Oracle® Communications Digital Business Experience

Order to Cash Implementation Guide

Release 26.4
G47940-01
May 2026

ORACLE

Oracle Communications Digital Business Experience Order to Cash Implementation Guide, Release 26.4

G47940-01

Contents

3 Request to Change Business Process

4 Termination to Confirmation Business Process

5 Implementing the Order to Cash Business Processes

6 Configuring the Order to Cash Business Process

7 Using the Order to Cash Reference Library

About This Content

This document describes the business process, feature set and implementation of Order to Cash.

Audience

This document is intended for:

- Customer Service Representatives
- Individuals who are responsible for configuring, managing, and maintaining tasks and process flows for the Order to Cash business process.

You should be familiar with the TMF ODA Order to Cash business process and its flows.

1

Order to Cash Implementation Overview

The Order to Cash business process covers the complete lifecycle of customer onboarding, ordering, asset management and monetization. This guide describes how to implement the Order to Cash business process in which you perform run-time tasks related to order processing, fulfillment and billing.

About Order to Cash

The Order to Cash journey starts after a product offering is launched in the preceding Business Proposal-to-Launch journey and a sales order is received. It covers all process flows that begin after an order for a product offering is received.

The Order to Cash business processes are aligned with the Telemanagement Forum (TMF) Open Digital Architecture (ODA) business architecture.

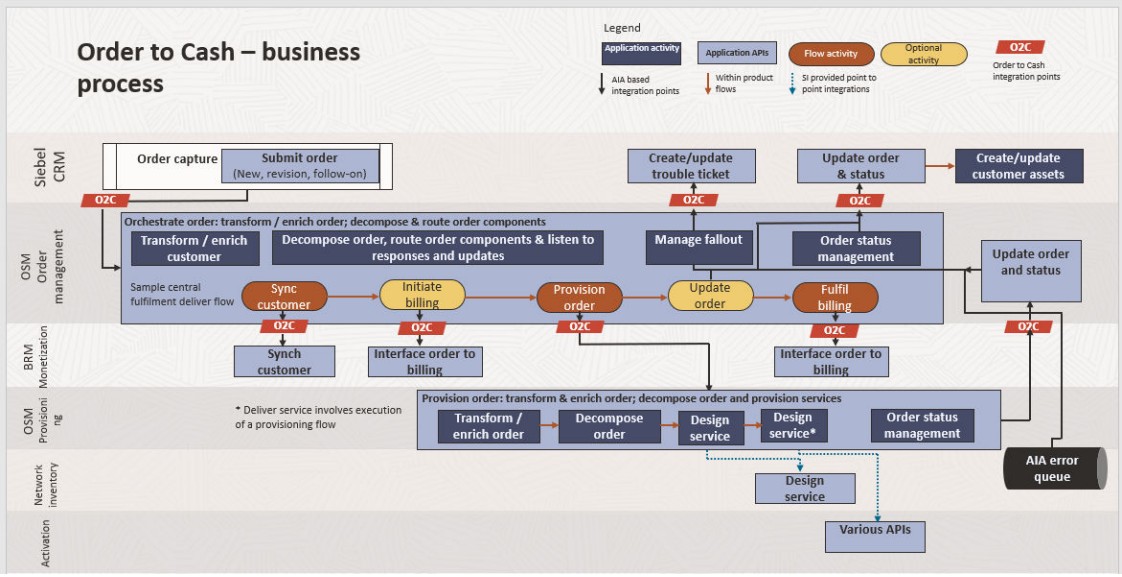
The solution supports the following processes, which are realized by the functional capabilities and integration provided within and across Siebel CRM, Oracle Communications Order and Service Management (OSM) and Oracle Communications Billing and Revenue Management (BRM):

- Order to Payment
- Request to Change
- Termination to Confirmation

The solution also uses Oracle Communications Application Integration Architecture (AIA), which is an integration framework, to integrate flows across Siebel CRM, OSM and BRM.

Figure 1-1 depicts the flow of Order to Cash journey.

Figure 1-1 Overview of Order to Cash



The chapters that follow provide conceptual and procedural information about tasks related to order processing, fulfillment and billing.

2

Order to Payment Business Process

This chapter describes the Order to Payment business process and features.

Overview of the Order to Payment Business Process

The Order to Payment business process encompasses activities related to order capturing, processing, fulfilling, and billing. This process deals with capturing the subscriber information, creating the order with suitable products and services as per their requirements, fulfilling the order to provision the required services and generating bills for the same.

The Order to Payment business process includes support for the following features:

- Subscriber Onboarding
- First-time Purchase of Products and Services
- Support for Special Rating (Friends and Family List) Products
- Support for Family Share Plans with Differential Pricing
- Support for Orders with Zone-Based Usage Pricing
- Support for Multiple Pricelist and Currencies
- Applying Discounts as per the Discount Matrix
- Promotion Component Discount
- Time-Based Offers
- Override of Price and Discounts
- Offer Aggregation
- Compatibility Rules
- Eligibility Rules
- Support for Bulk Order
- Order Fallout Management

The sections that follow describe these features.

About Subscriber Onboarding

The Subscriber Onboarding feature creates subscriber data in BRM from order data while processing orders as part of order lifecycle management.

The journey of a subscriber’s purchase begins when they are prospects targeted by CSPs. This process starts with targeted marketing efforts to raise awareness about unique offerings and gain interest. Prospects then explore their options by reviewing detailed plan information, trial offers, and customer support resources. After a decision is made, prospects provide their details to create accounts with the CSP, becoming subscribers. Subscribers can then place orders for the desired services via Agent assistance. Ongoing follow-ups address concerns, ensure satisfaction, and foster long-term loyalty through personalized engagement strategies.

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 subscriber.

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 hierarchy relationships.

This feature creates subscriber data in BRM when you submit orders for processing and is supported for orders from Siebel CRM.

See About the Synchronize Fulfillment Order Billing Account Flow for more information about where subscriber synchronization fits in to order fulfillment.

About Create/Sync New Subscriber Account

Figure 2-1 illustrates the Create/Sync New Subscriber Account flow.

Figure 2-1 Create/Sync New Subscriber Account

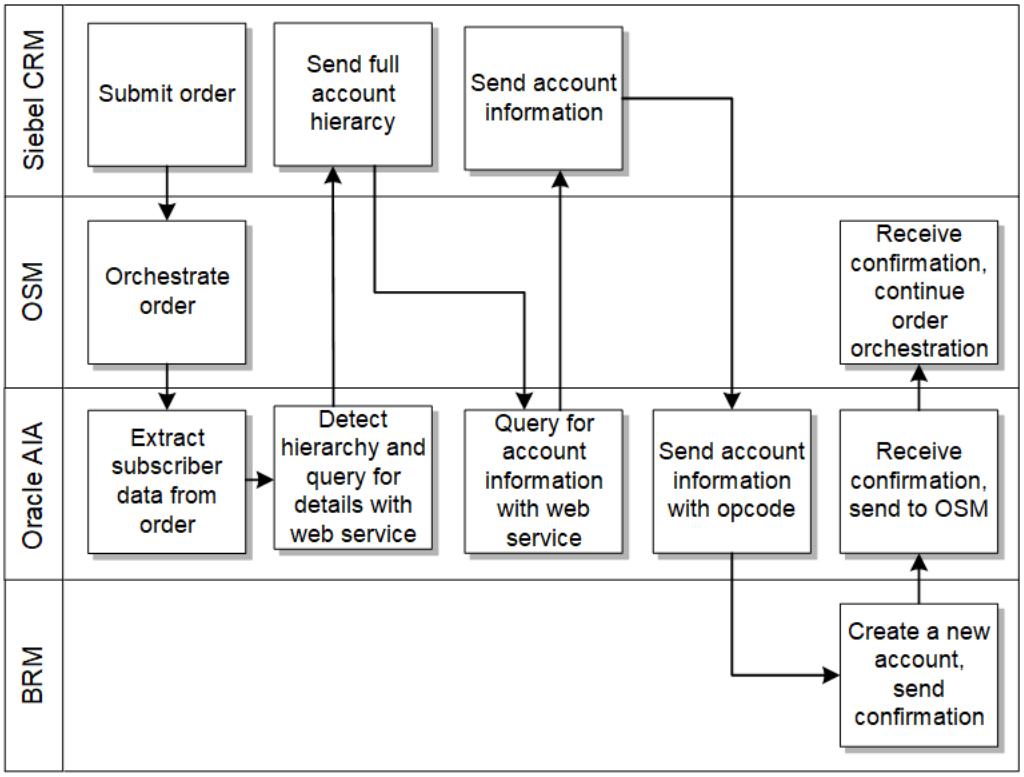


Table 2-1 provides information on Siebel CRM attributes mapped to BRM as part of the Create/Sync New Subscriber Account flow.

Table 2-1 Siebel Entities Created or Synchronized to BRM

Entity or Attribute in the Siebel CRM	Entity or Attribute in BRM	Notes
Account	**Account**	BRM sets the account status to **Active** by default. If the billing account and service account on the order line are different, the Order to Cash business process creates a creates a **/billinfo** 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.
--	Account Number	Oracle AIA sets this to the Common ID.
Account Type	Account Type	Valid Siebel CRM Account Type values are **Residential** and **Business**. Uses the CUSTOMERPARTY_TYPECODE DVM.
Name	Company Name	Only set for **Residential** Account Types.
Currency	Currency	Uses the CURRENCY_CODE DVM.
Contact	--	The Order to Cash business process 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 Order to Cash business process maps different Siebel CRM phone number types (home, work, fax, mobile) to BRM Phone Type and Number using the PHONENUMBER_TYPE DVM. The phone number format should match the supported format in BRM. See "Using BRM with Oracle Application Integration Architecture", "Validating Customer Contact Information" in Oracle Communications Billing and Revenue Management for more information about phone number formats.
Job Title	Job Title	--
Email	Email	--
Address	--	The Order to Cash business process synchronizes the account's primary address to BRM.
Address	Address	In addition to Address, fields for City, State, Postal Code, and Country are mapped. Uses the following DVMs: ADDRESS_COUNTRYID, ADDRESS_COUNTRYSUBDIVID, PROVINCE, STATE.
Billing Profile	**BillInfo**	--
Name	Name	--
Frequency	Billing Frequency in Months	Uses the CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE DVM

Table 2-1 (Cont.) Siebel Entities Created or Synchronized to BRM

Entity or Attribute in the Siebel CRM	Entity or Attribute in BRM	Notes
--	Currency	Integration passes account-level currency. Uses the CURRENCY_CODE DVM
Billing Schedule	Billing Day of Month	If the Billing Schedule is not set in and sent from Siebel CRM, then BRM defaults the Billing Day of Month. See "Setting Business Policies for Billing" in *Oracle Communications Billing and Revenue Management Configuring and Running Billing Guide* for more information about the billing schedule.
--	**PayInfo**	--
Payment Method	Payment Method	Only **Bill Me**, **Credit Card** or **Auto-Debit** is supported. Uses the CUSTOMERPARTY_PAYPROFILE_PAYMETHODECODE DVM.
Contact Last Name, First Name	Name	When the payment method is Bill Me, the Contact Name on the Siebel Billing profile is mapped to BRM PayInfo Contact Name. When the payment method is Credit Card or Auto-Debit, either the Credit Card owner name or Debit Account name is mapped to BRM PayInfo Contact Name.
Bill Media	Delivery Preference	Applicable only when the payment method is Bill Me. Uses the CUSTOMERPARTY_PAYPROFILE_DELIVERYPREF DVM.
Email Bill To	Email Address	Applicable only when the payment method is Bill Me.
Address	Address	In addition to Address, fields for City, State, Postal Code, and Country are mapped. Uses the following DVMs: ADDRESS_COUNTRYID, ADDRESS_COUNTRYSUBDIVID, PROVINCE, STATE.
Credit Card #	Credit Card Number	Applicable only when the payment method is Credit Card.
Expiration Month & Year	Credit Card Exp	Applicable only when the payment method is Credit Card.
Security Code	Security ID	Applicable only when the payment method is Credit Card.
Account #	Debit Num	Applicable only when the payment method is Auto-Debit.
Bank Routing #	Bank No	Applicable only when the payment method is Auto-Debit.
Bank Account Type	Type	Applicable only when the payment method is Auto-Debit.

Data Requirements for Creating a New Subscriber Account

The Order to Cash business process requires the following data to successfully create subscriber data in BRM:

- Account Type: **Residential**
- 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's 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 subscriber 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 the following for more information about expectations for order management systems other than OSM

- Table 5-3
- Communications Orders Dictionary
- Expectations from an Order Management System for Billing Integration

About Subscriber Management

Subscriber Management lets you synchronize subscriber information to BRM from Siebel CRM.

You create and update subscriber data in Siebel CRM and the Order to Cash business process synchronizes the account information to BRM during order processing. This is a one-way synchronization process; changes made to subscribers in BRM are not synchronized to Siebel CRM.

The process does not load subscriber data in bulk into BRM. Instead, it synchronizes new accounts and billing information to BRM while processing the first order for those accounts.

It also synchronizes updates to the accounts from Siebel CRM to BRM.

About Synchronizing Subscriber Account

When synchronizing subscriber account information, subscriber data is created in BRM from order data while processing orders as part of 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 subscriber.

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 hierarchy relationships.

Synchronizing subscriber account lets you do the following:

- Create/Sync Subscriber Account: Creates subscriber data in BRM when you submit orders for processing. Supported for orders from Siebel CRM.

- Update Subscriber Account: Updates existing subscriber data in BRM when you when you update accounts in Siebel CRM.

See About the Synchronize Fulfillment Order Billing Account Flow for more information about where subscribers synchronization fits in to order fulfillment.

For more information about implementing this flow, see Implementing the Synchronize Subscriber Account Flow.

About the Update Subscriber Account

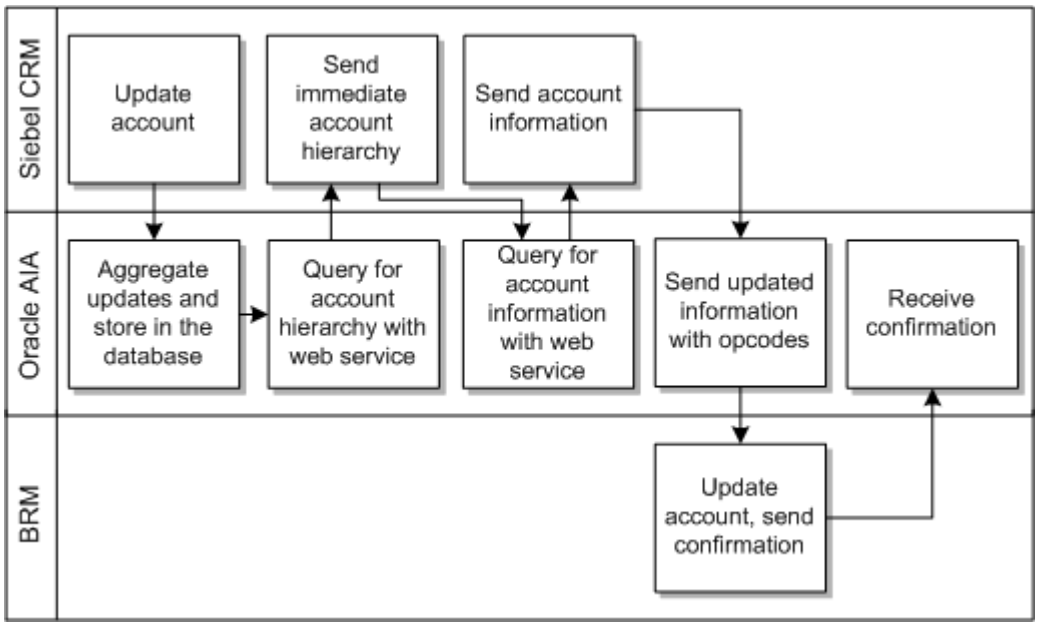
When subscribers call to change their account information, such as contact, payment, billing, and hierarchical information, a customer service representative (CSR) updates the accounts in Siebel CRM. The Order to Cash business process synchronizes subscriber account updates to BRM in real time through the Update Subscriber Account integration flow.

Updates are synchronized to BRM only for accounts that have already been created with Siebel CRM through the Create/Sync Subscriber 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 2-2 illustrates how updating a subscriber account works.

Figure 2-2 Update Subscriber Account



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 Cash to Care Business Process business process. Oracle recommends that you enable account status synchronization only if you are also using the process integration for collections management.

The Cash to Care Business Process synchronizes collections actions generated by BRM as credit alerts in Siebel CRM, where a CSR can take actions on the subscriber'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. You can also 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 subscriber 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 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 canceled 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: Cash to Care Business Process Implementation Guide* for more information about collections management.

About First-Time Purchase of Products and Services

The first-time purchase of products and services extends from the time a quote or order is created to the time when the goods and services are delivered and billed. Order to Cash 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 systems, such as supply chain management and workforce management, or with alternative billing and order management systems as an extension project at implementation time.

> **Note**
>
> First-time purchase is when a new customer is placing an order for products and services.

About Order Capture

The first-time purchase of products begins with the creation of a sales order to capture the particulars of the telecom order, which entails the services or assets requested by the subscriber, changes in service, or disconnection of services. It is captured and validated in Siebel CRM then submitted to OSM in the central order management (COM) role for fulfillment.

Figure 2-3 shows a typical order capture flow. The flow can vary depending on, for example, service family, subscriber 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 Subscriber Order: Validates the availability of a service design and the capacity to fulfill the subscriber order.
- Deliver Subscriber Order: Fulfills the products and services purchased by the subscriber or fulfills actions on existing subscriber assets.

Figure 2-3 Typical Order Capture Flow

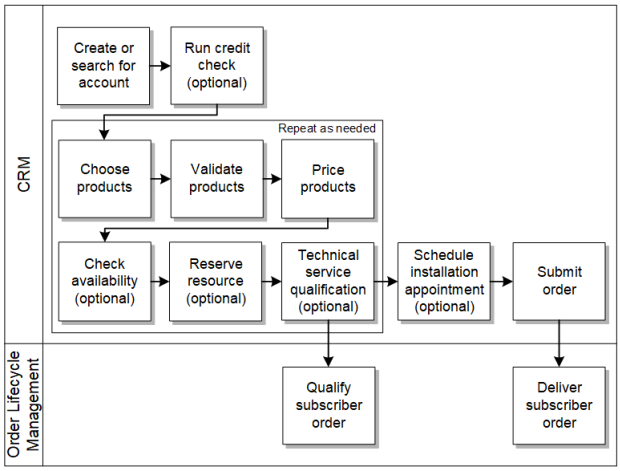


Figure 2-3 shows typical order activities for the CRM system and their integration points with the order lifecycle management activities.

A typical order capture progresses as follows:

1. A Customer Service Representative (CSR) creates a new subscriber account or searches for an existing subscriber account. CSRs can also capture subscriber 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 subscriber.
3. The subscriber 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 process 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 process starts.

About Sales Orders

Sales orders are orders that purchase products and services for subscribers. 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 subscriber 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 product offerings), 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/services**.

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 2-2 defines and maps the terms from Siebel CRM to OSM.

Table 2-2 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 About Supplemental Orders.

Table 2-2 (Cont.) Order Term Mapping

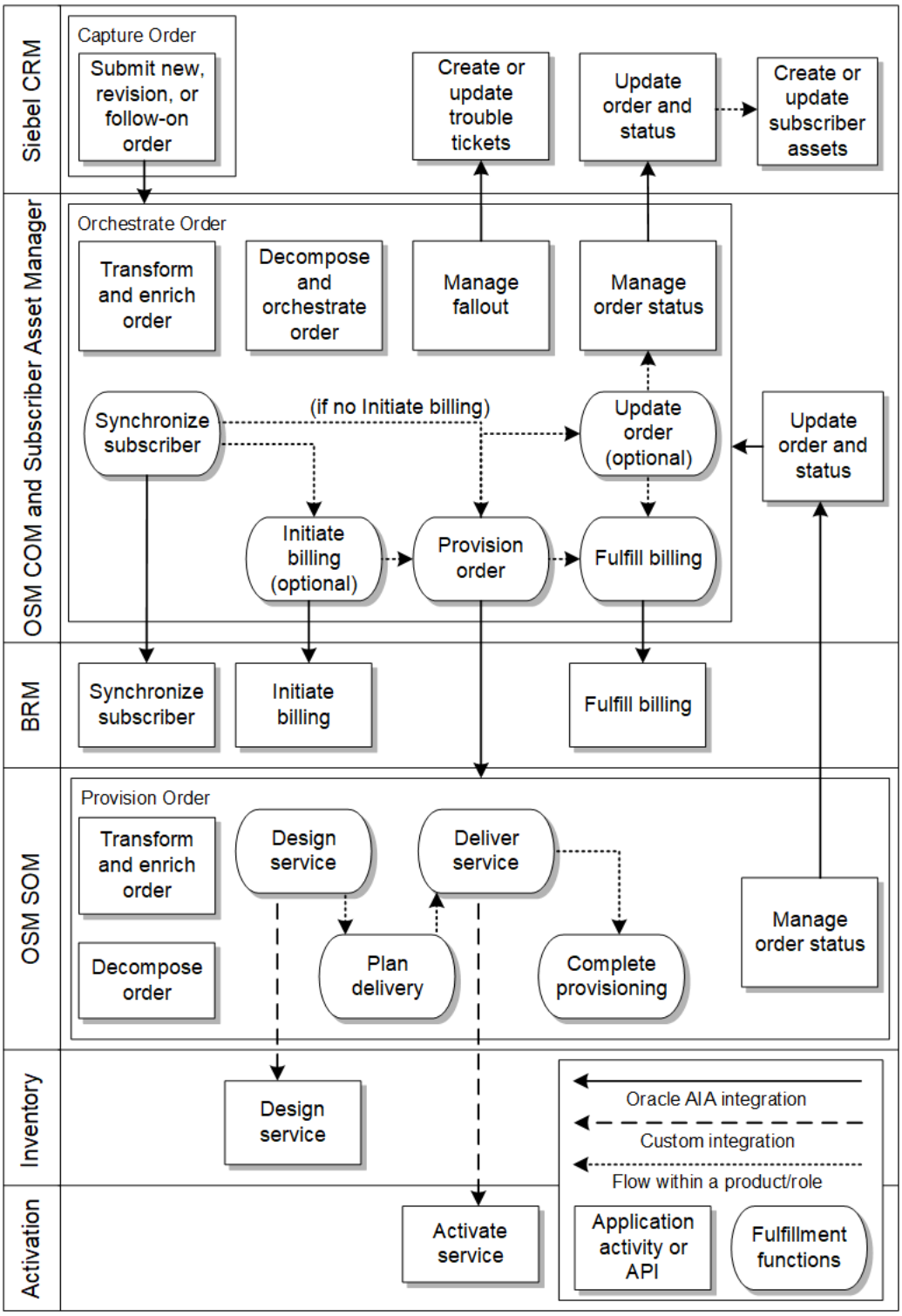
Siebel CRM Term	OSM Term	Description
Follow-on order	Follow-on order	A changed version of an in-flight/open order that has passed the point of no return. Fulfillment of the follow-on order waits until the fulfillment of the order item on which the follow-on order depends is complete. See About Follow-On Orders.
Modify order	--	An order to modify the attributes of assets on a completed order. There is no direct correlation 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 About Future-Dated Orders.

Sales orders are processed by the Process Sales Order Fulfillment Business Flow. For more information on implementing this flow, see Implementing the Process Sales Order Fulfillment.

About Deliver Subscriber Order

Figure 2-4 shows the typical application activities and interactions involved in delivering subscriber orders.

Figure 2-4 Deliver Subscriber Order



The integration delivers subscriber 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 pre-configured functions, such as synchronizing the subscriber 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 2-4. See the discussion of orchestration in Oracle Communications Order and Service Management Concepts for 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.

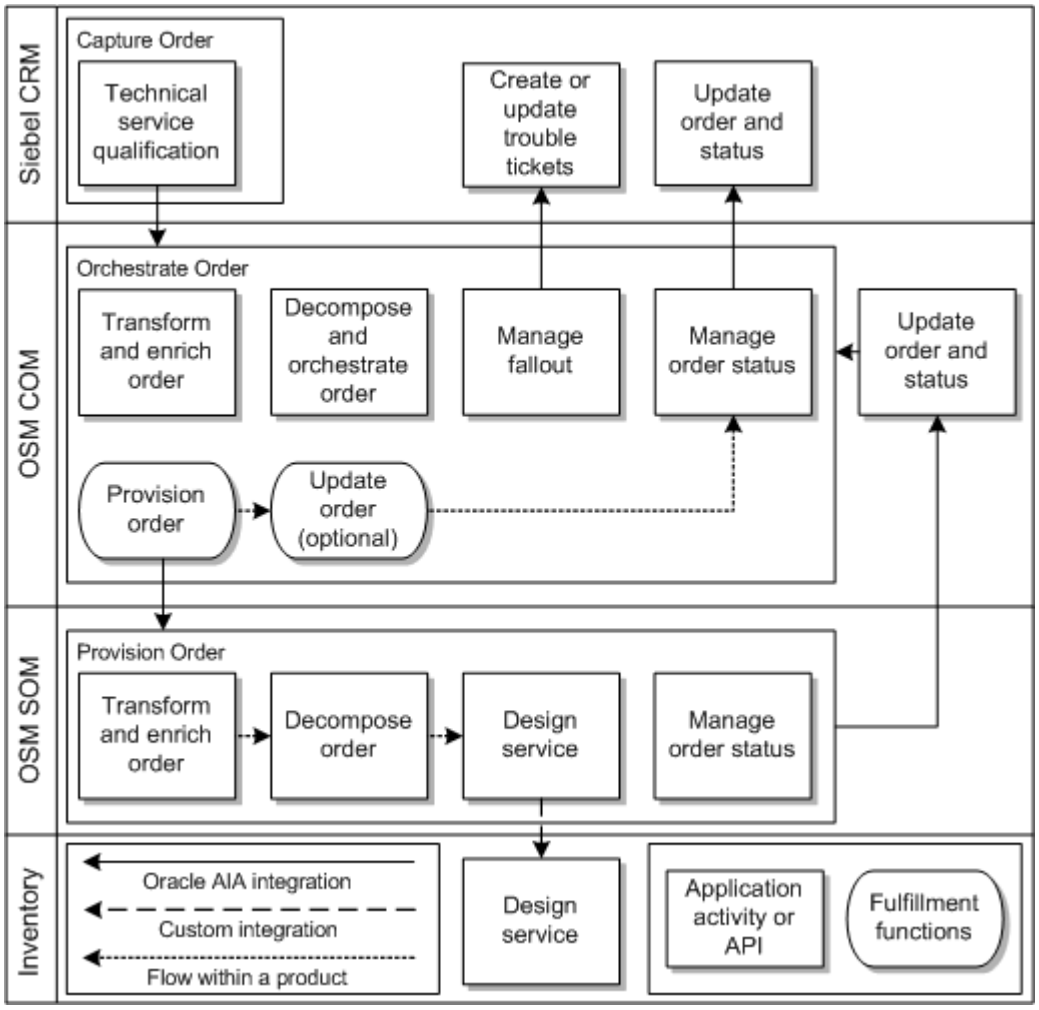
   See Using Error Type to Control Response to Order Fallout for more information.

d. Manages order status by mapping fulfillment function responses to common statuses. OSM updates Siebel CRM with relevant subscriber 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 Subscriber Order

Figure 2-5 shows the typical application activities and interactions involved in delivering subscriber orders.

Figure 2-5 Qualify Subscriber Order



This flow starts with a request to qualify the technical validity of a subscriber order submitted from Siebel CRM to OSM.

OSM performs the same functions described in Figure 2-5, except that the metadata and the fulfillment functions are for qualifying the subscriber order rather than delivering the subscriber 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.

For more information about defining products, see *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

About Mapping Orders to Fulfillment Functions

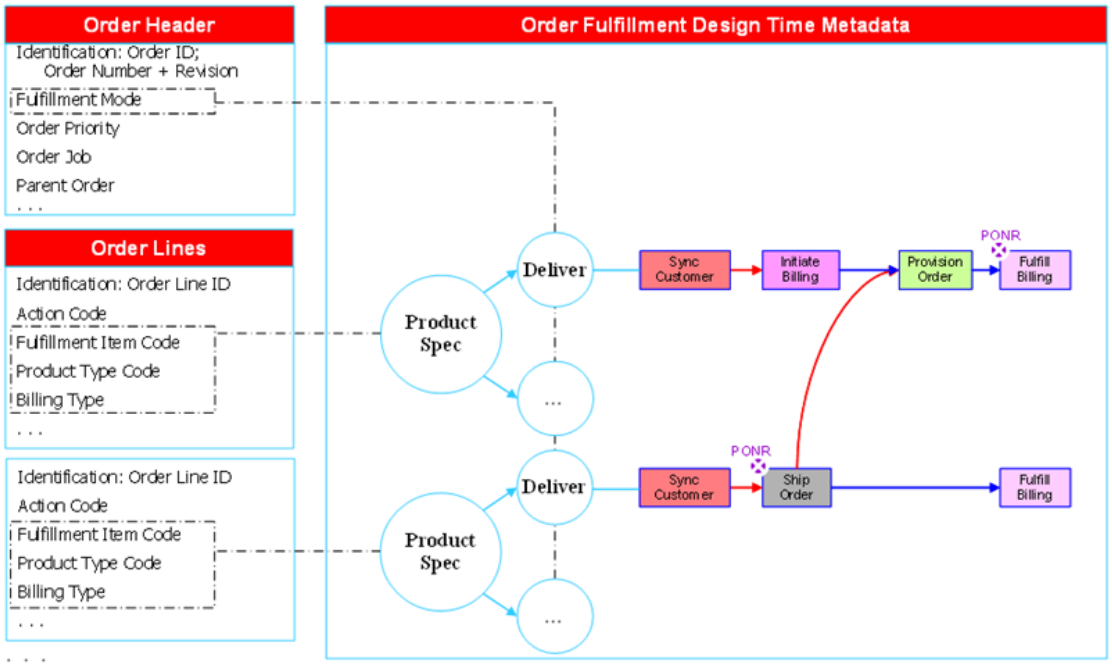
Order management systems act on subscriber orders, which are composed of order lines. Each order line is represented by an action and a subject. Actions are verbs that represent the nature of the subscriber request, such as ADD to purchase an offering or UPDATE to modify the subscription to an offering. 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 subscriber 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 2-6 illustrates how an order management system in the integration uses the product model to map customer order lines to fulfillment functions.

Figure 2-6 Mapping Order Lines to Fulfillment Functions



When a subscriber 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 flow. See *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for more information.

Data Requirements for Order Lifecycle Management

Orders must include the following data 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.

About Supporting Multiple Price Lists on Orders

At order creation in Siebel CRM, the price list assigned to the subscriber'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 when the price lists are of the same currency.

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 *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* 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 when these price lists are of the same currency. 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 2-3 shows an example of the order lines for a new Siebel CRM order. The table shows only the attributes relevant to this example.

Table 2-3 Example of Specifying Price Lists on a New Order

Line Number	Product	Action	Price List
1	Internet Access	Add	NA
2	Home Phone Service	Add	Premium Consumer Price List
2.1	Home Phone Access	Add	NA
2.2	Voicemail	Add	NA

When you submit the order in Table 2-3, Siebel CRM populates the price list for Home Phone Access and 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.

For more information about the product models used, see "About the Product Models" in the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

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. 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.

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.

See the following topics for more information:

- 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 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.

About Interfacing Orders to BRM

This section describes how order information is interfaced to BRM.

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.

About Price and Discount Overrides

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.

For more information on applying one-time and penalty charges, see Applying One-time and Penalty Charges.

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 About Order Fallout Management for more information about order fallout.

Supporting Friends and Family Lists

Friends and Family lists, implemented as special rating products and included in service bundles in Launch, allow subscribers to call designated phone numbers at discounted rates.

When Customer Service Representatives (CSRs) create orders for service bundles 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 with the special rating product on the order.

When the order is successfully interfaced to BRM and is automatically 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 Home Phone and Wireless Voice, those services are fulfilled in the same BRM instance.

See "About Friends and Family Lists" and the discussion of profiles in *Siebel Communications Guide* for more information about creating special rating products and profiles at design time.

Supporting Family Share Plan with Differential Pricing

This feature allows a Customer Service Representative (CSR) to create a family plan wherein the members of the family added in the plan can avail the services of a Communications Service Provider (CSP) under a single plan with additional benefits, such as differential pricing for the lines.

Differential pricing refers to the strategy of offering the secondary and add-on lines of a Family share plan at reduced prices.

The CSR first creates a primary or main line for the family plan owner, and then creates secondary lines or add-on lines for other family members as part of the family share service bundle. These lines are priced differently based on the defined criteria or discount. For example, the main line can be priced at $20 and other lines can be priced at $15 after giving a discount.

For example, consider creating a Supremo Silver Family Plan. After the order is successfully completed:

- Validate that the charges for primary, secondary, and add-on members reflect the defined differential pricing.
- Validate the assets on Siebel and BRM.
- Validate unbilled usage data on Siebel and BRM.
- Validate individual bills for primary, secondary, and add-on accounts on Siebel and BRM.

Supporting Orders with Zone-Based Pricing

The zone-based pricing feature enables you to charge subscribers for calls made by them in accordance with the rates defined for predefined geographic zones. The system uses attribute-based pricing, where the zone acts as the determining attribute. You can configure zone models for attribute-based pricing using Standard Zone or Value Maps.

When a subscriber purchases a package that includes a product with a usage fee, the system applies zone-based rules. For example, if the attribute is the country being called from the subscriber’s home country, the system charges different call rates according to the zone-specific pricing defined.

The system validates these rules at run time and calculates charges based on the subscriber's usage. After assets are created, the system checks the billing amount against the usage pricing defined for the relevant zones.

This feature ensures accurate, automated billing by applying real-time zone-based pricing to subscriber usage.

See Configuring Zone Models for Attribute-Based Pricing for more information about configuring zone models.

See Real-Time Rating Using Event Attributes for more information about real-time rating using event attributes.

Zone-Based Pricing Based on Standard Zone Pricing Attributes
When creating a first-time purchase order with standard zone-based pricing, you must:

- Create a sales order by selecting an appropriate plan (for example, Supremo Starter Home Phone or Supremo Premium Home Phone).
- Select the payment method as BillMe.
- After the sales order is created:
  - Validate the assets on Siebel and BRM.
  - Validate unbilled usage data on Siebel and BRM.
  - Create a Call Detail Record (CDR) file for voice usage. The calls must be charged as per zone-based pricing.
  - Verify all the charges applied are correct on Siebel and BRM.
  - Generate bill on demand from BRM and validate the billed amount on Siebel and BRM.
- Run usage with the number from defined zones and validate pricing based on standard zone pricing attribute as follows:

Table 2-4 Standard Zone Pricing Attributes

zone_Origin	zone_Dest	IC	Charge
001	001	IC_LOCAL	$2
001	EU; Greece; 0030	IC_Europe	$4
001	EU; Germany; 0049	IC_Europe	$4
001	EU; Italy; 0039	IC_Europe	$4
001	EU; Spain; 0034	IC_Europe	$4
001	Mexico; 0052	IC_Mexico	$3
001	91 (India)	IC_India	$5

- After usage run is successful, generate bill on demand from BRM and validate the billed amount on BRM and Siebel.

Zone-Based Pricing Based on Value Map Attributes
When creating a first-time purchase order with zone-based pricing based on value maps attribute, you must:

- Create a sales order by selecting an appropriate plan (for example, Supremo 5G Unlimited or Supremo 5G Premium).
- Customize the order by selecting **Voice Roaming** option from the **Voice Service** drop-down list.

- Select the payment method as BillMe.
- After the sales order is created:
  - Validate the assets on Siebel and BRM.
  - Validate unbilled usage data on Siebel and BRM.
  - Create a CDR file for Voice consumption. The calls must be charged as per zone-based pricing based on value map attributes.
  - Verify all the charges applied are correct on Siebel and BRM.
  - Generate bill on demand from BRM and validate the billed amount on Siebel and BRM.
- Run usage with the number from defined zones and validate pricing based on value map attribute as follows:

Table 2-5 Value Map Attributes

zone_Dest	IC	Charge
Mexico; 0052	IC_ROAM_MEXICO	$3
EU; Germany; 0049	IC_ROAM_GERMANY	$4
EU; Greece; 0030	IC_ROAM_GREECE	$4
EU; Italy; 0039	IC_ROAM_Italy	$4.5
EU; Spain; 0034	IC_ROAM_Spain	$4.5
ROW#	IC_ROAM_ROW	$5

> **Note**
>
> For Rest of the World (ROW), the country code should start with **0**.

- After usage run is successful, generate bill on demand from BRM and validate the billed amount on BRM and Siebel.

About Supporting Balance Groups

The Order to Cash business process 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 Order to Cash business process 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 Order to Cash business process creates a subscriber 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 Order to Cash business process 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_AIA_HOME*/**source/soainfra/apps/AIAMetaData/config/AIAConfigurationProperties.xml** file in AIA.
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 "Update Files to MDS" in the *Oracle Application Integration Architecture Cloud Native Deployment 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

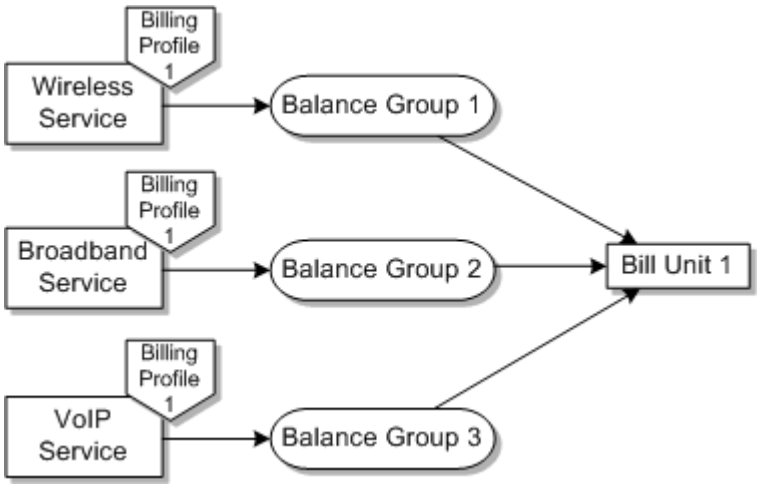
About Tracking and Billing Services with Service-Level Balance Groups Enabled

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.

