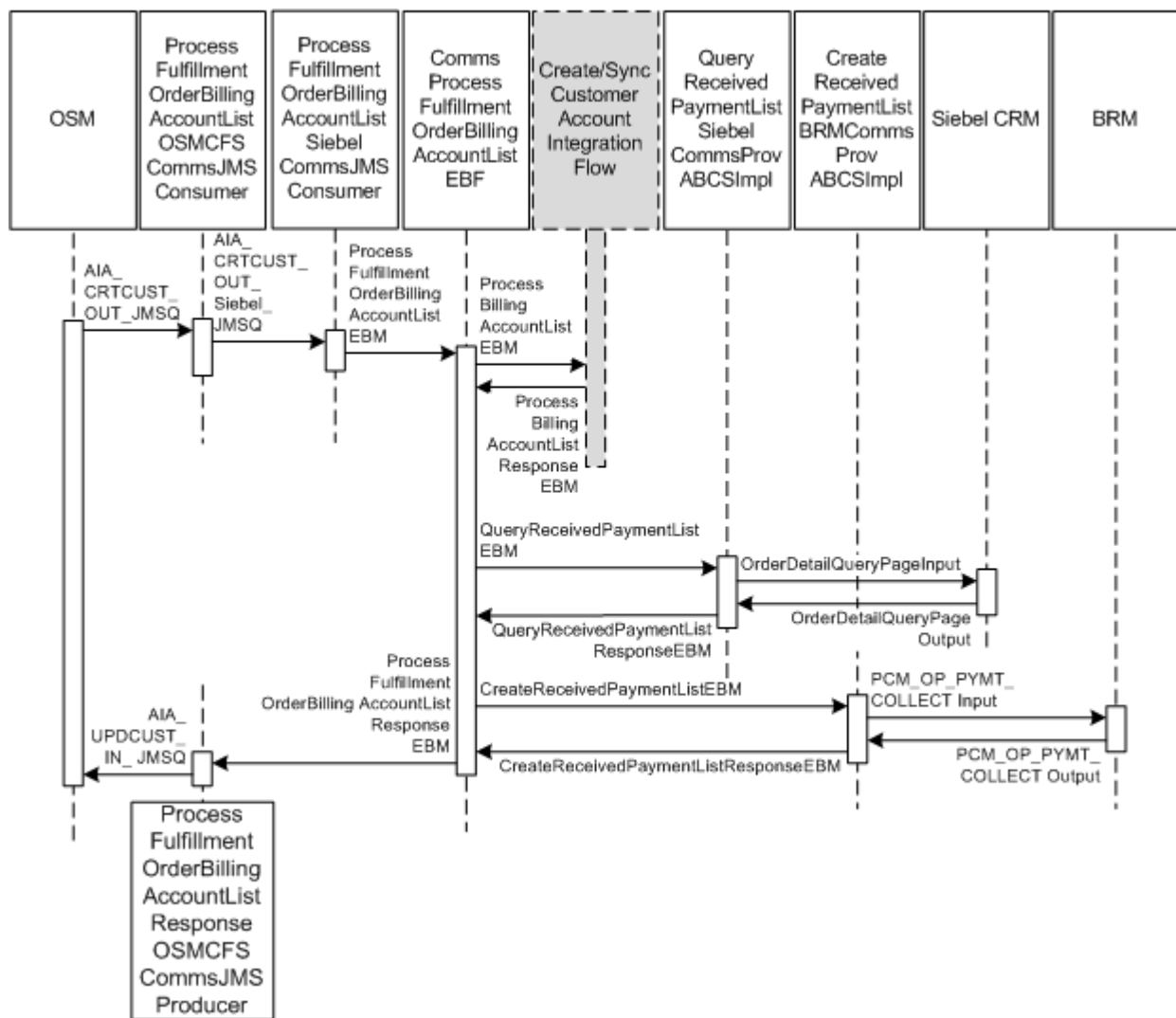
## About the Payment on Order Integration Flow

This integration flow uses the following services:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- QueryReceivedPaymentListSiebelCommsProvABCImpl
- CreateReceivedPaymentListBRMCommsProvABCImpl
- ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer

Figure 11-2 shows the sequence of integration services for the Payment on Order integration flow. This flow is initiated as part of the Synchronize Fulfillment Order Billing Account business flow after the customer has been synchronized to BRM.

Figure 11-2 Payment on Order Integration Flow Sequence



The integration registers payments submitted at order time as follows:

1. OSM drops a message into the AIA\_CRTCUST\_OUT\_JMSQ JMS queue.
2. The ProcessFulfillmentOrderBillingAccountListOSMCFSCCommsJMSConsumer service picks up the message and sends it to the CommsProcessFulfillmentOrderBillingAccountListEBF service.
3. CommsProcessFulfillmentOrderBillingAccountListEBF transforms the ProcessFulfillmentOrderBillingAccountListEBM message into the ProcessBillingAccountListEBM message and initiates the Create/Sync Customer Account integration flow.
4. After the Create/Sync Customer Account integration flow is complete, if ProcessFulfillmentOrderBillingAccountListEBM contains a payment-related billing profile ID:
  - a. CommsProcessFulfillmentOrderBillingAccountListEBF transforms ProcessFulfillmentOrderBillingAccountListEBM into the QueryReceivedPaymentListEBM message and sends it to the QueryReceivedPaymentListSiebelCommsProvABCSImpl service.
  - b. QueryReceivedPaymentListSiebelCommsProvABCSImpl transforms QueryReceivedPaymentListEBM into the OrderDetailQueryPageInput ABM and uses it to invoke the Siebel CRM OrderDetailWebService web service with the query operation.
  - c. OrderDetailWebService gets the payment details corresponding to the payment-associated billing profile ID and sends OrderDetailQueryPageOutput to QueryReceivedPaymentListSiebelCommsProvABCSImpl.
  - d. QueryReceivedPaymentListSiebelCommsProvABCSImpl transforms OrderDetailQueryPageOutput into the QueryReceivedPaymentListResponseEBM message and sends it to CommsProcessFulfillmentOrderBillingAccountListEBF.
  - e. CommsProcessFulfillmentOrderBillingAccountListEBF transforms QueryReceivedPaymentListResponseEBM into the CreateReceivedPaymentListEBM and sends it to the CreateReceivedPaymentListBRMCommsProvABCSImpl service.
  - f. CreateReceivedPaymentListBRMCommsProvABCSImpl invokes the PCM\_OP\_PYMT\_COLLECT opcode to record the payment.
  - g. PCM\_OP\_PYMT\_COLLECT sends a response message, which CreateReceivedPaymentListBRMCommsProvABCSImpl transforms into the CreateReceivedPaymentListResponseEBM message and sends to CommsProcessFulfillmentOrderBillingAccountListEBF.
5. CommsProcessFulfillmentOrderBillingAccountListEBF transforms ProcessBillingAccountListResponseEBM and CreateReceivedPaymentListResponseEBM into ProcessFulfillmentOrderBillingAccountListResponseEBM messages and sends it to the ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSProducer service.
6. ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSProducer drops a message into the AIA\_UPDCUST\_IN\_JMSQ queue where OSM receives updates.

## Defining Transaction Boundaries and Recovery Details

For this business 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.

**Table 11-1 Transaction Boundaries and Recovery Details**

| Transaction  | DB Operations  | In Case of Error  | Recovery   |
|--|--|---|--|
| <p>ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer sends the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which extracts relevant customer data.</p> <p>If account hierarchy is required, the EBF sends the message to ProcessAccountHierarchyListSiebelCommsProvABCImpl, which queries Siebel CRM for the hierarchy.</p> <p>Siebel CRM responds and the ABCS sends the response to CommsProcessFulfillmentOrderBillingAccountListEBF.</p> <p>The EBF then sends the message to CommsProcessBillingAccountListEBF. This EBF sends the message to QueryCustomerPartyListSiebelProvABCImplV2, which queries account details from Siebel CRM. Siebel CRM responds, and the ABCS sends the response to CommsProcessBillingAccountListEBF. The EBF sends the message to CommunicationsCustomerPartyEBSV2Resequencer.</p> | <p>Oracle AIA cross-reference tables updated.</p> <p>Message goes into the sequencer table.</p>  | <p>Message available in Enterprise Manager Fusion Middleware Control in recoverable state.</p>  | <p>Retry or abort the message from Enterprise Manager Fusion Middleware Control.</p> |
| <p>CommunicationsCustomerPartyEBSV2Resequencer sends a message to SyncCustomerPartyListBRMCommsProvABCImpl, which invokes BRM opcodes to create accounts.</p> <p>BRM responds with a confirmation, which the ABCS sends to CommsProcessBillingAccountListEBF.</p> <p>CommsProcessBillingAccountListEBF sends a message to ProcessCollectionSharingBRMCommsProvABCImpl, which invokes BRM opcodes to create collections sharing groups.</p> <p>BRM responds with a confirmation, which the ABCS sends to CommsProcessBillingAccountListEBF.</p> <p>The EBF sends the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which sends it to ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSProducer. The producer drops the message into AIA_UPDCUST_IN_JMSQ.</p>   | <p>Oracle AIA cross reference tables updated.</p> <p>Message goes to AIA_UPDCUST_IN_JMSQ.</p> <p>Accounts and collections sharing groups created in BRM.</p> | <p>Roll back Oracle AIA cross reference updates.</p> <p>Roll back BRM account and collections sharing group creation.</p> <p>Return message to sequencer table.</p> | <p>Resubmit message from sequencer table.</p>  |

 **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 the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about rollback procedures.

## BRM Interfaces

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

**Table 11-2 BRM Interfaces**

| Opcode                                 | Purpose   |
|--|---|
| PCM_OP_BILL_GROUP_MOVE_MEMBER          | As part of the Create/Sync Account integration flow, modifies a hierarchical account group (moves an account between groups, creates a group if one doesn't exist, deletes a group if it is empty). |
| PCM_OP_COLLECTIONS_GROUP_CREATE        | As part of the Create/Sync Account integration flow, creates a collections sharing group with the customer account as the owner and the billing account as a member.                                |
| PCM_OP_COLLECTIONS_GROUP_ADD_MEMBER    | As part of the Create/Sync Account integration flow, adds the billing account as a member of a collections sharing group owned by the customer account.   |
| PCM_OP_COLLECTIONS_GROUP_DELETE_MEMBER | As part of the Create/Sync Account integration flow, removes the billing account as a member of a collections sharing group owned by the customer account.  |
| PCM_OP_COLLECTIONS_GROUP_DELETE        | As part of the Create/Sync Account integration flow, deletes the collections sharing group owned by the customer account.   |
| PCM_OP_CUST_COMMIT_CUSTOMER            | As part of the Create/Sync Account integration flow, creates a new account with one or more <b>/billinfo</b> and <b>/payinfo</b> objects.   |
| PCM_OP_CUST_UPDATE_CUSTOMER            | As part of the Create/Sync Account integration flow, updates account information (name, address, phone), contact information, and billing information.  |
| PCM_OP_CUSTCARE_MOVE_ACCT              | As part of the Create/Sync Account integration flow, moves a child account to a new parent account.   |
| PCM_OP_PYMT_COLLECT                    | Records a payment for the billing account and billing profile on the order.   |

See *Oracle Communications Billing and Revenue Management Developer's Reference* for descriptions of these opcodes.

## Siebel CRM Interfaces

The Synchronize Fulfillment Order Billing Account business flow uses the following Siebel CRM interface:

- OrderDetailWebService: This web service queries orders and retrieves payment details.

See *Siebel CRM Web Services Reference* for more information about this web service.

See "[Siebel CRM Interfaces](#)" in "[Implementing the Synchronize Customer Account Business Flow](#)" for more information about the web services 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
- ProcessFulfillmentOrderBillingAccountListEBM
- ProcessBillingAccountListEBM
- ProcessFulfillmentOrderBillingAccountListResponseEBM
- ReceivedPaymentEBM
- ReceivedPaymentEBO

The following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home\comms\_home\source\soainfra\apps\AIAMetaData\AIAComponents\EnterpriseObjectLibrary\Industry\Communications\EBO\***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home\comms\_home\source\soainfra\apps\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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending 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 as part of the Interfacing Orders to Create Customer Data integration flow:

- [ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMConsumer](#)
- [ProcessFulfillmentOrderBillingAccountListSiebelCommsJMConsumer](#)
- [CommsProcessFulfillmentOrderBillingAccountListEBF](#)
- [QueryReceivedPaymentListSiebelCommsProvABCImpl](#)
- [CreateReceivedPaymentListBRMCommsProvABCImpl](#)
- [ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMProducer](#)

Additional services are delivered with the Synchronize Fulfillment Order Billing Account business flow as part of the Create/Sync Customer Account integration flow, which is part of the Synchronize Customer Account business flow. See "[Integration Services](#)" in "[Implementing the Synchronize Customer Account Business Flow](#)" for more information about these services.

## ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMConsumer

This consumer listens to the AIA\_CRTCUST\_OUT\_JMSQ queue. It picks up ProcessFulfillmentOrderBillingAccountListEBM messages and adds the messages into the AIA\_CRTCUST\_OUT\_Siebel\_JMSQ queue.

## ProcessFulfillmentOrderBillingAccountListSiebelCommsJMConsumer

This consumer listens to the AIA\_CRTCUST\_OUT\_Siebel\_JMSQ queue. It picks up ProcessFulfillmentOrderBillingAccountListEBM messages and sends them to the CommsProcessFulfillmentOrderBillingAccountListEBF service.

## CommsProcessFulfillmentOrderBillingAccountListEBF

This service is an asynchronous delayed response Business Process Execution Language (BPEL) process extracts customer data from the order. It loops through every order line and extracts customer accounts and billing profiles. It uses these data to call the services that query Siebel CRM for account information.

This service does the following:

1. Receives ProcessFulfillmentOrderBillingAccountListEBM from ProcessFulfillmentOrderBillingAccountListOSMCommsJMConsumer or ProcessFulfillmentOrderBillingAccountListSiebelCommsJMConsumer.
2. Transforms ProcessFulfillmentOrderBillingAccountListEBM into ProcessBillingAccountListEBM.
3. If the value for the **O2C.CorporateHierarchyAccountType** Oracle AIA configuration property matches the account type on the order, the service:
  - a. Sends ProcessBillingAccountListEBM to ProcessAccountHierarchyListSiebelCommsProvABCImpl.
  - b. Receives ProcessBillingAccountListResponseEBM from ProcessAccountHierarchyListSiebelCommsProvABCImpl.
  - c. Transforms ProcessBillingAccountListResponseEBM into ProcessBillingAccountListEBM.
4. Sends ProcessBillingAccountListEBM to CommsProcessBillingAccountListEBF.
5. Receives ProcessBillingAccountListResponseEBM from CommsProcessBillingAccountListEBF.

6. If there is a payment-related billing profile ID on the order, the service:
  - a. Transforms ProcessFulfillmentOrderBillingAccountListEBM into QueryReceivedPaymentListEBM and sends it to QueryReceivedPaymentListSiebelCommsProvABCImpl
  - b. Receives QueryReceivedPaymentListResponseEBM from QueryReceivedPaymentListSiebelCommsProvABCImpl
  - c. Transforms QueryReceivedPaymentListResponseEBM to CreateReceivedPaymentListEBM and sends it to CreateReceivedPaymentListBRMCommsProvABCImpl
  - d. Receives CreateReceivedPaymentListResponseEBM from CreateReceivedPaymentListBRMCommsProvABCImpl
7. Transforms ProcessBillingAccountListResponseEBM and CreateReceivedPaymentListResponseEBM into ProcessFulfillmentOrderBillingAccountListResponseEBM and sends it to ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCCommsJMSProducer.

OSM does not expect a response for errors. OSM uses Oracle AIA order fallout notification to generate a trouble ticket and change the order and line status to indicate failure.

If you are using an order management system other than OSM, you can send a response message for errors to the order management system based on the responseCode attribute of the DataArea of the ProcessFulfillmentOrderBillingAccountListEBM message from the order management system.

If the responseCode value is REQUIRED\_FOR\_BUSINESS\_AND\_SYSTEM\_ERRORS, the response message is sent back to the order management system for all errors. If the responseCode value is REQUIRED\_FOR\_BUSINESS\_ERRORS, the response message is only sent back to the order management system for business errors.

## QueryReceivedPaymentListSiebelCommsProvABCImpl

This BEPL process queries Siebel CRM for payment information recorded on sales orders. It processes only payments with the payment status of New.

This service does the following:

1. Receives QueryReceivedPaymentListEBM from CommsProcessFulfillmentOrderBillingAccountListEBF
2. Transforms QueryReceivedPaymentListEBM into an OrderDetailQueryPage\_Input message containing an order ID, and sends the message to OrderDetailWebService
3. Receives the OrderDetailQueryPage\_Output response message from OrderDetailWebService
4. Transforms OrderDetailQueryPage\_Output into QueryReceivedPaymentListResponseEBM and sends it to CommsProcessFulfillmentOrderBillingAccountListEBF

## CreateReceivedPaymentListBRMCommsProvABCImpl

This BEPL process invokes opcodes to create payments in BRM.

This service does the following:

1. Receives CreateReceivedPaymentListEBM from CommsProcessFulfillmentOrderBillingAccountListEBF

2. Transforms CreateReceivedPaymentListEBM into the PCM\_OP\_PYMT\_COLLECT input message containing payment information, and invokes the opcode
3. Receives the response from the opcode
4. Transforms the opcode response into CreateReceivedPaymentListResponseEBM and sends it to CommsProcessFulfilmentOrderBillingAccountListEBF

## ProcessFulfilmentOrderBillingAccountListResponseOSMCFSCommsJMSP roducer

This JMS producer service receives the ProcessFulfilmentOrderBillingAccountListResponseEBM and drops it into the AIA\_UPDCUST\_IN\_JMSQ queue.



# Understanding the Bill Fulfillment Order Business Flow

This chapter provides an overview of the Bill Fulfillment Order business flow. It describes how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) supports and synchronizes order data from Oracle Communications Order and Service Management (OSM) to Oracle Communications Billing and Revenue Management (BRM).

## Overview of the Bill Fulfillment Order Business Flow

The Bill Fulfillment Order business flow is enabled by the following pre-built integration options:

- Order to Cash for Oracle BRM
- Order to Cash for OSM

The Bill Fulfillment Order business flow is initiated when OSM drops a transformed order into a Java Messaging Service (JMS) queue, where a JMS consumer picks it up and passes it to the integration. The integration transforms the messages and passes it to BRM for billing.

## About Interfacing Orders to BRM

This section describes how the integration interfaces order information to BRM as part of the Bill Fulfillment Order business flow.

## Creating and Updating Service Instances

The integration creates or updates service instances, purchased product instances, and discount instances in BRM depending on the action on the order line as follows:

- For order lines with the Add action, the integration creates new service instances, purchased product instances, and discount instances in BRM.
- For order lines with the Update action, the integration updates to the service identifier, billing account, billing profile, or price on existing service instances, purchased product instances, and purchased discount instances in BRM.
- For order lines with the Delete action, the integration cancels the BRM service instances, purchased product instances, and discount instances in BRM. Any refunds and proration are determined in BRM by product level controls.
- For order lines with a Move-Add or Move-Delete action, which are the result of transferring a service from one location to another in Siebel CRM, the integration moves the service instances, purchased product instances, and purchased discount instances in BRM. The integration also makes any updates to the service identifier, billing account, and billing profile. Move-Add and Move-Delete actions cannot include adding new assets or cancelling existing assets, only transferring assets.

## Sending Pricing Information

When interfacing orders, the integration sends pricing information such as price or discount overrides, discounts, and one time penalty charges.

For price changes that occur within a billing cycle, the integration sends the price or discount overrides on a purchased product, the new price goes into effect from the following billing period, and no credits or debits are issued for the current period. To apply the new price immediately, submit an order to cancel the existing product, then submit another order to purchase the product at the new price.

## Applying One-time and Penalty Charges

The integration applies one-time charges for Suspend and Resume actions as service-level charges and penalty charges for changing a promotion agreement as account-level charges. See "[About One-Time Charges for Activating and Changing Services](#)" for information about one-time and penalty charges at design time.

In Siebel CRM, you can define charges for Suspend, Resume, Move, and Delete actions by default. You can extend Siebel CRM to define charges for other actions, such as Update. For example, you can charge a customer a fee for updating their phone number or billing profile. The integration supports one-time and penalty charges regardless of the action that triggered the charge.

Order lines representing one-time and penalty charges are linked to the service bundle line using the asset integration ID and due date from the Siebel CRM order line and the charge parent line from the order enterprise business message (EBM). The integration applies order lines linked to service bundle lines in this way to the corresponding service instance in BRM to create the one-time or penalty 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 is 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 is the same.

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.

## Applying Pricing or Discount Overrides

Pricing and discount overrides are controlled by the following order line attributes:

- **Pricing Commit Type:** 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 applied in BRM as a price or discount override. The BRM General Ledger component accounts for discount overrides but not price overrides.

The integration uses this attribute as follows:

- If the pricing commit type is set to **Committed**, the integration applies a price override in BRM.
- If the pricing commit type is set to **Dynamic**, the integration applies a discount override in BRM and the Dynamic Discount Method attribute is used.

- **Dynamic Discount Method:** Controls whether a discount override is applied by percent or amount.

To use BRM pricing without any overrides for an order line, set the Pricing Commit Type to **Dynamic** and leave the discount blank.

BRM allows only one override for each charge or discount type for a single product. For example, if a BRM product is mapped to multiple events of the same type and synchronized to Siebel CRM as a complex product with multiple simple child products, the integration applies any override applied for one of the child products to all children of the complex product with the same charge or discount type.

## Sending 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 includes the price list ID from the Siebel CRM sales order on the order messages sent to OSM and BRM.

BRM uses the rate plans associated with the price lists to charge the appropriate amount for the products or services purchased on the order lines.

## Using Service Identifiers

When interfacing orders, the integration includes the service identifiers on the service bundle line to BRM. For telephony services, the service identifier is used as the phone number. For other services, it is used as the log in and password.

## Communicating 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 there is a purchase date on 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 integration 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 an 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 are synchronized as bill units (**/billinfo** objects) in BRM.

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 "[About Tracking 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 `Comms_home/source/soainfra/apps/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 uploading changed files to the Oracle Metadata Services repository in *Oracle Application Integration Architecture Installation Guide* 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
- Pay for services of a single service account using multiple billing accounts
- Transfer services from one account to another
- Use sharing groups to share discounts, charges, and extended rating attributes

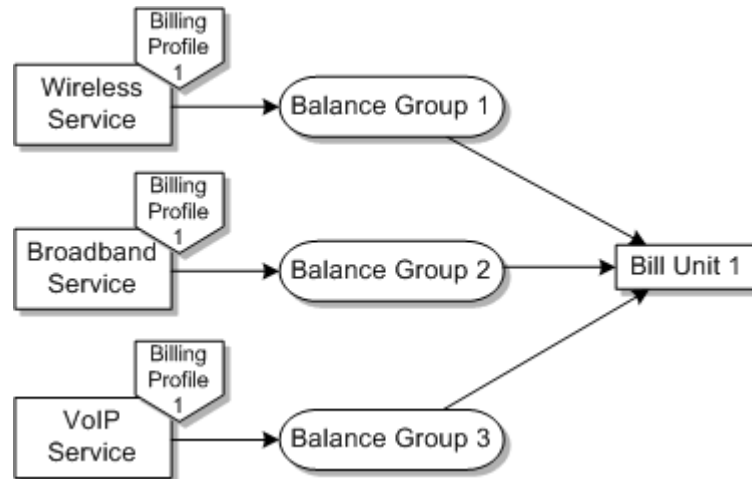
## About Tracking and Billing Services with Service-Level Balance Groups Enabled

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.

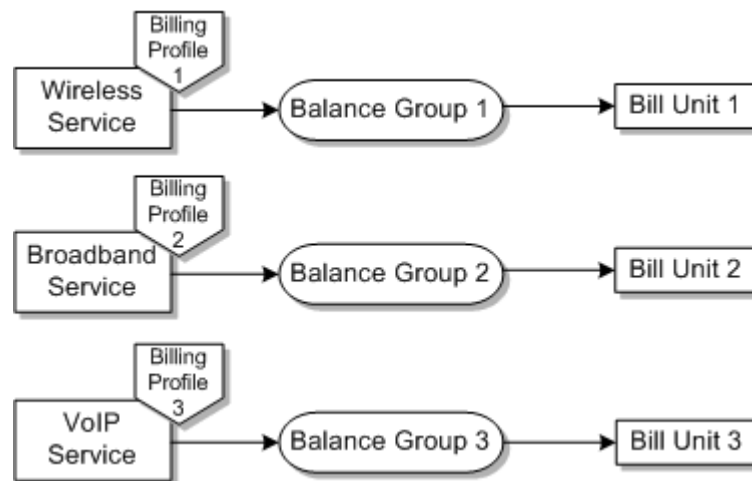
When you submit the order, BRM tracks each service in a separate balance group and the balance groups all point to the same bill unit. [Figure 12-1](#) illustrates this option.

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



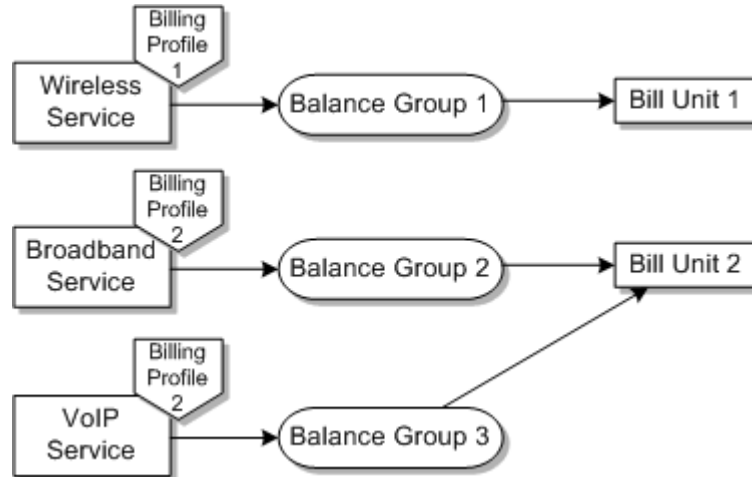
- **Billing all services separately:** You assign each service a separate billing profile in Siebel CRM. When you submit the order, BRM tracks each service in a separate balance group and each balance group points to a separate bill unit. [Figure 12-2](#) illustrates this option.

**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 in Siebel CRM. 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**



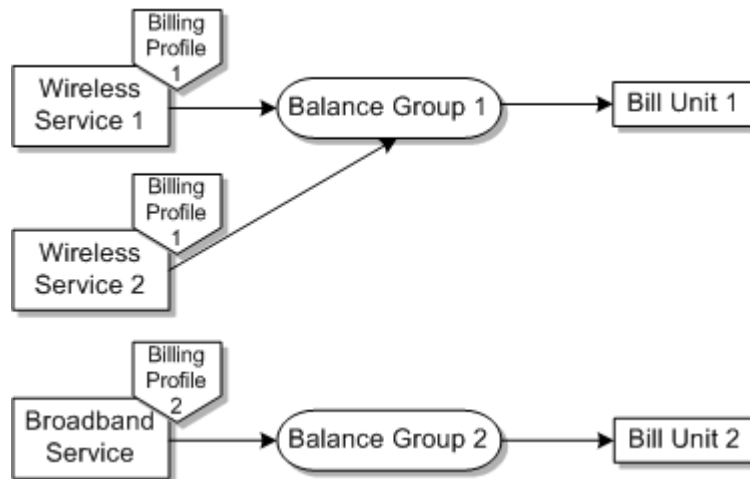
**About Tracking Services in Nested Service Bundles in Service-Level Balance Groups**

Nested service bundles, including nested simple service bundles, must have the same billing profile as their parent service bundle. BRM tracks the nested service bundles in the same balance group as the parent service bundle.

When you submit an order from Siebel CRM, you must manually assign the nested service bundles the same billing profile as their parent service bundle.

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.

## About Tracking 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 Move-Add 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.

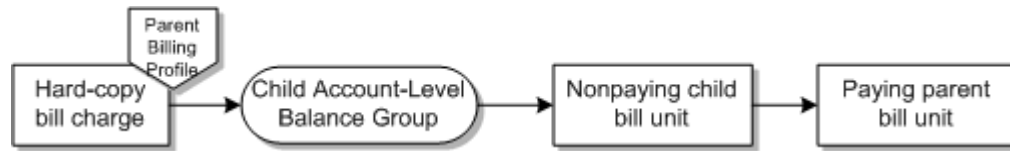
For more information about service bundles and their components, see "[About Service Bundles](#)".

## About Tracking 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 in Siebel CRM, 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.

The account-level balance group of a nonpaying child account is associated with a non-paying bill unit, which points to the parent's paying bill unit. This parent bill unit is unrelated to the parent's account-level balance group. [Figure 12-5](#) illustrates how BRM tracks account-level products for nonpaying child accounts.



**Figure 12-5 Account-Level Products in Nonpaying Child Accounts**

## About Transferring Services in Siebel CRM

The integration supports transferring services tracked in service-level balance groups.

You can transfer services to:

- A different billing profile on the same billing account
- A different billing account
- A different service account

To transfer services, submit a change order to change the service account, billing profile, or billing account 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. 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 Billing Profile

To transfer a service to a different billing profile on the same billing 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.

#### **About Transferring Services to a Different Billing Account**

To transfer a service to a different billing account, submit a change order that lists the target billing account and billing profile on the service bundle order line that you want to transfer.

When you submit the change order, the integration creates a new balance group and bill unit in BRM and assigns it to the target billing account and billing profile. The integration also identifies that the service account is different than the billing account and creates a **/billinfo** hierarchy.

See "[Examples of Changing the Paying Account for Child Accounts](#)" for examples of transferring services to different billing accounts.

#### **About Transferring Services to a Different Service Account**

To transfer a service to a different service 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.

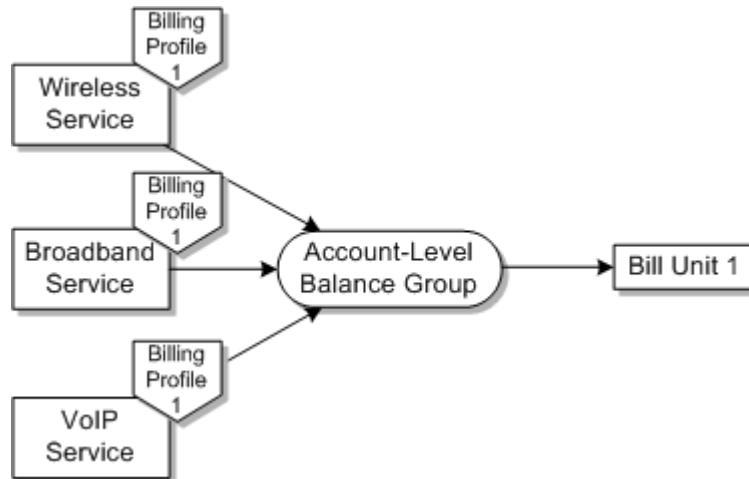
## Working with Service-Level Balance Groups Disabled

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. You cannot use sharing groups or split billing when service-level balance groups are disabled.

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 12-6](#) illustrates how services are tracked under the account-level balance group.

**Figure 12-6 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 12-6](#) 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.

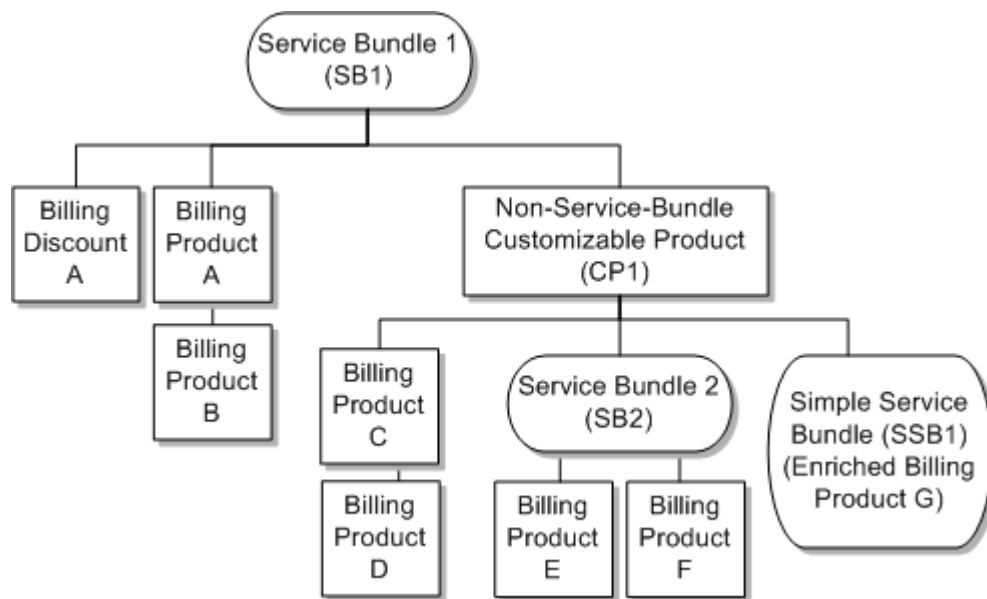
## Examples of Mapping for Bundled Products

This example shows how the integration maps a Siebel CRM 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 12-7](#), 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 12-7 Example of Siebel CRM Service Bundle Hierarchy**

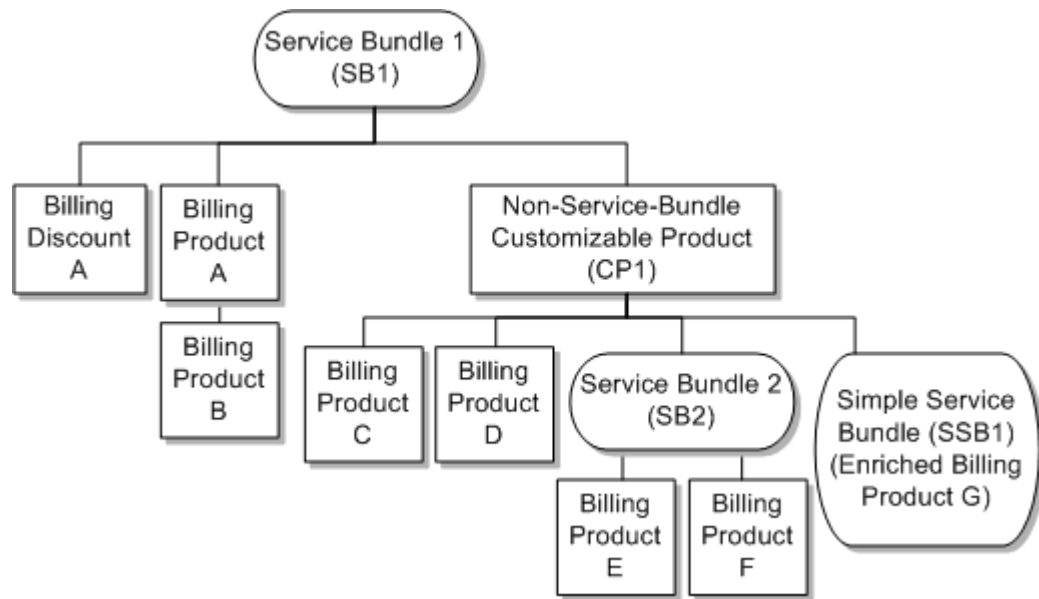


When you submit an order for SB1, the integration creates the following elements in BRM:

- 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 12-8](#).

**Figure 12-8 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 Move-Add or Move-Delete 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 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.

## Synchronizing Promotion Groups

When you submit an order from Siebel CRM containing promotion groups, the integration creates sharing groups in BRM. Sharing groups are collections of accounts consisting of one owner account and one or more member accounts that share a discount, charge, or profile.

The integration uses promotion group line items on sales orders to create sharing group objects in the BRM database as follows:

- **Reward product:** For each reward in the promotion group, the integration creates a separate sharing group object. A single promotion group definition in Siebel CRM can include several reward products of different types, but each BRM sharing group includes only one type of reward, so a single promotion group on a sales order can result in multiple sharing groups in BRM.
- **Owner membership product:** The integration adds the service or account associated with the owner membership product as the sharing group owner. The integration adds the service-level or account-level balance group for that service or account as the sharing group's owning balance group.
- **Member membership product:** The integration adds the services associated with the member membership products as sharing group members. If a bundled promotion is associated with the member membership product, the integration adds each service bundle nested within the bundled promotion as a sharing group member.

You can use change orders to add or delete optional components of bundled promotions that are associated with promotion group member membership products. The integration adds or deletes the components as sharing group members.

The integration also creates an ordered balance group for each member of a sharing group to set the order in which BRM applies shared rewards to usage events. By default, BRM applies discounts first, then chargseshares, then extended rating attributes. Any discounts that a sharing group member owns that are separate from the shared discounts are consumed before the shared discounts.

For example, for the order in [Table 8-5](#), the integration creates the sharing group objects shown in [Table 12-2](#) in BRM.

**Table 12-2 Sharing Group Objects Created in BRM for an Example Promotion Group Order**

| Sharing Group Object     | Owner                    | Owning Balance Group                                     | Reward   | Member                |
|--------------------------|--------------------------|--|--|-----------------------|
| /group/sharing/discounts | Corporate's VoIP Service | Service-level balance group for corporate's VoIP Service | Corporate's VoIP 5000 Free Minutes discount                        | Andrew's VoIP Service |
| /group/sharing/charges   | Corporate's VoIP Service | Service-level balance group for corporate's VoIP Service | Corporate's VoIP 50% Sponsorship chargeshare                       | Andrew's VoIP Service |
| /group/sharing/profiles  | Corporate's VoIP Service | Service-level balance group for corporate's VoIP Service | Corporate's VoIP Employee Special Rating extended rating attribute | Andrew's VoIP Service |

The integration also creates an ordered balance group for Andrew's VoIP service. This ordered balance group is represented in the BRM database by an **ordered\_balgrp** object that ranks the three sharing groups created on the order.

When Andrew uses his VoIP service and generates a usage charge, BRM discounts the charge according to the ordered balance group as follows:

1. BRM applies any discounts Andrew owns that are not part of the sharing group.
2. BRM applies the shared VoIP 5000 Free Minutes discount.
3. If the shared discount is used up, BRM applies the VoIP 50% Sponsorship chargeshare.
4. BRM applies any discounts from an extended rating attribute.
5. BRM bills Andrew for any remaining amount on the usage charge.

For more information about sharing groups and ordered balance groups in BRM, see:

- The discussion of managing resource sharing groups in *BRM Managing Accounts Receivable*
- The discussion of working with profile sharing groups in *BRM Managing Customers*
- The discussion of how discounts and charges are applied in *BRM Managing Accounts Receivable*

## Synchronizing Family Share Plans

When you submit an order from Siebel CRM containing service bundles that have the Community Member attribute enabled and discounts that have the Community Offer attribute enabled, the integration creates sharing groups in the BRM database.

For each order line with the Community Offer attribute enabled, the integration creates a sharing group in BRM. The integration adds the service of the ordering service account as the sharing group owner and the services on the same order with the Community Member attribute as members of that sharing group.

For example, for the order in [Table 8-6](#), the integration detects that the 5 GB Free Data discount has the Community Offer attribute enabled and creates the sharing group object shown in [Table 12-3](#) in BRM.

**Table 12-3 Sharing Group Objects Created in BRM for an Example Family Share Plan Order**

| Sharing Group Object     | Owner                 | Owning Balance Group                                  | Reward                           | Members                             |
|--------------------------|-----------------------|---|----------------------------------|-------------------------------------|
| /group/sharing/discounts | Denise's Primary Line | Service-level balance group for Denise's Primary Line | Denise's 5 GB Free Data discount | Michelle and Jessica's Add-on Lines |

The integration also creates ordered balance groups for Denise's Primary Line and Michelle and Jessica's Add-on Line services, ranking the new shared discount in order with any other discounts that they own.

## Synchronizing Service Grouping

When you submit an order from Siebel CRM containing product promotions with service bundles that have the Service Grouping attribute enabled, the integration creates subscription groups in the BRM database (called service groups in PDC).

For each order line with the Service Grouping attribute enabled, the integration creates a subscription group in BRM, with nested services added as members under the parent service. All of the members of the subscription group are tracked together under one balance group in BRM.

For example, for the order shown in [Table 8-6](#), the integration detects that the Primary Line and Add-on Lines have the Service Grouping attribute enabled and creates the subscription groups shown in [Figure 12-9](#).

**Figure 12-9 Subscription Groups Created in BRM for an Example Family Plan Order**



BRM tracks Denise's three services under one balance group for the Primary Line, Jessica's three services under one balance group for her Add-on Line, and Michelle's three services under one balance group for her Add-on Line.

See the discussion of grouping services by subscription in *BRM Managing Customers* for more information about subscription groups in BRM.

## 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 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).

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 12-4 summarizes revision actions.

**Table 12-4 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

- 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).
- Because only new purchases (lines with action ADD) are processed by billing initiation, revisions are only processed for new purchases.

- 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

Friends and family lists, implemented as special rating products and included in service bundles in Siebel CRM, let your customers call certain phone numbers at discounted rates.

Your customers can share a common friends and family list by purchasing promotion groups that include special rating products as rewards. See "[About Promotion Groups](#)" for more information about promotion groups.

When customer service representatives create orders for service bundles and 0 promotion groups that include special rating products, they create the friends and family lists, optionally add numbers to the lists, and associate the lists with the special rating products.

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 special rating 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 integration assumes that if the same special rating list is referenced by multiple services, such as VoIP and Wireless Voice, those services are fulfilled in the same BRM instance. See "[Configuring Multiple BRM Instances for Communications Integrations](#)" for more information.

See "[About Friends and Family Lists](#)" and the discussion of profiles in *Siebel Communications Guide* for more information about creating special rating product and profiles at design time.

## Using Change Orders with Special Rating Products

You can use change orders in to make changes to special rating products in the following situations:

- Changing special rating list items:
  - You can associate a completely different list with the special rating product using a modify 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 BRM with contents from the new list.

- You can add or remove items from a list currently associated with a special rating product. Use the Siebel Special Rating Profile subview to make changes to the list and synchronize them to BRM.
- Upgrading or downgrading promotions:  
Upgrading or downgrading promotions can add or cancel special rating products for existing services. When special rating products are cancelled, the integration deletes the respective list profile in BRM. When special rating products are added, the integration creates new list profiles in billing for the given service instance.
- Cancelling services:  
When you cancel a service bundle that includes a special rating product, the integration deletes the list profile in BRM.

## Supporting Split Billing

You can split bills among multiple parent and child accounts using **/billinfo** hierarchies. When you submit an order that has order lines that include a billing account that is different from the service account, the integration creates a **/billinfo** hierarchy under the service account while creating the account in BRM.

The **/billinfo** hierarchy includes an entry for each paying **/billinfo** associated with the billing account and billing profile on the order line. When BRM generates the bill for the child account's services, it uses the **/billinfo** hierarchy to assign the bill to the correct paying account.

For example, for the order described in [Table 8-7](#), the **/billinfo** hierarchy under Scott's account would include entries for the **/billinfo** objects corresponding to the Duncan-Scott billing profile and the Cathy-Scott billing profile. When BRM generates the bills for Scott's account, it uses Scott's **/billinfo** hierarchy to send the bill for the broadband service to Duncan and the bill for the wireless service to Cathy.

For more information about **/billinfo** hierarchies and how they relate to account hierarchies in BRM, see the discussion of hierarchical bill units in *BRM Managing Accounts Receivable*.

## 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 integration 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 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.
- For self-paying accounts, the service account, billing account, and billing profiles must be 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.
- For nonpaying child accounts, when service-level balance groups are enabled, the billing account can be different for service bundles and simple service bundles nested within a service bundle. For example, if two parents pay for the services of one child, one service bundle component could list one billing account, while a different component in the same service bundle could list a different billing account. An order including multiple billing accounts for the same service account results in a **/billinfo** hierarchy in BRM.
- The integration does not support changing from nonpaying child account to self-paying account or changing from self-paying account to nonpaying child account. Changing accounts in this way does not produce an error but results in data that breaks the billing management integration flows.
- A nonpaying child account can have multiple paying parents. In Siebel CRM, this relationship is represented in the account hierarchy, and by assigning different billing

accounts to the services for one service account. In BRM, this relationship is represented by a **/billinfo** hierarchy.

- 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.

# 13

## 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 pre-built integration (the integration) implements the business flow.

### About the Bill Fulfillment Order Business Flow

The following pre-built integration options enable the Bill Fulfillment Order business flow:

- Oracle Communications Order to Cash for Oracle Communications Billing and Revenue Management (BRM)
- Oracle Communications Order to Cash for Oracle Communications Order and Service Management (OSM)

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

- Interfacing Orders to Create Transaction Data in BRM

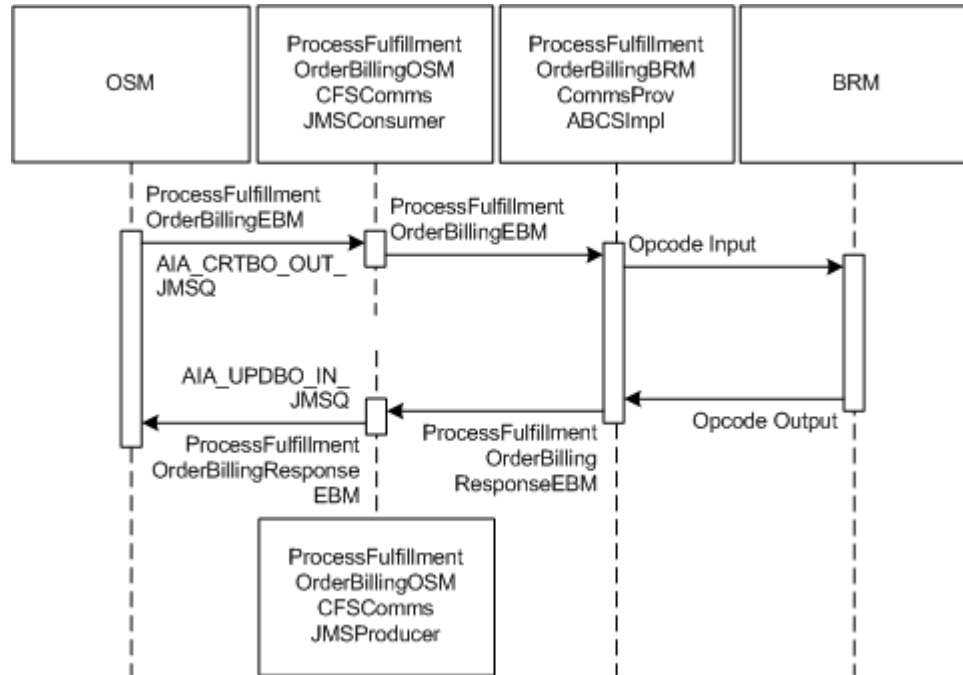
### About the Interfacing Orders to Create Transaction Data in BRM Integration Flow

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

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer

[Figure 13-1](#) shows the sequence of integration services for the Interfacing Orders to Create Transaction Data in BRM integration flow.

**Figure 13-1 Bill Fulfillment Order Sequence**



The integration interfaces orders to create transaction data in BRM as follows:

1. While decomposing and orchestrating an order created by the Process Sales Order Fulfillment business flow, OSM creates a ProcessFulfillmentOrderBillingEBM message and drops it into the AIA\_CRTBO\_OUT\_JMSQ JMS queue.
2. The ProcessFulfillmentOrderBillingOSMCFSCCommsJMSConsumer service picks up ProcessFulfillmentOrderBillingEBM and sends it to the ProcessFulfillmentOrderBillingBRMCommsProvABCsImpl service.
3. ProcessFulfillmentOrderBillingBRMCommsProvABCsImpl uses ProcessFulfillmentOrderBillingEBM to invoke opcodes and create billing data in BRM, including billing artifacts, service instances with balance group information, purchased products, purchased discounts, and sharing groups.
4. The opcodes send output response messages, which ProcessFulfillmentOrderBillingBRMCommsProvABCsImpl transforms into the ProcessFulfillmentOrderBillingResponseEBM message and sends back to the ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer service.
5. ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer drops ProcessFulfillmentOrderBillingResponseEBM into the AIA\_UPDBO\_IN\_JMSQ queue where OSM receives updates.

## 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:

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

**Table 13-1 Transaction Boundaries and Recovery Details**

| Transaction  | DB Operations  | In Case of Error   | Recovery   |
|--|--|--|--|
| The ProcessFulfillmentOrderBillingOSMCFSCCommsJMConsumer 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.<br>BRM data created.<br>Message goes to the AIA_UPDBO_IN_JMSQ queue. | Rollback AIA cross-references.<br>Rollback data created in BRM.<br>Message goes back to the originating queue (AIA_CRTBO_OUT_JMSQ_ErrorQ). | Resubmit the order from the AIA_CRTBO_OUT_JMSQ_ErrorQ queue. |

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about rollback procedures.

## BRM Interfaces

The Bill Fulfillment Order business flow uses these services:

- PCM\_OP\_CUST\_CREATE\_PROFILE
- PCM\_OP\_CUST\_DELETE\_PROFILE
- PCM\_OP\_CUST\_MODIFY\_CUSTOMER
- PCM\_OP\_CUST\_MODIFY\_PROFILE
- PCM\_OP\_CUST\_SET\_STATUS
- PCM\_OP\_CUST\_UPDATE\_SERVICES
- PCM\_OP\_READ\_FLDS
- PCM\_OP\_READ\_OBJ
- PCM\_OP\_SEARCH
- PCM\_OP\_SUBSCRIPTION\_CANCEL\_DISCOUNT
- PCM\_OP\_SUBSCRIPTION\_CANCEL\_PRODUCT
- PCM\_OP\_SUBSCRIPTION\_ORDERED\_BALGRP

- PCM\_OP\_SUBSCRIPTION\_PURCHASE\_DEAL
- PCM\_OP\_SUBSCRIPTION\_SERVICE\_BALGRP\_TRANSFER
- PCM\_OP\_SUBSCRIPTION\_SET\_PRODINFO
- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_CREATE
- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_DELETE
- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_MODIFY
- PCM\_OP\_SUBSCRIPTION\_SET\_BUNDLE
- PCM\_OP\_SUBSCRIPTION\_SET\_DISCOUNT\_STATUS
- PCM\_OP\_SUBSCRIPTION\_SET\_PRODUCT\_STATUS
- PCM\_OP\_SUBSCRIPTION\_TRANSFER\_SUBSCRIPTION

See *Oracle Communications Billing and Revenue Management Developer's Reference* for descriptions of these opcodes.

