Oracle® Application Integration Architecture

Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide

Release 11.4

E54228-02

August 2015

Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide, Release 11.4

E54228-02

Contents

4 Invoice Integration Flow

5 Service Usage Integration Flow

6 Payment Integration Flow

7 Adjustment Integration Flow

8 Understanding the Process Integration for Collection Management

Part II Configuring the Delivered Integrations

9 Configuring the Process Integration for Billing Management

10 Configuring the Process Integration for Collections Management

A Mapping Siebel Billing Management UI Elements to BRM Customer Center

B Composite Application Validation System Changes

C Reintroducing Enterprise Business Services

Preface

This document describes how to implement and use the Agent Assisted Billing Care (AABC) Pre-built integration.

Audience

This document is intended for customer service representatives, billing and pricing administrators, and other individuals who are responsible for configuring, managing and maintaining AIA Communications pre-built integrations.

Downloading Oracle Documentation

Product documentation is available from Oracle Help Center:

`http://docs.oracle.com`

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

`http://edelivery.oracle.com`

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

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

- Product guides
- Alerts
- Troubleshooting details
- FAQs

- Patches
- Community links

To see the Information Center for the Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care, see My Oracle Support note 1392661.2 at:

`http://support.oracle.com/epmos/faces/DocumentDisplay?id=1392661.2`

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

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

1

Siebel CRM Integration Pack for Oracle Communications BRM: Agent Assisted Billing Care Overview

This chapter provides an overview of the Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care.

About Agent Assisted Billing Care

The Agent Assisted Billing Care integrates the following business processes between Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM):

- About the Billing Management Process Integration
- About the Customer Management Process Integration
- About the Collections Management Process Integration

About the Billing Management Process Integration

The billing management process integration enables you to access all customer-related billing information by using the Siebel CRM interface, including data that is created and maintained within the BRM system. It includes support for the following key processes that can be performed from Siebel CRM:

- Real-time view of accounts and customer balance summaries, balance groups, and balance details
- Real-time display of unbilled usage information
- View of invoice details
- View of payment and adjustment history
- Invoice payment capture
- Adjustment request capture

See "Understanding the Process Integrations" for more information.

About the Customer Management Process Integration

The customer management process integration enables the synchronization of customer information between Siebel CRM and BRM. It provides two integration flows:

- Create or synchronize new customer accounts, which interface customers to BRM (performed during the order management processing flow).
- Update a customer account integration flow, which updates account profile information (such as address, name, and contact) from Siebel CRM to BRM.

See *Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management* for more information.

About the Collections Management Process Integration

The collections management process integration provides synchronization of collection actions (based on the specified collection scenarios) defined in BRM to Siebel CRM, and administration of these collection actions and credit alerts.

See "Understanding the Process Integration for Collection Management" for more information.

Part I

Understanding the Process Integrations

Part I includes the following chapters:

2

Understanding the Process Integration for Billing Management

This chapter describes the process integration for billing management, provides the billing management business process flow, and discusses the solution assumptions and constraints.

Process Integration for Billing Management Overview

The Agent Assisted Billing Care pre-built integration provides the following process integrations:

- Billing management
- Customer management
- Collections management

The Agent Assisted Billing Care pre-built integration also includes required touch points between Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM).

The Order to Cash pre-built integration is required for the process integration for billing management.

The process integration for billing management enables a customer service representative (CSR) to retrieve account balances, invoices, and unbilled data at a header, summary, and detail level. The process integration for billing management also let a CSR adjust and pay invoices.

After a service is activated, usage events (for example, a phone call, text message, or a data session) are sent from the communications network to the billing system. These events are rated, bills are generated, and then bills are sent to customers.

The service cycle of the process integration for billing management starts when customers call to question the content of their bills, query on unbilled usage, make payments, or file disputes. Billing information must be sent from BRM to Siebel CRM so that CSRs can respond to billing questions. Payment information and bill dispute requests must then be sent from Siebel CRM to BRM.

Most of the billing integration touch points do not replicate the billing data in Siebel CRM. Instead, the integration retrieves the billing data on demand from BRM and displays it within Siebel CRM.

For this integration, Oracle supports Siebel CRM and BRM as the participating applications. You may use routing rules to retrieve billing data from multiple instances of BRM.

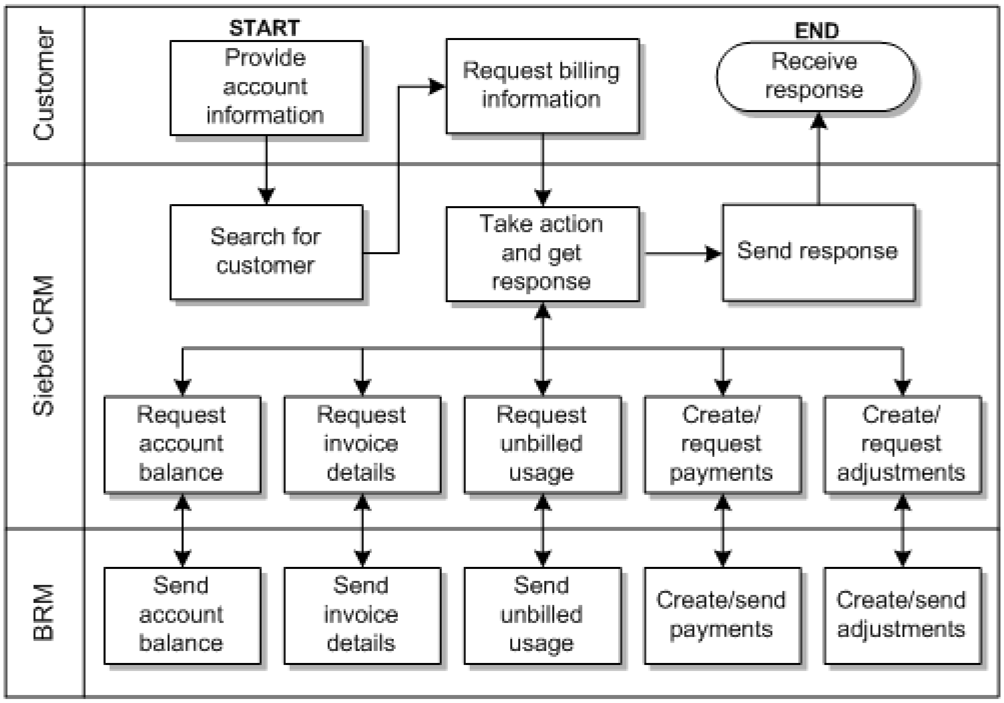
The process integration for billing management consists of the following integration flows and their respective integration scenarios:

- The Account Balance integration flow provides the QueryBalanceGroupList, QueryBalanceGroupServices, QueryBalanceSummary, and QueryBalanceDetails integration scenarios.
- The Invoice integration flow provides the QueryInvoiceList, QueryPDFInvoice, QueryInvoice, QueryInvoiceUsageAllocation, SearchInvoiceUsageAllocation, and QueryInvoiceUsageAllocationResource integration scenarios.
- The Service Usage integration flow provides the QueryServiceUsage, QueryServiceUsageAllocation, SearchServiceUsageAllocationResource, and QueryServiceUsageAllocationsResource integration scenarios.
- The Payment integration flow provides the CreateReceivedPayment, QueryReceivedPaymentList, and SearchPayment integration scenarios.
- The Adjustment integration flow provides the QueryAccountBalanceAdjustment and CreateAccountBalanceAdjustment integration scenarios.

Billing Management Business Process Flow

Figure 2–1 illustrates the overall flow for the process integration for billing management.

Figure 2–1 Business Process Flow for the Process Integration for Billing Management



Solution Assumptions and Constraints

These are the assumptions and constraints for the process integration for billing management.