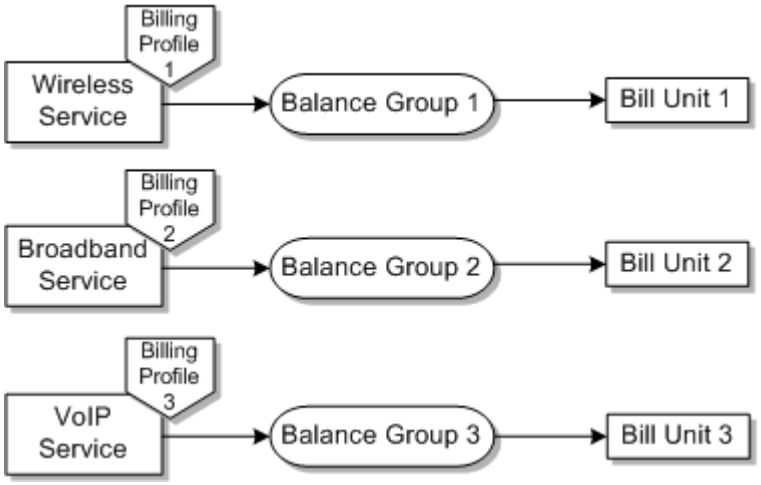
  When you submit the order, BRM tracks each service in a separate balance group and the balance groups all point to the same bill unit. Figure 2-7 illustrates this option.

Figure 2-7 Service-Level Balance Groups with a Shared Bill Unit



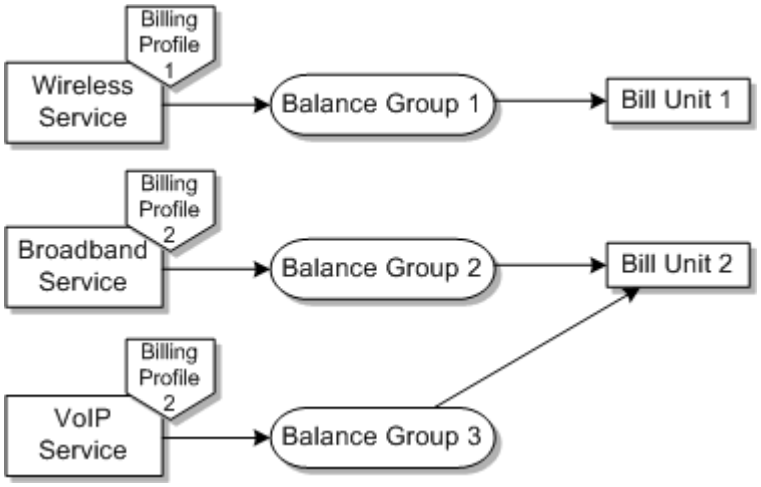
- **Billing all services separately**: You assign each service a separate billing profile in Siebel CRM. When you submit the order, BRM tracks each service in a separate balance group and each balance group points to a separate bill unit. Figure 2-8 illustrates this option.

Figure 2-8 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 2-9 illustrates this option.

Figure 2-9 Service-Level Balance Groups with Shared and Separate Bill Units



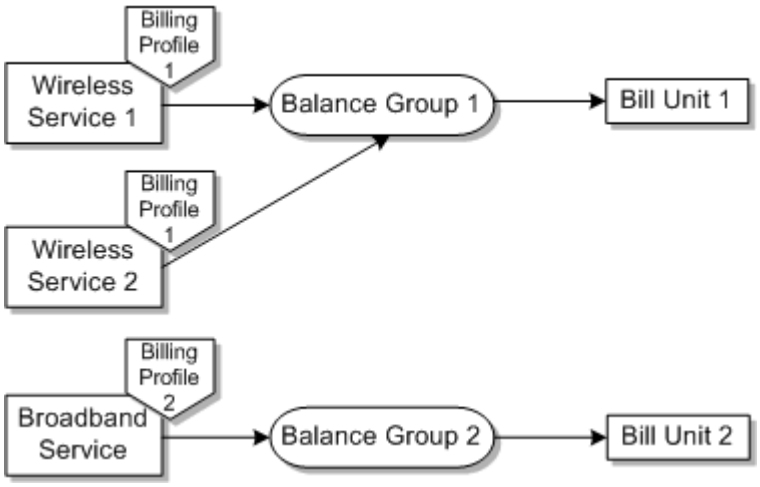
About Tracking Services in Nested Service Bundles in Service-Level Balance Groups

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 2-10 illustrates how BRM tracks nested service bundles when service-level balance groups are enabled.

Figure 2-10 Balance Groups for Nested Service Bundles



In Figure 2-10, 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 *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

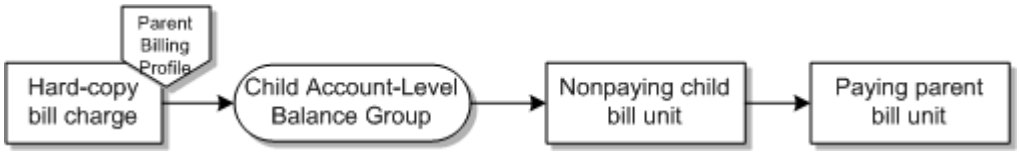
About Tracking Account-Level Products in the Default Account-Level Balance Group

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 nonpaying 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 2-11 illustrates how BRM tracks account-level products for nonpaying child accounts.

Figure 2-11 Account-Level Products in Nonpaying Child Accounts



Working with Service-Level Balance Groups Disabled

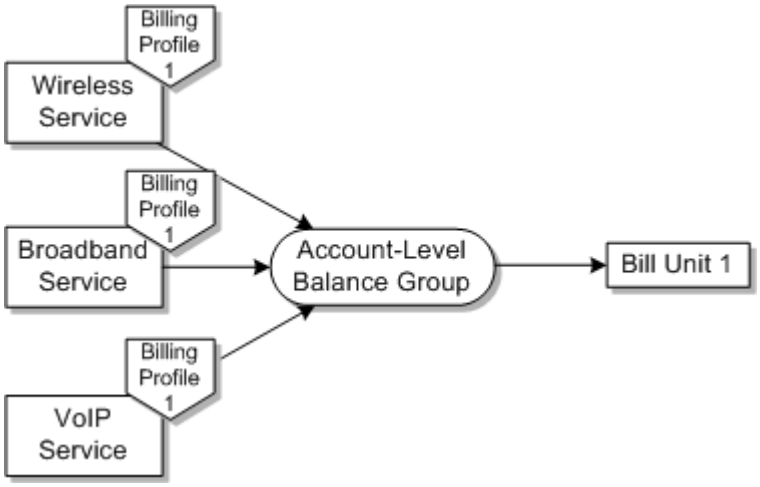
When you work with service-level balance groups disabled, BRM uses the default account-level balance group to track and pay for all services for the account.

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 2-12 illustrates how services are tracked under the account-level balance group.

Figure 2-12 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 2-12 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.

About Supporting Product Bundling

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

The Order to Cash business process 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 *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for more information about product bundling in Siebel CRM.

About Single-Phase and Two-Phase Billing

The Order to Cash business process 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 subscriber.

  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 the delay between when a service is made available for subscriber 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, the promotion 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 are 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 subscriber 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 subscribers 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 Move-Add line on a Siebel CRM order has a matching Move-Delete (and vice versa). The order management system sends Move-Add lines along with the Move-Delete lines to billing.
6. After order lines are submitted for 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.

About 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.

For information about creating time-based offerings and managing expired time-based offerings, see *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

> **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.

For more information on time-based offerings for change orders, see Supporting Time-Based Offerings on Change Orders.

About Order Provisioning

Siebel CRM sends subscriber 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.

For more information, see Implementing the Provision Order and Update Fulfillment Order Flows.

About Updating the Sales Order

This feature enables you to do the following:

- Update Sales Order Data
- Update 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 subscriber, 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 Order to Cash business process supports the order status attributes listed in Table 2-6.

Table 2-6 Order Status Attributes

Level	Attribute Name	Usage
Order Header	Fulfillment Status	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. Possible values include but are not limited to In Progress, Pending, Complete, Canceled, and Failed. 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. The OSM cartridge implementer can configure the values for Fulfillment Status attributes.
Order Header	Status Context	Provides details about the current order status. OSM cartridge implementers can configure this value.

Table 2-6 (Cont.) Order Status Attributes

Level	Attribute Name	Usage
Order Line	Fulfillment Status	Updates Siebel CRM with the current status of order line fulfillment at a high level. Possible values include but are not limited to In Progress, Pending, Created, Complete, Canceled, and Failed. The OSM cartridge implementer can configure the values for Fulfillment Status attributes.
Order Line	Milestone	Indicates the most recent fulfillment milestone reached. Example milestones in OSM: SYNC CUSTOMER START: reached PROVISIONING START: reached FULFILL BILLING COMPLETE: reached The OSM cartridge implementer can configure the values for Milestone attributes.
Order Line	Status Context	Provides details about the current order line status. OSM cartridge implementers can configure this value to indicate: • Required customer interaction. • If delivery is expected to be delayed. • The milestone or fulfillment function in which a failure occurred. • The cause of a cancellation or who canceled an order.
Order Line	Point-of-no-return	Indicates if Siebel CRM should allow revisions to an order line or submission of previously created revisions to an order line. 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.
Order Line	Actual Delivery Date-Time	Indicates the date when the purchased product or service is considered available to the customer. This date may be the date physical goods are shipped, delivered, or their receipt acknowledged. For service-based products, this date is when the service is activated. This date is computed in the 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. OSM uses this date to communicate changes for specific order line dates to Siebel CRM.

Because of the increased processing complexity of using different fulfillment status values for different services, Oracle recommends that you use a set of streamlined status values across product specifications. Using the streamlined values makes the status updates easier for your customers and customer service representatives (CSRs) to understand and lets you reuse the flow.

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.

About Support for Discount Matrix

Discount matrix allows you to define pricing and discounts specific to subscriber and market segments. The discounts may vary by region or channels, other attributes outside of the product dimension to pricing and discounts. The discount matrix is defined by the product manager. If any discounts are defined according to discount matrix, then those discounts will be applied when orders are placed.

For more information about discount matrix, see *Oracle Communications Digital Business Experience Concept to Market Guide*.

About Promotion Component Discount

Promotion Component Discounts are the discounts that apply on the price of components (part of the package hierarchy) in the context of the package.

For example, if a voice service is given a discount of 10% when Package A is ordered by the subscriber but no discount when Package B is ordered. Here, the voice service is a component in packages, A and B. Thus, promotion component discounts are in the context of the package.

See *Oracle Communications Digital Business Experience Concept to Market Guide* for more information about promotion component discounts.

About Offer Aggregation

Offer aggregation refers to grouping of similar products that allow subscribers to select a given product offer from this group. Typically, offer aggregation is based on Product Line or Product Specification.

For example, grouping of similar devices based on Product Line, such as a mobile phone.

When ordering a package of choice, an asset that is part of the offer aggregation could be added as default, but the subscriber can choose to add any other asset as long as it is part of the same offer.

Offer aggregation could be modeled within a package, commercial bundle, or service bundle.

For more information, see the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

About Compatibility Rules

Compatibility rules are defined to ensure that the appropriate combinations of products and services are offered to subscribers. The common use cases of such rules are a specific plan or package that may require the selection of a particular device, or choosing a device while limiting the accessories supported.

For example, if a customer wants to order a specific mobile accessory, a compatibility rule can be defined, stipulating that a specific mobile phone must be ordered as well.

Compatibility rules for product offerings are invoked during order creation in Siebel CRM.

For more information, see the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

About Eligibility Rules

Eligibility rules define if a particular product offer can be purchased by a subscriber, based on various parameters such as Account Type, Country, City, Postal Code, etc.

For example, there might be eligibility rules that are defined to allow a particular product may only be purchased in selected cities of a particular country. So, while ordering, the application validates if the account data conforms to these rules and accordingly allows the order to be created. If the account data does not conform to the defined eligibility rules, then the product offer cannot be purchased by the subscriber.

Eligibility rules for product offerings are invoked during order creation in Siebel CRM. For more information, see the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

About Supporting Bulk Orders

Bulk orders can be created by CSR for multiple orders for different subscriber accounts by raising a bulk request. The bulk request uses a bulk request template and the CSR enters all the details into it.

Bulk orders can also be used to create new orders or asset-based (MACD) orders. For more information about asset-based orders, see About MACD Orders (Asset-Based Orders).

The bulk request template can also be used to create or modify multiple orders against multiple accounts.

For more information on using this feature, see the *Siebel Order Management Guide*.

About Order Fallout Management

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.

Order Fallout Management allows you to do the following:

- **Create Trouble Ticket from Oracle AIA**

  See About the Create Trouble Ticket from Oracle AIA Flow for more information.

- **Create and Manage Trouble Ticket from OSM**

  See About the Create and Manage Trouble Ticket from OSM Flow.

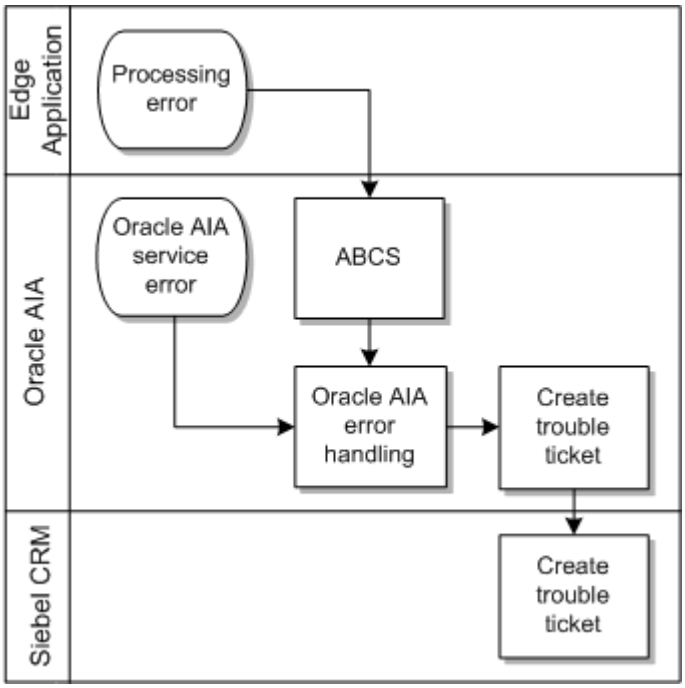
About the Create Trouble Ticket from Oracle AIA Flow

The Create Trouble Ticket from Oracle AIA flow provides trouble ticketing when OSM is not the central fulfillment system and is not used for order fulfillment and fallout management. In this flow, the Order to Cash business process provides services and artifacts to handle order fallout detection and notification.

The Order to Cash business process 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 Order to Cash business process creates a trouble ticket for every fault message notification, you must model your processes to generate only one notification for each order failure.

Figure 2-13 illustrates the Create Trouble Ticket from Oracle AIA flow at a high level.

Figure 2-13 Creating Trouble Tickets from Oracle AIA



When an error occurs in an edge application, such as in BRM while synchronizing subscribers 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.

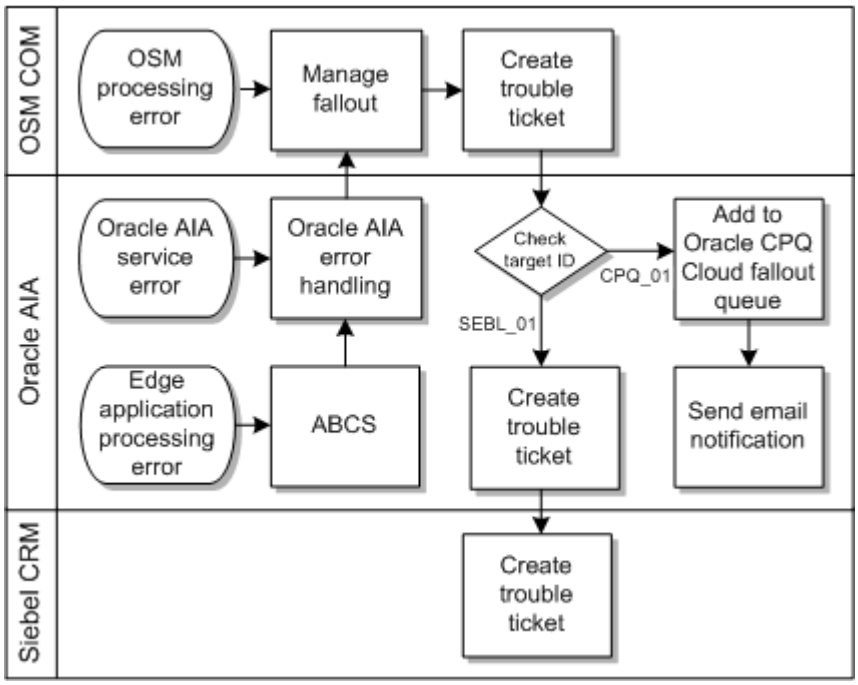
For more information about the implementation of this flow, see Implementing the Create Trouble Ticket from Oracle AIA Flow.

About the Create and Manage Trouble Ticket from OSM Flow

The Create and Manage Trouble Ticket from OSM flow provides trouble ticketing when OSM is the central fulfillment system. When you install the Order to Cash for OSM business process option, the Order to Cash business process automatically uses OSM to manage order fallout.

Figure 2-14 illustrates the high-level process flow involved in using OSM for order fallout management.

Figure 2-14 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 Order to Cash business process 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 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.

For more information about the implementation of this flow, see Implementing the Create and Manage Trouble Ticket from OSM Flow.

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 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

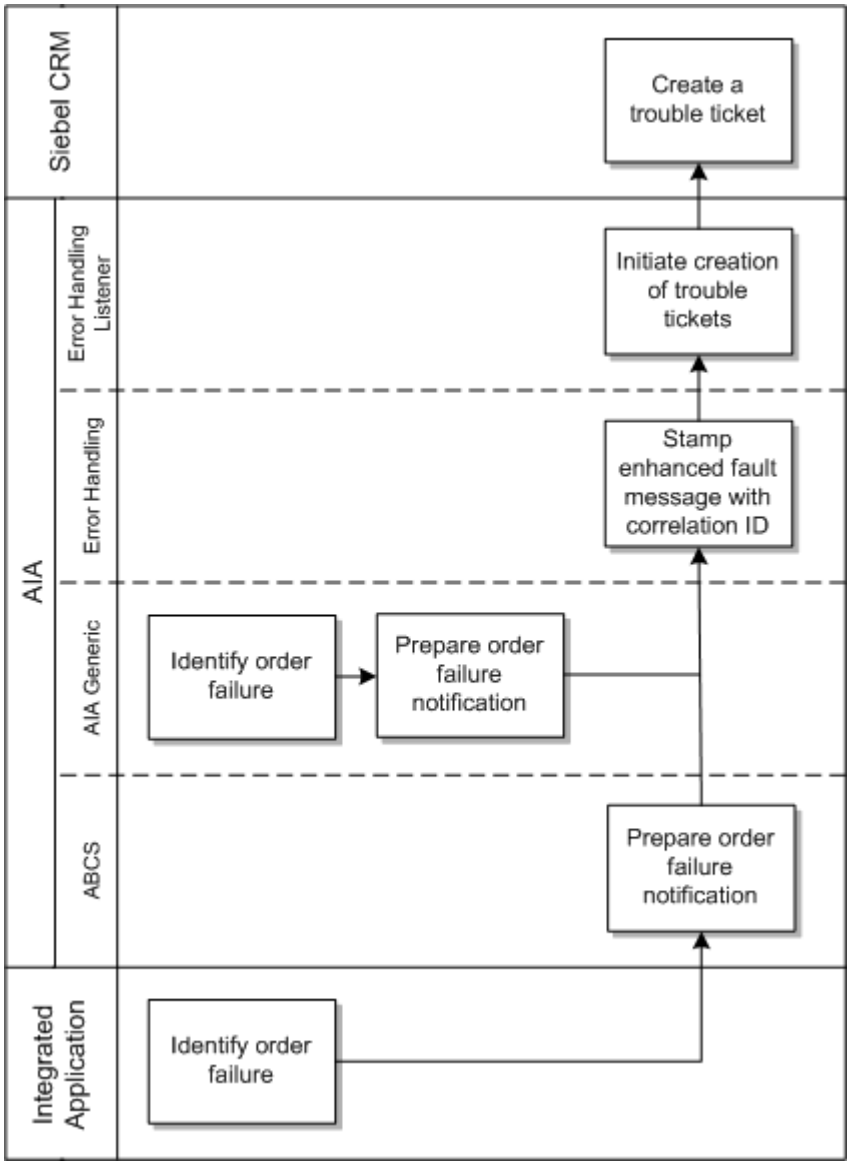
See Extending Fault Messages to Capture Order Fallout Information for more information about extending fault messages. See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about the Oracle 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 2-15 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 2-15 Detection Flow from Oracle AIA for Order Fallout

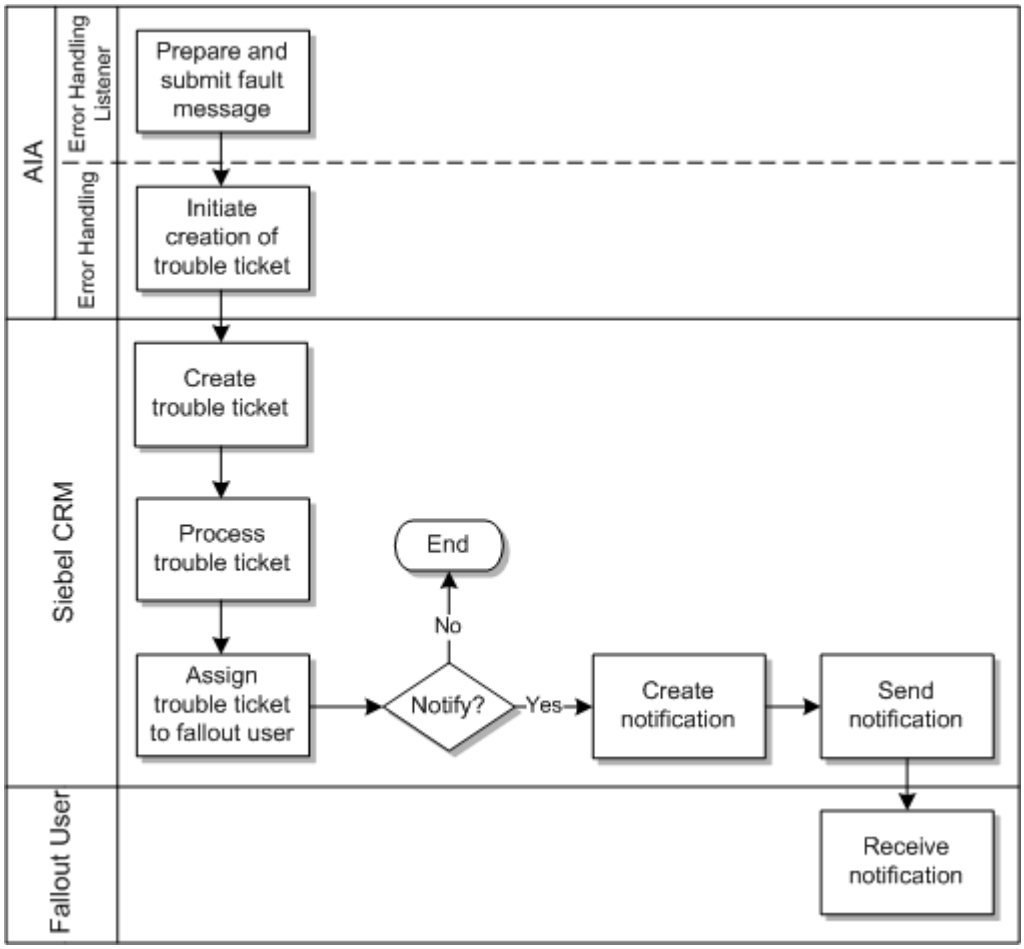


About Order Fallout Notification

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 2-16 illustrates the process for order fallout notification.

Figure 2-16 Order Fallout Notification Flow



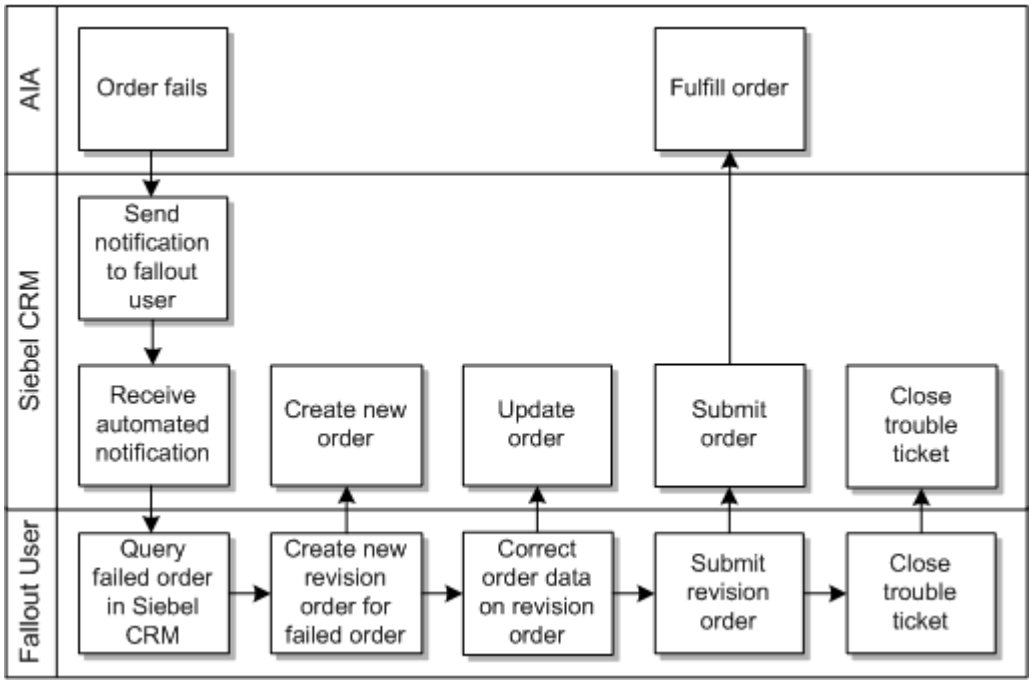
About Order Correction

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 2-17 illustrates the steps a fallout user takes to correct errors in a base order.

Figure 2-17 Siebel CRM Correction Flow for Order Fallout



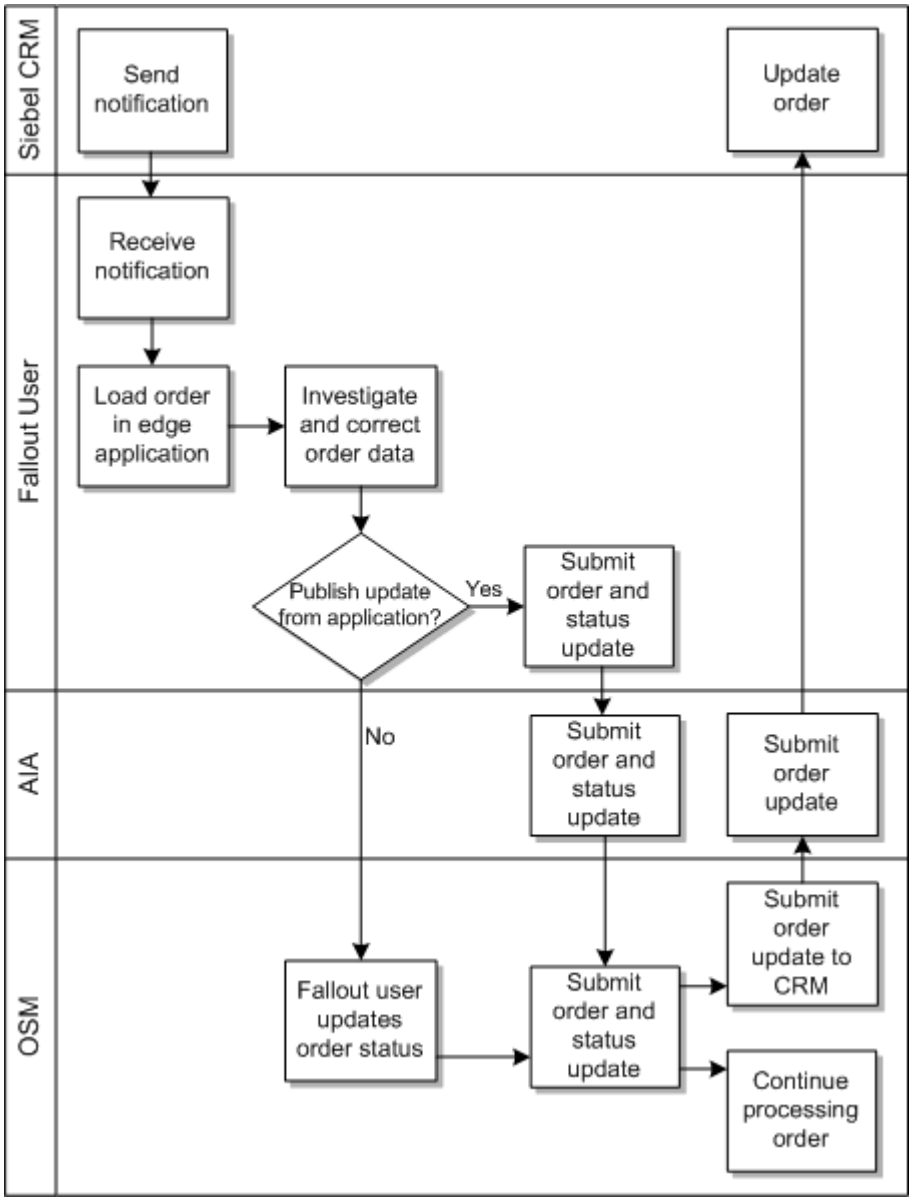
To correct errors in the data from participating edge applications (such as BRM, Siebel CRM, or an inventory or activation system), update the data in the edge application and resume the order OSM in the central order management role (OSM COM).

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 2-18 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 2-18 Local Correction Flow



Extending Fault Messages to Capture Order Fallout Information

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 2-7 and Table 2-8 list the additional fields that you must add to the fault messages to capture order failure information.

Table 2-7 lists the order header-level fields to include in the fault messages.

Table 2-7 Order Header-Level Data

Field Name	Type	Description	Source	Optional
Order Originating System Code	ID	The system code of the Siebel CRM system from which the order was placed. It is required to cross-reference the IDs back to the appropriate Siebel CRM IDs.	Oracle AIA service	No
Sales Order Number	Alphanumeric	Alphanumeric identifier for the sales order number (Siebel CRM value).	Siebel CRM	Yes
Sales Order Revision Number	Numeric	Numeric field storing the sales order number (Siebel CRM value).	Siebel CRM	Yes
SalesOrderID	ID	Siebel CRM Sales Order ID. Required to create trouble tickets for the orders that fail even before hitting the central fulfillment system.	Siebel CRM	Yes
Account Name	Alphanumeric	Alphanumeric value identifying the Siebel CRM account name.	Siebel CRM	Yes
Account ID	ID	Siebel CRM Account ID. Required to create trouble tickets for the orders that fail even before hitting the central fulfillment system.	Siebel CRM	Yes
SalesOrderID (Common)	ID	Common Order ID. (Required when Oracle AIA creates the trouble tickets).	Oracle AIA service	No
AccountID (Common)	ID	Common Account ID.	Oracle AIA service	Yes
Order ID	ID	Alphanumeric identifier for the order. Assigned by fulfillment system to the order. The fulfillment system uses it to correlate the order back to the common order ID received for the original order. The common order ID is then mapped to the Siebel order ID by the Siebel ABCS.	Fulfillment System	No
Order Number	Alphanumeric	User-friendly identifier for the order in the fulfillment system.	Fulfillment System	Yes
ProductID	Alphanumeric	Alphanumeric identifier for the product used for the failed line or the product for the first order line in case of multiple line failures.	Siebel CRM or Oracle AIA service	Yes
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

Table 2-7 (Cont.) Order Header-Level Data

Field Name	Type	Description	Source	Optional
Message	Alphanumeric	Used for the message (error, warning, or other). It can also be used to return notification to subscribers or other systems. Not to be confused with the original input order message.	Fulfillment System of failure or Oracle AIA Service	Yes
Error Code	Alphanumeric	Used to return the error code from the downstream fulfillment system (if any).	Fulfillment System of failure or Oracle AIA service	No
Error Severity	LOV	Used to return the error severity from the downstream fulfillment system (if any).	Fulfillment System of failure or Oracle AIA service	Yes
Processing Number	ID	Identifier of the job ID assigned in case of batch or bulk orders.	Siebel CRM	Yes
Processing Type Code	Code	Code to identify the job type.	Siebel CRM	Yes
Processing Quantity	Quantity	Job cardinality - Total number of orders within the job.	Siebel CRM	Yes

See Guidelines for Ensuring that Oracle AIA Processes are Fallout-Compliant for more information about how to pass this information from the edge application or Oracle AIA service to the process integration for Order Fallout Management.

Table 2-8 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 2-8 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 subscribers 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

Table 2-8 (Cont.) Order-Line Item-Level Data

Field Name	Type	Description	Source	Optional
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 2-7 and Table 2-8.
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.

Using Promotion Component Discounts

Promotion component discounts provide a way to incorporate targeted discounts within certain packages as part of your ordering process. When an order includes a package with eligible components, any applicable component discounts are automatically applied during order creation and pricing.

This feature ensures that subscribers receive the appropriate discounts for specific services or product features included in their selected packages. As you configure and process subscriber orders, the system evaluates the packages in the order, applies any defined component-level discounts, and accurately reflects the discounted prices throughout the order-to-cash flow.

For details on setting up promotion component discounts or configuring promotion structures, see *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

Example

Consider a sales order is created for a Supremo TV Basic package with the following details:

- Promotion Ordered: Supremo TV Local Channels
- Product Specification: Digital TV Local PS
- Billing Type: Subscription
- Billing Service Type: /service/digitaltv
- Base Recurring Charge: $3.00 per month
- Promotion Component Discount: 25% off on Monthly Cycle Fee (MCF), recurring

The discount is applied on the above order as follows:

- The 25% recurring discount is applied to the Supremo TV Local Channels monthly charge.
  - Calculation: 25% of $3.00 = $0.75
  - Discounted Monthly Charge: $3.00 - $0.75 = $2.25 per month

After the promotion component discount is applied:

- Validate the assets on Siebel and Oracle Communications Billing Revenue Management (BRM).
- Verify all the charges are applied appropriately on BRM and Siebel.
- Verify promotion component discount of 25% off on MCF (Recurring) is applied on Supremo TV Local Channels.
- Validate unbilled usage data on Siebel and BRM.
- Create bill on demand from BRM and validate the billed amount on BRM and Siebel.

Using Product Offer Cardinality

Product offer cardinality specifies the required quantity and selection constraints for products and services within a package. The Product Manager configures cardinality rules at any level of the package hierarchy, which can include the Commercial Bundle (CB), Service Bundle (SB), or simple offer level. These settings determine whether a specific component is mandatory or optional within the package.

During the ordering process, the cardinality definitions set by the Product Manager must be strictly followed. For example, if a product or service is marked as mandatory at any hierarchy level, it cannot be excluded from the order. Conversely, if the item is marked as optional, Customer Service Representatives (CSRs) can choose to include or exclude it according to customer needs. This ensures consistency and compliance with the intended product structure.

Example

A CSR creates a new sales order for the Supremo TV - Premium package. This package includes the Supremo Digital Music Pack component as follows:

- Default Cardinality: [0,1,0]
  - Minimum: 0 (optional)

  - Maximum: 1 (only one can be selected)
  - Mandatory Indicator: 0 (optional by default)

Because the mandatory indicator is 0, the Supremo Digital Music Pack is optional to be included in the order. The CSR can select only one instance of the Supremo Digital Music Pack for this order (as the maximum is 1).

This behavior applies consistently, regardless of which level (CB, SB, or Simple Offer) the cardinality is defined. It ensures that all packages created and ordered in the system strictly observe the cardinality guidelines as specified by the Product Manager.

After the cardinality is defined:

- Validate the assets on Siebel and Oracle Communications Billing Revenue Management (BRM).
- Validate unbilled usage data on Siebel and BRM.
- Create bill on demand from BRM and validate the billed amount on BRM and Siebel.

Using Commitment Terms Override

Commitment terms specify the duration for which a subscriber is obligated to retain a product or service, including any associated penalty clauses for early termination. Commitment terms can be configured at multiple levels within the product hierarchy, including individual services, devices, accessories (simple offers), service bundles (SB), commercial bundles (CB), and complete package. This flexibility allows for detailed and tailored agreements that reflect commercial realities and customer preferences.

During the new order process, you can assign different commitment terms to different components within a package. For example, a main package might require a two-year commitment, while a specific commercial bundle within that package might require only a one-year commitment. Additionally, penalty amounts for early termination can be set based on these varying terms.

The system ensures that, when an order is placed and completed, the correct commitment terms and any defined penalties are captured and reflected in the customer’s record, as per the product model configuration.

Example

Consider a Customer Service Representative (CSR) creates a sales order for a Supremo Silver Family Plan package with the following commitment terms:

- Commitment Period: 1 Year
- Grace Period: 15 Days
- Penalty Type: Penalty
- Penalty Amount: $50
- Pro Rate Plan: Fixed Proration

This means that:

- A subscriber orders the Supremo Silver Family Plan package, agreeing to a 1-year commitment.
- The plan includes a 15-day grace period at the start of the contract. If the subscriber cancels the plan within the first 15 days, no penalty applies.

- If the subscriber disconnects the plan any time after the grace period but before the 1-year commitment ends, the penalty amount is calculated using a fixed proration logic as per the configuration.

After the commitment terms are applied:

- Validate if the assets that got created have the corresponding commitment term and penalty information appropriately for Supremo Silver Family Plan on Siebel, once the order is complete.
- Validate unbilled usage data on Siebel and Oracle Communications Billing Revenue Management (BRM).
- Create bill on demand from BRM and validate the billed amount on BRM and Siebel.

3

Request to Change Business Process

This chapter describes the features of the Request to Change business process and its features.