## Industry Oracle AIA Components

The Bill Fulfillment Order business flow uses these industry components:

- FulfillmentOrderEBO
- ProcessFulfillmentOrderBillingEBM
- ProcessFulfillmentOrderBillingResponseEBM

The following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home/comms\_home/source/soainfra/apps/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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

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

- [ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer](#)
- [ProcessFulfillmentOrderBillingBRMCommsProvABCImpl](#)
  - [ProcessFulfillmentOrderBillingBRMCommsAddSubProcess](#)

- [ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess](#)
- [ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess](#)
- [ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess](#)
- [ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess](#)
- [ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess](#)
- [ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer](#)

## ProcessFulfillmentOrderBillingOSMCFSCCommsJMSSConsumer

The `ProcessFulfillmentOrderBillingOSMCFSCCommsJMSSConsumer` service is a Mediator process that has a JMS Adapter Service, which continuously polls the `AIA_CRTBO_OUT_JMSQ` queue. The `ProcessFulfillmentOrderBillingOSMCFSCCommsJMSSConsumer` 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 performs the following activities:

- Evaluates the product type of the order line and the action code. If the particular order line is a service bundle, subscription, discount, or item and if this line has never been interfaced to BRM, it calls the subprocesses as described in [Table 13-2](#).

**Table 13-2 Subprocesses Called by Action Code, Billing Mode, and Fulfillment Mode**

| Action Code | Billing Mode     | Fulfillment Mode | Subprocess   |
|-------------|------------------|------------------|--|
| ADD         | INITIATE BILLING | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsAddSubProcess</a> and <a href="#">ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess</a> |
| ADD         | FULFILL BILLING  | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsAddSubProcess</a> and <a href="#">ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess</a> |
| ADD         | INITIATE BILLING | REDO             | <a href="#">ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess</a>   |
| ADD         | INITIATE BILLING | UNDO             | <a href="#">ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess</a>   |



**Table 13-2 (Cont.) Subprocesses Called by Action Code, Billing Mode, and Fulfillment Mode**

| Action Code       | Billing Mode    | Fulfillment Mode | Subprocess  |
|-------------------|-----------------|------------------|---|
| DELETE            | FULFILL BILLING | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess</a> and <a href="#">ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess</a> |
| SUSPEND or RESUME | FULFILL BILLING | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess</a>   |
| UPDATE            | FULFILL BILLING | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess</a> and <a href="#">ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess</a> |
| MOVE-ADD          | FULFILL BILLING | DO               | <a href="#">ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess</a> and <a href="#">ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess</a>      |

- When an order has one or multiple special rating products, `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` calls the following opcodes:
  - `PCM_OP_CUST_CREATE_PROFILE`: called on new orders. The POID on the opcode response is populated in the AIA XREF database.
  - `PCM_OP_CUST_DELETE_PROFILE`: called to deleting a special rating product. The POID on the opcode response is deleted from the AIA XREF database.
  - `PCM_OP_CUST_MODIFY_PROFILE`: called to modify the customer's profile after deleting a special rating product.
- For Promotion on Invoice, `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` calls the `PCM_OP_SUBSCRIPTION_SET_BUNDLE` opcode and different values are passed depending on the particular functional operation.
- When the order has promotion group or family plan order lines, `ProcessFulfillmentOrderBillingBRMCommsProvABCImpl` calls the `ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess` service. See "[ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess](#)".
- After all of these activities, the data is cross-referenced to the AIA XREF database.

In case of errors, AIA can send a response message to the order management system, depending on the value of the `responseCode` attribute of the `DataArea` on `ProcessFulfillmentOrderBillingEBM`.

If the value of `responseCode` is `REQUIRED_FOR_BUSINESS_AND_SYSTEM_ERRORS`, AIA sends a response message to the order management system for all errors. If the value of `responseCode` is `REQUIRED_FOR_BUSINESS_ERRORS`, AIA only sends a response message to the order management system for business errors.

 **Note:**

OSM does not expect a response for errors. Instead, OSM uses 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](#) 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, service account, or billing account to another when service-level balance groups are enabled. This includes changing the paying account for services on a child account.

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

- Changing the paying account for services on a child 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 reprinted.

- 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 account for the services on a child account.

ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess converts the ProcessFulfillmentOrderBillingEBM into a BRM-specific message including new billing profile, service account, billing 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 account 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 SUSCRIPTION 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingDelete.

When the action code is Delete and the product type is a service bundle or an account-level subscription or account-level discount, then the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess is called.

The following operations are done by this process:

- When the action code is Delete and the product type is Service Bundle, the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess converts the ProcessFulfillmentOrderBillingEBM into 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.

## ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess

ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess is a BPEL process called by ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl for order lines containing promotion groups, promotion group memberships, and promotion group rewards or for family plan order lines with the Community Member or Community Offer attributes enabled.

ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess calls the following opcodes:

- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_CREATE: called to create a new sharing group. The sharing group POID on the opcode response is added to the Oracle AIA XREF database.
- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_MODIFY: called to add or remove a member or reward to or from an existing sharing group.
- PCM\_OP\_SUBSCRIPTION\_SHARING\_GROUP\_DELETE: called to delete an existing sharing group. The sharing group POID on the opcode response is deleted from the Oracle AIA XREF database.
- PCM\_OP\_SUBSCRIPTION\_ORDERED\_BALGRP: called after adding or deleting a member to or from a sharing group to create, update, or delete an ordered balance group for the member that sets the order in which BRM applies discounts, chargeshares, and extended rating attributes.

## ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer

The ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer 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 describes the Provision Order and Update Fulfillment Order business flows.

## About Order Provisioning

Siebel CRM sends customer order fulfillment requests 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.

While provisioning the order, OSM SOM manages the order lifecycle events of the order. For Cancel and Revision requests from Siebel CRM, OSM generates and executes compensation plans to match the change. OSM also manages order data updates, status updates, and fallout incidents. Throughout the fulfillment process, OSM SOM sends status and data updates to OSM COM.

OSM manages provisioning using OSM Order-to-Activate cartridges, which interact with the Oracle Application Integration Architecture (Oracle AIA) interfaces. This document assumes that you have deployed the required cartridge. See *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture* for more information about Order-to-Activate cartridges.

## About the Provision Order Business Flow

In this business flow, OSM COM creates provisioning orders by passing a transformed sales order to OSM SOM.

OSM COM drops the ProcessProvisioningOrderEBM message, which includes most of the SalesOrderEBO attributes, into the AIA\_CRTFO\_OUT\_JMSQ store-and-forward queue, where OSM SOM picks it up.

There is no response for this message other than error responses that come through the Oracle AIA common error schema. OSM SOM sends updates in the ProcessFulfillmentOrderUpdateEBM message. For interface errors and request failures, the responses are passed to an Oracle AIA Error Handling service, which passes order failure information from OSM SOM to customer order management for order fallout handling.

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

## About the Update Fulfillment Order Business Flow

In this business flow, OSM SOM sends milestone, status, and data updates to OSM COM.

You use OSM COM to manage order status and milestones. You can configure order status management across different fulfillment systems, including OSM SOM. You can also use milestones to track order fulfillment progress. OSM uses fulfillment system responses and



status updates to trigger evaluation of rules that progress the order item status and reach new milestones. These statuses and milestones trigger aggregation rules that update the order-level status.

OSM SOM also sends data updates during order provisioning to populate order attributes used by Siebel CRM and other fulfillment systems.

# Implementing the Provision Order and Update Fulfillment Order Business Flows

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Provision Order and Update Fulfillment Order business flows.

## About the Provision Order and Update Fulfillment Order Business Flows

The following pre-built integration options enable the Provision Order and Update Fulfillment Order business flow:

- Order to Cash for Oracle Communications Order and Service Management (OSM)

The Provision Order and Update Fulfillment Business Order business flows include the following integration flow:

- OSM Fulfillment to OSM Provisioning integration flow

## About the OSM Fulfillment to OSM Provisioning Integration Flow

This integration flow participates in both the Provision Order business flow and the Update Fulfillment Order business flow.

For the Provision Order business flow, this integration flow passes provision order requests from OSM in the central order management role (OSM COM) to OSM in the service order management role (OSM SOM).

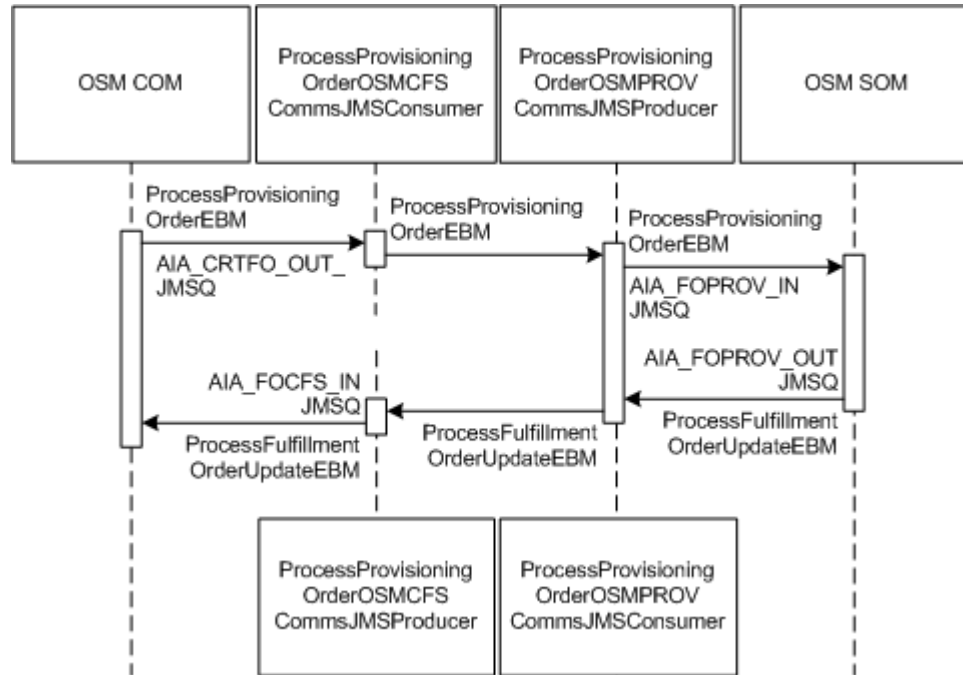
For the Update Fulfillment Order business flow, this integration flow passes milestone, status, and data updates from OSM SOM to OSM COM.

This integration flow uses the following services:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

Figure 15-1 shows the sequence of integration services for the OSM Fulfillment to OSM Provisioning integration flow.

**Figure 15-1 OSM Fulfillment to OSM Provisioning Sequence**



The integration passes provisioning orders to OSM SOM and receives updates in OSM COM as follows:

1. For each new provisioning order, OSM COM creates a `ProcessProvisioningOrderEBM` message and drops it into the `AIA_CRTFO_OUT_JMSQ` queue.
2. The `ProcessProvisioningOrderOSMCFSCommsJMSProducer` service picks up the `ProcessProvisioningOrderEBM` and sends it to the `ProcessProvisioningOrderOSMPROVJMSProducer` service.
3. `ProcessProvisioningOrderOSMPROVJMSProducer` drops the message into the `AIA_FOPROV_IN_JMSQ` queue, where OSM SOM picks it up and processes it.
4. While OSM SOM provisions the order, it sends updates by dropping the `ProcessFulfillmentOrderUpdateEBM` message into the `AIA_FOPROV_OUT_JMSQ` queue.
5. The `ProcessFulfillmentOrderUpdateOSMPROVCommsJMSProducer` service picks up `ProcessFulfillmentOrderUpdateEBM` and sends it to the `ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer` service.
6. `ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer` service drops `ProcessFulfillmentOrderUpdateEBM` into the `AIA_FOCFS_IN_JMSQ` queue where OSM COM picks it up to update the status of the order.

## 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:

- ProcessProvisioningOrderOSMCFSCCommsJMSSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSSConsumer
- ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer

**Table 15-1 Transaction Boundaries and Recovery Details**

| Transaction  | DB Operations | In Case of Error  | Recovery   |
|--|---------------|---|--|
| ProcessProvisioningOrderOSMCFSCCommsJMSSConsumer 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. ProcessFulfillmentOrderUpdateOSMPROVCommsJMSSConsumer routes the message to ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer, 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 the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/*

- Enterprise business service (EBS) WSDL files:

*Oracle\_home/comms\_home/source/soainfra/apps/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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

The following services are delivered with the Provision Order and Update Fulfillment Order business flows:

- ProcessProvisioningOrderOSMCFSCCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer
- ProcessFulfillmentOrderUpdateOSMCFSCCommsJMSProducer

### ProcessProvisioningOrderOSMCFSCCommsJMSConsumer

For interacting with OSM Provisioning, OSM Fulfillment pushes ProcessProvisioningOrderEBM message into AIA\_CRTFO\_OUT\_JMSQ using store-and-forward.

The ProcessProvisioningOrderOSMCFSCCommsJMSConsumer is a Mediator process that has a JMS Adapter Service. This Mediator service continuously polls the AIA\_CRTFO\_OUT\_JMSQ. The ProcessProvisioningOrderOSMCFSCCommsJMSConsumer 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.

## ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

The ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer is a BPEL process that has a JMS Adapter Service. This BPEL process is responsible for pushing the ProcessFulfillmentOrderUpdateEBM Message into the AIA\_FOCFS\_IN\_JMSQ. Using the SAF mechanism, it gets into the appropriate OSM queue.

This service has one operation: Initiate

# 16

## Understanding the Update Sales Order Business Flow

This chapter describes the Update Sales Order business flow and explains the concepts from Siebel CRM that are related to the business flow.

### About the Update Sales Order Business Flow

The Update Sales Order business flow enables the following integration flows:

- Updating sales order data.
- Updating sales order status

### About Updating Sales Order Data

OSM sends updated order data provided by downstream provisioning systems. For example, the service instance ID is blank on the sales order message from Siebel CRM. When a provisioning system assigns the service instance ID and sends a response to OSM, OSM sends an updated sales order message that includes the service instance ID to Siebel CRM.

Because revisions on the order can be submitted up until the point of no return, data updates sent from OSM before the point of no return could be lost. By default, OSM sends data updates only after an order line reaches the point of no return but before the order line is complete. If you are using an order management system other than OSM, ensure that your system follows this restriction.

### About Updating Sales Order Status

OSM sends order and order-line level status updates to Siebel CRM to give visibility into order progress to customer service representatives and self-service customers. OSM automatically limits updates to those that are significant to the customer, and you can customize which updates are sent in the OSM Order to Activate cartridges.

You can configure and send order fulfillment statuses from OSM 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.

The Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) supports the order status attributes listed in [Table 16-1](#).

**Table 16-1 Order Status Attributes**

| Level        | Attribute Name     | Usage   |
|--------------|--------------------|---|
| Order Header | Fulfillment Status | <p>Updates Siebel CRM with the current status of order fulfillment at a high level. The Fulfillment Status attribute tracks the order status while in fulfillment.</p> <p>Possible values include but are not limited to In Progress, Pending, Complete, Canceled, and Failed.</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     | <p>Provides details about the current order status. OSM cartridge implementers can configure this value.</p>  |
| Order Line   | Fulfillment Status | <p>Updates Siebel CRM with the current status of order line fulfillment at a high level.</p> <p>Possible values include but are not limited to In Progress, Pending, Created, Complete, Canceled, and Failed.</p> <p>The OSM cartridge implementer can configure the values for Fulfillment Status attributes.</p>  |
| Order Line   | Milestone          | <p>Indicates the most recent fulfillment milestone reached.</p> <p>Example milestones in OSM:<br/>           SYNC CUSTOMER START: reached<br/>           PROVISIONING START: reached<br/>           FULFILL BILLING COMPLETE: reached</p> <p>The OSM cartridge implementer can configure the values for Milestone attributes.</p>   |
| Order Line   | Status Context     | <p>Provides details about the current order line status. OSM cartridge implementers can configure this value to indicate:</p> <ul style="list-style-type: none"> <li>• Required customer interaction.</li> <li>• If delivery is expected to be delayed.</li> <li>• The milestone or fulfillment function in which a failure occurred.</li> <li>• The cause of a cancellation or who canceled an order.</li> </ul>   |
| 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.) Order Status Attributes**

| Level      | Attribute Name              | Usage  |
|------------|-----------------------------|--|
| Order Line | Actual Delivery Date-Time   | Indicates the date when the purchased product or service is considered available to the customer.<br>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.<br>This date is computed in the OSM fulfillment flow. |
| Order Line | Expected Delivery Date-Time | Indicates the expected delivery date for an order line. Siebel CRM provides this value by default.<br>OSM uses this date to communicate changes for specific order line dates to Siebel CR.  |

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.

Use the following guidelines to help determine which status updates to send to Siebel CRM:

- Not all status changes are relevant to the CSR or the customer. Do not send irrelevant status updates.
- Not all status changes must be reflected instantly. Determine which status changes need to be sent instantly, such as reaching the point of no return, and which do not, and use a throttling mechanism to prevent performance and throughput problems which could result from too many status updates being sent at once.
- Always send Complete and Canceled statuses to Siebel CRM. These trigger actions in Siebel CRM as follows:
  - The Complete status indicates that a service is fulfilled and triggers Siebel CRM to create and update assets. The status can be set 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 status excludes the order from a Siebel CRM calculation of the future state of the asset when creating follow-on or future-dated orders.

# 17

## Implementing the Update Sales Order Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Update Sales Order business flow.

### About the Update Sales Order Business Flow

The following pre-built integration options enable the Update Sales Order business flow:

- Order to Cash for Siebel CRM
- Order to Cash for Oracle Communications Order and Service Management (OSM)

The Update Sales Order business flow includes the following integration flow:

- Updating Statuses from OSM to Siebel CRM

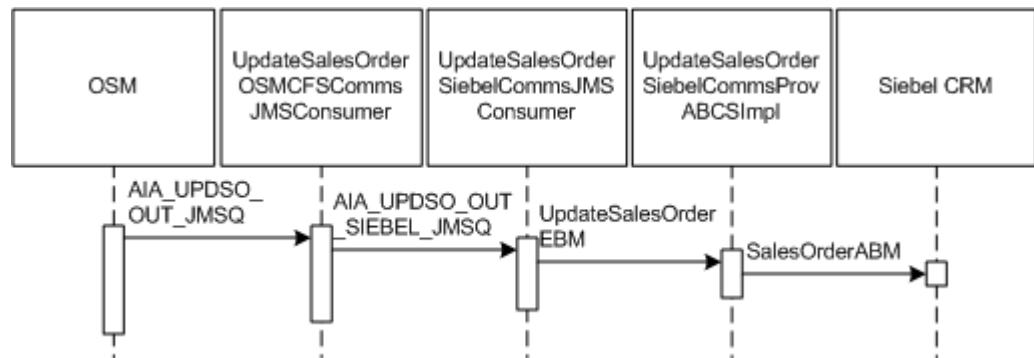
### About the Updating Statuses from OSM to Siebel CRM Integration Flow

This integration flow uses the following services:

- Integration services for OSM:
  - UpdateSalesOrderOSMCFSCommsJMSConsumer
- Integration services for Siebel CRM:
  - UpdateSalesOrderSiebelCommsJMSConsumer
  - UpdateSalesOrderSiebelCommsProvABCImpl

Figure 17-1 shows the sequence of integration services for the Updating Status from OSM to Siebel CRM integration flow.

**Figure 17-1 Updating Statuses from OSM to Siebel CRM**



The integration sends order status updates from OSM in the central order management role (OSM COM) to Siebel CRM as follows:

1. When OSM COM receives status update messages from OSM SOM or BRM, it creates an UpdateSalesOrderEBM message and adds it to the AIA\_UPDSO\_OUT\_JMSQ queue.
2. The UpdateSalesOrderOSMCFSCCommsJMSConsumer service subscribed to the queue picks up the EBM.
3. The consumer service adds the EBM to the AIA\_UPDSO\_OUT\_SIEBEL\_JMSQ queue.
4. The UpdateSalesOrderSiebelCommsJMSConsumer service subscribed to the queue picks up the EBM and sends it to the UpdateSalesOrderSiebelCommsProvABCImpl service.

 **Note:**

UpdateSalesOrderSiebelCommsJMSConsumer has an error resequencer. 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 "[Supporting Order Priority](#)" and "[Using the Oracle Mediator Resequencer Feature](#)" for more information about using sequencing logic to make updates to Siebel CRM.

5. The provider service transforms the EBM into a Siebel CRM application business message (ABM) and invokes the Siebel CRM web service to update the order.

During the transformation, UpdateSalesOrderSiebelCommsProvABCImpl copies fulfillment status from the EBM to the DeliveryStatus attribute of the ABM. The Siebel CRM user interface shows this status in the Fulfillment Status field, which indicates the status of the order in OSM COM while the order is being fulfilled. The Fulfillment Status is a sub-status to the overall order status, which represents the overarching status throughout order capture and order fulfillment. The Siebel CRM user interface shows the overarching status in the Status field.

## 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
- UpdateSalesOrderSiebelCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCImpl

**Table 17-1 Transaction Boundaries and Recovery Details for the Siebel CRM Flow**

| Transaction  | DB Operations                            | In Case of Error  | Recovery  |
|--|--|---|---|
| UpdateSalesOrderOSMCFSComm<br>sJMSConsumer picks up<br>UpdateSalesOrderEBM from<br>AIA_UPDSO_OUT_JMSQ and<br>adds it to<br>AIA_UPDSO_OUT_SIEBEL_JMSQ<br>.  | Message added to<br>queue.               | Roll back the message to<br>AIA_UPDSO_OUT_JMS<br>Q_ErrorQ.  | Resubmit the order from<br>AIA_UPDSO_OUT_JMSQ_Err<br>orQ.                           |
| UpdateSalesOrderSiebelCommsJ<br>MSConsumer picks up<br>UpdateSalesOrderEBM from<br>AIA_UPDSO_OUT_SIEBEL_JMSQ<br>and adds it to the sequencer table<br>defined by the<br>UpdateSalesOrderSiebelCommsJ<br>MSConsumer_RS routing service. | Message added to the<br>sequencer table. | Message available in<br>Enterprise Manager<br>Fusion Middleware<br>Control in recoverable<br>state. | Retry or abort the message<br>from Enterprise Manager<br>Fusion Middleware Control. |
| UpdateSalesOrderSiebelCommsJ<br>MSConsumer_RS sends the<br>message to<br>UpdateSalesOrderSiebelCommsPr<br>ovABCImpl, which invokes the<br>Siebel web service to update the<br>order.   | AIA cross-reference<br>entries.          | Roll back the message to<br>the sequencer table.  | Resubmit the order from the<br>sequencer table.                                     |

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

**`Oracle_home/comms_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/`**

- Enterprise business service (EBS) WSDL files:

**`Oracle_home/comms_home/source/soainfra/apps/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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

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

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderSiebelCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCImpl

### UpdateSalesOrderOSMCFSCommsJMSConsumer

UpdateSalesOrderOSMCFSCommsJMSConsumer is a Mediator process with a JMS Adapter Service. It is subscribed to the AIA\_UPDSO\_OUT\_JMSQ queue.

This process does the following:

1. Picks up the UpdateSalesOrderEBM message.
2. Adds UpdateSalesOrderEBM into the AIA\_UPDSO\_OUT\_SIEBEL\_JMSQ queue.

### UpdateSalesOrderSiebelCommsJMSConsumer

UpdateSalesOrderSiebelCommsJMSConsumer service is a Java Messaging Service (JMS) consumer service. It is subscribed to the AIA\_UPDSO\_OUT\_CPQ\_JMSQ into which order update EBMs are added.

This consumer picks up UpdateSalesOrderEBM and sends it to the UpdateSalesOrderSiebelCommsProvABCImpl service by using a Best Effort resequencer.

### UpdateSalesOrderSiebelCommsProvABCImpl

The UpdateSalesOrderSiebelCommsProvABCImpl is service accepts the UpdateSalesOrderEBM message and uses the information in the input message to update order 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 enabled to use Session Pool Manager.

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 describes the process integration for customer management and related concepts from Siebel CRM and Oracle Communications Billing and Revenue Management (BRM).

## About the Process Integration for Customer Management

The process integration for Customer Management lets you synchronize customer information to BRM from Siebel CRM.

You create and update customer data in Siebel CRM and the Oracle Application Integration Architecture (Oracle AIA) Communications Order to Cash pre-built integration (the integration) synchronizes the account information to BRM during order processing. This is a one-way synchronization process; changes made to customers in BRM are not synchronized to Siebel CRM.

The integration does not load customer data in bulk into BRM. Instead, the integration synchronizes new accounts and billing information to BRM while processing the first order for those accounts.

The integration also synchronizes updates to the accounts from Siebel CRM to BRM.

The process integration for Customer Management includes the following business flows:

- Synchronize Customer Account
- Synchronize Customer Special Rating Profile

## About the Synchronize Customer Account Business Flow

The Synchronize Customer Account business flow creates customer data in BRM from order data while processing orders as part of the process integration for order lifecycle management.

You capture account information when creating orders. You begin creating an order in Siebel CRM by searching for an existing account or creating a new account for a new customer.

Account information that you capture at order time includes billing preferences such as bill medium and frequency, payment type, billing type, billing contact, and bill cycle data. It also includes parent-child or corporate hierarchy relationships.

The Synchronize Customer Account business flow includes the following integration flows:

- Create/Sync Customer Account: Creates customer data in BRM when you submit orders for processing. Supported for orders from Siebel CRM.
- Update Customer Account: Updates existing customer data in BRM when you when you update accounts in Siebel CRM.

See "[Understanding the Synchronize Fulfillment Order Billing Account Business Flow](#)" for more information about where customer synchronization fits in to order fulfillment.

## About the Create/Sync Customer Account Integration Flow

Figure 18-1 illustrates the Create/Sync Customer Account integration flow.

Figure 18-1 Create/Sync Customer Account Integration Flow

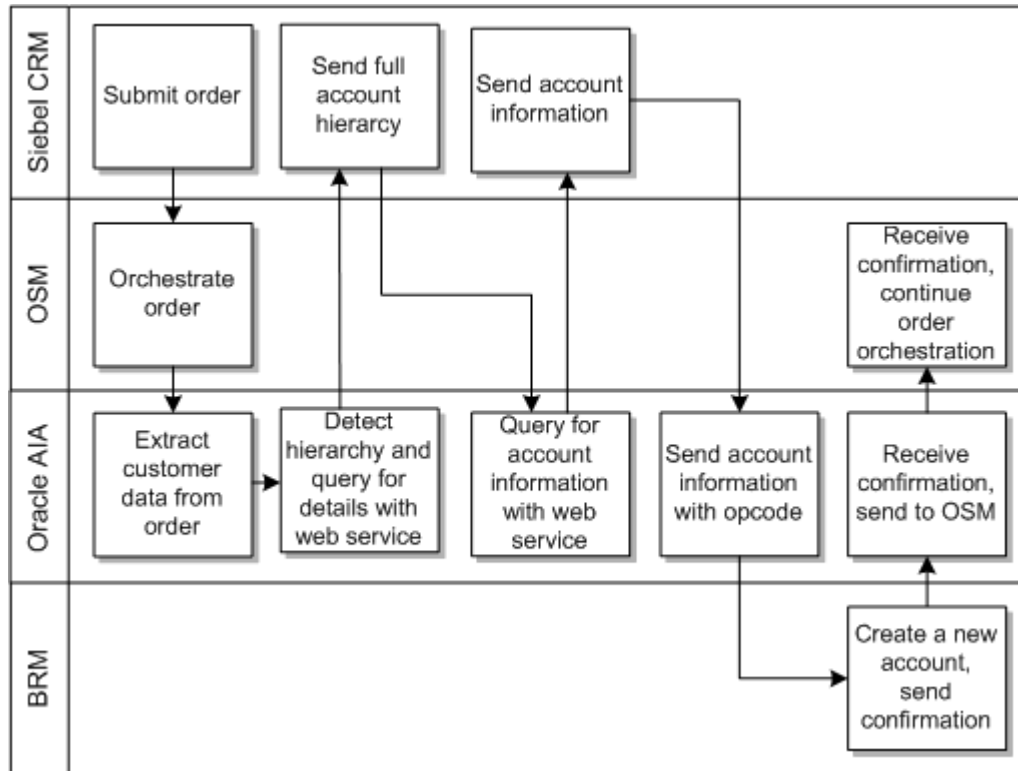


Table 18-1 provides information on Siebel CRM attributes mapped to BRM as part of the Create/Sync Account integration flow.

Table 18-1 Siebel Entities Created or Synchronized to BRM

| Entity or Attribute in the Siebel CRM | Entity or Attribute in BRM | Notes  |
|---------------------------------------|----------------------------|--|
| Account                               | Account                    | <p>BRM sets the account status to <b>Active</b> by default.</p> <p>For accounts of the type specified in the <b>O2C.CorporateHierarchyAccountType</b> property, the integration queries the linear hierarchy of accounts included on the order from Siebel CRM and uses the hierarchy to create hierarchical account groups in BRM.</p> <p>For all other account types, if the billing account and service account on the order line are different, the integration creates a <b>billinfo</b> hierarchy and a two-level account hierarchy in BRM. The Siebel CRM service account is a BRM child account and the Siebel CRM billing account is the BRM parent account. If there is more than one billing account on the order, the first billing account is the parent account.</p> |



**Table 18-1 (Cont.) Siebel Entities Created or Synchronized to BRM**

| Entity or Attribute in the Siebel CRM | Entity or Attribute in BRM  | Notes  |
|---------------------------------------|-----------------------------|--|
| --                                    | Account Number              | Oracle AIA sets this to the Common ID.   |
| Account Type                          | Account Type                | Valid Siebel CRM Account Type values are <b>Residential</b> and <b>Business</b> .<br>Uses the CUSTOMERPARTY_TYPECODE DVM.  |
| Name                                  | Company Name                | Only set for Account Type of <b>Business</b> .   |
| Currency                              | Currency                    | Uses the CURRENCY_CODE DVM.  |
| <b>Contact</b>                        | --                          | The integration synchronizes the account's primary contact to BRM.   |
| Mr/Mrs                                | Salutation                  | Uses the CONTACT_SALUTATION DVM.   |
| First Name                            | First Name                  | --   |
| Last Name                             | Last Name                   | --   |
| Phone                                 | Phone Number                | The integration maps different Siebel CRM phone number types (home, work, fax, mobile) to BRM Phone Type and Number using the PHONENUMBER_TYPE DVM.<br>The phone number format should match the supported format in BRM.<br>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. |
| Job Title                             | Job Title                   | --   |
| Email                                 | Email                       | --   |
| <b>Address</b>                        | --                          | The integration synchronizes the account's primary address to BRM.   |
| Address                               | Address                     | In addition to Address, fields for City, State, Postal Code, and Country are mapped.<br>Uses the following DVMs:<br>ADDRESS_COUNTRYID,<br>ADDRESS_COUNTRYSUBDIVID, PROVINCE, STATE.  |
| <b>Billing Profile</b>                | <b>BillInfo</b>             | --   |
| Name                                  | Name                        | --   |
| Frequency                             | Billing Frequency in Months | Uses the CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE DVM   |
| --                                    | Currency                    | Integration passes account-level currency.<br>Uses the CURRENCY_CODE DVM   |
| Billing Schedule                      | Billing Day of Month        | If the Billing Schedule is not set in and sent from Siebel CRM, then BRM defaults the Billing Day of Month.<br>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.   |
| --                                    | <b>PayInfo</b>              | --   |

**Table 18-1 (Cont.) Siebel Entities Created or Synchronized to BRM**

| Entity or Attribute in the Siebel CRM | Entity or Attribute in BRM | Notes   |
|---------------------------------------|----------------------------|---|
| Payment Method                        | Payment Method             | Only Bill Me, Credit Card or Auto-Debit is supported.<br>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.<br>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.<br>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.<br>Uses the following DVMs:<br>ADDRESS_COUNTRYID,<br>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

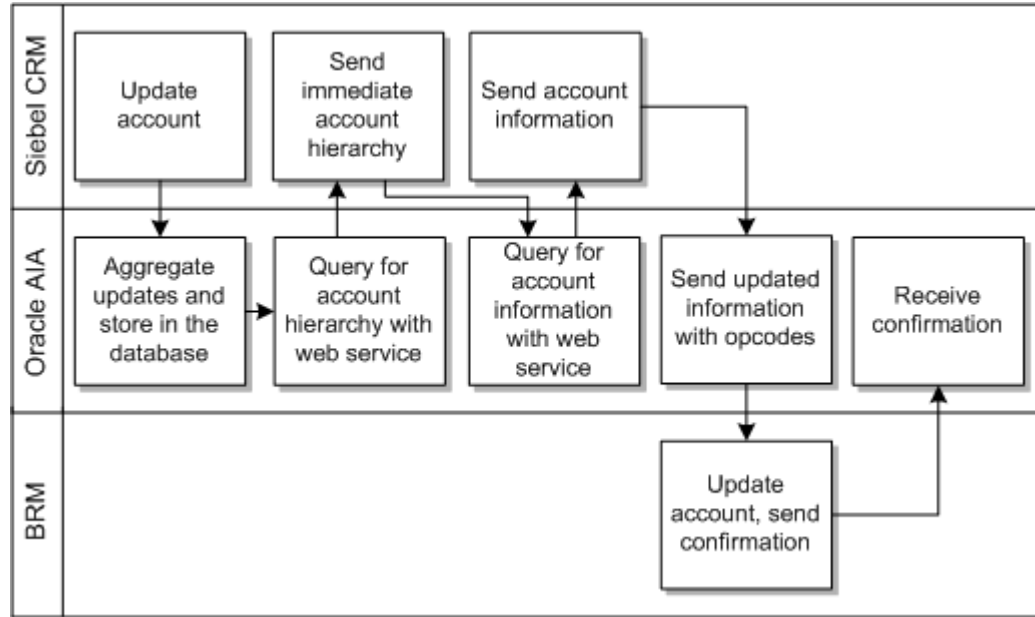
When customers call to change their account information, such as contact, payment, billing, and hierarchical information, a customer service representative (CSR) updates the accounts in Siebel CRM. The integration synchronizes customer account updates to BRM in real time through the Update Customer Account integration flow.

Updates are synchronized to BRM only for accounts that have already been created with Siebel CRM through the Create/Sync Customer Account integration flow as part of the order fulfillment flow.

The process integration can optionally synchronize account status updates from Siebel CRM to BRM. See "[About Account Status Synchronization](#)" for more information.

[Figure 18-2](#) illustrates the Update Customer Account integration flow.

Figure 18-2 Update Customer Account Integration Flow



## About Account Status Synchronization

You can synchronize account status changes from Siebel CRM to BRM. Account status synchronization enhances the process integration for collections management, which is delivered by the Agent Assisted Billing Care pre-built integration. Oracle recommends that you enable account status synchronization only if you are also using the process integration for collections management.

The Agent Assisted Billing Care pre-built integration synchronizes collections actions generated by BRM as credit alerts in Siebel CRM, where a CSR can take actions on the customer's account such as suspending or canceling services.

You can suspend or cancel services with change orders that are either manually submitted by a CSR or automatically generated based on credit alerts. extend Siebel CRM to automatically generate change orders based on credit alerts. Using change orders ensure that service state changes are synchronized from Siebel CRM to BRM.

If you must inactivate a customer account due to continued delinquency, enabling account status synchronization ensures that account status change in Siebel CRM is synchronized to BRM.

Synchronizing account status to BRM is disabled by default. You can enable it by changing the value of the **EnableAccountStatusSync** property in the **AIAConfigurationProperties.xml** file. See "[Configuring the Process Integration for Customer Management](#)" for more information.

When inactivating accounts in Siebel CRM, Oracle recommends the following:

- Inactivate accounts in Siebel CRM only after canceling all the services and account-level subscription products for that account in Siebel CRM. When you inactivate an account in Siebel CRM, the status change is immediately synchronized to BRM. BRM cascades status changes from the account to all of its **billinfo** objects, so the services and products in BRM are cancelled as well. If you inactivate the account before cancelling the services and products in Siebel CRM, they continue to appear active in Siebel CRM even after BRM cancels them.

- To avoid inadvertent inactivation of accounts with active services, Oracle recommends restricting the ability to inactivate accounts to particular Siebel CRM users and roles. Siebel CRM does not let you restrict account status changes in other ways.

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 the process integration for collections management.

## About the Synchronize Customer Special Rating Profile Business Flow

After a service that supports special rating has been purchased and the order fulfilled and asseted, you can synchronize changes to friends and family lists from Siebel CRM to BRM by using the Synchronize Customer Special Rating Profile business flow.

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.

## Data Requirements

The integration requires the following data to successfully create customer data in BRM:

- Account Type: **Residential** or **Business**
- Account Class: **Customer**, **Service**, or **Billing**
- In Siebel CRM, accounts can have any number of contacts or addresses associated with them. BRM requires at least the following:
  - The primary contact for the account, including last name
  - The primary address for the account, including city, state or province, country, and zip or postal code
  - The contact and address associated with the billing profile, including city, state, and zip code
  - For a credit card payment method, the credit card number, expiration month and year, and cardholder name (card verification value number is optional).
  - For an automatic debit payment method, the bank routing number and account number.

All billing profiles for an account and its related parent and child accounts must have the same value for Bill Frequency.

Oracle AIA expects OSM to initiate the synchronization of customer accounts while it orchestrates orders. If you are using an order management system other than OSM, you must ensure that your system recognizes changes in the accounts that appear on sales orders, such as owner accounts, service accounts, and billing accounts, by identifying old and new accounts that appear in the `ProcessSalesOrderFulfillmentEBM`.

See [Table 10-1, Communications Orders Dictionary](#), and [Expectations from an Order Management System for Billing Integration](#) for more information about expectations for order management systems other than OSM.

## About Synchronizing Accounts to Multiple BRM Instances

The integration synchronizes accounts to one BRM instance at a time. If you have multiple BRM instances, the integration can synchronize the same customer to additional instances by calling the customer synchronization application business connector service multiple times. See ["Configuring Multiple BRM Instances for Communications Integrations"](#) for information about configuring multiple BRM instances.

## About Account and Billing Hierarchies

Account and billing hierarchies represent different types of relationships between accounts in Siebel CRM and BRM.

### About Account Hierarchies

An account hierarchy represents the relationship in Siebel CRM between a parent and child account (or a business account and remote site accounts). In an account hierarchy, a parent can have more than one child, but a child can have only one parent.

In Siebel CRM, you can create account hierarchies with multiple levels of parents and children.

The integration does not automatically synchronize the account hierarchy to BRM. Instead, it creates account hierarchies in BRM when you submit an order where the billing account and service account on an order line are different. The integration creates the billing account as a BRM parent account and the service account as a BRM child account. If there are different billing accounts for the same service account on the different order lines, the integration sets the first billing account it encounters as the parent account in the hierarchy.

You can configure the integration to synchronize the entire account hierarchy for a particular type of account, such as business accounts in a corporate hierarchy. See ["About Corporate Account Hierarchies"](#) for more information.

### About Billing Hierarchies

A billing hierarchy represents the relationship in BRM between the **/billinfo** object for a child account and the **/billinfo** objects for one or more parent accounts. When you submit an order where the billing account and service account on an order line are different, the integration creates a billing hierarchy in BRM for the service account. Unlike an account hierarchy, the **/billinfo** object for a child account in a billing hierarchy can have multiple parent **/billinfo** objects.

See the following topics for more information:

- See ["Supporting Split Billing on Orders"](#) and ["Supporting Split Billing"](#) for more information about split billing and how the integration creates **/billinfo** hierarchies in BRM.
- See the discussion of hierarchical bill units in *BRM Managing Accounts Receivable* for more information about **/billinfo** hierarchies and how they relate to account hierarchies in BRM.

## About Corporate Account Hierarchies

A corporate account hierarchy represents the relationship in Siebel CRM between a corporation and its employees.

You use different classes of accounts to define a corporate account hierarchy that the integration synchronizes as hierarchical account groups in BRM. You can use the hierarchical account groups to produce invoices and reports that include detailed billing information at the different levels of the hierarchy.

The integration synchronizes most account hierarchies as parent-child relationships by using the accounts on order lines. If you intend to create corporate account hierarchies, you can specify the type of corporate account for which to synchronize the entire hierarchy in the **O2C.CorporateHierarchyAccountType** property. Hierarchical relationships for accounts with an account type that matches this value are synchronized as described in "[Synchronizing Corporate Account Hierarchies to BRM](#)". Hierarchical relationship for accounts with any other account type are synchronized as described in "[About Account and Billing Hierarchies](#)".

## Enabling and Disabling Corporate Account Hierarchy Synchronization

To enable or disable the synchronization of corporate account hierarchies:

1. Open the **AIAConfigurationProperties.xml** file.
2. Search for the **O2C.CorporateHierarchyAccountType** property.
3. Do one of the following:
  - To enable synchronization for corporate account hierarchies, set the value of the property to the account type for which you want to synchronize account hierarchies.
  - To disable synchronization for corporate account hierarchies, remove any value for the property, leaving it blank.

By default, the value of the property is **BUSINESS**.

4. Save and close the file.

See "[Configuring the Process Integration for Customer Management](#)" for more information about **O2C.CorporateHierarchyAccountType** and working with **AIAConfigurationProperties.xml**.

## Creating Corporate Account Hierarchies

To create corporate account hierarchies in Siebel CRM, associate accounts with parent accounts as described in the discussion of creating an account hierarchy in *Siebel Communications Guide*.

Use the same account type for all of the accounts in a corporate account hierarchy, and ensure that this account type is set as the value for the **O2C.CorporateHierarchyAccountType** property. Setting this property enables synchronization for corporate account hierarchies. See "[Enabling and Disabling Corporate Account Hierarchy Synchronization](#)" for more information.

Use the following account classes for the different levels of the hierarchy:

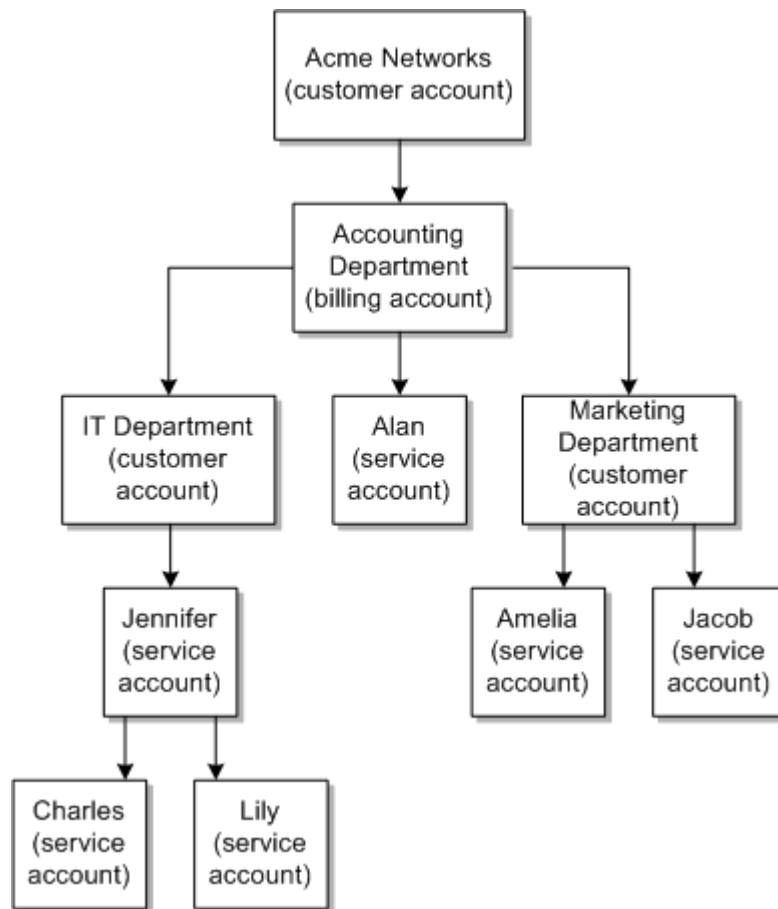
- **Customer:** A department or corporation-level account that represents a tier in the hierarchy, including an account at the top of the hierarchy that represents the entire organization. Maintain the distinction between corporate customers and consumer by using different customer accounts in the different scenarios.

- **Billing:** A department-level account that represents the person or department who pays for the services, in the middle of the hierarchy.
- **Service:** An employee-level account, represents the person who uses the service, below the billing account in the hierarchy.

Although only the billing account actually pays for the services, the other accounts in the hierarchy are also synchronized as self-paying accounts.

For example, Acme Networks is a small software corporation with several departments. The departments have managers and employees under them. Figure 18-3 shows a simple corporate account hierarchy for three departments of Acme Networks.

**Figure 18-3 Example Corporate Account Hierarchy**



## Synchronizing Corporate Account Hierarchies to BRM

The integration synchronizes corporate account hierarchies while creating accounts in BRM through the Create/Sync Customer Account integration flow, or while updating accounts in BRM through the Update Customer Account integration flow.

## Synchronizing Hierarchies for New Corporate Accounts

To synchronize account hierarchies for new corporate accounts, submit an order from the customer account for the service accounts in the account hierarchy. The integration

synchronizes the account hierarchy of only the accounts included on the order, not those in the entire account hierarchy.

For example, assume you create the corporate account hierarchy in Siebel CRM or the OSM account repository shown in [Figure 18-3](#). You then submit an order for Amelia's account from the corporate account, as shown in [Table 18-2](#).

**Table 18-2 Example Order Line with Corporate Account Hierarchy**

| Service | Customer Account | Billing Account       | Service Account |
|---------|------------------|-----------------------|-----------------|
| Mobile  | Acme Networks    | Accounting Department | Amelia          |

The integration retrieves the linear hierarchy for Amelia's account and creates hierarchical account groups for each account in BRM. Acme Networks' group includes the accounting department as a child, the accounting department's group includes the marketing department as a child, and the marketing department's group includes Amelia as a child. Likewise, each group includes an association with its parent account. Amelia's group includes the marketing department as a parent, the marketing departments group includes the accounting department as a parent, and the accounting department's group includes Acme Networks as a parent.

The integration does not create hierarchical account groups for the other accounts from the account hierarchy until you submit orders for those accounts.

You can create an account hierarchy using a combination of new accounts and existing accounts but you must always submit an order to synchronize any new accounts.

For example, using the corporate account hierarchy in [Figure 18-3](#) again, after submitting the order in [Table 18-2](#), you create a service account for Jacob and set the parent on Jacob's account to the marketing department. To synchronize this addition to the hierarchy, you must submit an order that includes Jacob's account. When you submit the order, the integration creates a hierarchical account group for Jacob with the marketing department as a parent. It also adds Jacob as a child in the marketing department's hierarchical group.

## Synchronizing Hierarchies for Existing Corporate Accounts

You synchronize account hierarchies for existing corporate accounts by setting the parent on the accounts in Siebel CRM.

The integration synchronizes the immediate parent and child in the Siebel CRM account hierarchy of only the accounts that you update, not those in the entire hierarchy.

For example, assume you have already created accounts and submitted orders for Acme Networks, the accounting department, the IT department, and Jennifer, but no hierarchical relationship exists between them. To create the corporate account hierarchy shown in [Figure 18-3](#):

1. Set Jennifer's parent to the IT department account.

Siebel CRM initiates the Update Customer Account integration flow and the integration creates hierarchical account groups for Jennifer and the IT department.

2. Set the IT department's parent to the accounting department.

Siebel CRM initiates the Update Customer Account integration flow and the integration creates a hierarchical account group for the accounting department, and updates the hierarchical account group for the IT department.

3. Set the accounting department's parent to Acme Networks.



Siebel CRM initiates the Update Customer Account integration flow and the integration creates a hierarchical account group for Acme Networks, and updates the hierarchical account group for the accounting department.

For more information about hierarchical account groups in BRM, see *BRM Managing Accounts Receivable*.

## Updating Corporate Account Hierarchies

You can update existing corporate account hierarchies by setting or changing the parents on the accounts. Updates to corporate account hierarchies include:

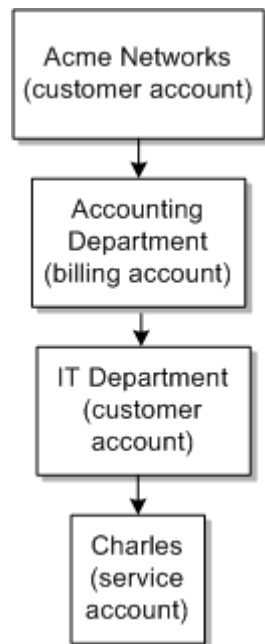
- Adding an account to the hierarchy. If the account is a new account, submit an order that includes the new account.
- Removing an account from the hierarchy.
- Moving an account to a different hierarchy.

## Adding Accounts to Existing Corporate Account Hierarchies

To add an account to an existing corporate account hierarchy, set the parent of the account you are adding to the account immediately above it in the hierarchy, and change the parent of any accounts immediately below it in the hierarchy.

For example, the integration has already synchronized the account hierarchy shown in [Figure 18-4](#) to BRM, and Acme Networks wants to add Charles' manager Jennifer and his coworker Lily to the hierarchy. Jennifer's account has already been synchronized to BRM, but Lily's account is new.

**Figure 18-4 Example Corporate Account Hierarchy Updates: Base Hierarchy**



To add Jennifer and Lily's accounts to the hierarchy:

1. Set Jennifer's parent to the IT department account.

The integration creates a hierarchical account group for Jennifer and updates the hierarchical account group for the IT department using the Update Customer Account integration flow.