- One billing profile in Siebel CRM is associated with a single billing system.

- Billing data is not replicated in Siebel CRM. Rather, it is retrieved on demand from the billing system.
- In cases where the deployment topology has multiple billing systems, each Siebel billing profile should map 1:1 with a billing system (assumption stated in #1). This is because the integration does not have any provisions to converge or consolidate billing data between multiple billing systems. If this premise is ignored, the solution cannot guarantee fetching the billing data consistently from the same billing system.

  See the discussion of configuring multiple BRM instances in *Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management* for more information.
- Monetary and nonmonetary adjustments within a single adjustment request from Siebel CRM are not possible.
- A single adjustment request can have adjustments from only one level and not across multiple levels such as header, item, and event.
- Adjustment requests always originate in Siebel CRM and are sent to BRM for processing through the process integration.
- Adjustment requests that are absolute and percentage value are supported at bill/header and event level only. Adjustment request of only absolute value are supported at the item level.
- Invoice adjustment requests are applicable at header, item, and event levels.
- Account-level adjustments are not supported.
- Unbilled adjustments are applicable only at the event level.
- Payment validation occurs in the billing system.
- Create Payments is used for making one-time payments and supports credit card and automatic debit payment methods.
- Although the Invoice integration flow supports uploading batches of invoices from Oracle Business Intelligence Publisher (BI Publisher) into a central repository for viewing in Siebel CRM, if a batch upload fails, you cannot resubmit the entire batch. You must re-send each invoice individually.
- The invoice report job history information in BI Publisher is not intended for auditing purposes, such as reconciliation. The integration does not have comprehensive support for job execution reporting.

3

Account Balance Integration Flow

This chapter provides an overview of the account balance integration flows and discusses Siebel customer relationship management (Siebel CRM) and Oracle Communications Billing and Revenue Management (BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Account Balance Integration Flows Overview

The account balance integration between Siebel CRM and BRM supports the following integration flows:

- QueryBalanceSummary enables a customer service representative (CSR) to view an account billing profile balance summary in Siebel CRM.
- QueryBalanceGroupList enables a CSR to view the list of balance groups under an account billing profile.
- QueryBalanceDetails enables a CSR to view balance details under a balance group.
- QueryBalanceGroupServices enables a CSR to view services under a balance group.

The account balance integration enables the CSR to retrieve balance information from BRM. The CSR can obtain balance information at a summary or detail level. They can also query a list of balance groups for a selected billing profile.

After the CSR obtains the balance group information from BRM, they can navigate and query additional details, such as monetary and nonmonetary balances, credit limits, and validity dates.

When the CSR accesses the Siebel Billing Profile screen, a request is made from Siebel CRM to retrieve an account balance summary. When the CSR clicks the Balance Group tab, a call is made to fetch the list of balance groups. When the CSR clicks the view detail button for a balance group, the corresponding balance group detail data appears on the Balance Group detail applet.

With the add-on support for viewing account balance and other billing data for service accounts, the account balance integration now enables the CSR to retrieve balance and billing information for the service account (in addition to the billing account) from BRM.

From the Account Summary view in Siebel CRM, the CSR can retrieve both the account balance information of the billing (paying) account and the service (nonpaying) account. The click stream action determines which account billing information is retrieved from the billing system. By default there are the following three navigation paths available:

- From the Billing Account Summary page, navigate to the Siebel Billing Portal page for the billing account.

  If the intent is to see the billing account's billing data (account balance, bills, unbilled usage, adjustments and payments), then the user must use the Billing Profile link under the Billing Profile applet.

  In cases were a single billing profile is used to pay for both the billing account and service account(s) services, then the billing data that is displayed using this navigation option contains the consolidated billing information of account balance, bills, unbilled usage, adjustments and payments across all accounts that are tied to this billing profile.

  Alternatively, if different parent billing profiles are used to pay for self and other service accounts, then clicking a specific billing profile displays the billing data of account(s) that are tied to that billing profile.

- From the Billing Account Summary page, navigate to the Siebel Billing Portal page for one service account.

  If the intent is to see a particular service account's billing data then the user must use the Billing Profile link under the Billing Items applet.

  The user must identify the correct billing item that is mapped to the service account for which the billing data is to be seen. Billing data like account balance, bills, unbilled usage and adjustments that are specific to the selected service account can be viewed using this navigation option.

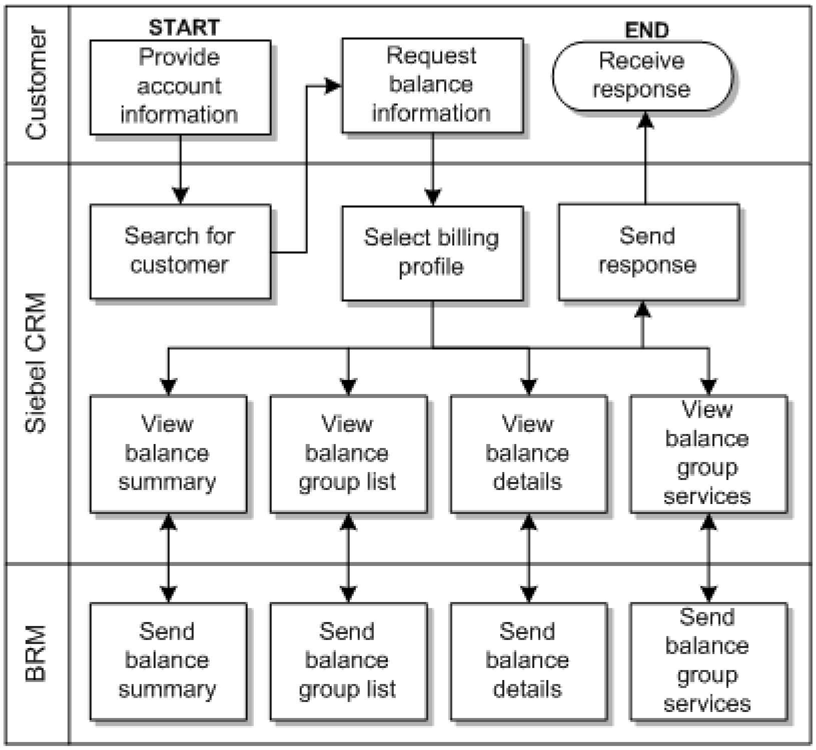
- From the Service Account Summary page, navigate to the Siebel Billing Portal page for the service account.

  If the intent is to see the billing data of the service account that is currently accessed in Siebel CRM, the user must use the Billing Profile link under the Installed Asset applet.

  Clicking on any of the Billing Profile links under this applet navigates the user to the service account's billing portal screen. Billing data like account balance, bills, unbilled usage, and adjustments that are specific to a service account can be viewed using this navigation option.

Figure 3–1 illustrates the account balance integration flow.

Figure 3–1 The Account Balance Integration Flow



To view account balance information for an account using the billing profile portal screen in Siebel CRM:

1. Navigate to the Accounts screen and query an account.
2. Scroll down to the Billing Profile applet for this account and click the Billing Profile **Name** link.

   You can now view the account's billing profile portal screen which has the applet with the balance summary for this account.
3. Click the **Balance Group** tab to view the list of balance groups under an account billing profile. Click **View Detail** to view the corresponding balance group detail on the Balance Detail applet.

See "Mapping Siebel Billing Management UI Elements to BRM Customer Center" for more information about the mapping of Siebel CRM elements to BRM elements.