Overview of the Request to Change Business Process

The Request to Change business process comprises activities related to the subscriber's request for changes to in-progress orders, assets or services, as well as account updates.

The Request to Change business process includes the following features:

- Revision Order
- Follow-on Order
- Cancellation Order
- Move, Add, Change, Disconnect (MACD) Orders for Asset-Based Ordering
- Updates to Subscriber Account Information

The sections that follow describe these features.

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 order completion or another revision is submitted.

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

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 is resubmitted, then the order management system compares the revised line against what was submitted to billing initiation. It 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 About MACD Orders (Asset-Based Orders) 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).

Table 3-1 summarizes revision actions.

Table 3-1 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.

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 3-2 shows an example of a revision of the order shown in Table 2-3.

Table 3-2 Example of Changing Price Lists on a Supplemental Order

Line Number	Product	Action	Price List
1	Internet Access	Add	Premium Consumer Price List
2	Home Phone Service	Add	-

Table 3-2 (Cont.) Example of Changing Price Lists on a Supplemental Order

Line Number	Product	Action	Price List
2.1	Home Phone Access	Add	-
2.2	Voicemail	Add	-

When you submit the order in Table 3-2, OSM changes the price list for the Internet Access product to **Premium Consumer Price List** and changes the price list for the Home Phone Service and its components to the price list specified for the order header. OSM sends the revised order through the integration to BRM for billing.

For more information about the product models used, see "About the Product Models" in the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

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 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. See About MACD Orders (Asset-Based Orders) for more information about modifying orders.

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 3-3 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 3-2.

Table 3-3 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	Home Phone Service	Update	Premium Consumer Price List
2.1	Home Phone Access	Update	-
2.2	Voicemail	Update	-

When you create the order in Table 3-3, 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 Home Phone Access and 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 Home Phone Service 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.

For more information about the product models used, see "About the Product Models" in the *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

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 to 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 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.

About Cancel Order

Order cancellation for existing sales orders is possible as long as the sales order has not reached the point of no-return. For more information about the point of no return, see *OSM Concepts*.

When a subscriber requests for cancelling an existing order, the cancellation request is sent to fulfillment as a cancellation order by removing all order items in the existing order. Fulfillment receives the request and generates the orchestration plan based on the cancellation order. The fulfillment status provides details about the order cancellation progress. Order cancellation is successful when no new services are added.

About MACD Orders (Asset-Based Orders)

This section provides information about the Move, Add, Change, Disconnect (MACD) orders and the subsequent line actions that are supported for existing 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 orders apply to asset-based orders where subscribers request changes to existing subscribed services/assets created from the original (base) order.

Orders for disconnecting services are described under the Termination to Confirmation Business Process. For more information, see Termination to Confirmation Business Process.

About Adding and Removing Services

When a subscriber wishes to purchase additional services as an add-on to their existing order (**Add** action), the request is subjected to eligibility rules that determine if such a request is valid.

These eligibility rules are pre-defined and are validated during the creation of both new orders and change orders where change orders include Supplemental or Revision orders, Follow-on orders, and Modify orders. If the order passes the validation, then the chosen services are added to, or removed from, the order. For more information about eligibility rules, see About Eligibility Rules.

Once the modified order is processed and the services are activated or disconnected, invoices are generated based on the modified services or assets and according to the billing frequency opted by the subscriber.

For more information about the applicability of the **Add** and **Delete** action, see Supporting MACD Actions and Attribute Changes.

About Moving Existing Services

When a subscriber calls to change the service address of the services or assets in their existing order to a new address, the request can be processed after the following requirements are validated:

- Eligibility for changing the service address.
- Serviceability at the new address.

For the change in service address, a **Move** order is created in reference to the existing order and the service/asset to be moved. Upon completion of the **Move** order, invoices are

generated based on the moved services or assets and according to the billing frequency opted by the subscriber.

For more information about the applicability of the **Move** action, see Supporting MACD Actions and Attribute Changes.

About Upgrading and Downgrading Services

Subscribers can choose to change their package to either upgrade or downgrade services or assets. Services or assets can be upgraded or downgraded based on migration rules for packages (such as compatibility or eligibility rules). These rules are used to validate upgrade or downgrade requests made by the subscriber; if it passes validation, a MACD order is created, and the subscriber's new package is activated and the older package is disconnected.

Migration may also entail penalties, which are computed based on the subscriber request and associated rules. See *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for more information on these rules.

Furthermore, upgrade or downgrade fees are applicable when there is a commitment period defined for the packages. When both upgrade or downgrade fees as well as disconnect fees are defined for a package, only upgrade fees may be applicable.

For more information on defining migration rules during product definition, see *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*. For more information about support for upgrade or downgrade, see Supporting MACD Actions and Attribute Changes.

About Suspending Services

This functionality allows you to suspend a subscriber's service (or asset) temporarily so they are not billed for the duration it is not in use.

A **Suspend** order is created with reference to the existing order of a subscriber and the service or asset to be suspended. It may include a service suspension date which indicates the date of service suspension. Optionally, a date for resuming services may also be indicated.

When you submit the suspend order, a sales order is created to pass on the fulfillment. The provisioning system consumes the order to suspend the service and informs BRM to halt the billing for the service suspension duration.

For more information about the applicability of the **Suspend** action, see Supporting MACD Actions and Attribute Changes.

About Resuming Services

This functionality allows you to resume suspended services.

A resume order reinstates the suspended service based on the recorded date of service resumption. The fulfillment system orchestrates the order to the provisioning system which resumes the service.

Upon service resumption, the subscriber is invoiced for the service based on the billing frequency and date.

For more information about the applicability of the **Resume** action, see Supporting MACD Actions and Attribute Changes.

About Updating Subscriber Account Profile

Account profile updates are key use cases that must be handled as part of the Account Management business flow. As part of Subscriber Onboarding, a new account is created when a first time order is placed for the customer.

Over the subscriber lifecycle, there could be scenarios when an update is needed to the account profile. The most common or frequent updates to be made could be changes to contact name, email IDs, address, etc. The Account Update business flow manages these changes directly between Siebel CRM and BRM.

About Interfacing Orders to BRM

Interfacing order information to BRM for changes orders follows the same flow as the Order to Payment sales orders. For more information, see About Interfacing Orders to BRM.

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 *DBE Concept to Market Implementation Guide* 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 subscriber 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 baseline 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.

Modifying Orders with Special Rating Products

You can do the following changes on orders with special rating products:

Modifying special rating list items:

- You can associate a different list with the special rating product by submitting a modify order that updates the special rating list reference on the existing special rating product

asset. When the integration processes the change, it updates the list profile in BRM with contents from the new list.

  - You can add or remove phone numbers from a list currently associated with a special rating product. Use the Siebel Special Rating Profile sub view to make changes to the list and synchronize them to BRM.

Modifying Orders with Differential Pricing (Family Plan)

To modify an order with Differential Pricing (Family Plan), you must:

- Create one parent account (primary line bundle) and one child service account (secondary line bundle) as part of the hierarchy.
- Create Billing Profile only for the parent account. Select the payment method as Credit Card.
- Create a sales order with a family plan (for example, Supremo Gold Family Plan) in the parent account. By default, one primary and one secondary bundle are added to the plan.
- Validate the assets on Siebel and BRM.
- Validate unbilled usage data on Siebel and BRM.
- Validate if the charges for the members (primary or secondary) are reflecting appropriately in line with the differential pricing.
  These lines are priced differently based on the defined criteria or discount.
- Generate a bill on demand from BRM and validate the billed amount on Siebel and BRM.
- Create a modify order by adding an add-on commercial bundle. Now, the promotion will have one primary, one secondary, and one add-on line bundle.
- Submit the order.
- Validate the differential pricing while querying the respective accounts in BRM.
- Validate the assets on Siebel and BRM.
- Validate unbilled usage data on Siebel and BRM.
- Generate a bill on demand from BRM and validate the billed amount on Siebel and BRM.

Modifying Orders with Zone-Based Pricing

When you modify an existing service order (such as adding or removing components), the system continues to apply the same zone-based pricing rules as it does during the first-time purchase.

The system ensures that calls made to the country specified in the attribute-based pricing (with zone as the attribute) continue to be charged according to the existing plan. This maintains pricing consistency after any order modifications.

You can create a modify order with zone-based pricing based on value maps attribute, as follows:

- Create a sales order by selecting an appropriate plan (for example, Supremo 5G Lite).
- Select the payment method as BillMe.
- After the sales order is created:
  - Validate the assets on Siebel and BRM.

  - Validate unbilled usage data on Siebel and BRM.
  - Verify all the charges applied are correct on Siebel and BRM.
  - Generate bill on demand from BRM and validate the billed amount on Siebel and BRM.
- Modify the order by selecting **Voice Roaming** option from the **Voice Service** drop-down list.
- Run usage with the number from defined zones and validate pricing based on value map attribute. See Table 2-5 for more information.
- After usage run is successful, generate bill on demand from BRM and validate the billed amount on BRM and Siebel.

Modifying Orders with Promotion Component Discounts

Promotion component discounts can be impacted when you revise orders prior to asset creation and Point of No Return (PONR) or modify existing assets or services after asset creation, that is Asset-Based Ordering (ABO). In both cases, you must validate that the applicable promotion component discount is accurately reflected in the order and related billing systems.

Revising Orders Defined with Promotion Component Discount

Order revisions are changes made to a sales order that has been submitted for processing but has not yet reached PONR and has not resulted in asset creation. Examples of order revisions include:

- Adding or customizing optional services, or changing the package before PONR:
  Consider a subscriber submits an order for the Supremo TV Premium package, with payment via BillMe. Before reaching PONR, the order is revised to change the package to Supremo TV Basic and add the optional service Supremo TV Local Channels, which qualifies for a 25% off recurring monthly charge fee (MCF) discount. At each stage, validate the billing in Siebel and Oracle Communications Billing Revenue Management (BRM) to ensure the correct charges and discount are applied.

Modifying Services Defined with Promotion Component Discount

Order modifications are performed after assets have been created, using ABO. In this scenario, the subscriber already owns products/services (assets) and requests changes. Examples of asset modifications include:

- Deleting an optional service from the package:
  Consider a customer who has an active Supremo TV Basic package (which includes the optional service Supremo TV Local Channels), which consists of a 50% off recurring MCF discount. Create a modify order by deleting the optional service Supremo TV Local Channels from the package.

  After the modify order is processed successfully:

  - Validate the assets on Siebel and BRM.
  - Create bill on demand from BRM and validate that the optional service and its associated discount are not reflecting in the billed amount.

Modifying Orders with Product Offer Cardinality

Product offer cardinality determines the allowed quantity and mandatory or optional status for each component in a package. When you revise an order (before asset creation and before PONR) or modify existing assets (after asset creation, via ABO), you must ensure that cardinality rules for each component, as defined in the product model, are enforced and reflected accurately in both the order and subsequent billing.

> **Note**
>
> If a component is marked as mandatory, you cannot remove that component as part of Asset-Based Ordering. However, you can remove the optional components from existing assets or add to the existing assets.

Revising Orders Defined with Product Offer Cardinality

Order revisions are changes to a sales order that is still in process (not yet fulfilled, assets not yet created), and allowed only before reaching the PONR. Cardinality rules are validated in real time to ensure compliance with product model definitions.

Example:

A subscriber submits an order for the Supremo Starter Home Phone package with components defined by the following cardinality:

- Call Conferencing: [0,1,0]
- 3-Way Calling: [0,1,0]
- Visual Voice Mail: [0,1,0]
- Web Conferencing: [0,1,0]

Before PONR, the subscriber decides to add the optional component Web Conferencing to their existing order for Supremo Starter Home Phone. The revision is allowed because:

- Web Conferencing is optional (minimum=0, maximum=1, default=0).
- A maximum of one can be added.

After the revised order is submitted:

- Validate the system enforces the [0,1,0] cardinality for the component.
- At every step, verify that the asset structure and charges, including any changes due to the added component are consistent in both Siebel and BRM.
- Use bill-on-demand to validate the billed amount after the revision.

Modifying Services Defined with Product Offer Cardinality

Order modifications occur when the subscriber already has active assets (services) and requests changes, such as adding or removing optional components from the promotion. All changes must comply with the product's cardinality rules as defined at the time of package configuration.

Example:

A subscriber submits an order for the Supremo Starter Home Phone package with components defined by the following cardinality:

- Call Conferencing: [0,1,0]
- 3-Way Calling: [0,1,0]
- Visual Voice Mail: [0,1,0]
- Web Conferencing: [0,1,0]

The subscriber decides to delete the optional component Web Conferencing from their existing Supremo Starter Home Phone package and creates a Modify order. The deletion is allowed because:

- Web Conferencing is optional (minimum=0, maximum=1, default=0).

After the optional component Web Conferencing is deleted:

- Validate the assets on Siebel and Oracle Communications Billing Revenue Management (BRM).
- At every step, verify that the asset structure and charges, including any changes due to the deleted component are consistent in both Siebel and BRM.
- Use bill-on-demand to validate the billed amount after the deletion.

Modifying Orders with Commitment Term Override

Commitment term overrides can be applied at various levels in the package hierarchy. You must validate the appropriate commitment term attributes, such as commitment period, start or end date, penalty information, and any overrides during order revision (before asset creation and PONR).

Revising an Order with Commitment Terms Override

Order revisions are changes made to a sales order that has not yet been fulfilled (assets not yet created) and is still before the Point of No Return (PONR). Any changes are subject to PONR rules and validations.

Example:

- Create a sales order for a parent account (with a primary bundle) and a child account (with a secondary bundle), with the billing profile assigned to the parent account.
- Select the Supremo Silver Family Plan package; by default, one primary and one secondary line bundle are added.
- Before the order reaches PONR, revise the order to change the package to Supremo Gold Family Plan. Again, one primary and one secondary line bundle are included by default.
- Select BillMe as the payment method.
- Validate, prior to order completion, that the following attributes related to commitment terms in Siebel are set correctly for each bundle:
  - Commitment Period
  - Start and End Dates
  - Grace Period
  - Penalty Type and Amount
  - Payment Method

After order completion:

- Validate the assets, any unbilled usage data, and ensure all charges and penalties (if any) are correct in Siebel and Oracle Communications Billing Revenue Management (BRM).
- Use bill-on-demand in BRM to validate that the billed amount matches expected results.

4

Termination to Confirmation Business Process

This chapter describes the Termination to Confirmation business process and its features.

Overview of the Termination to Confirmation Business Process

The Termination to Confirmation business process deals with all activities related to carrying out a subscriber's request for termination of assets or services, as well as deactivating their engagement with the CSP.

The Termination to Confirmation business process supports the following features:

- Disconnecting services
  - Terminating/Disconnecting a package or services in a package
  - Terminating/Disconnecting a service
  - Terminating/Disconnecting services for member(s) in account hierarchy
- Terminating Subscriber Account

About Disconnecting Services

This functionality allows a subscriber's assigned services to be disconnected (terminated) when not required.

The subscriber can call the agent to raise a request for terminating their services. The agent can view the services to be disconnected (this can be the entire package or specific services within the package), as well as any penalty (if applicable) on the termination order, and advise the subscriber on proceeding with the termination order.

A termination order is created with reference to the subscriber's original order and the service to be disconnected. The termination date associated with this order can either be the current date (for immediate disconnection) or a future date.

Once the order is submitted to OSM for fulfillment, OSM consumes it and orchestrates it to the provisioning system to terminate the service. OSM then informs BRM about the disconnected service to generate the invoice for any disconnection penalty and to also stop future invoices for the services being disconnected.

About Disconnecting a Package

When a subscriber wishes to disconnect an active package with a commitment period, the package components will also be disconnected.

The total penalty that is levied on the subscriber is determined by:

- **Package level penalty:** This may be a flat amount. However, if the package is associated with a prorated plan, the penalty amount will depend on how long after service activation the disconnect request was placed.

- **Component level penalty**: This is applied if any package components have a component charge penalty defined. The services associated with the package are now in an inactive state.

About Disconnecting a Service

In the cases where a subscriber has multiple assets/services as components in their existing package, they can choose to only disconnect the components no longer required. This only applies to the optional services included as add-ons, and not mandatory services of the package.

Disconnection of services is carried out according to the following two scenarios:

- **Disconnection of a Component with Commitment:** If a component within a package has commitment enabled, it cannot be disconnected in isolation. So disconnection of this component requires disconnection of the entire package.
  For example, consider a package named Supremo Voice 500 that has the components Voice and SMS Bundles (commitment enabled) and Data and OTT Bundles (with no commitment). If a request is made to disconnect either the Voice Bundle or SMS Bundle, the entire Supremo Voice 500 package must be disconnected. Additionally, component-level charges may apply as a disconnect penalty.
- **Disconnection of a Component with no Commitment:** If a component within a package does not have commitment enabled, it can be disconnected independently without affecting the rest of the package.
  For example, if a subscriber wants to disconnect the Data Bundle or OTT Bundle from the Supremo Voice 500 package, they can be removed in isolation without terminating the entire package. However, any component-level charges may still apply as a penalty.

About Disconnecting Services for a Member in Account Hierarchy

In the cases where account hierarchy is defined, it is possible for subscribers to disconnect services of one or more members in the hierarchy if no longer required.

In such cases, the termination order is created with reference to the services being terminated for a specific member in the hierarchy. The termination can be done immediately or on a future date.

The billing system generates the invoice for any termination penalty and also stops generation of future invoices for the specific member. The agent will then view the subscriber account to ensure that the services for the specific family member are reflected as **Inactive**.

About Termination of Subscriber Account

A subscriber can choose to terminate their accounts. This action is done in Siebel, when the subscriber places a request to terminate their order via the CSR. The CSR must have the privileges to do this activity.

In order to deactivate a subscriber's account, all associated services must be disconnected first. Once the services are disconnected, the CSR terminates the account on Siebel CRM and the account termination status is sent to BRM, so that invoices are not generated in the future.

Terminating Services with Special Rating Products

When you terminate a service bundle or package that includes a special rating product, the integration terminates the associated assets in BRM and updates their status to **Inactive**.

Terminating Family Share Plan with Differential Pricing

When you terminate a service bundle or package that includes a family share plan with differential pricing, the integration disconnects the lines of one or more members in BRM, and updates their status to **Inactive**.

> **Note**
>
> - The commitment terms for the primary and secondary or add-on lines may be different and in this case, validate if the penalty is duly computed depending upon the termination of a given line.
> - Termination of primary line can be done independently even though the commitment flag is enabled. Secondary and add-on lines continue to be Active.
> - You must validate the asset status in BRM and Siebel after the termination is completed.

The following are the possible termination scenarios:

- Disconnecting primary line of a family plan.
- Disconnecting the secondary or add-on line bundle of the family share plan.
- Disconnecting the entire family share plan or promotion.

Terminating Services with Zone-Based Pricing

When you terminate a service bundle or package that includes zone-based pricing, the integration terminates the associated assets in Siebel and BRM, and updates their status to **Inactive**.

The possible termination scenario for services with zone-based pricing is as follows:

- Create a sales order by selecting an appropriate plan (for example, Supremo Starter Home Phone).
- Select the payment method as BillMe.
- After the sales order is created:
  - Validate the assets on Siebel and BRM.
  - Validate unbilled usage data on Siebel and BRM.
  - Verify all the charges applied are correct on Siebel and BRM.
  - Generate bill on demand from BRM and validate the billed amount on Siebel and BRM.
- Run usage with the number from defined zones and validate pricing based on standard zone pricing attribute. See Table 2-4 for details about various standard zone pricing attributes.
- After usage run is successful, disconnect all services and assets of the account.

> **Note**
>
> A penalty will be applied based on the commitment term and the grace period. See Commitment Terms and Rules for more information about eligible commitment terms and rules.

- After the services get disconnected, the corresponding assets become inactive, and if there is any penalty associated, it will be computed and levied.
- Generate bill on demand from BRM and validate the billed amount on BRM and Siebel.

Terminating Services with Promotion Component Discounts

You can terminate services (assets) defined with promotion component discounts at any level within the package hierarchy. Upon termination, it is important to validate that all related status and billing information is correctly updated in both Siebel and Oracle Communications Billing Revenue Management (BRM), including the application of any penalties, discounts, or adjustments as per the defined rules and terms.

Example

Suppose you have submitted a sales order for the Supremo Broadband Premium package and included the optional Supremo Broadband Installation service, which qualifies for a 15% discount on the purchase fee. After the order is completed and the services are active, proceed to disconnect all the associated services and assets for the account.

After the termination order is processed, validate that:

- In Siebel
  - Confirm that the status of each terminated service and asset is updated to reflect disconnection/termination.
  - Check that the applied 15% discount is reflected on the final invoice or billing record for the installation service.
  - Review the Commitment Term and Penalty attributes associated with the terminated assets to confirm if any early termination penalty is applicable according to commitment rules.
  - Verify that service end dates and any relevant package or discount identifiers are recorded accurately in the asset details.
- In BRM
  - Validate asset/service termination has been propagated and is consistent with Siebel.
  - Check that final charges, pro-rated amounts, discounts (such as the 15% installation discount), and any penalties are calculated as per the product's rules and reflected in the customer's account balance and final bill.
  - Run bill on demand (if required) to generate a final bill and review all applied adjustments for accuracy.

Terminating Services with Commitment Term Override

This section describes how to handle the termination of assets or services that are associated with product offerings having different commitment terms within the package hierarchy.

The following are the possible termination scenarios:

- Terminate the Entire Package: The subscriber ends all services from a package.
- Terminate a Single Component Bundle: The subscriber ends only one component bundle while retaining the rest.

Example

Consider a subscriber submits an order with Supremo Silver Family Plan package, which includes a primary line bundle with the **Commitment** flag enabled and a secondary line bundle with commitment terms defined as follows:

Table 4-1 Example of Termination of Services for Supremo Silver Family Plan

Product Offer Name	Commitment Period	Grace Period	Disconnect Package	Apply Component Charge	Type	Penalty Amount	Pro Rate Plan
Supremo Silver Family Plan	1 year	15 days	NA	NA	Penalty	$50	Fixed Proration
Wireless Family Primary Line Bundle	1 year	15 days	Y	Y	Penalty	$500	Fixed Proration
Wireless Family Secondary Line Bundle	1 year	15 days	Y	Y	Penalty	$250	Fixed Proration

After the order is processed with above details, the subscriber submits a request to disconnect the Wireless Family Primary Line Bundle. As the **Commitment** flag is enabled, the Wireless Family Primary Line Bundle and the Package associated with it will be disconnected with the associated penalty amount.

After the disconnection happens:

- Validate the status of the package and all installed assets of the Wireless Primary Line Bundle in Siebel and Oracle Communications Billing Revenue Management (BRM).
- Validate the penalty charges applied on BRM.
- Validate the status of all installed assets of Wireless Secondary Line Bundle in Siebel and BRM.
- Wireless Secondary Line Bundle should be **Active** in both Siebel and BRM.
- Create bill on demand from BRM and validate the billed amount on BRM and Siebel.

> **Note**
>
> After a successful disconnection, neither the Package nor the Wireless Primary Line Bundle should be present in the installed assets. Only the Wireless Secondary Line Bundle should remain active and visible.

5

Implementing the Order to Cash Business Processes

This chapter describes the implementation of Order to Cash business processes and their features through the implementation process of order lifecycle management.

About the Order Lifecycle Management

Order lifecycle management includes the following:

- **Process Sales Order Fulfillment**: Siebel CRM submits orders to OSM.
  - Enabled using the Order to Cash business process options for OSM and Siebel CRM.
  - Used when submitting orders from Siebel CRM to OSM for order fulfillment processing.

  See About the Process Sales Order Fulfillment Flow.
- **Synchronize Fulfillment Order Billing Account**: OSM decomposes orders to create customer data in BRM.
  - Enabled using the Order to Cash business process options for OSM, BRM and Siebel CRM.
  - Used when interfacing orders to create customer data in BRM.

  See About the Synchronize Fulfillment Order Billing Account Flow.
- **Bill Fulfillment Order**: OSM decomposes orders to create transaction data in BRM.
  - Enabled using the Order to Cash business process options for OSM and BRM.
  - Used when interfacing orders to create transaction data in BRM.

  See About the Bill Fulfillment Order Flow.
- **Provision Order and Update Fulfillment Order**: OSM in the Central Order Management (COM) role decomposes and sends orders to OSM in the Service Order Management (SOM) role for provisioning.
  - Enabled using the Order to Cash business process 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 About the Provision Order and Update Fulfillment Order Flows.
- **Update Sales Order**: OSM sends order updates to Siebel CRM.
  - Enabled using the Order to Cash business processs options for OSM and Siebel CRM.
  - Used when sending order updates from OSM COM to Siebel CRM.

  See About the Update Sales Order Flow.

The following sections describe how the above business flows are implemented.

For more information about configuring order lifecycle management for Order to Cash, see Configuring Order Lifecycle Management.

Implementing the Process Sales Order Fulfillment

This section explains how the Order to Cash business process implements the Process Sales Order Fulfillment flow. It also describes how the Order to Cash business process supports order priorities and multiple price lists.

About the Process Sales Order Fulfillment Flow

The Process Sales Order Fulfillment 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 Order to Cash business process transforms the message and passes it to OSM for processing.

The Order to Cash business process 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.

For more information about creating Sales Orders, see About Sales Orders.

The following business process options enable the Process Sales Order Fulfillment flow:

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

The Process Sales Order Fulfillment flow includes the following:

- Submitting Orders from Siebel CRM to OSM

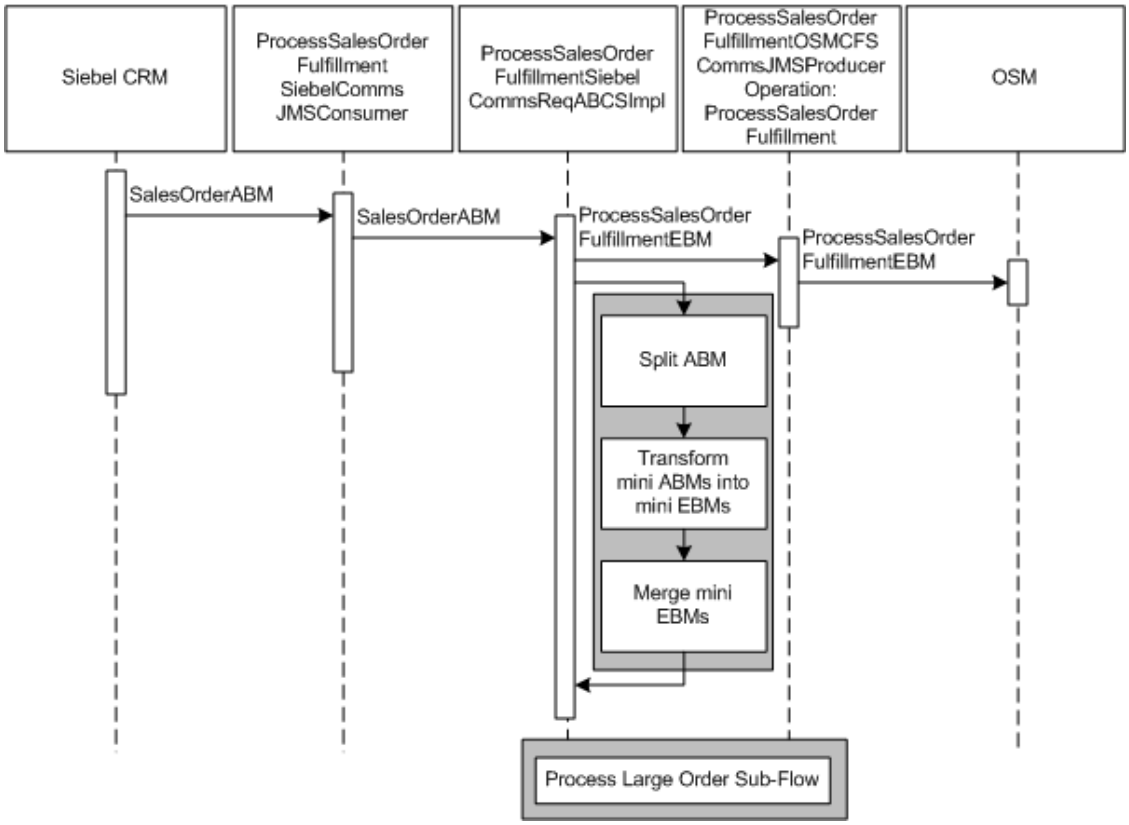
About Submitting Orders from Siebel CRM to OSM Integration

This flow uses the following services:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer

Figure 5-1 shows the sequence of integration services for the Submitting Orders from Siebel CRM to OSM integration flow.

Figure 5-1 Submitting Orders from Siebel CRM to OSM



The Order to Cash business process 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 ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer service picks up the ABM and passes it on to the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl service.
4. ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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**, ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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, ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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 About First-Time Purchase of Products and Services for more information on how OSM processes the order.

Defining Transaction Boundaries and Recovery Details

Table 5-1 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 5-1 Transaction Boundaries and Recovery Details for Siebel CRM Flow

Transaction	DB Operations	In Case of Error	Recovery
ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer passes the Siebel CRM message to ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl, 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. 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.

Assumptions and Constraints for the Process Sales Order Fulfillment Flow

The assumptions and constraints for the Process Sales Order Fulfillment 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.

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 ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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 Flow for information about how this message is used.

Siebel CRM Interfaces

The Process Sales Order Fulfillment 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 flow uses 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:

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

- OptimizedSalesOrder EBO files:

  COMMS_AIA_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 flow:

- ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- 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 ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl.

ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl

ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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, ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl 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:
  - subscriber 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 6-2 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.

Implementing the Synchronize Fulfillment Order Billing Account Flow

This section explains how the Order to Cash business process implements the Synchronize Fulfillment Order Billing Account flow.

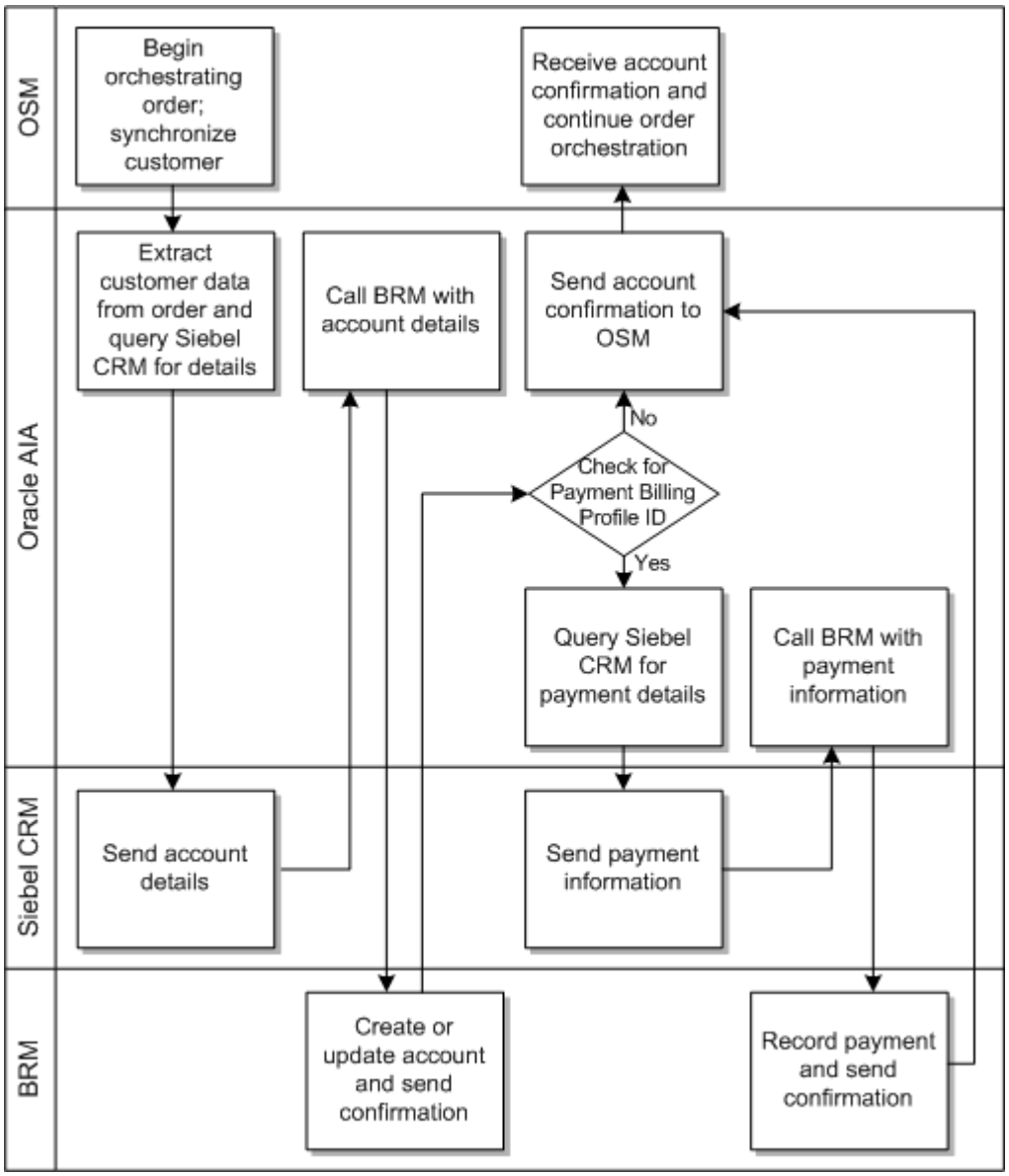
About the Synchronize Fulfillment Order Billing Account Flow

The Synchronize Fulfillment Order Billing Account flow lets you create subscriber data 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 flow creates the subscriber data necessary for order fulfillment without overburdening BRM with all of the subscriber information available in Siebel CRM. It also lets you process payments accepted at order time.

Figure 5-2 shows the sequence of events in this flow.

Figure 5-2 Synchronize Fulfillment Order Billing Account Flow



The following business process options enable the Synchronize Fulfillment Order Billing Account 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 flow includes Interfacing Orders to Create Subscriber Data.

> **Note**
>
> In this 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 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 Subscriber Data Integration Flow

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

See About Subscriber Management for more information about Subscriber Management.

The Interfacing Orders to Create Subscriber Data Integration integration flow processes lines with ADD, UPDATE, and MOVE-ADD.

The lines are processed as follows:

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

The Interfacing Orders to Create Subscriber Data 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.

Defining Transaction Boundaries and Recovery Details

For the Synchronize Fulfillment Order Billing Account flow there are two transaction boundaries. Table 5-2 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 5-2 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer sends the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which extracts relevant subscriber data. If account hierarchy is required, the EBF sends the message to ProcessAccountHierarchyListSiebelCommsProvABCSImpl, which queries Siebel CRM for the hierarchy. Siebel CRM responds and the ABCS sends the response to CommsProcessFulfillmentOrderBillingAccountListEBF. The EBF then sends the message to CommsProcessBillingAccountListEBF. This EBF sends the message to QueryCustomerPartyListSiebelProvABCSImplV2, 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.	Oracle AIA cross-reference tables updated. Message goes into the sequencer table.	Message available in Enterprise Manager Fusion Middleware Control in recoverable state.	Retry or abort the message from Enterprise Manager Fusion Middleware Control.
CommunicationsCustomerPartyEBSV2Resequencer sends a message to SyncCustomerPartyListBRMCommsProvABCSImpl, which invokes BRM opcodes to create accounts. BRM responds with a confirmation, which the ABCS sends to CommsProcessBillingAccountListEBF. CommsProcessBillingAccountListEBF sends a message to ProcessCollectionSharingBRMCommsProvABCSImpl, which invokes BRM opcodes to create collections sharing groups. BRM responds with a confirmation, which the ABCS sends to CommsProcessBillingAccountListEBF. The EBF sends the message to CommsProcessFulfillmentOrderBillingAccountListEBF, which sends it to ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer. The producer drops the message into AIA_UPDCUST_IN_JMSQ.	Oracle AIA cross reference tables updated. Message goes to AIA_UPDCUST_IN_JMSQ. Accounts and collections sharing groups created in BRM.	Roll back Oracle AIA cross reference updates. Roll back BRM account and collections sharing group creation. Return message to sequencer table.	Resubmit message from sequencer table.

> **Note**
>
> If any order contains more than one account and a failure occurs after any account is processed successfully but the subsequent account fails, then error recovery may become difficult based on the point of failure. Customers must first examine the point of failure and then determine if it's necessary to recover the BPEL instance from the recovery console.

For more information about rollback procedures, see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension*, which covers error handling and trace logging in Oracle AIA processes.

Assumptions and Constraints for the Synchronize Fulfillment Order Billing Account

The Synchronize Fulfillment Order Billing Account has the same assumptions and constraints as the Bill Fulfillment Order. 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 5-3 summarizes what is expected for the action on order lines. OSM and OSM O2A cartridges conform to these expectations.

Table 5-3 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	The order management system creates customer data in BRM if it does not already exist. If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM. If the owner account is changed, the order management system changes the collections sharing group in BRM. The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.

Table 5-3 (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
ADD	Revision	Cancellation. Manifests as a missing line on the revision.	DELETE	This action is ignored. 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. 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.
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.
UPDATE	Revision	Changes to service account, billing account, billing profile, or owner account.	UPDATE	The order management system creates customer data in BRM if it does not already exist. If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM. If the owner account is changed, the order management system changes the collections sharing group in BRM. The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.

Table 5-3 (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	Cancellation. Manifests as a missing line on the revision or the action changing to a "-" (NONE).	UPDATE	If the original update line created a new account and billing profile in BRM, then it cannot be undone. For the attributes that have changed on the original line, the order management system flips the values (old, new) on the compensation line. If a hierarchy has been updated, this reverts that update.
MOVE-ADD	New, but can change billing account and billing profile as part of a move-add.	Not Applicable	MOVE-ADD	Expects prior value fields to be populated for values that are changing from an existing asset.
MOVE-ADD	Revision	No changes to service account, billing account, or billing profile.	NONE	No changes.
MOVE-ADD	Revision	Changes to billing account or billing profile.	MOVE-ADD	The order management system creates customer data in BRM if it does not already exist. If revisions are made to attributes affecting account hierarchy, the order management system makes the change in BRM. The order management system indicates which attributes have changed by populating the prior value fields for the changed attributes.