2. Set Charles' parent to Jennifer's account.

The integration updates the hierarchical account groups for Jennifer, Charles, and the IT department using the Update Customer Account integration flow.

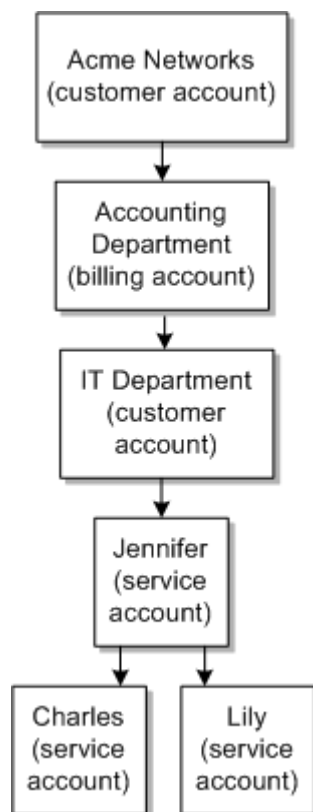
3. Set Lily's parent to Jennifer's account.

4. Submit an order from Acme Networks for Jennifer with the accounting department as the billing account.

The integration creates a hierarchical account group for Lily using the Create/Sync Customer Account integration flow and updates the hierarchical account group for Jennifer using the Update Customer Account integration flow.

Figure 18-5 shows the new account hierarchy.

**Figure 18-5 Example Corporate Account Hierarchy: Added Accounts**



## Removing Accounts from Corporate Account Hierarchies

To remove an account from a corporate account hierarchy, remove the value of the parent of the account, leaving it blank. If you want to keep any accounts under the account you are removing in the hierarchy, change the parent on those accounts to the parent account one level above the account you are removing.

For example, Jennifer leaves Acme Networks. To remove Jennifer from the hierarchy shown in [Figure 18-5](#), while keeping Charles and Lily in the hierarchy, set the parents on the accounts as follows:

1. Set Charles and Lily's parent to the IT department account.
2. Set Jennifer's parent to no parent, removing the IT department account.

The integration updates the hierarchical account groups for Charles, Lily, Jennifer, and the IT department using the Update Customer Account integration flow.

If you simply remove Jennifer's parent, Charles and Lily remain under Jennifer and are no longer part of the corporate hierarchy under Acme Networks.

## Moving Accounts to Different Corporate Account Hierarchies

To move an account to a different corporate account hierarchy, change the parent on the account you want to move.

For example, beginning with the account hierarchy in [Figure 18-5](#), Jennifer, Charles, and Lily transfer to the marketing department, which is a sibling of the IT department in the corporate account hierarchy under Acme Networks. To move their accounts to the new hierarchy, set Jennifer's parent to the marketing department account. The integration updates the hierarchical account groups for Jennifer, the IT department, and the marketing department using the Update Customer Account integration flow. Because hierarchical account groups only track the immediate parent, and Charles and Lily remain under Jennifer, the integration does not need to update Charles and Lily's groups.

To move only Jennifer's account, set Jennifer's parent to the marketing department account and set Charles and Lily's parents to the IT department account.

If you move an account to a place in the hierarchy under a new billing account, the integration transfers the services to the new billing account and updates the billing hierarchy. It does not remove the original account from the billing hierarchy. In BRM, you must either remove it manually or suppress this relationship in reports.

## About Legal Hierarchies

A legal hierarchy represents the relationship between a child account and the parent account that is held legally responsible for the child account's bills. Use a legal hierarchy when customers who pay their own bills cannot legally be held responsible for the charges that they incur, such as in jurisdictions where you cannot assign a debt collection agency to collect from minors, or when an employee pays their own bills but the corporate account is legally responsible.

By using a legal hierarchy to assign a legal owner to the minor or employee account, you can take collections actions against the legal owner rather than the minor or employee.

You create account hierarchies in Siebel CRM to represent legal hierarchies. The integration synchronizes the legal hierarchies as collections sharing groups in BRM when you submit orders that include both accounts in the hierarchy.

For information about how Oracle AIA supports integrated collections, see *Oracle AIA Siebel CRM Integration Pack for BRM: Agent Assisted Billing Care Implementation Guide*.

## Enabling and Disabling Legal Hierarchy Synchronization

To enable or disable the synchronization of legal hierarchies:

1. Open the **AIAConfigurationProperties.xml** file.
2. Search for the **O2C.LegalGroup** property.
3. Set the value of the property to one of the following:
  - **TRUE**: Enables the synchronization of legal hierarchies.
  - **FALSE**: Disables the synchronization of legal hierarchies.

By default, the value of the property is **FALSE**.

4. Save and close the file.

## Creating and Synchronizing Legal Hierarchies

To create and synchronize a legal hierarchy:

1. In Siebel CRM, associate the minor's account with a parent account as described in the discussion of creating an account hierarchy in *Siebel Communications Guide*.  
Each account can have only one parent or legal entity.
2. Submit a new order or a change order for services for the child account that includes the child account as the billing account and the parent as the owner account on the order lines.

When the integration detects that the billing account and owner account on an order line are different, it creates a collections sharing group in BRM to represent the legal hierarchy. Because a child account can have only one parent account, all services for one child account on a single order or multiple orders must use the same owner account. Because the integration creates only one collections sharing group for each owner account, it adds the child account as a member of the existing collections sharing group on any subsequent orders for the same owner account.

Collections sharing groups consist of a parent **/billinfo** object and child **/billinfo** objects. The integration creates the collections sharing group with the **/billinfo** of the owner account's primary billing profile as the parent and the **/billinfo** of the billing profile on the order for the child or employee account as the child. During collections processing, BRM ignores the child **/billinfo** and only enters the parent **/billinfo** into a collections scenario.

## Example of Legal Hierarchies

[Table 18-3](#) shows an example order line that would create a collections sharing group. In the example, Helen is a minor who wants to pay for her own mobile service. Her mother, Lisa, is an existing customer with her primary billing profile set to LisaBP. Lisa agrees to be the legal owner of Helen's charges.

To create the hierarchy, you set the parent on Helen's account to Lisa's account and submit an order including the account details shown in [Table 18-3](#).

**Table 18-3 Example Order Line with Legal Hierarchy**

| Service  | Service Account | Billing Account | Billing Profile | Owner Account |
|----------|-----------------|-----------------|-----------------|---------------|
| Wireless | Helen           | Helen           | HelenBP         | Lisa          |

The integration creates Helen's account and the associated HelenBP billing profile. Because the integration detects that the billing account and owner account on the order line are different, it then creates a collections sharing group in BRM. The collections sharing group consists of the **/billinfo** created for LisaBP as the parent and the **/billinfo** created for HelenBP as a child.

You could also include a service for Lisa's son Alex on the same order, as shown in [Table 18-4](#).

**Table 18-4 Example Siebel CRM Order Line with Legal Hierarchy**

| Service  | Service Account | Billing Account | Billing Profile | Owner Account |
|----------|-----------------|-----------------|-----------------|---------------|
| Wireless | Helen           | Helen           | HelenBP         | Lisa          |
| Wireless | Alex            | Alex            | AlexBP          | Lisa          |

In this order, the integration would include the **/billinfo** created for both AlexBP and HelenBP as members of LisaBP's collections sharing group.

You must ensure that any subsequent services for Helen and Alex also specify Lisa as the owner account.

For more information about collections sharing groups, see *BRM Collections Manager*.

## Updating Legal Hierarchies

You can update existing legal hierarchies by adding members or changing the parents on the accounts and submitting change orders. Updates to account hierarchies include:

- [Adding New Members to Legal Hierarchies](#)
- [Changing the Legal Owner of a Child Account](#)
- [Removing the Legal Owner from a Child Account](#)

When you disconnect the services of an account in a legal hierarchy, the integration does not delete the collections sharing group or remove the account's **/billinfo** as a member. This behavior lets you preserve the collections sharing group and memberships if only some of the member's services are disconnected. You are responsible for purging disconnected or inactive accounts and collections sharing groups from BRM.

## Adding New Members to Legal Hierarchies

You can add new members to existing legal hierarchies using new accounts or existing accounts. In the order capture and account management applications, there is no difference between creating and synchronizing a hierarchy for the first time and adding a new member to an existing hierarchy. In both cases, you submit an order where the billing account and owner or customer account on the order lines are different.

After creating any new accounts on the order, the integration calls the `PCM_OP_COLLECTIONS_GROUP_ADD_MEMBER` opcode. This opcode adds the **/billinfo** for the billing profile on the order line as a member of the collections sharing group that belongs to the **/billinfo** of the primary billing profile on the owner or customer account.

## Changing the Legal Owner of a Child Account

To change the legal owner of a child account:

1. Change the parent on the child account in the Siebel CRM hierarchy.
2. Submit a change order for the services in the child account that includes the new parent as the owner account on the order lines and order header.

For example, after you have submitted the order shown in [Table 18-3](#), Helen's father, David, offers to be the legal owner of her charges. To make this change, you change the parent on

Helen's account to David's account and submit a change order with the account details shown in [Table 18-5](#).

**Table 18-5 Example Change Order Line with Legal Hierarchy**

| Service  | Service Account | Billing Account | Billing Profile | Owner Account |
|----------|-----------------|-----------------|-----------------|---------------|
| Wireless | Helen           | Helen           | HelenBP         | David         |

Siebel CRM includes both the old and new owner accounts on the order.

The integration does the following:

1. Removes the **/billinfo** for HelenBP from Lisa's collections sharing group.
2. Deletes Lisa's collections sharing group.

The integration deletes the collections sharing group only if Helen was the only member. If the collections sharing group has other members, the integration does not delete the group.

3. Creates a collections sharing group with David as the owner and the **/billinfo** for HelenBP as a member.

The integration creates a new collections sharing group only if David is not already the owner of a collections sharing group. If David is already the owner of a collections sharing group, the **/billinfo** for HelenBP is added as a member of the existing group.

## Removing the Legal Owner from a Child Account

To remove the legal owner of a child account, submit a change order for the services in the child account that includes the child account as the owner account.

You are not required to change the account hierarchy in Siebel CRM.

For example, Helen reaches the age of majority and can be held legally responsible for her bills. To mark this change in BRM, you submit a change order based on the order in [Table 18-5](#) with the account details shown in [Table 18-6](#).

**Table 18-6 Example Change Order Line Removing Legal Hierarchy**

| Service  | Service Account | Billing Account | Billing Profile | Owner Account |
|----------|-----------------|-----------------|-----------------|---------------|
| Wireless | Helen           | Helen           | HelenBP         | Helen         |

The integration does one of the following:

- If Helen is the only member of David's collections sharing group, the integration deletes the entire group.
- If Helen is not the only member of David's collections sharing group, the integration removes her from the group.

# Implementing the Synchronize Customer Account Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Synchronize Customer Account business flow.

## About the Synchronize Customer Account Business Flow

The following Pre-Built Integration options enable the Synchronize Customer Account business flow:

- Order to Cash for Oracle Communications Order and Service Management (OSM)
- Order to Cash for Oracle Communications Billing and Revenue Management (BRM)
- Order to Cash for Siebel CRM

The Synchronize Customer Account business flow supports the following integration flows:

- Create/Sync Customer Account
- Update Customer Account

## About the Create/Sync Customer Account Integration Flow

The Create/Sync Customer Account integration flow synchronizes customer information from Siebel CRM to BRM.

This integration flow uses the following services:

- CommunicationsCustomerPartyEBSV2Resequencer
- QueryCustomerPartyListSiebelProvABCSImplV2
- ProcessAccountHierarchyListSiebelCommsProvABCSImpl
- SyncCustomerPartyListBRMCommsProvABCSImpl
- ProcessCollectionSharingBRMCommsProvABCSImpl

This flow is called during the Interfacing Orders to Create Customer Data integration flow of the Synchronize Fulfillment Order Billing Account business flow.

See "[About the Synchronize Fulfillment Order Billing Account Business Flow](#)" for information about the sequence of events for this business flow.

## About the Update Customer Account Integration Flow

This flow synchronizes account updates from Siebel CRM to BRM.

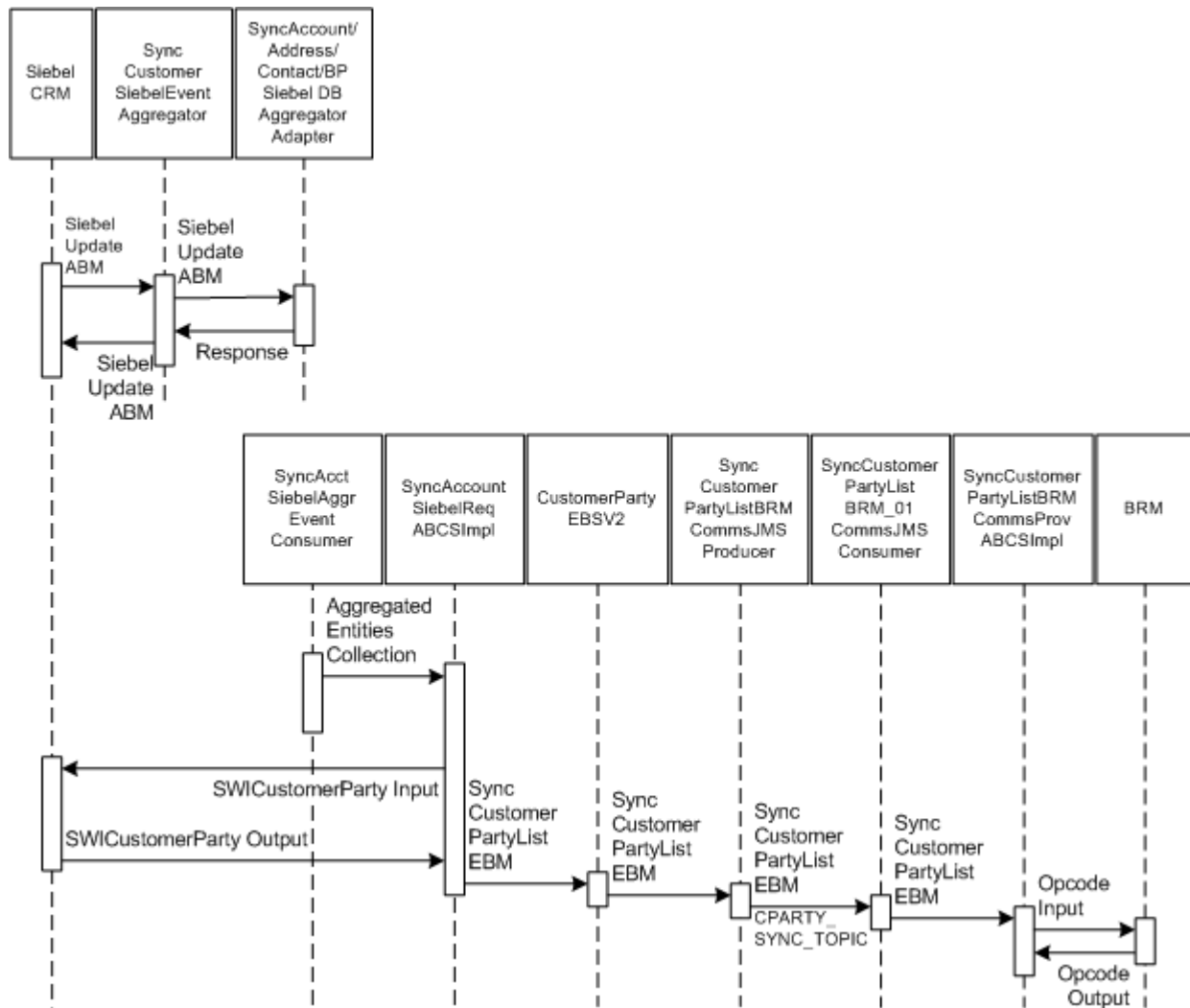
The Update Customer Account integration flow uses the following services:

- SyncCustomerSiebelEventAggregator

- SyncAcctSiebelEventAggrConsumer
- ProcessAccountHierarchyListSiebelCommsProvABCSImpl
- SyncAccountSiebelReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListBRM\_01CommsJMSConsumer
- SyncCustomerPartyListBRMCommsProvABCSImpl

Figure 19-1 illustrates the Update Customer Account integration flow.

Figure 19-1 Update Customer Account Sequence



The integration updates customer accounts in BRM as follows:

1. A Siebel CRM user updates an attribute on a customer's account such as account name, address, contact information, or billing profile.
2. Siebel CRM sends the SiebelUpdateABM message containing the updated attribute to the SyncCustomerSiebelEventAggregator service. SiebelUpdateABM takes one of the following forms, depending on the updated attribute:



- ListOfSWICustomerIO for name and hierarchy updates
  - ListOfSWIAddressIO for address updates
  - ListOfSWIContactIO for contact updates
  - ListOfSWIBillingProfileIO for billing profile updates
3. SyncCustomerSiebelEventAggregator send the SiebelUpdateABM to one of the following database adapters, depending on the updated attribute:
    - SyncAccountSiebelAggregatorAdapter for name and hierarchy updates
    - SyncAddressSiebelAggregatorAdapter for address updates
    - SyncContactSiebelAggregatorAdapter for contact updates
    - SyncBPSiebelAggregatorAdapter for billing profile updates
  4. The adapter extracts the account, contact, address, or billing profile ID from the ABM, stores it in the AIA\_AGGREGATED\_ENTITIES database table, and sends an acknowledgment to Siebel CRM.
 

The IDs in the database table are stored in the same hierarchy as in Siebel CRM (for example, BillingProfileID is always the child of an account ID).
  5. The SyncAcctSiebelEventAggrConsumer service picks up the account ID and all of its child IDs. This service sends updates for the same customer in sequence.
  6. SyncAcctSiebelEventAggrConsumer sends the account ID and child IDs to the SyncAccountSiebelReqABCServiceImpl service.
  7. SyncAccountSiebelReqABCServiceImpl sends the account and child IDs as an input application business message (ABM) for the Siebel CRM SWICustomerPartyQueryByExample web service operation.
  8. SWICustomerPartyQueryByExample sends an output ABM, which includes the entire account data, back to SyncAccountSiebelReqABCServiceImpl.
  9. SyncAccountSiebelReqABCServiceImpl transforms the output into the SyncCustomerPartyListEBM and sends it to the CustomerPartyEBSV2 service.
  10. CustomerPartyEBSV2 sends SyncCustomerPartyListEBM to SyncCustomerPartyListBRMCommsJMSProducer, which publishes it to the CPARTY\_SYNC\_TOPIC JMS topic.
  11. The consumer subscribed to this topic by default, SyncCustomerPartyListBRM\_01CommsJMSConsumer, picks up SyncCustomerPartyListEBM, checks which service it should be sent to, stamps it with the target ID, and sends it to SyncCustomerPartyListBRMCommsProvABCServiceImpl.
 

You can define more consumers for multiple instances of BRM or other billing systems and subscribe them to CPARTY\_SYN\_TOPIC.
  12. SyncCustomerPartyListBRMCommsProvABCServiceImpl calls one of the following opcodes to update the data in BRM:
    - PCM\_OP\_CUST\_UPDATE\_CUSTOMER
    - PCM\_OP\_CUST\_DELETE\_PAYINFO
    - PCM\_OP\_CUST\_SET\_STATUS
    - PCM\_OP\_BILL\_GROUP\_MOVE\_MEMBER
  13. The opcode sends confirmation to SyncCustomerPartyListBRMCommsProvABCServiceImpl.

## BRM Interfaces

Table 19-1 lists the BRM interfaces used by the Synchronize Customer Account business flow.

**Table 19-1 BRM Interfaces**

| Opcode                                 | Purpose  |
|--|--|
| PCM_OP_CUST_COMMIT_CUSTOMER            | Creates a new account with one or more <b>/billinfo</b> and <b>/payinfo</b> objects.   |
| PCM_OP_CUST_UPDATE_CUSTOMER            | Updates account information (name, address, phone), contact information, and billing information.  |
| PCM_OP_CUST_DELETE_PAYINFO             | Delete a <b>/payinfo</b> from an account.  |
| PCM_OP_CUSTCARE_MOVE_ACCT              | Moves a child account to a new parent account.   |
| PCM_OP_CUST_SET_STATUS                 | Modifies the account status.   |
| PCM_OP_BILL_GROUP_MOVE_MEMBER          | Modifies a hierarchical account group (moves an account between groups, creates a group if one doesn't exist, deletes a group if it is empty). |
| PCM_OP_COLLECTIONS_GROUP_CREATE        | Creates a collections sharing group with the owner account as the owner and the billing account as a member.                                   |
| PCM_OP_COLLECTIONS_GROUP_ADD_MEMBER    | Adds the billing account as a member of the owner account's collections sharing group.   |
| PCM_OP_COLLECTIONS_GROUP_DELETE_MEMBER | Removes the billing account as a member of the owner account's collections sharing group.  |
| PCM_OP_COLLECTIONS_GROUP_GET_BILLINFO  | Checks whether an account already owns or is a member of a collections sharing group.  |
| PCM_OP_COLLECTIONS_GROUP_DELETE        | Deletes the owner account's collections sharing group.   |

See *BRM Developer's Reference* for detailed descriptions of individual opcodes.

## 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-<br>(SWICustomerParty)                 | Retrieves account, bill profile, contact, and address data from Siebel CRM. | <ul style="list-style-type: none"> <li>QueryCustomerPartyListSiebelProvABCImplV2 as part of creating a new account or adding a new billing profile to an existing account.</li> <li>SyncAccountSiebelReqABCImpl</li> </ul> |
| Query account hierarchy -<br>(SWI Account Hierarchy) | Retrieves the full linear account hierarchy from Siebel CRM.                | ProcessAccountHierarchyListSiebelCommProvABCImpl   |

See *Siebel CRM Web Services Reference* for more information about this web service.

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.agg regateaccountevent 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.agg regatebpevent 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.agg regatecontactevent 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.agg regateadressevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ENTITIES).   |

See the discussion of 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

**`Oracle_home/comms_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/`**

- Enterprise business service (EBS) WSDL files:

**`Oracle_home/comms_home/source/soainfra/apps/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 such as adding new data elements. These extensions are protected and will remain intact even after a patch or an upgrade, so long as the extensibility guidelines are followed.

See the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

The following services are delivered with the Synchronize Customer Account business flow:

- [CommsProcessBillingAccountListEBF](#)
- [CommunicationsCustomerPartyEBSV2Resequencer](#)
- [SyncCustomerPartyListBRMCommsProvABCSImpl](#)
- [SyncCustomerPartyListBRMCommsJMSProducer](#)
- [SyncCustomerPartyListBRM\\_01CommsJMSConsumer](#)
- [CustomerPartyEBSV2](#)
- [ProcessCollectionSharingBRMCommsProvABCSImpl](#)
- [ProcessAccountHierarchyListSiebelCommsProvABCSImpl](#)
- [QueryCustomerPartyListSiebelProvABCSImplV2](#)
- [SyncCustomerSiebelEventAggregator](#)
- [SyncAccountSiebelAggregatorAdapter](#)
- [SyncContactSiebelAggregatorAdapter](#)
- [SyncAddressSiebelAggregatorAdapter](#)
- [SyncBPSiebelAggregatorAdapter](#)
- [SyncAcctSiebelAggrEventConsumer](#)
- [SyncAccountSiebelReqABCSImpl](#)

Additional services are delivered with the Synchronize Customer Account business flow as part of the Interfacing Orders to Create Customer Data integration flow, which is part of the Synchronize Fulfillment Order Billing Account business flow. See "[Integration Services](#)" in "[Implementing the Synchronize Fulfillment Order Billing Account Business Flow](#)" for more information about these services.

### CommsProcessBillingAccountListEBF

This service invokes Siebel CRM services to query account information and BRM services to create accounts.

This service does the following:

1. Receives `ProcessBillingAccountListEBM`, which contains a list of customer account IDs, billing profile IDs, and the target system ID, and transforms it into `QueryCustomerPartyListEBM`.
2. Sends `QueryCustomerPartyListEBM` to `QueryCustomerPartyListSiebelProvABCSImplV2`.
3. Receives `QueryCustomerPartyListResponseEBM`, which contains detailed account information, and transforms it into `SyncCustomerPartyListEBM`.
4. Sends `SyncCustomerPartyListEBM` to `CommunicationsCustomerPartyEBSV2Resequencer`.
5. Receives `SyncCustomerPartyListResponseEBM` from `SyncCustomerPartyListBRMCommsProvABCSImpl`.
6. If the owner account in `ProcessBillingAccountListEBM` is different than the billing account, this service does the following:
  - a. Sends `ProcessBillingAccountListEBM` to `ProcessCollectionsSharingBRMCommsProvABCSImpl`.
  - b. Receives `ProcessBillingAccountListResponseEBM`.
7. Combines the response messages into `ProcessBillingAccountListResponseEBM` and sends the message back to `CommsProcessFulfillmentOrderBillingAccountListEBF`.

## CommunicationsCustomerPartyEBSV2Resequencer

This service groups account messages from `CommsProcessBillingAccountListEBF` by account ID. This process receives `SyncCustomerPartyListEBM` and sends it to `SyncCustomerPartyListBRMCommsProvABCSImpl`.

See "[Using the Oracle Mediator Resequencer Feature](#)" for more information about Oracle Mediator Resequencer.

## SyncCustomerPartyListBRMCommsProvABCSImpl

This service creates or updates customer accounts in BRM. It does following:

1. Receives the `SyncCustomerPartyListEBM`.
2. For child accounts, checks whether the parent account has been synchronized. The integration can only synchronize a child account after the parent account has been synchronized.
3. For each account in the EBM with the ADD action code, the service creates the account as follows:
  - a. Transforms the `SyncCustomerPartyListEBM` to the BRM-specific account creation ABM.
  - b. Calls `PCM_OP_CUST_COMMIT_CUSTOMER` to create the customer account.
  - c. For child accounts where the billing account on the order is different from the owner account and the **O2C.LegalGroup** configuration property is set to **TRUE**, calls any of the following opcodes:
    - `PCM_OP_COLLECTIONS_GROUP_GET_BILLINFO` to check if the owner account already owns a collections sharing group
    - `PCM_OP_COLLECTIONS_GROUP_CREATE` to create a collections sharing group if the owner account doesn't own one

- PCM\_OP\_COLLECTIONS\_GROUP\_ADD\_MEMBER to add the billing account to the owner account's collections sharing group
- d. Transforms the response from the opcode calls to SyncCustomerPartyListResponseEBM and populates the following cross-reference tables with the BRM IDs obtained:
    - CUSTOMERPARTY\_ACCOUNTID
    - CUSTOMERPARTY\_ADDRESSID
    - CUSTOMERPARTY\_CONTACTID
    - CUSTOMERPARTY\_BILLPROFILEID
    - CUSTOMERPARTY\_PAYPROFILEID
4. For each account in the EBM with the UPDATE action code, the service updates the account as follows:
    - a. For child accounts where the parent in the EBM is different from the parent obtained in the opcode response, moves the child account to the parent in the EBM by calling the PCM\_OP\_CUSTCARE\_MOVE\_ACCT.
    - b. Transforms the SyncCustomerPartyListEBM into a BRM-specific account update ABM.
    - c. Calls PCM\_OP\_CUST\_UPDATE\_CUSTOMER to update the account.
    - d. For child accounts where the billing account on the order is different from the customer account and the **O2C.LegalGroup** configuration property is set to **TRUE**, calls any of the following opcodes:
      - PCM\_OP\_COLLECTIONS\_GROUP\_GET\_BILLINFO to check if the owner account already owns a collections sharing group or the billing account is already a member of a collections sharing group
      - PCM\_OP\_COLLECTIONS\_GROUP\_CREATE to create a collections sharing group if the owner account doesn't own one
      - PCM\_OP\_COLLECTIONS\_GROUP\_ADD\_MEMBER to add the billing account to the owner account's collections sharing group
      - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE\_MEMBER to remove the billing account from a collections sharing group if the owner account on the EBM and the owner account in the PCM\_OP\_COLLECTIONS\_GROUP\_GET\_BILLINFO response are different
      - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE to delete the collections sharing group if the billing account being moved was the only member
    - e. Checks whether the AIA configuration 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.
    - f. 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.
    - g. Transforms the SyncCustomerPartyListEBM to SyncCustomerPartyListResponseEBM.
  5. Sends SyncCustomerPartyListResponseEBM back to the calling service.

## SyncCustomerPartyListBRMCommsJMSProducer

This producer service receives the SyncCustomerPartyListEBM message with account update information from CustomerPartyEBSV2. This service drops the message into the C\_PARTY\_SYNC\_TOPIC queue, where SyncCustomerPartyListBRM\_01CommsJMSProducer picks it up.

## SyncCustomerPartyListBRM\_01CommsJMSProducer

This is a mediator process subscribed to the CPARTY\_SYNC\_TOPIC queue. It does the following:

1. Picks up SyncCustomerPartyListEBM from CPARTY\_SYNC\_TOPIC.
2. Checks the cross-reference table to identify the BRM ID that corresponds to the common ID.
3. Stamps the EBM with the target BRM ID.
4. Sends the EBM to the SyncCustomerPartyListBRMCommsProvABCImpl service.

This process is intended for a topology with multiple BRM instances. Deploy a version of this consumer for each BRM instance. See "[Configuring Multiple BRM Instances for Communications Integrations](#)" for more information about multiple BRM instances.

## CustomerPartyEBSV2

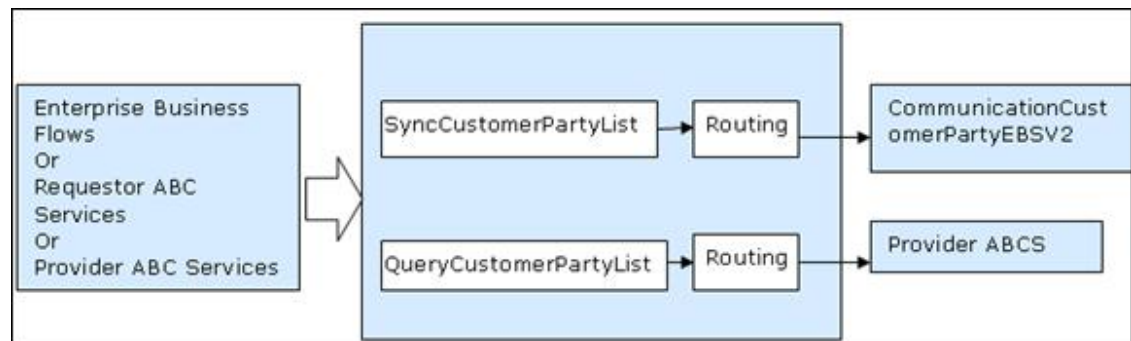
CustomerPartyEBSV2 a lightweight EBS routing service that exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object.

CustomerPartyEBSV2 service uses the following operations:

- SyncCustomerPartyList
- QueryCustomerPartyList

Figure 19-2 illustrates the relationship of CustomerPartyEBSV2 with the other services in the integration flow.

**Figure 19-2 CustomerPartyEBSV2**



The process checks whether the incoming message has a target system identifier. If the target system identifier is not present, then the routing rule assumes multiple BRM systems and routes the incoming requests to SyncCustomerPartyListBRMCommsJMSProducer.

If you have only one BRM instance, you can change the routing rule to route incoming requests to the SyncCustomerPartyListBRMCommsProvABCSEImpl directly. You must also apply a transformation before routing to stamp the target system identifier in the EBM. Apply the transformation in the following file:

**esb:///ESB\_Projects/Customer\_CustomerPartyEBSV2/AddTargetID\_BRM01.xsl**

## ProcessCollectionSharingBRMCommsProvABCSEImpl

This service invokes opcodes in BRM to create collections sharing groups when the owner account on an order line is different from the billing account. This service does the following:

1. Receives ProcessBillingAccountListEBM.
2. Transforms ProcessBillingAccountListEBM into the correct input and invokes any of the following opcodes:
  - PCM\_OP\_COLLECTIONS\_GROUP\_GET\_BILLINFO to check if the owner account already owns a collections sharing group or the billing account is already a member of a collections sharing group
  - PCM\_OP\_COLLECTIONS\_GROUP\_CREATE to create a collections sharing group if the owner account doesn't own one
  - PCM\_OP\_COLLECTIONS\_GROUP\_ADD\_MEMBER to add the billing account to the owner account's collections sharing group
  - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE\_MEMBER to remove the billing account from a collections sharing group if the owner account on the EBM and the owner account in the PCM\_OP\_COLLECTIONS\_GROUP\_GET\_BILLINFO response are different
  - PCM\_OP\_COLLECTIONS\_GROUP\_DELETE to delete the collections sharing group if the billing account being moved was the only member
3. Receives the collections sharing group opcode response and transforms it into ProcessBillingAccountListResponseEBM.
4. Sends ProcessBillingAccountListResponseEBM to CommsProcessBillingAccountListEBF.

## ProcessAccountHierarchyListSiebelCommsProvABCSEImpl

This service calls the Siebel CRM web service to query the account hierarchy for a given account ID.

It does the following:

1. Receives ProcessBillingAccountListEBM from CommsProcessFulfillmentOrderBillingAccountListEBF.
2. Transforms ProcessBillingAccountListEBM into a Siebel CRM ABM.
3. Calls the Siebel CRM web services to query the complete linear account hierarchy for the account IDs listed in the ABM.
4. Receives the response from the Siebel CRM web service.
5. Transforms the response into ProcessBillingAccountListResponseEBM.
6. Sends ProcessBillingAccountListResponseEBM to CommsProcessFulfillmentOrderBillingAccountListEBF.



## QueryCustomerPartyListSiebelProvABCImplV2

This service does the following:

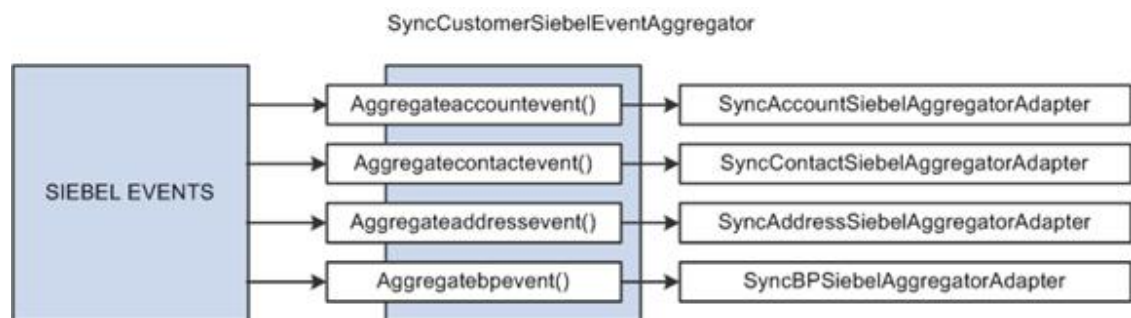
1. Receives QueryCustomerPartyListEBM from CommsProcessBillingAccountListEBF.
2. Transforms QueryCustomerPartyListEBM into a Siebel CRM ABM.
3. Invokes the Siebel CRM SWICustomerParty web service.
4. Receives the response from SWICustomerParty.
5. Transforms the response into QueryCustomerPartyListResponseEBM and sends it to CommsProcessBillingAccountListEBF.

## SyncCustomerSiebelEventAggregator

This service receives Siebel CRM update account events and collates them into the AIA\_AGGREGATED\_ENTITIES database table.

Figure 19-3 illustrates the relationship of the SyncCustomerSiebelEventAggregator with the other services in the integration flow.

**Figure 19-3 SyncCustomerSiebelEventAggregator**



This service provides the following operations:

- **Aggregateaccountevent:**
  1. Receives the Account Updated message from Siebel CRM.
  2. Extracts the account ID, contact IDs, and address IDs from the message.
  3. Invokes the SyncAccountSiebelAggregatorAdapter to store these IDs into the AIA\_AGGREGATED\_ENTITIES database table.
- **Aggregatecontactevent:**
  1. Receives the Contact Update message from Siebel CRM.
  2. Extracts the account IDs, billing profile IDs, and contact IDs from the message.
  3. Invokes the SyncContactSiebelAggregatorAdapter to store these IDs in the AIA\_AGGREGATED\_ENTITIES database table.
- **Aggregateaddressevent:**
  1. Receives the Address Update message from Siebel CRM.

2. Extracts the account IDs, billing profile IDs, and address IDs from the message.
  3. Invokes the SyncAddressSiebelAggregatorAdapter to store these IDs into the AIA\_AGGREGATED\_ENTITIES database table.
- Aggregatebpevent:
    1. Receives the BillingProfile Updated message from Siebel CRM.
    2. Extracts the billing profile ID and the associated account ID from the message.
    3. Invokes the SyncBPSiebelAggregatorAdapter to store these IDs in the AIA\_AGGREGATED\_ENTITIES database table.

## SyncAccountSiebelAggregatorAdapter

This service invokes the AIA\_AGGREGATOR\_PUB.SIEBEL\_AGGREGATE\_ACCOUNT PL/SQL procedure, which aggregates the account events generated in Siebel CRM when an account is created or updated.

## SyncContactSiebelAggregatorAdapter

This service invokes the AIA\_AGGREGATOR\_PUB.SIEBEL\_AGGREGATE\_CONTACT PL/SQL procedure, which aggregates the account events generated in Siebel CRM when an account is created or updated.

## SyncAddressSiebelAggregatorAdapter

This service invokes the AIA\_AGGREGATOR\_PUB.SIEBEL\_AGGREGATE\_ADDRESS PL/SQL procedure, which aggregates the account events generated in Siebel CRM when an account is created or updated.

## SyncBPSiebelAggregatorAdapter

This service invokes the AIA\_AGGREGATOR\_PUB.SIEBEL\_AGGREGATE\_BP PL/SQL procedure, which aggregates the account events generated in Siebel CRM when an account is created or updated.

## SyncAcctSiebelAggrEventConsumer

This service extracts the account IDs stored in the AIA\_AGGREGATED\_ENTITIES database table and sends them to the SyncAccountSiebelReqABCSImpl service.

Sequencing is enabled for this service. When this consumer calls the requester service for further processing and the requester service fails, any subsequent update for that customer is not processed until action is taken on the messages in the sequencer.

If the failure is due to a business error, you must remove the messages from the queue for the subsequent messages to process. If the failure is system related, you can retry messages in the resequencer, which moves the messages from the resequencer queue and enables 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 transforms the Siebel CRM aggregated message into the SyncCustomerPartyListEBM message and sends it to the CustomerPartyEBSV2 service.

# 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 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

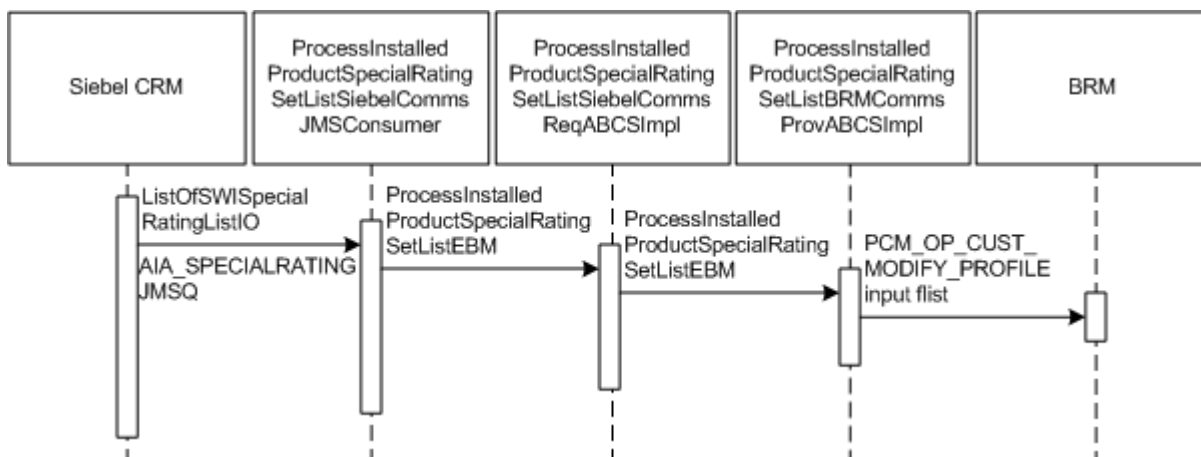
### Synchronize Friends and Family List Updates to BRM Integration Flow

This integration flow uses the following services:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

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**



The integration synchronizes friends and family list updates to BRM in a one-way asynchronous pattern as follows:

1. A Siebel CRM user updates a special rating list. Siebel CRM drops the ListOfSWISpecialRatingListIO message into the AIA\_SPECIALRATINGJMSQ Oracle Advanced Queuing (AQ) queue.
2. The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer service picks up ListOfSWISpecialRatingListIO and sends it to the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl service.
3. ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl transforms ListOfSWISpecialRatingListIO into the ProcessInstalledProductSpecialRatingSetListEBM message and sends it to the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl service.
4. ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl 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
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl

**Table 20-1 Transaction Boundaries and Recovery Details**

| Transaction   | DB Operations                 | In Case of Error   | Recovery                                    |
|---|-------------------------------|--|---|
| ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer picks up message and routes it to ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl, which transforms message and routes to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl. 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/*

- Enterprise business service (EBS) WSDL files:

*Oracle\_home/comms\_home/source/soainfra/apps/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:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

## ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer

The ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer 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 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl service.

## ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl

The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl 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 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl 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.

## ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl

The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl 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.

# 21

## Understanding the Process Integration for Order Fallout Management

This chapter describes the process integration for order fallout management and related order fallout concepts.

### About the Process Integration for Order Fallout Management

The process integration for Order Fallout Management helps you detect order fallout and notify the appropriate person so that you can correct failed orders. Oracle Application Integration Architecture (Oracle AIA) handles order exceptions with a detection and notification process that uses trouble ticketing for notification and tracking.

You can use Oracle AIA or Oracle Communications Order and Service Management (OSM) to initiate the creation of trouble tickets in Siebel CRM.

### About the Order Fallout Management Business Flows

The process integration for Order Fallout Management includes the following business flows:

- **Create Trouble Ticket from Oracle AIA:**
  - Used when the request to create a trouble ticket is initiated by Oracle AIA for messages with the JMS Correlation ID of AIA\_ORDERFALLOUT\_TTS.
  - Enabled by the Order to Cash for Siebel CRM pre-built integration options.See "[About the Create Trouble Ticket from Oracle AIA Business Flow](#)" for more information.
- **Create and Manage Trouble Ticket from OSM.**
  - Used when the request to create a trouble ticket is initiated by OSM for messages with the JMS Correlation ID of AIA\_ORDERFALLOUT\_CFS.
  - Enabled by the Order to Cash for Siebel CRM and Order to Cash for OSM pre-built integration options.See "[About the Create and Manage Trouble Ticket from OSM Business Flow](#)".

### About the Create Trouble Ticket from Oracle AIA Business Flow

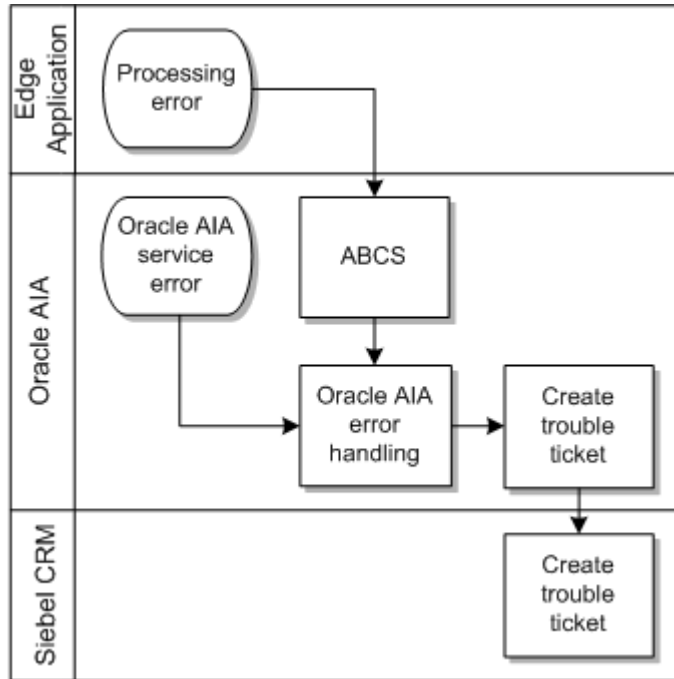
The Create Trouble Ticket from Oracle AIA business flow provides trouble ticketing when OSM is not the central fulfillment system and is not used for order fulfillment and fallout management. In this business flow, the integration provides services and artifacts to handle order fallout detection and notification.

The integration can also create trouble tickets in Siebel CRM when an order fails and an error is detected by the Oracle AIA error handler. Because the integration creates a trouble ticket for every fault message notification, you must model your processes to generate only one notification for each order failure.

[Figure 21-1](#) illustrates the Create Trouble Ticket from Oracle AIA business flow at a high level.



**Figure 21-1 Creating Trouble Tickets from Oracle AIA**



When an error occurs in an edge application, such as in BRM while synchronizing customers or initiating billing, the edge application sends an error notification back to the ABCS that provided the order data. The ABCS sends a fault message to the Oracle AIA error handling framework.

When an error occurs in an Oracle AIA service, such as an ABCS or EBS, the service sends a fault message to the Oracle AIA error handling framework.

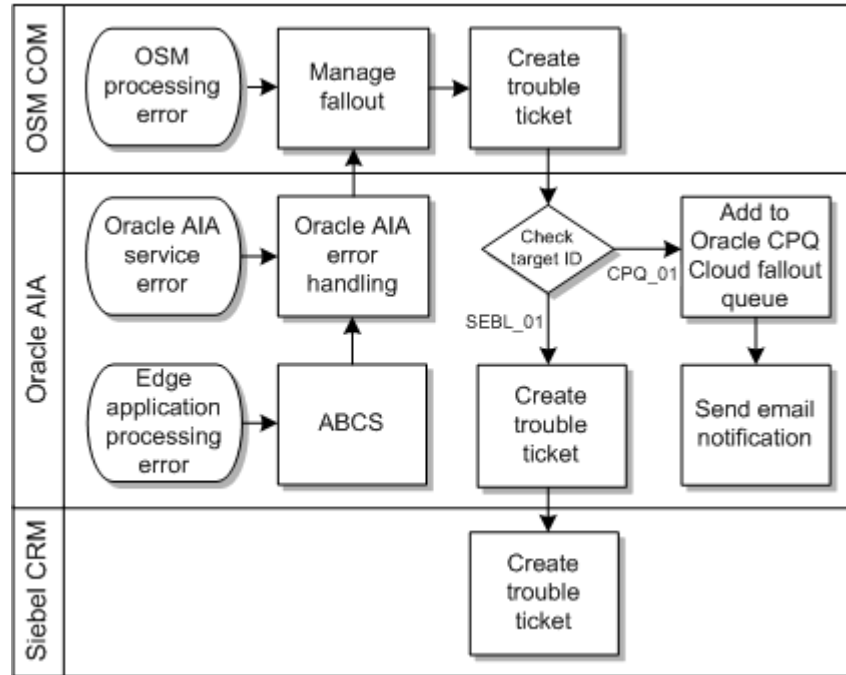
The Oracle AIA error handling framework initiates the Oracle AIA services that create trouble tickets in Siebel CRM.

## About the Create and Manage Trouble Ticket from OSM Business Flow

The Create and Manage Trouble Ticket from OSM business flow provides trouble ticketing when OSM is the central fulfillment system. When you install the Order to Cash for OSM pre-built integration option, the integration automatically uses OSM to manage order fallout.

Figure 21-2 illustrates the high-level process flow involved in using OSM for order fallout management.

**Figure 21-2 Creating Trouble Tickets from OSM**



When an error occurs in an edge application, such as in BRM while synchronizing customers or initiating billing, the edge application sends an error notification back to the ABCS that provided the order data. The ABCS sends a fault message to the Oracle AIA error handling framework, which sends the message to OSM in the central order management role (OSM COM).

When an error occurs in an Oracle AIA service, such as an ABCS or EBS, the service sends a fault message to the Oracle AIA error handling framework, which sends the message to OSM COM.

When OSM COM receives errors from Oracle AIA or processing errors from within OSM, OSM COM manages the fallout with OSM-internal fallout management mechanisms, including compensation and orchestration of fallout orders. As part of fallout management, OSM can create trouble tickets and send them to Siebel CRM through Oracle AIA.

The integration creates trouble tickets on a per-order or per-application basis for failed orders that have been submitted from Siebel CRM as follows:

- The failure of different orders in the same application generates different trouble tickets.
- The failure of the same order in a different application generates a different trouble ticket. For example, if an order fails while initiating billing in BRM, and the same order fails while provisioning a service in OSM, the integration creates two trouble tickets for that order: one for the failure in BRM and one for the failure in OSM.
- Multiple order line item failures for the same order in the same application generate only one trouble ticket with the additional order line item failure information appended. For example, if three line items on an order fail during the service design phase of provisioning in OSM, the integration creates only one trouble ticket containing all three order line failures.

This business flow operates under the following constraints:

- To cancel a failed order as part of error correction, you must cancel the order from OSM rather than Siebel CRM.
- You must ensure that any custom process flow that creates an order failure notification creates only one notification for each order failure.
- When an order revision from Siebel CRM 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. You must manually close the trouble ticket for the revision that failed upon arrival.

## About Order Fallout Management for Different Error Types

Oracle AIA manages order fallout for the following error types:

- System errors: Errors caused by infrastructure outages. See "[About Order Fallout Management for System Errors](#)".
- Business errors: Errors caused by missing data. See "[About Order Fallout Management for Business Errors](#)".
- Service: Errors caused by failed Oracle AIA services. See "[About Order Fallout Management for Service Errors](#)".

## About Order Fallout Management for System Errors

System errors occur when infrastructure outages cause the participating application, network, or Fusion Middleware engine to be unavailable. Because order fallout due to system errors occurs when there is nothing inherently wrong with the original order message, you can resubmit the same message for processing.

Unlike business errors, system errors do not result in trouble tickets. They are managed using the standard Oracle AIA error handling framework and service fault policies, which automatically resubmit orders and roll back transactions to the error queue.