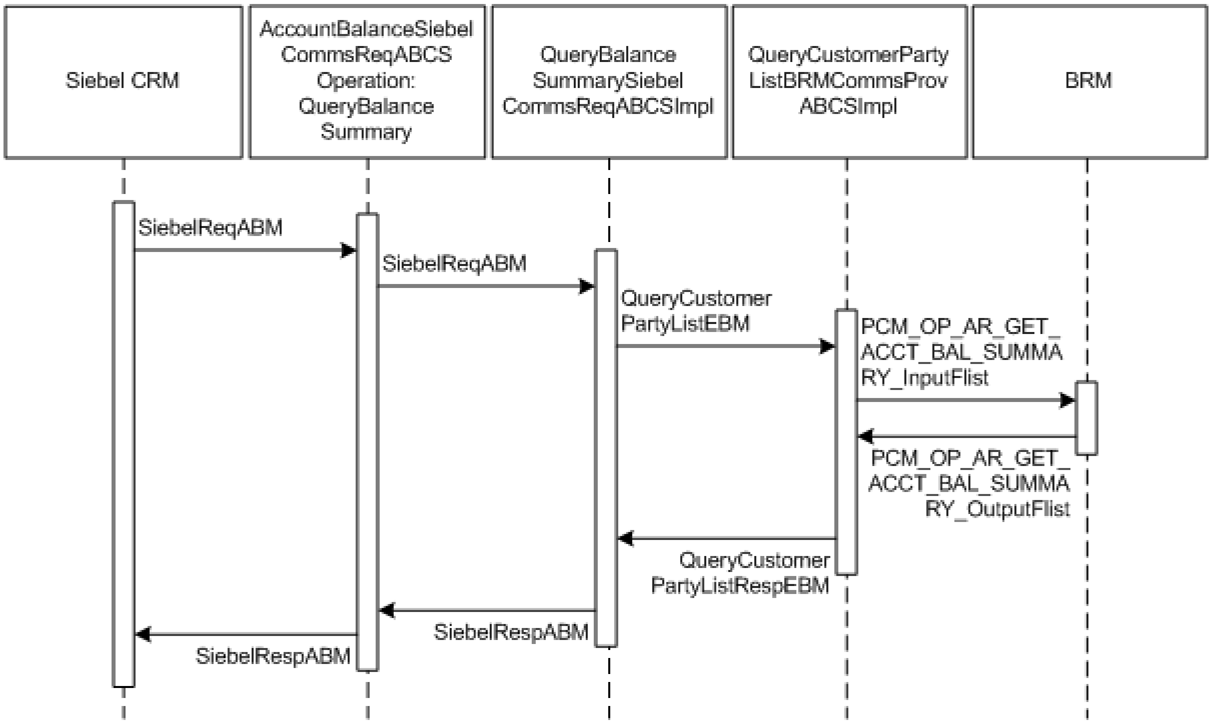
QueryBalanceSummary Integration Flow

This integration flow uses the following interfaces:

- AccountBalanceSiebelCommsReqABCS with operation QueryBalanceSummary
- QueryBalanceSummarySiebelCommsReqABCSImpl
- QueryCustomerPartyListBRMCommsProvABCSImpl

Figure 3–2 illustrates the QueryBalanceSummary integration scenario:

Figure 3–2 QueryBalanceSummary Integration Flow Sequence Diagram


Siebel CRM
AccountBalanceSiebelCommsReqABCS Operation: QueryBalance Summary
QueryBalance SummarySiebel CommsReqABCSImpl
QueryCustomerParty ListBRMCommsProv ABCSImpl
BRM
SiebelReqABM
SiebelReqABM
QueryCustomer PartyListEBM
PCM_OP_AR_GET_ ACCT_BAL_SUMMA RY_InputFlist
PCM_OP_AR_GET_ ACCT_BAL_SUMMA RY_OutputFlist
QueryCustomer PartyListRespEBM
SiebelRespABM
SiebelRespABM

When you initiate the QueryBalanceSummary process, the following events occur:

1. In Siebel CRM, a user navigates to the Accounts screen, queries an account, and clicks a Billing Profile for the account.

   This opens up the Billing Profile BRM screen, and a web service call is made to query the balance summary from the billing system.

2. Navigating to the Billing BRM screen invokes AccountBalanceSiebelCommsReqABCS web service with the QueryBalanceSummary method, which in turn calls the AccountBalanceSiebelCommsReqABCS with operation QueryBalanceSummary.

   AccountBalanceSiebelCommsReqABCS is a generic Siebel Account Balance interface service with several operations defined on the application business message (ABM).

3. Invoking AccountBalanceSiebelCommsReqABCS with operation QueryBalanceSummary routes the QueryBalanceSummaryReqMsg to the QueryBalanceSummarySiebelCommsReqABCSImpl.

4. The QueryBalanceSummarySiebelCommsReqABCSImpl first transforms the QueryBalanceSummaryReqMsg into QueryCustomerPartyListReqMsgEBM and routes the QueryCustomerPartyListReqMsg to the appropriate billing systems.

   As delivered, QueryCustomerPartyListReqMsg is routed to the QueryCustomerPartyListBRMCommsProvABCSImpl.

5. QueryCustomerPartyListBRMCommsProvABCSImpl transforms QueryCustomerPartyListReqMsg into the input of PCM_OP_AR_GET_ACCT_BAL_SUMMARY and calls the opcode PCM_OP_AR_GET_ACCT_BAL_SUMMARY.

6. QueryCustomerPartyListBRMCommsProvABCSImpl then transforms the application programming interface (API) output PCM_OP_AR_GET_ACCT_BAL_

SUMMARY _outputFlist into enterprise business message (EBM) QueryCustomerPartyListRespMsg and returns it to QueryBalanceSummarySiebelCommsReqABCSImpl

7. QueryBalanceSummarySiebelCommsReqABCSImpl transforms the QueryCustomerPartyListRespMsg into QueryBalanceSummaryRespMsg, which is returned to the AccountBalanceSiebelCommsReqABCS.

8. AccountBalanceSiebelCommsReqABCS returns the QueryBalanceSummarySiebelMsg to the calling Siebel web service AccountBalanceSiebelCommsReqABCS.