Table 5-3 (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	Manifests as a missing line on the revision or the action changing to a "-" (The line is canceled).	MOVE-ADD	If the original MOVE-ADD line created a new account and billing profile in BRM, then it cannot be undone. For the attributes that have changed on the original line, the order management system flips the values (old, new) on the compensation line. If a hierarchy has been updated, this reverts that update.

> ⚠ **Caution**
>
> The process integration for billing management (delivered in the Cash to Care Business Process business process) 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: Cash to Care Business Process Implementation Guide* for more information.

BRM Interfaces

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

Table 5-4 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 subscriber account as the owner and the billing account as a member.

Table 5-4 (Cont.) BRM Interfaces

Opcode	Purpose
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 subscriber 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 subscriber account.
PCM_OP_COLLECTIONS_GROUP_DELETE	As part of the Create/Sync Account integration flow, deletes the collections sharing group owned by the subscriber account.
PCM_OP_CUST_COMMIT_CUSTOMER	As part of the Create/Sync Account integration flow, creates a new account with one or more **/billinfo** and **/payinfo** 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 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 Synchronized Subscriber Account 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 flow uses the following communications industry-specific Oracle AIA components:

- FufillmentOrderEBO
- 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:

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flow as part of the Interfacing Orders to Create Subscriber Data integration flow:

- ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer
- ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer
- CommsProcessFulfillmentOrderBillingAccountListEBF
- QueryReceivedPaymentListSiebelCommsProvABCSImpl
- CreateReceivedPaymentListBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer

Additional services are delivered with the Synchronize Fulfillment Order Billing Account flow as part of the Create/Sync Subscriber Account integration flow, which is part of the Synchronize Subscriber Account flow. See Integration Services in "Implementing the Synchronize Subscriber Account Flow" for more information about these services.

ProcessFulfillmentOrderBillingAccountListOSMCFSCommsJMSConsumer

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.

ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer

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 subscriber data from the order. It loops through every order line and extracts subscriber 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 ProcessFulfillmentOrderBillingAccountListOSMCommsJMSConsumer or ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer.
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 ProcessAccountHierarchyListSiebelCommsProvABCSImpl.
    b. Receives ProcessBillingAccountListResponseEBM from ProcessAccountHierarchyListSiebelCommsProvABCSImpl.
    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 QueryReceivedPaymentListSiebelCommsProvABCSImpl
    b. Receives QueryReceivedPaymentListResponseEBM from QueryReceivedPaymentListSiebelCommsProvABCSImpl
    c. Transforms QueryReceivedPaymentListResponseEBM to CreateReceivedPaymentListEBM and sends it to CreateReceivedPaymentListBRMCommsProvABCSImpl
    d. Receives CreateReceivedPaymentListResponseEBM from CreateReceivedPaymentListBRMCommsProvABCSImpl
7. Transforms ProcessBillingAccountListResponseEBM and CreateReceivedPaymentListResponseEBM into ProcessFulfillmentOrderBillingAccountListResponseEBM and sends it to ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer.

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.

QueryReceivedPaymentListSiebelCommsProvABCSImpl

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 CommsProcessFulfilmentOrderBillingAccountListEBF.

CreateReceivedPaymentListBRMCommsProvABCSImpl

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.

ProcessFulfillmentOrderBillingAccountListResponseOSMCFSCommsJMSProducer

This JMS producer service receives the ProcessFulfillmentOrderBillingAccountListResponseEBM and drops it into the AIA_UPDCUST_IN_JMSQ queue.

Implementing the Bill Fulfillment Order Flow

This section describes the Bill Fulfillment Order flow and explains how the Oracle Communications Order to Cash business process implements it.

About the Bill Fulfillment Order Flow

The Bill Fulfillment Order 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.

The following business process options enable the Bill Fulfillment Order 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 flow supports the following:

- Interfacing Orders to Create Transaction Data in BRM

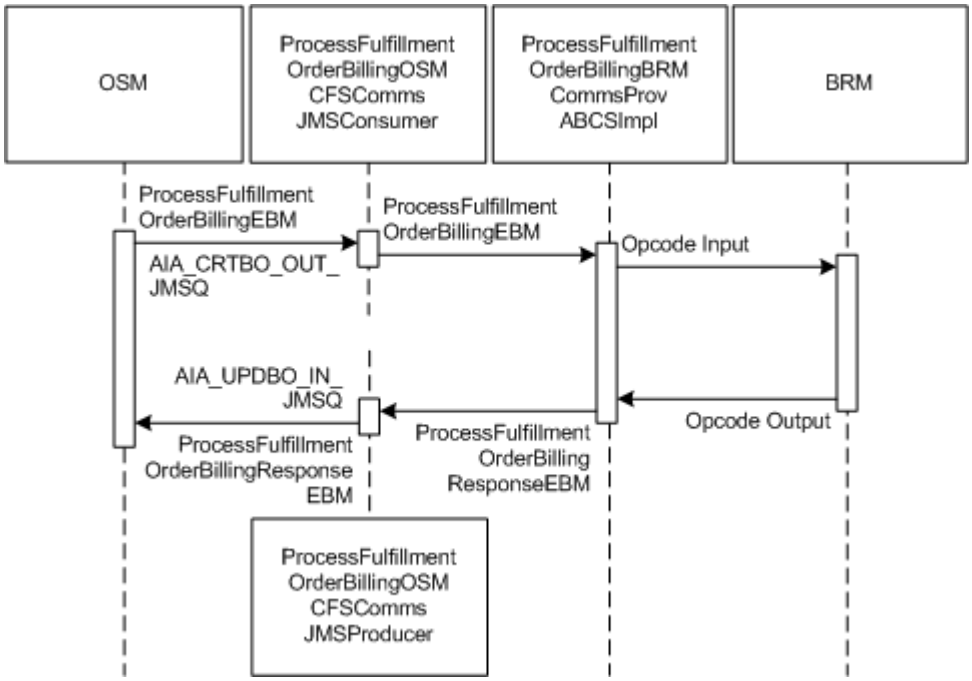
About the Interfacing Orders to Create Transaction Data in BRM Integration Flow

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

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingOSMCFSCommsJMSProducer

Figure 5-3 shows the sequence of integration services for the Interfacing Orders to Create Transaction Data in BRM integration flow.

Figure 5-3 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 flow, OSM creates a ProcessFulfillmentOrderBillingEBM message and drops it into the AIA_CRTBO_OUT_JMSQ JMS queue.
2. The ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer 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 ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer service.
5. ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer 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 5-5 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:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer

Table 5-5 Transaction Boundaries and Recovery Details

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

See 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.

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 Cash to Care Business Process 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. This means 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 *Oracle Communication Digital Business Experience Concept to Market Implementation Guide* for more information.

BRM Interfaces

The Bill Fulfillment Order 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_SUBCRIPTION

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

Industry Oracle AIA Components

The Bill Fulfillment Order flow uses these industry components:

- FufillmentOrderEBO
- ProcessFulfillmentOrderBillingEBM
- ProcessFulfillmentOrderBillingResponseEBM

The following directories contain the industry component files:

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

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flow:

- ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
  - ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
  - ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
  - ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
  - ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

  - ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
  - ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess
- ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer

ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer

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

This service has one operation: Consume_Message.

ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl

The ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 5-6.

Table 5-6 Subprocesses Called by Action Code, Billing Mode, and Fulfillment Mode

Action Code	Billing Mode	Fulfillment Mode	Subprocess
ADD	INITIATE BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsAddSubProcess and ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess
ADD	FULFILL BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsAddSubProcess and ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess
ADD	INITIATE BILLING	REDO	ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
ADD	INITIATE BILLING	UNDO	ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
DELETE	FULFILL BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess and ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess
SUSPEND or RESUME	FULFILL BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

Table 5-6 (Cont.) Subprocesses Called by Action Code, Billing Mode, and Fulfillment Mode

Action Code	Billing Mode	Fulfillment Mode	Subprocess
UPDATE	FULFILL BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess and ProcessFulfillmentOrderBillingBRMCommsSharingGroupSubProcess
MOVE-ADD	FULFILL BILLING	DO	ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess and ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess

- For Promotion on Invoice, ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 order lines, ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product.

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is called for a service bundle, account-level product, or account-level discount that is being newly added either as a part of a new order or an update order and that has an action code of Add.

The ProcessFulfillmentOrderBillingBRMCommsAddSubProcess is never called for any 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 6-2 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingMove.

The structure of the message coming in ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess comprises:

- ProcessFulfillmentOrderBillingEBM
- XREFPopulate
- XREFDelete

When the action code on the order line is 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, ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl invokes ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess.

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

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

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

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

ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess

The structure of the message coming in the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess comprises:

- ProcessFulfillmentOrderBillingEBM
- XREFPopulate
- XREFDelete

The ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess is a synchronous BPEL process that is called by the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl. This call depends on the action code present on the order line and also the type of product. It has one operation: processBillingSuspendResume.

When the action code is Suspend or Resume and the ProductType is a service bundle or an account-level subscription or account-level discount, then the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess is called.

The following operations are done by this process:

- When the action code is Suspend or Resume and the product type is a service bundle.

  ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess converts the ProcessFulfillmentOrderBillingEBM into 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 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.

ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer

The ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer is a BPEL process that has an adapter service, which produces the order response messages to the AIA_UPDBO_IN_JMSQ.

This process has only one operation: Produce_ProcessFOBResponse to produce the message into the AIA_UPDBO_IN_JMSQ AIA queue. This operation is called after the order is interfaced into BRM.

Implementing the Provision Order and Update Fulfillment Order Flows

This section explains how the Order to Cash business process implements the Provision Order and Update Fulfillment Order flows.

About the Provision Order and Update Fulfillment Order Flows

The following sections provide an overview of Provision Order and Update Fulfillment Order flows.

About the Provision Order Flow

In this 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 About Order Fallout Management for more information about order fallout.

About the Update Fulfillment Order Flow

In this flow, OSM SOM sends milestone, status, and data updates to OSM COM.

You use OSM COM to manages 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.

The following business process options enable the Provision Order and Update Fulfillment Order flow:

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

The Provision Order and Update Fulfillment Business Order 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 flow and the Update Fulfillment Order flow.

For the Provision Order 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).

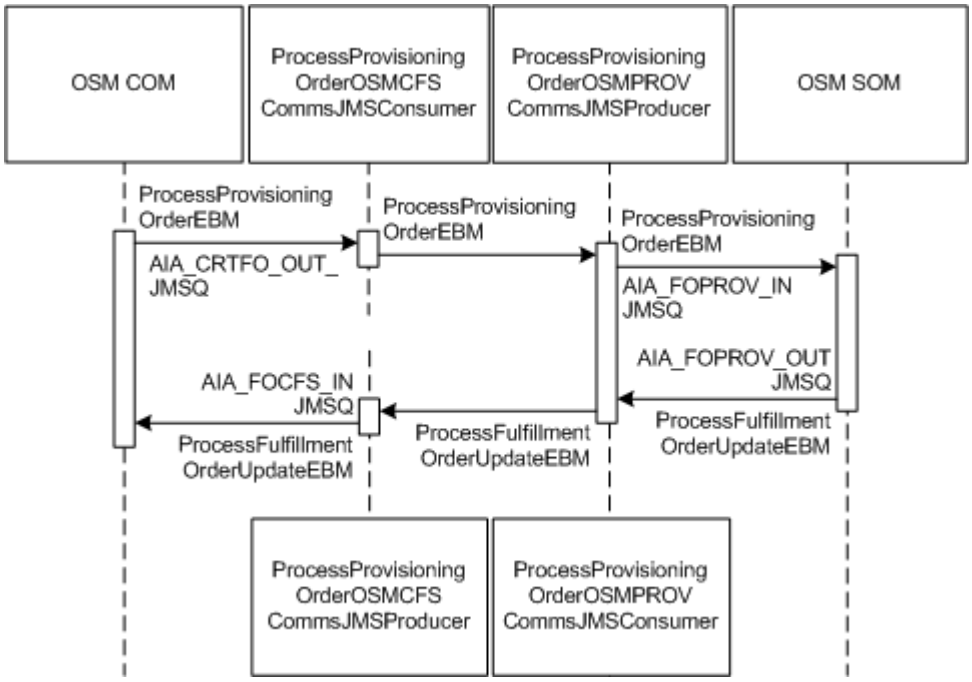
For the Update Fulfillment Order flow, the Interfacing Orders to Create Subscriber Data Integration flow passes milestone, status, and data updates from OSM SOM to OSM COM.

This integration flow uses the following services:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer

Figure 5-4 shows the sequence of integration services for the OSM Fulfillment to OSM Provisioning integration flow.

Figure 5-4 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 ProcessProvisioningOrderOSMCFSCommsJMSConsumer 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 ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer service picks up ProcessFulfillmentOrderUpdateEBM and sends it to the ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer service.
6. ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer service drops ProcessFufillmentOrderUpdateEBM 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. Defining Transaction Boundaries and Recovery Details describes the transactions involved, the database operations, and what actions to take in case of an error.

See Using Error Type to Control Response to Order Fallout for more information about system and business errors.

The following services are involved:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

Table 5-7 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessProvisioningOrderOSMCFSCommsJMSConsumer routes the message to ProcessProvisioningOrderOSMPROVCommsJMSProducer, which produces message into AIA_FOPROV_IN_JMSQ. SAF puts message in OSM.	None	Rollback JMS message to originating queue AIA_CRTFO_OUT_JMSQ_ErrorQ.	Resubmit the message from AIA_CRTFO_OUT_JMSQ_ErrorQ.
During provisioning, update messages are dequeued by OSM and eventually moves to AIA_FOPROV_OUT_JMSQ. ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer routes the message to ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer, which produces the message to AIA_FOCFS_IN_JMSQ.	None	Rollback JMS message to the originating queue AIA_FOPROV_OUT_JMSQ_ErrorQ.	Resubmit the order from AIA_FOPROV_OUT_JMSQ_ErrorQ.

See 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 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:

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flows:

- ProcessProvisioningOrderOSMCFSCommsJMSConsumer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMPROVCommsJMSConsumer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer

ProcessProvisioningOrderOSMCFSCommsJMSConsumer

For interacting with OSM Provisioning, OSM Fulfillment pushes ProcessProvisioningOrderEBM message into AIA_CRTFO_OUT_JMSQ using store-and-forward.

The ProcessProvisioningOrderOSMCFSCommsJMSConsumer is a Mediator process that has a JMS Adapter Service. This Mediator service continuously polls the AIA_CRTFO_OUT_JMSQ. The ProcessProvisioningOrderOSMCFSCommsJMSConsumer dequeues the ProcessProvisioningOrderEBM and routes it to the ProcessProvisioningOrderOSMPROVCommsJMSProducer.

This service has one operation: Consume_Message.

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 ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer.

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

Implementing the Update Sales Order Flow

This section explains how the Order to Cash business process implements the Update Sales Order flow.

About the Update Sales Order Flow

The following process options enable the Update Sales Order flow:

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

The Update Sales Order flow includes the following integration flow:

- Updating Statuses from OSM to Siebel CRM

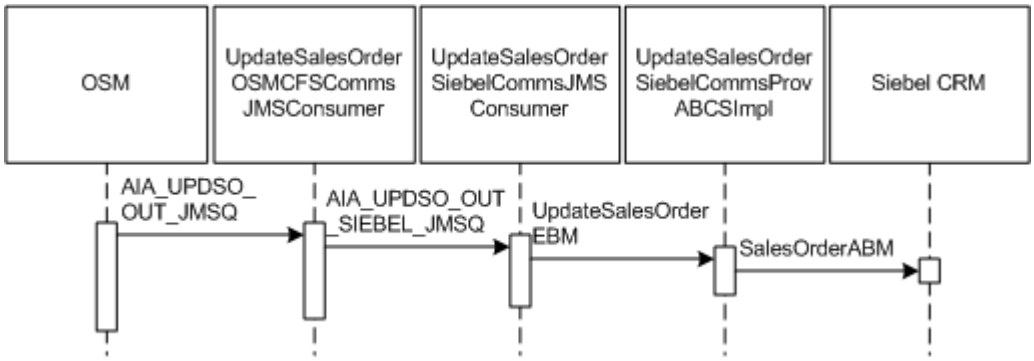
About the Updating Statuses from OSM to Siebel CRM Integration Flow

This integration flow uses the following services:

- Integration services for OSM:
  - UpdateSalesOrderOSMCFSCommsJMSConsumer
- Integration services for Siebel CRM:
  - UpdateSalesOrderSiebelCommsJMSConsumer
  - UpdateSalesOrderSiebelCommsProvABCSImpl

Figure 5-5 shows the sequence of integration services for the Updating Status from OSM to Siebel CRM integration flow.

Figure 5-5 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 UpdateSalesOrderOSMCFSCommsJMSConsumer 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 UpdateSalesOrderSiebelCommsProvABCSImpl 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 About Order Priority and Implementing the Synchronize Subscriber Account Flow 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, UpdateSalesOrderSiebelCommsProvABCSImpl 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 5-8 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:

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderSiebelCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCSImpl

Table 5-8 Transaction Boundaries and Recovery Details for the Siebel CRM Flow

Transaction	DB Operations	In Case of Error	Recovery
UpdateSalesOrderOSMCFSCommsJMSConsumer picks up UpdateSalesOrderEBM from AIA_UPDSO_OUT_JMSQ and adds it to AIA_UPDSO_OUT_SIEBEL_JMSQ.	Message added to queue.	Roll back the message to AIA_UPDSO_OUT_JMSQ_ErrorQ.	Resubmit the order from AIA_UPDSO_OUT_JMSQ_ErrorQ.
UpdateSalesOrderSiebelCommsJMSConsumer picks up UpdateSalesOrderEBM from AIA_UPDSO_OUT_SIEBEL_JMSQ and adds it to the sequencer table defined by the UpdateSalesOrderSiebelCommsJMSConsumer_RS routing service.	Message added to the sequencer table.	Message available in Enterprise Manager Fusion Middleware Control in recoverable state.	Retry or abort the message from Enterprise Manager Fusion Middleware Control.
UpdateSalesOrderSiebelCommsJMSConsumer_RS sends the message to UpdateSalesOrderSiebelCommsProvABCSImpl, which invokes the Siebel web service to update the order.	AIA cross-reference entries.	Roll back the message to the sequencer table.	Resubmit the order from the 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 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 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:

COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flow:

- UpdateSalesOrderOSMCFSCommsJMSConsumer
- UpdateSalesOrderSiebelCommsJMSConsumer
- UpdateSalesOrderSiebelCommsProvABCSImpl

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 UpdateSalesOrderSiebelCommsProvABCSImpl service by using a Best Effort resequencer.

UpdateSalesOrderSiebelCommsProvABCSImpl

The UpdateSalesOrderSiebelCommsProvABCSImpl 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 subscriber 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.

Implementing the Synchronize Subscriber Account Flow

This section explains how the Order to Cash business process implements the Synchronize Subscriber Account flow.

About the Synchronize Customer Account Flow

The following business process options enable the Synchronize Subscriber Account 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 Subscriber Account flow supports the following integration flows:

- Create/Sync Subscriber Account
- Update Subscriber Account

About the Create/Sync Subscriber Account Integration Flow

The Create/Sync Subscriber Account integration flow synchronizes subscriber 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 Subscriber Data integration flow of the Synchronize Fulfillment Order Billing Account business flow.

See About the Synchronize Fulfillment Order Billing Account Flow for information about the sequence of events for this business flow.

About the Update Subscriber Account Integration Flow

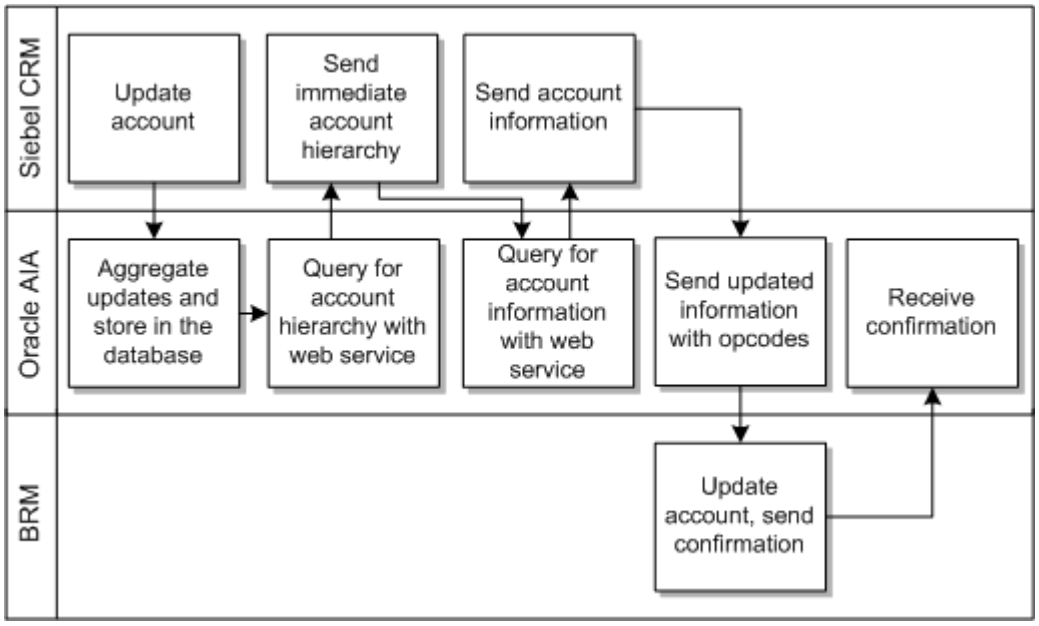
This flow synchronizes account updates from Siebel CRM to BRM.

The Update Subscriber Account integration flow uses the following services:

- SyncCustomerSiebelEventAggregator
- SyncAcctSiebelEventAggrConsumer
- ProcessAccountHierarchyListSiebelCommsProvABCSImpl
- SyncAccountSiebelReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListBRM_01CommsJMSConsumer
- SyncCustomerPartyListBRMCommsProvABCSImpl

Figure 5-6 illustrates the Update Subscriber Account integration flow.

Figure 5-6 Update Subscriber Account Sequence



The integration updates subscriber 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 sends 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 subscriber in sequence.
6. SyncAcctSiebelEventAggrConsumer sends the account ID and child IDs to the SyncAccountSiebelReqABCSImpl service.
7. SyncAccountSiebelReqABCSImpl 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 SyncAccountSiebelReqABCSImpl.
9. SyncAccountSiebelReqABCSImpl 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 SyncCustomerPartyListBRMCommsProvABCSImpl.

    You can define more consumers for multiple instances of BRM or other billing systems and subscribe them to CPARTY_SYN_TOPIC.

12. SyncCustomerPartyListBRMCommsProvABCSImpl 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 SyncCustomerPartyListBRMCommsProvABCSImpl.

BRM Interfaces

Table 5-9 lists the BRM interfaces used by the Synchronize Subscriber Account flow.

Table 5-9 BRM Interfaces

Opcode	Purpose
PCM_OP_CUST_COMMIT_CUSTOMER	Creates a new account with one or more **/billinfo** and **/payinfo** objects.
PCM_OP_CUST_UPDATE_CUSTOMER	Updates account information (name, address, phone), contact information, and billing information.

Table 5-9 (Cont.) BRM Interfaces

Opcode	Purpose
PCM_OP_CUST_DELETE_PAYINFO	Delete a **/payinfo** 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 *Oracle Communications Billing and Revenue Management Developer's Reference* for detailed descriptions of individual opcodes.

Siebel CRM Interfaces

Table 5-10 describes the Siebel CRM web service interface.

Table 5-10 Siebel CRM Web Service Interface

Web Service	Description	Used by
Query Account- (SWICustomerParty)	Retrieves account, bill profile, contact, and address data from Siebel CRM.	• QueryCustomerPartyListSiebelProvABCSImplV2 as part of creating a new account or adding a new billing profile to an existing account. • SyncAccountSiebelReqABCSImpl
Query account hierarchy - (SWI Account Hierarchy)	Retrieves the full linear account hierarchy from Siebel CRM.	ProcessAccountHierarchyListSiebelCommsProvABCSImpl

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

Table 5-11 describes the Siebel CRM workflow event interfaces.

Table 5-11 Siebel CRM Workflow Event Interfaces

Event	Description	Consumed by
SWI Account Updated	This workflow event is started when an account is updated in Siebel CRM.	This event message is consumed by the SyncCustomerSiebelEventAggregator.aggregateaccountevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ENTITIES).
SWI Bill Profile Updated	This workflow event is started when a bill profile is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator.aggregatebpevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ENTITIES).
SWI Contact Updated	This workflow event is started when a contact is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator.aggregatecontactevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ENTITIES).
SWI Address Updated	This workflow event is started when an address is updated in Siebel CRM.	The event message is consumed by the SyncCustomerSiebelEventAggregator.aggregateaddressevent service, which extracts all the relevant IDs from the input payload and stores them in a database table (AIA_AGGREGATED_ENTITIES).

See 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 Subscriber Account 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:

COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 Subscriber Account flow:

- CommsProcessBillingAccountListEBF
- CommunicationsCustomerPartyEBSV2Resequencer
- SyncCustomerPartyListBRMCommsProvABCSImpl
- SyncCustomerPartyListBRMCommsJMSProducer
- SyncCustomerPartyListBRM_01CommsJMSConsumer
- CustomerPartyEBSV2
- ProcessCollectionSharingBRMCommsProvABCSImpl
- ProcessAccountHierarchyListSiebelCommsProvABCSImpl
- QueryCustomerPartyListSiebelProvABCSImplV2
- SyncCustomerSiebelEventAggregator
- SyncAccountSiebelReqABCSImpl
- SyncContactSiebelAggregatorAdapter
- SyncAddressSiebelAggregatorAdapter
- SyncBPSiebelAggregatorAdapter
- SyncAcctSiebelAggrEventConsumer
- SyncAccountSiebelReqABCSImpl

Additional services are delivered with the Synchronize Subscriber Account flow as part of the Interfacing Orders to Create Customer Data integration flow, which is part of the Synchronize Fulfillment Order Billing Account flow. See Integration Services in Implementing the Synchronize Fulfillment Order Billing Account 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 subscriber 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 subscriber 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 subscriber 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 subscriber 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_01CommsJMSConsumer picks it up.

SyncCustomerPartyListBRM_01CommsJMSConsumer

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 SyncCustomerPartyListBRMCommsProvABCSImpl service.

This process is intended for a topology with multiple BRM instances. Deploy a version of this consumer for each BRM instance.

CustomerPartyEBSV2

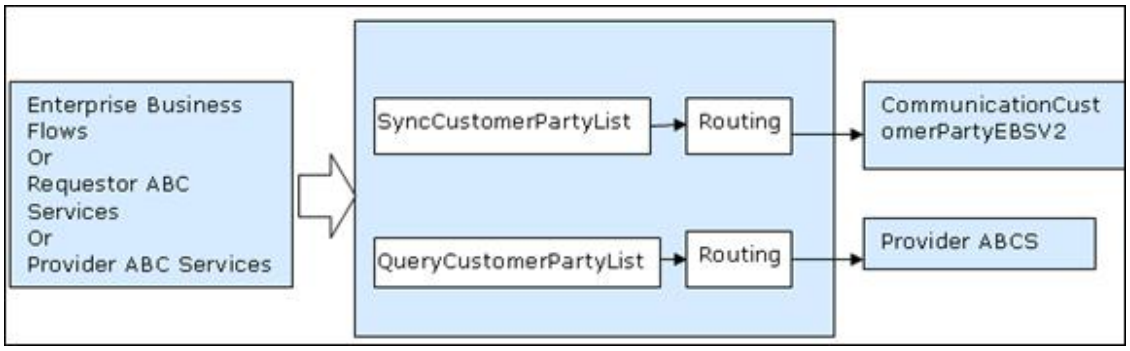
CustomerPartyEBSV2 a lightweight EBS routing service that exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object.

CustomerPartyEBSV2 service uses the following operations:

- SyncCustomerPartyList
- QueryCustomerPartyList

Figure 5-7 illustrates the relationship of CustomerPartyEBSV2 with the other services in the integration flow.

Figure 5-7 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 SyncCustomerPartyListBRMCommsProvABCSImpl 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

ProcessCollectionSharingBRMCommsProvABCSImpl

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.

ProcessAccountHierarchyListSiebelCommsProvABCSImpl

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.

QueryCustomerPartyListSiebelProvABCSImplV2

This service does the following:

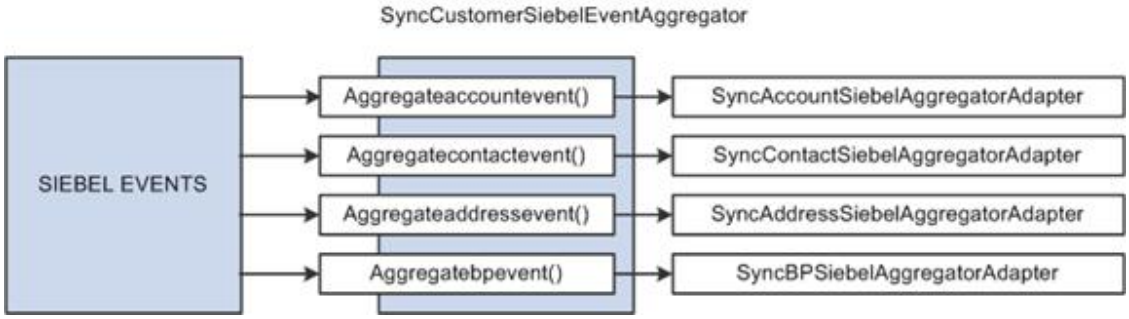
1. Receives QueryCustomerPartyListEBM from CommsProcessBillingAccountListEBF.
2. Transforms QueryCustomerPartyListEBM into a Siebel CRM ABM.
3. Invokes the Siebel CRM SWICustomerParty web service.
4. Receives the response from SWICustomerParty.
5. Transforms the response into QueryCustomerPartyListResponseEBM 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 5-8 illustrates the relationship of the SyncCustomerSiebelEventAggregator with the other services in the integration flow.

Figure 5-8 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.
- Aggregatecontacttevent:
  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.
- Aggregateaddresstevent:
  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 subscriber 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.

SyncAccountSiebelReqABCSImpl

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 Flow

This chapter describes the Synchronize Customer Special Rating Profile flow. It explains how the Oracle Communications Order to Cash business process for Siebel CRM and Oracle Communications Billing and Revenue Management (BRM) implements this flow using BRM interfaces, Siebel CRM interfaces, industry-specific Oracle Application Integration Architecture (AIA) components, and integration services.

Overview of the Synchronize Customer Special Rating Profile Flow

The Synchronize Customer Special Rating Profile flow supports the following integration flow:

- Synchronize Friends and Family List Updates to BRM

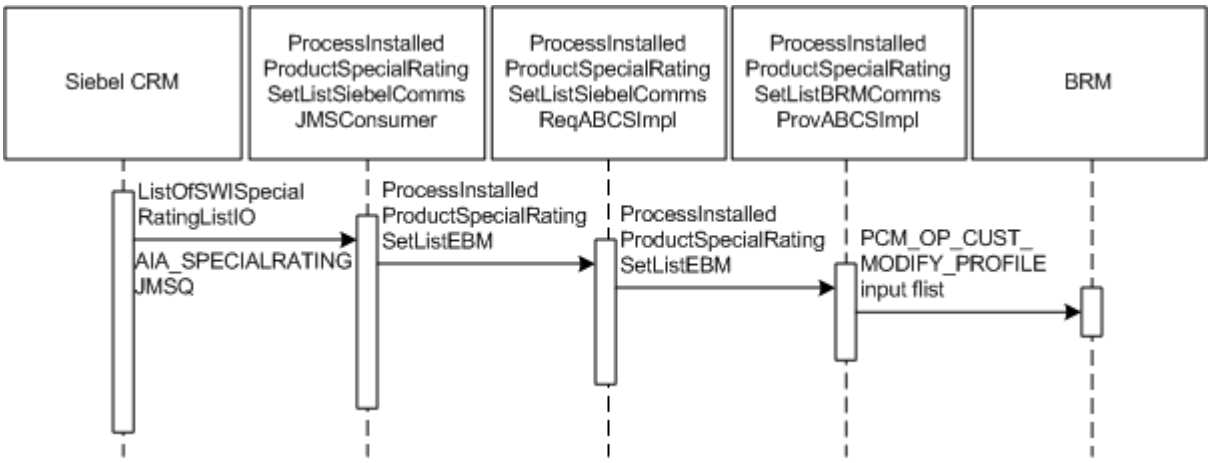
Synchronize Friends and Family List Updates to BRM

This flow uses the following services:

- ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

Figure 5-9 illustrates the sequence of events in the integration flow for synchronizing Friends and Family List Updates to BRM.

Figure 5-9 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 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl service.
3. ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl transforms ListOfSWISpecialRatingListIO into the ProcessInstalledProductSpecialRatingSetListEBM message and sends it to the ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl service.
4. ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl 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 5-12 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
- ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl
- ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

Table 5-12 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
ProcessInstalledProductSpecialRatingSetListSiebelCommsJMSConsumer picks up message and routes it to ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl, which transforms message and routes to ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl. The opcode is called to update information in 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" and "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 the 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 provided 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 ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl service.

ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl

The ProcessInstalledProductSpecialRatingSetListSiebelCommsReqABCSImpl 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 ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl service.

This service is invoked when the existing special rating (Friends and Family) profile for an account, present in Siebel CRM and synchronized with BRM, is modified.

The service looks up the cross-reference values for the customer account ID and installed product ID to identify the corresponding common IDs used to populate the EBM appropriately.

ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl

The ProcessInstalledProductSpecialRatingSetListBRMCommsProvABCSImpl 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 for an account, present in Siebel CRM and synchronized with 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.

Implementing the Create Trouble Ticket from Oracle AIA Flow

This section describes the Create Trouble Ticket from Oracle AIA flow and explains how the Order to Cash for Siebel CRM and Oracle Communications Billing and Revenue Management (BRM) business process options implements it using Siebel CRM interfaces, communications industry-specific Oracle Application Integration Architecture (Oracle AIA) components, integration services, and fallout-enabled services.

About the Create Trouble Ticket from Oracle AIA 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. This interface is used for the creation of trouble tickets in Siebel CRM.

This feature is composed of the following services:

- CreateTroubleTicketAIACommsReqImpl - Oracle AIA Requestor application business connector service (ABCS).
- CreateTroubleTicketSiebelCommsProvABCSImpl - Oracle AIA Provider ABCS invoked to create a trouble ticket in Siebel CRM.

On an error, the order fallout process (detection) within Oracle AIA passes the order fault message that is queued in the Oracle AIA Error JMS Topic to the

CreateTroubleTicketAIACommsReqImpl ABCS. The service then routes the Oracle AIA message to the Siebel provider, which in turn calls the Siebel web service to create the trouble ticket in Siebel.

This flow supports the following integration flows:

- Creating a Trouble Ticket in Siebel CRM

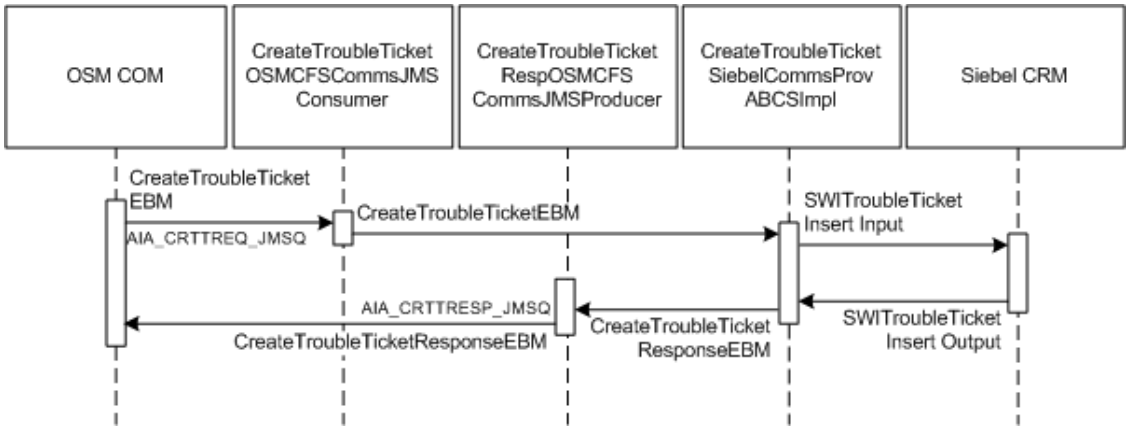
Creating a Trouble Ticket in Siebel CRM Integration Flow

This integration flow uses the following interfaces:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCSImpl

Figure 5-10 illustrates the create trouble ticket integration flow.

Figure 5-10 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 5-13 describes the transactions involved, the database operations, and what actions to take in case of an error.

See Table 6-2 for more information on system and business errors.

The following services are involved:

- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- CreateTroubleTicketSiebelCommsProvABCSImpl

Table 5-13 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
The AIAOrderFalloutJMSBridgeService picks up the messages with the JMSCorrelationID and publishes to AIA_ORDERFALLOUT_JMSQ.	Message enqueued in AIA_ORDERFALLOUT_JMSQ.	Rollback JMS message to AIA_ERROR_TOPIC.	Resubmit from AIA_ERROR_TOPIC.
AIACOMOrderFalloutNotificationJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_TTS and invokes CreateTroubleTicketAIACommsReqImpl, which parses fault message and routes to CreateTroubleTicketSiebelCommsProvABCSImpl.	AIA cross-reference entries.	Rollback the message to AIA_ORDERFALLOUT_JMSQ.	Resubmit from AIA_ORDERFALLOUT_JMSQ.

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 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 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:

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flow:

- CreateTroubleTicketSiebelCommsProvABCSImpl
- AIAOrderFalloutJMSBridgeService
- AIACOMOrderFalloutNotificationJMSConsumer
- CreateTroubleTicketAIACommsReqImpl
- AIAOrderFalloutErrorHandlerExtension - Java Class

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.

CreateTroubleTicketSiebelCommsProvABCSImpl

The CreateTroubleTicketSiebelCommsProvABCSImpl 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 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.

Fallout-Enabled Services

The following Create Trouble Ticket from Oracle AIA flow services are fallout-enabled:

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess
- ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl
- ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- QueryCustomerPartyListSiebelProvABCSImplV2
- SyncCustomerPartyListBRMCommsProvABCSImpl

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.