Some Oracle AIA flows communicate with integrated applications in a synchronous pattern and include consumer services that can be retired when system errors occur. Retiring the consumer service prevents it from picking up subsequent messages and sending them to an ABCS that will attempt to connect to an unavailable application. After the system error is resolved, you reactivate the consumer service and it resumes picking up messages from the queue. See "[Reactivating Retired Consumer Services](#)" for information about how to reactivate consumer services.

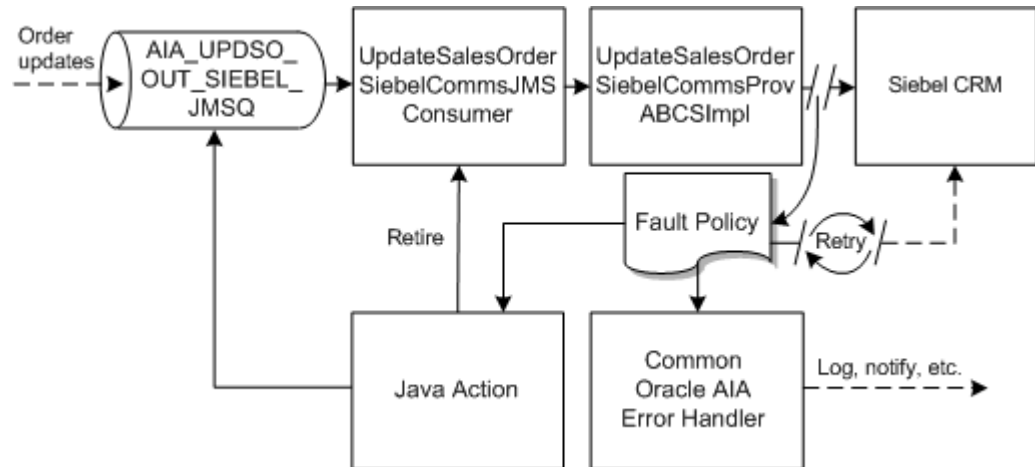
If you do not want Oracle AIA to handle system errors in this way, you can disable the Java action which retires the consumers. See "[Disabling Consumer Service Retirement](#)" for more information.

The following consumer services are enabled to handle system errors by default:

- UpdateSalesOrderSiebelCommsJMSConsumer
- SyncItemCompositionListSiebelCommsJMSConsumer
- ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer
- ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer

[Figure 21-3](#) shows an example of how errors are handled in a part of the flow for updating order status in Siebel CRM. The figure uses UpdateSalesOrderSiebelCommsJMSConsumer as an example, but the same pattern applies to the other consumers enabled to handle system errors.

**Figure 21-3 System Error Handling in Order Update Flow**



The figure illustrates the following process:

1. The UpdateSalesOrderSiebelCommsJMSConsumer consumer picks up a message from the AIA\_UPDSO\_OUT\_SIEBEL\_JMSQ queue and sends it to the UpdateSalesOrderSiebelCommsProvABCImpl provider service.
2. The provider service attempts to send the message to Siebel CRM, but Siebel CRM is unavailable.
3. The fault policy for the provider service tries to resend the message a configured number of times.  
Because Siebel CRM remains unavailable, the retries fail.
4. The service's fault policy throws the error to the Oracle AIA common error handler, which performs the basic activities of logging the fault, sending a notification, and dropping it into the Oracle AIA error topic.
5. Because it is a system fault, the service's fault policy initiates a Java action to retire the consumer.  
UpdateSalesOrderSiebelCommsJMSConsumer is retired and no longer picks up messages from AIA\_UPDSO\_OUT\_SIEBEL\_JMSQ.
6. The Java action also throws the fault to the error queue, which persists messages that have been rolled back.

## Reactivating Retired Consumer Services

After you have resolved the system error and the edge application is available, you must reactivate the retired consumer services.

To reactivate retired consumer services:

1. In a web browser, navigate to the Oracle Enterprise Manager Fusion Middleware Control console for your Oracle AIA domain at the following location:

**http://host.domain:port/lem**

where *host*, *domain*, and *port* are the host, domain name, and port number for your Oracle AIA deployment.

2. In the Target Navigation tree, expand the SOA folder, then expand the soa-infra component, and then click the SOA partition component for your Oracle AIA deployment. The SOA Partition home page appears.
3. Click the **Deployed Composites** tab.
4. Locate any of the following consumer services that have **Retired** in the Mode column:
  - UpdateSalesOrderSiebelCommsJMSSConsumer
  - SyncItemCompositionListSiebelCommsJMSSConsumer
  - ProductOptimizedSyncPriceListListSiebelCommsJMSSConsumer
  - ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSSConsumer
5. For each consumer service from the list above with **Retired** in the Mode column, do the following:
  - a. Click the consumer service's name. The SOA Composite home page appears.
  - b. Click **Activate**. The consumer service is reactivated and resumes picking up messages from the queue to which it is subscribed.

## Disabling Consumer Service Retirement

To disable the error handling mechanism that retires consumers:

1. Open the fault policy file for the consumer for which you want to disable the Java action that retires the service.
2. Search for the following line:
 

```
<retryFailureAction ref="aia-custom-java"/>
```
3. Change **aia-custom-java** to **aia-no-action**. For example:
 

```
<retryFailureAction ref="aia-no-action"/>
```
4. Save and close the file.
5. Upload the file to the Oracle Metadata Services repository. See *Oracle AIA Installation Guide* for more information about uploading changed files to the Oracle Metadata Services repository.

## About Order Fallout Management for Business Errors

Business failures are caused by business reasons unrelated to the Oracle AIA infrastructure. For example, a sales order that is missing critical data fails while being sent to Oracle Communications Billing and Revenue Management (BRM). Because the error occurs in the original message, you must revise the order in Siebel CRM to correct the error, then resubmit the order.

## About Order Fallout Management for Service Errors

If an error occurs during an Oracle AIA service call, such as when an application business connector service (ABCS) calls an integrated application, the following occurs:

1. Detecting order fallout: An error occurs within an ABCS, it creates an error message, and the Oracle AIA error handling framework uses the message to create an enhanced fault message. See "[About Order Fallout Detection](#)" for more details.
2. Notifying someone of the fallout: For orders submitted from Siebel CRM, Oracle AIA creates a trouble ticket in Siebel CRM and Siebel CRM assigns the trouble ticket to a user and notifies the user. See "[About Order Fallout Notification](#)" for more details.
3. Correcting orders: The notified user resolves the issue and either resubmits the order or cancels it. See "[About Order Correction](#)" for more details.

## About Order Fallout Detection

When an error occurs within any of the Oracle AIA order services, the ABCS creates an error message. The Oracle AIA Error Handling framework detects the error message and uses it to create an enhanced fault message that contains the following information:

- Faulting Service
- Error Code
- Error Severity
- Error Text
- Time Of Failure
- Order ID
- Order Number
- Order Originating System Code
- Account ID
- Account Name

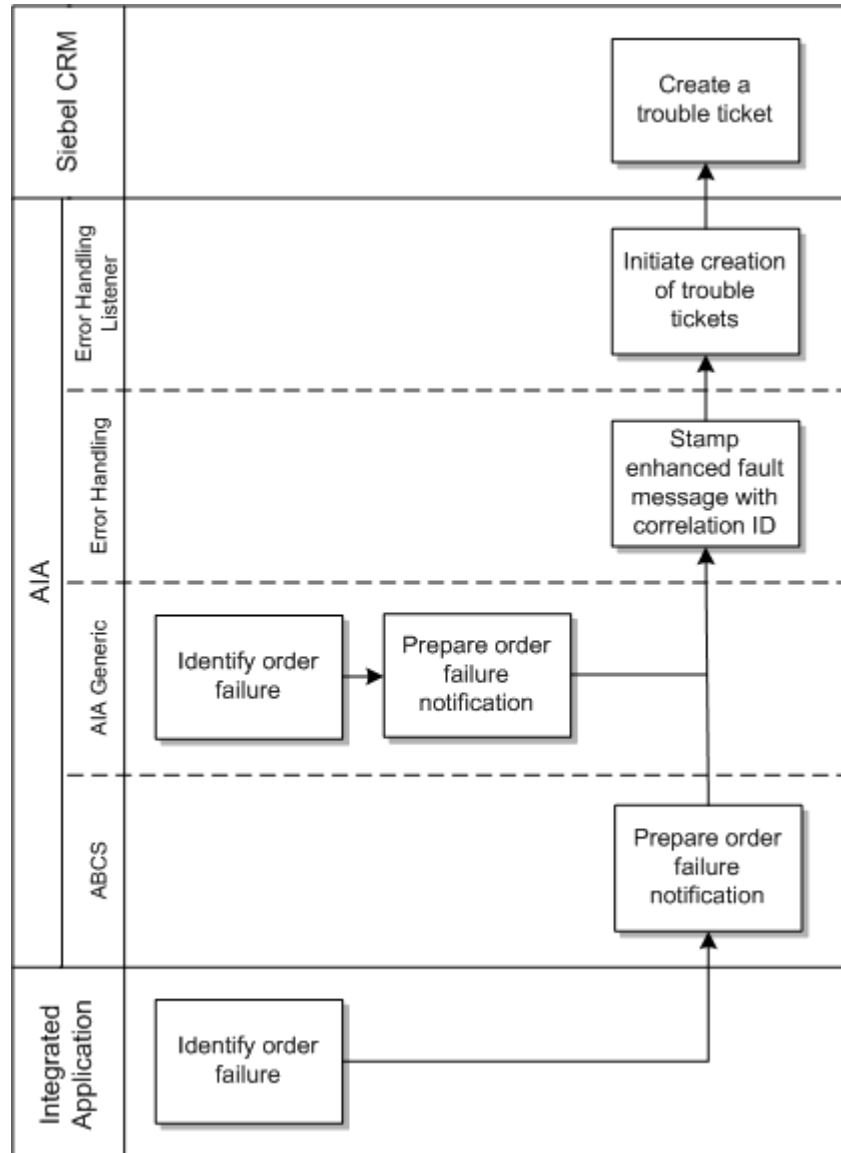
See "[Extending Fault Messages to Capture Order Fallout Information](#)" for more information about extending fault messages. See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about the Oracle AIA Error Handling Framework.

The framework publishes the enhanced fault message to the AIA Error JMS topic, where the error handling listener picks it up. The listener submits the fault message to the order fallout services to create a trouble ticket in Siebel CRM.

[Figure 21-4](#) illustrates order fallout detection when Oracle AIA initiates trouble ticket creation. When OSM initiates trouble ticket creation, the Oracle AIA error handling framework sends a notification to OSM, and OSM begins the process of notifying Siebel CRM.

Although the order can fail in any of the application tiers shown in the figure, this chapter discusses order failure only within Oracle AIA. Other applications and systems are outside the scope of the process integration for order fallout management.

Figure 21-4 Detection Flow from Oracle AIA for Order Fallout

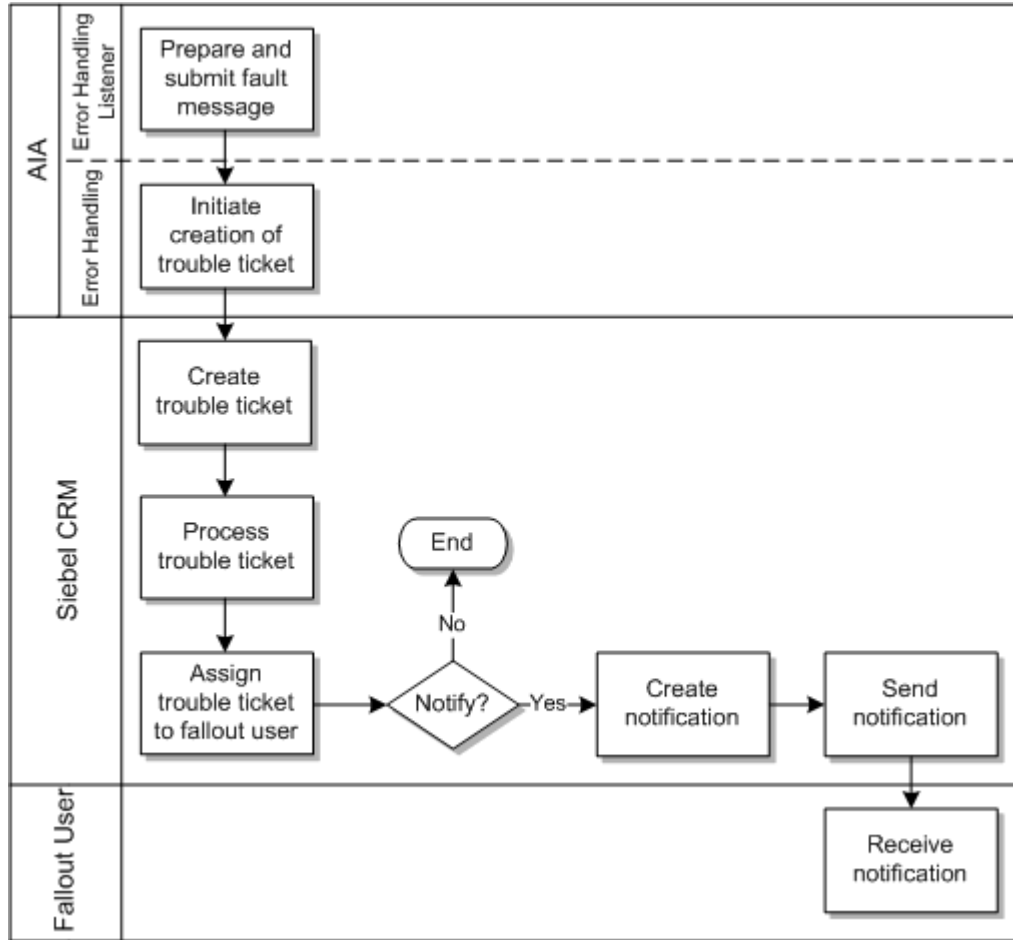


## About Order Fallout Notification

Siebel CRM handles order fallout notification. After Oracle AIA sends an enhanced fault message to Siebel CRM and creates a trouble ticket using a web service operation, Siebel CRM processes the trouble ticket and assigns it to a fallout user according to the Siebel CRM assignment rules. Siebel CRM can create and send notification to the fallout user, who can then investigate the trouble ticket to correct the issue.

Figure 21-5 illustrates the process for order fallout notification.

Figure 21-5 Order Fallout Notification Flow



## About Order Correction

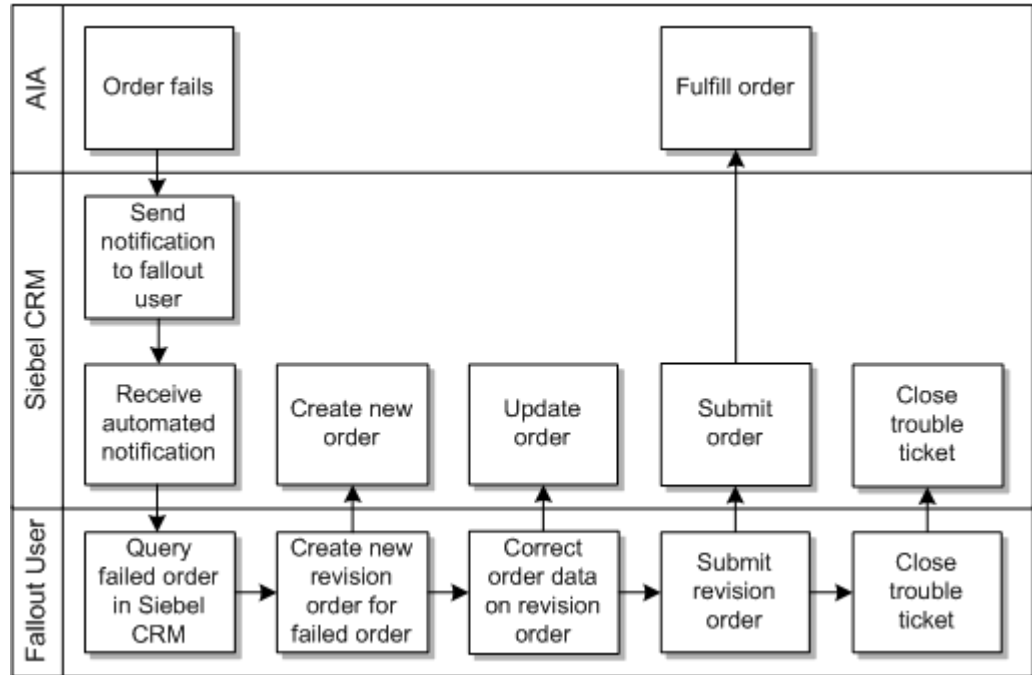
After the trouble ticket is created in Siebel CRM and assigned to a fallout user, the fallout user can investigate the failure and correct the error to resolve the trouble ticket.

To correct errors in the base order, submit a revision order with updated data. The OSM Order to Activate cartridge closes any trouble tickets created to report the order fallout and fulfills the revision order.

Figure 21-6 illustrates the steps a fallout user takes to correct errors in a base order.



Figure 21-6 Siebel CRM Correction Flow for Order Fallout



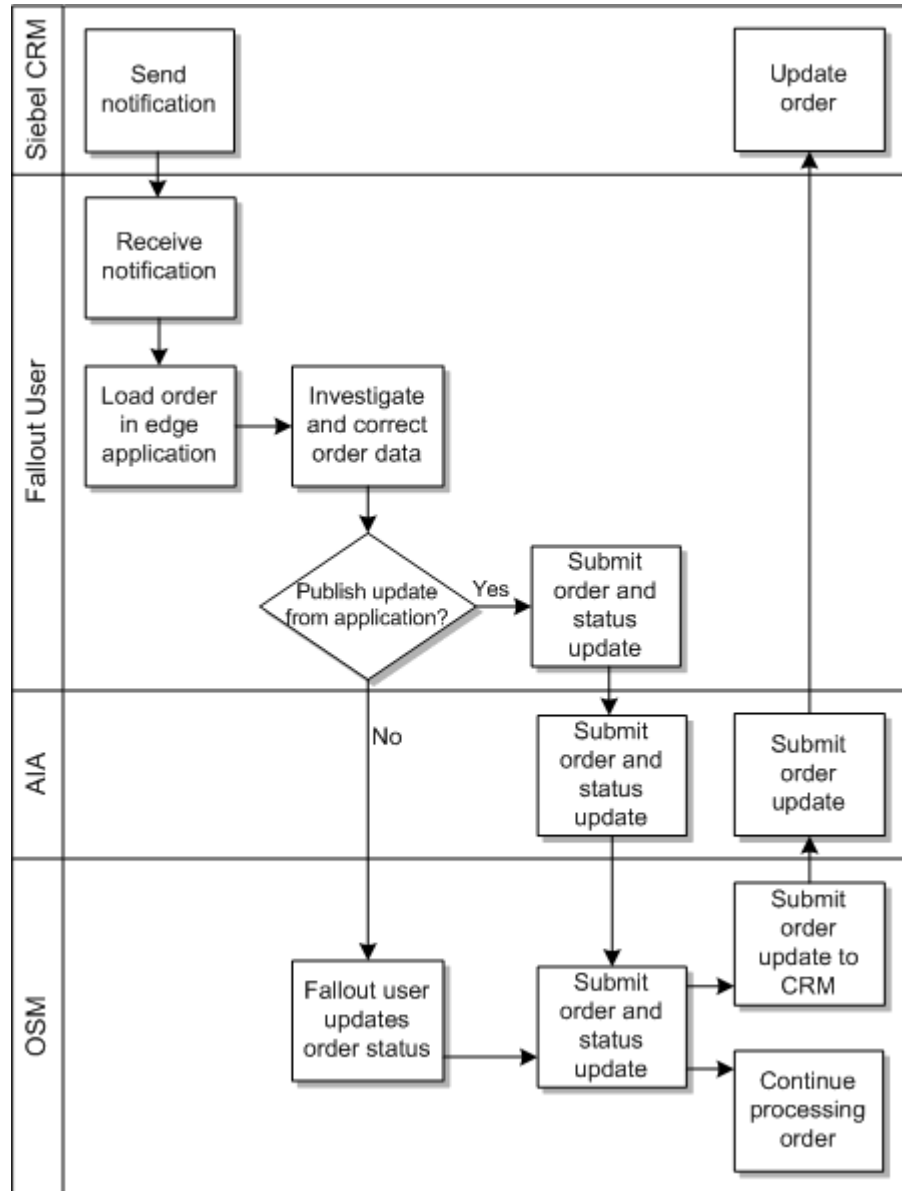
To correct errors in the data from participating edge applications (such as BRM, Siebel CRM, or an inventory or activation system), update the data in the edge application and resume the order OSM in the central order management role (OSM COM).

For example, if an order fails because of bad billing profile information, correct the billing profile information in Siebel CRM. Oracle AIA synchronizes the changed information to BRM, and you can resume the order in OSM.

For errors that occur and are corrected in local fulfillment systems, submitting a revision order from Siebel CRM does not correct the error or close the trouble tickets. Because the revision order is identical to the base order, OSM ignores the revision. You must resume the order from OSM.

Figure 21-7 illustrates process of correcting order data errors in edge applications. The figure shows an option for updating the order status from the edge application, which sends the order update through Oracle AIA, and an option for updating the order status directly in OSM.

Figure 21-7 Local Correction Flow

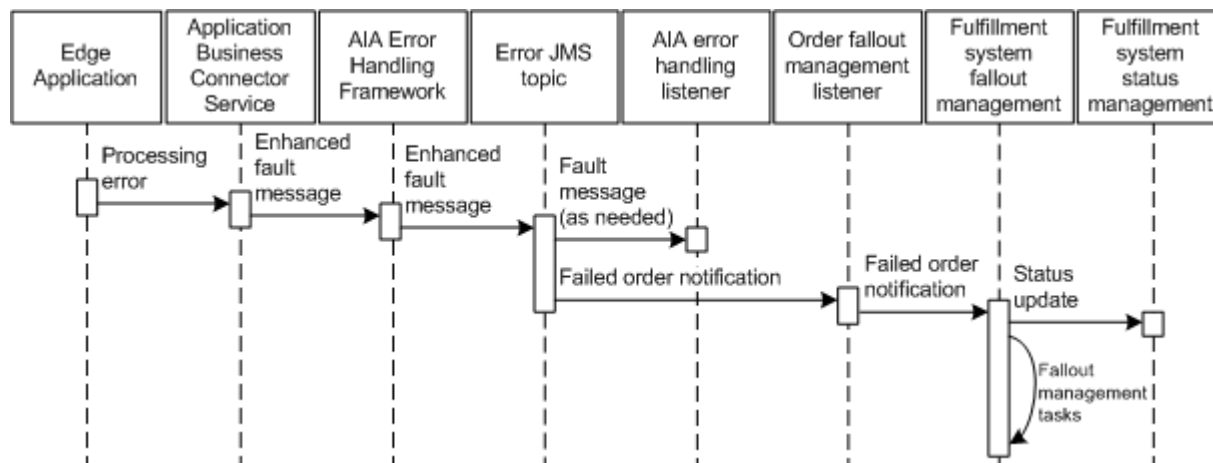


## How Oracle AIA Error Handling Framework Captures Faults

The Oracle AIA Error Handling Framework is used to capture faults across order processing.

Figure 21-8 illustrates the interactions taking place when an order failure is detected by an edge application, such as a billing or inventory system.

Figure 21-8 Capturing the Fault Sequence Diagram



The integration captures faults as follows:

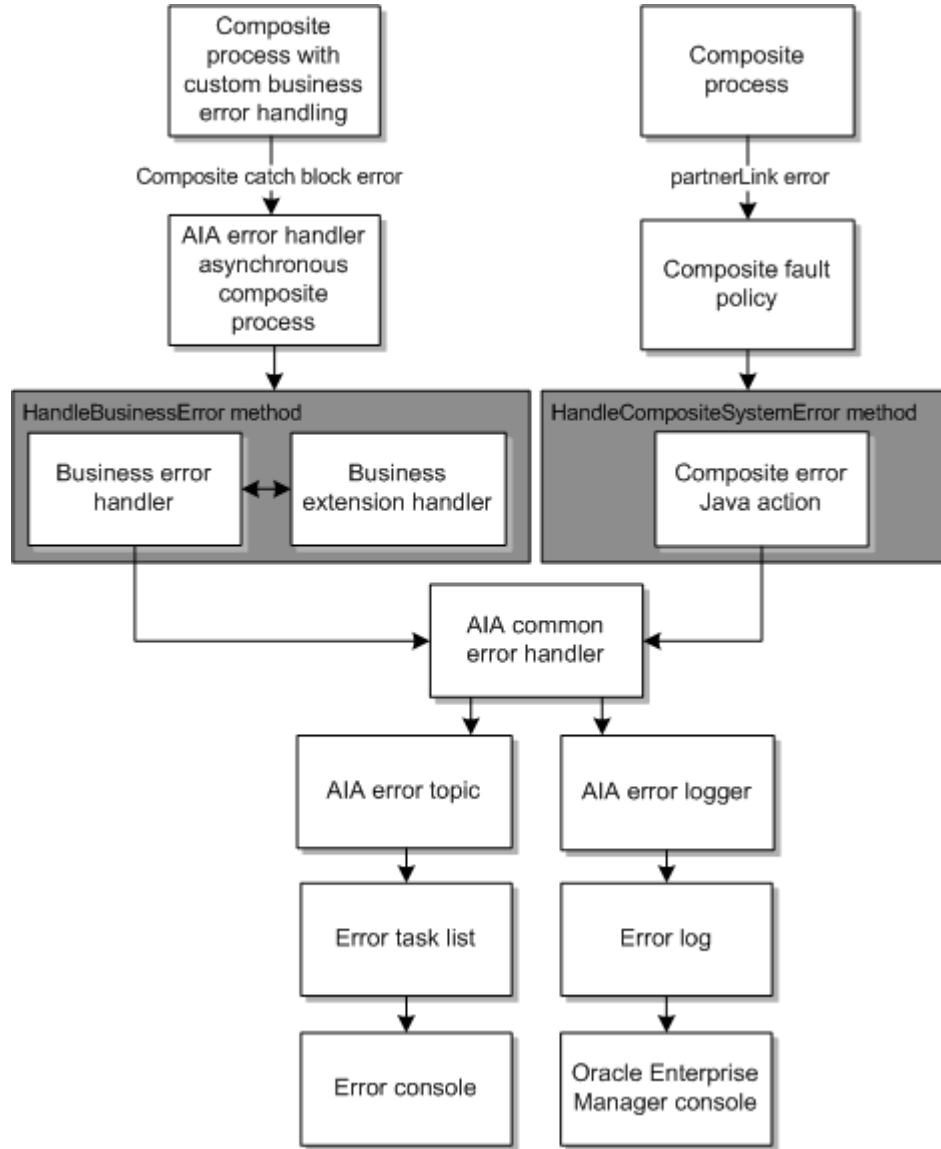
1. An edge application, such as BRM, encounters a processing error and sends the error to the Oracle AIA ABCS.
2. The ABCS creates an enhanced fault message from the error message and sends it to the Oracle AIA error handling framework.
3. The error handling framework sends the enhanced fault message to the error JMS topic.
4. If needed, the Oracle AIA error handling listener subscribed to the topic picks up the fault message.
5. The order fallout management listener subscribed to the topic picks up the failed order notification and sends it to the fulfillment system responsible for order fallout management.
6. The fulfillment system sends a status update to the system responsible for status management and proceeds with fallout management tasks such as assigning time of first failure, calculating and assigning recovery priority, mapping the application error code to a central error code, and generating a central error message based on the central error code.

The Oracle AIA error handling framework supports order fallout management by:

- Allowing custom enrichments to the fault message.
- Publishing the enriched fault message to the Oracle AIA Error topic.
- Using the order fallout listener to pick up only the messages that are relevant to the order failure.

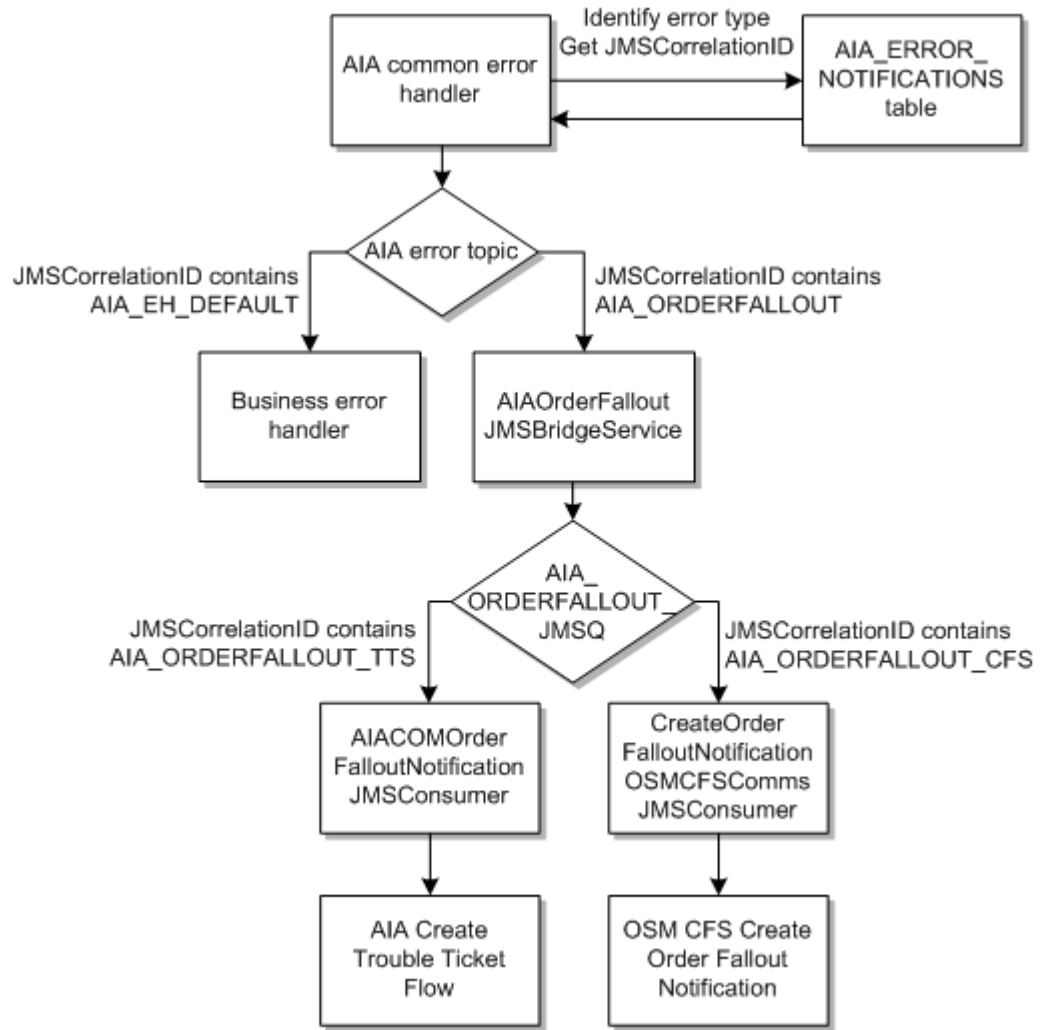
Figure 21-9 illustrates how the Oracle AIA Error Handling Framework submits an order failure notification to the Oracle AIA Error Topic.

Figure 21-9 Creation and Submission of a Fault Message to the Oracle AIA Error Topic



The custom listener selectively picks up the messages from the Oracle AIA Error Topic and initiates the appropriate Create Trouble Ticket Business flow, as shown in [Figure 21-10](#).

**Figure 21-10 Initiating Appropriate Create Trouble Ticket Flow**



The integration initiates the appropriate trouble ticket integration flow as follows:

1. The Oracle AIA common error handler reads the AIA\_ERROR\_NOTIFICATIONS table to identify the error type and adds the error type to the JMSHeader JMSCorrelationID attribute.
2. The Oracle AIA common error handler posts all of the enriched fault messages to the AIA\_ERROR\_TOPIC topic.
3. One of the following services picks up the fault message from AIA\_ERROR\_TOPIC:
  - The AIAOrderFalloutJMSBridgeService listener picks up messages with a JMSCorrelationID that includes AIA\_ORDERFALLOUT and publishes the fault messages to the AIA\_ORDERFALLOUT\_JMSQ queue for further filtering before creating trouble tickets. This queue ensures that the fault messages are not lost if there are more errors.
  - A business error handler picks up messages with other JMSCorrelationID values. These messages are managed outside of the process integration for Order Fallout Management.

4. One of the following services picks up the fault messages from AIA\_ORDERFALLOUT\_JMSQ:
  - The AIACOMOrderFalloutNotificationJMSConsumer picks up the fault messages with a JMSCorrelationID value that includes AIA\_ORDERFALLOUT\_TTS and initiates the Create Trouble Ticket from Oracle AIA business flow.
  - The CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer picks up the fault messages with JMSCorrelationID value that includes AIA\_ORDERFALLOUT\_CFS and initiates the Create and Manage Trouble Ticket from OSM business flow.

See "Using Error Type to Control Response to Order Fallout" for information about how to set up the seed data so that the trouble ticket is created from either Oracle AIA or OSM.

## Extending Fault Messages to Capture Order Fallout Information

The process integration for Order Fallout Management uses the Oracle AIA error handling framework to capture order failure notifications when an ABCS or an Oracle AIA service fails.

For faults that occur within Oracle AIA, the fault messages contain all the required details of the failed order and do not require enrichment by the Oracle AIA error handling framework.

For faults that occur in edge applications, you must extend the messages to capture additional order failure information. See the discussion of extending fault messages in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for information about extending error handling.

[Table 21-1](#) and [Table 21-2](#) list the additional fields that you must add to the fault messages to capture order failure information.

[Table 21-1](#) lists the order header-level fields to include in the fault messages.

**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       |

**Table 21-1 (Cont.) Order Header-Level Data**

| Field Name                              | Type         | Description  | Source  | Optional |
|---|--------------|--|---|----------|
| AccountID (Common)                      | ID           | Common Account ID.   | Oracle AIA service                                  | Yes      |
| Order ID                                | ID           | Alphanumeric identifier for the order. Assigned by fulfillment system to the order. The fulfillment system uses it to correlate the order back to the common order ID received for the original order. The common order ID is then mapped to the Siebel order ID by the Siebel ABCS. | Fulfillment System                                  | No       |
| Order Number                            | AlphaNumeric | User-friendly identifier for the order in the fulfillment system.  | Fulfillment System                                  | Yes      |
| ProductID                               | AlphaNumeric | Alphanumeric identifier for the product used for the failed line or the product for the first order line in case of multiple line failures.  | Siebel CRM or Oracle AIA service                    | Yes      |
| 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.<br>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 from the edge application or Oracle AIA service to the process integration for Order Fallout Management.

Table 21-2 lists order line item level fields to include in the fault messages. when the Oracle AIA service or the edge application 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 are not required.

**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      |

To extend error handling, at a high level:

1. Extend the Oracle AIA fault message to capture the additional information in [Table 21-1](#) and [Table 21-2](#).
2. Extend the common error handler to:
  - Identify when a fault message is related to order failures.
  - Stamp the error type in the fault message as a JMSCorrelationID and invoke the appropriate fault extension handlers (in case of a partner link fault).
  - Publish to the AIA Error JMS Topic.
3. Create 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. Create the AIACOMOrderFalloutNotificationConsumer listener for the Order fallout queue 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

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 CRM and Oracle Communications Billing and Revenue Management (BRM) pre-built integration options (the integration) implement 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)
- CreateTroubleTicketSiebelCommsProvABCServiceImpl - 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

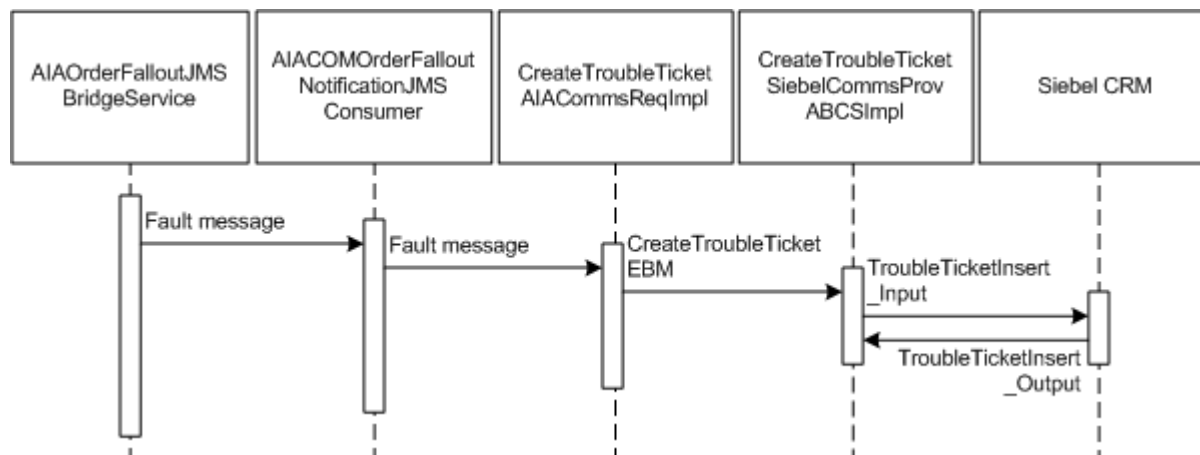
## Creating a Trouble Ticket in Siebel CRM Integration Flow

This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCServiceImpl

[Figure 22-1](#) illustrates the create trouble ticket integration flow.

Figure 22-1 Create Trouble Ticket Sequence Diagram



The integration creates a trouble ticket in Siebel CRM as follows:

1. The AIAOrderFalloutJMSBridgeService listener picks up an enriched fault message with a JMSCorrelationID value of AIA\_ORDERFALLOUT\_TTS from the AIA\_ERROR\_TOPIC topic and drops it into the AIA\_ORDERFALLOUT\_JMSQ queue.
2. The AIACOMOrderFalloutNotificationJMSConsumer picks up the message and sends it to the CreateTroubleTicketAIACommsReqImpl service.
3. CreateTroubleTicketAIACommsReqImpl transforms the fault message into the CreateTroubleTicketEBM message, and sends it to the CreateTroubleTicketSiebelCommsProvABCSImpl service.
4. CreateTroubleTicketSiebelCommsProvABCSImpl calls the Siebel CRM SWITroubleTicketIO web service with the SWITroubleTicketInsert operation.
5. The Siebel CRM web service operation responds with the trouble ticket ID in an application business message (ABM).
6. Depending on the response code in CreateTroubleTicketEBM, CreateTroubleTicketSiebelCommsProvABCSImpl may transform the ABM into CreateTroubleTicketResponseEBM and send it to CreateTroubleTicketAIACommsReqImpl.

## 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
- CreateTroubleTicketSiebelCommsProvABCSImpl

**Table 22-1 Transaction Boundaries and Recovery Details**

| Transaction  | DB Operations                              | In Case of Error                               | Recovery                             |
|--|--|--|--------------------------------------|
| The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID and publishes to AIA_ORDERFALLOUT_JMSQ.  | Message enqueued in AIA_ORDERFALLOUT_JMSQ. | Rollback JMS message to AIA_ERROR_TOPIC.       | Resubmit from AIA_ERROR_TOPIC.       |
| AIAComOrderFalloutNotificationJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_TTS and invokes CreateTroubleTicketAIACommsReqImpl, which parses fault message and routes to CreateTroubleTicketSiebelCommsProviderABCImpl. | AIA cross-reference entries.               | Rollback the message to AIA_ORDERFALLOUT_JMSQ. | Resubmit from AIA_ORDERFALLOUT_JMSQ. |

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about rollback procedures.

## 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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home/comms\_home/source/soainfra/apps/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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

The following services are delivered with the Create Trouble Ticket from Oracle AIA business flow:

- CreateTroubleTicketSiebelCommsProvABCServiceImpl
- 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.

## CreateTroubleTicketSiebelCommsProvABCServiceImpl

The CreateTroubleTicketSiebelCommsProvABCServiceImpl 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.

If the **TroubleTicket.GenerateTroubleTicketResponse** configuration property is set to **True** or if the response code attribute is specified on the EBM, this process creates a trouble ticket response message and a cross-reference for the trouble ticket ID with the Siebel ID. Otherwise, this process ignores any response.

This service is 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.

## 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 persists the enhanced fault message into a fallout queue so that it can be retried in case of errors in the downstream process. In this business flow, the messages in this queue are used by Oracle AIA to create a trouble ticket in Siebel CRM.

## 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
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCImplV2
- SyncCustomerPartyListBRMCommsProvABCImpl

# Implementing the Create and Manage Trouble Ticket from OSM Business Flow

This chapter explains how the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash Integration Pack (the integration) implements the Create and Manage Trouble Ticket from OSM business flow.

## About 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:

- Order to Cash for Oracle Communications Order and Service Management (OSM)
- Order to Cash for Siebel CRM

The Create and Manage Trouble Ticket from OSM business flow includes the following integration flows:

- 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

## About the 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`. Errors or faults with a correlation ID of `AIA_ORDERFALLOUT_CFS` are enriched and published to OSM.

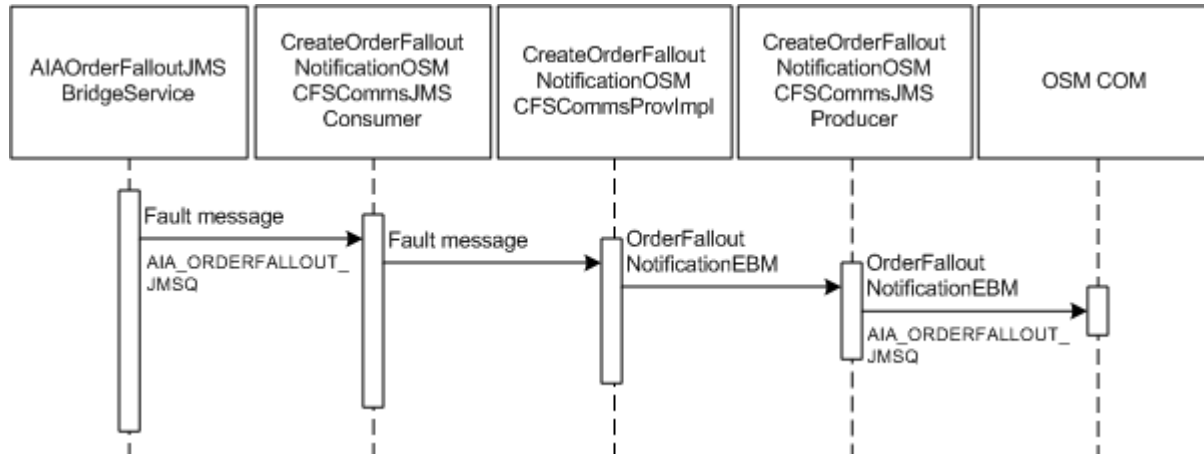
This integration flow includes the following services:

- `AIAOrderFalloutJMSBridgeService`
- `CreateOrderFalloutNotificationOSMCFSCommsJMSSConsumer`
- `CreateOrderFalloutNotificationOSMCFSCommsProvImpl`
- `CreateOrderFalloutNotificationOSMCFSCommsJMSProducer`

[Figure 23-1](#) shows the sequence of integration services through which the fallout notification message passes for the Order Failure Notification to OSM integration flow.



**Figure 23-1 Order Failure Notification to OSM Sequence**



When this process initiates, the following events occur:

1. The AIAOrderFalloutJMSBridgeService listener picks up an enriched fault message with a JMSCorrelationID value of AIA\_ORDERFALLOUT\_CFS from the AIA\_ERROR\_TOPIC topic and adds it to the AIA\_ORDERFALLOUT\_JMSQ queue.
2. The CreateOrderFalloutNotificationOSMCFSCCommsJMSConsumer service picks up the message and sends it to the CreateOrderFalloutNotificationOSMCFSCCommsProvImpl service.
3. CreateOrderFalloutNotificationOSMCFSCCommsProvImpl transforms the message into the OrderFalloutNotificationEBM message and sends it to the CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer service.
4. CreateOrderFalloutNotificationOSMCFSCCommsJMSProducer adds the message to the AIA\_FALLOUT\_JMSQ queue, where OSM in the central order management role (OSM COM) picks it up.
5. OSM COM marks the order as failed and initiates the request to create a trouble ticket for Siebel CRM.

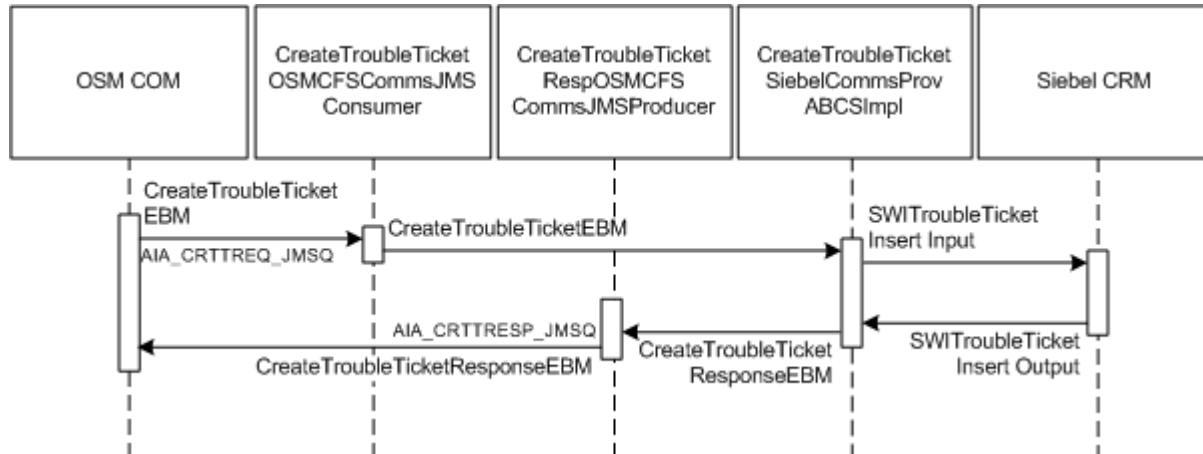
## About the Creating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow includes the following services:

- CreateTroubleTicketOSMCFSCCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCImpl
- CreateTroubleTicketOSMCFSCCommsJMSProducer

Figure 23-2 shows the sequence of integration services through which the fallout notification message passes for the Creating a Trouble Ticket in Siebel CRM from OSM integration flow.

Figure 23-2 Creating a Trouble Ticket in Siebel CRM from OSM



The integration creates trouble tickets in Siebel CRM from OSM as follows:

1. OSM COM adds the CreateTroubleTicketEBM message to the AIA\_CRTTREQ\_JMSQ queue.
2. The CreateTroubleTicketOSMCFSCommsJMSConsumer service picks up the message and sends it to the CreateTroubleTicketSiebelCommsProvABCSImpl service.
3. CreateTroubleTicketSiebelCommsProvABCSImpl transforms CreateTroubleTicketEBM into an application business message (ABM) and sends it to the SWITroubleTicket Siebel CRM web service with the SWITroubleTicketInsert operation.
4. SWITroubleTicket sends a response ABM that includes the trouble ticket ID to CreateTroubleTicketSiebelCommsProvABCSImpl.
5. CreateTroubleTicketSiebelCommsProvABCSImpl transforms the ABM into the CreateTroubleTicketResponseEBM message and sends it to the CreateTroubleTicketRespOSMCFSCommsJMSProducer service.
6. CreateTroubleTicketRespOSMCFSCommsJMSProducer adds CreateTroubleTicketResponseEBM to the AIA\_CRTTRESP\_JMSQ queue.
7. OSM COM picks up the message and stores the trouble ticket ID for reference.

## Defining Transaction Boundaries and Recovery Details

Table 23-1 describes the transactions, the database operations, and what actions to take in case of an error for these flows.

The following services are involved:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer
- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCSImpl
- CreateTroubleTicketRespOSMCFSCommsJMSProducer

**Table 23-1 Transaction Boundaries and Recovery Details**

| Transaction  | DB Operations   | In Case of Error                                | Recovery                             |
|--|---|---|--------------------------------------|
| The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID and publishes to AIA_ORDERFALLOUT_JMSQ.  | Message added to AIA_ORDERFALLOUT_JMSQ.   | Roll back JMS message to AIA_ERROR_TOPIC.       | Resubmit from AIA_ERROR_TOPIC.       |
| CreateOrderFalloutNotificationOSM CFSCommsJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_CFS and sends it to CreateOrderFalloutNotificationOSM CFSCommsProvImpl, which parses fault message and sends it to CreateOrderFalloutNotificationOSM CFSCommsJMSProducer.   | Message is submitted to OSM through AIA_FALLOUT_JMSQ.   | Roll back the message to AIA_ORDERFALLOUT_JMSQ. | Resubmit from AIA_ORDERFALLOUT_JMSQ. |
| CreateTroubleTicketOSMCFSCommsJMSConsumer picks up message and sends the EBM to CreateTroubleTicketSiebelCommsProvABCImpl. The ABCS invokes a Siebel web service., which sends a response message with the trouble ticket ID to CreateTroubleTicketSiebelCommsProvABCImpl. The ABCS sends the response to CreateTroubleTicketRespOSMCFSCommsJMSProducer. | AIA cross-references created.<br>Message added to AIA_CRTTTREQ_JMSQ<br>Message added to AIA_CRTTTRESP_JMSQ. | Roll back the message to AIA_CRTTTREQ_JMSQ.     | Resubmit from AIA_CRTTTREQ_JMSQ.     |

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about rollback procedures.