9. The response is then written to the Siebel Balance Summary virtual business component (VBC) for the user.

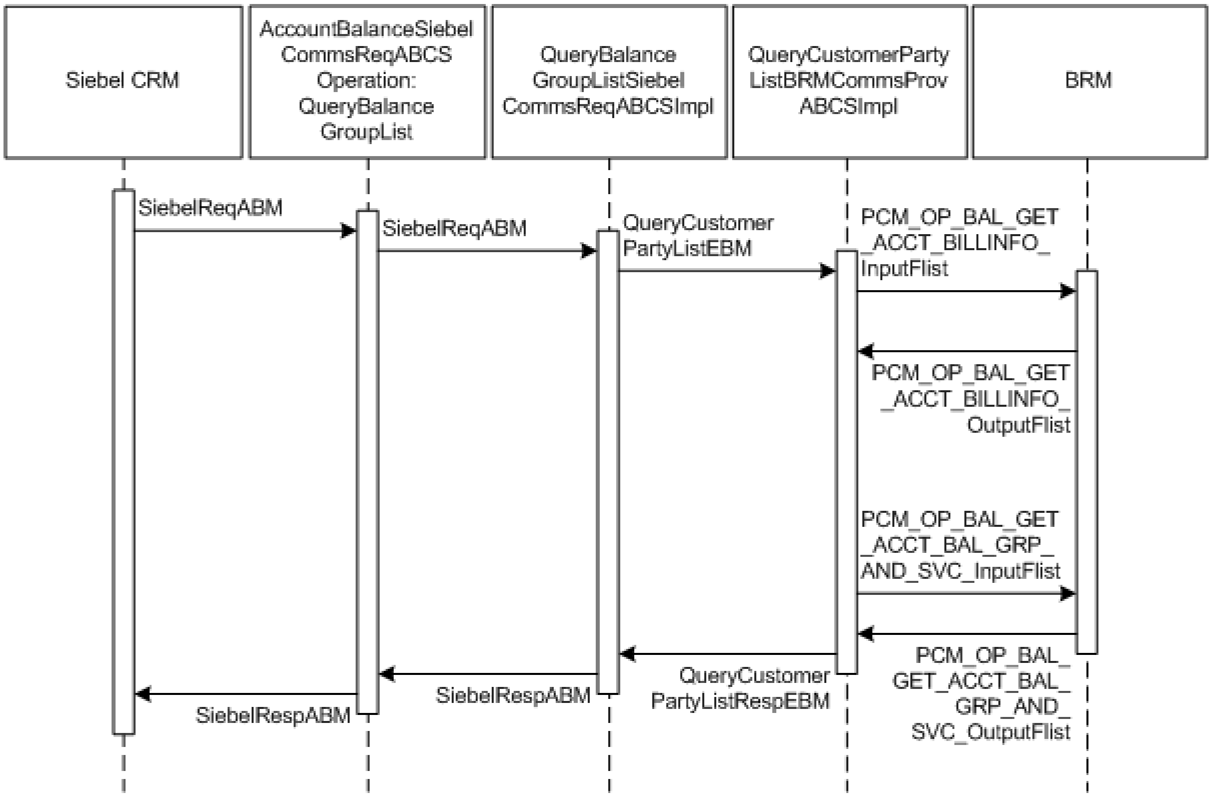
QueryBalanceGroupList Integration Flow

This integration uses the following service interfaces:

- AccountBalanceSiebelCommsReqABCS with operation QueryBalanceGroupList
- QueryBalanceGroupListSiebelCommslReqABCSImpl
- QueryCustomerPartyListBRMCommsProvABCSImpl

Figure 3–3 displays the QueryBalanceGroupList integration scenario.

Figure 3–3 QueryBalanceGroupList Integration Flow Sequence Diagram


Siebel CRM
AccountBalanceSiebel CommsReqABCS Operation: QueryBalance GroupList
QueryBalance GroupListSiebel CommsReqABCSImpl
QueryCustomerParty ListBRMCommsProv ABCSImpl
BRM
SiebelReqABM
SiebelReqABM
QueryCustomer PartyListEBM
PCM_OP_BAL_GET _ACCT_BILLINFO_ InputFlist
PCM_OP_BAL_GET _ACCT_BILLINFO_ OutputFlist
PCM_OP_BAL_GET _ACCT_BAL_GRP_ AND_SVC_InputFlist
PCM_OP_BAL_ GET_ACCT_BAL_ GRP_AND_ SVC_OutputFlist
QueryCustomer PartyListRespEBM
SiebelRespABM
SiebelRespABM

When you initiate the QueryBalanceDetails process, the following events occur:

1. In Siebel CRM, a user navigates to the Accounts screen, queries an account, and clicks a billing profile for the account.

This displays the Billing Profile BRM screen. On the Billing Profile screen, when the user clicks the Balance Group tab, a web service call is made to query the complete list of balance groups for that account billing profile.

2. Navigating to the Billing BRM screen and clicking the Balance Group tab invokes the AccountBalanceSiebelCommsReqABCS web service, which in turn invokes the Siebel Account Balance Interface service AccountBalanceSiebelCommsReqABCS with operation QueryBalanceGroupList.

   AccountBalanceSiebelCommsReqABCS is a generic Oracle AIA Application Business Connector Service (ABCS) interface service with several operations on the Siebel ABM.

3. Invoking AccountBalanceSiebelCommsReqABCS with operation QueryBalanceGroupList routes the QueryBalanceGroupListReqMsg to the QueryBalanceGroupListSiebelCommsReqABCSImpl.

4. The QueryBalanceGroupListSiebelCommslReqABCSImpl transforms the QueryBalanceGroupListReqMsg into QueryCustomerPartyListReqMsgEBM and routes the QueryCustomerPartyListReqMsg to the appropriate billing system.

   As delivered, QueryCustomerPartyListReqMsg is routed to the QueryCustomerPartyListBRMCommsProvABCSImpl.

5. QueryCustomerPartyListBRMCommsProvABCSImpl checks the Query Criteria code. If it is QueryBalanceGroupList, the QueryInvoiceListReqMsg is transformed into PCM_OP_BAL_GET_ACCT_BILLINFO_inputflist.

   This opcode call returns the list of BILLINFO and AR_BILLINFO of that account.

6. QueryCustomerPartyListBRMCommsProvABCSImpl first checks the Query Criteria code. If it is QueryBalanceGroupList, then it transforms QueryCustomerPartyListReqMsg into the input of PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC and then invokes the BRM API PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC to query the list of balance groups of the account billing profile.

7. From the response of PCM_OP_BAL_GET_ACCT_BILLINFO opcode, the appropriate BILLINFO and AR_BILLINFO are picked.

   QueryCustomerPartyListReqMsg is transformed into the input of PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC and calls the BRM opcode PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC. Several balance groups can be in the billing system for an account billing profile. Based on the value of *n* passed from Siebel CRM, the API returns <=n number of balance groups.

8. The list of balance groups from the BRM output list is transformed into QueryCustomerPartyListRespMsg and returned to the QueryBalanceGroupListSiebelCommslReqABCSImpl service.

9. QueryBalanceGroupListSiebelCommslReqABCSImpl then transforms the response messages into QueryBalanceGroupListRespMsg, which is returned to the AccountBalanceSiebelCommsReqABCS.

10. AccountBalanceSiebelCommsReqABCS returns the QueryBalanceGroupListRespMsg to the calling Siebel web service AccountBalanceSiebelCommsReqABCS.

11. The response message is then written to the Siebel Balance Group VBCs for the user.

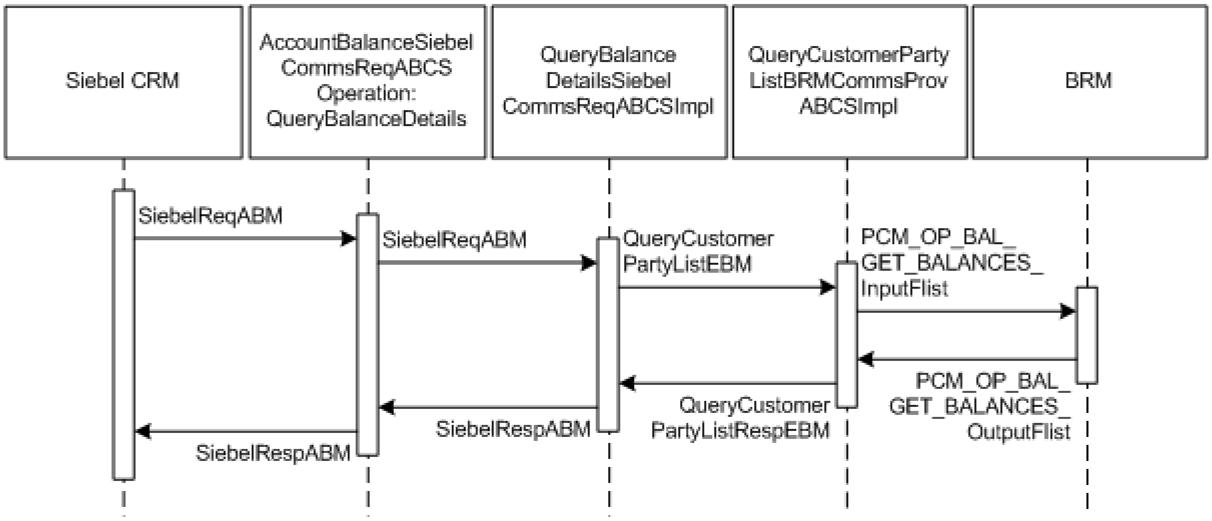
QueryBalanceDetails Integration Flow

This integration uses the following service interfaces:

- AccountBalanceSiebelCommsReqABCS with operation QueryBalanceDetails
- QueryBalanceDetailsSiebelCommsReqABCSImpl
- QueryCustomerPartyListBRMCommsProvABCSImpl

Figure 3–4 displays the QueryBalanceDetails integration scenario.