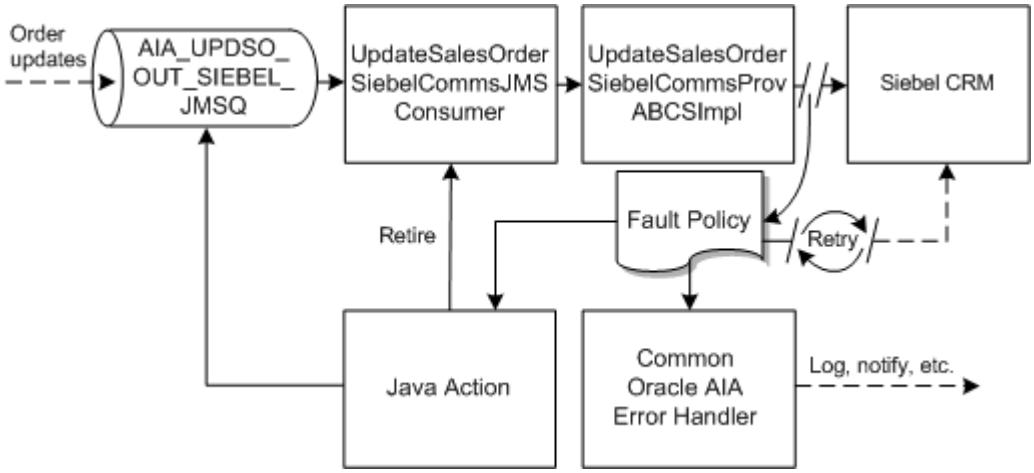
If you do not want Oracle AIA to handle system errors in this way, you can disable the Java action which retires the consumers. See Disabling Consumer Service Retirement for more information.

The following consumer services are enabled to handle system errors by default:

- UpdateSalesOrderSiebelCommsJMSConsumer
- SyncItemCompositionListSiebelCommsJMSConsumer
- ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer
- ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer

Figure 5-11 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 5-11 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 UpdateSalesOrderSiebelCommsProvABCSImpl 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***/em**

   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:
   - UpdateSalesOrderSiebelCommsJMSConsumer
   - SyncItemCompositionListSiebelCommsJMSConsumer
   - ProductOptimizedSyncPriceListListSiebelCommsJMSConsumer
   - ProcessFulfillmentOrderBillingAccountListSiebelCommsJMSConsumer
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"/>
   ```

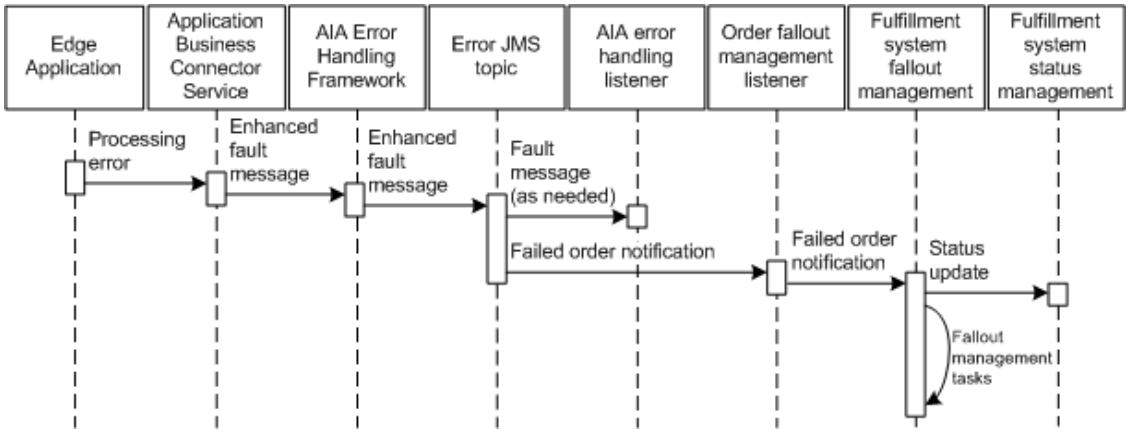
4. Save and close the file.
5. Upload the file to the Oracle Metadata Services repository. See "Update Files to MDS" in the *Oracle Application Integration Architecture Cloud Native Deployment Guide* for more information about uploading changed files to the Oracle Metadata Services repository.

How Oracle AIA Error Handling Framework Captures Faults

The Oracle AIA Error Handling Framework is used to capture faults across order processing.

Figure 5-12 illustrates the interactions taking place when an order failure is detected by an edge application, such as a billing or inventory system.

Figure 5-12 Capturing the Fault Sequence Diagram



The integration captures faults as follows:

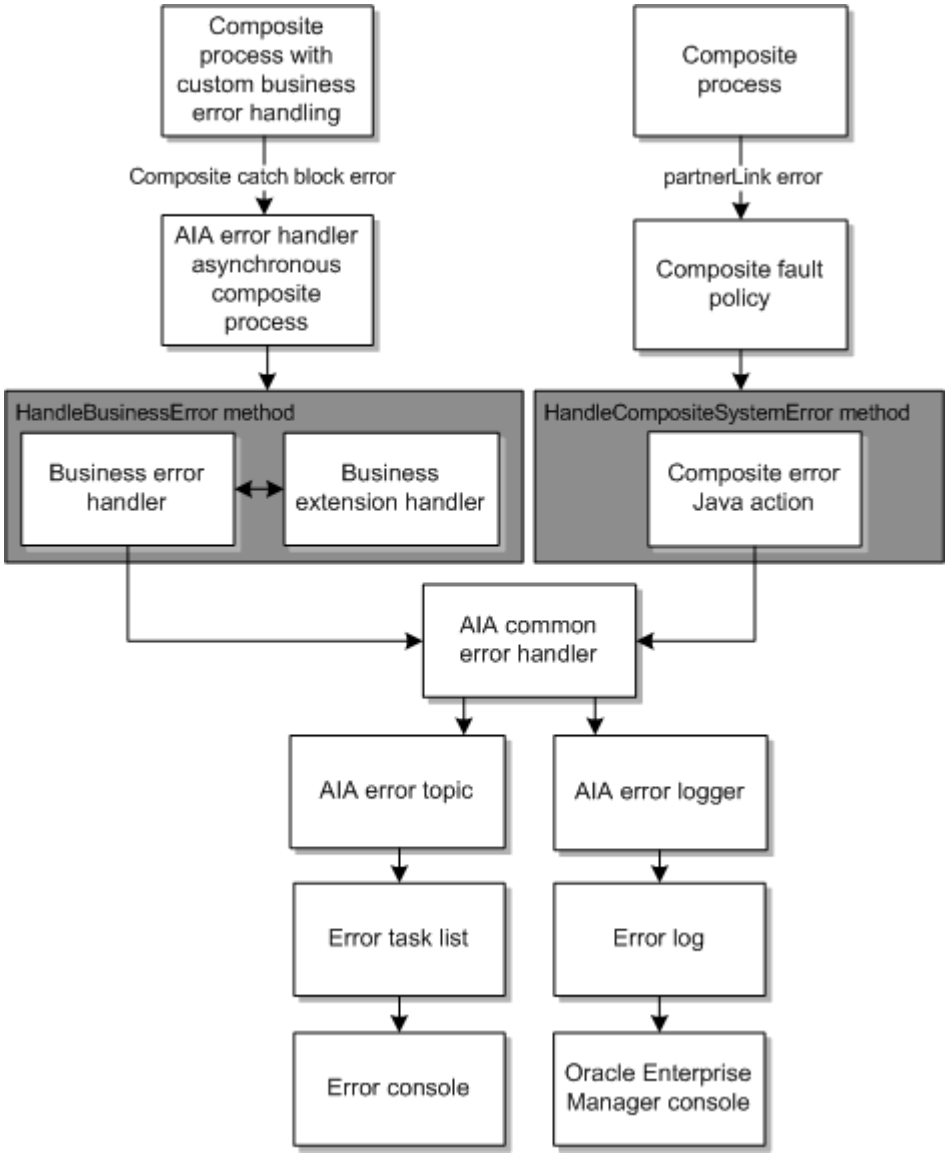
1. An edge application, such as BRM, encounters a processing error and sends the error to the Oracle AIA ABCS.
2. The ABCS creates an enhanced fault message from the error message and sends it to the Oracle AIA error handling framework.
3. The error handling framework sends the enhanced fault message 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.

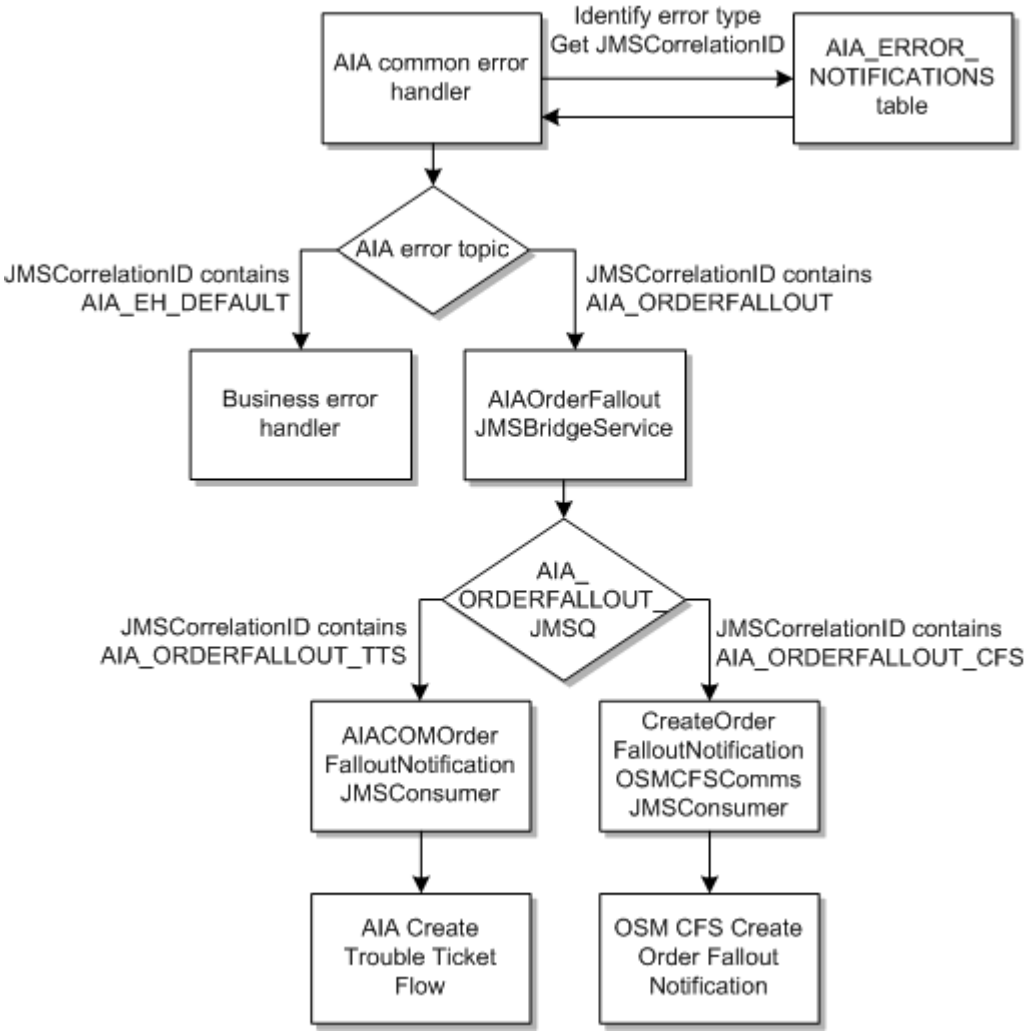
Figure 5-13 illustrates how the Oracle AIA Error Handling Framework submits an order failure notification to the Oracle AIA Error Topic.

Figure 5-13 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 flow, as shown in Figure 5-14.

Figure 5-14 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 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 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.

Implementing the Create and Manage Trouble Ticket from OSM Flow

This section explains how the Order to Cash business process implements the Create and Manage Trouble Ticket from OSM flow.

About the Create and Manage Trouble Tickets from OSM Flow

The following business process options enable the Create and Manage Trouble Ticket from OSM 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 flow includes the following 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

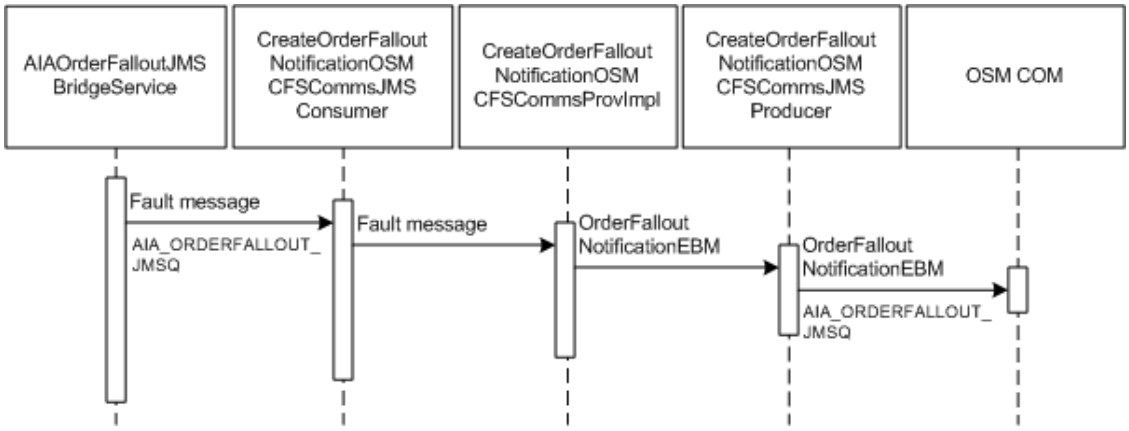
The Oracle AIA order fallout listener, AIAOrderFalloutJMSBridgeService, listens to all messages published to the Oracle AIA Error JMS Topic, AIA_ERROR_TOPIC. Errors or faults with a correlation ID of AIA_ORDERFALLOUT_CFS are enriched and published to OSM.

This integration flow includes the following services:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer

Figure 5-15 shows the sequence of integration services through which the fallout notification message passes for the Order Failure Notification to OSM integration flow.

Figure 5-15 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 CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer service picks up the message and sends it to the CreateOrderFalloutNotificationOSMCFSCommsProvImpl service.
3. CreateOrderFalloutNotificationOSMCFSCommsProvImpl transforms the message into the OrderFalloutNotificationEBM message and sends it to the CreateOrderFalloutNotificationOSMCFSCommsJMSProducer service.
4. CreateOrderFalloutNotificationOSMCFSCommsJMSProducer 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.

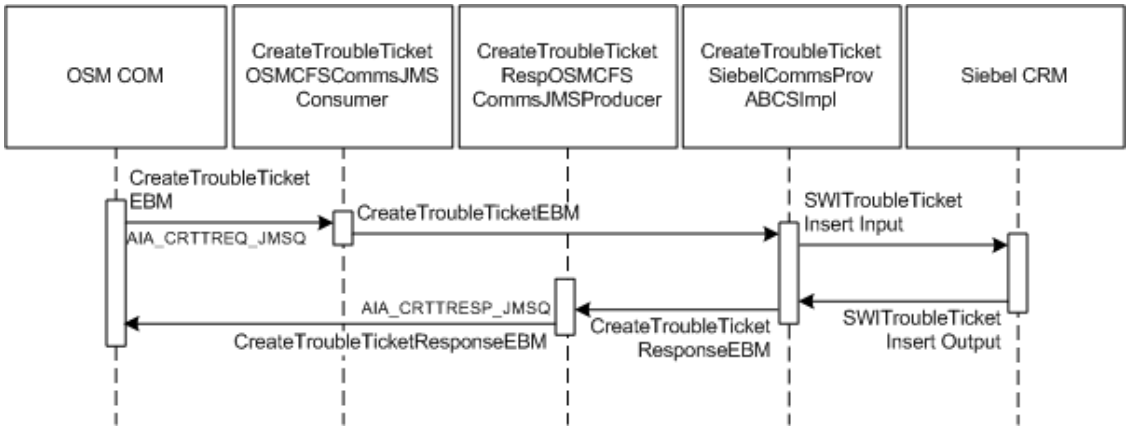
About the Creating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow includes the following services:

- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCSImpl
- CreateTroubleTicketOSMCFSCommsJMSProducer

Figure 5-16 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 5-16 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_CRTTTREQ_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_CRTTTRESP_JMSQ queue.
7. OSM COM picks up the message and stores the trouble ticket ID for reference.

Defining Transaction Boundaries and Recovery Details

Table 5-14 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 5-14 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.
CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer picks up messages with the JMS Correlation ID AIA_ORDERFALLOUT_CFS and sends it to CreateOrderFalloutNotificationOSMCFSCommsProvImpl, which parses fault message and sends it to CreateOrderFalloutNotificationOSMCFSCommsJMSProducer.	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 CreateTroubleTicketSiebelCommsProvABCSImpl. The ABCS invokes a Siebel web service., which sends a response message with the trouble ticket ID to CreateTroubleTicketSiebelCommsProvABCSImpl. The ABCS sends the response to CreateTroubleTicketRespOSMCFSCommsJMSProducer.	AIA cross-references created. 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.

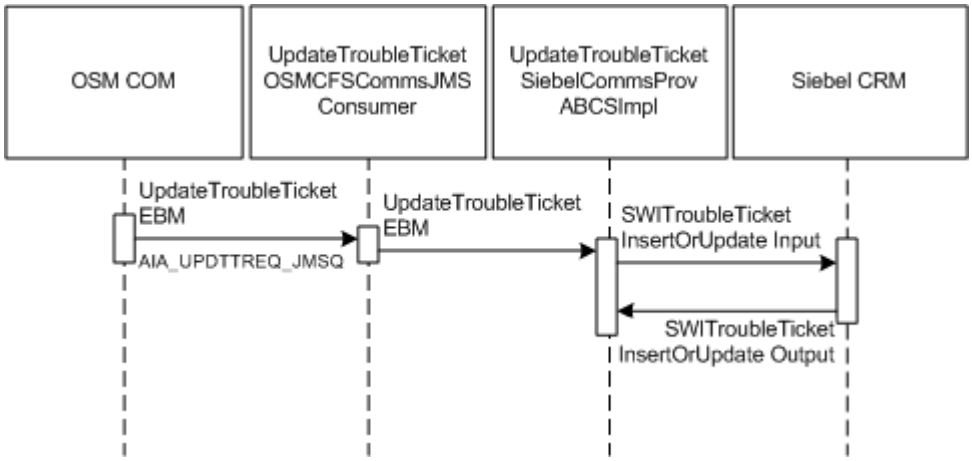
Updating a Trouble Ticket in Siebel CRM from OSM Integration Flow

This integration flow uses the following interfaces:

- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCSImpl

Figure 5-17 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 5-17 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 UpdateTroubleTicketOSMCFSCommsJMSConsumer 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 5-15 describes the transaction involved, the database operations, and what actions to take in case of an error.

See Table 6-2 for more information about system and business errors.

The following services are involved:

- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCSIm

Table 5-15 Transaction Boundaries and Recovery Details

Transaction	DB Operations	In Case of Error	Recovery
UpdateTroubleTicketOSMCFSCommsJMSConsumer picks up the message from the queue and invokes UpdateTroubleTicketSiebelCommsProvABCSImpl, which invokes the Siebel web service to update the trouble ticket.	AIA cross-references updated. Trouble Ticket updated in Siebel.	Message goes back to the originating queue AIA_UPDTTREQ_JMSQ.	Resubmit from AIA_UPDTTREQ_JMSQ.

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

Siebel CRM Interfaces

The Create and Manage Trouble Ticket from OSM business flow uses the following Siebel CRM interface:

- SWI Trouble Ticket Service

This inbound web service is invoked by the Siebel ABCS to create or update a trouble ticket in Siebel CRM. If the request is for creating a new trouble ticket, then a new trouble ticket is created and the trouble ticket number is returned. If the request is to update a specific trouble ticket, typically to close the trouble ticket, then the trouble ticket is updated.

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

Industry Oracle AIA Components

The Create and Manage Trouble Ticket from OSM business flow uses the following communications industry-specific Oracle AIA components:

- TroubleTicketEBO
- CreateTroubleTicketEBM
- CreateTroubleTicketResponseEBM
- UpdateTroubleTicketEBM

The following directories contain the industry component files:

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

  COMMS_AIA_HOME/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/TroubleTicket/V1

- Enterprise business service (EBS) WSDL files:

  COMMS_AIA_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 flow:

- AIAOrderFalloutJMSBridgeService
- CreateOrderFalloutNotificationOSMCFSCommsJMSConsumer
- CreateOrderFalloutNotificationOSMCFSCommsProvImpl
- CreateOrderFalloutNotificationOSMCFSCommsJMSProducer
- CreateTroubleTicketOSMCFSCommsJMSConsumer
- CreateTroubleTicketSiebelCommsProvABCSImpl
- CreateTroubleTicketRespOSMCFSCommsJMSProducer
- UpdateTroubleTicketOSMCFSCommsJMSConsumer
- UpdateTroubleTicketSiebelCommsProvABCSImpl
- 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 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.

CreateTroubleTicketRespOSMCFSCommsJMSProducer

The CreateTroubleTicketRespOSMCFSCommsJMSProducer is a BPEL process that does the following:

1. Receives CreateTroubleTicketResponseEBM from CreateTroubleTicketSiebelCommsProvABCSImpl.
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.

CreateTroubleTicketOSMCFSCommsJMSConsumer

The CreateTroubleTicketOSMCFSCommsJMSConsumer 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 CreateTroubleTicketSiebelCommsProvABCSImpl.

This service acts as a consumer, listening to the messages produced in the AIA_CRTTTREQ_JMSQ SAF queue.

CreateTroubleTicketSiebelCommsProvABCSImpl

The CreateTroubleTicketSiebelCommsProvABCSImpl service is an asynchronous BPEL process that does the following:

1. Receives CreateTroubleTicketEBM from CreateTroubleTicketOSMCFSCommsJMSConsumer.
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.

UpdateTroubleTicketOSMCFSCommsJMSConsumer

The UpdateTroubleTicketOSMCFSCommsJMSConsumer is a Mediator service that does the following:

1. Picks up the UpdateTroubleTicketEBM message from the AIA_UPDTTREQ_JMSQ queue.

2. Sends UpdateTroubleTicketEBM to UpdateTroubleTicketSiebelCommsProvABCSImpl.

This service acts as a consumer, listening to the messages produced in the AIA_UPDTTREQ_JMSQ queue.

UpdateTroubleTicketSiebelCommsProvABCSImpl

The UpdateTroubleTicketSiebelCommsProvABCSImpl is a BPEL process that does the following:

1. Receives UpdateTroubleTicketEBM from UpdateTroubleTicketOSMCFSCommsJMSConsumer.
2. Transforms the EBM into the TroubleTicketInsertorUpdate_Input ABM.
3. Invokes the Siebel CRM web service to update the trouble ticket.

CreateFaultNotificationLFCommsJMSConsumer

The CreateFaultNotificationLFCommsJMSConsumer 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.

Fallout-Enabled Services

The following Create and Manage Trouble Ticket from OSM flow services are fallout-enabled:

- UpdateSalesOrderSiebelCommsProvABCSImpl
- ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl
- ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer
- ProcessFOBillingAccountListRespOSMCFSCommsJMSProducer
- ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer
- TestOrderOrchestrationEBF
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsJMSConsumer_RS

  default.ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl.ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl_1_0
- OSM.ABCS.ConsumeCustomerCFO_JMSAdapter
- OSM.ABCS.ConsumeCustomerCFO_JMSAdapter_RS
- OSM.ABCS.ConsumeBillingCFO_JMSAdapter
- OSM.ABCS.ConsumeBillingCFO_JMSAdapter_RS

- OSM.ABCS.ConsumeUpdateFulfillmentOrder_JMSAdapter_RS
- OSM.ABCS.ConsumeUpdateFulfillmentOrder_JMSAdapter
- OSM.ABCS.BillingResponseConsumer
- OSM.ABCS.BillingResponseConsumer_RS
- OSM.ABCS.CustomerResponseConsumer
- OSM.ABCS.CustomerResponseConsumer_RS
- OSM.ABCS.OrderOrchestrationConsumer
- OSM.ABCS.OrderOrchestrationConsumer_RS
- default.TestOrderOrchestrationEBF.TestOrderOrchestrationEBF_1_0
- Siebel.ProcessSalesOrderFulfillmentSiebelCommsSequencer
- ProcessProvisioningOrderOSMPROVCommsJMSProducer
- ProcessFulfillmentOrderUpdateOSMCFSCommsJMSProducer
- OSMPROV
- OSM.ABCS.ConsumeProvUpdate_RS
- OSM.ABCS.ConsumeProvUpdate
- OSM.ABCS.Consume_ProcessProvOrder
- OSM.ABCS.Consume_ProcessProvOrder_RS

6

Configuring the Order to Cash Business Process

This chapter describes how to configure the Order to Cash business process.

Configuring Order Lifecycle Management

This section describes how to configure order lifecycle management in Order to Cash business process.

About Configuring 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 Application Integration Architecture Cloud Native Deployment 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 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 (DVMs) 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 6-1 lists the DVMs for the process integration for order lifecycle management.

Table 6-1 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_REVISIONPERMISIBLECODE	Revision Permissible Code
SALESORDER_SERVICEINDICATOR.	Service Indicator
SALESORDER_STRTBILLSERVICEUSAGE	Start Billing Service Usage
SALESORDER_STATUS	Status
SALESORDER_TYPECODE	Type Code
STATE	State
PROVINCE	Province
ADDRESS_COUNTRYID	Country Code
CUSTOMERPARTY_TYPECODE	Account Type Code
ITEM_BILLINGTYPECODE	Billing Type Code. Examples values include Subscription, Discount, Item, and Sponsorship.
SALESORDER_CHANGEDIND	Order Changed Indicator. Values are **True** or **False**. Used to validate the OrderChangedIndicator attribute. For example, The order management system can set this attribute to *True* if, as part of fulfillment, the order changes significantly such that Siebel CRM must make a copy of the customer order to preserve the customer intent before updating the working version of the order.
SALESORDER_ACTIONCODE	Sales Order Line Action Code
SALESORDER_REVISIONPERMISIBLECODE	Revision Permissible Code
SALESORDER_LINESTATUS	Order Line Status
DISCOUNT_METHODCODE	Discount Method Code
CURRENCY_CODE	Currency Code
PRICE_TYPE	Price Type
PRICELIST	Maps price list Row IDs from Siebel CRM to logical common price list names for BRM and Oracle Product Hub. Used to communicate price lists from order lines for products synchronized from BRM using a rate plan selector. See Working with the PRICELIST DVM for details.

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

Working with the PRICELIST DVM

You must update the PRICELIST DVM before submitting any orders so that the integration can map Siebel CRM price lists to BRM rate plans. You can migrate the PRICELIST DVM information between environments.

Updating the PRICELIST DVM

To update the PRICELIST DVM:

1. Log in to Oracle SOA Composer and open **PRICELIST.dvm**.

   For information about using Oracle SOA Composer with DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

2. Replace the placeholder domain values for the first row in **PRICELIST.dvm** as follows:
   - For SEBL_01, enter the Siebel CRM price list row ID for one of the price lists you created in Siebel CRM before synchronizing the products to BRM. You can find price list row IDs using a query in Siebel CRM.
   - For COMMON and BRM_01, enter the logical name of the price list that corresponds to the Siebel CRM price list.
3. Add and fill in new rows for all additional price lists that you created in Siebel CRM with the exception of the default price list that you entered into the **AIAConfigurationProperties.xml** file. For more information about configuring Siebel CRM for integrated product lifecycle management, see *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide*
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 6-2 lists the order lifecycle management cross-references.

Table 6-2 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	SalesOrderLine/Identification	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 (Populated only when service-level balance groups are enabled)	InstalledProductIdentification (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. For nested service bundles, the integration populates only the values for the parent service bundle. 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 (Entry for default account-level balance group)	CustomerPartyAccountIdentification	ServiceAccountId/ 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 6-2 (Cont.) Order Lifecycle Management Cross-References

Cross-Reference Table Name	COMMON	SEBL_01	BRM_01	Description
CUSTOMERPARTY_ADDRESSID	CustomerPartyAccountAddressIdentification	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). The integration uses this XREF to determine if a bundled promotion is associated with a promotion group membership. 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. 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 **/billinfo**	The common ID of the billing profile for the owner account on the sales order is mapped to the corresponding **/billinfo** POID. When creating new collections sharing groups, the integration checks if the owner account's **/billinfo** 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 **/billinfo**	The common ID of the billing profile for the member account on the sales order is mapped to the corresponding **/billinfo** 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 About Order Fallout Management for more information about order fallout.

Table 6-3 lists the error messages that are issued when order billing integration is called in billing initiation mode.

Table 6-3 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 *Subscription* or *Discount*.
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 *Item*.
AIA_ERR_AIACOMOMPI_0003	Purchased promotion instance does not exist for a promotion that was previously purchased. A data upgrade script was not run.	ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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.	ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 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 **COMMS_AIA_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 6-4 shows the system-level configuration properties for order lifecycle management.

Table 6-4 System Configuration Properties for Order Lifecycle Management

Property Name	Value/Default Values	Description
O2C.AccountLevelBalanceGroup	TRUE/FALSE Default value: **FALSE**	A value of **FALSE** 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 6-5 shows the properties for the UpdateSalesOrderSiebelCommsProvABCSImpl service.

Table 6-5 UpdateSalesOrderSiebelCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	URL for Siebel CRM Instance web service for "Order spcLine spcItem spcUpdate spc-_spcComplex" web service.
Routing.SWI_spcOrder_spcUpsert.RouteToCAVS	true/false. Default = false.	Controls whether UpdateSalesOrderSiebelCommsProvABCSImpl routes messages to the CAVS or to the Siebel CRM system.
Routing.SWI_spcOrder_spcUpsert.SEBL_01.EndpointURI	Target Endpoint URL for the Siebel CRM upsert web service. example: http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1Status OpenFixedClosed	Target Endpoint URL for the Siebel CRM upsert web service.
ABCSExtension.PreXformEBMtoABM	true/false. Default = false.	Whether there is any extension in the ABCS before transformation of EBM to ABM.
ABCSExtension.PreInvokeABS	true/false. Default = false.	Indicates whether there is any extension in the ABCS before invoking application business service.

Table 6-6 shows the properties for the ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl service.

Table 6-6 ProcessSalesOrderFulfillmentSiebelCommsReqABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	Possible value: A Siebel CRM system instance code. Default value: **SEBL_01**	Default Siebel CRM system instance code (defined in BSR). This is used only if the Siebel CRM Order message does not contain the EnterpriseServerName.
ABCSExtension.PreXformABMtoEBM	Possible values: **true** or **false** Default value: **false**	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: **true** or **false** Default value: **false**	Indicates whether there is any extension in the ABCS before invoking application business service.
ABCSExtension.PostInvokeEBS	Possible values: **true** or **false** Default value: **false**	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. Default value: **http://<soa_server>:<soa port>/soa-infra/services/default/ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer/client**	This property is used to dynamically invoke any web service from this service. This holds the address of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.PortType	Possible value: PortType of the web service that must be invoked. Default value: **ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer**	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. Default value: **http://xmlns.oracle.com/ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer}ProcessSalesOrderFulfillmentOSMCFSCommsJMSProducer**	This value should be in consistent with EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address.

Configuration Properties for BRM Services

Table 6-7 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess service.

Table 6-7 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.

Table 6-7 (Cont.) ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Properties

Property Name	Value/Default Values	Description
BRM_01.FutureTimeThresholdForBillingDates	8640	This property is used for future date validation in Billing Initiation. It is set to a default value of 8640 hours (360 days). This property is billing-instance-specific and must be set for any instance that the order must be sent for billing integration. See Using Single-Phase Billing or Two-Phase Billing for more information on how this property is used.
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PostprocessAddPCM_OP_SUBSSCRIPTION_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 6-8 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess service.

Table 6-8 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.

Table 6-8 (Cont.) ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess Properties

Property Name	Value/Default Values	Description
ABCSExtention.PostprocessPCM_OP_SUBSCRIPTION_CANCEL_DISCOUNTABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_CANCEL_PRODUCTABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT.
ABCSExtention.PostProcessPCM_OP_SUBSCRIPTION_CANCEL_PRODUCTABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT.
ABCSExtension.PreProcessPCM_OP_CUST_SET_STATUSABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_SET_STATUS.
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 6-9 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsMoveAddSubProcess service.

Table 6-9 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 6-10 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl service.

Table 6-10 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	BRM_01	Default target billing system instance code (defined in BSR). This is used only if the request message does not contain the target information.
Routing.BRMSUBSCRIPTIONService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMSUBSCRIPTIONService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMSUBSCRIPTIONService.BRM_02.EndpointURI	End point for BRM Adapter. Example: eis/BRM2	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMSUBSCRIPTIONService.BRM_03.EndpointURI	End point for BRM Adapter. Example: eis/BRM3	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMCUSTService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMCUSTService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMCUSTService.BRM_02.EndpointURI	End point for BRM Adapter. Example: eis/BRM2	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMCUSTService.BRM_03.EndpointURI	End point for BRM Adapter. Example: eis/BRM3	End point for BRM Adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMBALService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMBALService_ptt.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMBALService_ptt.BRM_02.EndpointURI	End point for BRM Adapter. Example: eis/BRM2	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMBALService_ptt.BRM_03.EndpointURI	End point for BRM Adapter. Example: eis/BRM3	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMARService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.
Routing.BRMARService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
Routing.BRMBASEService.RouteToCAVS	true/false. Default = false	CAVS simulator to be enabled or disabled for this partner link.

Table 6-10 (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Routing.BRMBASEService.BRM_01.EndpointURI	End point for BRM Adapter. Example: eis/BRM	End point for BRM adapter. Example: Update with CAVS endpoint URL to route to CAVS along with changing the above property to "**true**"
ABCSExtension.PreInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension before transformation of EBM to ABM.
ABCSExtension.PostInvokeEBM	true/false. Default = false	To indicate whether the ABCS has any extension after transformation of EBM to ABM.
ABCSExtension.PreProcessPCM_OP_CUST_CREATE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_CREATE_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_CREATE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_CREATE_PROFILE.
ABCSExtension.PreProcessPCM_OP_CUST_MODIFY_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_MODIFY_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_MODIFY_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_MODIFY_PROFILE.
ABCSExtension.PreProcessPCM_OP_CUST_DELETE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension before calling BRM opcode PCM_OP_CUST_DELETE_PROFILE.
ABCSExtension.PostProcessPCM_OP_CUST_DELETE_PROFILEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_CUST_DELETE_PROFILE.
ABCSExtension.PreProcessPCM_OP_SUBSCRIPTION_SET_BUNDLEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_BUNDLEABM.
ABCSExtension.PostProcessPCM_OP_SUBSCRIPTION_SET_BUNDLEABM	true/false. Default = false	To indicate whether the ABCS has any extension after calling BRM opcode PCM_OP_SUBSCRIPTION_SET_BUNDLEABM.
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.Address	Address of the web service that must be invoked. Example (Default): `{http://host:port/soa-infra/services/default/ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer/client`	This property is used to dynamically invoke any web service from this service. This holds the Address of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.PortType	PortType of the web service that must be invoked. Example (Default): ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	This property is used to dynamically invoke any web service from this service. This holds the PortType of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.

Table 6-10 (Cont.) ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.CommunicationsBillingEBSV1.ProcessFulfillmentOrderBillingResponse.ServiceName	ServiceName of the web service that must be invoked. Example (Default): `{http://xmlns.oracle.com/ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer}`ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	This property is used to dynamically invoke any web service from this service. This holds the ServiceName of the web service that must be invoked dynamically. To invoke CAVS/EBS or any other web service this property must be updated accordingly.

Table 6-11 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess service.

Table 6-11 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.

Table 6-11 (Cont.) ProcessFulfillmentOrderBillingBRMCommsSuspendResumeSubProcess Properties

Property Name	Value/Default Values	Description
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.
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 6-12 shows the properties for the ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess service.

Table 6-12 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.

Table 6-12 (Cont.) ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess Properties

Property Name	Value/Default Values	Description
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.
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.

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. 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.

Configuring Customer Management

This section describes how to configure customer management in the Order to Cash business process.

About Configuring Customer Management

Configuring customer management includes:

- Working with domain value maps (DVMs) and cross-references.
- Handling error notifications.
- Setting Oracle AIA configuration properties.

This section assumes that you have already installed Oracle Communications Billing and Revenue Management (BRM) and Siebel CRM. See *Oracle Application Integration Architecture Compatibility Matrix* 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 6-13 lists the DVMs for the process integration for customer management.

Table 6-13 Customer Management DVMs

DVM	Columns	Description
CUSTOMERPARTY_ACCOUNTTYPECODE.dvm	SEBL_01, COMMON, BRM_01	Used to get the type of the account, such as **Business** or **Residential**.
PROVINCE.dvm	SEBL_01, COMMON, BRM_01	Province name.
STATE.dvm	SEBL_01, COMMON, BRM_01	State name.
ADDRESS_COUNTRYID.dvm	SEBL_01, COMMON, BRM_01	Country codes.
ADDRESS_COUNTRYSUBDIVID.dvm	SEBL_01, COMMON, BRM_01	State codes.
CONTACT_SALUTATION.dvm	SEBL_01, COMMON, BRM_01	Salutation (such Mr., Mrs.). In BRM, Salutation is not a language-independent code. If BRM requires salutations in a language other than English, then you must update the DVM with the appropriate BRM values.
CURRENCY_CODE.dvm	SEBL_01, COMMON, BRM_01	Currency codes.

Table 6-13 (Cont.) Customer Management DVMs

DVM	Columns	Description
CUSTOMERPARTY_BILLPROFILE_BILLTYPECODE.dvm	SEBL_01, COMMON, BRM_01	Bill type (**summary** and **detailed**).
CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE.dvm	SEBL_01, COMMON, BRM_01	Billing frequency (**monthly, yearly, quarterly**, and so on).
CUSTOMERPARTY_PAYPROFILE_BANKACCOUNTTYPE.dvm	SEBL_01, COMMON, BRM_01	Bank account type (**checking, savings**, and so on).
CUSTOMERPARTY_PAY PROFILE CREDIT_CARDTYPE.dvm	SEBL_01, COMMON	Credit Card type **(Visa, Mastercard**, and so on).
CUSTOMERPARTY_PAYPROFILE_DELIVERYPREF.dvm	COMMON, BRM_01	Bill media/delivery preference (**Email** or **Mail)**.
CUSTOMERPARTY_PAYPROFILE_PAYMETHODCODE.dvm	SEBL_01, COMMON, BRM_01	Payment profile payment method types (**credit card, direct debit**, and **invoice/bill me**).
CUSTOMERPARTY_PAYPROFILE_PAYTERMCODE.dvm	COMMON, BRM_01	Payment term codes.
CUSTOMERPARTY_STATUSCODE.dvm	SEBL_01, COMMON, BRM_01	Account status codes.
PHONENUMBER_TYPE.dvm	SEBL_01, COMMON, BRM_01	Phone number type codes (**home, work, mobile, fax**, and so on).

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 6-14 lists the customer management cross-references.

Table 6-14 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
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) Pay-info POID (if this is the pay-info address)
CUSTOMERPARTY_CONTACTID	Contact ID	Contact ID	Account POID (if this is the main contact) Pay-info POID (if this is the pay-info contact)

Table 6-14 (Cont.) Customer Management Cross-References

Cross-Reference Table Name	COMMON	SEBL_01	BRM_01
CUSTOMERPARTY_DEFAULTBALANCEGROUPID	Account ID	Not applicable	Balance Group POID
CUSTOMERPARTY_PARTYID	Customer Party ID	Customer Party ID	Not applicable
CUSTOMERPARTY_PARTYLOCATIONID.xref	Customer Party Location ID	Customer Party Location ID	Not applicable
CUSTOMERPARTY_LOCATIONREFID	Customer Party Location Reference ID	Customer Party Location Reference ID	Not applicable
CUSTOMERPARTY_ACCOUNT_PHONECOMMID	Customer Account's phone number	Customer Account's phone number	Not applicable
CUSTOMERPARTY_ACCOUNT_FAXCOMMID	Customer Account's fax number	Customer Account's fax number	Not applicable
CUSTOMERPARTY_ACCOUNT_WEBCOMMID	Customer Account's email address	Customer Account's email address	Not applicable
CUSTOMERPARTY_CONTACT_PHONECOMMID	Customer Contact's phone number	Customer Contact's phone number	Not applicable
CUSTOMERPARTY_CONTACT_FAXCOMMID	Customer Contact's fax number	Customer Contact's fax number	Not applicable
CUSTOMERPARTY_CONTACT_EMAILCOMMID	Customer Contact's email address	Customer Contact's 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 6-15 lists the errors that are caused by the process integration for customer management services.

Table 6-15 Errors Caused by Customer Management Services

Service Name	Error Code	Possible Cause
SyncCustomerPartyListBRMCommsProvABCSImpl	AIA_ERR_AIACOMCMPI_0004	Subordinate account cannot have multiple parent accounts.
SyncCustomerPartyListBRMCommsProvABCSImpl	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.
SyncCustomerPartyListBRMCommsProvABCSImpl	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.

Table 6-15 (Cont.) Errors Caused by Customer Management Services

Service Name	Error Code	Possible Cause
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 About 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/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 6-16 shows the system-level properties for the process integration for customer management.

Table 6-16 Customer Management System-level Properties

Property Name	Value/Default Value	Description
O2C.LegalGroup	TRUE/FALSE. Default: FALSE	Determines whether the integration creates collections sharing groups in BRM. If set to **TRUE**, the integration creates collections sharing groups when the owner account and billing account on an order line are different. If set to **FALSE**, the integration ignores the owner account and does not create collections sharing groups.

Configuration Properties for BRM Services

Table 6-17 shows the properties for the ProcessCollectionSharingBRMCommsProvABCSImpl service.

Table 6-17 ProcessCollectionSharingBRMCommsProvABCSImpl 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	**true/false** Default: **false**	Specifies whether the end point routes to CAVS.
Routing.BRMCOLLECTIONSService.BRM_01.EndpointURI	Default: eis/BRM	Specifies the end point URL for the CAVS.
ABCSExtension.ABCSExtension.PreXformABMtoEBM	true/false Default: false	Indicates whether the ABCS has been extended before the EBM to ABM transformation.
ABCSExtension.PostXformEBMtoABM	true/false Default: false	Indicates whether the ABCS has been extended after the EBM to ABM transformation.

Table 6-18 shows the properties for the SyncCustomerPartyListBRMCommsProvABCSImpl service.

Table 6-18 SyncCustomerPartyListBRMCommsProvABCSImpl 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.

Table 6-18 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties

Property Name	Value/Default Value	Description
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.
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. Default: false	Indicates whether the integration synchronizes account status. When set to **true**, 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.
Routing.BRMCUSTService.RouteToCAVS	true/false. Default: false	Specifies whether the end point routes to CAVS for the CUST opcodes.
AccountLevelBalanceGroupName	Account Level Balance Group	Names the default balance group created in BRM when an account is created.
ABCSExtension.PreXFormEBMtoABM	true/false. Default: false	Indicates whether the ABCS has been extended before the EBM to ABM transformation.
ABCSExtension.PreInvokePCM_OP_BILL_GROUP_GET_PARENTABM	true/false. 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. 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. 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. 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. 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. 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. Default: false	Indicates whether the ABCS has been extended before the invocation of the opcode in the property name.
ABCSExtension.PostInvokePCM_OP_CUSTCARE_MOVE_ACCTABM	true/false. Default: false	Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.

Table 6-18 (Cont.) SyncCustomerPartyListBRMCommsProvABCSImpl Properties

Property Name	Value/Default Value	Description
ABCSExtension.PreInvokePCM_OP_CUST_UPDATE_CUSTOMERABM	true/false. 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. 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. 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. Default: false	Indicates whether the ABCS has been extended after the invocation of the opcode in the property name.
ABCSExtension.PostXFormABMtoEBM	true/false. Default: false	Indicates whether the ABCS has been extended after the ABM to EBM transformation.
Routing.BRMCUSTService_ptt.RouteToCAVS	true/false. Default: false	Specifies whether to route to CAVS for the CUST opcodes.
Routing.BRMCUSTCAREService.RouteToCAVS	true/false. Default: false	Specifies whether to route to CAVS for the CUSTCARE opcodes.
Routing.BRMBILLService.RouteToCAVS	true/false. Default: false	Specifies whether to route to CAVS for BILL opcodes.
Routing.BRMBASEService.RouteToCAVS	true/false. Default: false	Specifies whether to route to CAVS for BASE opcodes.
Routing.BRMTXNService.RouteToCAVS	true/false. Default: false	Specifies whether to route to CAVS for TXN opcodes.

Configuration Properties for Siebel CRM Services

Table 6-19 shows the properties for the SyncAccountSiebelReqABCSImpl service.

Table 6-19 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. Default: false	Indicates whether the ABCS has been extended before the ABM to EBM transformation.
ABCSExtension.PreInvokeEBSEBM	true/false. Default: false.	Indicates whether the ABCS has been extended before the EBS invocation.
Routing.SWICustomerParty.SEBL_01.EndpointURI	http://*Siebel_host*:*port*/eai_*Siebel_language*/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	Endpoint URI of the SEBL_01 Siebel instance.

Table 6-19 (Cont.) SyncAccountSiebelReqABCSImpl Properties

Property Name	Value/Default Values	Description
Routing.SWICustomerParty.RouteToCAVS	true/false. Default: false.	Specifies whether the end point routes to CAVS.
Routing.SWICustomerParty.CAVS.EndpointURI	http://*SOA_host*:*port*/AIAValidationSystemServlet/syncresponsesimulator	Specifies the end point URL for the CAVS service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	NA
RoutingCustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${fp.server.soaserverhostname}:{fp.server.soaserverport}/AIAValidationSystemServlet/asyncrequestrecipient	Specifies whether the end point routes to CAVS.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	true/false. Default: false.	Specifies whether the end point routes to CAVS.
Routing.TransformAppContextSiebelService.RouteToCAVS	true/false. Default: false.	Specifies whether the end point routes to CAVS.
Routing.TransformAppContextSiebelService.CAVS.EndpointURI	http://${fp.server.soaserverhostname}:{fp.server.soaserverport}/AIAValidationSystemServlet/asyncrequestrecipient	Specifies whether the end point routes to CAVS.
Account.ProcessUpdateEventsOnly	true/false. Default: true.	To optimize the integration flow, leave this property set to **true**. 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. Setting this property to **false** results in a less optimized flow, and the default integration behavior remains the same. See *Oracle Application Integration Architecture Pre-Built Integrations Functional Interoperability Configuration Guide* for more information.
Contact.QueryAllEntities	true/false. Default: false.	NA
Address.QueryAllEntities	true/false. Default: false.	NA

Table 6-20 shows the properties for the ProcessAccountHierarchyListSiebelCommsProvABCSImpl service.

Table 6-20 ProcessAccountHierarchyListSiebelCommsProvABCSImpl 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 Default: false	Indicates whether the ABCS has been extended before the EBM to ABM transformation.
ABCSExtension.PostXformEBMtoABM	true/false Default: false	Indicates whether the ABCS has been extended after the EBM to ABM transformation.
Routing.SWI_spcAccount_spcHierarchy.RouteToCAVS	true/false Default: false	Specifies whether the end point routes to CAVS.
Routing.SWI_spcAccount_spcHierarchy.CAVS.EndpointURI	http://*SOA_server*:*port*/AIAValidationSystemServlet/syncresponsesimulator.	Specifies the CAVS endpoint URI for the Siebel CRM adapter.
Routing.SWI_spcAccount_spcHierarchy.SEBL_01.EndpointURI	http://*Siebel_server*:*port*/eai_*Siebel_language*/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=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.PreInvokeABS	true/false Default: false	Indicates whether the ABCS invokes the extension service before invoking the application.
ABCSExtension.PostInvokeABS	true/false Default: false	Indicates whether the ABCS invokes the extension service after invoking the application.

Table 6-21 shows the properties for the QueryCustomerPartyListSiebelProvABCSImpleV2 service.

Table 6-21 QueryCustomerPartyListSiebelProvABCSImpleV2 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 Default: false	Indicates whether the ABCS has been extended before the EBM to ABM transformation.
ABCSExtension.PostXformABMtoEBMEBM	true/false Default: false	Indicates whether the ABCS has been extended after the EBM to ABM transformation.
Routing.SWI_spcCustomer_spcParty_spcService.RouteToCAVS	true/false Default: false	Specifies whether the end point routes to CAVS.

Table 6-21 (Cont.) QueryCustomerPartyListSiebelProvABCSImpleV2 Properties

Property Name	Value/Default Value	Description
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?SWEExtSource=SecureWebService&SWEExtCmd=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 Default: false	Indicates whether the ABCS invokes the extension service before invoking the Siebel CRM web service.
ABCSExtension.PostInvokeSWICustServiceABM	true/false Default: false	Indicates whether the ABCS invokes the extension service after invoking the Siebel CRM web service.

Configuring Order Fallout Management

This section describes how to configure order fallout management in the Order to Cash business process.

About Configuring Order Fallout Management

Configuring 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 section 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

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 6-22 lists the DVMs for the process integration for order fallout.

Table 6-22 Order Fallout Management DMVs

DVM	Description
TROUBLETICKET_AREA	DVM to map the Area of the trouble ticket. SEBL_01 column maps to the Area element in Siebel CRM. COMMON column points to the SystemCode column of the corresponding system in the AIA Systems page.
TROUBLETICKET_SUBAREA	DVM to map the SubArea of the trouble ticket. SEBL_01 column maps to the Sub-Area element in Siebel CRM. COMMON column points to the appropriate FailureSubSystemCode or the faulting service.
TROUBLETICKET_STATUS	DVM to map the status of the trouble ticket. SEBL_01 column maps to the Status element in Siebel CRM. COMMON column maps to the appropriate status in Oracle AIA.

Table 6-22 (Cont.) Order Fallout Management DMVs

DVM	Description
TROUBLETICKET_SEVERITY	DVM to map the severity of the trouble ticket. SEBL_01 column maps to the Severity element in Siebel CRM. COMMON column maps to the appropriate severity (1-5) in Oracle AIA.
TROUBLETICKET_PRIORITY	DVM to map the recovery priority of the trouble ticket. SEBL_01 column maps to the Priority element in Siebel CRM. COMMON column maps to the appropriate priority (1-4) in Oracle AIA.

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 6-23 lists the order fallout cross-reference.

Table 6-23 Order Fallout Management Cross-Reference

Cross-Reference Table Name	Column Name COMMON	Column Value SEBL_01	Description
TROUBLETICKET_TROUBLETICKETID	CreateTroubleTicketResponseEBM/DataArea/CreateTroubleTicketResponse/Identification/BusinessComponentID stores this value. A randomly generated ID is used as the COMMON value for the trouble ticket and referenced with the Siebel value.	The row ID for the trouble ticket created in Siebel, which is returned in the ListOfSWITroubleTicketIO/TroubleTicket/Id element of the response of the web service, is cross-referenced.	The trouble ticket ID returned by the Siebel web service is cross-referenced to the BusinessComponentID of the TroubleTicket Response enterprise business message (EBM). The idea is to use this cross-referenced value for making any updates to this trouble ticket. So this cross-referencing is done only when the response is sought from the process CreateTroubleTicketSiebelCommsProvABCSImpl.

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 6-24 lists the localized custom errors that are caused by the order fallout management services for data insufficiency for creating a trouble ticket.

Table 6-24 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 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 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 business process option and the Oracle Communications Order to Cash OSM business process options are installed, the seeded value for ERROR_TYPE is AIA_ORDERFALLOUT_CFS. If the Oracle Communications Order to Cash: Siebel CRM business process option is installed alone (without the Oracle Communications Order to Cash OSM business process 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 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 6-25 is an example entry for the ProcessFulfillmentOrderBillingBRMCommsAddSubProcess order service.

Table 6-25 Example Entry for ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Order Service

Error Code	Service Name	Error Type	Error Extn Handler
--	ProcessFullmentOrderBillingBRMCommsAddSubProcess	AIA_EH_DEFAULT	AIACOM_OFM_EXT

Table 6-25 (Cont.) Example Entry for ProcessFulfillmentOrderBillingBRMCommsAddSubProcess Order Service

Error Code	Service Name	Error Type	Error Extn Handler
{http://schemas.oracle.com/bpel/extension}bindingFault	ProcessFullmentOrderBillingBRMCommsAddSubProcess	AIA_EH_DEFAULT,AIA_ORDERFALLOUT_CFS	AIACOM_OFM_EXT

If additional error codes are also classified as business errors, you can add new entries into the AIA_ERROR_NOTIFICATIONS table with the appropriate Error Code value.

> **Note**
>
> The Error Extn Handler value for all order service entries must be AIACOM_OFM_EXT. This is required so that the correct information is in the fallout and the standard error notification.

You can perform a bulk update or 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 Order Fallout Services

Configure the properties described in this section in the **COMMS_AIA_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 6-26 shows the properties for the CreateTroubleTicketAIACommsReqImpl service.

Table 6-26 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.

Table 6-26 (Cont.) CreateTroubleTicketAIACommsReqImpl Properties

Property Name	Value/Default Values	Description
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.PortType	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCSImpl/V1}CreateTroubleTicketSiebelCommsProvABCSImplService	Port Type of the provider ABCS/EBS service of the CreateTroubleTicket operation.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.Address	http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketSiebelCommsProvABCSImpl/CreateTroubleTicketSiebelCommsProvABCSImpl	Address of the provider ABCS/EBS service CreateTroubleTicket operation.
EBSOverride.TroubleTicketEBS.CreateTroubleTicket.ServiceName	{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/CreateTroubleTicketSiebelCommsProvABCSImpl/V1}CreateTroubleTicketSiebelCommsProvABCSImplService	Service Name of the provider ABCS/EBS service of the CreateTroubleTicket operation.
TroubleTicket.DefaultSeverity	Any number from 1 to 5. Default = 2.	If the fault message does not have any severity associated with it, the default severity is assigned to the fault message and reflects the same in the trouble ticket.
TroubleTicket.DefaultPriority	Any number from 1 to 4. Default = 2.	This service assigns the recovery priority for the trouble ticket by default to the value specified in this configuration property.

Table 6-27 shows the properties for the CreateTroubleTicketSiebelCommsProvABCSImpl service name.

Table 6-27 CreateTroubleTicketSiebelCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property. TroubleTicket.UseDefaultInstance is set to true.
ABCSExtension.PreXformEBMtoABMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).
ABCSExtension.PostXformABMtoEBMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation).
ABCSExtension.PreInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).

Table 6-27 (Cont.) CreateTroubleTicketSiebelCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
ABCSExtension.PostInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PostInvoke Application).
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.PortType	{http://xmlns.oracle.com/Industry/Comms/CreateTroubleTicketRespOSMCFSCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCommsJMSProducer	Port Type of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.Address	http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/CreateTroubleTicketRespOSMCFSCommsJMSProducer	Address of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.
EBSOverride.TroubleTicketEBSResponse.CreateTroubleTicketResponse.ServiceName	{http://xmlns.oracle.com/Industry/Comms/CreateTroubleTicketRespOSMCFSCommsJMSProducer/V1}CreateTroubleTicketRespOSMCFSCommsJMSProducer	Service Name of the provider ABCS/EBS service of the CreateTroubleTicketResponse operation.
Routing.SWI_spcTrouble_spcTicket_spcService.RouteToCAVS	true/false Default: false	Indicates whether the Partner link SWI_spcTrouble_spcTicket_spcService should be routed to CAVS or the actual application.
Routing.SWI_spcTrouble_spcTicket_spcService.SEBL_01.EndpointURI	Endpoint URI of the SEBL_01 Siebel instance	Endpoint URI of the SEBL_01 Siebel instance or CAVS simulator (if RouteToCAVS is true).
TroubleTicket.GenerateTroubleTicketResponse	true/false Default: false	CreateTroubleTicketSiebelCommsProvABCSImpl creates a trouble ticket response message (creates a cross-reference for the trouble ticket ID with the Siebel ID) and invokes the CommunicationsTroubleTicketResponseEBSV1 if this property is set to true or if the response code attribute is not null. Otherwise, this service acts only as a fire-and-forget flow and ignores the response.
TroubleTicket.UseDefaultInstance	true/false Default: false	If set to true, overwrites the target Siebel instance to the default instance indicated by the property Default.SystemID. Gives the user an option to create a trouble ticket in a Siebel instance different from the one where the order was placed.
TroubleTicket.SR_TYPE	Order Failure	SR_TYPE identifies that the trouble ticket is for Order Failure. Siebel web service expects this value to be Order Failure for Order Failure Trouble Tickets.

Table 6-28 shows the properties for the UpdateTroubleTicketSiebelCommsProvABCSImpl service name.

Table 6-28 UpdateTroubleTicketSiebelCommsProvABCSImpl Properties

Property Name	Value/Default Values	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR). Used when the target system cannot be identified from the request message or if the configuration property TroubleTicket.UseDefaultInstance is set to true.
ABCSExtension.PreXformEBMtoABMTroubleTicketEBM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation).
ABCSExtension.PreInvokeABSSWITroubleTicketIOABM	true/false Default: false	Value determines whether the ABCS should invoke the Extension service (PreInvoke Application).
Routing.SWI_spcTrouble_spcTicket_spcService.RouteToCAVS	true/false Default: false	Indicates whether the Partner link SWI_spcTrouble_spcTicket_spcService should be routed to CAVS or the actual application.
Routing.SWI_spcTrouble_spcTicket_spcService.SEBL_01.EndpointURI	Endpoint URI of the SEBL_01 Siebel instance	Endpoint URI of the SEBL_01 Siebel instance or CAVS simulator (if RouteToCAVS is true).

Table 6-29 shows the properties for the CreateOrderFalloutNotificationOSMCFSCommsProvImpl service.

Table 6-29 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.EndpointURI	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 enqueuing or producing service that enqueues or produces the message in the OSM JMS Queue.
Fault.DefaultSeverity	Any number from 1 to 5 Default - 2	If the fault message has no severity associated with it, assigns the default severity to the fault message and reflects the same in the trouble ticket.

The following fields extract the localized values using the **aia:getAIALocalizedString** xpath function:

```
EBM Field Name: DataArea / CreateTroubleTicket / Description
Siebel Field Name: Description
ResourceBundle - oracle.apps.aia.core.i18n.AIAListResourceBundle
ResourceBundle Key - TROUBLETICKET_DESCRIPTION
Resource Bundle Value: SalesOrder- {OrderNumber} # {OrderRevision}for Account
{AccountName} failed at {Timestamp}
--
EBM Field Name: EBMHeader/BusinessScope/ID
Siebel Field Name: Abstract
ResourceBundle - oracle.apps.aia.core.i18n.AIAListResourceBundle
ResourceBundle Key - TROUBLETICKET_ABSTRACT
```

Resource Bundle Value: **[{Timestamp}] Trouble Ticket for (Sales)Order - {OrderNumber} # {OrderRevision}**

7

Using the Order to Cash Reference Library

This chapter provides information about some sample order to cash features that are pre-configured in the Digital Business Experience Reference Solution (the reference solution) and will be available in your Digital Business Experience environment after installing the reference solution. These sample features help you get familiarized with the Digital Business Experience applications and business processes.

The following are some order to cash features that are pre-configured in the reference solution:

- First-Time Purchase (New Subscriber, New Order)
- Change Order
- Modify Order
- Suspend Services
- Resume Services
- Disconnect Services

Before you begin:

- Learn about the reference solution. See About the Reference Solution in *Oracle Communications Digital Business Experience Concepts* for more details.
- Learn about the reference product models and seed data available in the reference solution. See Reference Product Models and Seed Data in *Oracle Communications Digital Business Experience Concept to Market Guide* for more details.
- Download and deploy the reference solution. See Deploying the Reference Solution Package in *Oracle Communications Digital Business Experience Solution Deployment Guide* for more details.

The following sections provide more details about how to use each sample pre-configured order to cash feature, and how to create those orders.

Creating a First-Time Purchase Order

Amy Watkins is a new subscriber who wants to subscribe to a voice service from Supremo Telecom Ltd. She calls the service center to speak to a CSR to request the subscription. The CSR verifies Amy Watkins is not an existing subscriber and creates a new account for Amy Watkins based on the information provided. The CSR captures details such as:

- Name
- Address
- Phone numbers (if any)
- Email address
- Billing details

After creating the new account, the CSR creates a new order and adds the Supremo 5G Unlimited promotion to which Amy Watkins wants to subscribe to the order, and submits the order. The order flows through the Digital Business Experience stack to completion. After order

completion, bills are generated in BRM. The CSR will access the Account view in Siebel CRM to view the installed assets and bill details.

To create a first-time purchase order for a prospect, there are various tasks that need to be performed as follows.

Creating an Account in Siebel CRM

To create an account in Siebel CRM, sign in to Siebel CRM and create an account. See Creating an Account in *Siebel CRM Administration Guide* for instructions on creating an account.

1. On the **Accounts List** page, provide the values for the following fields:
   - Account Name
   - Account Type
   - Address
   - Account Class
   - Currency
   - Price List
2. From the **Account Summary** drop-down menu, select **Contacts**, and provide the values for the following fields to create a contact:
   - First Name
   - Last Name
   - Mr/Ms
3. On the **Account Summary** page, **Billing Profile** section, provide the values for the following fields to create a billing profile:
   - Billing Name.
   - Payment Method: Based on the Payment Method, provide the details, such as Bill Type, Frequency, Bill Media, Credit Card Number, Expiration Month, and Expiration Year.
   - Select the **Primary** check box.

Creating an Order in Siebel CRM

This procedure assumes that you have created an account in Siebel CRM and are currently on the Account Summary page for the created account.

To create an order in Siebel CRM, sign in to Siebel CRM, and create an order. See Creating an Order in *Siebel CRM Order Management Guide* for instructions on creating an order.

1. On the **Accounts Summary** page, **Orders** section, click the **Order Number** link and provide the values for the following fields:
   - Due Date (select the Due Date as Order Date to complete the order instantly).
   - In the **Line Items** section, from the **Products** menu, select the required **Promotion**.
   - Check the Eligibility and Compatibility rules.
   - Add Service IDs for each commercial bundle, service bundle, and simple service bundle.

> **Note**
>
> - Orders can be placed using different types of products, such as **Commercial Bundles**, **Simple Service Bundles**, or **Promotions**. In the following steps, you add a promotion to the product lines to create and place an order. For more information on these terms, refer to *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.
> - Ensure that all product data used in the examples in this document exist in your environment after installing the reference solution.
> - If the **Eligibility Status** column displays **Y**, you can create the order. Otherwise, cancel the order and create a new order with a different promotion eligible for subscriber's address.
> - In the **Line Items** section, verify the Compatibility rules, discounts, or time-based offerings applied to the promotion.

2. Click **Submit**.
   After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances:

1. Sign to the AIA EM Console, **soa-infra** page.
2. Verify if the **Flow State** column for all the respective instances is **Completed**.

To verify the Order status in Siebel CRM:

1. Navigate back to the **Account Summary** page, **Order** section.
2. In the **Line Items** section, verify if the **Status** column for all line items is **Complete**.
3. Verify if the **Status** field in the Order header displays **Complete**.

To verify the Order status in OSM:

1. Sign in to OSM.
2. In the **Order Number** field, provide the order number, and click **Search**.
3. Verify if the **Order Status** column for COM and SOM displays **Complete**.

Verifying a Completed Order in BRM

To verify a completed order in BRM:

1. Sign in to the BRM Billing Care application by entering the provided user Id and password.
2. Click **Search** and search for the account.
3. On the **Results** page, select the account and click Open.
4. On the **Home** tab, click **Assets** to view all the installed assets of the promotion.
5. On the **Home** tab, click **Offers** to view all offers, discounts, and promotions.
6. On the **Bills** tab, verify the **Account Charges** and **Service Charges** sections.
7. On the **Home** tab, select **Account History**.
8. Verify the order details.
9. Click **Bill Now** to generate the bill for the account.

The bill details appear.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify all the installed services and assets of the order.

Creating a Change Package (Upgrade/Downgrade)

Upgrade request: Amy Watkins is an existing subscriber who subscribed to Supremo Triple Play Unlimited promotion. She liked the services and wants to upgrade to the Supremo Connect Platinum promotion. She calls the service center to speak to a CSR to request this change. The CSR creates a new order and customizes it to reflect Amy's request, and submits the order.

To create a change package order for an existing subscriber, there are various tasks that need to be performed as follows.

Creating a Change Package Order in Siebel CRM

This procedure assumes that the order for the account was previously placed, submitted, and completed; therefore, you can scroll down to the **Installed Assets** section in Siebel CRM and view the products the account has subscribed to.

To create a change package order and verify the completion of change package order in Siebel, sign in to Siebel CRM and query the order details. See Querying an Order in *Siebel CRM Order Management Guide* for instructions on querying an order:

1. To upgrade a promotion, from the **Install Assets** section, select the **Promotion**.
2. Click the **Settings** icon for the selected promotion, and select **Upgrade**.
   The **Promotion Upgrades** window appears.
3. Select the required promotion.

> **Note**
>
> If you select to downgrade your promotion, there will be penalty charges. For upgrade promotion, there are no penalty charges.

4. Click **OK**.
   For **Residential** account type, the **Quotes** page is bypassed and you are taken directly to the **Sales Order** page, where a new order is auto-generated.
5. Enter the following data:
   - **Order#** is pre-populated, but you can edit it per business requirements.
   - For demo purposes, set the **Due date** to the order creation date. The date also needs to be updated for all Line Items under the **Due** column.
   - In **Price List**, search for and select the appropriate price list. The change order is now created.
6. Click **Submit**.

After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances, see Verifying the Order Instances.

To verify the order status in Siebel CRM, see Verifying the Order Status in Siebel CRM.

To verify the order status in OSM, see Verifying the Order Status in OSM.

Verifying a Completed Order in BRM

To verify the change package order results in BRM:

1. Log in to BRM Billing Care.
2. Click the **Assets** tab.
   If you are already in BRM, click **Refresh** to update the page with the latest changes.
3. Verifythe following:
   - The new services or assets are added to the list and the status is **Active**.
   - The existing services or assets are still on the list and the status is **Active**.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify the services and assets of the changed package order.

Creating a Modify Order

Amy Watkins is an existing subscriber who subscribed to a Supremo 5G Unlimited promotion. She wants to modify her existing services to add optional services that were not ordered earlier. She calls the service center to speak to a CSR to request this modification. The CSR creates a new order and modifies it to reflect Amy's request, and submits the order.

To create a modify order for an existing subscriber, there are various tasks that need to be performed as follows.

Creating a Modify Order in Siebel CRM

This procedure assumes that the order for the account was previously placed, submitted, and completed; therefore, you can scroll down to the **Installed Assets** section in Siebel CRM and view the products the account has subscribed to.

To create a modify order, customize the modify order, and verify the completion of modify order in Siebel CRM, sign in to Siebel CRM, and query the order details. See Querying an Order in *Siebel CRM Order Management Guide* for instructions on querying an order:

1. To modify any of the installed assets listed in the **Install Assets** section, select the **Promotion**, and click **Modify**.
2. Click **Continue** without making any changes.
3. Click **Done**.
   For **Residential** account type, the **Quotes** page is bypassed and you are taken directly to the **Sales Order** page, where a new order is auto-generated.
4. Enter the following data:

- **Order#** is pre-populated, but you can edit it per business requirements.
- For demo purposes, set the **Due date** to the order creation date. The date also needs to be updated for all Line Items under the **Due** column.
- In **Price List**, search for and select the appropriate price list. The change order is now created.
- Customize the bundle and click **Done**.
- Verify the updates in the **Line Items** section.

> **Note**
>
> There is no need to enter the **Service IDs** here. The **ServiceIDs** added as part of the new order are retained. However, you can change the **Service IDs** if needed, and the change is treated as an update.

5. Click **Submit**.
   After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances, see Verifying the Order Instances.

To verify the order status in Siebel CRM, see Verifying the Order Status in Siebel CRM.

To verify the order status in OSM, see Verifying the Order Status in OSM.

Verifying a Completed Order in BRM

To verify the modified order results in BRM:

1. Log in to BRM Billing Care.
2. Click the **Assets** tab.
   If you are already in BRM, click **Refresh** to update the page with the latest changes.
3. Verifythe following:
   - The new service or asset has been added to the list and the status is **Active**.
   - The old service or asset is still on the list and the status is **Canceled**.
   - The Promotion reference is carried forward to the newly added product.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify the modified services and assets of the order.

Suspending Services

Amy Watkins is an existing subscriber who subscribed to a Supremo 5G Unlimited promotion. She is planning to go for a vacation and wants to suspend a few services which are not required during her vacation. She calls the CSR and requests for suspending a few services from her promotion. The CSR places a suspend services order on Amy's account, which suspends the requested services.

To create a suspend services order for an existing subscriber, there are various tasks that need to be performed as follows.

Creating a Suspend Services Order in Siebel CRM

This procedure assumes that the order for the account was previously placed, submitted, and completed; therefore, you can scroll down to the **Installed Assets** section in Siebel CRM and view the products the account has subscribed to.

To create a suspend services order and verify the completion of the order in Siebel CRM, sign in to Siebel CRM, and query the order details. See Querying an Order in *Siebel CRM Order Management Guide* for instructions on querying an order:

1. In the **Installed Assets** section, select the services you want to suspend from the promotion.
2. Click the **Settings** icon, and select **Suspend**.
   The **Change Order Due Date** window appears.
3. Set the **Due Date** to **Order date**, to cause the action to happen immediately after order submission.
4. Click **Continue**.
5. Click **Done**.
   For **Residential** account type, the **Quotes** page is bypassed and you are taken directly to the **Sales Order** page, where a new order is auto-generated.
6. Enter the following data:
   - **Order#** is pre-populated, but you can edit it per business requirements.
7. In the **Line Items** section, verify the following:
   - The **Action** column displays **Suspend** for all the existing line items.
8. Click **Submit**.
   After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances, see Verifying the Order Instances.

To verify the order status in Siebel CRM, see Verifying the Order Status in Siebel CRM.

To verify the order status in OSM, see Verifying the Order Status in OSM.

Verifying a Completed Order in BRM

To verify the suspended services order results in BRM:

1. Log in to BRM Billing Care.
2. Click the **Assets** tab.
   If you are already in BRM, click **Refresh** to update the page with the latest changes.
3. Verifythe following:
   - The status for the services that are suspended is **Inactive**.
   - Any services that were in the **Canceled** status remain in the same status.
   - Any services that were in the **Not Set** status remain in the same status.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify if the respective services and assets are suspended.

Resuming Services

Amy Watkins is an existing subscriber whose services were previously suspended based on her request to suspend them. She calls the CSR to request for resuming the suspended services as she is back from her vacation. The CSR places an order to resume the services on Amy's account.

To create a resume services order for an existing subscriber, there are various tasks that need to be performed as follows.

Creating a Resume Services Order in Siebel CRM

This procedure assumes that the order for the account was previously placed, submitted, and completed; therefore, you can scroll down to the **Installed Assets** section in Siebel CRM and view the products the account has subscribed to.

To create a resume services order and verify the completion of the order in Siebel CRM, sign in to Siebel CRM, and query the order details. See Querying an Order in *Siebel CRM Order Management Guide* for instructions on querying an order:

1. In the **Installed Assets** section, select the services you want to resume from the promotion.
2. Click the **Settings** icon, and select **Suspend**.
   The **Change Order Due Date** window appears.
3. Set the **Due Date** to **Order date**, to cause the action to happen immediately after order submission.
4. Click **Continue**.
5. Click **Done**.
   For **Residential** account type, the **Quotes** page is bypassed and you are taken directly to the **Sales Order** page, where a new order is auto-generated.
6. Enter the following data:
   - **Order#** is pre-populated, but you can edit it per business requirements.
   - In **PriceList**, search for and select the appropriate price list.
7. In the **Line Items** section, verify the following:
   - The **Action** column displays **Resume** for all the existing line items.
8. Click **Submit**.
   After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances, see Verifying the Order Instances.

To verify the order status in Siebel CRM, see Verifying the Order Status in Siebel CRM.

To verify the order status in OSM, see Verifying the Order Status in OSM.

Verifying a Completed Order in BRM

To verify the suspended services order results in BRM:

1. Log in to BRM Billing Care.
2. Click the **Assets** tab.
   If you are already in BRM, click **Refresh** to update the page with the latest changes.
3. Verifythe following:
   - The status for the services that are resumed is **Active**.
   - Any services that were in the **Canceled** status remain in the same status.
   - Any services that were in the **Not Set** status remain in the same status.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify if the respective services and assets are resumed.

Disconnecting Services

Amy Watkins is an existing subscriber who wants to disconnect all of her services related to Supremo 5G Unlimited promotion and use a different Service Provider. She calls the service center to speak to a CSR to request to disconnect her services. The CSR places an order to disconnect all the services.

To create a disconnect services order for an existing subscriber, there are various tasks that need to be performed as follows.

Creating a Disconnect Services Order in Siebel CRM

This procedure assumes that the order for the account was previously placed, submitted, and completed; therefore, you can scroll down to the **Installed Assets** section in Siebel CRM and view the products the account has subscribed to.

To create a disconnect services order and verify the completion of the order in Siebel CRM, sign in to Siebel CRM, and query the order details. See Querying an Order in *Siebel CRM Order Management Guide* for instructions on querying an order:

> **Note**
>
> To disconnect a service, you must disconnect the commercial bundle, OTTs, mobile devices, and then the promotion.

1. In the **Installed Assets** section, first select the commercial bundle you want to disconnect from the promotion.
2. Click **Disconnect**.
   The **Change Order Due Date** window appears.
3. Set the **Due Date** to **Order date**, to cause the action to happen immediately after order submission.
4. Click **Continue** without making any changes.
5. Click **Done**.

For **Residential** account type, the **Quotes** page is bypassed and you are taken directly to the **Sales Order** page, where a new order is auto-generated.

6. Enter the following data:
   - **Order#** is pre-populated, but you can edit it per business requirements.
7. In the **Line Items** section, verify the following:
   - The **Action** column displays **Delete** for all line items for the commercial bundle.
8. Click **Submit**.

> **Note**
>
> - Repeat Steps 1 to 8 for disconnecting OTTs, mobile devices, and then the promotion.
> - When creating a Disconnect Services order for disconnecting the promotion, a Penalty charge line item gets added.

After you submit the order, the order transitions through various statuses as it proceeds through the process.

To verify the order instances, see Verifying the Order Instances.

To verify the order status in Siebel CRM, see Verifying the Order Status in Siebel CRM.

To verify the order status in OSM, see Verifying the Order Status in OSM.

Verifying a Completed Order in BRM

To verify the disconnect services order results in BRM:

1. Log in to BRM Billing Care.
2. Click the **Assets** tab.
   If you are already in BRM, click **Refresh** to update the page with the latest changes.
3. Verifythe following:
   - All services that were in the **Active** status are now in the **Canceled** status.
   - Any services that were in the **Canceled** status remain in the same status.
   - Any services that were in the **Not Set** status remain in the same status.

Verifying the Services and Assets in Siebel CRM

To verify the services and assets in the completed order:

1. Sign in to the Siebel CRM customer portal.
2. Navigate to the **Accounts Summary** page, **Installed Assets** section.
3. Verify if all the services and assets are disconnected.

About MACD Framework

The MACD Framework provides the foundational extensions required to support multiple Move, Add, Change, and Disconnect (MACD) transaction scenarios within the Oracle Communications Digital Business Experience reference solution. While this framework does

not constitute a business use case on its own, it enables sales teams, product managers, engineers, and end users to efficiently manage a wide range of customer service changes throughout the service lifecycle.

The framework assumes assets are present in the reference solution, and each product includes a resource type update, now available both in the product and the associated asset records.

> **Note**
>
> Currently, only orders are supported for all MACD transactions.

To support multiple MACD scenarios, the Siebel UI has been enhanced as follows:

- MACD actions can be enabled or disabled based on whether the service is either a **Mobile Service** or **Fixed Service**.
- Ensure all Digital Business Experience definable products at the service bundle level include any one of the following values:
  - Define **Transaction Charge** to the **PRODUCT_TYPE** LOV.
  - Define the following **DBE_MACD_EVENT** LOV values for transaction types supported by the framework:
    - New (for new orders only)
    - Modify service
    - Modify promotion
    - Upgrade/downgrade
    - Change service number
    - Change SIM
    - Change ownership
    - Change address
    - Barring
    - Contract renewal
    - Suspend
    - Resume
    - Disconnect
- **MACD Event Configuration**:

  > **Note**
  >
  > The fields mentioned below are maintained by Administrators from a dedicated list view under Administration – Order Management.

  - Event name (from **DBE_MACD_EVENT** LOV)
  - Ordering entity (Order)
  - Charge product name and ID (filter for products where type = “MACD Charge”)

  - Approval required flag

- The following actions are available under the Installed Assets applet in the Customer Summary view and the Installed Assets view:
  - **New**: If the Transaction Type is set to **New**, no need to select a value from the MACD action list.
  - **Modify**: It is mandatory to select a value from the MACD action list. Based on the value selected the transaction type will be set.
  - **Disconnect**: If the Transaction Type is set to **Disconnect**, no need to select a value from the MACD action list.
  - **Suspend**: If the Transaction Type is set to **Suspend**, no need to select a value from the MACD action list. It is mandatory to select a suspension reason.
  - **Resume**: If the Transaction Type is set to **Resume**, no need to select a value from the MACD action list. It is mandatory to select a suspension reason to resume the service.
- Menu options are dynamically enabled based on the current asset record's attributes, such as asset status, product type, service type, and resource type.
- The following table lists events or actions and their enablement conditions before initiating a modification:

Table 7-1 MACD Conditions

Action/s	Enable Conditions
Modify service Barring	Asset status = Active Root asset Id = asset id (or is null) Product type is not "Promotion"
Modify promotion Upgrade/downgrade Renewal Change ownership	Asset status = Active Product type is "Promotion"
Change service number (change MSISDN/ landline)	Asset status = Active Root asset Id = asset id (or is null) Resource type is "MSISDN" or "Landline"
Change address	Asset status = Active Root asset Id = asset id (or is null) Service type is "Fixed Service"
Suspend	Should follow the "Suspend" control on SIS OM Products & Services Root List Applet (Service) AND The Suspend action is enabled if the asset status is either **Active** or **Suspend**
Resume	Should follow the "Resume" control on SIS OM Products & Services Root List Applet (Service)
Disconnect	Should follow the "Disconnect" control on SIS OM Products & Services Root List Applet (Service)
Change SIM	Asset status = Active Root asset Id = asset id (or is null) Service type is "Mobile Service"

> **Note**
>
> – You can select the above-mentioned action or event using a drop-down menu, available next to the Modify action.
>
> – Suspend or Resume action will be initiated as a separate action or as a modification event.

> **Note**
>
> - When any MACD action is triggered, it will always create a new order.
> - If there are any chargeable products in the order, you need proper approvals to perform MACD actions.

About Modifying Services

This feature enables Enterprise Sales Representative (ESR) to initiate changes to components or attribute values of subscriber-owned services in response to subscriber requests. This process streamlines the handling of service modifications by providing a controlled workflow for managing service adjustments through either a quote or an order, depending on the configuration.

The process begins when an ESR receives a request to modify a subscriber's service. The ESR navigates to the Customer view, identifies the specific service, and initiates a modification transaction (either a Quote or an Order). The process concludes when the ESR completes the required modifications and is ready to submit the order or generate one from the modification quote.

> **Note**
>
> For all MACD scenarios, the term Document refers to both Quote and Order, as both types may be created based on the configuration.

To support this feature, the Siebel UI has been enhanced as follows:

- When processing a document with the transaction type **Modify Service**, the following controls and restrictions apply:
  - To prevent unintended order composition changes:
    - The Catalog view is disabled; ESRs cannot add new line items to the document.
    - The document header is read-only; ESRs cannot update header information.
    - Adding or removing line items from the document is not permitted.
    - Changing line item details (such as address, paying customer, or billing profile) is disabled.
  - Price adjustment options remain available for added components, following the standard price adjustment process.
  - Resource or inventory reservation is available only for added components (line items where Action = **ADD**).

  - Feasibility checks are required only for line items with the action **ADD** or **MODIFY**.
- The following service configurations are mandatory:
  - Use the **CfgEval** UI property as described in the *Siebel Product Administration Guide* to dynamically show or hide configurator controls.
  - Hide “Barring” products and the “MSISDN Validity number” product for **Modify Service** transactions.
  - Only display barring-related products in the configuration session when the transaction type is **Barring**.
- If the **Approval Required** flag is set to **Yes**, the approval process must be triggered according to the approval workflow. Else, no approval is required.

> **Note**
>
> Existing Siebel capabilities, such as future-dated orders and projected asset cache, remain available and are unaffected by the MACD Modify Service process.

About Barring Services (Manual)

This feature allows Enterprise Sales Representatives (ESRs) to partially or fully disable specific service features on a subscriber’s active account (for example, Bar Roaming, Bar Data, Bar Outgoing Calls, Full Bar, and so on). Barring differs from service suspension in that the service remains active, charges continue to apply, and the service commitment period is not affected by the bar’s activation or removal. This process supports both subscriber-initiated and policy-driven bars, enabling ESRs to efficiently handle various business scenarios involving usage restrictions.

Types of Barring

- **Customer Bars**:
  - Applied or removed by the subscriber (service owner) or any ESR.
  - May be requested freely and through self-service or agent channels (subject to transaction fees, if applicable).
- **Policy Bars**:
  - Applied by the provider in response to policy events, such as fraud or collections.
  - Managed through automated processes or authorized ESR intervention.
  - Removal by the service owner requires support from an authorized ESR or resolution of the policy event.

The process begins when an ESR receives a request to bar a subscriber’s service. The ESR navigates to the Customer view to locate the installed service (asset) to be barred; if the asset does not satisfy the MACD Framework enablement rules for Barring, the Barring action is unavailable and no further processing occurs.

If the asset is eligible, the ESR selects it from the Installed Assets applet (Customer Summary) and initiates Barring, where a transaction charge is automatically added, and the order header is populated with Transaction Type = Barring and Approval Required as defined in the matrix. The ESR is then taken into the service configuration session to choose the required barring options, and upon completing configuration is returned to the Order view to validate, obtain approval if applicable, and submit the order.

> **Note**
>
> Barring services are initiated and processed exclusively through orders; quotes are not used for this feature.

To support this feature, the Siebel UI has been enhanced as follows:

- When the transaction type is set to **Bar Service**:
  - All document actions are disabled except for configuration, validation, approval, and submission.
  - The document header is read-only.
  - Adding or removing line items, except through the configuration session is disabled.
  - Editing line item details (address, billing profile, paying customer, and so on) is not permitted.
  - The Catalog view is disabled.
- The following fields in the Order header should be read-only:
  - **Account Details**: Customer, Billing and Service account, First and Last name, Billing profile
  - **Pricing Details**: Price list, Manual discounting options, Currency
- The following fields in the Document line item should be read-only:
  - **Account Details**: Customer, Billing and Service account, First and Last name, Billing profile
  - Addresses fields
  - Product, Service ID, Service Point, Service Address
  - Any price and discount-related fields
  - **Pricing Details**: Price list, Manual discounting options, Currency
- Barring products (for example, for Roaming, Data, Outgoing Calls) are defined and included as separate product relationships:
  - One group for subscriber-initiated bars.
  - One group for policy bars.
  - All other product relationships are hidden or view-only when the transaction type is **Bar Service**, using the **CfgEval** property as described in the *Siebel Product Administration Guide*.
- Before submission, the system verifies that all required configurations are complete and valid (mandatory selections, constraints met).
- If the **Approval Required** flag is set to **Yes**, the order follows the approval workflow; otherwise, approval is not triggered.
- All other standard order submission validations are skipped for barring transactions.

> **Note**
>
> User roles and responsibilities must be configured to allow or restrict ESR access to apply or manage policy bars, as defined in catalog settings.

About the Reference Solution Test Catalog

This chapter provides information about the reference solution test catalog, which consists of a list of automated test cases (created using product models, seed data, and sample orders) that are packaged and delivered to you within the reference solution package. This test catalog helps you to validate whether the contents of the reference solution are deployed correctly. This test catalog can be deployed and executed using Oracle Communications Solution Test Automation Platform (STAP).

Introduction to STAP

STAP is a powerful automation platform that allows you to automate your end-to-end business use cases without writing a single line of code. By providing a no-code automation solution, STAP enables you to automate your workflows easily with an inbuilt Behavior Driven Development (BDD) language without any technical expertise. This makes it an ideal automation platform for improving efficiency and productivity.

The following are the benefits of using STAP for deploying the reference solution package:

- STAP helps to improve the reference solution quality by automating the test cases and identifying potential issues. It provides accurate results that help to ensure that the reference solution is functioning as expected.
- STAP automates testing, saving time and effort for testing teams. It enables teams to focus on other critical tasks, such as improving the reference solution use cases.
- STAP handles high traffic and growing demands, making it an ideal automation solution for diverse testing requirements.

For more information about STAP:

- See Installing STAP in *Oracle Communications Solution Test Automation Platform Deployment Guide*.
- See *Oracle Communications Solution Test Automation Platform User Operations Guide* for information about using STAP.
- See Troubleshooting STAP Deployment in *Oracle Communications Solution Test Automation Platform Deployment Guide*.

Reference Solution Automation Test Cases

This section provides a list of reference solution test cases that are automated using STAP.

The test cases are categorized into design time and run time categories as follows:

- Design-time test cases include the following:
  - Postpaid Mobile
  - Broadband
  - Digital TV
  - Home Phone
  - Dual Play
  - Triple Play
  - Quad Play

- Run-time test cases include the following:
  - First-Time Purchase
  - Modify
  - Disconnect Services
  - Upgrade or Downgrade
  - Payment
  - Adjustment
  - Update Contact and Address

The Table 7-2 table provides details of total number of automation test cases available in the reference solution package.

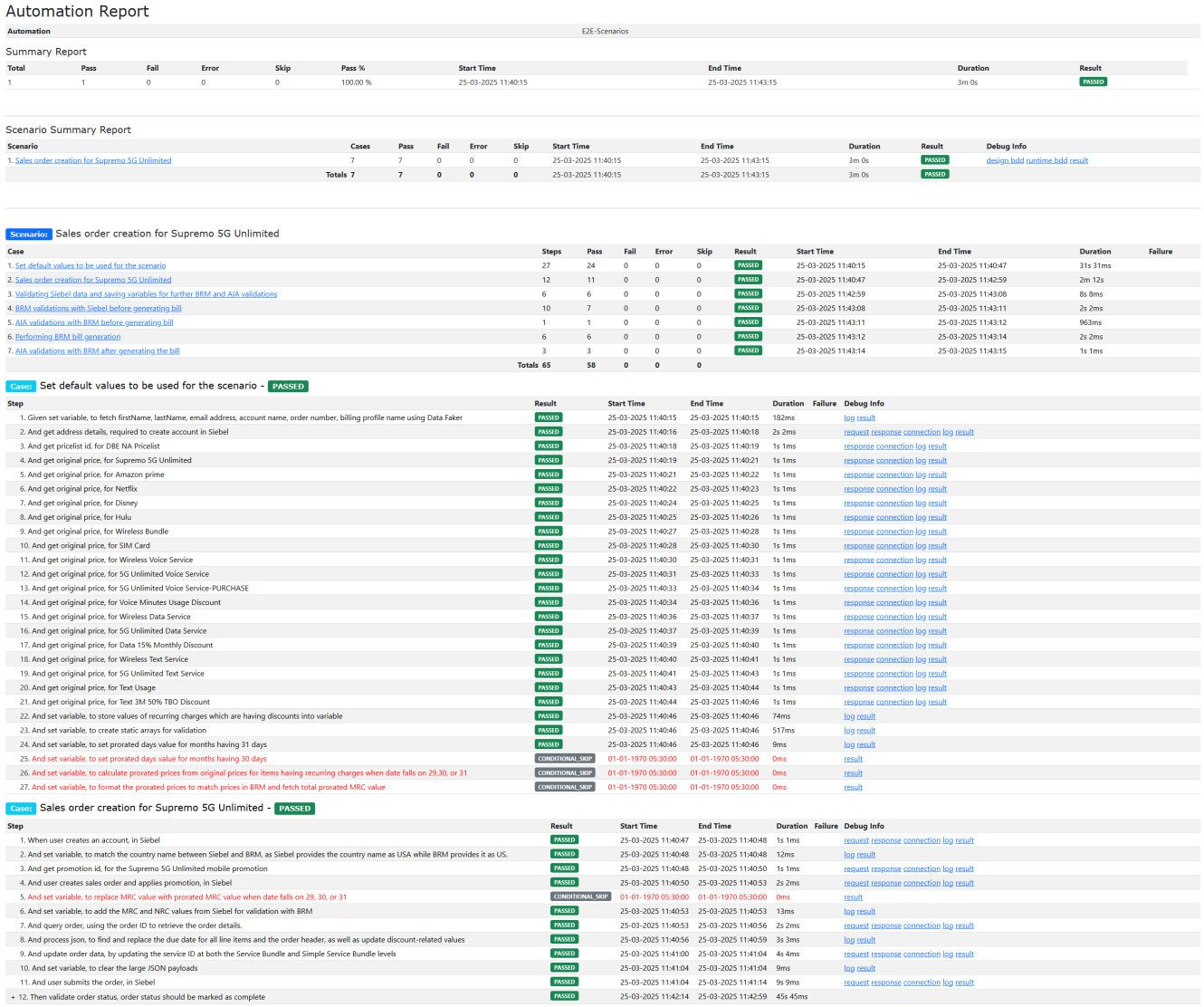
Table 7-2 Automation Test Case Coverage

Product Models	Creation of Product Models	First-Time Purchase	Modify	Disconnect Services	Downgrade or Upgrade	Payment	Adjustment	Update Contact and Address	Total Test cases for each Model
Postpaid Mobile	3	8	1	1	1	NIL	NIL	NIL	14
Broadband	4	6	3	NIL	1	NIL	NIL	NIL	14
Digital TV	2	2	NIL	NIL	NIL	1	1	2	8
Home Phone	2	2	NIL	1	NIL	NIL	NIL	NIL	5
Dual Play	8	8	NIL	NIL	NIL	NIL	NIL	NIL	16
Triple Play	3	3	NIL	1	NIL	NIL	NIL	NIL	7
Quad Play	3	3	NIL	NIL	NIL	NIL	NIL	NIL	6
Total Count	25	32	4	3	2	1	1	2	70

After executing the above automation test cases, all the design-time product models will be created in Launch, Siebel, and PDC, accounts will be created for each run-time feature, and a test report is generated, which is published using STAP.

The following figure displays a sample test report of a sample automation test case.

Figure 7-1 Sample Automation Test Report



Automation Report

Automation	E2E-Scenarios

Summary Report

Total	Pass	Fail	Error	Skip	Pass %	Start Time	End Time	Duration	Result
1	1	0	0	0	100.00 %	25-03-2025 11:40:15	25-03-2025 11:43:15	3m 0s	PASSED

Scenario Summary Report

Scenario	Cases	Pass	Fail	Error	Skip	Start Time	End Time	Duration	Result	Debug Info
1. Sales order creation for Supremo 5G Unlimited	7	7	0	0	0	25-03-2025 11:40:15	25-03-2025 11:43:15	3m 0s	PASSED	design bdd runtime bdd result
Totals	7	7	0	0	0	25-03-2025 11:40:15	25-03-2025 11:43:15	3m 0s	PASSED	

Scenario: Sales order creation for Supremo 5G Unlimited

Case	Steps	Pass	Fail	Error	Skip	Result	Start Time	End Time	Duration	Failure
1. Set default values to be used for the scenario	27	24	0	0	0	PASSED	25-03-2025 11:40:15	25-03-2025 11:40:47	31s 31ms	
2. Sales order creation for Supremo 5G Unlimited	12	11	0	0	0	PASSED	25-03-2025 11:40:47	25-03-2025 11:42:59	2m 12s	
3. Validating Siebel data and saving variables for further BRM and AIA validations	6	6	0	0	0	PASSED	25-03-2025 11:42:59	25-03-2025 11:43:08	8s 8ms	
4. BRM validations with Siebel before generating bill	10	7	0	0	0	PASSED	25-03-2025 11:43:08	25-03-2025 11:43:11	2s 2ms	
5. AIA validations with BRM before generating bill	1	1	0	0	0	PASSED	25-03-2025 11:43:11	25-03-2025 11:43:12	963ms	
6. Performing BRM bill generation	6	6	0	0	0	PASSED	25-03-2025 11:43:12	25-03-2025 11:43:14	2s 2ms	
7. AIA validations with BRM after generating the bill	3	3	0	0	0	PASSED	25-03-2025 11:43:14	25-03-2025 11:43:15	1s 1ms	
Totals	65	58	0	0	0					

Case: Set default values to be used for the scenario - PASSED

Step	Result	Start Time	End Time	Duration	Failure	Debug Info
1. Given set variable, to fetch firstName, lastName, email address, account name, order number, billing profile name using Data Faker	PASSED	25-03-2025 11:40:15	25-03-2025 11:40:15	182ms		log result
2. And get address details, required to create account in Siebel	PASSED	25-03-2025 11:40:16	25-03-2025 11:40:18	2s 2ms		request response connection log result
3. And get pricelist id, for DBE NA Pricelist	PASSED	25-03-2025 11:40:18	25-03-2025 11:40:19	1s 1ms		response connection log result
4. And get original price, for Supremo 5G Unlimited	PASSED	25-03-2025 11:40:19	25-03-2025 11:40:21	1s 1ms		response connection log result
5. And get original price, for Amazon prime	PASSED	25-03-2025 11:40:21	25-03-2025 11:40:22	1s 1ms		response connection log result
6. And get original price, for Netflix	PASSED	25-03-2025 11:40:22	25-03-2025 11:40:23	1s 1ms		response connection log result
7. And get original price, for Disney	PASSED	25-03-2025 11:40:24	25-03-2025 11:40:25	1s 1ms		response connection log result
8. And get original price, for Hulu	PASSED	25-03-2025 11:40:25	25-03-2025 11:40:26	1s 1ms		response connection log result
9. And get original price, for Wireless Bundle	PASSED	25-03-2025 11:40:27	25-03-2025 11:40:28	1s 1ms		response connection log result
10. And get original price, for SIM Card	PASSED	25-03-2025 11:40:28	25-03-2025 11:40:30	1s 1ms		response connection log result
11. And get original price, for Wireless Voice Service	PASSED	25-03-2025 11:40:30	25-03-2025 11:40:31	1s 1ms		response connection log result
12. And get original price, for 5G Unlimited Voice Service	PASSED	25-03-2025 11:40:31	25-03-2025 11:40:33	1s 1ms		response connection log result
13. And get original price, for 5G Unlimited Voice Service-PURCHASE	PASSED	25-03-2025 11:40:33	25-03-2025 11:40:34	1s 1ms		response connection log result
14. And get original price, for Voice Minutes Usage Discount	PASSED	25-03-2025 11:40:34	25-03-2025 11:40:36	1s 1ms		response connection log result
15. And get original price, for Wireless Data Service	PASSED	25-03-2025 11:40:36	25-03-2025 11:40:37	1s 1ms		response connection log result
16. And get original price, for 5G Unlimited Data Service	PASSED	25-03-2025 11:40:37	25-03-2025 11:40:39	1s 1ms		response connection log result
17. And get original price, for Data 15% Monthly Discount	PASSED	25-03-2025 11:40:39	25-03-2025 11:40:40	1s 1ms		response connection log result
18. And get original price, for Wireless Text Service	PASSED	25-03-2025 11:40:40	25-03-2025 11:40:41	1s 1ms		response connection log result
19. And get original price, for 5G Unlimited Text Service	PASSED	25-03-2025 11:40:41	25-03-2025 11:40:43	1s 1ms		response connection log result
20. And get original price, for Text Usage	PASSED	25-03-2025 11:40:43	25-03-2025 11:40:44	1s 1ms		response connection log result
21. And get original price, for Text 3M 50% TBO Discount	PASSED	25-03-2025 11:40:44	25-03-2025 11:40:46	1s 1ms		response connection log result
22. And set variable, to store values of recurring charges which are having discounts into variable	PASSED	25-03-2025 11:40:46	25-03-2025 11:40:46	74ms		log result
23. And set variable, to create static arrays for validation	PASSED	25-03-2025 11:40:46	25-03-2025 11:40:46	517ms		log result
24. And set variable, to set prorated days value for months having 31 days	PASSED	25-03-2025 11:40:46	25-03-2025 11:40:46	9ms		log result
25. And set variable, to set prorated days value for months having 30 days	CONDITIONAL_SKIP	01-01-1970 05:30:00	01-01-1970 05:30:00	0ms		result
26. And set variable, to calculate prorated prices from original prices for items having recurring charges when date falls on 29,30, or 31	CONDITIONAL_SKIP	01-01-1970 05:30:00	01-01-1970 05:30:00	0ms		result
27. And set variable, to format the prorated prices to match prices in BRM and fetch total prorated MRC value	CONDITIONAL_SKIP	01-01-1970 05:30:00	01-01-1970 05:30:00	0ms		result

Case: Sales order creation for Supremo 5G Unlimited - PASSED

Step	Result	Start Time	End Time	Duration	Failure	Debug Info
1. When user creates an account, in Siebel	PASSED	25-03-2025 11:40:47	25-03-2025 11:40:48	1s 1ms		request response connection log result
2. And set variable, to match the country name between Siebel and BRM, as Siebel provides the country name as USA while BRM provides it as US.	PASSED	25-03-2025 11:40:48	25-03-2025 11:40:48	12ms		log result
3. And get promotion id, for the Supremo 5G Unlimited mobile promotion	PASSED	25-03-2025 11:40:48	25-03-2025 11:40:50	1s 1ms		request response connection log result
4. And user creates sales order and applies promotion, in Siebel	PASSED	25-03-2025 11:40:50	25-03-2025 11:40:53	2s 2ms		request response connection log result
5. And set variable, to replace MRC value with prorated MRC value when date falls on 29, 30, or 31	CONDITIONAL_SKIP	01-01-1970 05:30:00	01-01-1970 05:30:00	0ms		result
6. And set variable, to add the MRC and NRC values from Siebel for validation with BRM	PASSED	25-03-2025 11:40:53	25-03-2025 11:40:53	13ms		log result
7. And query order, using the order ID to retrieve the order details.	PASSED	25-03-2025 11:40:53	25-03-2025 11:40:56	2s 2ms		request response connection log result
8. And process json, to find and replace the due date for all line items and the order header, as well as update discount-related values	PASSED	25-03-2025 11:40:56	25-03-2025 11:40:59	3s 3ms		log result
9. And update order data, by updating the service ID at both the Service Bundle and Simple Service Bundle levels	PASSED	25-03-2025 11:41:00	25-03-2025 11:41:04	4s 4ms		request response connection log result
10. And set variable, to clear the large JSON payloads	PASSED	25-03-2025 11:41:04	25-03-2025 11:41:04	9ms		log result
11. And user submits the order, in Siebel	PASSED	25-03-2025 11:41:04	25-03-2025 11:41:14	9s 9ms		request response connection log result
+ 12. Then validate order status, order status should be marked as complete	PASSED	25-03-2025 11:42:14	25-03-2025 11:42:59	45s 45ms		

For information about troubleshooting, see Troubleshooting STAP in *Oracle Communications Solution Test Automation Platform Deployment Guide*.

A

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 A-1 defines the terms used in this appendix.

Table A-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 subscriber.
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
CommsOrder	A variable signifying one of the following: • Sales Order • Fulfillment Order • Provisioning Order

Order Header Component Attributes

Table A-2 lists the attributes for order headers in communications orders.

Table A-2 Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Order ID	Uniquely identifies each order.	NA	No	None	Produces a unique identifier for all orders, including revision orders. Unlike Order Number, Order ID is different for revisions of the same base order. Used by Oracle AIA for cross-reference.	SaleOrderEBO/ Identification/ BusinessComponentID

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Order Number	Identifies an order across revisions.	NA	No	None	A revision number >1 does not necessarily mean that this is a revision order from OM Fulfillment. You can create an order in Siebel CRM and revise it several times before submitting it. If an Order Number matches an in-flight order, then the order is treated as a revision order. When an order is revised, this number stays the same. OM uses this number to identify the base order. If the same order number with the same revision is submitted, then OM rejects the revision order and places it in fallout.	*CommsOrder*EBO/ Identification/ID
Revision	A revision sequence number that, with the order number, represents the user key to an order.	NA	No	None	If an order is received with an Order Number equal to that of an in-flight order and the newly received order has a higher revision number, then OM assumes the order is a revision order and proceeds to analyze the Order Lines. If the revision number is equal or lower than that of the base order, the revision is rejected.	*CommsOrder*EBO/ Identification/Revision/ Number
Success Dependency	Declares if all order lines must fulfill successfully or else the whole order fails (all or none). When the order level Success Dependency is set to All or None, it takes precedence over Order Line Success Dependency designations because it is more restrictive.	DEFAULT ALL OR NONE	No	None	None	*CommsOrder*EBO/ PartialFulfillmentAllowedIndicator

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Fulfillment Mode	Qualifies the nature of fulfillment request.	Deliver, Qualify, Cancel, Initiate billing, Fulfill billing	No	None	Communications service providers (CSPs) may extend support to other modes, such as Design, Schedule and Cost. CRM can cancel an order through a revision order with no order lines or by resending the order with Fulfillment Mode = *Cancel*. OM is expected to honor the two alternatives for canceling an order, providing no order lines reaches the point of no return. When used on billing EBS, Fulfillment Mode has a different meaning. It determines the type of Billing request: *Initiate* or *Fulfill.*	*CommsOrder*EBO/ FulfillmentModeCode
subscriber Class	Identifies type of subscriber: Residential, Business, etc.	Residential Business	No	None	None	*CommsOrder*EBO/ CustomerPartyReference / CustomerPartyAccountTypeCode
Organization ID	Identifies the organization/LOB generating the order. No cross-reference exists.	NA	No	None	No cross-reference. OM should use the application-specific ID if required in any of the rules; if Organization IDs are unique and synchronized across all order capture systems.	*CommsOrder*EBO/ BusinessUnitReference/ BusinessUnitIdentification /ID
Sales Channel	Identifies the sales channel.	NA	No	None	None	*CommsOrder*EBO/ SalesChannelCode
Job ID	A string or number that uniquely identifies the job to orchestration.	NA	No	None	Track orders that belong to a bulk or batch job.	*CommsOrder*EBO/ ProcessingNumber
Sequence in Job	A number that identifies the order sequence within the job.	NA	No	None	None	*CommsOrder*EBO/ ProcessingSequenceNumber

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Job Type	Identifies the type of job. This information identifies the threshold for creating a consolidated SR for Bulk or Batch Orders. This value is optional for orders whose Job Cardinality is 1. By default, this value is HETROGENEOUS.	Heterogeneous,homogeneous, third-party homogeneous, third-party heterogeneous, correlated	No	None	None	*CommsOrder*EBO/ProcessingTypeCode
Job Cardinality	Indicates the total number of orders within the job.	NA	No	None	None	*CommsOrder*EBO/ProcessingQuantity
Parent Order ID	Order ID of another order that indicates the fulfillment for this order does not start before the parent order fulfillment completes.	NA	No	None	This attribute applies to explicit order-to-order dependencies and is not limited to follow-on orders. For example, in a B2B scenario, a large order can be divided into some smaller orders, with one order acting as the root order for all other orders and the remainder of the orders chained using the parent order ID attribute.	*CommsOrder*EBO/Parent*CommsOrder*Reference/*CommsOrder*Identification/BusinessComponentID
Fulfillment Priority	Indicates relevant priority of order fulfillment across orders. A lower value indicates a higher priority. Accepts values 0 to 9 in accordance with JMS Queue support.	9,7,5,3	No	None	EBM value: Siebel value 9: Urgent. Used for expedited orders. 7: High. CSP determines its use. 5: Medium. CSP determines its use. 3: Low. Recommended for job orders. Note: Oracle Advanced Queuing (AQ) and JSM priority values have the inverse order of precedence.	*CommsOrder*EBO/FulfillmentPriorityCode

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Order Type	Sometimes indirectly determines sales channel to drive compensation process.	Sales Order	No	None	None	*CommsOrder*EBO/TypeCode
Requested Delivery Date Time	Overall order level due date that provides the default due date at each line level. Can be overridden at each line.	NA	Yes	None	None	*CommsOrder*EBO/RequestedDeliveryDateTime
Fulfillment Status	Reports aggregate order fulfillment status.	In Progress, Failed, Canceled, Complete	Yes	None	This is different from the Status attribute tracked within Siebel CRM.	*CommsOrder*EBO/Status/Code
Status Context	Provides details about the current status. The implementer configures this value.	NA	Yes	None	OM can use this to track the milestone causing the status change, along with context details such as *error message, cause for cancel*. One primary scenario that the Order Header/Status Context is populated with revision orders that cancels Order Lines by dropping them from the revision and if the revision is rejected. In that case the orchestration system does not have a line on the revision order to provide fallout status and context. In such a case the header level status context is used to identify the base line the cause for the fallout.	*CommsOrder*EBO/Status/Description
Owner Account ID	Identifies the owner account.	NA	Yes	None	Cross-referenced.	*CommsOrder*EBO/CustomerPartyReference/CustomerPartyAccountIdentification/BusinessComponentID

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
Owner Account Name	Identifies the Account Name. You can enter or derive this value from contact first name + last name of primary contact associated with the account.	NA	Yes	None	Required for network inventory tracking of service owner.	*CommsOrder*EBO/ CustomerPartyReference / CustomerPartyAccountName
Owner Account Number	Identifies account number to subscriber.	NA	Yes	None	None	*CommsOrder*EBO/ CustomerPartyReference / CustomerPartyAccountIdentification/ID
Account Contact ID	Foreign key to contact record that holds personal and contact details of the subscriber/ company representative who is placing the order and is the contact person for anything related to the order process.	NA	Yes	None	None	*CommsOrder*EBO/ CustomerPartyReference / CustomerPartyAccountContactIdentification/ BusinessComponentID
Account Contact Address (component)	Identifies the address used to communicate with the Contact ID.	NA	Yes	None	--	*CommsOrder*EBO/ CustomerPartyReference / CustomerPartyAccountContactAddressCommunication/ AddressCommunication/ Address
Project ID	Identifies project record if the order to be delivered is part of a project that contains related orders. Foreign key reference. No cross-reference.	NA	Yes	None	No cross-reference for 2.4.	*CommsOrder*EBO/ ProjectReference/ ProjectIdentfication/ID

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
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	FulfillmentOrderEBO/ FulfillmentSystemTypeCode
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	FulfillmentOrderEBO/ FulfillmentTargetSystemID
Order Changed Indicator	OM sets this attribute to Yes if the order changed significantly such that CRM should make a copy of the subscriber order to preserve the subscriber 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 subscriber's intent is compromised.	*CommsOrder*EBO/ OrderChangedIndicator
Sales Representative ID	CRM User ID that identifies the sales representative who entered the order.	NA	No	None	No cross-reference. Use the application ID.	*CommsOrder*EBO/ SalespersonPartyReference/PartyIdentification/ID

Table A-2 (Cont.) Order Header Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values	Asset-able	Prior Value Available	Comments	EBO Structure XPath
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	*CommsOrder*EBO/CustomerPartyReference/CustomerPartyAccountContact/FirstName *CommsOrder*EBO/CustomerPartyReference/CustomerPartyAccountContact/LastName *CommsOrder*EBO/CustomerPartyReference/CustomerPartyAccountContactPhoneCommunication/PhoneCommunication/CompleteNumber *CommsOrder*EBO/CustomerPartyReference/CustomerPartyAccountContactEmailCommunication/EmailCommunication/
Price List	Indicates default price list on order.	NA	No	No	If no price list value is supplied for order lines for products synchronized with rate plan selectors, the price list on the order header is used. If products on order lines were not synchronized with rate plan selectors, the default price list from the **AIAConfigurationProperties.xml** file is used.	*CommsOrder*EBM/DataArea/*CommsOrder*/PriceListReference/PriceListIdentification/ID
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 subscriber order to a service order.	*CommsOrder*EBO/OrderSubject

Order Line Component Attributes

Table A-3 lists the attributes for order lines in communications orders.

Table A-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.	*CommsOrder*EBO/ *CommsOrder*Line/ Identification/ BusinessComponentID
Base Line ID	References base order line revised by this order line.	NA	No	None	Uses a cross-reference.	*CommsOrder*EBO/ *CommsOrder*Line/ Original*CommsOrder*Line Reference/ *CommsOrder*LineIdentification/ BusinessComponentID
Asset Integration ID	Uniquely identifies an instance of a product that was or is being purchased.	NA	Yes	AIA 2.0	Cross-referenced. CRM populates the Asset Integration ID on all Order Lines, regardless of the Assetable state on the subject of the Order Line or whether the Order Line is for a new or existing service. A revision should never change the Asset Integration ID. When a product is dropped as part of one product hierarchy (CP or Promotion) and then added through another product hierarchy (CP or Promotion), the Asset Integration ID for the two line items are different, although for the same product.	*CommsOrder*Line/ InstalledProductReference / InstalledProductIdentification/BusinessComponentID
Line Number	Identifies the line regarding its position in the line item tree.	NA	No	None	Line number establishes the parent child relationship between Order Lines of the same order, but it may vary across revisions. Therefore, do not rely on it for matching Order Lines across revisions.	*CommsOrder*EBO/ *CommsOrder*Line/ Identification/ID

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Parent Line	References parent order line in the line items tree instantiated according to the product model definition. Points to itself if the item does not have an associated parent item.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ Parent*CommsOrder*LineIdentification/ BusinessComponentID
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	*CommsOrder*EBO/ *CommsOrder*Line/ RootParent*CommsOrder*LineIdentification/ BusinessComponentID
Related Line ID	BRM adaptors use to relate one-time charges to base line ID.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ChargeParentLineIdentification/ BusinessComponentID
Related Asset Integration ID	Links Move-Add to Move-Delete line items.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ InstalledProductReference/ PriorInstalledProductIdentification/ BusinessComponentID
Depends On Line ID	Indicates order line item ID of a previous order line item that is changed by this order. Follow-on orders use this value to capture dependencies of the order line items in the follow-on order-to-order line items of original orders.	NA	No	None	Cross-referenced	*CommsOrder*EBO/ *CommsOrder*Line/ *DependingCommsOrder*LineReference/ *CommsOrder*LineIdentification/ BusinessComponentID

Table A-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
Depends On Order ID	Identifies order ID of an in-flight order, which is the basis for this follow-on order line item.	NA	No	None	Cross-referenced	*CommsOrder*EBO/ *CommsOrder*Line/ Depending*CommsOrder*Reference/ *CommsOrder*Identification/ BusinessComponentID
Promotion Line ID	References an order line that represents the promotion/ marketing offer under which the order line is being purchased.	NA	No	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ Promotion*CommsOrder*LineReference/ Promotion*CommsOrder*LineIdentification/ Identification/ BusinessComponentID
Promotion Asset Integration ID	References an asset that represents the promotion/ marketing offer under which the order line is being purchased.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *PromotionCommsOrder*LineReference/ InstalledProductReference/ InstalledProductIdentification/BusinessComponentID
Product ID	References product record based on which order line is instantiated. Foreign key reference.	NA	Yes	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ Identification/ BusinessComponentID
Quantity	Identifies the quantity of the item requested by a subscriber. Default value is 1.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ OrderQuantity
Action Code	Specify action required to meet subscriber request.	None, Add, Update,Suspend Resume, Delete,Move-Add, Move-Delete	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ServiceActionCode
Deliver To Address	Address record that represents the delivery/ service installation address.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ ServiceAddress/Address

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Requested Delivery Date Time	When Null, the requested date for delivery of the goods or service is ASAP; otherwise, it is the specified date. This date is not guaranteed. Typically, it is a future date; if it is a past date, then the default behavior equals a Null value.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ RequestedDeliveryDateTime
Usage Start Date	Determines the date when usage events should start being rated. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to the current date.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ ServiceUsageStartDate
Cycle State Date	Determines the date when cycle charges should start being billed. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to the current date according to previous patterns.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ CycleStartDate

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Purchase Date	Determines the date when one-time purchase charges should be billed. The value for this attribute is populated by CRM, OM Fulfillment flows, or kept to Null for BRM default to current date according to previously mentioned patterns.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ PurchaseDate
Service Start Date	Indicates effective start date of service.	NA	Yes	None	Initially computed by Siebel based on Due Date and then updated by Order Management based on Actual Delivery Date.	*CommsOrder*EBO/ *CommsOrder*Line/ EffectiveTimePeriod/ StartDateTime
Earliest Delivery Date	Identifies the date when the work associated to the order can start. Typically used for fulfillment actions that require subscriber presence such as in cases subscriber must be available to install service or deliver shipment.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ EarliestDeliveryDateTime
Service End Date	Indicates the effective end date of service. Applies to services with a specified duration.	NA	Yes	None	Initially computed in Siebel and then updated by Order Management. Update is sent to Siebel.	*CommsOrder*EBO/ *CommsOrder*Line/ EffectiveTimePeriod/ EndDateTime

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Actual Delivery Date Time	Determines the date when the purchased product or service is considered available to the subscriber by the CSP. This date may be when physical goods are shipped, delivered, or their receipt is acknowledged. For service-based products, the service is activated on this date. This date is computed in the OM Fulfillment flow according to previous patterns.	NA	Yes	None	Oracle BRM does not allow for starting any charges before the Purchase Date; therefore, the ABCS for Oracle BRM always overrides the Purchase Date if it is later than any of the Cycle or Usage start dates. OM should facilitate calculation of Order Line level Actual Delivery Date and Order Line attributes for billing Usage Start Date, Cycle Start Date, and Purchase Date.	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ ActualDeliveryDateTime
Expected Delivery Date Time	Indicates the due date expected by the system because of Design and Assign. The default is the Order Due Date when the order is created by CRM.	NA	No	None	Computed by OM based on preconfigured time estimates on fulfillment actions. Used by OM to communicate to CRM changes to expected delivery date of specific Order Lines.	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ ExpectedDeliveryDateTime
Fulfillment Status	Updates orchestration and CRM regarding the current status of order line fulfillment at a high level.	In Progress, Failed,Canceled Complete	Yes	None	Additional values can be added as an extension.	*CommsOrder*EBO/ *CommsOrder*Line/Status/ Code
Milestone	Fulfillment passes the last reached milestone into this field.	NA	No	None	None	*CommsOrder*/ *CommsOrder*Line/ MilestoneCode

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Status Context	Provides details about the current status of the order line. The implementer configures this value.	NA	Yes	None	OM can include the reached milestone (from the fulfillment system, the cause for the status update that is necessary because of dynamic nature of fulfillment plan) and a textual string for context per current status as follows (canonical Status / status context): Submitted / NA In Progress / <milestone>: context text Failed / <milestone>: reason text Canceled / <milestone>: reason text Complete / NA In Progress: Context Text could be used to indicate any of the following among others: Requires subscriber interaction Delivery is expected to be delayed	*CommsOrder*EBO/ *CommsOrder*Line/Status/ Description
Point-of-no-return	Determines if Siebel should allow order line revisions to be submitted.	Not yet, Hard	No	None	OM Fulfillment flows allow configuration of setting a hard PONR when a condition is met for a particular service. When a hard PONR is reached for an Order Line in OM, a status update is issued to reflect the same in CRM. Additional values such as SOFT can be added as an extension.	*CommsOrder*EBO/ *CommsOrder*Line/ RevisionPermissibleCode

Table A-3 (Cont.) Order Line Component Attributes

Functional Attribute Name	Attribute Usage Semantics	Seeded Values and Value Type	Asset-able	Prior Value Available	Remarks	EBO Structure XPath
Billing Account	References an account record that represents the bill payer or the branch of a company responsible for bill payment. This value may be a subscriber account or an account from the account hierarchy.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ BillToPartyReference/ CustomerPartyAccountIdentification/ BusinessComponentID
Billing Profile	References the billing profile record that holds the subscriber's billing/payment preferences. This value may be associated to the subscriber account or to a separate billing account.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ BillToPartyReference/ BillingProfileReference/ BillingProfileIdentification/ BusinessComponentID
Payment Profile	Identifies the Payment Profile.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ BillToPartyReference/ BillingProfileReference/ PaymentProfileReference/ PaymentProfileIdentification/BusinessComponentID
Service Account	References an account record that represents a service user or the branch of the company where service is installed. This value may be subscriber account or an account from the account hierarchy.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ OwnerPartyReference/ CustomerPartyAccountIdentification/ BusinessComponentID

Table A-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. The value must be the parent account in the hierarchy of the service account.	NA	Yes	Yes	None	New value: ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/SalesOrderLine/OwnerAccountId Prior value: ProcessSalesOrderFulfillmentEBM/DataArea/ProcessSalesOrderFulfillment/PriorSalesOrderLine/OwnerAccountId
Owner Contact	Represents a contact of the subscriber account or service account who should be contacted during fulfillment of the line if required.	NA	Yes	None	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*Schedule/OwnerPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID
Shipping Contact	Represents a contact of the subscriber account or service account who should be contacted for shipping purposes.	NA	Yes	None	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*Schedule/ShipToPartyReference/CustomerPartyAccountContactIdentification/BusinessComponentID
Node	Alphanumerically references the root order line that corresponds to access at site A of a connection. This value is relevant for network ordering only.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="Node"]/ValueText

Table A-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 Node	Alphanumerically references the root order line that corresponds to access at site B of a connection. This value is relevant for network ordering only.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToNode"]/ValueText
Network ID	Unique compound product number that represents the virtual network ID. Relevant for network orders. Provided by default from the order number and cascaded to network connection items.	NA	Yes	Yes	Identifies which Access and Nodes belong to the same network. This information may be of value to decomposition.	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="NetworkID"]/ValueText
Port Number	Identifies the port number allocated to the access circuit connected to provide (starting) edge router during the fulfillment process.	NA	Yes	Yes	For new services, port number comes back from Network Inventory through provisioning.	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="PortNumber"]/ValueText
To Port Number	Identifies the port number allocated to the access circuit connected to provide (ending) edge router during the fulfillment process.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToPortNumber"]/ValueText
Service Address Prefix	Identifies the area code/NPA for the access circuits on starting or two ends of the connection.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ServiceAddressPrefix"]/ValueText

Table A-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 Address Prefix	Identifies the area code/NPA for the access circuits on the end of the connection.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressPrefix"]/ValueText
Access Circuit	Provides the Common Language Location Identification (CLLI) for the access circuit on two sides or starting side of the connection.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="AccessCircuit"]/ValueText
To Access Circuit	Provides the CLLI for the access circuit on ending side of the connection.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToAccessCircuit"]/ValueText
To Service Account ID	Identifies the Service Account ID associated with the end side of a network.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAccountID"]/ValueText
From Service Address ID	Identifies the Service Address ID for the starting point of a network.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="FromServiceAddressID"]/ValueText

Table A-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 Address ID	Identifies the Service Address ID for the ending point of a network.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServiceAddressID"]/ValueText
To Service Point ID	References a dummy asset record that represents the access point to which the starting side of a network service is connected on the subscriber's premises.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/*CommsOrder*LineSpecificationGroup/SpecificationGroup[./name="ExtensibleAttributes"]/Specification[./name="ToServicePointID"]/ValueText
Service Point	References a dummy asset record that represents the access point to which this service is connected on the subscriber'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.	*CommsOrder*EBO/*CommsOrder*Line/ServicePointCode
Promotion Description	Provides short description that appears on the invoice.	NA	Yes	Yes	None	*CommsOrder*EBO/*CommsOrder*Line/ItemReference/Description This is Promotion Description used for display purposes on subscriber invoice

Table A-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 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 subscriber accounts.	NA	Yes	Yes	Can be populated as part of order capture process or during fulfillment, but before interface an order to billing.	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ItemInstance /Identification/ID
Balance Bundle Identification	Identifies the Balance Bundle to which a service instance belongs.	NA	NA	None	Not Used by Oracle AIA for Communications.	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ BalanceBundleIdentification/BusinessComponentID
Line Description	Provides additional description for an order line. For example, to indicate that a charge is being applied for a penalty.	NA	No	None	Not used by Oracle AIA for Communications.	*CommsOrder*EBO/ *CommsOrder*Line/ Description
Service Length	Indicates requested service length in Service Length Unit of Measure.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ ServiceTimePeriod/ Duration
Service Length Unit of Measure	Indicates the service length unit of measure.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ ServiceTimePeriod/ Duration
Fulfillment Mode	Designates compensation operations for Initiate Billing. May be used in the future to provide explicit revision operations at the line level.	DO, NOOP,REDO,UNDO	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ FulfillmentModeCode
Product Name	Provides the name of the product.	NA	NA	NA	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/Name

Table A-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.	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ FulfillmentCompositionTypeCode
Product Type	Classifies products into Products, Discounts, Bundles, Promotion (Offer), etc.	Product, Offer, Bundle	No	None	Used part of fulfillment to determine the order lines Subject Type, which drives the mapping to Product Specifications.	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/TypeCode
Billing Type	Classifies products for Billing into Service Bundles, Subscriptions, Items, Discounts.	Service Bundle, Subscription, Item, Discount	No	None	Used with Product Type.	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ ClassificationCode [listID="BillingProductTypeCode"]
Billing Service Type	Specifies the service type so that when a corresponding product is created in Billing, it is associated to the specified service.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ ClassificationCode [listID="PermittedTypeCode"]

Table A-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.	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ ServiceIndicator
Vendor	Identifies the vendor supplying the product when the product is supplied by a third-party.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ SupplierPartyReference/ PartyIdentification/ID
Vendor Part Number	Identifies the product part number to the vendor.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ ItemIdentification/ SupplierItemID
Fulfillment Item Code	Uniquely identifies the mapping of an Order Line Subject to a Product Specification.	1) Null 2) A unique code that identifies the Product Spec to OM	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ ClassificationCode [listID = "FulfillmentItemCode"]
Item Class Name	Determines business classification of a product.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ PrimaryClassificationCode
Success Dependency	Declares if all order lines of a bundle or offer must fulfill successfully or else the whole bundle or offer fails (all or none).	Default,All Or None	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ FulfillmentSuccessCode

Table A-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.	*CommsOrder*EBO/ *CommsOrder*Line/ StartBillingOnFirstServiceUsageIndicator. We have added BillingStartCode to ItemReference, if this requirement is at the item/ itemReference level and not line level then BillingStartCode from ItemReference is necessary.
Smart Part Number	Automatically generated based on a predefined scheme. Mainly, drives dynamic product configuration/ pricing rules in CRM. The billing system may use it to dynamically derive a price/ discount value.	NA	Yes	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ AlternateObjectKey [ContextID=SmartPartNumber]
Network Product Flag	Indicates if this is a network product, which helps determine which user-defined attributes to expect.	True, False	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ NetworkIndicator
Network Element Type	Indicates if this network product represents a node, a connection, or a network.	NA	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ ItemReference/ NetworkItemTypeCode
Charge Frequency Code	Indicates charge frequency unit of measure, for example, monthly, quarterly, yearly.	NA	NA	NA	None	*CommsOrder*EBO/ *CommsOrder*>Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ ChargeFrequencyCode
List Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/TypeCode

Table A-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	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/UnitListPrice/ Amount
Sale Price Type	Identifies price type.	One-Time, Recurring, Usage	No	None	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/TypeCode
Sale Price	Identifies net price of the item.	NA	Yes	Yes	None	*CommsOrder*EBO/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ UnitSalePrice/Amount
Pricing Commit Type	Indicates whether the pricing is Committed or Dynamic.	Common/ Siebel values are true/ Dynamic, false/ Committed.	Yes	Yes	None	*CommsOrder*/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ DynamicPricingIndicator
Dynamic Discount Method	Indicates whether the discount is of type amount or percent.	Amount, Percent	Yes	Yes	NA	*CommsOrder*/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ DiscountMethodCode
Discount Percent	Indicates the percent by which the list price is discounted.	NA	Yes	Yes	NA	*CommsOrder*/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ DiscountPercent
Discount Amount	Indicates the amount by which the list price is discounted.	NA	Yes	Yes	NA	*CommsOrder*/ *CommsOrder*Line/ *CommsOrder*Schedule/ *CommsOrder*ScheduleCharge/Charge/ DiscountAmount

Table A-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
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 **AIAConfigurationProperties.xml** file is used regardless of values specified for this attribute.	*CommsOrder*EBM/ DataArea/ *CommsOrder*Line/ CommsOrderSchedule/ PriceListReference/ PriceListIdentification/ID
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.	*CommsOrder*EBO/ *CommsOrder*Line/ ServiceFamily
Technical Inventory ID	Correlates assets across technical inventory systems.	NA	No	No	Set by service and resource management system.	*CommsOrder*EBO/ *CommsOrder*Line/ InstalledProductReference /TechnicalInventoryId
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA Name	*CommsOrder*/ *CommsOrder*Line/ ItemReference/ SpecificationGroup[name= "ExtensibleAttributes"]/ Specification/Name
User Defined Attributes	Indicates attribute is common across all Specification components.	Add, Update, Delete	Yes	None	UDA Action Code (Expected to change to a Service Action Code element to allow additional value NONE).	*CommsOrder*/ *CommsOrder*Line/ ItemReference/ SpecificationGroup[name= "ExtensibleAttributes"]/ Specification[name="<OrderLine.XA.Attribute>"]/ @actionCode
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	Has Previous LIC Value	UDA language-independent code Value	*CommsOrder*/ *CommsOrder*Line/ ItemReference/ SpecificationGroup[name= "ExtensibleAttributes"]/ Specification[name="<OrderLine.XA.Attribute>"]/ Value

Table A-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.	String, Date, Number	Yes	None	UDA Data Type	*CommsOrder*/ Prior*CommsOrder*/ *CommsOrder*Line/ ItemReference/ SpecificationGroup[name= "ExtensibleAttributes"]/ Specification[name="<OrderLine.XA.Attribute"]/ DataTypeCode
User Defined Attributes	Indicates attribute is common across all Specification components.	NA	Yes	None	UDA language-independent code Prior Value.	*CommsOrder*/ Prior*CommsOrder*/ *CommsOrder*Line/ ItemReference/ SpecificationGroup[name= "ExtensibleAttributes"]/ Specification[name="<OrderLine.XA.Attribute>"]/ Value

B

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 flow.

How Dates are Set in BRM

Table B-1 defines the terms used in Table B-2 to explain about how dates are set in BRM.

Table B-1 Billing Date Mapping Terminology

Term	Abbreviation	Definition
Order Datetime	ODT	The date that the order was placed by the 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 subscriber; it is captured on the order in Siebel CRM. It is also known as **Due Date**.
Actual Delivery Datetime	RDDT	The actual delivery date time; it is supplied by the order management system that fulfills the order, and is updated in Siebel CRM.
Purchase Start Date	-	The date as of which BRM applies purchase fees.
Cycle Start Date	-	The date as of which BRM applies cycle fees.
Usage Start Date	-	The date as of which BRM rates usage and applies usage fees.

Table B-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
Subscriber data creation	Oracle AIA uses order date as the effective date for subscriber data creation	PCM_OP_CUST_COMMIT_CUSTOMER	Pass Order Date coming from Siebel CRM.
CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service	CommunicationsBillingEBSV1.ProcessFulfillmentOrderBilling service

Table B-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 Promotion Purchase	Oracle AIA passes the Purchase Date as the Valid From date for bundle purchase (that represents purchased promotion). If Purchase Date is null, then it passes Requested Delivery Date and if that is null, it passes no date and BRM defaults current date.	PCM_OP_SUBSCRIPTION_SET_BUNDLE	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Purchase Date to Actual Delivery Datetime.
Single Phase Billing - Billing Fulfillment	If all three of the billing dates are set, then Oracle AIA uses Order Date as Effective Date, and sets respective offset (Order Date - respective billing date). Billing dates are: Purchase Date, Cycle Start Date and Usage Start Date. If any of the three billing dates are not set, then Oracle AIA passes no dates to BRM and lets BRM default the Purchase, Cycle Start and Usage Start dates. For purchase of a service bundle, this check for existence of billing dates applies to ALL products and discounts included in the service bundle.	PCM_OP_CUST_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Purchase Date, Start Cycle, and Start Usage to Actual Delivery Datetime to explicitly control setting of billing dates.
Single Phase Billing - Billing Fulfillment Time Based Account or Service level Subscription Product/Discount Purchase	In addition to setting of billing dates as described previously, if Service End Date is passed, then Oracle AIA additionally sets the Purchase, Cycle and Usage end date offsets (difference between the respective billing date and service end date). If any of the billing dates (Purchase, Cycle, or Usage start) are not set then Oracle AIA uses the Order Date to calculate the Purchase, Cycle and Usage end date offsets (difference between the Order Date and Service End Date).	PCM_OP_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Populate Purchase, Cycle and Usage Start dates (this is required for enabling time-based offerings. Calculate the Service End Date based on TBO attributes as documented in TBO section. See "About Time-Based Offerings on Orders" for more information.

Table B-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 Time Based Account or service-level Subscription Product/Discount Update (of end date due to promotion upgrade or downgrade, or other pricing changes)	If Service End Date is passed (and prior value is set), then Oracle AIA uses that to reset the Purchase, Cycle and Usage end dates.	PCM_OP_SUBSCRIPTION_SET_PRODINFO PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO	Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate prior value to trigger update. See "About Time-Based Offerings on Orders" for more information about TBO attributes.
Single Phase Billing - Billing Fulfillment Promotion Cancellation	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.	PCM_OP_SUBSCRIPTION_SET_BUNDLE	Pass Order Date and Requested Delivery Date coming from Siebel CRM. Set Actual Delivery Datetime.
Single Phase Billing - Billing Fulfillment Application of Promotion Penalties or MACD One Time Charge (Suspend, Resume, Disconnect, or Move charge) **Note:** These are processed only in Billing Fulfillment	If ADDT (on service bundle line) is passed, Oracle AIA sets the effective date to ADDT (from service bundle line). If ADDT (on service bundle line) is not passed, then Oracle AIA lets BRM default the purchase date (to current date).	PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Set Actual Delivery Datetime.
Single Phase Billing - Billing Fulfillment Suspend, Resume, or Cancellation of Service or account-level or service-level Subscription Product/ Discount	If ADDT is passed, then Oracle AIA uses that as the effective date for the operation, else it lets BRM default the date (to current date).	PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS PCM_OP_CUST_SET_STATUS	Set Actual Delivery DateTime.
Two-Phase Billing - Billing Initiation Promotion Purchase	Oracle AIA passes Purchase Date as the Valid From date. If Purchase Date is null, then Oracle AIA passes Requested Delivery Date and if that is null, Oracle AIA passes no date and BRM defaults current date.	PCM_OP_SUBSCRIPTION_SET_BUNDLE	Pass Order Date and Requested Delivery Date coming in from Siebel CRM. Set Purchase Date to Expected Delivery Date.

Table B-2 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
Two Phase Billing - Billing Initiation Account-level or service-level Item Type Product Purchase	Oracle AIA validates that Purchase Date is set to future (based on value of configuration property - FutureTimeThreshold). Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date). Billing Dates are - Purchase Date, Cycle Start Date and Usage Start Date.	PCM_OP_CUST_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Pass Order Date coming in from Siebel CRM. Set Purchase, Cycle, and Usage Date to Future (one year out to match default threshold).
Two Phase Billing - Billing Initiation Account-level or service-level Subscription Type Product/Discount Purchase	Oracle AIA validates that Cycle Start Date is set to future (based on value of configuration property - FutureTimeThresholdForBillingDates). Uses Order Date as Effective Date, and sets respective offset for each billing date (calculated as Order Date - respective billing date). Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.	PCM_OP_CUST_MODIFY_CUSTOMER PCM_OP_SUBSCRIPTION_PURCHASE_DEAL	Pass Order Date coming in from Siebel CRM. To support validation mode, set all three billing dates to the future (one year out to match default threshold). To support latency mode, set Purchase and Usage Start Date to Current, but set Cycle Start Date to Future (one year out to match threshold).
Two-Phase Billing - Billing Fulfillment Promotion Purchase	Oracle AIA uses purchase date to reset Valid From date.	PCM_OP_SUBSCRIPTION_SET_BUNDLE	If purchase date had been set to Expected Delivery Date in Billing Initiation, reset purchase date to Actual Delivery Date.
Two Phase Billing - Billing Fulfillment. Account-level or service-level Item Type Product Purchase.	If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset. Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.	PCM_OP_SUBSCRIPTION_SET_PRODINFO	Reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update).
Two Phase Billing - Billing Fulfillment Account-level or service-level Subscription Type Product/Discount Purchase	If prior values are set, Oracle AIA resets respective billing date by passing in absolute values for each billing date that must be reset. Billing Dates are - Purchase Date, Cycle Start Date, and Usage Start Date.	PCM_OP_SUBSCRIPTION_SET_PRODINFO PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO	To support validation mode, reset all three billing dates to Actual Delivery Datetime (set prior values to trigger update). To support latency mode, reset Cycle date to Actual Delivery Datetime (set prior value to trigger update).

Table B-2 (Cont.) Mapping Billing Dates

Operation Being Performed in BRM	Dates Set by Oracle AIA When the Service is Called	BRM Opcodes Invoked	Expectations of the Order Management System
Two Phase Billing - Billing Fulfillment Time-Based account-level or service-level Subscription Product/ Discount Purchase	If Service End Date is passed, then Oracle AIA uses that to set the Purchase, Cycle, and Usage end dates for products/ discounts purchased.	PCM_OP_SUBSCRIPTION_SET_PRODINFO PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO	Calculate the Service End Date based on TBO attributes as documented in TBO section. Populate Purchase, Cycle, and Usage start dates. See "About Time-Based Offerings on Orders" for more information about TBO attributes.

C

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 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 **Canceled** in Siebel CRM and **Disconnected** in BRM.

Table C-1 shows which line actions are supported for marketing bundles.

Table C-1 MACD Line Actions Supported for Marketing Bundles

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	N/A	N/A	Yes	Yes Cross-reference tables updated to reflect new Siebel Customer Relationship Management (Siebel CRM) asset.	Ignored other than to determine original BRM asset.

Table C-2 shows which line actions are supported for service bundles.

Table C-2 MACD Line Actions Supported for Service Bundles

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	Yes	Yes	Yes	Yes Same as Update with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

Table C-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 C-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 C-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.

Table C-4 shows which line actions are supported for service-level billing subscription products.

Table C-4 MACD Line Actions Supported for Service-Level Billing Subscription Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	Yes	N/A Unsupported by Siebel CRM and BRM.	N/A Unsupported by Siebel CRM and BRM.	Yes	Yes Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

Table C-5 shows which line actions are supported for service-level billing discount products.

Table C-5 MACD Line Actions Supported for Service-Level Billing Discount Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes	Yes	N/A Unsupported by Siebel CRM and BRM.	N/A Unsupported by Siebel CRM and BRM.	Yes	Yes Same as UPDATE with communicating changes to line attributes.	Ignored other than to determine original BRM asset.

Table C-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 C-6 MACD Line Actions Supported for Service-Level Billing Item Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	N/A	N/A	N/A	N/A If a line is billing-initiated and a revision is processed, pricing information and billing dates can change.	N/A	N/A

MACD Line Actions Supported for Account-Level Products

This section shows which line actions are supported for account-level products.

Table C-7 shows which line actions are supported for account-level billing subscription products.

Table C-7 MACD Line Actions Supported for Account-Level Billing Subscription Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	Yes	Yes	Yes	Yes	Ignored	Ignored

Table C-8 shows which line actions are supported for account-level billing discounts.

Table C-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 C-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 C-9 MACD Line Actions Supported for Account-Level Billing Item Products

Add	Delete	Suspend	Resume	Update	Move-Add	Move-Delete
Yes Can communicate price or discount override as part of this action.	NA	NA	NA	NA 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 the changes made to product attributes that 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 with 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 About 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

The integration does not create a purchased product instance in BRM for service-level billing item products like one-time charges. Therefore, 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 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.

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 re-points 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

The integration does not create a purchased product instance in BRM for account-level billing item products like penalties,. Therefore, 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.

D

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 D-1 defines the abbreviations used in the examples.

Table D-1 Abbreviations Used in the Examples

Abbreviation	Description
SA	Service account: the child account
BA	Billing account: the parent account

Table D-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 **/billinfo** object: a **/billinfo** 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 there services in the child account.

Table D-2 shows the base order. A separate billing profile is assigned to each service.

Table D-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 D-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 D-1 Result of Base Order for Services with Separate Billing Profiles

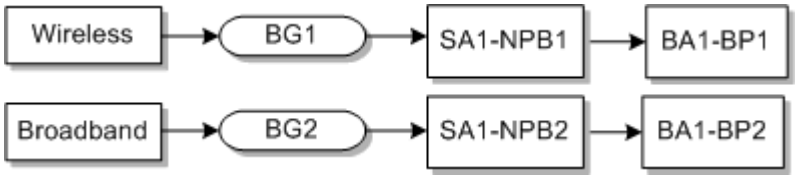


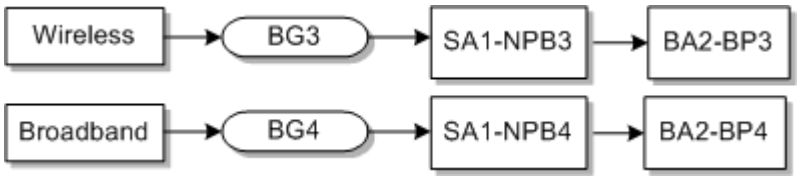
Table D-3 shows the change order to change the paying account for the services while maintaining separate billing profiles.

Table D-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 D-2 shows the result of sending the change order to billing.

Figure D-2 Result of Change Order to Change the Paying Account with Separate Billing Profiles



Alternative: Single Billing Profile

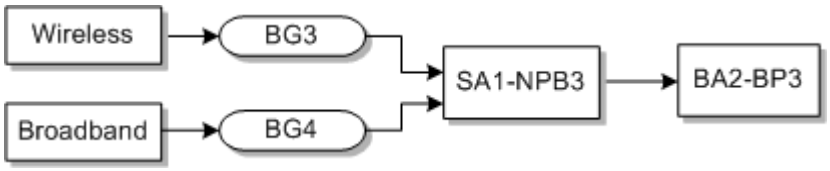
Alternatively, the change order can change the paying account for the services and assign the same billing profile to both. Table D-4 shows this alternative order.

Table D-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 D-3 shows the results of sending the alternative order to billing.

Figure D-3 Result of Alternative Change Order to Change the Paying Account with a Single Billing Profile



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 D-5 shows the base order. The same billing profile is assigned to both services.

Table D-5 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 D-4 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 D-4 Result of Base Order for Services with a Single Billing Profile

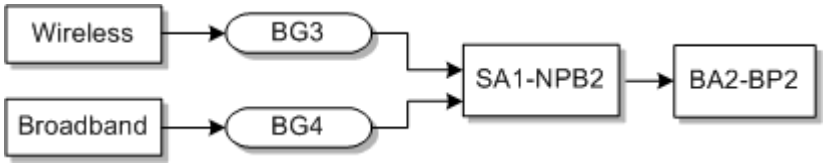
Table D-6 shows the change order to change the paying account for the service son the child account while maintaining a single billing profile for the services.

Table D-6 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 D-5 shows the result of sending the change order to billing.

Figure D-5 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 D-7 shows this alternative order.

Table D-7 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 D-6 shows the results of sending the alternative order to billing.

Figure D-6 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 D-8 shows the base order. A separate billing profile is assigned to each service.

Table D-8 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 D-7 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 D-7 Result of Base Order for Services in Multiple Child Accounts

Table D-9 shows the change order to change the paying account for the services on both child accounts at once while maintaining separate billing profiles.

Table D-9 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 D-8 shows the results of sending the change order to billing.

Figure D-8 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 D-10 shows this alternative order.

Table D-10 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 D-6 shows the result of sending the alternative order to billing.

Figure D-9 Result of Alternative Order to Change the Paying Account for Multiple Child Accounts with a Single Billing Profile

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 D-11 shows the base order. The same billing profile is assigned to both services.

Table D-11 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 D-10 shows the result of sending the base order to billing. The services are tracked together under the default account-level balance group.

Figure D-10 Result of Base Order for Services in a Child Account

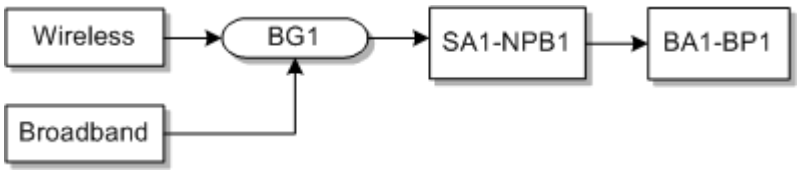


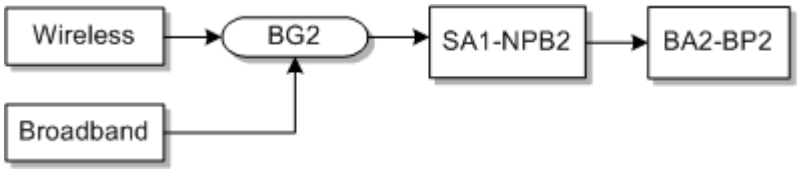
Table D-12 shows the change order to change the paying parent for the child account.

Table D-12 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 D-11 shows the result of sending the change order to billing.

Figure D-11 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 D-13 shows the base order. A separate billing profile is assigned to each child account.

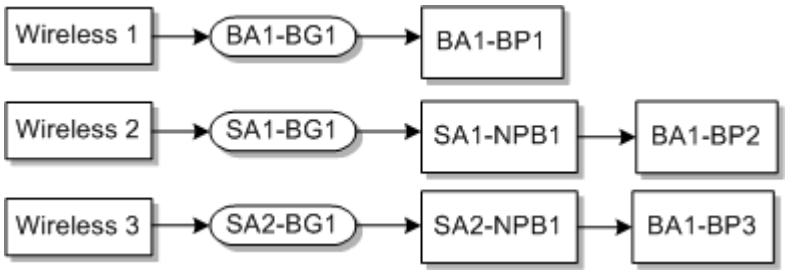
> **Note**
>
> When service-level balance groups are disabled, a parent account can only use multiple billing profiles to pay for services under separate child accounts. A single child account cannot have multiple billing profiles. See Problems When Integrating Separate Billing Profiles on the Same Account for an example of this restriction.

Table D-13 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 D-12 shows the result of sending the base order to billing. Each service is tracked under the default account-level balance group for its account.

Figure D-12 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 D-14 shows the change order to change the paying parent for both child accounts using one billing profile under the new parent.

Table D-14 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 D-13 shows the result of sending the change order to billing.

Figure D-13 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 D-15 shows the base order. A separate billing profile is assigned to each service.

Table D-15 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 D-15, the integration uses billing account and billing profile on the first service purchased for all the remaining services on the order. The billing account and billing profile specified on the second service are ignored. The result of sending the order to billing is the same as in Table D-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 D-16 shows the revision order to resolve the failure or data mismatch by assigning the broadband service to BP1.

Table D-16 Resubmitted Order

Action	Service Number	Service Account	Billing Account	Billing Profile
ADD	Wireless	SA1	BA1	BA1-BP1
ADD	Broadband	SA1	BA1	**BA1-BP1**

After the order is processed to billing, the result is the same as in Table D-16.

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.

E

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.

F

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 flows).

For feature-specific expectations, see the respective flow feature sections (for example, see Order to Payment Business Process 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl 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 UpdateSalesOrderSiebelCommsProvABCSImpl 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 (UpdateSalesOrderOSMCFSCommsJMSConsumer) picks up the message and sends it to UpdateSalesOrderSiebelCommsProvABCSImpl 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 About the Update Sales Order 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 ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl for information on how to achieve this.

G

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 G-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 G-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	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']) Order of Processing: FIFO (First in First Out). Composite Name: UpdateSalesOrderOSMCFSCommsJMSConsumer.	**Note:** The consumer in the Create Trouble Ticket for Order Fallout flow is only a sample. The resequencer in this flow ensures that multiple updates for the same order are processed in the right sequence.

Table G-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 subscriber 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. concat($in.SyncCustomerPartyListEBM/ns0:SyncCustomerPartyListEBM/ns0:DataArea/ns0:SyncCustomerPartyList/ns0:CustomerPartyAccount/corecom:Identification/corecom:ApplicationObjectKey/corecom:ID[@schemeID='AccountId'], $in.SyncCustomerPartyListEBM/ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ID) Order of Processing: FIFO (First in First Out). Composite Name: CommunicationsCustomerPartyEBSV2Resequencer.	The resequencer in this flow ensures that the solution can successfully handle processing of concurrent orders for the same subscriber.
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 ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer to implement custom sequencing.	NA
AIA_UPDBO_IN_JMSQ	Order flow from AIA (from billing) to OSM.	Set by ProcessFulfillmentOrderBillingResponseOSMCFSCommsJMSProducer	None (Onus is on OSM).	NA
AIA_UPDCUST_IN_JMSQ	Response of the subscriber creation in billing from AIA to OSM.	Set by ProcessFOBillingAccountListRespOSMCFSCommsJMSProducer	None (Onus is on OSM).	NA

Table G-1 (Cont.) Queues and Flows Enabled for Sequencing

Oracle AIA Queue	Flow	JMS Priority	Sequencing Criteria	Comments
AIA_CRTFO_OUT_JMSQ	Create Fulfillment Order flow from OSM to Oracle AIA for the provisioning system.	Set by OSM.	None as delivered. Subscriber 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. Subscriber 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.

H

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 H-1 is a sample EBMHeader section.

Example H-1 Sample EBMHeader Section