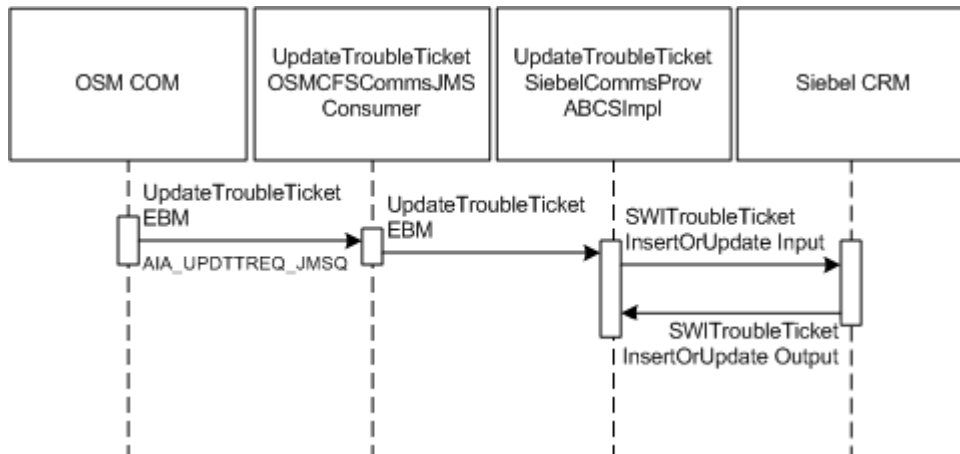
## Updating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow uses the following interfaces:

- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCImpl

**Figure 23-3** shows the sequence of integration services through which the trouble ticket update message passes for the Updating a Trouble Ticket in Siebel CRM from OSM integration flow.

**Figure 23-3 Updating a Trouble Ticket in Siebel CRM from OSM**



The integration updates trouble tickets in Siebel CRM from OSM as follows:

1. OSM COM adds the UpdateTroubleTicketEBM message to the AIA\_UPDTTREQ\_JMSQ queue.
2. The UpdateTroubleTicketOSMCFSCCommsJMSConsumer services picks up the message and sends it to the UpdateTroubleTicketSiebelCommsProvABCSImpl service.
3. UpdateTroubleTicketSiebelCommsProvABCSImpl service transforms UpdateTroubleTicketEBM into an ABM and sends it to SWITroubleTicket with the SWITroubleTicketInsertOrUpdate operation.
4. SWITroubleTicket sends a response ABM to UpdateTroubleTicketSiebelCommsProvABCSImpl.

## 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
- UpdateTroubleTicketSiebelCommsProvABCSIm

**Table 23-2 Transaction Boundaries and Recovery Details**

| Transaction   | DB Operations  | In Case of Error  | Recovery                         |
|---|--|---|----------------------------------|
| UpdateTroubleTicketOSMCFSCCommsJMSConsumer 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.<br>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 following directories contain the industry component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:

***Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1***

- Enterprise business service (EBS) WSDL files:

***Oracle\_home/comms\_home/source/soainfra/apps/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 the discussion of Oracle AIA assets extensibility patterns in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about extending EBOs.

## Integration Services

The following services are delivered with the Create and Manage Trouble Ticket from OSM business flow:

- [AIAOrderFalloutJMSBridgeService](#)
- [CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer](#)
- [CreateOrderFalloutNotificationOSMCFSCommsProvImpl](#)
- [CreateOrderFalloutNotificationOSMCFSCommsJMSProducer](#)
- [CreateTroubleTicketOSMCFSCommsJMSConsumer](#)
- [CreateTroubleTicketSiebelCommsProvABCImpl](#)
- [CreateTroubleTicketRespOSMCFSCommsJMSProducer](#)
- [UpdateTroubleTicketOSMCFSCommsJMSConsumer](#)
- [UpdateTroubleTicketSiebelCommsProvABCImpl](#)
- [CreateFaultNotificationLFCommsJMSConsumer](#)

### AIAOrderFalloutJMSBridgeService

The AIAOrderFalloutJMSBridgeService service is a mediator service that does the following:

1. Picks up the fault message from the AIA Error Topic.
2. Adds the fault message to the AIA\_ORDERFALLOUT\_JMSQ.

This service persists the enhanced fault message into a fallout queue so that it can be retried in case of errors in the downstream process. In this business flow, the messages in this queue are used to send an order failure notification to OSM.

### CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer

The CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer service is a Mediator service that does the following:

1. Picks up the fault message from the AIA\_ORDERFALLOUT\_JMSQ.
2. Sends the fault message to the CreateOrderFalloutNotificationOSMCFSCommsProvImpl process based on the JMS Correlation ID.

This service acts as a consumer, listening to the messages produced in AIA\_ORDERFALLOUT\_JMSQ.

### CreateOrderFalloutNotificationOSMCFSCommsProvImpl

The CreateOrderFalloutNotificationOSMCFSCommsProvImpl service is a business process execution language (BPEL) process that does the following:

1. Picks up the fault message from CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer.
2. Transforms the fault message into the OrderFalloutNotificationEBM message.
3. Sends OrderFalloutNotificationEBM to the CreateOrderFalloutNotificationOSMCFSCommsJMSProducer service.

## CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

The CreateOrderFalloutNotificationOSMCFSCommsJMSProducer is a BPEL process that does the following:

1. Receives OrderFalloutNotificationEBM from CreateOrderFalloutNotificationOSMCFSCommsProvImpl.
2. Adds the EBM to the AIA\_FALLOUT\_JMSQ queue from which OSM picks up the message to trigger an OSM fallout event

## CreateTroubleTicketOSMCFSCommsJMSSConsumer

The CreateTroubleTicketOSMCFSCommsJMSSConsumer is a Mediator service that does the following:

1. Picks up the CreateTroubleTicketEBM message from the AIA\_CRTTTREQ\_JMSQ queue.
2. Checks the value of the target ID on the EBM and does one of the following:
  - If the target ID is **CPQ\_01**, sends the EBM to CreateTroubleTicketCQCommsJMSProducer.
  - If the target ID is **SEBL\_01**, sends the EBM to CreateTroubleTicketSiebelCommsProvABCImpl.

This service acts as a consumer, listening to the messages produced in the AIA\_CRTTTREQ\_JMSQ SAF queue.

## CreateTroubleTicketSiebelCommsProvABCImpl

The CreateTroubleTicketSiebelCommsProvABCImpl service is an asynchronous BPEL process that does the following:

1. Receives CreateTroubleTicketEBM from CreateTroubleTicketOSMCFSCommsJMSSConsumer.
2. Transforms CreateTroubleTicketEBM into the Siebel CRM web service input.
3. Invokes the Siebel web service to create the trouble ticket.
4. Receives a response from the web service with the trouble ticket ID.
5. If the **TroubleTicket.GenerateTroubleTicketResponse** configuration property is set to **True** or if the response code attribute is specified on the EBM, transforms the response ABM into the CreateTroubleTicketResponseEBM and sends it to CreateTroubleTicketRespOSMCFSCommsJMSProducer.

This service is 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.

## CreateTroubleTicketRespOSMCFSCommsJMSProducer

The CreateTroubleTicketRespOSMCFSCommsJMSProducer is a BPEL process that does the following:

1. Receives CreateTroubleTicketResponseEBM from CreateTroubleTicketSiebelCommsProvABCImpl.

2. Adds CreateTroubleTicketResponseEBM to the AIA\_CRTTTRESP\_JMSQ SAF queue from which OSM picks up the message and updates the order task with the created trouble ticket ID.

## UpdateTroubleTicketOSMCFSCCommsJMSSConsumer

The UpdateTroubleTicketOSMCFSCCommsJMSSConsumer is a Mediator service that does the following:

1. Picks up the UpdateTroubleTicketEBM message from the AIA\_UPDTTREQ\_JMSQ queue.
2. Sends UpdateTroubleTicketEBM to UpdateTroubleTicketSiebelCommsProvABCImpl.

This service acts as a consumer, listening to the messages produced in the AIA\_UPDTTREQ\_JMSQ queue.

## UpdateTroubleTicketSiebelCommsProvABCImpl

The UpdateTroubleTicketSiebelCommsProvABCImpl is a BPEL process that does the following:

1. Receives UpdateTroubleTicketEBM from UpdateTroubleTicketOSMCFSCCommsJMSSConsumer.
2. Transforms the EBM into the TroubleTicketInsertorUpdate\_Input ABM.
3. Invokes the Siebel CRM web service to update the trouble ticket.

## CreateFaultNotificationLFCommsJMSSConsumer

The CreateFaultNotificationLFCommsJMSSConsumer is a Mediator service that does the following:

1. Picks up the fault message from the AIA\_LFERROR\_JMSQ queue.
2. Sends the fault message to the AIAAsyncErrorHandlingBPELProcess service, which adds the message to the AIA\_ERROR\_TOPIC topic.

Provisioning systems that want to notify the central fulfillment system about an error in processing the order create a fault message that includes order details which is added to the AIA\_LFERROR\_JMSQ. This fault message is processed by the Order Fallout Management framework, and OSM is notified about the errors in the provisioning system for the 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
- ProcessSalesOrderFulfillmentOSMCFSCCommsJMSSProducer
- ProcessFOBillingAccountListRespOSMCFSCCommsJMSSProducer
- ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSSProducer



- TestOrderOrchestrationEBF
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSSConsumer
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSSConsumer\_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 describes how to configure the process integration for product lifecycle management in the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash pre-built integration (the integration).

## About Configuring the Process Integration for Product Lifecycle Management

Configuring the process integration for product lifecycle management includes:

- Configuring Oracle Communications Billing and Revenue Management (BRM) to integrate with 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.

This chapter assumes that you have already installed BRM and Siebel CRM. See the Oracle AIA software compatibility matrix available from My Oracle Support for information about required versions and patches.

## Configuring Siebel CRM for Integrated Product Lifecycle Management

To configure Siebel CRM for product lifecycle management:

1. 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](#)".

2. 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](#).
3. 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](#).
4. Set up friends and family products. See "[About Friends and Family Lists](#)" for more information.
5. 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.
6. Define simple Special Rating products and set their composition type to **Partial**.
7. 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.
8. Set up promotions, bundling service bundles, account level products, and discounts.
9. Add service bundles and promotions to the price list used by the product synchronization integration flow.

## Configuring BRM for Integrated Product Lifecycle Management

To configure BRM for product lifecycle management:

1. Install configure, and run Synchronization Queue Data Manager. You may have completed this task as part of your BRM installation.

This data manager lets you synchronize changes in the BRM database to Siebel CRM by using Oracle AIA. See *Oracle Communications Billing and Revenue Management Synchronization Queue Manager* for more information.

2. In the `Infranet.properties` file for the BRM Payload Configuration, set the `infranet.eai.xml_zero_epoch_as_null` parameter to **TRUE**.

This causes immediate effective start dates and end dates with infinite effectivity to be set as a null date value in Siebel CRM.

See the discussion of defining infinite start date and end date values in *Oracle Communications Billing and Revenue Management Developer's Guide* for more information about setting this parameter.

3. Set up data in BRM as described in the BRM documentation. This includes creating the following objects and data in the BRM database:
  - Services and events: BRM includes internet access and email services by default. Configure a list of events to track each service, and new events to track new service.
  - Resources: Create resources to supplement the rate plans that are associated with each product. Resources include currency resources, such as USD, and noncurrency resources, such as minutes.
  - Currency exchange rates
  - 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.
  - Tax codes and tax suppliers: Optionally, define tax codes and tax suppliers if you are calculating taxes using Taxware.
  - Rateable Usage Metrics (RUMs) and mappings: BRM uses RUMs to identify the event attributes that define rates for each event. RUM definitions are stored in the BRM database.
  - Mappings:
    - Map event types to RUMs: Associate each event with a list of RUMs. When you create products, you will associate a rate plan structure with every RUM that is linked for the event.
    - Map event types to services: Map event types to the valid services. Not all event types are valid for all services. When you create products, you will select a set of services and events to be rated. Mapping event types to valid services prevents you from selecting invalid combinations.
  - 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.
  - Impact categories: For real-time rating, impact categories are used to specify that particular groups of balance impacts within rates must be used. To use attribute value grouping during rating, you must create impact categories.
  - Pipeline data: For pipeline rating, create several types of data and pricing components.
  - Friends and family pricing
  - Pricing components, such as products, discounts, and chargeshares. You associate pricing components with events and rate plans.

BRM triggers functional events that identify changes (additions, deletions, modification) that trigger synchronization of the new or changed pricing components to Siebel CRM.

## 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 product lifecycle management.

**Table 24-1 Product Lifecycle Management DVMs**

| DVM                  | Description   |
|----------------------|---|
| PRICECHARGETYPE      | Price Charge Type, such as One-Time or Recurring.               |
| PRICECHARGETYPEUOM   | Price Charge Type Unit Of Measure such as Per Day or Per Month. |
| PRICETYPE_EVENT      | Price Type Event such as Purchase or Cancel.                    |
| PRODUCTTYPECODE      | Product Type Code such as Item or Subscription.                 |
| ITEM_BILLINGTYPECODE | Maps Billing Type from BRM to Siebel CRM.                       |
| RESOURCE             | Non-monetary resources such as free minutes or text messages.   |
| 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 lifecycle 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  | COMMON                                   | SEBL_01  | 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 Product ID.   |
| 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. |
| SIEBELPRODUCTEVENTXREF | ITEM_ID_COMMON<br>From<br>ITEM_ID.COMMON | LINEPRICETYPE_CODE<br>PRICELINETYPE_ID.COMMON    | Not applicable                       | 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 Oracle AIA-specific errors are caused by product lifecycle management services for Siebel CRM.

For more information about the errors caused by BRM or Siebel CRM, see the documentation for BRM and Siebel CRM.

### About Error Notification Roles and Users

The integration uses the following default roles and users to issue error notifications:

- **Role:** AIAIntegrationAdmin
- **User:** AIAIntegrationAdminUser

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about configuring error notifications.

## Configuring Properties for Product Lifecycle Management

Configure the properties described in this section in the `Comms_home/source/soainfra/apps/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

## Configuration Properties for the Synchronize Product and Price Business Flow

This section describes the properties for the services in the Synchronize Product and Price business flow.

### Configuration Properties for BRM Services

Table 24-3 shows the properties for the SyncProductBRMCommsReqABCSEImpl service.

**Table 24-3 SyncProductBRMCommsReqABCSEImpl Properties**

| Property Name   | Value/Default Values  | Description  |
|---|---|--|
| Default.SystemID                                      | BRM_01  | Default target billing system instance code defined in BSR. Used if the request message does not contain the system instance ID.   |
| ABCSExtension.PreXFormABMtoSyncItemCompositionListEBM | true/false. Default = false   | Enables or disables the application business connector service (ABCS) extension at the predefined plug-into-point.<br>If set to true, the Extension process defined in Oracle AIA ABCS extension guidelines is invoked. The name of the property indicates which extension point is enabled. |
| ABCSExtension.PreInvokeItemCompositionEBS             | true/false. Default = false   | Enables or disables the ABCS extension at the predefined plug-into-point.<br>If set to true, the Extension process is invoked. This property is required for extensibility. The name of the property indicates which extension point is enabled.   |
| ABCSExtension.PreXFormABMtoPriceListListEBM           | true/false. Default = false   | Enables or disables the ABCS extension at the predefined plug-into-point.<br>If set to true, 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   | Enables or disables the ABCS extension at the predefined plug-into-point.<br>If set to true, the extension process defined in AIA ABCS Extension guidelines is invoked. The name of the property indicates which extension point is enabled.   |
| CallBackAddress                                       | http://host:port/soa-infra/services/default/SyncProductBRMCommsReqABCSEImpl/SyncProductBRMCommsReqABCSEImpl | Sets the ReplyTo element in the EBM header. The provider ABCS uses this WSAAddress, if present, when sending the response.   |



**Table 24-3 (Cont.) SyncProductBRMCommsReqABCImpl Properties**

| Property Name  | Value/Default Values  | Description   |
|--|---|---|
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList             | true/false. Default = true  | Indicates if EBS call needs to be bypassed. If true, it uses the properties below to identify the service it should invoke. |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.Address     | http://host:port/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl                                   | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.PortType    | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl                     | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.      |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.ServiceName | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl                     | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property.   |
| EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList                         | true/false. Default = true  | Indicates if EBS call needs to be bypassed. If true, it uses the properties below to identify the service it should invoke. |
| EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.Address                 | http://host:port/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl               | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.PortType                | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.      |
| EBSOverride.CommunicationsPriceListEBSV2.SyncPriceListList.ServiceName             | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property.   |

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  | The Siebel CRM system instance code defined in BSR. Used if the request message does not contain the system instance ID.   |
| ABCSExtension.PreXFormABMtoEBM   | true/false. Default = false   | Enables or disables the ABCS extension at the predefined plug-into-point.<br>If set to true, 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   | Enables or disables the ABCS extension at the predefined plug-into-point.<br>If set to true, 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  | Indicates if EBS call needs to be bypassed. If true, it uses the properties below to identify the service it should invoke.  |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.Address     | http://host.port/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl               | The address endpoint URI of the web service to be invoked dynamically.   |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.PortType    | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.   |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.ServiceName | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property.  |

## Configuration Properties for Siebel CRM Services

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. Used if the request message does not contain the target system ID.  |
| Routing.SiebelProductService.SEBL_01.EndpointURI   | http://host:port/eai_enu/start.swe?SWExtSource=SecureWebService&SWExtCmd=Execute&WSSOAP=1 | The SOAP end point URL for the Siebel CRM product import web service. Used if the request message does not contain the target URL.  |
| Routing.SiebelProductService.RouteToCAVS           | true/false. Default = false   | If true, invokes the Siebel CRM end point in the Routing.SiebelProductService.SEBL_01.EndpointURI property for this service.<br>If false, invokes the verification system end point in the system-level SyncResponseSimulator.Soap.EndpointURL property.  |
| Routing.SiebelPriceListService.SEBL_01.EndpointURI | http://host:port/eai_enu/start.swe?SWExtSource=SecureWebService&SWExtCmd=Execute&WSSOAP=1 | The SOAP end point URL for the Siebel CRM price list web service. Used if the request message does not contains the target URL.   |
| Routing.SiebelPriceListService.RouteToCAVS         | true/false. Default = false   | If true, invokes the Siebel CRM end point in the Routing.SiebelProductService.SEBL_01.EndpointURI property for this service.<br>If false, invokes the verification system end point in the system-level SyncResponseSimulator.Soap.EndpointURL property.  |
| 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 synchronizing products.   |
| Siebel.SEBL_01.Product.Workspace Name              | Demo Workspace  | The name of the Siebel CRM workspace. Create a workspace and update this property before synchronizing products.  |
| 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 in the Siebel.SEBL_01.PriceList.ID property. If the currency of the prices in PriceListEBM does not match this currency, the price in Siebel CRM is set to zero. Set this value before synchronizing products.  |

**Table 24-5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl Properties**

| Property Name  | Value/Default Values   | Description   |
|--|--|---|
| ABCSExtension.PreXFormEBMtoABM   | true/false. Default = false  | Determines whether the ABCS should invoke the extension service before the EBM to ABM transformation.                       |
| ABCSExtension.PostXFormEBMtoABM  | true/false. Default = false  | Determines whether the ABCS should invoke the extension service after the EBM to ABM transformation.                        |
| ABCSExtension.PreInvokeABS   | true/false. Default = false  | Determines whether the ABCS should invoke the extension service before invoking the application.                            |
| ABCSExtension.PostInvokeABS  | true/false. Default = false  | Determines whether the ABCS should invoke the extension service after invoking the application.                             |
| ABCSExtension.PreXFormPriceListListEBMtoItemCompositionEBM                         | true/false. Default = false  | Determines whether the ABCS should invoke the extension service before the EBM to EBM transformation.                       |
| ABCSExtension.PreInvokeItemCompositionEBS  | true/false. Default = false  | Determines whether the ABCS should invoke the extension service before invoking the application.                            |
| ABCSExtension.PreXFormPriceListListEBMtoProductABM                                 | true/false. Default = false  | Determines whether the ABCS should invoke the extension service before the EBM to ABM transformation.                       |
| ABCSExtension.PreInvokeProductABM  | true/false. Default = false  | Determines whether the ABCS should invoke the extension service after invoking the application.                             |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList             | true/false. Default = true   | Indicates if EBS call needs to be bypassed. If true, it uses the properties below to identify the service it should invoke. |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.Address     | <code>http://host.port/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCImpl/SyncItemCompositionListSiebelCommsProvABCImpl</code>               | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.PortType    | <code>{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl</code> | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.      |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionList.ServiceName | <code>{http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCImpl/V1}SyncItemCompositionListSiebelCommsProvABCImpl</code> | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property.   |

**Table 24-5 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl Properties**

| Property Name  | Value/Default Values  | Description   |
|--|---|---|
| EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse             | true/false. Default = true  | Not used in the product lifecycle management flow.<br>ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS. |
| EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.Address     | http://host:port/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl               | Not used in the product lifecycle management flow.<br>ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS. |
| EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.PortType    | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl | Not used in the product lifecycle management flow.<br>ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl does not invoke CommunicationsPriceListResponseEBS. |
| EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.ServiceName | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl | Not used in the product lifecycle management flow.<br>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   | The Siebel CRM system instance code defined in BSR. Used if the request message does not contain the target system ID.   |
| Routing.SiebelProductService.SEBL_01.EndpointURI | http://host:port/eai_enu/start.swe?SWExtSource=SecureWebService&SWExtCmd=Execute&WSSOAP=1 | The SOAP end point for the Siebel CRM product import web service. Used if the request message does not contain the target URL.   |
| Routing.SiebelProductService.RouteToCAVS         | true/false. Default = false   | If true, invokes the Siebel CRM end point in the Routing.SiebelProductService.SEBL_01.EndpointURI property for this service.<br>If false, invokes the verification system end point in the system-level SyncResponseSimulator.Soap.EndpointURL property. |

**Table 24-6 (Cont.) SyncItemCompositionListSiebelCommsProvABCSEImpl Properties**

| Property Name   | Value/Default Values  | Description  |
|---|---|--|
| 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 synchronizing products.  |
| Siebel.Product.WorkspaceName  | Demo Workspace  | The name of the Siebel CRM workspace. Create a workspace and update this property before synchronizing products.   |
| 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 synchronized.   |
| ABCSEExtension.PreXFormEBMtoABM   | true/false. Default = false   | Determines whether the ABCS should invoke the Extension service before the EBM to ABM transformation.  |
| ABCSEExtension.PostXFormABMtoEBM  | true/false. Default = false   | Determines whether the ABCS should invoke the Extension service after the ABM to EBM transformation.   |
| ABCSEExtension.PreInvokeABS   | true/false. Default = false   | Determines whether the ABCS should invoke the Extension service before invoking the application.   |
| ABCSEExtension.PostInvokeABS  | true/false. Default = false   | Determines whether the ABCS should invoke the Extension service after invoking the application.  |
| EBSOverride.CommunicationsItemCompositionEBSV1.<br>CommunicationsItemCompositionResponseEBSV1.SyncItemCompositionListResponse | true/false. Default = true  | Indicates if the response EBS call needs to be bypassed.<br><br>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 properties below to identify the service it should invoke to send the response. |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.Address  | http://host:port/soa-infra/services/default/SyncItemCompositionListSiebelCommsProvABCSEImpl/SyncItemCompositionListSiebelCommsProvABCSEImpl                 | The address endpoint URI of the web service to be invoked dynamically.   |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.PortType                                       | {http://xmlns.oracle.com/ABCSEImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSEImpl/V1}SyncItemCompositionListSiebelCommsProvABCSEImpl | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.   |
| EBSOverride.CommunicationsItemCompositionEBSV1.SyncItemCompositionListResponse.ServiceName                                    | {http://xmlns.oracle.com/ABCSEImpl/Siebel/Industry/Comms/SyncItemCompositionListSiebelCommsProvABCSEImpl/V1}SyncItemCompositionListSiebelCommsProvABCSEImpl | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property.  |

## Configuration 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                      | The Design Studio instance code. Used 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 | Determines whether the ABCS invokes the Extension service before invoking the application.   |
| ABCSExtension.PostInvokeCommunicationsClassificationEBS   | true/false. Default = false | Determines whether the ABCS invokes the Extension service after invoking the application.  |
| ABCSExtension.PreInvokeEBSQueryClassificationListEBM  | true/false. Default = false | Determines whether the ABCS invokes the Extension service before invoking the application.   |
| ABCSExtension.PostXFormQueryClassificationListResponseEBMtoProductClassAndAttributesResponseABM | true/false. Default = false | Determines whether the ABCS invokes the Extension service before the EBM to ABM transformation.  |
| ABCSExtension.PreXFormQueryClassificationListResponseEBMtoQuerySpecificationListEBM             | true/false. Default = false | Determines whether the ABCS invokes the Extension service before the EBM to EBM transformation.  |
| ABCSExtension.PreInvokeCommunicationsSpecificationEBS   | true/false. Default = false | Determines whether the ABCS invokes the Extension service before invoking the application.   |
| ABCSExtension.PostInvokeCommunicationsSpecificationEBS  | true/false. Default = false | Determines whether the ABCS invokes the Extension service after invoking the application.  |
| ABCSExtension.PreXformQueryClassificationListResponseEBMtoQuerySpecificationValueSetListEBM     | true/false. Default = false | Determines whether the ABCS invokes the Extension service before the EBM to EBM transformation.  |
| ABCSExtension.PreInvokeCommunicationsSpecificationValueSetEBS                                   | true/false. Default = false | Determines whether the ABCS invokes the Extension service before invoking the application.   |
| ABCSExtension.PostInvokeCommunicationsSpecificationValueSetEBS                                  | true/false. Default = false | Determines whether the ABCS invokes the Extension service after invoking the application.  |

**Table 24-7 (Cont.) QueryProductClassAndAttributesSCECommsReqABCImpl Properties**

| Property Name   | Value/Default Values  | Description   |
|---|---|---|
| EBSOverride.CommunicationsClassificationEBSV1.QueryClassificationList.Address                   | http://host.port/soa-infra/services/default/QueryClassificationListSiebelCommsProvABCImpl/QueryClassificationListSiebelCommsProvABCImpl                             | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsClassificationEBSV1.QueryClassificationList.PortType                  | QueryClassificationListSiebelCommsProvABCImplService  | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.    |
| EBSOverride.CommunicationsClassificationEBSV1.QueryClassificationList.ServiceName               | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QueryClassificationListSiebelCommsProvABCImpl/V1}QueryClassificationListSiebelCommsProvABCImpl               | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property. |
| EBSOverride.CommunicationsSpecificationEBSV1.QuerySpecificationList.Address                     | http://host.port/soa-infra/services/default/QuerySpecificationListSiebelCommsProvABCImpl/QuerySpecificationListSiebelCommsProvABCImpl                               | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsSpecificationEBSV1.QuerySpecificationList.PortType                    | QuerySpecificationListSiebelCommsProvABCImplService   | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.    |
| EBSOverride.CommunicationsSpecificationEBSV1.QuerySpecificationList.ServiceName                 | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QuerySpecificationListSiebelCommsProvABCImpl/V1}QuerySpecificationListSiebelCommsProvABCImpl                 | The service name of the web service to be invoked dynamically. This value should be consistent with the Address.          |
| EBSOverride.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.Address     | http://host.port/soa-infra/services/default/QuerySpecificationValueSetListSiebelCommsProvABCImpl/QuerySpecificationValueSetListSiebelCommsProvABCImpl               | The address endpoint URI of the web service to be invoked dynamically.  |
| EBSOverride.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.PortType    | QuerySpecificationValueSetListSiebelCommsProvABCImplService   | The port type of the web service to be invoked dynamically. This value should be consistent with the Address property.    |
| EBSOverride.CommunicationsSpecificationValueSetEBSV1.QuerySpecificationValueSetList.ServiceName | {http://xmlns.oracle.com/ABCImpl/Siebel/Industry/Comms/QuerySpecificationValueSetListSiebelCommsProvABCImpl/V1}QuerySpecificationValueSetListSiebelCommsProvABCImpl | The service name of the web service to be invoked dynamically. This value should be consistent with the Address property. |

Table 24-8 shows the properties for the QueryClassificationListSiebelCommsProvABCImpl service.



**Table 24-8 QueryClassificationListSiebelCommsProvABCSImpl Properties**

| Property Name  | Value/Default Values  | Description  |
|--|---|--|
| Default.SystemID   | SEBL_01   | The Siebel CRM system instance code defined in BSR. Used 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://host:port/eai_enu/start.swe?<br>SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1 | The SOAP end point URL for the Siebel CRM product class query import web service. Used if the request message does not contain the target URL. |
| ABCSExtension.PreXFormEBMtoABMClassificationListEBM            | true/false. Default = false   | Determines whether the ABCS invokes the Extension service before the EBM to ABM transformation.  |
| ABCSExtension.PostXFormABMtoEBMClassificationListEBM           | true/false. Default = false   | Determines whether the ABCS invokes the Extension service after the ABM to EBM transformation.   |
| ABCSExtension.PreInvokeABSListOfSwiAdminIssClassDefinitionABM  | true/false. Default = false   | Determines whether the ABCS invokes the Extension service (PreInvoke Application).   |
| ABCSExtension.PostInvokeABSListOfSwiAdminIssClassDefinitionABM | true/false. Default = false   | Determines whether the ABCS invokes the Extension service (PostInvoke Application).  |

Table 24-9 shows the properties for the QuerySpecificationListSiebelCommsProvABCSImpl service.

**Table 24-9 QuerySpecificationListSiebelCommsProvABCSImpl Properties**

| Property Name  | Value/Default Values  | Description   |
|--|---|---|
| Default.SystemID   | SEBL_01   | The Siebel CRM system instance code defined in BSR. Used 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://host:port/eai_enu/start.swe?<br>SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1 | The SOAP end point URL for the Siebel CRM query product class attributes import web service. Used if the request message does not contain the target URL. |
| ABCSExtension.PreXFormEBMtoABMSpecificationListEBM         | true/false. Default = false   | Determines whether the ABCS invokes the Extension service before the EBM to ABM transformation.   |
| ABCSExtension.PostXFormABMtoEBMSpecificationListEBM        | true/false. Default = false   | Determines whether the ABCS invokes the Extension service after the ABM to EBM transformation.  |
| ABCSExtension.PreInvokeABSAttributeQueryByExample_InputABM | true/false. Default = false   | Determines whether the ABCS invokes the Extension service (PreInvoke Application).  |

**Table 24-9 (Cont.) QuerySpecificationListSiebelCommsProvABCImpl Properties**

| Property Name   | Value/Default Values        | Description   |
|---|-----------------------------|---|
| ABCSExtension.PostInvokeABSAttributeQueryByExample_InputABM | true/false. Default = false | Determines whether the ABCS invokes the Extension service (PostInvoke Application). |

Table 24-10 shows the properties for the QuerySpecificationValueSetListSiebelCommsProvABCImpl service.

**Table 24-10 QuerySpecificationValueSetListSiebelCommsProvABCImpl Properties**

| Property Name   | Value/Default Values  | Description   |
|---|---|---|
| Default.SystemID  | SEBL_01   | The Siebel CRM system instance code defined in BSR. Used 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://host:port/eai_enu/start.swe?SWEEExtSource=SecureWebService&SWEEExtCmd=Execute&WSSOAP=1 | The SOAP end point URL for the Siebel CRM query product class attributes import web service. Used if the request message does not contain the target URL. |
| ABCSExtension.PreXFormEBMtoABMSpecificationValueSetListEBM  | true/false. Default = false   | Determines whether the ABCS invokes the Extension service before the EBM to ABM transformation.   |
| ABCSExtension.PostXFormABMtoEBMSpecificationValueSetListEBM | true/false. Default = false   | Determines whether the ABCS invokes the Extension service after the ABM to EBM transformation.  |
| ABCSExtension.PreInvokeABSAttributeQueryByExample_InputABM  | true/false. Default = false   | Determines whether the ABCS invokes the Extension service (PreInvoke Application).  |
| ABCSExtension.PostInvokeABSAttributeQueryByExample_InputABM | true/false. Default = false   | Determines whether the ABCS invokes the Extension service (PostInvoke Application).   |

# Configuring the Process Integration for Order Lifecycle Management

This chapter describes how to configure the process integration for order lifecycle management in for the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash pre-built integration (the integration).

## About Configuring the Process Integration for Order Lifecycle Management

This chapter assumes that you have already installed Oracle Communications Billing and Revenue Management (BRM), Siebel CRM, and Oracle AIA and performed any Oracle AIA post-installation tasks described in *Oracle AIA Installation Guide*.

See the Oracle AIA software compatibility matrix available from My Oracle Support for information about required versions and patches.

In addition to basic setup, configuring the process integration for order lifecycle management includes:

- Working with domain value maps (DVMs) and cross-references.
- Handling error notifications.
- Setting Oracle AIA configuration properties.

## 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 order lifecycle management.

**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   |
| SALESORDER_TYPECODE                 | Type Code  |
| STATE                               | State  |
| PROVINCE                            | Province   |
| ADDRESS_COUNTRYID                   | Country Code   |
| CUSTOMERPARTY_TYPECODE              | Account Type Code  |
| ITEM_BILLINGTYPECODE                | Billing Type Code. Examples values include Subscription, Discount, Item, Special Rating, and Sponsorship.  |
| SALESORDER_CHANGEDIND               | Order Changed Indicator. Values are <b>True</b> or <b>False</b> . 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 " <a href="#">Working with the PRICELIST DVM</a> " 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 order lifecycle management cross-references.

**Table 25-2 Order Lifecycle Management Cross-References**

| Cross-Reference Table Name  | COMMON   | SEBL_01                               | BRM_01   | Description   |
|---|--|---------------------------------------|--|---|
| SALESORDER_ID   | SalesOrderIdentification   | Id                                    | N/A  | The Siebel CRM Sales Order ID is cross-referenced.  |
| SALESORDER_LINEID   | SalesOrderLineIdentification   | OrderItem/OrderId                     | N/A  | The Siebel CRM OrderItem/OrderId is mapped to the SalesOrderLine Identification in the EBM.   |
| INSTALLED_PRODUCT_ID  | InstalledProductIdentification   | AssetIntegrationId                    | Product, service, or discount OBJ, or sharing group POID | The Siebel CRM Asset Integration ID is mapped to the product/service/discount OBJ or sharing group POID of the BRM object.  |
| ITEM_ITEMID   | ItemIdentification   | ProductId                             | Product or 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<br>(Populated only when service-level balance groups are enabled) | InstalledProductIdentification<br>(Populated from INSTALLEDPRODUCT_ID) | AssetIntegrationID                    | Balance group POID                                       | 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.<br><br>For nested service bundles, the integration populates only the values for the parent service bundle.<br><br>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. |
| CUSTOMERPARTY_DEFAULTBALANCEGROUPID<br>(Entry for default account-level balance group)                | CustomerPartyAccountIdentification                                     | ServiceAccountId/<br>BillingAccountId | Balance group POID                                       | The default balance group POID is mapped to the common ID of the account.   |
| CUSTOMERPARTY_PAYPROFILEID  | PaymentProfileIdentification   | BillingProfileId                      | /payinfo POID  | The billing profile ID is mapped to the pay info POID from BRM.   |
| CUSTOMERPARTY_BILLPROFILEID   | BillingProfileIdentification   | BillingProfileId                      | /billinfo POID   | The billing profile ID is mapped to the bill info POID from BRM.  |

**Table 25-2 (Cont.) Order Lifecycle Management Cross-References**

| Cross-Reference Table Name | COMMON   | SEBL_01                 | BRM_01                              | Description  |
|----------------------------|--|-------------------------|-------------------------------------|--|
| CUSTOMERPARTY_ADDRESSID    | CustomerParty AccountAddressIdentification   | AddressId               | Address POID                        | The address ID is mapped to the BRM Contact POID.  |
| PROMOTIONGROUP_MEMBER_ID   | The common ID of a promotion group (Populated from INSTALLEDPRODUCT_ID) combined with the common ID of a BRM bundled promotion POID. | Promotion group RowId   | BRM bundled promotion POID          | The Siebel CRM RowId for a promotion group is mapped to the POID of a bundled promotion (but not the components of the bundled promotion).<br>The integration uses this XREF to determine if a bundled promotion is associated with a promotion group membership.<br>A bundled promotion can be associated with multiple promotion group memberships. In this case, the BRM value is included only for the first promotion group membership.<br>A promotion group membership can be associated with multiple bundled promotions. In this case, the BRM value is included for each bundled promotion. |
| COMMUNITYOFFER_ID          | Bundled promotion common ID  | Bundled promotion RowId | POID of the discount sharing group  | The Siebel CRM RowId for a bundled promotion that includes a discount with the Community Offer attribute enabled is mapped to the POID of the corresponding discount sharing group.  |
| COLLECTION_GROUP_OWNER_ID  | Common ID of the owner billing profile   | N/A                     | POID of the owner <b>/billinfo</b>  | The common ID of the billing profile for the owner account on the sales order is mapped to the corresponding <b>/billinfo</b> POID.<br>When creating new collections sharing groups, the integration checks if the owner account's <b>/billinfo</b> POID already exists in this table. If so, the member is added to the existing group.   |
| COLLECTION_GROUP_MEMBER_ID | Common ID of the member billing profile  | N/A                     | POID of the member <b>/billinfo</b> | The common ID of the billing profile for the member account on the sales order is mapped to the corresponding <b>/billinfo</b> 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 ProcessFulfillmentOrderBillingBRMComms 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> .                |
| AIA_ERR_AIACOMOMPI_0002 | Date Validation Failed: Purchase Date should be set to the future.   | In Billing Initiation mode, the ProcessFulfillmentOrderBillingBRMComms AddSubProcess 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.  | ProcessFulfillmentOrderBillingBRMCommsPro vABCImpl 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. | ProcessFulfillmentOrderBillingBRMCommsPro vABCImpl 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.

## About Error Notification Roles and Users

The integration uses the following default roles and users to issue error notifications:

- **Role:** AIAIntegrationAdmin
- **User:** AIAIntegrationAdminUser

See the discussion of configuring Oracle AIA processes for error handling and trace logging in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension* for more information about configuring error notifications.

## Configuring the Process Integration for Order Lifecycle Management

This section provides instructions for setting the Oracle AIA configuration properties and setting the BRM version number for backward compatibility.



## Setting Oracle AIA Configuration Properties

Configure the properties described in this section in the *Oracle\_home/comms\_home/source/soainfra/apps/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.

## System-Level Configuration Properties

Table 25-4 shows the system-level configuration properties for order lifecycle management.

**Table 25-4 System Configuration Properties for Order Lifecycle Management**

| Property Name                | Value/Default Values                | Description  |
|------------------------------|-------------------------------------|--|
| O2C.AccountLevelBalanceGroup | TRUE/FALSE<br>Default value = FALSE | A value of <b>FALSE</b> enables service-level balance groups. A value of TRUE disables service-level balance groups and tracks all services in one account-level balance group in BRM. |

## Configuration Properties for Siebel CRM Services

Table 25-5 shows the properties for the UpdateSalesOrderSiebelCommsProvABCSImpl service.

**Table 25-5 UpdateSalesOrderSiebelCommsProvABCSImpl Properties**

| Property Name                                       | Value/Default Values   | Description   |
|---|--|---|
| Default.SystemID                                    | SEBL_01  | URL for Siebel CRM Instance web service for "Order spcLine spcltem spcUpdate spc_spcComplex" web service.         |
| Routing.SWI_spcOrder_spcUpsert.RouteToCAVS          | true/false. Default = false.   | Controls whether UpdateSalesOrderSiebelCommsProvABCSImpl routes messages to the CAVS or to the Siebel CRM system. |
| Routing.SWI_spcOrder_spcUpsert.SE BL_01.EndpointURI | Target Endpoint URL for the Siebel CRM upsert web service.<br>example:<br>http://{siebel.http.host}:{siebel.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1StatusOpenFixedClosed | Target Endpoint URL for the Siebel CRM upsert web service.  |

**Table 25-5 (Cont.) UpdateSalesOrderSiebelCommsProvABCImpl Properties**

| Property Name                  | Value/Default Values         | Description  |
|--------------------------------|------------------------------|--|
| 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 25-6 shows the properties for the ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl service.

**Table 25-6 ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl Properties**

| Property Name  | Value/Default Values  | Description   |
|--|---|---|
| Default.SystemID   | Possible value: A Siebel CRM system instance code.<br>Default value: <b>SEBL_01</b>   | Default Siebel CRM system instance code (defined in BSR). This is used only if the Siebel CRM Order message does not contain the EnterpriseServerName.  |
| ABCSExtension.PreXformABMtoEBM   | Possible values: <b>true</b> or <b>false</b><br>Default value: <b>false</b>   | Indicates whether there is any extension in the ABCS before transformation of EBM to ABM.   |
| ABCSExtension.PostXformABMtoEBM  | true/false. Default = false   | Indicates whether there is any extension in the ABCS after transformation of EBM to ABM.  |
| ABCSExtension.PreInvokeEBS   | Possible values: <b>true</b> or <b>false</b><br>Default value: <b>false</b>   | Indicates whether there is any extension in the ABCS before invoking application business service.  |
| ABCSExtension.PostInvokeEBS  | Possible values: <b>true</b> or <b>false</b><br>Default value: <b>false</b>   | Indicates whether there is any extension in the ABCS after invoking application business service.   |
| EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address     | Possible value: address of web service that must be invoked.<br>Default value: <b>http://&lt;soa_server&gt;:&lt;soa_port&gt;/soa-infra/services/default/ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer/client</b>        | 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    | Possible value: PortType of the web service that must be invoked.<br>Default value: <b>ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer</b>  | This value should be in consistent with EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address.   |
| EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.ServiceName | Possible value: ServiceName of the web service that must be invoked.<br>Default value: <b>http://xmlns.oracle.com/ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer}ProcessSalesOrderFulfillmentOSMCFSCCommsJMSProducer</b> | This value should be in consistent with EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address.   |

**Table 25-6 (Cont.) ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl Properties**

| Property Name                     | Value/Default Values  | Description  |
|-----------------------------------|---|--|
| Default.handleLargeOrderEnabled   | Possible values: <b>true</b> or <b>false</b><br>Default value: <b>false</b> | Indicates whether ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl uses the Process Large Order sub-flow to split large orders and decrease the time it takes to transform ABMs for large orders.   |
| Default.numOrderLinesInLargeOrder | Possible values: any positive integer<br>Default value: <b>1000</b>         | When handleLargeOrderEnabled is set to <b>true</b> , and an order has a number of order lines greater than or equal to the value of this property, ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl uses the Process Large Order sub-flow.  |
| Default.numOrderLinesInMiniABM    | Possible values: any positive integer<br>Default value: <b>150</b>          | When using the Process Large Order sub-flow, ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl splits the sales order ABM into mini ABMs with a number of order lines equal to the value of this property.<br><br><b>Note:</b> Because ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl keeps order lines that are part of the same promotions in the same mini ABM, the number of order lines in a mini ABM may sometimes exceed the value of this property. |

Table 25-7 shows the properties for the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl service.

**Table 25-7 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.  |
| EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.PortType | PortType of the web service that must be invoked.<br>Example (Default):<br>ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImplService | 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. |

**Table 25-7 (Cont.) ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl Properties**

| Property Name   | Value/Default Values  | Description   |
|---|---|---|
| EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.ServiceName | ServiceName of the web service that must be invoked.<br>Example (Default): {http://xmlns.oracle.com/ABCServiceImpl/Industry/Comms/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl/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.<br>Example (Default):<br><br>http://host:port/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.     |

## Configuration Properties for BRM Services

Table 25-8 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess service.

**Table 25-8 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).<br><br>This property is billing-instance-specific and must be set for any instance that the order must be sent for billing integration.<br><br>See <a href="#">"Using Single-Phase Billing or Two-Phase Billing"</a> 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.   |

**Table 25-8 (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 before calling BRM opcode PCM_OP_SUBSCRIPTION_PURCHASE_DEAL. |
| 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 25-9 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess service.

**Table 25-9 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 25-9 (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 25-10 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess service.

**Table 25-10 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 25-11 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsProvABCImpl service.

**Table 25-11 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.   |

**Table 25-11 (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl Properties**

| Property Name   | Value/Default Values                            | Description   |
|---|---|---|
| Routing.BRMSUBSCRIPTIONService.BRM_01.EndpointURI     | End point for BRM Adapter.<br>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.BRMSUBSCRIPTIONService.BRM_02.EndpointURI     | End point for BRM Adapter.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.   |
| 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.                                |

**Table 25-11 (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCImpl Properties**

| Property Name   | Value/Default Values  | Description   |
|---|---|---|
| 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.<br>Example (Default):<br><pre>{http://host:port/soa-infra/services/default/ProcessFulfillmentOrderBillingResponseOSMCFSCCommsJMSProducer/client</pre>                          | 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.<br>Example (Default):<br>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.<br>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. |

Table 25-12 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess service.



**Table 25-12 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.                          |
| 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.                           |

**Table 25-12 (Cont.) ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess Properties**

| Property Name  | Value/Default Values        | Description  |
|--|-----------------------------|--|
| 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 25-13 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess service.

**Table 25-13 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 25-13 (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 25-14 shows the properties for the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl service.

**Table 25-14 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCImpl 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://host:port/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.

# 26

## Configuring the Process Integration for Customer Management

This chapter describes how to configure the process integration for customer management in the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash pre-built integration (the integration).

### About Configuring the Process Integration for Customer Management

Configuring the process integration for customer management includes:

- Working with domain value maps (DVMs) and cross-references.
- Handling error notifications.
- Setting Oracle AIA configuration properties.

This chapter assumes that you have already installed Oracle Communications Billing and Revenue Management (BRM) and Siebel CRM. See the Oracle AIA software compatibility matrix available from My Oracle Support for information about required versions and patches.

### 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_ACCOUNTTYPECOD<br>E.dvm | SEBL_01, COMMON,<br>BRM_01 | Used to get the type of the account, such as <b>Business</b> or <b>Residential</b> . |
| PROVINCE.dvm                          | SEBL_01, COMMON,<br>BRM_01 | Province name.   |
| STATE.dvm                             | SEBL_01, COMMON,<br>BRM_01 | State name.  |
| ADDRESS_COUNTRYID.dvm                 | SEBL_01, COMMON,<br>BRM_01 | Country codes.   |

**Table 26-1 (Cont.) Customer Management DVMs**

| DVM  | Columns                 | Description   |
|--|-------------------------|---|
| 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_BILLYPECODE.dvm      | SEBL_01, COMMON, BRM_01 | Bill type ( <b>summary</b> and <b>detailed</b> ).   |
| CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE.dvm    | SEBL_01, COMMON, BRM_01 | Billing frequency ( <b>monthly, yearly, quarterly</b> , and so on.)   |
| CUSTOMERPARTY_PAYPROFILE_BANKACCOUNTTYPE.dvm   | SEBL_01, COMMON, BRM_01 | Bank account type ( <b>checking, savings</b> , and so on).  |
| CUSTOMERPARTY_PAY_PROFILE_CREDIT_CARDTYPE.dvm  | SEBL_01, COMMON         | Credit Card type ( <b>Visa, Mastercard</b> , and so on.)  |
| CUSTOMERPARTY_PAYPROFILE_DELIVERYRYPREF.dvm    | COMMON, BRM_01          | Bill media/delivery preference ( <b>Email</b> or <b>Mail</b> ).   |
| CUSTOMERPARTY_PAYPROFILE_PAYMENTMETHODCODE.dvm | SEBL_01, COMMON, BRM_01 | Payment profile payment method types ( <b>credit card, direct debit</b> , and <b>invoice/bill me</b> .)   |
| CUSTOMERPARTY_PAYPROFILE_PAYMENTTERMCODE.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 ( <b>home, work, mobile, fax</b> , and so on.)  |

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  | COMMON                                | SEBL_01         | BRM_01         |
|-----------------------------|---------------------------------------|-----------------|----------------|
| CUSTOMERPARTY_ACCOUNTID     | Common GUID (generated by Oracle AIA) | Account ID      | Account POID   |
| CUSTOMERPARTY_BILLPROFILEID | Bill Profile ID                       | Bill Profile ID | Bill-info POID |

**Table 26-2 (Cont.) Customer Management Cross-References**

| Cross-Reference Table Name              | COMMON                                     | SEBL_01                                 | BRM_01   |
|---|--|---|--|
| CUSTOMERPARTY_PAYPROFILEID              | Payment Profile ID                         | Bill Profile ID                         | Pay-info POID  |
| CUSTOMERPARTY_ADDRESSID                 | Address ID                                 | Address ID                              | Account POID (if this is the billing address)<br>Pay-info POID (if this is the pay-info address) |
| CUSTOMERPARTY_CONTACTID                 | Contact ID                                 | Contact ID                              | Account POID (if this is the main contact)<br>Pay-info POID (if this is the pay-info contact)    |
| CUSTOMERPARTY_DEFAULTBALANCEGROU<br>PID | Account ID                                 | Not applicable                          | Balance Group POID   |
| CUSTOMERPARTY_PARTYID                   | Customer Party ID                          | Customer Party ID                       | Not applicable   |
| CUSTOMERPARTY_PARTYLOCATIONID.xref      | Customer Party<br>Location ID              | Customer Party<br>Location ID           | Not applicable   |
| CUSTOMERPARTY_LOCATIONREFID             | Customer Party<br>Location<br>Reference ID | Customer Party<br>Location Reference ID | Not applicable   |
| CUSTOMERPARTY_ACCOUNT_PHONECOMM<br>ID   | Customer<br>Account's phone<br>number      | Customer Account's<br>phone number      | Not applicable   |
| CUSTOMERPARTY_ACCOUNT_FAXCOMMID         | Customer<br>Account's fax<br>number        | Customer Account's<br>fax number        | Not applicable   |
| CUSTOMERPARTY_ACCOUNT_WEBCOMMID         | Customer<br>Account's email<br>address     | Customer Account's<br>email address     | Not applicable   |
| CUSTOMERPARTY_CONTACT_PHONECOMM<br>ID   | Customer<br>Contact's phone<br>number      | Customer Contact's<br>phone number      | Not applicable   |
| CUSTOMERPARTY_CONTACT_FAXCOMMID         | Customer<br>Contact's fax<br>number        | Customer Contact's<br>fax number        | Not applicable   |
| CUSTOMERPARTY_CONTACT_EMAILCOMMID       | Customer<br>Contact's email<br>address     | Customer Contact's<br>email address     | Not applicable   |

## 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  |
|--|-------------------------|---|
| SyncCustomerPartyListBRMCommsProvABCImpl | AIA_ERR_AIACOMCMPI_0004 | Subordinate account cannot have multiple parent accounts.   |
| SyncCustomerPartyListBRMCommsProvABCImpl | 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. |
| SyncCustomerPartyListBRMCommsProvABCImpl | 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.  |

## About Error Notification Roles and Users

The integration uses the following default roles and users to issue error notifications:

- **Role:** AIAIntegrationAdmin
- **User:** AIAIntegrationAdminUser

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.