Figure 3–4 QueryBalanceDetails Integration Flow Sequence Diagram


Siebel CRM
AccountBalanceSiebel CommsReqABCS Operation: QueryBalanceDetails
QueryBalance DetailsSiebel CommsReqABCSImpl
QueryCustomerParty ListBRMCommsProv ABCSImpl
BRM
SiebelReqABM
SiebelReqABM
QueryCustomer PartyListEBM
PCM_OP_BAL_ GET_BALANCES_ InputFlist
PCM_OP_BAL_ GET_BALANCES_ OutputFlist
QueryCustomer PartyListRespEBM
SiebelRespABM
SiebelRespABM

When you initiate the QueryBalanceDetails process, the following events occur:

1. Siebel web service calls AccountBalanceSiebelCommsReqABCS (operation - QueryBalanceDetails) with QueryBalanceDetailsReqMsg ABM comprising account ID, billing profile ID, and balance group ID.
2. AccountBalanceSiebelCommsReqABCS invokes QueryBalanceDetailsSiebelCommsReqABCSImpl with QueryBalanceDetailsReqMsg ABM.
3. QueryBalanceDetailsSiebelCommsReqABCSImpl transforms QueryBalanceGroupListReqMsg ABM to QueryCustomerPartyListReqMsgEBM and sets the value of the field, Query Criteria Code, to Query Balance Details.
4. QueryBalanceDetailsSiebelCommsReqABCSImpl then routes the QueryCustomerPartyListReqMsgEBM to QueryCustomerPartyListBRMCommsProvABCSImpl.
5. QueryCustomerPartyListBRMCommsProvABCSImpl ensures that the value in Query Criteria Code is Query Balance Details and transforms QueryCustomerPartyListReqMsgEBM into BRM ABM and calls the BRM API, PCM_OP_BAL_GET_BALANCES.
6. PCM_OP_BAL_GET_BALANCES takes balance group ID as the input and returns the balance and balance details for that balance group.
7. QueryCustomerPartyListBRMCommsProvABCSImpl transforms the BRM output to QueryCustomerPartyListResMsgEBM.

8. QueryCustomerPartyListResMsgEBM goes as a response to QueryBalanceDetailsSiebelCommsReqABCSImpl.

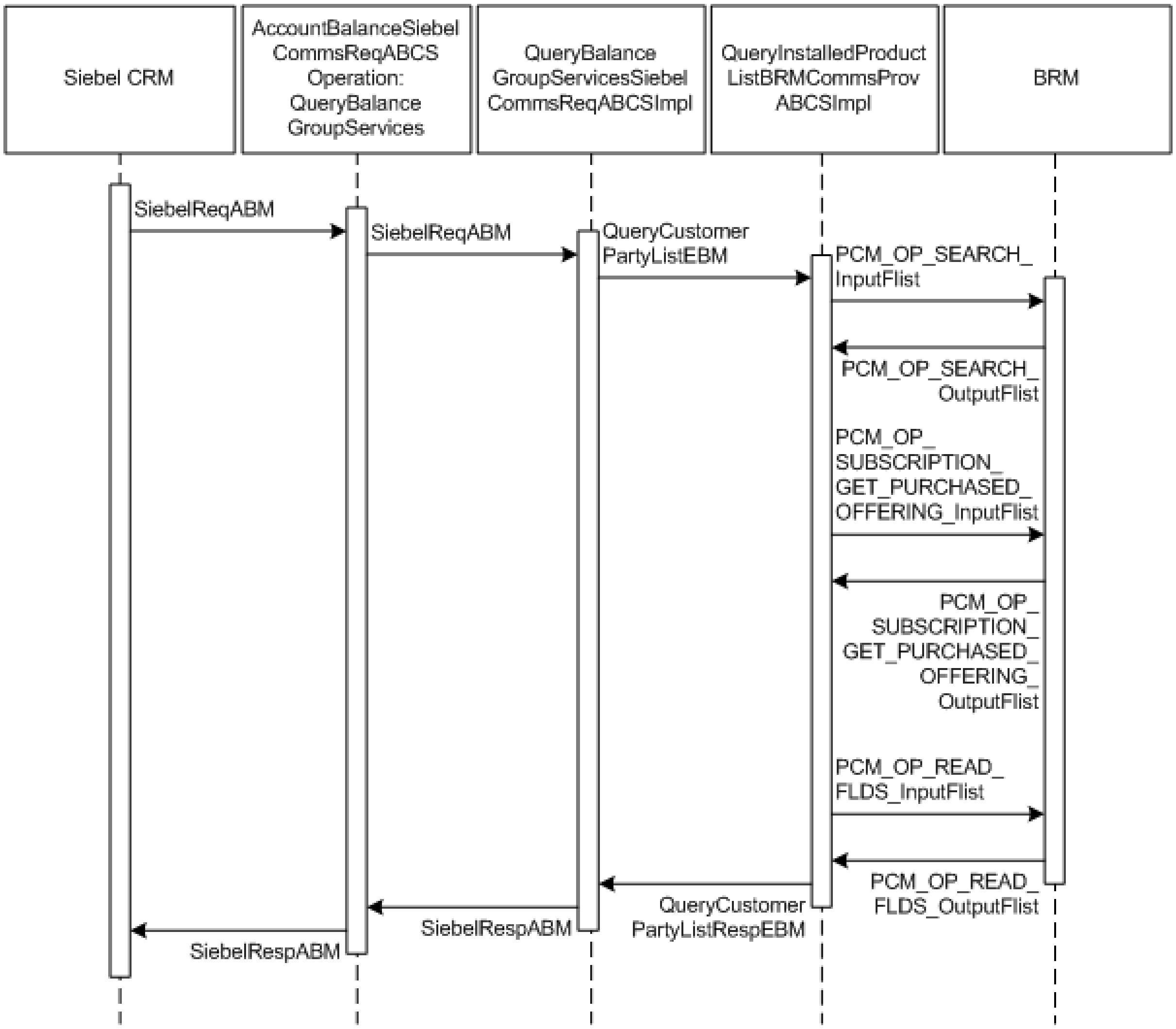
QueryBalanceGroupServices Integration Flow

This integration uses the following service interfaces:

- AccountBalanceSiebelCommsReqABCS
- QueryBalanceGroupServicesSiebelCommsReqABCSImpl
- QueryInstalledProductListBRMCommsProvABCSImpl

Figure 3–5 displays the QueryBalanceGroupServices integration scenario.

Figure 3–5 QueryBalanceGroupServices Integration Flow Sequence Diagram


Siebel CRM
AccountBalanceSiebel CommsReqABCS Operation: QueryBalance GroupServices
QueryBalance GroupServicesSiebel CommsReqABCSImpl
QueryInstalledProduct ListBRMCommsProv ABCSImpl
BRM
SiebelReqABM
SiebelReqABM
QueryCustomer PartyListEBM
PCM_OP_SEARCH_ InputFlist
PCM_OP_SEARCH_ OutputFlist
PCM_OP_ SUBSCRIPTION_ GET_PURCHASED_ OFFERING_InputFlist
PCM_OP_ SUBSCRIPTION_ GET_PURCHASED_ OFFERING_ OutputFlist
PCM_OP_READ_ FLDS_InputFlist
PCM_OP_READ_ FLDS_OutputFlist
QueryCustomer PartyListRespEBM
SiebelRespABM
SiebelRespABM

When you initiate the QueryBalanceGroupServices process, the following events occur:

1. Siebel web service calls AccountBalanceSiebelCommsReqABCS using operation QueryBalanceGroupServices with QueryBalanceGroupServicesReqMsg ABM comprising account ID, billing profile ID, and balance group ID.