```
<EBMHeader>
<Sender>
<ObjectCrossReference>
 <SenderObjectIdentification>
<BusinessComponentID> OrderId</BusinenessComponentID>
             <ID> Order# (if any)</ID>
<ApplicationObjectKey>
<ID schemeID="SALESORDER_ID"schemeAgencyID="SEBL_01">SalesOrderID</ID>
</ApplicationObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_ID " schemeAgencyID="COMMON">SalesOrderCommonID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_NUMBER" schemeAgenyId="SEBL_01">SalesOrderNumber</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgenyId="SEBL_01">SalesOrderRevision</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgenyId="COMMON">CommonAccountID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTID" schemeAgenyId="SEBL_01">Siebel Account ID</ID>
</AlternateObjectKey>
<AlternateObjectKey>
<ID schemeID="CUSTOMERPARTY_ACCOUNTNAME" schemeAgenyId="SEBL_01">Account Name</ID>
</AlternateObjectKey>
</SenderObjectIdentification>
</ObjectCrossReference>
</Sender>
</EBMHeader>
```

Only the underlined elements are required for the SalesOrder EBM.

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 in an xsd. Any field in the fault message is defined as:

Fault/FaultNotification/FaultMessage/ApplicationFaultData

The BPEL processes are expected to construct a variable of element type ApplicationFaultData defined in this xsd as: http://{<*httphostname*>}:{<*http portname*>}/AIAComponents/PIPS/Communications/Schemas/OrderFailureData.xsd

The fields defined in the xsd and how they must be used are listed as below.

- ApplicationFaultData / OrderFailureData / OrderID

  BusinessComponentID - SalesOrder / Provisioning Order / Fulfillment Order / Fulfillment Billing Order

  ID - SalesOrder # / Provisioning Order # / Fulfillment Order # / Fulfillment Billing Order # (If available)

ApplicationObjectKey - If available

AlternateObjectKey - SALESORDER_ID

AlternateObjectKey - SALESORDER_NUMBER

AlternateObjectKey - SALESORDER_REVISION

AlternateObjectKey - FULFILLMENTSYSTEM_ID

Example H-2 is a sample definition.

- ApplicationFaultData / OrderFailureData / AccountID

  BusinessComponentID - CommonAccountID

  ID - Account Name

  ApplicationObjectKey - Siebel AccountID (required only with SalesOrder EBM)

  Example H-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 H-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 H-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 H-2 Sample Definition 1