## 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 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 *Comms\_home/source/soainfra/apps/AIAMetaData/config/AIACConfigurationProperties.xml* file.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about working with *AIACConfigurationProperties.xml*.



### Note:

Entries in the *AIACConfigurationProperties.xml* file are case sensitive.

## System-Level Configuration Properties

Table 26-4 shows the system-level properties for the process integration for customer management.

**Table 26-4 Customer Management System-level Properties**

| Property Name                     | Value/Default Value                                 | Description  |
|-----------------------------------|---|--|
| O2C.LegalGroup                    | TRUE/FALSE.<br>Default: FALSE                       | Determines whether the integration creates collections sharing groups in BRM.<br><br>If set to <b>TRUE</b> , the integration creates collections sharing groups when the owner account and billing account on an order line are different.<br><br>If set to <b>FALSE</b> , the integration ignores the owner account and does not create collections sharing groups. |
| O2C.CorporateHierarchyAccountType | BUSINESS/RESIDENTIAL/no value.<br>Default: BUSINESS | Determines the account type for which the integration synchronizes the linear account hierarchy.<br><br>To disable the synchronization of account hierarchies, do not provide a value.   |

## Configuration Properties for BRM Services

Table 26-5 shows the properties for the *ProcessCollectionSharingBRMCommsProvABCServiceImpl* service.

**Table 26-5 ProcessCollectionSharingBRMCommsProvABCServiceImpl Properties**

| Property Name                                    | Value/Default Value                        | Description   |
|--|--|---|
| Default.SystemID                                 | BRM_01                                     | Specifies the BRM instance code. It is used if the request message does not contain the target system ID. |
| Routing.BRMCOLLECTIONSService.RouteToCAVS        | <b>true/false</b><br>Default: <b>false</b> | Specifies whether the end point routes to CAVS.   |
| Routing.BRMCOLLECTIONSService.BRM_01.EndpointURI | Default: eis/BRM                           | Specifies the end point URL for the CAVS.   |

**Table 26-5 (Cont.) ProcessCollectionSharingBRMCommsProvABCSEImpl Properties**

| Property Name                                | Value/Default Value          | Description  |
|--|------------------------------|--|
| ABCSExtension.ABCSExtension.PreXformABMtoEBM | true/false<br>Default: false | Indicates whether the ABCS has been extended before the EBM to ABM transformation. |
| ABCSExtension.PostXformEBMtoABM              | true/false<br>Default: false | Indicates whether the ABCS has been extended after the EBM to ABM transformation.  |

Table 26-6 shows the properties for the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSEImpl service.

**Table 26-6 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSEImpl Properties**

| Property Name                             | Value/Default Value                        | Description  |
|---|--|--|
| Default.SystemID                          | BRM_01                                     | Specifies the BRM instance code that is used if the request message does not contain the target system ID. |
| ABCSExtension.PreXFormEBMtoABM            | true/false.<br>Default: false              | Indicates whether the ABCS has been extended before the EBM to ABM transformation.                         |
| Routing.BRMCUSTSService.RouteToCAVS       | <b>true/false</b><br>Default: <b>false</b> | Specifies whether the end point routes to CAVS.  |
| Routing.BRMCUSTService.BRM_01.EndpointURI | Default: eis/BRM                           | Specifies the end point URL for the CAVS.  |
| ABCSExtension.PreInvokeABS                | true/false<br>Default: false               | Indicates whether the ABCS invokes the extension service before invoking the application.                  |

Table 26-7 shows the properties for the SyncCustomerPartyListBRMCommsProvABCSEImpl service.

**Table 26-7 SyncCustomerPartyListBRMCommsProvABCSEImpl Properties**

| Property Name  | Value/Default Value  | Description   |
|--|--|---|
| Default.SystemID   | BRM_01   | Specifies the BRM instance code that is used if the request message does not contain the target system ID.            |
| EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.PortType    | CommsProcessBillingAccountListEBF  | The port type of the web service to invoke dynamically. This value should be consistent with the Address property.    |
| EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.ServiceName | {http://xmlns.oracle.com/EnterpriseFlows/Industry/Comms/CommsProcessBillingAccountListEBF/V1}CommsProcessBillingAccountListEBF | The service name of the web service to invoke dynamically. This value should be consistent with the Address property. |

Table 26-7 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties

| Property Name  | Value/Default Value  | Description  |
|--|--|--|
| EBSOverride.CommunicationsCustomerPartyEBSV2.SyncCustomerPartyListResponse.Address | http://SOA_host.port/soa-infra/services/default/CommsProcessBillingAccountListEBF/client | The address endpoint URI of a web service, such as CAVS or another ABCS, to dynamically invoke from this service.  |
| EnableAccountStatusSync  | true/false.<br>Default: false  | Indicates whether the integration synchronizes account status.<br><br>When set to <b>true</b> , the integration synchronizes account status from Siebel CRM to BRM.  |
| Routing.BRMCUSTService.BRM_01.EndpointURI  | Default: eis/BRM   | Specifies the connection factory to connect to the BRM JCA adapter for the first of multiple instances of BRM for the CUST opcodes.<br><br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems. |
| Routing.BRMCUSTService.RouteToCAVS   | true/false.<br>Default: false  | Specifies whether the end point routes to CAVS for the CUST opcodes.   |
| Routing.BRMCUSTService_ptt.BRM_01.EndpointURI                                      | Default: eis/BRM   | Specifies the connection factory to connect to the BRM JCA adapter for the first of multiple instances of BRM for the CUST opcodes.<br><br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems. |
| AccountLevelBalanceGroupName   | Account Level Balance Group  | Names the default balance group created in BRM when an account is created.   |
| ABCSExtension.PreXFormEBMtoABM   | true/false.<br>Default: false  | Indicates whether the ABCS has been extended before the EBM to ABM transformation.   |
| ABCSExtension.PreInvokePCM_OP_BILL_GROUP_GET_PARENTABM                             | true/false.<br>Default: false  | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.   |
| ABCSExtension.PostInvokePCM_OP_BILL_GROUP_GET_PARENTABM                            | true/false.<br>Default: false  | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.  |
| ABCSExtension.PreInvokePCM_OP_SEARCHABM  | true/false.<br>Default: false  | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.   |
| ABCSExtension.PostInvokePCM_OP_SEARCHABM   | true/false.<br>Default: false  | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.  |
| ABCSExtension.PreInvokeABSPCM_OP_CUST_COMMIT_CUSTOMERABM                           | true/false.<br>Default: false  | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.   |
| ABCSExtension.PostInvokeABSPCM_OP_CUST_COMMIT_CUSTOMERABM                          | true/false.<br>Default: false  | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.  |
| ABCSExtension.PreInvokePCM_OP_CUSTCARE_MOVE_ACCTABM                                | true/false.<br>Default: false  | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.   |

**Table 26-7 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties**

| Property Name  | Value/Default Value           | Description   |
|--|-------------------------------|---|
| ABCSExtension.PostInvokePCM_OP_CUSTCARE_MOVE_ACCTABM   | true/false.<br>Default: false | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.   |
| ABCSExtension.PreInvokePCM_OP_CUST_UPDATE_CUSTOMERABM  | true/false.<br>Default: false | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.  |
| ABCSExtension.PostInvokePCM_OP_CUST_UPDATE_CUSTOMERABM | true/false.<br>Default: false | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.   |
| ABCSExtension.PreInvokePCM_OP_CUST_DELETE_PAYINFOABM   | true/false.<br>Default: false | Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.  |
| ABCSExtension.PostInvokePCM_OP_CUST_DELETE_PAYINFOABM  | true/false.<br>Default: false | Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.   |
| ABCSExtension.PostXFormABMtoEBM                        | true/false.<br>Default: false | Indicates whether the ABCS has been extended after the ABM to EBM transformation..  |
| Routing.BRMCUSTService_ptt.RouteToCAVS                 | true/false.<br>Default: false | Specifies whether to route to CAVS for the CUST opcodes.  |
| Routing.BRMCUSTCAREService.BRM_01.EndpointURI          | Default: eis/BRM              | Specifies the Connection factory to connect to the BRM JCA adapter for the first of multiple BRM instances for the CUSTCARE opcodes.<br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems. |
| Routing.BRMCUSTCAREService.RouteToCAVS                 | true/false.<br>Default: false | Specifies whether to route to CAVS for the CUSTCARE opcodes.  |
| Routing.BRMBILLService.BRM_01.EndpointURI              | Default: eis/BRM              | Specifies the Connection factory to connect to the BRM JCA adapter for the first of multiple BRM instances for the BILL opcodes.<br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems.     |
| Routing.BRMBILLService.RouteToCAVS                     | true/false.<br>Default: false | Specifies whether to route to CAVS for BILL opcodes.  |
| Routing.BRMBASEService.BRM_01.EndpointURI              | Default: eis/BRM              | Specifies the Connection factory to connect to the BRM JCA adapter for the first of multiple BRM instances for the BASE opcodes.<br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems.     |
| Routing.BRMBASEService.RouteToCAVS                     | true/false.<br>Default: false | Specifies whether to route to CAVS for BASE opcodes.  |

**Table 26-7 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties**

| Property Name                            | Value/Default Value           | Description  |
|--|-------------------------------|--|
| Routing.BRMTXNService.BRM_01.EndpointURI | Default: eis/BRM              | Specifies the Connection factory to connect to the BRM JCA adapter for the first of multiple BRM instances for the TXN opcodes.<br>See " <a href="#">Configuring Multiple BRM Instances for Communications Integrations</a> " for more information about multiple BRM systems. |
| Routing.BRMTXNService.RouteToCAVS        | true/false.<br>Default: false | Specifies whether to route to CAVS for TXN opcodes.  |

## Configuration Properties for Siebel CRM Services

Table 26-8 shows the properties for the SyncAccountSiebelReqABCSImpl service.

**Table 26-8 SyncAccountSiebelReqABCSImpl Properties**

| Property Name   | Value/Default Values  | Description   |
|---|---|---|
| Default.SystemID  | SEBL_01   | Specifies the Siebel CRM instance code that is used if the request message does not contain the target system ID. |
| ABCSExtension.PreXformABMtoEBMABM   | true/false.<br>Default: false   | Indicates whether the ABCS has been extended before the ABM to EBM transformation.                                |
| ABCSExtension.PreInvokeEBSEBM   | true/false.<br>Default: false.  | Indicates whether the ABCS has been extended before the EBS invocation.   |
| Routing.SWICustomerParty.SEBL_01.EndpointURI  | <code>http://Siebel_host:port/eai_Siebel_language/start.swe?SWEExtSource=SecureWebService&amp;SWEExtCmd=Execute&amp;WS SOAP=1</code>          | Endpoint URI of the SEBL_01 Siebel instance.  |
| Routing.SWICustomerParty.RouteToCAVS  | true/false.<br>Default: false.  | Specifies whether the end point routes to CAVS.   |
| Routing.SWICustomerParty.CAVS.EndpointURI   | <code>http://SOA_host:port/AIAValidationSystemServlet/syncreponsesimulator</code>   | Specifies the end point URL for the CAVS service.   |
| Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode | PRODUCTION  | --  |
| RoutingCustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI                              | <code>http://\${fp.server.soaserverhostname}:<br/>{fp.server.soaserverport}/<br/>AIAValidationSystemServlet/<br/>asyncrequestrecipient</code> | Specifies whether the end point routes to CAVS.   |
| Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS                                  | true/false.<br>Default: false.  | Specifies whether the end point routes to CAVS.   |
| Routing.TransformAppContextSiebelService.RouteToCAVS  | true/false.<br>Default: false.  | Specifies whether the end point routes to CAVS.   |

**Table 26-8 (Cont.) SyncAccountSiebelReqABCSEImpl Properties**

| Property Name   | Value/Default Values  | Description  |
|---|---|--|
| Routing.TransformAppContextSiebelService.CAVS.EndpointURI | http://\$<br>{fp.server.soaserverhostname}:<br>{fp.server.soaserverport}/<br>AIAValidationSystemServlet/<br>asyncrequestrecipient | Specifies whether the end point routes to CAVS.  |
| Account.ProcessUpdateEventsOnly                           | true/false.<br>Default: true.   | To optimize the integration flow, leave this property set to <b>true</b> .<br><br>This prevents the Siebel CRM ABCS from propagating create events. By default the integration supports creation of customers only as part of the order flow.<br><br>Setting this property to <b>false</b> results in a less optimized flow, and the default integration behavior remains the same.<br><br><i>See Oracle Application Integration Architecture Pre-Built Integrations Functional Interoperability Configuration Guide for more information.</i> |
| Contact.QueryAllEntities                                  | true/false.<br>Default: false.  | --   |
| Address.QueryAllEntities                                  | true/false.<br>Default: false.  | --   |

Table 26-9 shows the properties for the ProcessAccountHierarchyListSiebelCommsProvABCSEImpl service.

**Table 26-9 ProcessAccountHierarchyListSiebelCommsProvABCSEImpl Properties**

| Property Name   | Value/Default Value  | Description   |
|---|--|---|
| Default.SystemID  | SEBL_01  | Specifies the Siebel CRM instance code that is used if the request message does not contain the target system ID. |
| ABCSExtension.PreXformEBMtoABM                          | true/false<br>Default: false   | Indicates whether the ABCS has been extended before the EBM to ABM transformation.                                |
| ABCSExtension.PostXformEBMtoABM                         | true/false<br>Default: false   | Indicates whether the ABCS has been extended after the EBM to ABM transformation.                                 |
| Routing.SWI_spcAccount_spcHierarchy.RouteToCAVS         | true/false<br>Default: false   | Specifies whether the end point routes to CAVS.   |
| Routing.SWI_spcAccount_spcHierarchy.CAVS.EndpointURI    | http://SOA_server.port/<br>AIAValidationSystemServlet/<br>syncresponsesimulator.   | Specifies the CAVS endpoint URI for the Siebel CRM adapter.   |
| Routing.SWI_spcAccount_spcHierarchy.SEBL_01.EndpointURI | http://Siebel_server.port/<br>eai_Siebel_language/start.swe?<br>SWEExtSource=SecureWebService&SWE<br>ExtCmd=Execute&WSSOAP=1 | Specifies the Siebel CRM instance endpoint URI.   |

**Table 26-9 (Cont.) ProcessAccountHierarchyListSiebelCommsProvABCImpl Properties**

| Property Name  | Value/Default Value          | Description   |
|--|------------------------------|---|
| Routing.SWI_spcAccount_spcHierarchy.MessageProcessingInstruction.EnvironmentCode | PRODUCTION                   | Specifies the type of environment for which to route to CAVS.                             |
| ABCSExtension.PreInvokeABS   | true/false<br>Default: false | Indicates whether the ABCS invokes the extension service before invoking the application. |
| ABCSExtension.PostInvokeABS  | true/false<br>Default: false | Indicates whether the ABCS invokes the extension service after invoking the application.  |

Table 26-10 shows the properties for the QueryCustomerPartyListSiebelProvABCImplV2 service.

**Table 26-10 QueryCustomerPartyListSiebelProvABCImplV2 Properties**

| Property Name  | Value/Default Value  | Description   |
|--|--|---|
| Default.SystemID   | SEBL_01  | Specifies the Siebel CRM instance code that is used if the request message does not contain the target system ID. |
| ABCSExtension.PreXformEBMtoABMEBM  | true/false<br>Default: false   | Indicates whether the ABCS has been extended before the EBM to ABM transformation.                                |
| ABCSExtension.PostXformABMtoEBMEBM   | true/false<br>Default: false   | Indicates whether the ABCS has been extended after the EBM to ABM transformation.                                 |
| Routing.SWI_spcCustomer_spcParty_spcService.RouteToCAVS                          | true/false<br>Default: false   | Specifies whether the end point routes to CAVS.   |
| Routing.SWI_spcCustomer_spcParty_spcService.CAVS.EndpointURI                     | http://SOA_server:port/AIAValidationSystemServlet/syncresponsesimulator.   | Specifies the CAVS endpoint URI for the Siebel CRM adapter.   |
| Routing.SWI_spcCustomer_spcParty_spcService.SEBL_01.EndpointURI                  | http://Siebel_server:port/eai_Siebel_language/start.swe?SWEEExtSource=SecureWebService&SWEEExtCmd=Execute&WSSOAP=1 | Specifies the Siebel CRM instance endpoint URI.   |
| Routing.SWI_spcAccount_spcHierarchy.MessageProcessingInstruction.EnvironmentCode | PRODUCTION   | Specifies the type of environment for which to route to CAVS.   |
| ABCSExtension.PreInvokeSWICustServiceABM   | true/false<br>Default: false   | Indicates whether the ABCS invokes the extension service before invoking the Siebel CRM web service.              |
| ABCSExtension.PostInvokeSWICustServiceABM  | true/false<br>Default: false   | Indicates whether the ABCS invokes the extension service after invoking the Siebel CRM web service.               |

Table 26-11 shows the properties for the ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCImpl service.

**Table 26-11 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCServiceImpl Properties**

| Property Name   | Value/Default Value  | Description   |
|---|--|---|
| Default.SystemID  | SEBL_01  | Specifies the Siebel CRM instance code that is used if the request message does not contain the target system ID.                   |
| ABCSExtension.PreXformABMtoEBM  | true/false<br>Default: false   | Indicates whether the ABCS has been extended before the ABM to EBM transformation.  |
| EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.PortType    | ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImplService   | The port type of the web service that must be invoked dynamically. This value should be consistent with the Address property.       |
| EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.ServiceName | {http://xmlns.oracle.com/ABCServiceImpl/BRM/Industry/Comms/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl/V1}ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImplService | The service name of the web service that must be invoked dynamically. This value should be in consistent with the Address property. |
| EBSOverride.CommunicationsInstalledProductEBSV2.ProcessInstalledProductSpecialRatingSetList.Address     | http://SOA_host:port/soa-infra/services/default/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl/ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCServiceImpl                      | The address endpoint URI of a web service, such as CAVS or another ABCS, to dynamically invoke a web service from this service.     |
| ABCSExtension.PreInvokeEBS  | true/false<br>Default: false   | Indicates whether the ABCS should invoke the extension service before invoking the EBS.   |



# Configuring the Process Integration for Order Fallout Management

This chapter describes how to configure the process integration for order fallout management in the Oracle Application Integration Architecture (Oracle AIA) Oracle Communications Order to Cash pre-built integration (the integration).

## About Configuring the Process Integration for Order Fallout Management

Configuring the process integration for order fallout management includes:

- Configuring 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

This chapter assumes that you have already installed Siebel CRM. See the Oracle AIA software compatibility matrix available from My Oracle Support for information about required versions and patches.

## Configuring Oracle AIA for Order Fallout Management

To configure Oracle AIA for order fallout management:

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 order fallout management, 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 any additional values needed for your environment.

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.<br>SEBL_01 column maps to the Area element in Siebel CRM.<br>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.<br>SEBL_01 column maps to the Sub-Area element in Siebel CRM.<br>COMMON column points to the appropriate FailureSubSystemCode or the faulting service.       |
| TROUBLETICKET_STATUS   | DVM to map the status of the trouble ticket.<br>SEBL_01 column maps to the Status element in Siebel CRM.<br>COMMON column maps to the appropriate status in Oracle AIA.                                    |
| TROUBLETICKET_SEVERITY | DVM to map the severity of the trouble ticket.<br>SEBL_01 column maps to the Severity element in Siebel CRM.<br>COMMON column maps to the appropriate severity (1-5) in Oracle AIA.                        |
| TROUBLETICKET_PRIORITY | DVM to map the recovery priority of the trouble ticket.<br>SEBL_01 column maps to the Priority element in Siebel CRM.<br>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<br>COMMON   | Column Value<br>SEBL_01  | Description  |
|-------------------------------|---|--|--|
| TROUBLETICKET_TROUBLETICKETID | CreateTroubleTicketResponseEBM/DataArea/CreateTroubleTicketResponse/Identification/BusinessComponentID stores this value.<br><br>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).<br><br>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.

## About Error Notification Roles and Users

The integration uses the following default roles and users to issue error notifications:

- **Role:** AIAIntegrationAdmin
- **User:** AIAIntegrationAdminUser

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 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 |
|--|--|-------------------------------------|--------------------|
| --   | ProcessFullmentOrderBillingBRMCommsAddSubProcesses | AIA_EH_DEFAULT                      | AIACOM_OFM_EXT     |
| {http://schemas.oracle.com/bpel/extension}bindingFault | ProcessFullmentOrderBillingBRMCommsAddSubProcesses | 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.

You can perform a bulk update of the error types for all of the processes by using 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 `Oracle_home/comms_home/source/soainfra/apps/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/ABCServiceImpl/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.soaserverhostname}:\${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketSiebelCommsProvABCServiceImpl/CreateTroubleTicketSiebelCommsProvABCServiceImpl | Address of the provider ABCS/EBS service CreateTroubleTicket operation.  |
| EBSOverride.TroubleTicketEBS.CreateTroubleTicket.ServiceName | {http://xmlns.oracle.com/ABCServiceImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCServiceImpl/V1}CreateTroubleTicketSiebelCommsProvABCServiceImplService                       | Service Name of the provider ABCS/EBS service of the CreateTroubleTicket operation.  |
| TroubleTicket.DefaultSeverity                                | Any number from 1 to 5.<br>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. |

**Table 27-5 (Cont.) CreateTroubleTicketAIACommsReqImpl Properties**

| Property Name                 | Value/Default Values                    | Description   |
|-------------------------------|---|---|
| TroubleTicket.DefaultPriority | Any number from 1 to 4.<br>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 CreateTroubleTicketSiebelCommsProvABCServiceImpl service name.

**Table 27-6 CreateTroubleTicketSiebelCommsProvABCServiceImpl 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.PreXformEBMt oABMTroubleTicketEBM                                | true/false<br>Default: false   | Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).  |
| ABCSExtension.PostXformABMt oEBMTroubleTicketEBM                               | true/false<br>Default: false   | Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation).   |
| ABCSExtension.PreInvokeABS WITroubleTicketIOABM                                | true/false<br>Default: false   | Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).   |
| ABCSExtension.PostInvokeABS SWITroubleTicketIOABM                              | true/false<br>Default: false   | Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).  |
| EBSOverride.TroubleTicketEBSR esponse.CreateTroubleTicketRes ponse.PortType    | {http://xmlns.oracle.com/Industry/ Comms/ CreateTroubleTicketRespOSMCFSCo mmsJMSProducer/ V1}CreateTroubleTicketRespOSMCFSC ommsJMSProducer      | Port Type of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.   |
| EBSOverride.TroubleTicketEBSR esponse.CreateTroubleTicketRes ponse.Address     | http://\$ {fp.server.soaserverhostname};\$ {fp.server.soaserverport}/soa-infra/ services/default/ CreateTroubleTicketRespOSMCFSCo mmsJMSProducer | Address of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.   |
| EBSOverride.TroubleTicketEBSR esponse.CreateTroubleTicketRes ponse.ServiceName | {http://xmlns.oracle.com/Industry/ Comms/ CreateTroubleTicketRespOSMCFSCo mmsJMSProducer/ V1}CreateTroubleTicketRespOSMCFSC ommsJMSProducer      | Service Name of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.  |
| Routing.SWI_spcTrouble_spcTic ket_spcService.RouteToCAVS                       | true/false<br>Default: false   | Indicates whether the Partner link SWI_spcTrouble_spcTicket_spcService should be routed to CAVS or the actual application.   |

**Table 27-6 (Cont.) CreateTroubleTicketSiebelCommsProvABCServiceImpl Properties**

| Property Name   | Value/Default Values                        | Description   |
|---|---|---|
| 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<br>Default: false                | CreateTroubleTicketSiebelCommsProvABCServiceImpl creates a trouble ticket response message (creates a cross-reference for the trouble ticket ID with the Siebel ID) and invokes the CommunicationsTroubleTicketResponseEBSV1 if this property is set to true or if the response code attribute is not null. Otherwise, this service acts only as a fire-and-forget flow and ignores the response. |
| TroubleTicket.UseDefaultInstance                                | true/false<br>Default: false                | If set to true, overwrites the target Siebel instance to the default instance indicated by the property Default.SystemID.<br>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 UpdateTroubleTicketSiebelCommsProvABCServiceImpl service name.

**Table 27-7 UpdateTroubleTicketSiebelCommsProvABCServiceImpl 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<br>Default: false                | Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).   |
| ABCSExtension.PreInvokeABSSWITroubleTicketIOABM                 | true/false<br>Default: false                | Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).  |
| Routing.SWI_spcTrouble_spcTicket_spcService.RouteToCAVS         | true/false<br>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 CreateOrderFalloutNotificationOSMCFSCommsProvImpl Properties

| Property Name   | Value/Default Values   | Description  |
|---|--|--|
| Default.SystemID  | OSMCFS_01  | Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property TroubleTicket.UseDefaultInstance is set to true.                                |
| Routing.CreateOrderFalloutNotification.OSMCFS_01.Endpoint URI | http://\${http.host name}:\${http.port}/orabpel/default/CreateOrderFalloutNotificationOSMCFSCommsJMSProducer/1.0 | Endpoint URI of the service that sends the fallout notification to OSMCFS_01. The default value is the EndpointURI of the out-of-the-box JMS enqueueing or producing service that enqueues or produces the message in the OSM JMS Queue. |
| Fault.DefaultSeverity   | Any number from 1 to 5<br>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 Account for Child 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](#)
- [Reintroducing Enterprise Business Services](#)
- [Understanding Multischema Migration](#)

# A

## 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_COM MON Value | LINEPRICETYP ECODE 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                       |

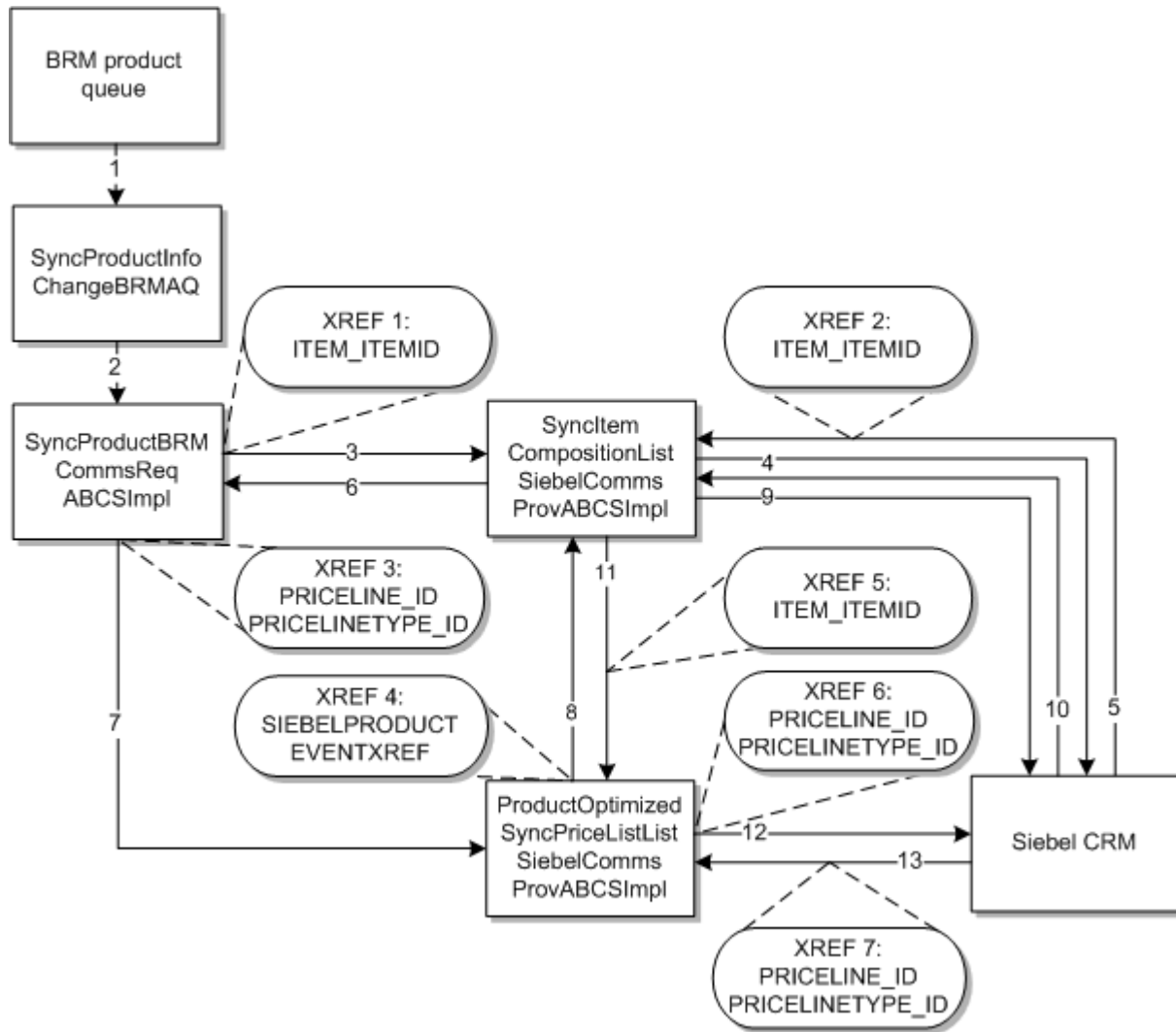
**Table A-2 (Cont.) Values of the Cross-References for Product Lifecycle Management**

| Entry                  | Description   | COMMON Value        | BRM_01 Value                           | SEBL_01 Value                                    | ITEM_ID_COMMON Value | LINEPRICETYPECODE Value      |
|------------------------|---|---------------------|--|--|----------------------|------------------------------|
| 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                           |
| PRICELINETYPE_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                           |
| SIEBELPRODUCTEVENTXREF | Cross-references BRM Product's Event that is associated with the main product in Siebel CRM.                                  | NA                  | NA                                     | NA   | From ITEM_ID.COMMON  | From PRICELINETYPE_ID.COMMON |

## Product Synchronization Flow

Figure A-1 illustrates the events that occur in the product synchronization integration flow and the points in the flow at which the integration updates the Oracle AIA XREF\_DATA database table.

Figure A-1 Product Synchronization Flow



As shown in the figure, the integration updates XREF\_DATA at the following points in the integration flow:

1. Before the SyncProductBRMCommsReqABCSImpl service makes the call to the SyncItemCompositionListSiebelCommsProvABCSImpl service, the integration makes the entries listed in Table A-3 in the XREF\_DATA table.

Table A-3 First Product Synchronization Update of XREF\_DATA

| XREF_TABLE_NAME | VALUE               |
|-----------------|---------------------|
| ITEM_ITEMID     | POID of BRM product |
| ITEM_ITEMID     | COMMON GUID1        |

2. During the response from Siebel CRM to the SyncItemCompositionListSiebelCommsProvABCSImpl service, the integration makes the entry listed in Table A-4 in the XREF\_DATA table.

**Table A-4 Second Product Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME | VALUE                |
|-----------------|----------------------|
| ITEM_ITEMID     | Siebel CRM ProductID |

- Before the SyncProductBRMCommsReqABCImpl service makes the call to the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service, the integration makes the entries listed in [Table A-5](#) in XREF\_DATA table.

**Table A-5 Third Product Synchronization Update of the XREF\_DATA Table**

| 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 ProductOptimizedSyncPriceListListSiebelProvABCImpl calls the SyncItemCompositionListSiebelCommsProvABCImpl service, the integration makes the entries listed in [Table A-6](#) in the XREF\_DATA table.

**Table A-6 Fourth Product Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME        | VALUE                   |
|------------------------|-------------------------|
| SIEBELPRODUCTEVENTXREF | LINEPRICETYPECODE GUID2 |
| SIEBELPRODUCTEVENTXREF | ITEM_ID_COMMON GUID1    |

- During the response from SyncItemCompositionListSiebelCommsProvABCImpl, the integration makes the entries listed in [Table A-7](#) in the XREF\_DATA table.

**Table A-7 Fifth Product Synchronization Update of the XREF\_DATA Table**

| 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 ProductOptimizedSyncPriceListListSiebelProvABCImpl calls Siebel CRM, the integration makes the entries listed in [Table A-8](#) in the XREF\_DATA table.

**Table A-8 Sixth Product Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME | VALUE          |
|-----------------|----------------|
| PRICELINE_ID    | ITEM_ID_COMMON |

**Table A-8 (Cont.) Sixth Product Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME                                      | VALUE          |
|--|----------------|
| PRICELINETYPE_ID<br>(in case of multi-event product) | ITEM_ID_COMMON |

7. During the response from Siebel CRM to the ProductOptimizedSyncPriceListListSiebelProvABCServiceImpl service, the integration makes the entries listed in [Table A-9](#) in the XREF\_DATA table.

**Table A-9 Seventh Product Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME                                      | VALUE                                  |
|--|--|
| PRICELINE_ID   | Siebel CRM ProductID for Event Product |
| PRICELINETYPE_ID<br>(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.

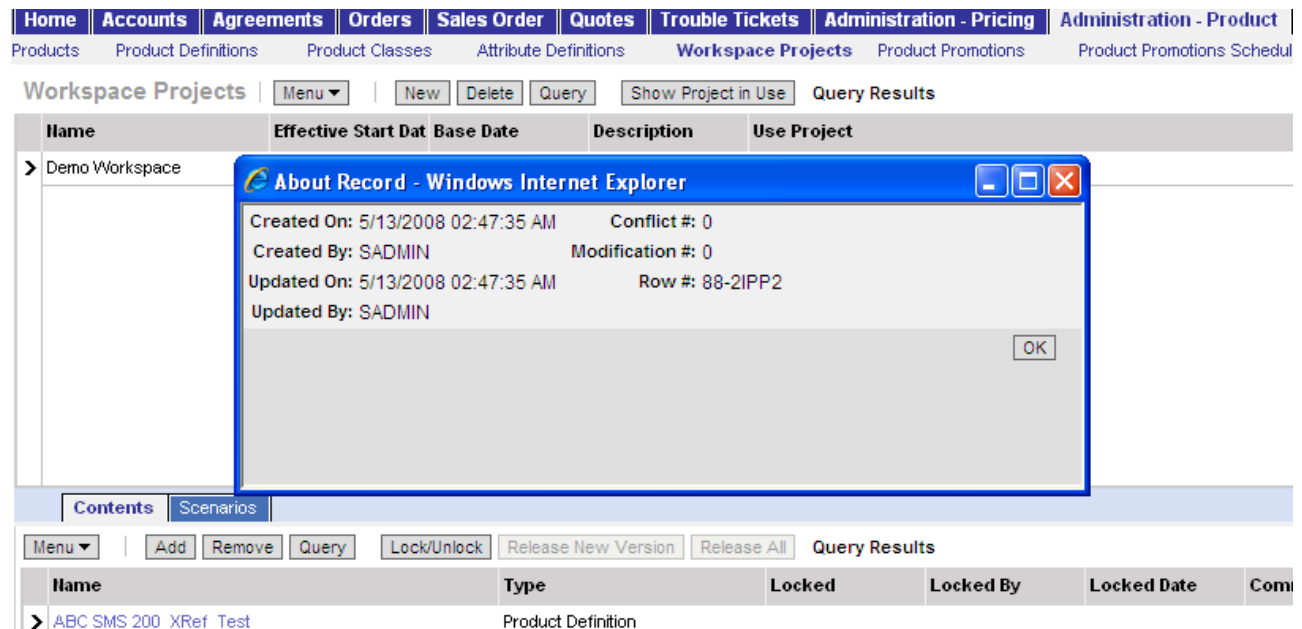
1. Create a simple product in BRM to be synchronized to Siebel CRM, as shown in [Figure A-2](#).

**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**



- 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**

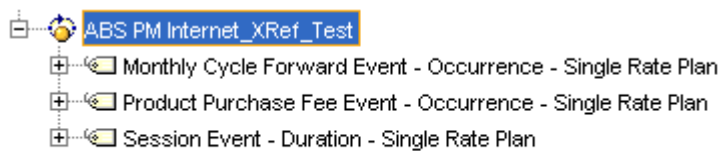
| XREF_TABLE_NAME               | XREF_COLUMN_NAME  | ROW_NUMBER | VALUE                |
|-------------------------------|-------------------|------------|----------------------|
| ITEM_ID                       | BRM_01            | ROWNUM_1   | BRM_PROD_01          |
| ITEM_ID                       | 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 |
| SIEBELPRODUCTEVEN<br>TXREF    | LINEPRICETYPECODE | ROWNUM_4   | COMMON_PRICETYPE_ID1 |
| SIEBELPRODUCTEVEN<br>TXREF_ID | ITEM_ID_COMMON    | ROWNUM_4   | COMMON_PROD_01       |
| PRICELINE_ID                  | ITEM_ID_COMMON    | ROWNUM_2   | COMMON_PROD_01       |
| 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.

- 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**



- Verify the synchronized records in Siebel CRM, as shown in [Figure A-5](#).

**Figure A-5 Synchronized Complex Product Records in Siebel CRM**

The screenshot shows the Siebel CRM interface. At the top, there is a navigation bar with various menu items like 'Products', 'Product Definitions', etc. Below that is a table with columns: Name, Service Type, Description, Product Line, Locked By, Check Eligibility, Inclusive Eligible, Revision, Status, and Compo. The table contains three rows related to 'ABS PM Internet\_XRef\_Test'. Below the table, three 'About Record' pop-up windows are displayed, each showing details for a specific row. The 'Row #' field in each window is circled in red: 88-26V97, 88-26VQ2, and 88-26VFR. The pop-ups also show creation and update dates and times, and the user 'SADHDI'.



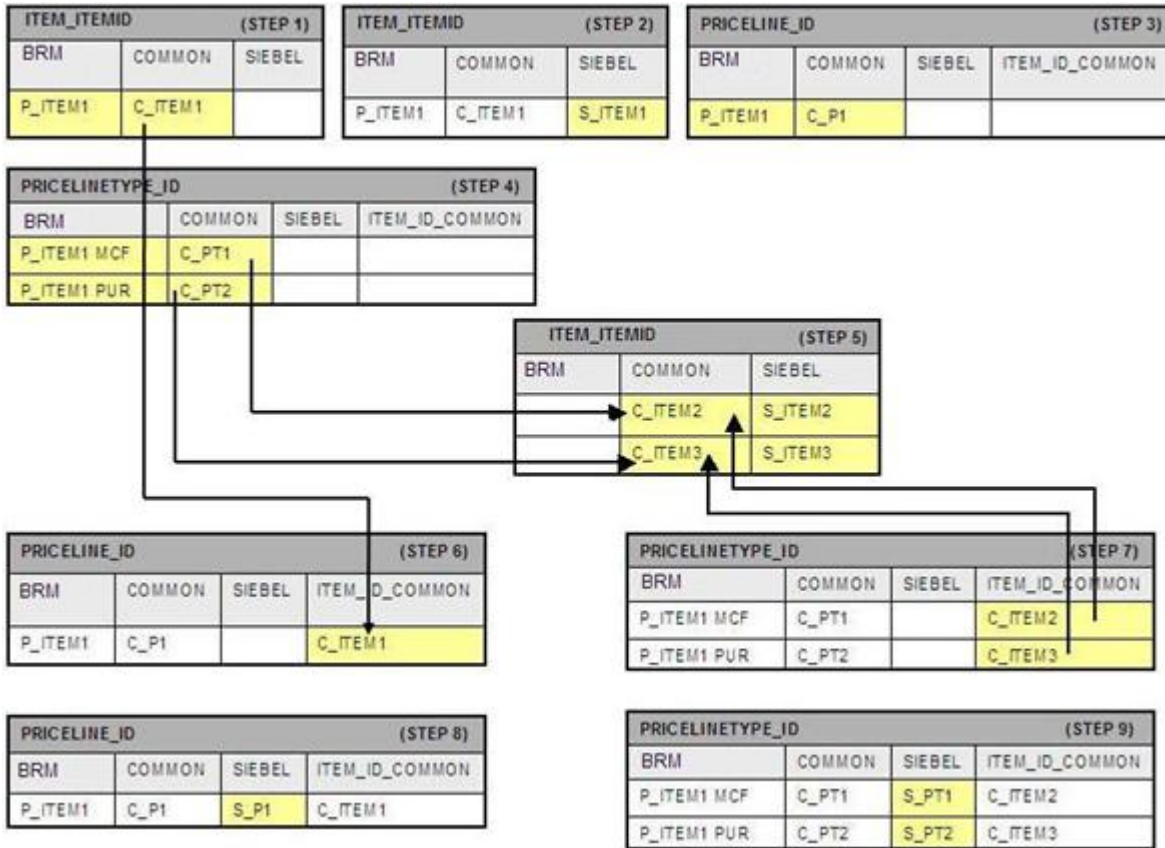
3. Verify the data entered into the XREF\_DATA table is correct as shown in table [Table A-11](#).

**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  |
| PRICELINETYPE_ID              | BRM_01            | ROWNUM_4   | BRM_PROD_01_EVENT2  |
| PRICELINETYPE_ID              | COMMON            | ROWNUM_4   | COMMON_PRICETYPE_02 |
| SIEBELPRODUCTEVE<br>NTXREF    | LINEPRICETYPECODE | ROWNUM_4   | COMMON_PRICETYPE_01 |
| SIEBELPRODUCTEVE<br>NTXREF_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.

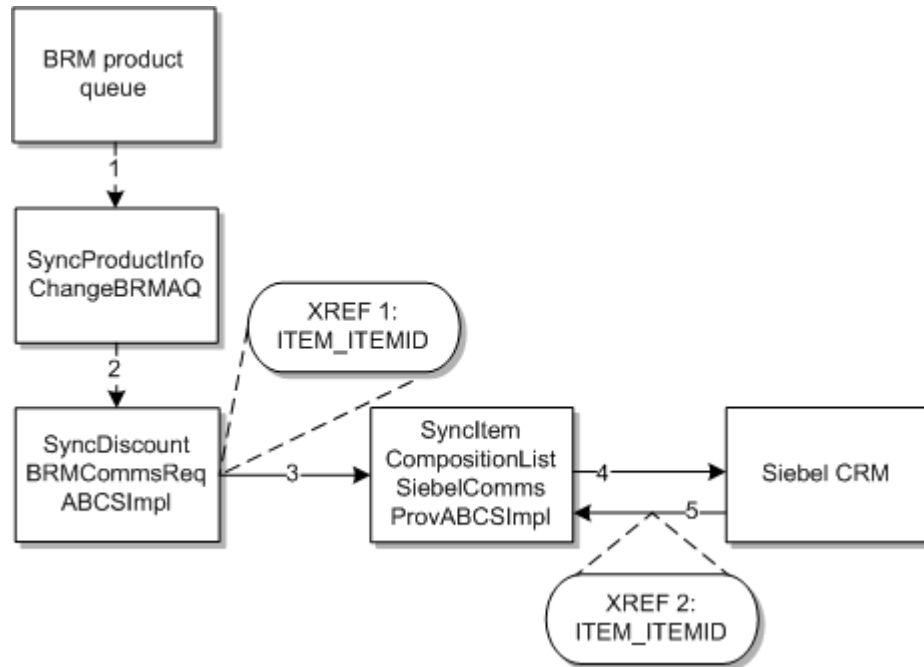
Figure A-6 Maintaining Mappings in the Cross-Reference Table



## Discount Synchronization Flow

Figure A-7 illustrates the events that occur in the discount synchronization integration flow and the points in the flow at which the integration updates the Oracle AIA XREF\_DATA database table.

**Figure A-7 Discount Synchronization Flow**



As shown in the figure, the integration updates XREF\_DATA at the following points in the integration flow:

1. Before the SyncDiscountBRMCommsReqABCSImpl service makes the call to the SyncItemCompositionListSiebelCommsprovABCSImpl service, the integration makes the entries listed in [Table A-12](#) in the XREF\_DATA table.

**Table A-12 First Discount Synchronization Update of the XREF\_DATA Table**

| XREF_TABLE_NAME | VALUE               |
|-----------------|---------------------|
| ITEM_ITEMID     | COMMON GUID         |
| ITEM_ITEMID     | POID Of BRM Product |

2. During the response from Siebel CRM to SyncItemCompositionListSiebelCommsProvABCSImpl, the integration makes the entry listed in [Table A-13](#) in the XREF\_DATA table.