2. AccountBalanceSiebelCommsReqABCS invokes QueryBalanceGroupServicesSiebelCommsReqABCSImpl with QueryBalanceGroupServicesReqMsg ABM.
3. QueryBalanceGroupServicesSiebelCommsReqABCSImpl transforms QueryBalanceGroupServicesReqMsg ABM to QueryInstalledProductListReqMsgEBM and sets the value of the field, Query Criteria Code, to Query Balance Group Services.
4. QueryBalanceGroupServicesSiebelCommsReqABCSImpl routes QueryInstalledProductListReqMsgEBM to QueryInstalledProductListBRMCommsProvABCSImpl.

   This service:

   a. Transforms QueryInstalledProductListReqMsgEBM to BRM ABM and calls the BRM API, PCM_OP_SEARCH, which takes a query statement involving balance group ID as input and returns the list of service IDs for that balance group.
   b. Calls the BRM API, PCM_OP_SUBSCRIPTION_GET_PURCHASED_ OFFERINGS for each of the service IDs queried, which accepts a service ID as input and returns the list of product IDs associated with that service.
   c. Calls the BRM API, PCM_OP_READ_FLDS for each of the product IDs queried, which takes a product ID as input and returns the product details for that ID.
5. QueryInstalledProductListBRMCommsProvABCSImpl merges and transforms the BRM output to QueryInstalledProductListResMsg enterprise business message (EBM).
6. QueryInstalledProductListResMsgEBM goes as a response to QueryBalanceGroupServicesSiebelCommsReqABCSImpl.

BRM Interfaces

The QueryBalanceSummary integration flow uses this opcode:

- PCM_OP_AR_GET_ACCT_BAL_SUMMARY

The QueryBalanceGroupList integration flow uses these opcodes:

- PCM_OP_AR_GET_ACCT_BILLS
- PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC

The QueryBalanceDetails integration flow uses this opcode:

- PCM_OP_BAL_GET_BALANCES

The QueryBalanceGroupServices integration flow uses these opcodes:

- PCM_OP_SEARCH
- PCM_OP_SUBSCRIPTION_GET_PURCHASED_OFFERINGS
- API, PCM_OP_READ_FLDS

See *Oracle Communications Billing and Revenue Management Opcode Flist Reference* for more information.

Siebel CRM Interfaces

The account balance integration flow uses this Siebel CRM interface:

- AccountBalanceSiebelCommsReqABCS

See "Web Services Reference" in the *Siebel Order Management Guide Addendum for Communications*.

Industry Oracle AIA Components

The account balance integration uses the following delivered enterprise business objects (EBOs) and enterprise business messages (EBMs):

- CustomerPartyEBO
- InstalledProductEBO
- QueryCustomerPartyListEBM
- QueryCustomerPartyListResponseEBM
- QueryInstalledProductListEBM
- QueryInstalledProductListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the AIA Reference Doc link.

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

Integration Services

These services are delivered with the account balance integration:

- AccountBalanceSiebelCommsReqABCS
- QueryBalanceSummarySiebelCommsReqABCSImpl
- QueryCustomerPartyListBRMCommsProvABCSImpl
- QueryBalanceGroupListSiebelCommsReqABCSImpl

- QueryBalanceDetailsSiebelCommsReqABCSImpl
- QueryBalanceGroupServicesSiebelCommsReqABCSImpl
- QueryInstalledProductListBRMCommsProvABCSImpl

AccountBalanceSiebelCommsReqABCS

AccountBalanceSiebelCommsReqABCS mediates calls between the clients and the provider. AccountBalanceSiebelCommsReqABCS exposes the following operations related to Account Balance integration on the Siebel ABM.

- QueryBalanceSummary:
  - Routes QueryBalanceSummaryReqMsg to the requester implementation service
  - Routes QueryBalanceSummaryRespMsg to the requester
- QueryBalanceDetails:
  - Routes QueryBalanceDetailsReqMsg to the requester implementation service
  - Routes QueryBalanceDetailsRespMsg to the requester
- QueryBalanceGroupList:
  - Routes QueryBalanceGroupListReqMsg to the requester implementation service
  - Routes QueryBalanceGroupListRespMsg to the requester
- QueryBalanceGroupServices:
  - Routes QueryBalanceGroupServicesReqMsg to the requester implementation service
  - Routes QueryBalanceGroupServicesRespMsg to the requester

QueryBalanceSummarySiebelCommsReqABCSImpl

QueryBalanceSummarySiebelCommsReqABCSImpl transforms the Siebel message into QueryBalanceSummaryEBM and calls the provider to query the balance summary response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryCustomerPartyListBRMCommsProvABCSImpl

QueryCustomerPartyListBRMCommsProvABCSImpl transforms the QueryBalanceSummaryEBM into BRM API input format and calls the API to query the balance summary output from the billing system. It then transforms the output from the API back to a CustomerPartyBalanceEBM message and returns it to the requestor.

QueryBalanceGroupListSiebelCommslReqABCSImpl

The QueryBalanceGroupListSiebelCommslReqABCSImpl transforms the QueryBalanceGroupListReqMsg into QueryCustomerPartyListReqMsgEBM.

QueryBalanceDetailsSiebelCommsReqABCSImpl

The QueryBalanceDetailsSiebelCommsReqABCSImpl is a Business Process Execution Language (BPEL) process that transforms the Siebel message into the

QueryBalanceDetailsEBM and calls the provider to query the balance group and balance group balance details response from BRM. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryBalanceGroupServicesSiebelCommsReqABCSImpl

QueryBalanceGroupServicesSiebelCommsReqABCSImpl transforms the Siebel message into QueryInstalledProductListEBM and calls the provider to query the balance group list from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryInstalledProductListBRMCommsProvABCSImpl

QueryInstalledProductListBRMProvABCSImpl transforms QueryInstalledProductListReqMsgEBM to BRM ABM and calls the BRM API, PCM_OP_SEARCH, which takes a query statement involving balance group ID as input and returns the list of service IDs for that balance group.

For each of the service IDs queried, QueryInstalledProductListBRMProvABCSImpl calls the BRM API, PCM_OP_SUBSCRIPTION_GET_PURCHASED_OFFERINGS, which takes a service ID as input and returns the list of product IDs associated with that service.

For each of the product IDs queried, QueryInstalledProductListBRMProvABCSImpl calls the BRM API, PCM_OP_READ_FLDS, which takes a product ID as input and returns the product details for that ID.

4

Invoice Integration Flow

This chapter provides an overview of the invoice integration flows and describes Siebel Customer Relationship Management (Siebel CRM) and Oracle Billing and Revenue Management (BRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Invoice Integration Overview

The invoice integration between Siebel CRM and BRM supports the following integration scenarios:

- QueryInvoiceList lets you view a list of invoices for an account billing profile in Siebel CRM.
- QueryPDFInvoice lets you view a PDF version of an invoice in Siebel CRM.
- QueryInvoice lets you view invoice details in Siebel CRM.
- QueryInvoiceUsageAllocation lets you view invoice event details (call detail (CDR) records) in Siebel CRM.
- SearchInvoiceUsageAllocation lets you search invoice event detail records (CDR records) in Siebel CRM.
- QueryInvoiceUsageAllocationResource lets you view nonmonetary resource balance details in Siebel CRM.

BRM generates invoices on a periodic basis by after a bill cycle. Once generated, an invoice does not change. An invoice usually consists of sections for a header, a summary, and details. You can view invoice data at any of these levels, and resolve most bill inquiries by viewing the header and summary. The integration between Siebel CRM and BRM presents the most current invoices in Siebel CRM and retrieves the header, summary, and details on demand from BRM to show in Siebel CRM.