```
<BusinessComponentID>  Order ID   </BusinessComponentID>
         <ID> Order# (if any)</ID>
         <ApplicationObjectKey>
<ID schemeID="SALESORDER_ID"schemeAgencyID="SEBL_01">SalesOrderID</ID>
         </ApplicationObjectKey>
        <AlternateObjectKey>
  <ID schemeID="SALESORDER_ID"           schemeAgencyID="COMMON">SalesOrderCommonID</ID>
         </AlternateObjectKey>
         <AlternateObjectKey>
<ID schemeID="SALESORDER_NUMBER" schemeAgenyId="SEBL_01">SalesOrderNumber</ID>
          </AlternateObjectKey>
          <AlternateObjectKey>
<ID schemeID="SALESORDER_REVISION" schemeAgenyId="SEBL_01">SalesOrderRevision</ID>
          </AlternateObjectKey>
          <AlternateObjectKey>
<ID schemeID="FULFILLMENTSYSTEM_ID " schemeAgenyId="FulfillmentSystemAppID">OrderID in
the Fulfillment System</ID>
          </AlternateObjectKey>
```

Example H-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 H-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 H-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* for more information on extending error handling.

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*.

J

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 J-1 shows an example of the MIGRATED_OBJECTS_T table data.

Table J-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 J-2. The table shows the cross-reference mapping between the Oracle AIA cross-reference identifier and BRM cross-reference object types.

Table J-2 Cross-Reference Mapping Between Oracle AIA and BRM

AIA Cross-Reference	BRM Cross-Reference
CUSTOMERPARTY_CONTACTID	**/account** POID
CUSTOMERPARTY_BILLPROFILEID	**/billinfo** POID
CUSTOMERPARTY_PAYPROFILEID	**/payinfo** POID
CUSTOMERPARTY_ADDRESSID	**/account** POID or **/payinfo** POID
CUSTOMERPARTY_CONTACTID	**/account** POID or **/payinfo** POID
CUSTOMERPARTY_DEFAULTBALANCEGROUPID	**/balance_group** POID This is the default account level balance group.
INSTALLEDPRODUCT_ID	**/purchased_product** POID
INSTALLEDPRODUCT_ID	**/purchased_discount** POID
INSTALLEDPRODUCT_ID	**/service/***object* POID Where *object* is the service object for the account.
INSTALLEDPRODUCT_ID	**/purchased_bundle** POID
INSTALLEDPRODUCT_ID	**/profile/***object* POID Where *object* is the profile object for the account.
INSTALLEDPRODUCT_ID	**/group/sharing/***object* POID Where *object* is the type of sharing group. Possible values are **discounts**, **charges**, and **profiles**.

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.

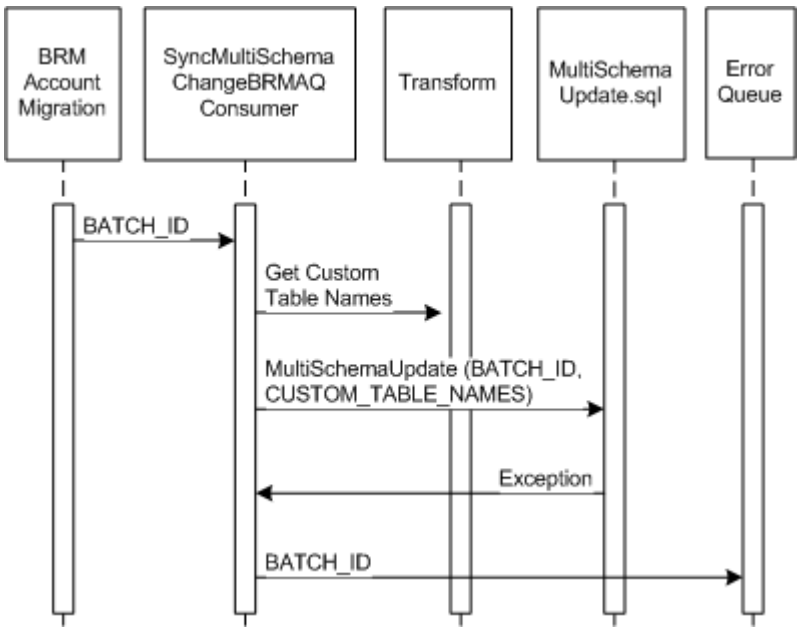
b. Connects to MIGRATED_OBJECTS_T cross-reference table using a database link.

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 J-2 illustrates the flow when updating cross-references in a multischema migration.

Figure J-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 Order to Cash business process, 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_AIA_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 J-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 Lifecycle 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.