**Table A-13 Second Discount Synchronization Update of the XREF\_DATA Table**

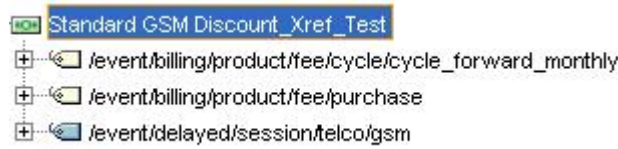
| XREF_TABLE_NAME | VALUE                |
|-----------------|----------------------|
| ITEM_ITEMID     | Siebel CRM PRODUCTID |

## 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 |
| 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                         |

# B

## 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 can be used to determine if a change has occurred and 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"><li>• Sales Order</li><li>• Fulfillment Order</li><li>• Provisioning Order</li></ul>                  |

### **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.<br>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.<br><br>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<br>ALL OR NONE                                      | No         | None                  | None  | <i>CommsOrderEBO/PartialFulfillmentAllowed</i> indicator                    |
| 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.<br><br>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.<br><br>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/</i> FulfillmentModeCode                                   |
| Customer Class            | Identifies type of customer: Residential, Business, and so on   | Residential<br>Business                                     | No         | None                  | None  | <i>CommsOrderEBO/</i> CustomerPartyReference / CustomerPartyAccountTypeCode |
| Organization ID           | Identifies the organization/LOB generating the order. No cross-reference exists.  | NA  | No         | None                  | No cross-reference. OM should use the application-specific ID if required in any of the rules; if Organization IDs are unique and synchronized across all order capture systems.  | <i>CommsOrderEBO/</i> BusinessUnitReference/ BusinessUnitIdentification /ID |

**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.<br>This value is optional for orders whose Job Cardinality is 1.<br>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<br>9: Urgent. Used for expedited orders.<br>7: High. CSP determines its use.<br>5: Medium. CSP determines its use.<br>3: Low. Recommended for job orders.<br>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/RequestedDeliveryDateTi<br/>me</i> |
| Fulfillment Status           | Reports aggregate order fulfillment status.  | In Progress<br>, Failed,<br>Canceled<br>, Complet<br>e | Yes        | None                  | This is different from the Status attribute tracked within Siebel CRM.  | <i>CommsOrderEBO/Status/<br/>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> .<br><br>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.  | NA            | NA         | NA                    | NA  | <i>CommsOrderEBO/ CustomerPartyReference / CustomerPartyAccountContact/FirstName</i><br><i>CommsOrderEBO/ CustomerPartyReference / CustomerPartyAccountContact/LastName</i><br><i>CommsOrderEBO/ CustomerPartyReference / CustomerPartyAccountContactPhoneCommunication/PhoneCommunication/CompleteNumber</i><br><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 <b>AIAConfigurationProperties.xml</b> file is used. | <i>CommsOrderEBM/DataArea/CommsOrder/PriceListReference/PriceListIdentification/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               |
|---------------------------|--|---------------|------------|-----------------------|--|-----------------------------------|
| Order Subject             | Indicates the type of fulfillment system to which the order is sent. | CUSTOMER      | No         | No                    | Used by OSM SOM to determine whether to transform an order from a customer order to a service order. | <i>CommsOrderEBO/OrderSubject</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> |

**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  |
|---------------------------|---|------------------------------|------------|-----------------------|---|--|
| Asset Integration ID      | Uniquely identifies an instance of a product that was or is being purchased.  | NA                           | Yes        | AIA2.0                | <p>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.</p> <p>A revision should never change the Asset Integration ID.</p> <p>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.</p> | <i>CommsOrderLine/InstalledProductReference / InstalledProductIdentification/BusinessComponentID</i> |
| Line Number               | Identifies the line regarding its position in the line item tree.   | NA                           | No         | None                  | <p>Line number establishes the parent child relationship between Order Lines of the same order, but it may vary across revisions.</p> <p>Therefore, do not rely on it for matching Order Lines across revisions.</p>  | <i>CommsOrderEBO/CommsOrderLine/Identification/ID</i>  |
| 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>           |

**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  |
|------------------------------|--|------------------------------|------------|-----------------------|-------------------|--|
| 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>                  |
| Promotion Line ID            | References an order line that represents the promotion/marketing offer under which the order line is being purchased.  | NA                           | No         | Yes                   | None              | <i>CommsOrderEBO/CommsOrderLine/PromotionCommsOrderLineReference/PromotionCommsOrderLineIdentification/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 Asset Integration ID | References an asset that represents the promotion/marketing offer under which the order line is being purchased.   | NA  | Yes        | Yes                   | 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        | Yes                   | 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        | Yes                   | None    | <i>CommsOrderEBO/CommsOrderLine/ServiceAddress/Address</i>  |
| 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        | Yes                   | None    | <i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/RequestedDeliveryDateTime</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  |
|---------------------------|--|------------------------------|------------|-----------------------|---|--|
| 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        | Yes                   | 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        | Yes                   | None  | <i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/CycleStartDate</i>        |
| 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        | Yes                   | 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>        |

**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   |
|-----------------------------|--|------------------------------|------------|-----------------------|---|---|
| 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>             |
| 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.<br><br>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> |

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   |
|---------------------------|---|--|------------|-----------------------|--|---|
| 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>              |
| 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:<br>o Requires customer interaction<br>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.<br><br>This value may be a customer account or an account from the account hierarchy. | NA                           | Yes        | Yes                   | None    | <i>CommsOrderEBO/CommsOrderLine/CommsOrderSchedule/BillToPartyReference/CustomerPartyAccountIdentification/BusinessComponentID</i>   |
| Billing Profile           | References the billing profile record that holds the customer's billing/payment preferences.<br><br>This value may be associated to the customer account or to a separate billing account.                 | NA                           | Yes        | Yes                   | 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.<br><br>This value may be customer account or an account from the account hierarchy.   | NA                           | Yes        | Yes                   | 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 Account             | References an account record that represents the legal owner of a service, against which collections actions can be taken.<br><br>The value must be the parent account in the hierarchy of the service account. | NA                           | Yes        | Yes                   | Only considered when the Oracle AIA configuration property <b>O2C.LegalGroup</b> is set to <b>TRUE</b> . | New value:<br>ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/SalesOrderLine/OwnerAccountId<br><br>Prior value:<br>ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/PriorSalesOrderLine/OwnerAccountId |
| 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.<br><br>This value is relevant for network ordering only.  | NA                           | Yes        | Yes                   | 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.<br><br>This value is relevant for network ordering only.  | NA                           | Yes        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | None    | <i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressID"]/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   |
|-------------------------------|---|------------------------------|------------|-----------------------|--|---|
| 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        | Yes                   | None   | <i>CommsOrderEBO/CommsOrderLine/CommsOrderLineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServicePointID"]/ValueText</i> |
| 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        | Yes                   | 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        | Yes                   | None   | <i>CommsOrderEBO/CommsOrderLine/ItemReference/Description</i><br>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        | Yes                   | 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>  |



**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   |
|--------------------------------|---|------------------------------|------------|-----------------------|---|---|
| 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> |
| 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, RE DO, 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>                            |

**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  |
|---------------------------|--|--|------------|-----------------------|---|--|
| 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>   |
| 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>      |

**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 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        | Yes                   | None  | <i>CommsOrderEBO/CommsOrderLine/ItemReference/SupplierPartyReference/PartyIdentification/ID</i>       |
| Vendor Part Number        | Identifies the product part number to the vendor.   | NA   | Yes        | Yes                   | 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<br>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>                           |
| 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>                              |

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   |
|------------------------------|--|------------------------------|------------|-----------------------|-----------------------------------|---|
| 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>   |
| Charge Frequency Code        | Indicates charge frequency unit of measure, for example, monthly, quarterly, yearly.   | NA                           | NA         | NA                    | None                              | <i>CommsOrderEBO/CommsOrderLine/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>   |

**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   |
|---------------------------|--|---|------------|-----------------------|---------|---|
| 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        | Yes                   | 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        | Yes                   | 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        | Yes                   | N A     | <i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DiscountMethodCode</i>      |
| Discount Percent          | Indicates the percent by which the list price is discounted. | NA  | Yes        | Yes                   | N A     | <i>CommsOrder/CommsOrderLine/CommsOrderSchedule/CommsOrderScheduleCharge/Charge/DiscountPercent</i>         |
| Discount Amount           | Indicates the amount by which the list price is discounted.  | NA  | Yes        | Yes                   | 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 <b>AIAConfigurationProperties.xml</b> file is used regardless of values specified for this attribute. | <i>CommsOrderEBM/DataArea/CommsOrderLine/CommsOrderSchedule/PriceListReference/PriceListIdentification/ID</i>  |
| Service Family            | Indicates the category of a service, such as broadband or wireless.   | NA                           | No         | No                    | Set by order management, used to determine actions of fulfillment systems.  | <i>CommsOrderEBO/CommsOrderLine/ServiceFamily</i>  |
| Technical Inventory ID    | Correlates assets across technical inventory systems.   | NA                           | No         | No                    | Set by service and resource management system.  | <i>CommsOrderEBO/CommsOrderLine/InstalledProductReference/TechnicalInventoryId</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="&lt;OrderLine.XA.Attribute&gt;"]/@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="&lt;OrderLine.XA.Attribute&gt;"]/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="&lt;OrderLine.XA.Attribute&gt;"]/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="&lt;OrderLine.XA.Attribute&gt;"]/Value</i>        |

# C

## 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 <b>Due Date</b> .                                    |
| 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_COM MIT_CUSTOMER   | Pass Order Date coming from Siebel CRM.                                      |
| CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service            | CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service            | CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service            | CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service            |



**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   |
|---|--|--|---|
| Single Phase Billing - Billing Fulfillment<br>Promotion Purchase  | Oracle AIA passes the Purchase Date as the Valid From date for bundle purchase (that represents purchased promotion).<br><br>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.  | PCM_OP_SUBSCRIPTI<br>ON_SET_BUNDLE   | Pass Order Date and Requested Delivery Date coming from Siebel CRM.<br><br>Set Purchase Date to Actual Delivery Datetime.   |
| Single Phase Billing - Billing Fulfillment  | If all three of the billing dates are set, then Oracle AIA uses Order Date as Effective Date, and sets respective offset (Order Date - respective billing date).<br><br>Billing dates are: Purchase Date, Cycle Start Date and Usage Start Date.<br><br>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.<br><br>For purchase of a service bundle, this check for existence of billing dates applies to ALL products and discounts included in the service bundle. | PCM_OP_CUST_MODI<br>FY_CUSTOMER<br><br>PCM_OP_SUBSCRIPTI<br>ON_PURCHASE_DEAL | Pass Order Date and Requested Delivery Date coming from Siebel CRM.<br><br>Set Purchase Date, Start Cycle, and Start Usage to Actual Delivery Datetime to explicitly control setting of billing dates.  |
| Single Phase Billing - Billing Fulfillment.<br>Time Based Account or Service level Subscription Product/Discount Purchase | In addition to setting of billing dates as described previously, if Service End Date is passed, then Oracle AIA additionally sets the Purchase, Cycle and Usage end date offsets (difference between the respective billing date and service end date).<br><br>If any of the billing dates (Purchase, Cycle, or Usage start) are not set then Oracle AIA uses the Order Date to calculate the Purchase, Cycle and Usage end date offsets (difference between the Order Date and Service End Date).   | PCM_OP_MODIFY_CU<br>STOMER<br><br>PCM_OP_SUBSCRIPTI<br>ON_PURCHASE_DEAL      | Populate Purchase, Cycle and Usage Start dates (this is required for enabling time-based offerings).<br><br>Calculate the Service End Date based on TBO attributes as documented in TBO section.<br><br>See " <a href="#">Supporting Time-Based Offerings on Orders</a> " for more information. |

**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 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_SUBSCRIPTI<br/>ON_SET_PRODINFO</p> <p>PCM_OP_SUBSCRIPTI<br/>ON_SET_DISCOUNTIN<br/>FO</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 "<a href="#">Supporting Time-Based Offerings on Orders</a>" 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_SUBSCRIPTI<br/>ON_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_SUBSCRIPTI<br/>ON_PURCHASE_DEAL</p>   | <p>Set Actual Delivery Datetime.</p>  |
| <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_SUBSCRIPTI<br/>ON_SET_PRODUCT_S<br/>TATUS</p> <p>PCM_OP_SUBSCRIPTI<br/>ON_SET_DISCOUNT_S<br/>TATUS</p> <p>PCM_OP_CUST_SET_<br/>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. 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_SUBSCRIPTI<br/>ON_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>   |

**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>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>  |
| <p>Two Phase Billing - Billing Fulfillment.</p> <p>Account-level or service-level Item Type Product Purchase.</p>                  | <p>If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset.</p> <p>Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.</p>   | <p>PCM_OP_SUBSCRIPTION_SET_PRODINFO</p>   | <p>Reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update).</p>  |
| <p>Two Phase Billing - Billing Fulfillment.</p> <p>Account-level or service-level Subscription Type Product/Discount Purchase.</p> | <p>If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset.</p> <p>Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.</p>   | <p>PCM_OP_SUBSCRIPTION_SET_PRODINFO</p> <p>PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO</p> | <p>To support validation mode, reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update).</p> <p>To support latency mode, reset Cycle date to Actual Delivery Datetime (set prior value to trigger update).</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>Two Phase Billing - Billing Fulfillment.</p> <p>Time-Based account-level or service-level Subscription Product/ Discount Purchase.</p> | <p>If Service End Date is passed, then Oracle AIA uses that to set the Purchase, Cycle, and Usage end dates for products/ discounts purchased.</p> | <p>PCM_OP_SUBSCRIPTI<br/>ON_SET_PRODINFO<br/>PCM_OP_SUBSCRIPTI<br/>ON_SET_DISCOUNTIN<br/>FO</p> | <p>Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate Purchase, Cycle, and Usage start dates.</p> <p>See "<a href="#">Supporting Time-Based Offerings on Orders</a>" for more information about TBO attributes.</p> |

# 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<br>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<br>Same as Update with communicating changes to line attributes. | Ignored other than to determine original BRM asset. |

[Table D-3](#) shows which line actions are supported for promotion groups. These actions apply to the promotion group instance, the membership products, and the rewards.

The MACD actions described in [Table D-2](#) are all supported for service bundles associated with promotion group membership products. However, if you suspend a service associated with a promotion group owner membership product, members of that promotion group cannot consume the rewards until you resume the service.

**Table D-3 MACD Line Actions Supported for Promotion Groups**

| Add | Delete | Suspend | Resume | Update | Move-Add | Move-Delete |
|-----|--------|---------|--------|--------|----------|-------------|
| Yes | Yes    | No      | No     | No     | No       | No          |

## 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 12-7](#) for an illustration of nested billing products and nested service bundles.

[Table D-4](#) shows which line actions are supported for service-level billing subscription products.

**Table D-4 MACD Line Actions Supported for Service-Level Billing Subscription Products**

| Add   | Delete | Suspend                                   | Resume                                    | Update | Move-Add   | Move-Delete   |
|---|--------|---|---|--------|--|---|
| Yes<br>Can communicate price or discount override as part of this action. | Yes    | N/A<br>Unsupported by Siebel CRM and BRM. | N/A<br>Unsupported by Siebel CRM and BRM. | Yes    | Yes<br>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 discount products.

**Table D-5 MACD Line Actions Supported for Service-Level Billing Discount Products**

| Add | Delete | Suspend                                   | Resume                                    | Update | Move-Add   | Move-Delete   |
|-----|--------|---|---|--------|--|---|
| Yes | Yes    | N/A<br>Unsupported by Siebel CRM and BRM. | N/A<br>Unsupported by Siebel CRM and BRM. | Yes    | Yes<br>Same as UPDATE with communicating changes to line attributes. | Ignored other than to determine original BRM asset. |

[Table D-6](#) 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-6 MACD Line Actions Supported for Service-Level Billing Item Products**

| Add   | Delete | Suspend | Resume | Update   | Move-Add | Move-Delete |
|---|--------|---------|--------|--|----------|-------------|
| Yes<br>Can communicate price or discount override as part of this action. | N/A    | N/A     | N/A    | N/A<br>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-7](#) shows which line actions are supported for account-level billing subscription products.

**Table D-7 MACD Line Actions Supported for Account-Level Billing Subscription Products**

| Add   | Delete | Suspend | Resume | Update | Move-Add | Move-Delete |
|---|--------|---------|--------|--------|----------|-------------|
| Yes<br>Can communicate price or discount override as part of this action. | Yes    | Yes     | Yes    | Yes    | Ignored  | Ignored     |

[Table D-8](#) shows which line actions are supported for account-level billing discounts.

**Table D-8 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-9](#) 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-9 MACD Line Actions Supported for Account-Level Billing Item Products**

| Add   | Delete | Suspend | Resume | Update   | Move-Add | Move-Delete |
|---|--------|---------|--------|--|----------|-------------|
| Yes<br>Can communicate price or discount override as part of this action. | N/A    | N/A     | N/A    | N/A<br>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 Marketing Bundle Attribute Changes

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 Service Bundle Attribute Changes

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 Service Bundle Component Attribute Changes

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 Service-Level Billing Subscription Product Attribute Changes

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 Service-Level Billing Discount Product Attribute Changes

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 Service-Level Billing Item Product Attribute Changes

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 Service-Level Special Rating Product Attribute Changes

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 "[About Friends and Family Lists](#)" for more information.

## Communicating Account-Level Product Attribute Changes

This section lists the changes to account-level products that the integration communicates to billing.

Subscription products and discounts are purchased at the account level if they are not in a service bundle. Subscription products and discounts nested up to two levels within a service bundle are purchased at the service level. See "[Examples of Mapping for Bundled Products](#)" for an example that includes nested products purchased at the account level and at the service level.

BRM does not support transferring account-level products or discounts from one account to another. Siebel CRM validates this.

## Communicating Account-Level Billing Subscription Product Attribute Changes

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 Account-Level Billing Discount Attribute Changes

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 Account-Level Billing Item Product Attribute Changes

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

# E

## Examples of Changing the Paying Account for Child Accounts

This appendix gives examples of changing the paying account for child accounts. When service-level balance groups are enabled, there are different options for changing the paying account 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 nonpaying child account under one or more paying parent accounts. When the service account on an order is different from the billing account, the integration creates the service account with a nonpaying **/billinfo** object as a child account under the billing account in BRM. The nonpaying **/billinfo** object points to the billing profile of the paying parent account.

The integration also creates a **/billinfo** hierarchy under the service account representing the billing relationship between the service account and the billing accounts.

The subsequent orders in each example are change orders that change the paying account on some or all of the services purchased in the first order. When these change orders are submitted, the integration creates new balance groups and nonpaying **/billinfo** objects for the services in the child account pointing to the paying parent account's chosen billing profile. The integration also updates the **/billinfo** hierarchy with the new parent account information.

When the paying account for all services on one service account is changed, the integration moves the child account under the new paying parent account in BRM.

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 group names would be identical.

Although most of the examples show nonpaying child accounts, child accounts can also be self-paying. A service account could pay for some of its own services and have multiple billing accounts paying for the rest of its services.

[Table E-1](#) defines the abbreviations used in the examples.

**Table E-1 Abbreviations Used in the Examples**

| Abbreviation | Description                         |
|--------------|-------------------------------------|
| SA           | Service account: the child account  |
| BA           | Billing account: the parent account |

**Table E-1 (Cont.) Abbreviations Used in the Examples**

| Abbreviation | Description  |
|--------------|--|
| BG           | Balance group: the BRM balance group in which each service is tracked  |
| BP           | Billing profile: the billing profile on the parent account   |
| NPB          | Nonpaying <b>/billinfo</b> object: a <b>/billinfo</b> object created for a child 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 account for child accounts when service-level balance groups are enabled.

### Changing the Paying Account for a Child Account with Separate Billing Profiles

This example shows:

- A base order that purchases two services with separate billing profiles on one parent account for a child account
- Change orders that change the paying account for these services in the child account

[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   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| ADD    | Wireless  | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband | 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 nonpaying **/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 account for the services while maintaining separate billing profiles.

**Table E-3 Order to Change the Paying Account with Separate Billing Profiles**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Wireless  | SA1             | BA2             | BA2-BP3         |
| UPDATE | Broadband | 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 Account with Separate Billing Profiles**



### Alternative: Single Billing Profile

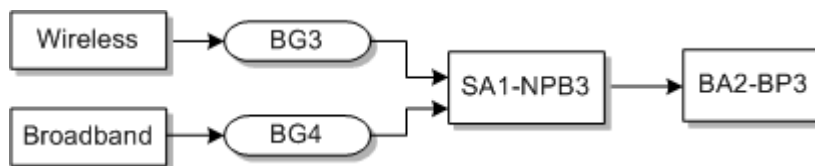
Alternatively, the change order can change the paying account for the services and assign the same billing profile to both. Table E-4 shows this alternative order.

**Table E-4 Alternative Change Order to Change the Paying Account with a Single Billing Profile**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Wireless  | SA1             | BA2             | BA2-BP3         |
| UPDATE | Broadband | 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 Account with a Single Billing Profile**



### Alternative: Multiple Parents

Alternatively, the change order can change the paying account for only one of the services. Table E-5 shows this alternative order.

**Table E-5 Alternative Change Order to Change the Paying Account for One Service**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Broadband | SA1             | BA2             | BA2-BP1         |

Figure E-4 shows the results of sending the alternative order to billing, and includes the wireless service from the original order for comparison.

**Figure E-4 Result of Alternative Order for Multiple Parents**



## Changing the Paying Account for a Child Account with a Single Billing Profile

This example shows:

- A base order that purchases two services with a single billing profile on a parent account for a child account
- Change orders that change the paying account for the services on the child account

Table E-6 shows the base order. The same billing profile is assigned to both services.

**Table E-6 Base Order for Services with a Single Billing Profile**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| ADD    | Wireless  | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband | SA1             | BA1             | BA1-BP1         |

Figure E-5 shows the result of sending the base order to billing. The services are tracked in separate balance groups that point to the same nonpaying **billinfo** object.

**Figure E-5 Result of Base Order for Services with a Single Billing Profile**

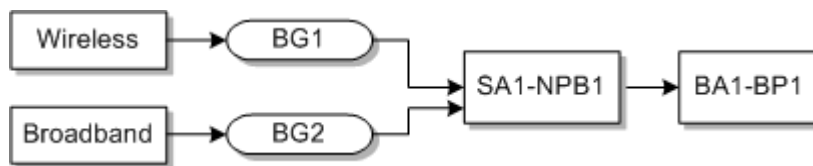


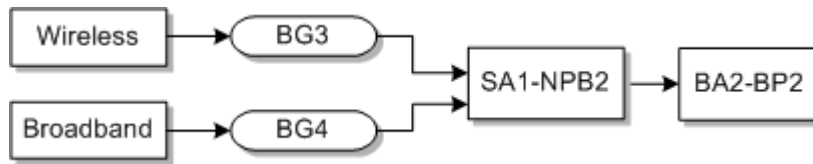
Table E-7 shows the change order to change the paying account for the service son the child account while maintaining a single billing profile for the services.

**Table E-7 Order to Change the Paying Account with a Single Billing Profile**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Wireless  | SA1             | BA2             | BA2-BP2         |
| UPDATE | Broadband | SA1             | BA2             | BA2-BP2         |

Figure E-6 shows the result of sending the change order to billing.

**Figure E-6 Result of Order to Change the Paying Account with a Single Billing Profile**



## Alternative: Multiple Billing Profiles

Alternatively, the change order can change the paying account for the services on the child account and assign separate billing profiles to the services. Table E-8 shows this alternative order.

**Table E-8 Alternative Order to Change the Paying Account with Multiple Billing Profiles**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Wireless  | SA1             | BA2             | BA2-BP2         |
| UPDATE | Broadband | SA1             | BA2             | BA2-BP3         |

Figure E-7 shows the results of sending the alternative order to billing.

**Figure E-7 Result of Alternative Order to Change the Paying Account with Multiple Billing Profiles**



## Changing the Paying Account for Multiple Child Accounts

This example shows:

- A base order that purchases one service each with separate billing profiles for two child accounts
- A change order that changes the paying account for the services on the child accounts



Table E-9 shows the base order. A separate billing profile is assigned to each service.

**Table E-9 Base Order for Services in Multiple Child Accounts**

| Action | Service    | Service Account | Billing Account | Billing Profile |
|--------|------------|-----------------|-----------------|-----------------|
| ADD    | Wireless 1 | SA1             | BA1             | BA1-BP1         |
| ADD    | Wireless 2 | SA2             | BA1             | BA1-BP2         |

Figure E-8 shows the results of sending the base order to billing. The services are tracked in separate balance groups that point to separate nonpaying /billinfo objects.

**Figure E-8 Result of Base Order for Services in Multiple Child Accounts**



Table E-10 shows the change order to change the paying account for the services on both child accounts at once while maintaining separate billing profiles.

**Table E-10 Order to Change the Paying Account for Multiple Child Accounts with Multiple Billing Profiles**

| Action | Service    | Service Account | Billing Account | Billing Profile |
|--------|------------|-----------------|-----------------|-----------------|
| UPDATE | Wireless 1 | SA1             | BA2             | BA2-BP3         |
| UPDATE | Wireless 2 | SA2             | BA2             | BA2-BP4         |

Figure E-9 shows the results of sending the change order to billing.

**Figure E-9 Result of Order to Change the Paying Account for Multiple Child Accounts with Multiple Billing Profiles**



## Alternative: Single Billing Profile

Alternatively, the change order can change the paying account for the services on the child accounts and assign the same billing profile to both services. Table E-11 shows this alternative order.

**Table E-11 Alternative Order to Change the Paying Account for Multiple Child Accounts with a Single Billing Profile**

| Action | Service    | Service Account | Billing Account | Billing Profile |
|--------|------------|-----------------|-----------------|-----------------|
| UPDATE | Wireless 1 | SA1             | BA2             | BA2-BP3         |
| UPDATE | Wireless 2 | SA2             | BA2             | BA2-BP3         |

Figure E-10 shows the result of sending the alternative order to billing.

**Figure E-10 Result of Alternative Order to Change the Paying Account for Multiple Child Accounts with a Single Billing Profile**



## Changing Multiple Paying Accounts

This example shows:

- A base order that purchases two services for a child account paid for by separate parent accounts
- Change orders that change the paying parent for one of the services

Table E-12 shows the base order. A different billing account is assigned to each service.

**Table E-12 Base Order for Services with Multiple Paying Accounts**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| ADD    | Wireless  | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband | SA1             | BA2             | BA2-BP1         |

Figure E-11 shows the result of sending the base order to billing. The services are tracked in separate balance groups that point to separate nonpaying **/billinfo** objects.

**Figure E-11 Result of Base Order for Services with Multiple Paying Accounts**



Table E-13 shows the change order to change the paying account for the Broadband service.

**Table E-13 Change Order to Change the Paying Parent for One Service**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Broadband | SA1             | BA1             | BA1-BP1         |

Figure E-12 shows the results of sending the change order to billing, and includes the wireless service from the original order for comparison.

**Figure E-12 Result of Change Order for a Single Paying Account**



## Alternative: Separate Billing Profiles

Alternatively, the change order can change the paying account for one service and use a new billing profile. Table E-14 shows this alternative order.

**Table E-14 Alternative Change Order to Change the Paying Account for One Service**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Broadband | SA1             | BA1             | BA1-BP2         |

Figure E-13 shows the results of sending the alternative order to billing, and includes the wireless service from the original order for comparison.

**Figure E-13 Result of Alternative Change Order for a Single Paying Parent**



## Alternative: Self Paying and Paying Parent

Alternatively, the change order can change the paying account for one service to be self paying. Table E-15 shows alternative order.

**Table E-15 Alternative Change Order to Change the Paying Account for One Service to Self Paying**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Broadband | SA1             | SA1             | SA1-BP1         |

Figure E-14 shows the results of sending the alternative order to billing, and includes the wireless service from the original order for comparison.

**Figure E-14 Result of Alternative Change Order for a Paying Parent and a Self Paying Account**



## Examples when Service-Level Balance Groups Are Disabled

This section gives examples of changing the paying parent on child accounts when service-level balance groups are disabled.

### Changing the Paying Parent for a Child Account

This example shows:

- A base order that purchases two services for a child account
- A change order that changes the paying parent for the child account and its services

Table E-16 shows the base order. The same billing profile is assigned to both services.

**Table E-16 Base Order for Services in a Child Account**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| ADD    | Wireless  | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband | SA1             | BA1             | BA1-BP1         |

Figure E-15 shows the result of sending the base order to billing. The services are tracked together under the default account-level balance group.

**Figure E-15 Result of Base Order for Services in a Child Account**



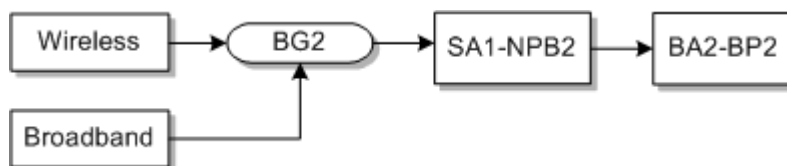
Table E-17 shows the change order to change the paying parent for the child account.

**Table E-17 Order to Change the Paying Parent for a Child Account**

| Action | Service   | Service Account | Billing Account | Billing Profile |
|--------|-----------|-----------------|-----------------|-----------------|
| UPDATE | Wireless  | SA1             | BA2             | BA2-BP2         |
| UPDATE | Broadband | SA1             | BA2             | BA2-BP2         |

Figure E-16 shows the result of sending the change order to billing.

**Figure E-16 Result of Order to Change the Paying Parent for a Child Account**



## Changing the Paying Parent for Multiple Child Accounts with Separate Billing Profiles

This example shows:

- A base order that purchases one service each for two child accounts
- A change order that changes the paying parent for both accounts and all their services

Table E-18 shows the base order. A separate billing profile is assigned to each child account.

 **Note:**

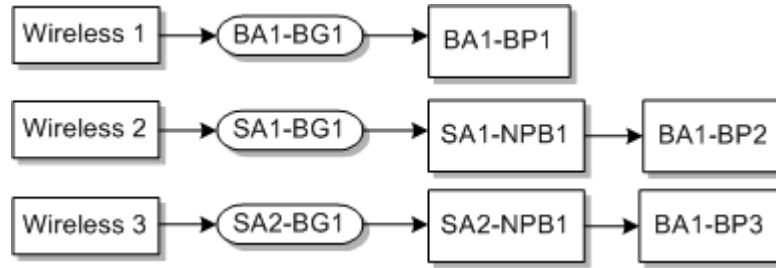
When service-level balance groups are disabled, a parent account can only use multiple billing profiles to pay for services under separate child accounts. A single child account cannot have multiple billing profiles. See "[Problems When Integrating Separate Billing Profiles on the Same Account](#)" for an example of this restriction.

**Table E-18 Base Order for Multiple Child Accounts**

| Action | Service    | Service Account | Billing Account | Billing Profile |
|--------|------------|-----------------|-----------------|-----------------|
| ADD    | Wireless 1 | BA1             | BA1             | BA1-BP1         |
| ADD    | Wireless 2 | SA1             | BA1             | BA1-BP2         |
| ADD    | Wireless 3 | SA2             | BA1             | BA1-BP3         |

Figure E-17 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-17 Result of Base Order for Multiple Child 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 Wireless 1 fails because it is using BP1.

Table E-19 shows the change order to change the paying parent for both child accounts using one billing profile under the new parent.

**Table E-19 Order to Change the Paying Parent for Multiple Child Accounts**

| Action | Service    | Service Account | Billing Account | Billing Profile |
|--------|------------|-----------------|-----------------|-----------------|
| UPDATE | Wireless 2 | SA1             | BA2             | BA2-BP4         |
| UPDATE | Wireless 3 | SA2             | BA2             | BA2-BP4         |

Figure E-18 shows the result of sending the change order to billing.

**Figure E-18 Result of Changing the Paying Parent for Multiple Child Accounts**



Because Wireless 1 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 child accounts.

When service-level balance groups are disabled, you cannot transfer child 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 child account
- A resubmitted version of the order to correct errors

Table E-20 shows the base order. Each service is assigned a separate billing profile.

**Table E-20 Base Order for Services with Separate Billing Profiles**

| Action | Service Number | Service Account | Billing Account | Billing Profile |
|--------|----------------|-----------------|-----------------|-----------------|
| ADD    | Wireless       | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband      | SA1             | BA1             | BA1-BP2         |

When the order is sent to billing, the integration creates two nonpaying **/billinfo** objects in BRM:

- NPB1 under the child account (SA1) pointing the primary billing profile (BP1) on the parent account (BA1)
- NPB2 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-20, 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-15.

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 the wireless service succeeds but the order for the broadband service fails.

The integration fails to process the broadband service because it is attempting to point the nonpaying **/billinfo** object for the broadband service to a billing profile different than the default account-level billing profile.

Table E-21 shows the revision order to resolve the failure or data mismatch by assigning the broadband service to BP1.

**Table E-21 Resubmitted Order**

| Action | Service Number | Service Account | Billing Account | Billing Profile |
|--------|----------------|-----------------|-----------------|-----------------|
| ADD    | Wireless       | SA1             | BA1             | BA1-BP1         |
| ADD    | Broadband      | SA1             | BA1             | <b>BA1-BP1</b>  |

After the order is processed to billing, the result is the same as in Figure E-15.

Though the revised order is processed successfully, the nonpaying **/billinfo** object for the broadband service (NPB2) 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 child account to be paid by the same parent, so all services of a child must be transferred to the new parent together. Because NPB2 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 nonpaying **/billinfo** objects and resubmit the change order.



# F

## 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

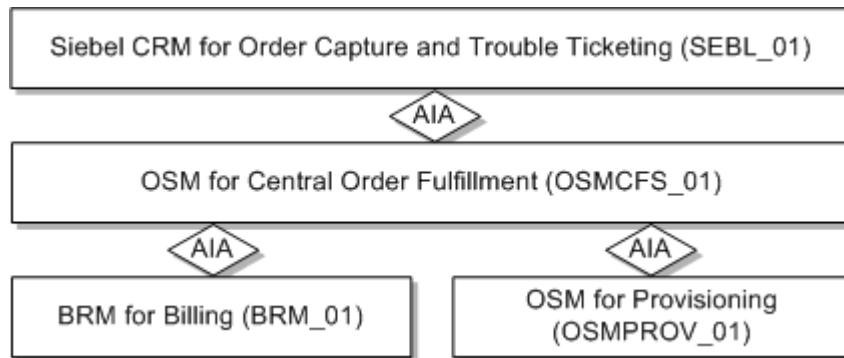
Each system instance is identified in Oracle AIA by a unique identifier, called a system code. The system codes help Oracle AIA identify the source or destination of a message.

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.   |
| OSMCSF_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\_OVERRIDETYPECODE.dvm  
ACCOUNTBALANCEADJUSTMENT\_STATUS.dvm  
RATEDATA\_DISCOUNTBRACKETCODE.dvm  
COLLECTION\_ACTIONNAME.dvm  
PRICE\_BILLINGPERIODCODE.dvm  
ACCOUNTBALANCEADJUSTMENT\_REASON.dvm  
SALESORDER\_LINESTATUS.dvm  
RATEPLAN\_TAXAPPLICABILITYCODE.dvm  
COLLECTION\_SUBSTATUS.dvm  
RECEIVEDPAYMENT\_TYPE.dvm  
ADDRESS\_COUNTRYID.dvm  
CUSTOMERPARTY\_PAYPROFILE\_DELIVERYPREF.dvm  
RESOURCE.dvm  
INSTALLEDPRODUCT\_STATUS.dvm  
ACCOUNTBALANCEADJUSTMENT\_USAGEALLOCATION\_TAXTREATMENT.dvm  
MULTIPLE\_DISCOUNT\_PER\_EVENT.dvm  
CUSTOMERPARTY\_PAYPROFILE\_PAYMETHODCODE.dvm  
COLLECTION\_PRIORITY.dvm  
CUSTOMERPARTY\_TYPECODE.dvm  
CUSTOMERPARTY\_PAYPROFILE\_PAYTERMCODE.dvm  
CUSTOMERPARTY\_STATUSCODE.dvm  
PRICETYPE\_EVENT.dvm  
STATE.dvm  
ACCOUNTBALANCEADJUSTMENT\_SUBSTATUS.dvm  
SALESORDER\_ACTIONCODE.dvm  
PHONENUMBER\_TYPE.dvm  
ENTITY\_TO\_TARGET\_APPLICATION.dvm  
STOP\_DISCOUNTING.dvm  
PROVISIONING\_TAG.dvm  
PRICECHARGETYPEUOM.dvm  
COLLECTION\_STATUS.dvm  
PLANTIER\_RESTRICTIONTYPE.dvm

APPLIES\_TO.dvm  
PRICE\_DISCOUNTTYPECODE.dvm  
DAYS\_OF\_WEEK.dvm  
CUSTOMERPARTY\_PAYPROFILE\_CREDIT\_CARDTYPE.dvm  
DISCOUNT\_VALIDITY.dvm  
RATEDATA\_CANCELATIONPRORATIONCODE.dvm  
LINEPRICE\_TYPECODE.dvm  
CUSTOMERPARTY\_BILLPROFILE\_BILLTYPECODE.dvm  
RATEPLAN\_TYPECODE.dvm  
PRICECHARGETYPE.dvm  
CUSTOMERPARTY\_ACCOUNTTYPECODE.dvm  
TARGET\_ID.dvm  
CURRENCY\_CODE.dvm  
RATEPLAN\_UNITCODE.dvm  
CUSTOMERPARTY\_BILLPROFILE\_FREQUENCYCODE.dvm  
PRICE\_IMPACTCLASSIFICATIONCODE.dvm  
SALESORDER\_STATUS.dvm  
RATEDATA\_PURCHASEPRORATIONCODE.dvm  
ACCOUNTBALANCEADJUSTMENT\_TYPE.dvm  
CUSTOMERPARTY\_PAYPROFILE\_BANKACCOUNTTYPE.dvm  
ADDRESS\_COUNTRYSUBDIVID.dvm  
GL\_CODE.dvm  
PRICE\_MARKUPBASISCODE.dvm  
ITEM\_BILLING\_TYPE\_CODE.dvm  
PLANIMPACT\_UNITCODE.dvm  
ITEM\_BILLINGTYPECODE.dvm  
PARTIAL\_PURCHASE.dvm  
LINEPRICE\_UOMCODE.dvm  
PROVINCE.dvm  
TROUBLETICKET\_AREA.dvm

- b. Once all the columns have been added, load the DVMs to the Metadata Services (MDS) repository using the update deployment plan.

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 port*. For example:
    - ip example.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