When you select the **Bills** tab of the Billing Profile screen, the integration shows a list of the latest invoices. You set the number of invoices to display in Siebel CRM, and the integration retrieves that many invoices from BRM. From this list, you can take the following actions:

- View a PDF of the invoice:

  a. Highlight an invoice in the list.

  b. Click **View Invoice**.

     A PDF corresponding to the invoice appears in a new window.

- View invoice details:

a. Click the **Bill Number** link.

   The Bill Detail view appears and the integration retrieves the invoice or bill header and item data, and displays it on the header and items applet of the Bill Detail view. A single request from Siebel CRM invokes two APIs in BRM that call the invoice header and items data and return a single hierarchical message to Siebel CRM.

b. Click the **Payment** tab or the **A/R items** tab.

   The integration retrieves payment or A/R items data for the selected invoice or bill.

- View item or event details:

  a. Click the **Bill Number** link.

  b. Select an invoice item.

  c. Click the **Net Amount** link associated with the item charge.

     The integration retrieves the item or event details record from BRM. Because this item detail data is generally large, you can filter the data using search criteria.

- View individual service charges:

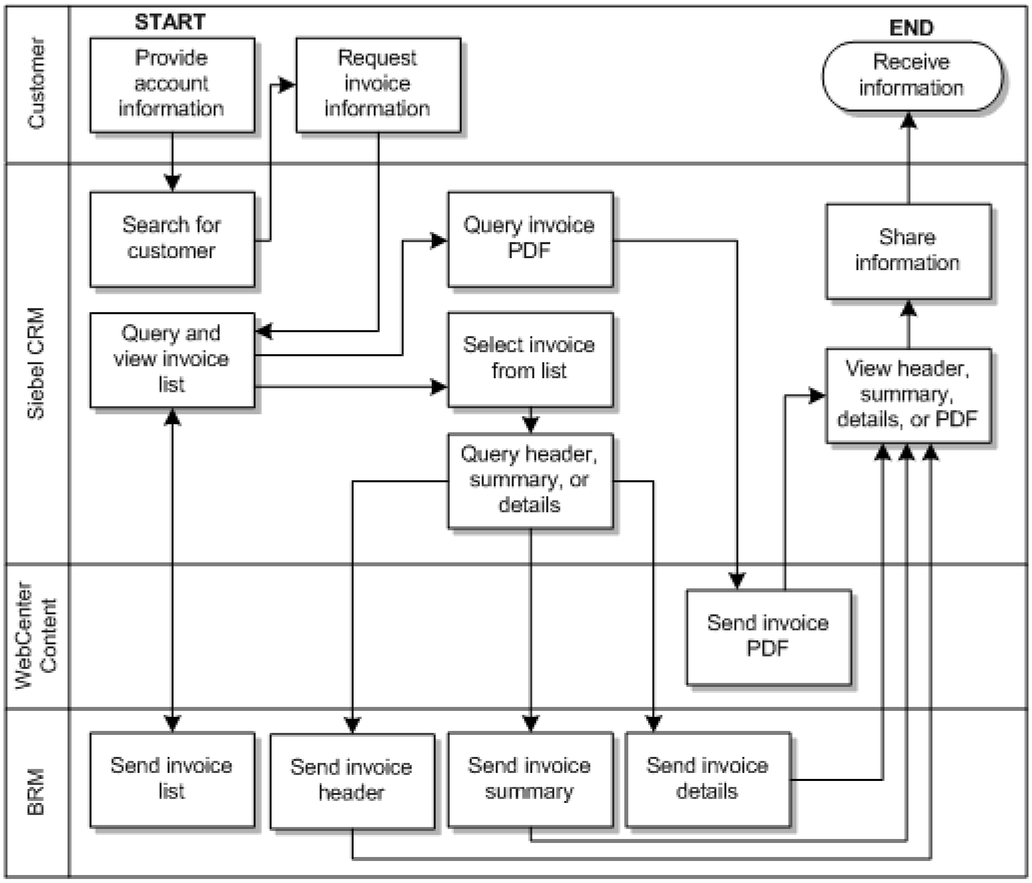
  a. Click the **Bill Number** link.

  b. Select the Service Charges applet.

  Note: The date and time information of the event details and CDR displayed in Siebel CRM is specific to the time zone from where the event originated.

Figure 4–1 illustrates the invoice integration flow.

Figure 4–1 The Invoice Integration Flow



To view invoice information for an account in the billing profile in Siebel CRM:

1. Navigate to the Accounts screen and query an account. Scroll down to the Billing Profile applet for this account and click the Billing Profile **Name** link.
2. Click the **Bills** tab to view a list of invoices.
3. Click the **Bill Number** link to view the details for that invoice.
4. Go to the Item Charges applet and click the **Net Amount** link to view the invoice usage allocation (CDRs) for that item.
5. Click the **Search** tab to search invoice event detail records.
6. In the Event Details, click **View Details** to query the resource impact details (non-monetary charges, for example free minutes, and so on) for an event from the billing system.

See "Mapping Siebel Billing Management UI Elements to BRM Customer Center" for more information on mapping Siebel CRM elements to BRM elements.

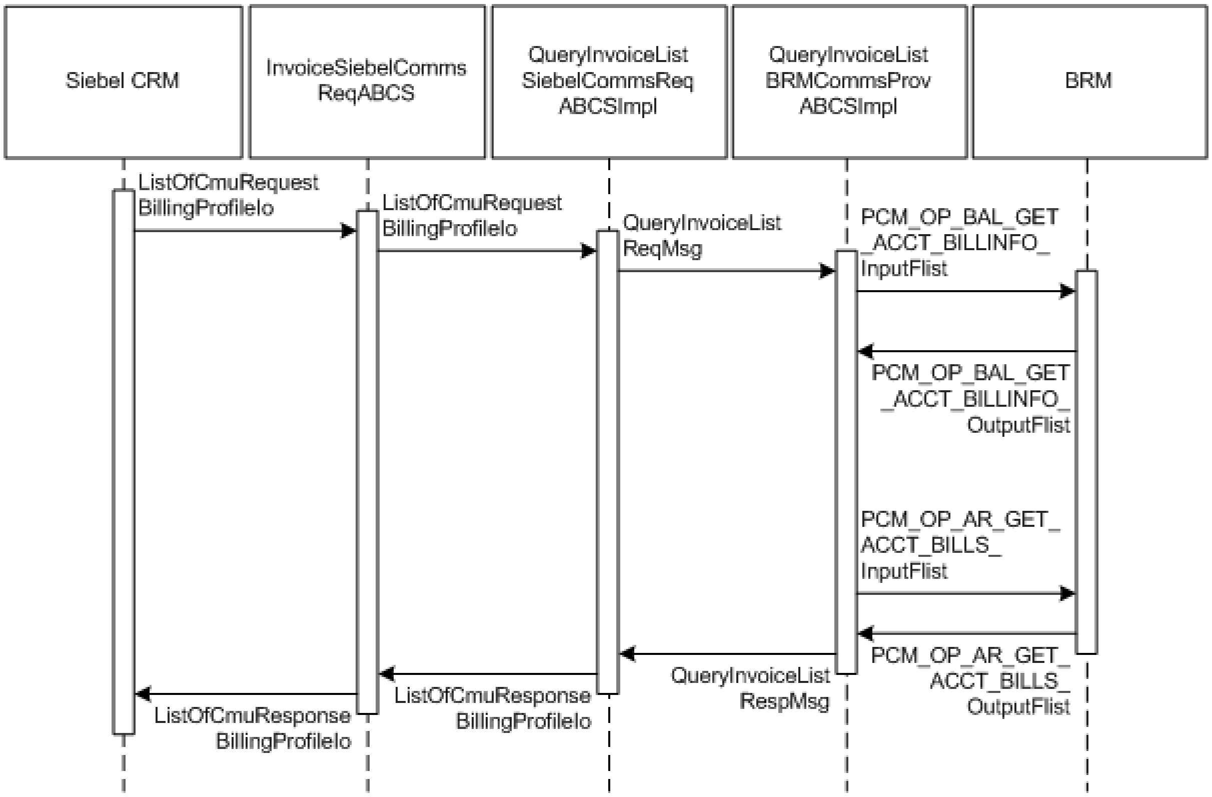
QueryInvoiceList Integration Flow

This integration flow uses the following interfaces:

- InvoiceSiebelCommsReqABCS with operation QueryInvoiceList
- QueryInvoiceListSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl

Figure 4–2 illustrates the QueryInvoiceList integration scenario.