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>
```

- b. Update the **AIAConfigurationProperties.xml** file in the MDS repository as described in the discussion of uploading changed files to the Oracle Metadata Services repository in *Oracle AIA Installation Guide*.

## Creating a New Consumer for Product Synchronization

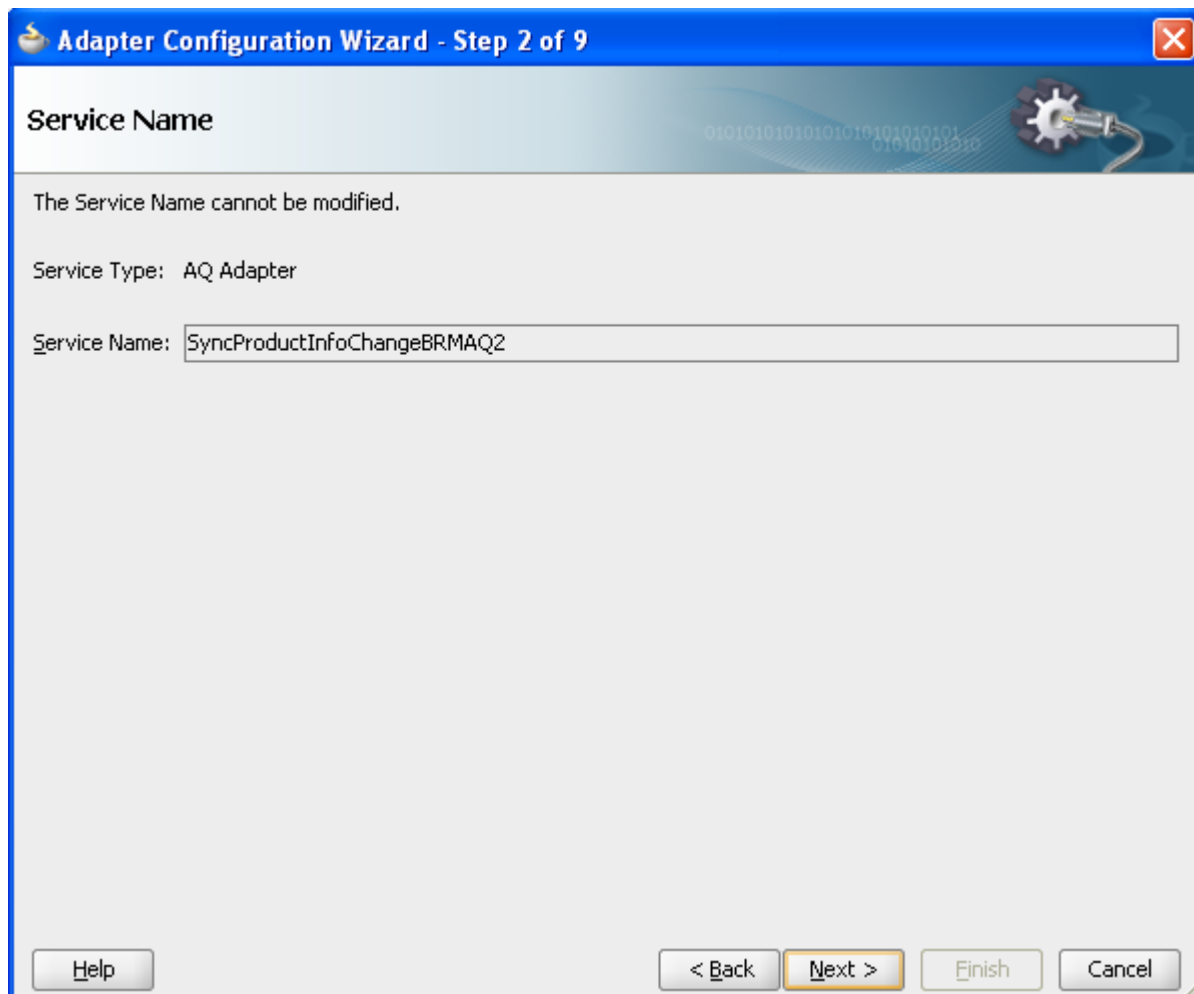
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for product synchronization:

### To create a consumer for product sync:

1. Create a new SOAPProject 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**



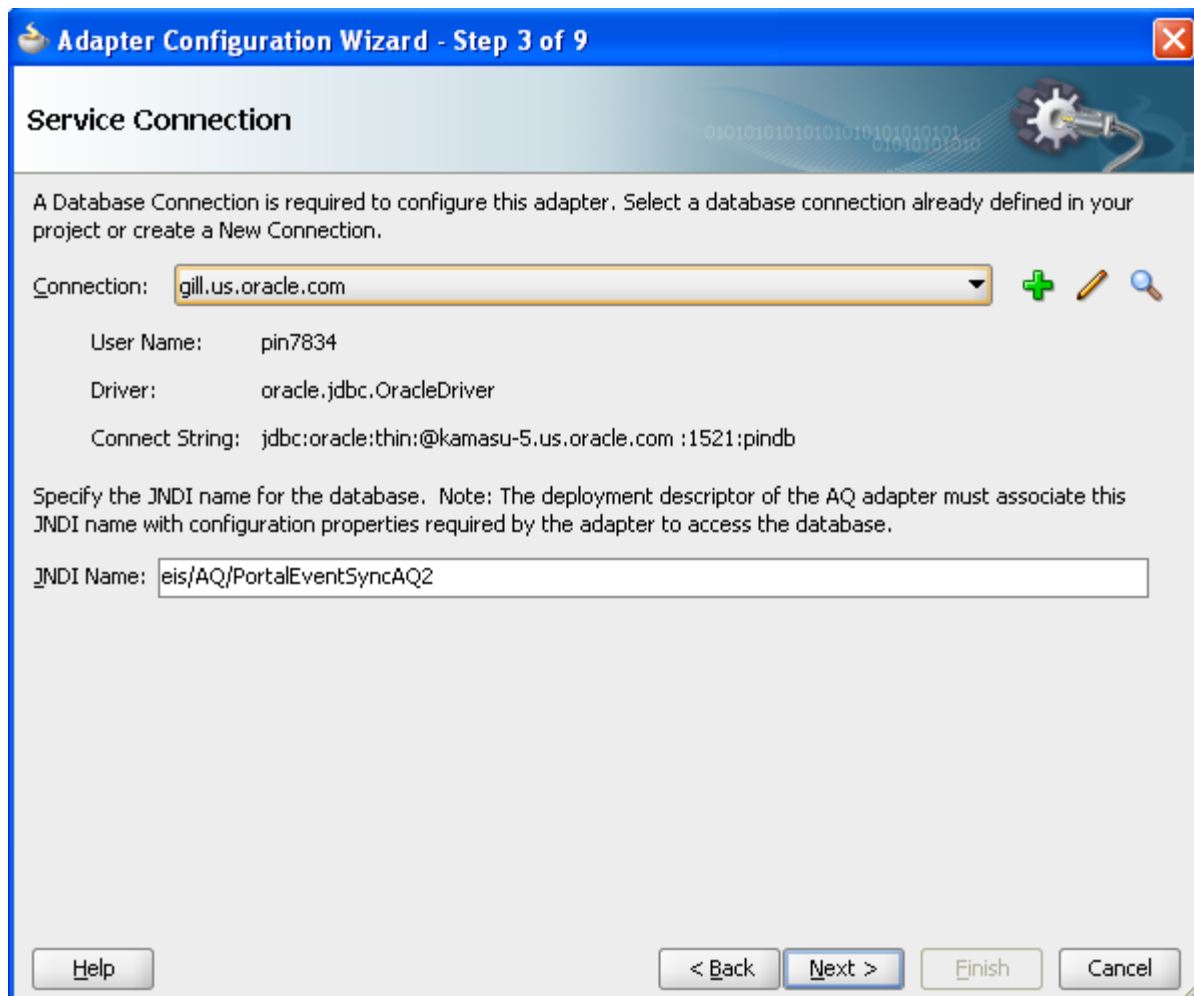
4. Click **Next** to display Step 3 (Service Connection).

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**

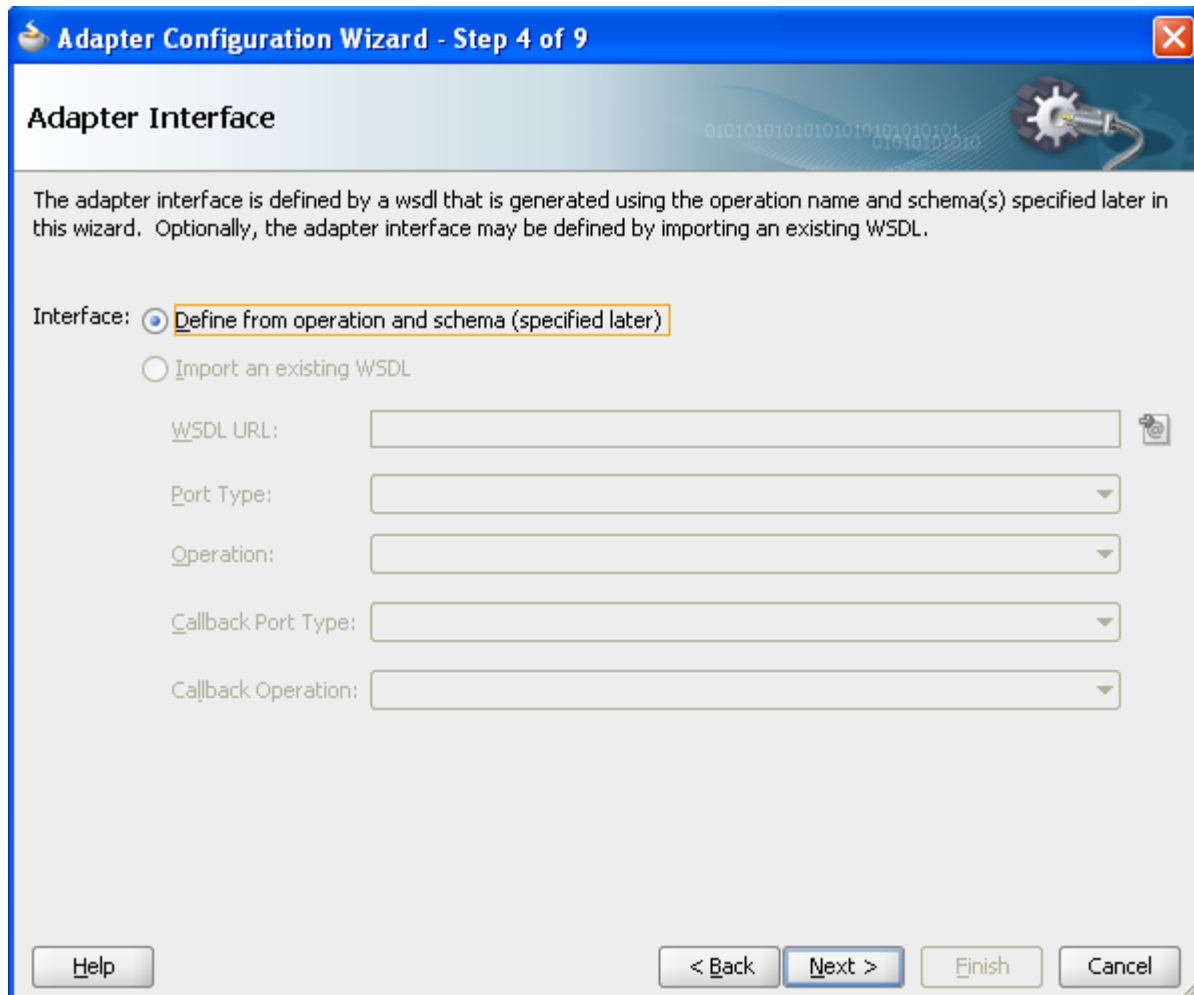


5. Click **Next** to display Step 4 (Adapter Interface).

Go to the **Interface** field and select **Define from operation and schema (specified later)**, as shown in [Figure F-4](#).

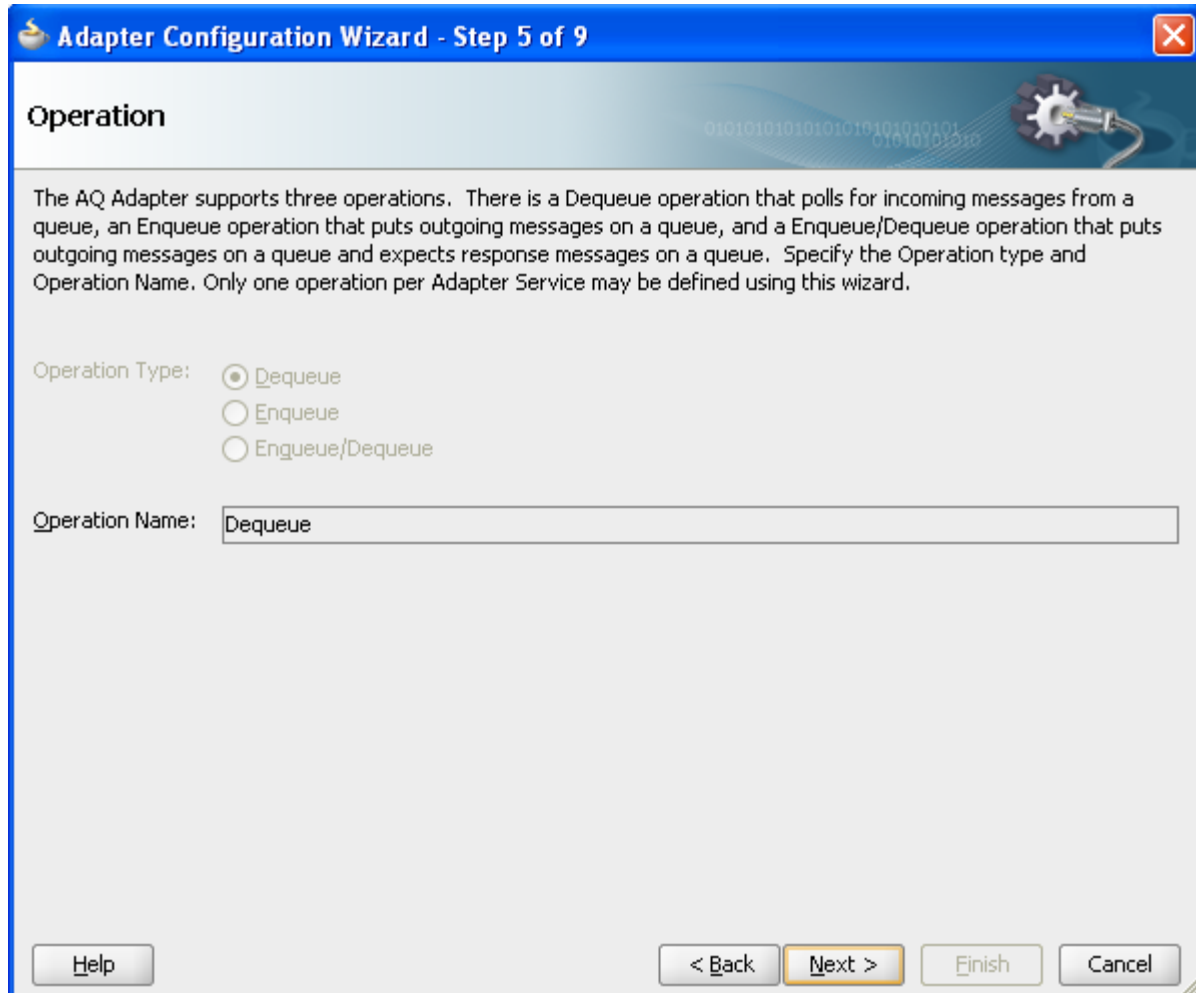


Figure F-4 Adapter Configuration Wizard - Step 4 of 9



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



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

Adapter Configuration Wizard - Step 6 of 9

### Queue Name

Specify the database schema and the queue to be used for the service. Use the Browse button to find the queue in the specified schema.

Browse

Queue Information

Database Schema: PIN7834

Queue Name: AQQPIN7834

Help < Back Next > Finish Cancel

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:

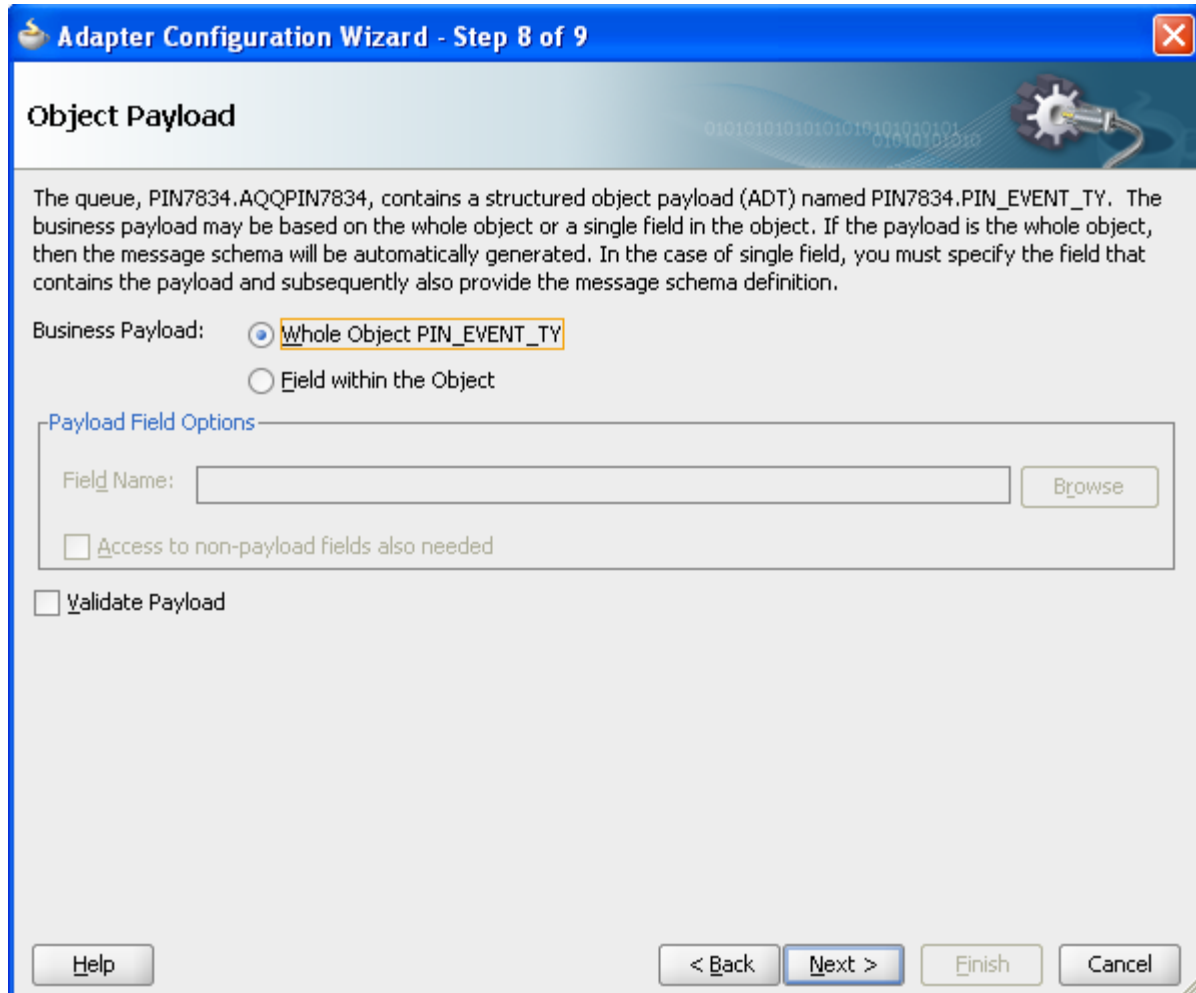
Dequeue Condition

Help < Back Next > Finish Cancel

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



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 SyncProductBRMCommsReqABCSEImpl 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/RequesterABCSE/
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/ABCSEImpl/BRM/Industry/Comms/
SyncProductBRMCommsReqABCSEImpl/V1#wSDL.endpoint(SyncProductBRMReqABCSEImpl/
SyncProductBRMReqABCSEImpl)" location="http://<hostname>:<Port>/soa-infra/services/
```

```
default/SyncProductBRMCommsReqABCSEImpl/SyncProductBRMCommsReqABCSEImpl?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

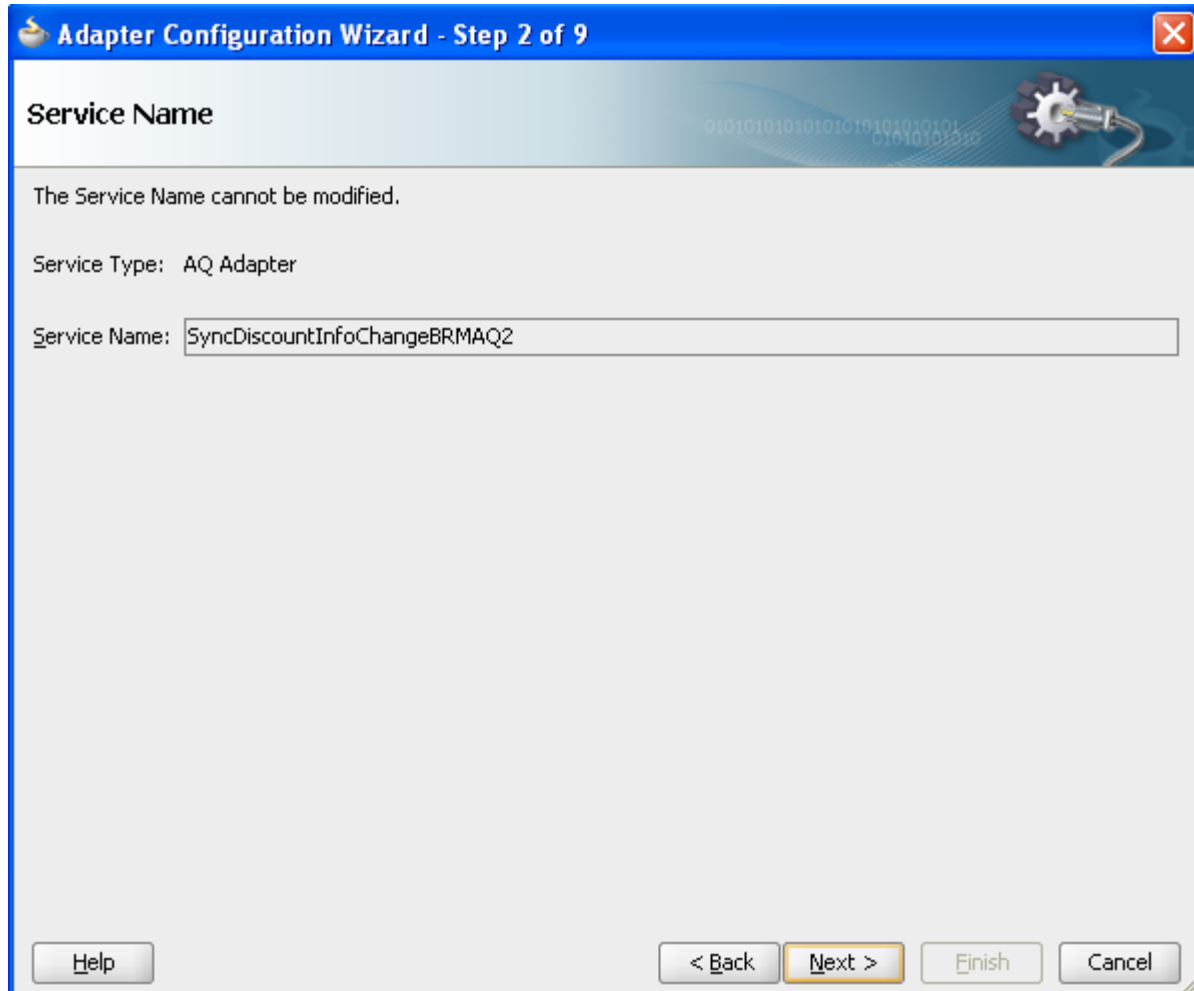
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for discount synchronization:

**To create a consumer for product sync:**

1. Create a new SOAPProject 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



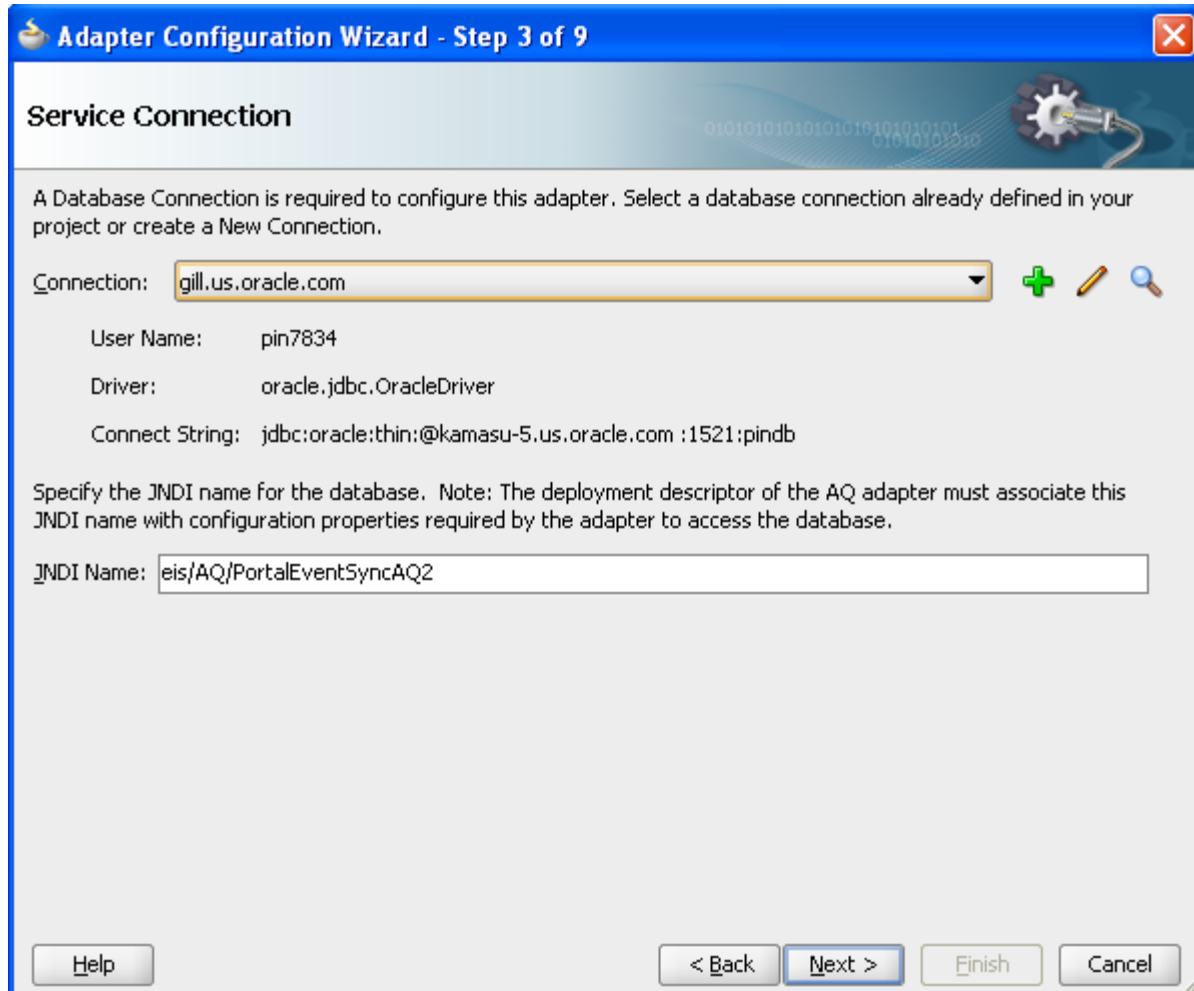
4. Click **Next** to display Step 3 (Service Connection).

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

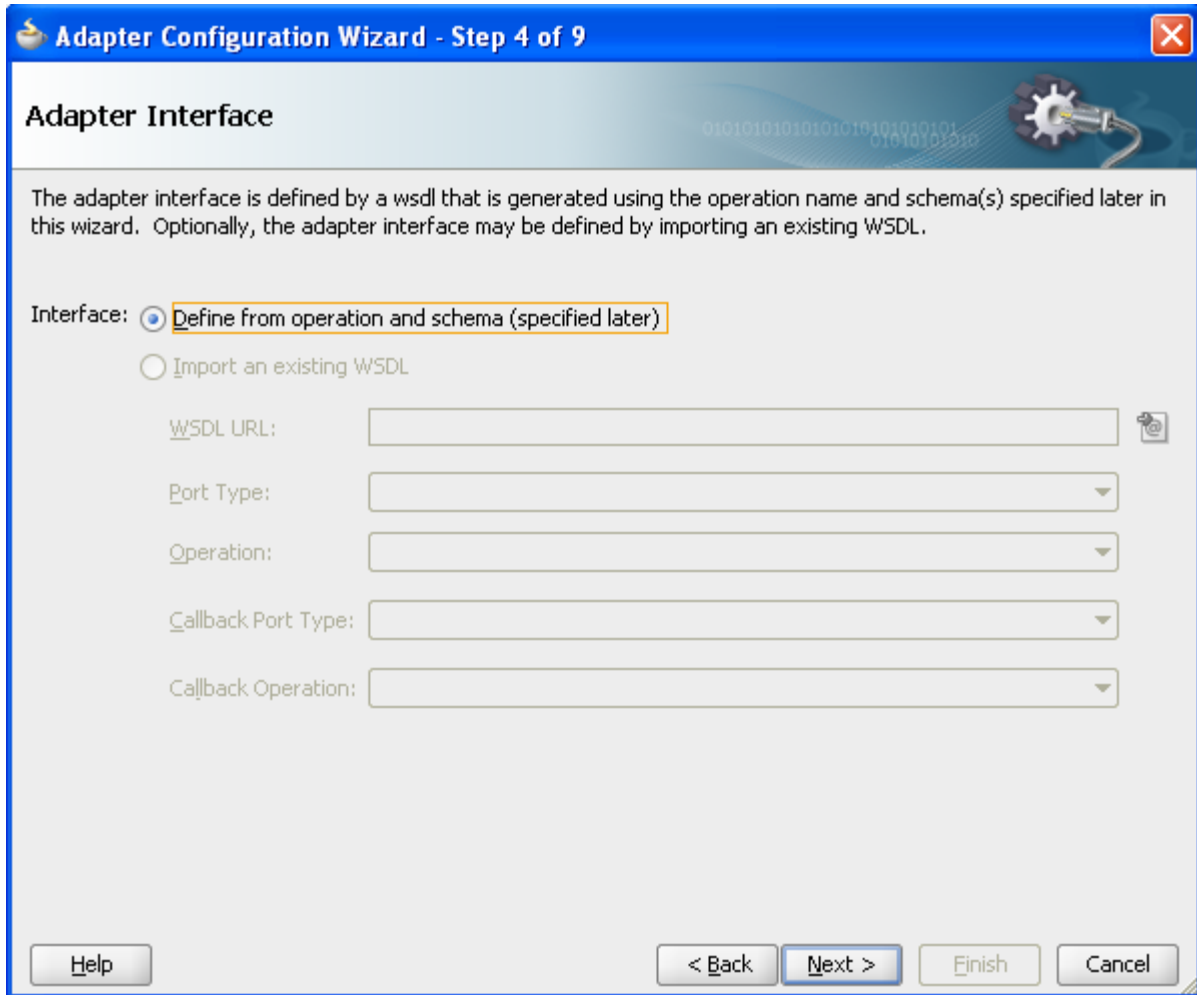


5. Click **Next** to display Step 4 (Adapter Interface).

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

**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 a 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

Adapter Configuration Wizard - Step 6 of 9

### Queue Name

Specify the database schema and the queue to be used for the service. Use the Browse button to find the queue in the specified schema.

Browse

Queue Information

Database Schema: PIN7834

Queue Name: AQQPIN7834

Help < Back Next > Finish Cancel

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:

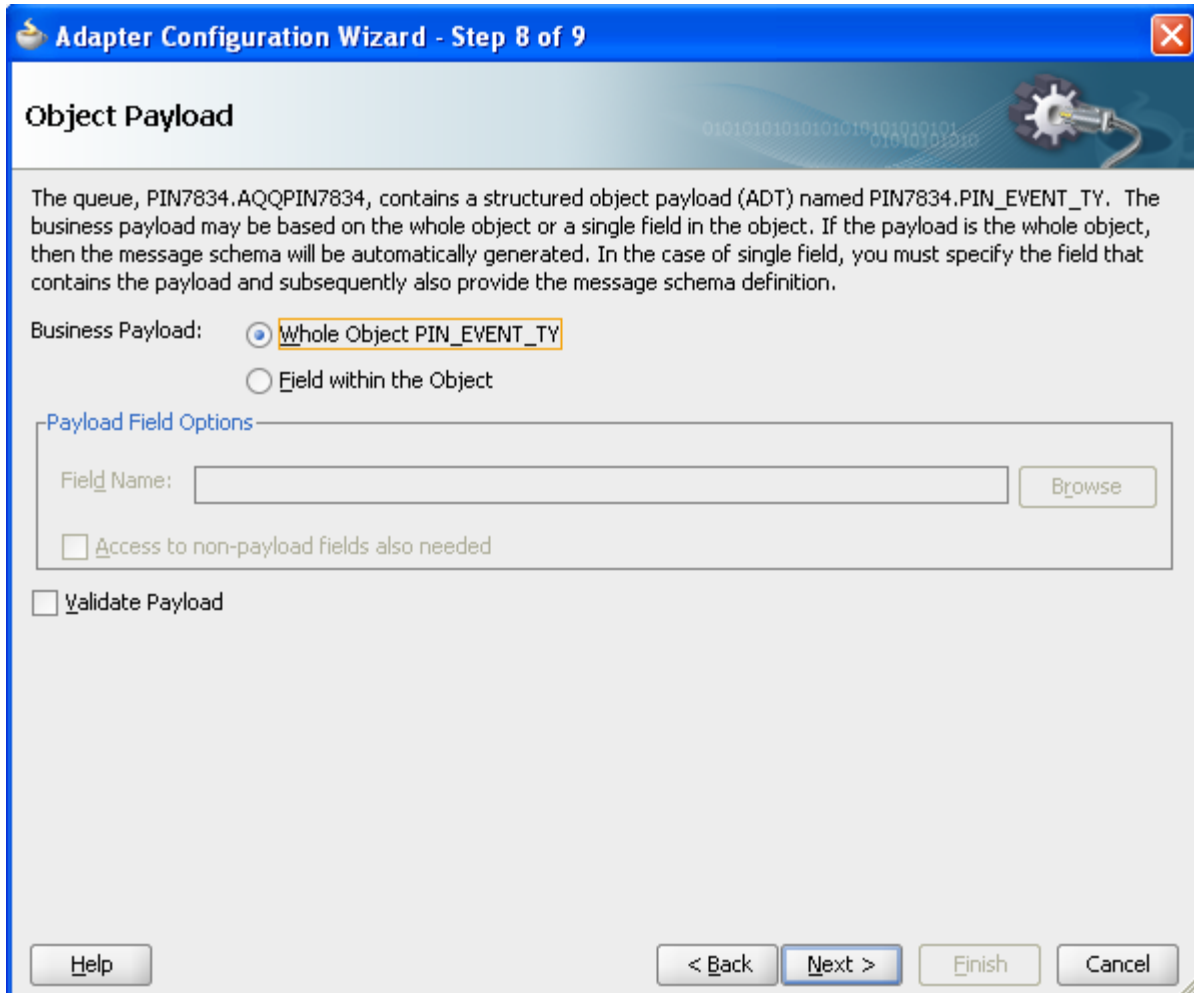
Dequeue Condition

Help < Back Next > Finish Cancel

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



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 SyncDiscountBRMCommsReqABCServiceImpl 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/SyncDiscountBRMCommsReqABCImpl/SyncDiscountBRMCommsReqABCImpl?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

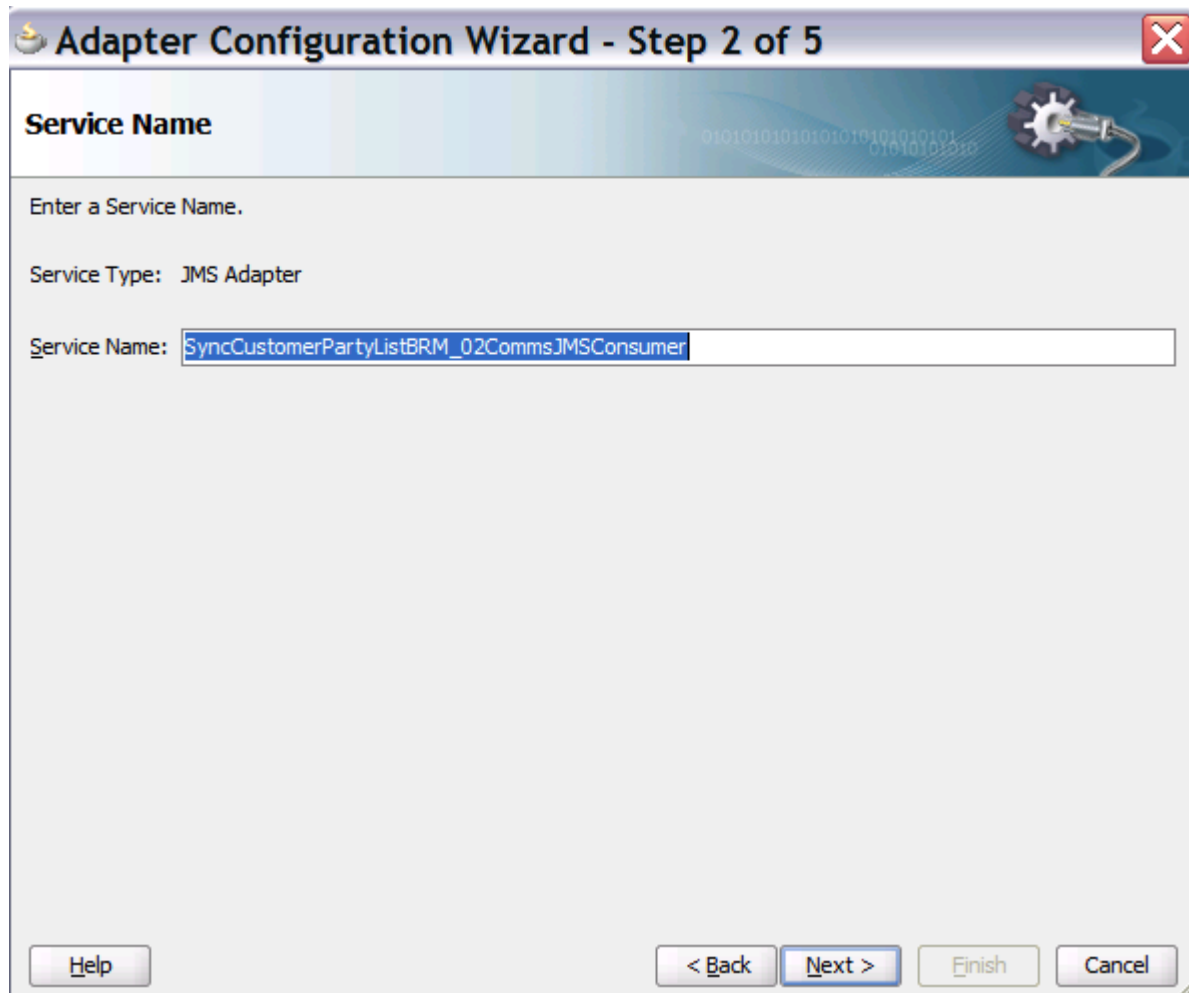
The following steps must be followed to create a consumer for every new BRM instance. This consumer is used for customer updates:

**To create a consumer for customer updates:**

1. Create a new SOAPProject 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](#).

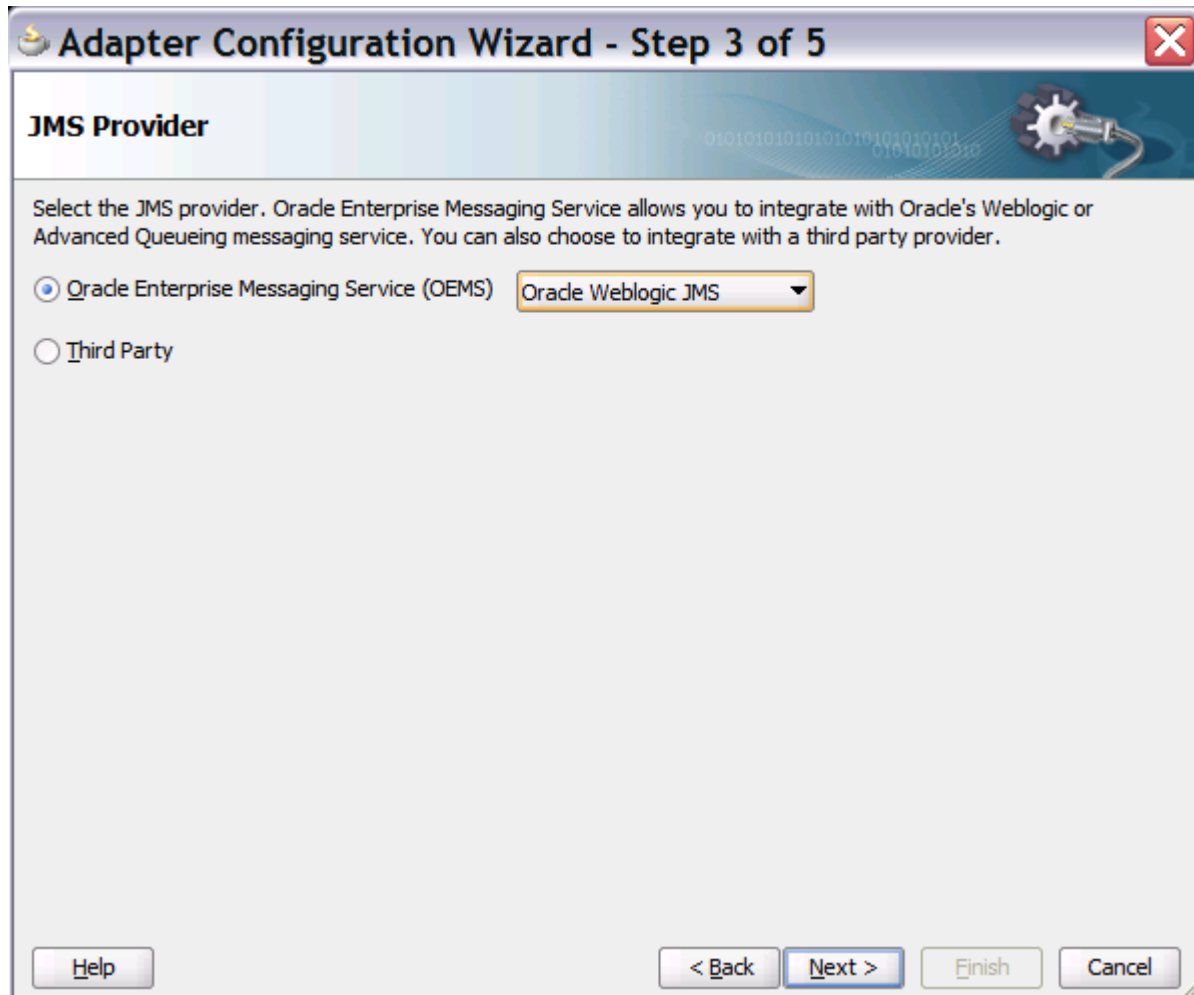
Figure F-16 Adapter Configuration Wizard - Step 2 of 5



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

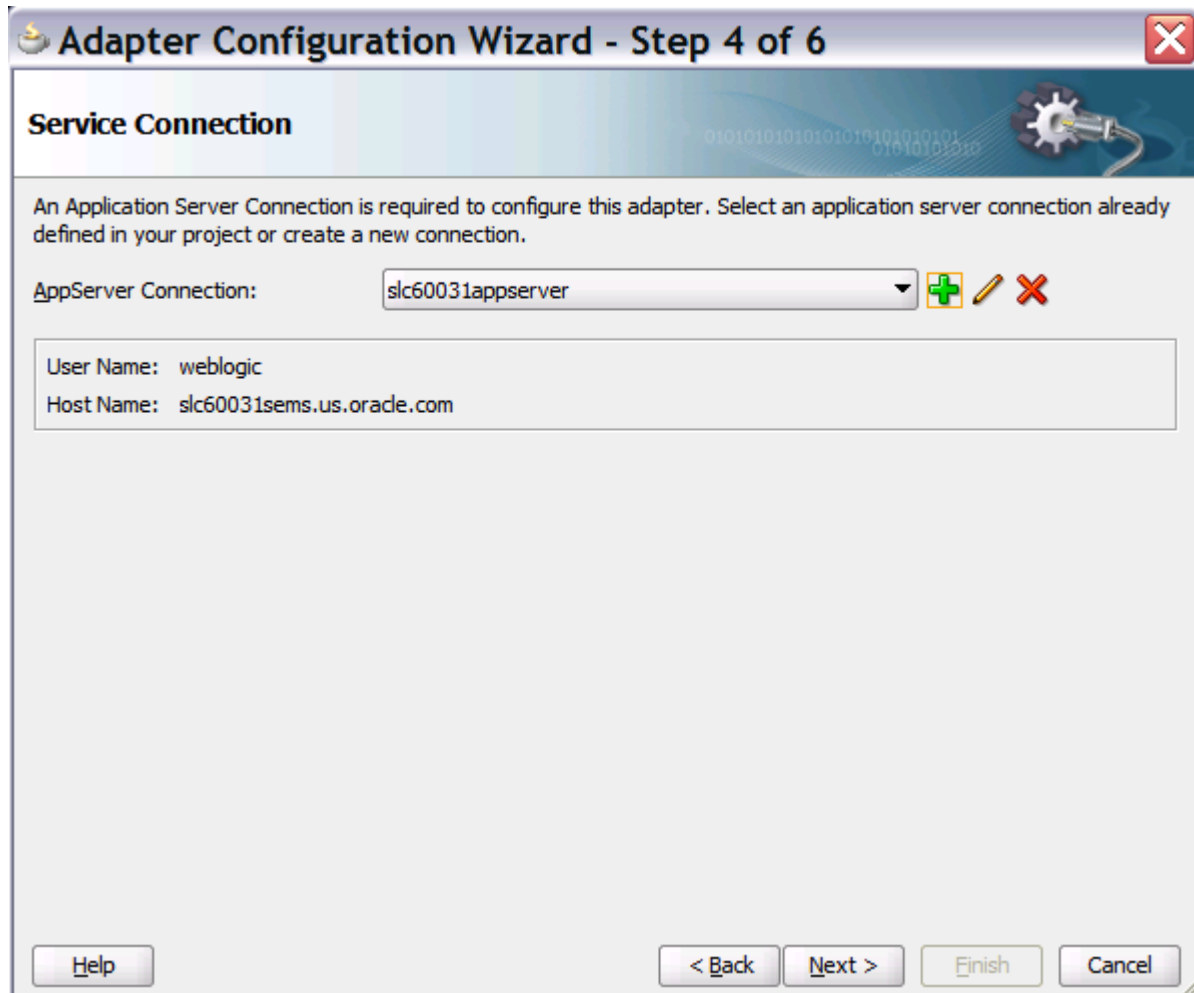


5. Click **Next** to display Step 4 (Service Connection).

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



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:

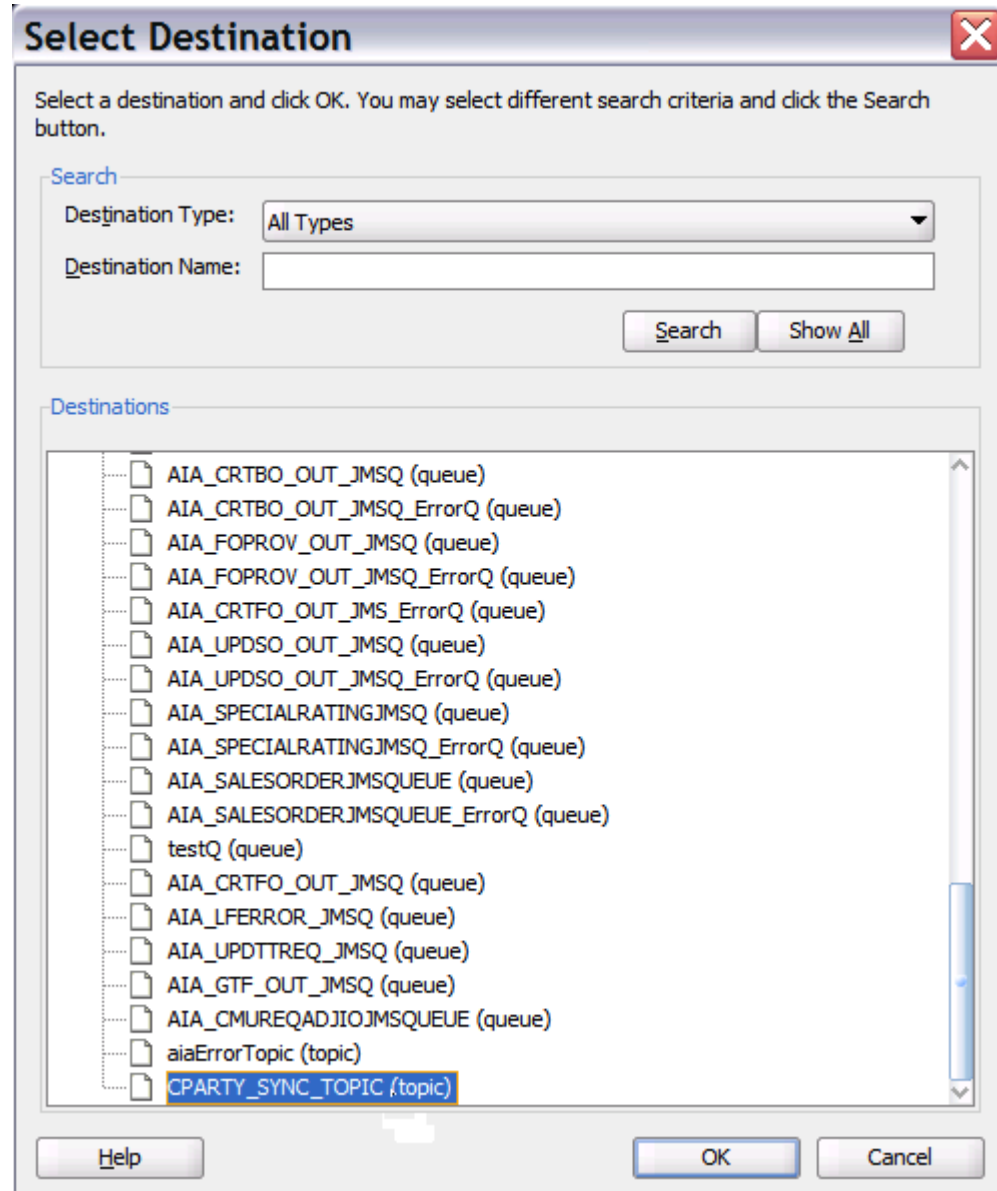
Durable Subscriber ID:

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`.

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('orams:/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.

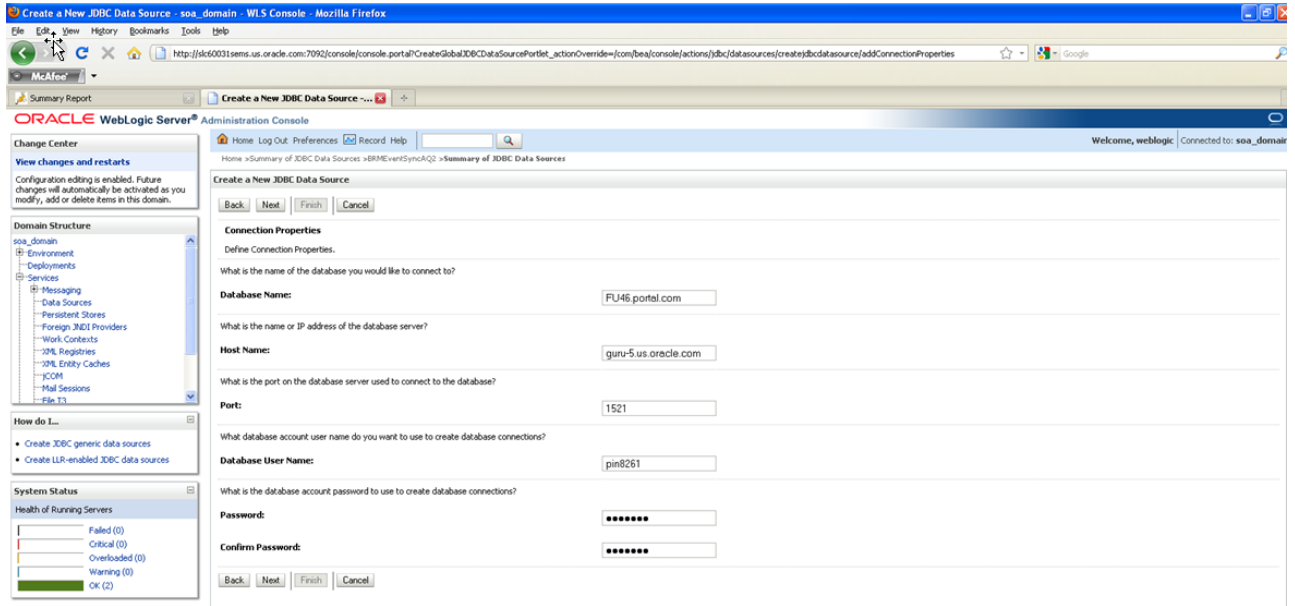
## Creating a Data Source and Connection Factory

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-23](#).

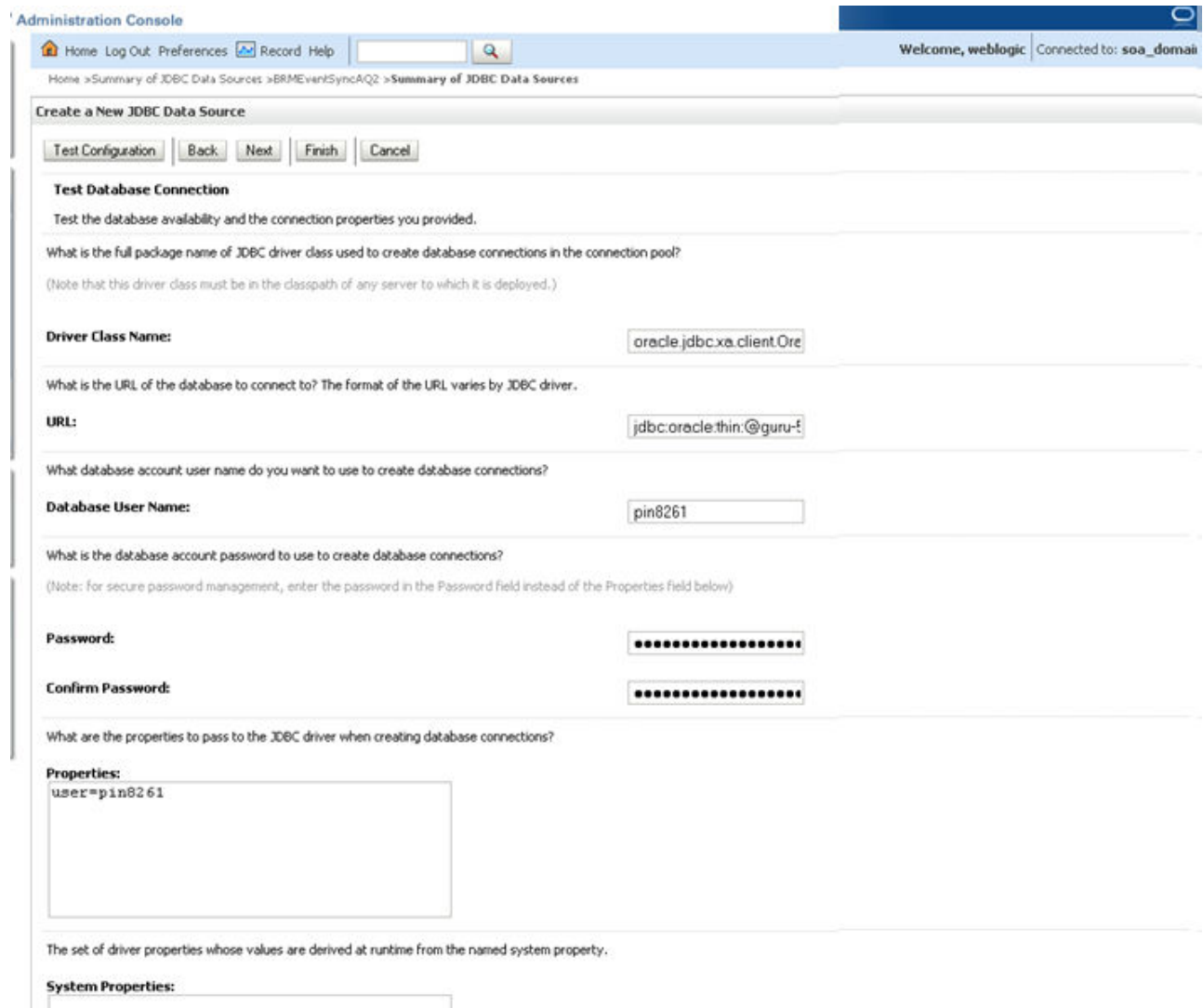
Figure F-23 WebLogic Server - Administration Console - Connection Properties



Enter all the BRM connection properties and then

4. Click **Next** to display the Test Configuration page, as shown in [Figure F-24](#).

Figure F-24 WebLogic Server - Administration Console - Test Configuration



Click **Test Configuration** to verify that the provided details are correct. If the test is successful, click **Finish**.

5. Navigate to **Deployments** and click **AQAdapter**.
6. Go to the Configuration tab and expand the connection factory. Click **New** and select the connection factor.
7. Click **Next**. Provide the JNDI name as *eis/AQ/PortalEventSyncAQ2*.
8. Click **Finish**.
9. Click the newly created connection factory and go to the **XADataSourceName** field and enter *jdbc/aia/BRMEventSyncAQ2*. Click **Enter** and then click **Save**.

The same process must be followed for each additional 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-25](#) 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-25](#) in the AIA\_SYSTEM table.

**Figure F-25 Logical Instance Example**

|        |        |                 |                   |                                |     |     |
|--------|--------|-----------------|-------------------|--------------------------------|-----|-----|
| BRM_02 | BRM_02 | BRM Instance 01 | gil.us.oracle.com | http://gil.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-25](#), 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**

| <b>Typical Topology</b>                                    |
|--|
| 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**

| <b>Reserved Topology</b>                                   |
|--|
| 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**

| <b>Simple Topology</b>  |
|---|
| 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>
```

# G

## 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.

# H

## 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, including updates to billing account, billing profile, service account, and owner account:
  - 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 (`ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer`) 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: UpdateSalesOrderOSMCFSC ommsJMSConsumer.</p> | <p><b>Note:</b> 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> |



**Table I-1 (Cont.) Queues and Flows Enabled for Sequencing**

| Oracle AIA Queue     | Flow   | JMS Priority  | Sequencing Criteria  | Comments  |
|----------------------|--|---|--|---|
| AIA_CRTCUST_OUT_JMSQ | Order flow from OSM to Oracle AIA for customer data creation in billing. | Set by OSM  | Group By: Account ID on the message (this is either the Billing account or the Service account on the order line that must be created in billing) and the target system identifier.<br><br>concat(\$in.SyncCustomerPartyListEBM/<br>ns0:SyncCustomerPartyListEBM/ns0:DataArea/<br>ns0:SyncCustomerPartyList/<br>ns0:CustomerPartyAccount/<br>corecom:Identification/<br>corecom:ApplicationObjectKey/<br>corecom:ID[@schemeID='AccountId'], \$in.SyncCustomerPartyListEBM/<br>ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/<br>corecom:Target/corecom:ID)<br><br>Order of Processing: FIFO (First in First Out).<br><br>Composite Name: CommunicationsCustomerPartyEBSV2Resequencer. | 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   | Group By: AccountID.<br><br>Order of Processing: FIFO (First in First Out).<br><br>Composite Name: SyncAcctSiebelAggrEventConsumer<br>SyncContSiebelAggrEventConsumer.   | Also available in the Agent Assisted Billing Care pre-built integration.<br><br>The resequencer in this flow ensures that multiple updates for the same customer are processed in the right sequence. |
| AIA_CRTFO_IN_JMSQ    | Order flow from Oracle AIA to OSM.                                       | 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. You can use ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer 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  |

**Table I-1 (Cont.) Queues and Flows Enabled for Sequencing**

| Oracle AIA Queue    | Flow  | JMS Priority   | Sequencing Criteria  | Comments |
|---------------------|---|--|--|----------|
| 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 ProcessProvisioningOrderOSMCFSCommsJMSConsumer 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 ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer 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.

# J

## 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}}:{{http portname}}/AIAComponents/PIPS/Communications/Schemas/OrderFailureData.xsd>

The fields defined in the xsd and how they must be used are listed here.

- ApplicationFaultData / OrderFailureData / OrderID  
BusinessComponentID - SalesOrder / Provisioning Order / Fulfillment Order / Fulfillment Billing Order  
ID - SalesOrder # / Provisioning Order # / Fulfillment Order # / Fulfillment Billing Order # (If available)

ApplicationObjectKey - If available  
 AlternateObjectKey - SALESORDER\_ID  
 AlternateObjectKey - SALESORDER\_NUMBER  
 AlternateObjectKey - SALESORDER\_REVISION  
 AlternateObjectKey - FULFILLMENTSYSYSTEM\_ID

[Example J-2](#) is a sample definition.

- ApplicationFaultData / OrderFailureData / AccountID  
 BusinessComponentID - CommonAccountID  
 ID - Account Name  
 ApplicationObjectKey - Siebel AccountID (required only with SalesOrder EBM)

[Example J-3](#) is a sample definition.

- 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.

- 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.

- 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.

**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>
```

**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>
```

**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>
```

**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>

```

## 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.



# K

## 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, upload the changed **`AIAConfigurationProperties.xml`** to the Oracle Metadata Services repository as described in *Oracle AIA Installation Guide*.

# L

## 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 L-1](#) shows an example of the MIGRATED\_OBJECTS\_T table data.

**Table L-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 BRM object types shown in [Table L-2](#). The table shows the cross-reference mapping between the Oracle AIA cross-reference identifier and BRM cross-reference object types.

**Table L-2 Cross-Reference Mapping Between Oracle AIA and BRM**

| AIA Cross-Reference                     | BRM Cross-Reference  |
|---|--|
| CUSTOMERPARTY_CONTACTID                 | <b>/account</b> POID   |
| CUSTOMERPARTY_BILLPROFILEID             | <b>/billinfo</b> POID  |
| CUSTOMERPARTY_PAYPROFILEID              | <b>/payinfo</b> POID   |
| CUSTOMERPARTY_ADDRESSID                 | <b>/account</b> POID<br>or<br><b>/payinfo</b> POID   |
| CUSTOMERPARTY_CONTACTID                 | <b>/account</b> POID<br>or<br><b>/payinfo</b> POID   |
| CUSTOMERPARTY_DEFAULTBALANCEG<br>ROUPID | <b>/balance_group</b> POID<br>This is the default account level balance group.   |
| INSTALLEDPRODUCT_ID                     | <b>/purchased_product</b> POID   |
| INSTALLEDPRODUCT_ID                     | <b>/purchased_discount</b> POID  |
| INSTALLEDPRODUCT_ID                     | <b>/service/object</b> POID<br>Where <i>object</i> is the service object for the account.  |
| INSTALLEDPRODUCT_ID                     | <b>/purchased_bundle</b> POID  |
| INSTALLEDPRODUCT_ID                     | <b>/profile/object</b> POID<br>Where <i>object</i> is the profile object for the account.  |
| INSTALLEDPRODUCT_ID                     | <b>/group/sharing/object</b> POID<br>Where <i>object</i> is the type of sharing group. Possible values are <b>discounts</b> , <b>charges</b> , and <b>profiles</b> . |

## Synchronizing Cross-References for Migrated Accounts

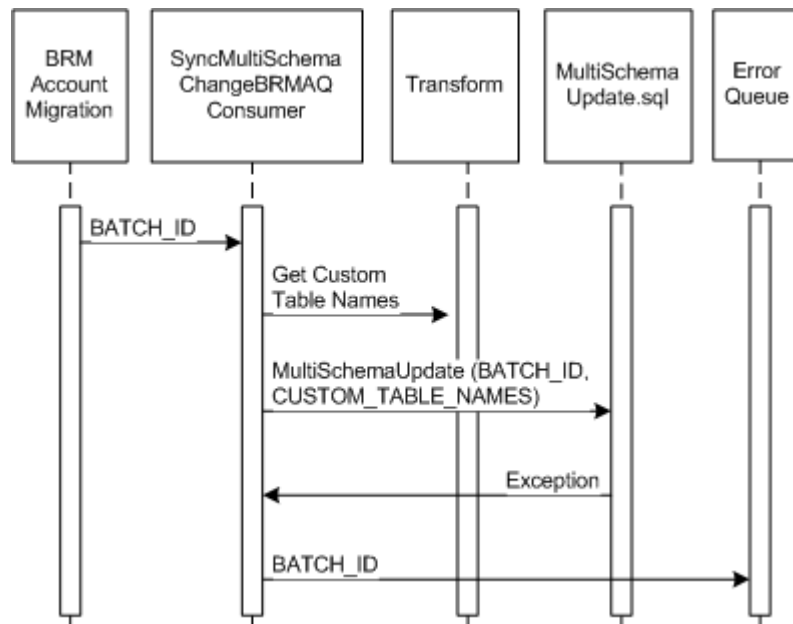
When you migrate accounts across BRM schemas, the cross-references for the accounts are synchronized from BRM to Oracle AIA as follows:

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. Calls the MultiSchemaUpdate PLSQL procedure.
  - c. 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.

Table L-2 illustrates the flow when updating cross-references in a multischema migration.

**Figure L-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, where *host*, *port*, and *sid* are the host, port, and SID for your environment:

```
create database link brm01 connect to brm identified by brm using
'(DESCRIPTION =
  (ADDRESS =
    (PROTOCOL = TCP)
    (Host = host)
    (Port = port)
  )
(CONNECT_DATA = (SID = sid)
)');
```

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 BRM cross-reference object types, see "[Mapping BRM POIDs to AIA Cross-References](#)".
  - b. Open the `Comms_home\source\soainfra\apps\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 L-2](#) 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. Invokes **MultiSchemaUpdate.sql**.

3. 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.