Figure 4–2 QueryInvoiceList Integration Flow Sequence Diagram


Siebel CRM
InvoiceSiebelComms ReqABCS
QueryInvoiceList SiebelCommsReq ABCSImpl
QueryInvoiceList BRMCommsProv ABCSImpl
BRM
ListOfCmuRequest BillingProfileIo
ListOfCmuRequest BillingProfileIo
QueryInvoiceList ReqMsg
PCM_OP_BAL_GET _ACCT_BILLINFO_ InputFlist
PCM_OP_BAL_GET _ACCT_BILLINFO_ OutputFlist
PCM_OP_AR_GET_ ACCT_BILLS_ InputFlist
PCM_OP_AR_GET_ ACCT_BILLS_ OutputFlist
QueryInvoiceList RespMsg
ListOfCmuResponse BillingProfileIo
ListOfCmuResponse BillingProfileIo

When you initiate the QueryInvoiceList process, the following events occur:

1. In Siebel CRM, a user navigates to the Accounts screen, queries an account, and clicks a billing profile for the account.

   This displays the Billing Profile BRM screen, Bills tab, a web service call is made to query the list of *n* invoices for that billing profile. Many invoices may be in BRM, but the number *n* of invoices to be fetched is sent from Siebel CRM as part of the request message and <=n number of invoices are returned from BRM. The default values of invoices (*n*) queried depends on the configuration in Siebel Integration Object CMU Request Billing Profile IO integration component Com Invoice Profile field Maximum Number Of Records.

2. Navigating to the Invoice applet in the Billing Profile BRM screen invokes outbound web service InvoiceSiebelCommsReqABCS, which in turn calls the InvoiceSiebelCommsReqABCS with operation QueryInvoiceList.

   InvoiceSiebelCommsReqABCS is a generic Invoice interface service with several operations defined on the Invoice application business message (ABM).

3. Invoking InvoiceSiebelCommsReqABCS with operation QueryInvoiceList routes the QueryInvoiceListReqMsg to the QueryInvoiceListSiebelCommsReqABCSImpl.

4. The QueryInvoiceListSiebelCommsReqABCSImpl first transforms the QueryInvoiceListReqMsg into QueryInvoiceListRequest enterprise business message (EBM) and routes the QueryInvoiceListReqMsg to the appropriate billing systems.

As delivered, QueryInvoiceListReqMsg is routed to the QueryInvoiceListBRMCommsProvABCSImpl.

5. QueryInvoiceListBRMCommsProvABCSImpl first checks the Query Criteria code.

   If it is Query Invoice List, QueryInvoiceListReqMsg is transformed into PCM_OP_BAL_GET_ACCT_BILLINFO_inputflist. This opcode call returns the list of BILLINFO and AR_BILLINFO of that account.

6. From the response of PCM_OP_BAL_GET_ACCT_BILLINFO opcode, the appropriate BILLINFO and AR_BILLINFO are picked.

   QueryInvoiceListReqMsg is transformed into the input of PCM_OP_AR_GET_ACCT_BILLS and calls the BRM opcode PCM_OP_AR_GET_ACCT_BILLS. Many invoices may be in the billing system for an account billing profile. Based on the value of *n* passed from Siebel CRM, the application programming interface (API) returns <=n number of invoices.

7. QueryInvoiceListBRMCommsProvABCSImpl then transforms the API output PCM_OP_AR_GET_ACCT_BILLS_RespMsg into EBM QueryInvoiceListRespMsg and returns it to QueryInvoiceListSiebelCommsReqABCSImpl.

8. QueryInvoiceListSiebelCommsReqABCSImpl then transforms the QueryInvoiceListRespMsg into QueryInvoiceListRespMsg, which is returned to the InvoiceSiebelCommsReqABCS.

9. InvoiceSiebelCommsReqABCS returns the QueryInvoiceListRespMsg to the calling Siebel web service InvoiceSiebelCommsReqABCSService.

10. The system then writes the list of bills to the Siebel Invoice virtual business component (VBC) for the user.

QueryPDFInvoice Integration Flow

This integration flow is supported by a process for generating and storing PDF invoices that does not use Oracle AIA services and interfaces. Instead, the process uses custom integration between supporting applications to deliver PDF invoices to Siebel CRM.

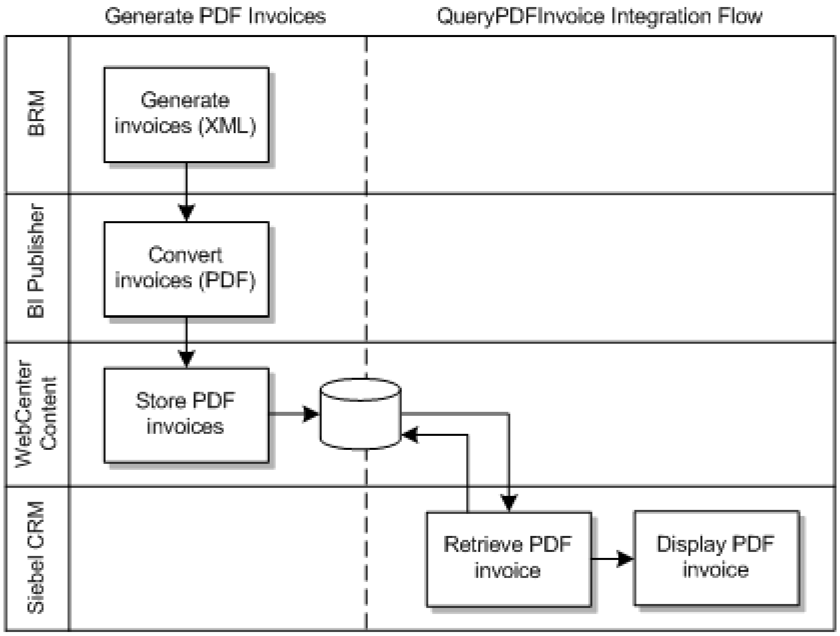
The following applications participate in the process of generating, storing, and displaying invoices:

- **BRM**: Generates invoice data in XML format according to a schedule and sends the data to Oracle Business Intelligence Publisher (BI Publisher) in batches.
- **BI Publisher**: Converts the BRM XML invoice data to PDFs according to a template and uploads the batch of invoices to Oracle WebCenter Content.
- **WebCenter Content**: Stores the PDF invoices in a central repository.
- **Siebel CRM**: Requests the PDF invoices from WebCenter Content and then displays them.

The QueryPDFInvoice integration flow handles the process of requesting the PDF invoices from WebCenter Content and displaying them in Siebel CRM.

Figure 4–3 illustrates the process for generating, storing, and displaying invoices alongside the QueryPDFInvoice integration flow.

Figure 4–3 Process for Generating, Storing, and Displaying Invoices



As part of the QueryPDFInvoice integration flow, you configure the applications to integrate directly as described in "Setting Up PDF Invoice Display."

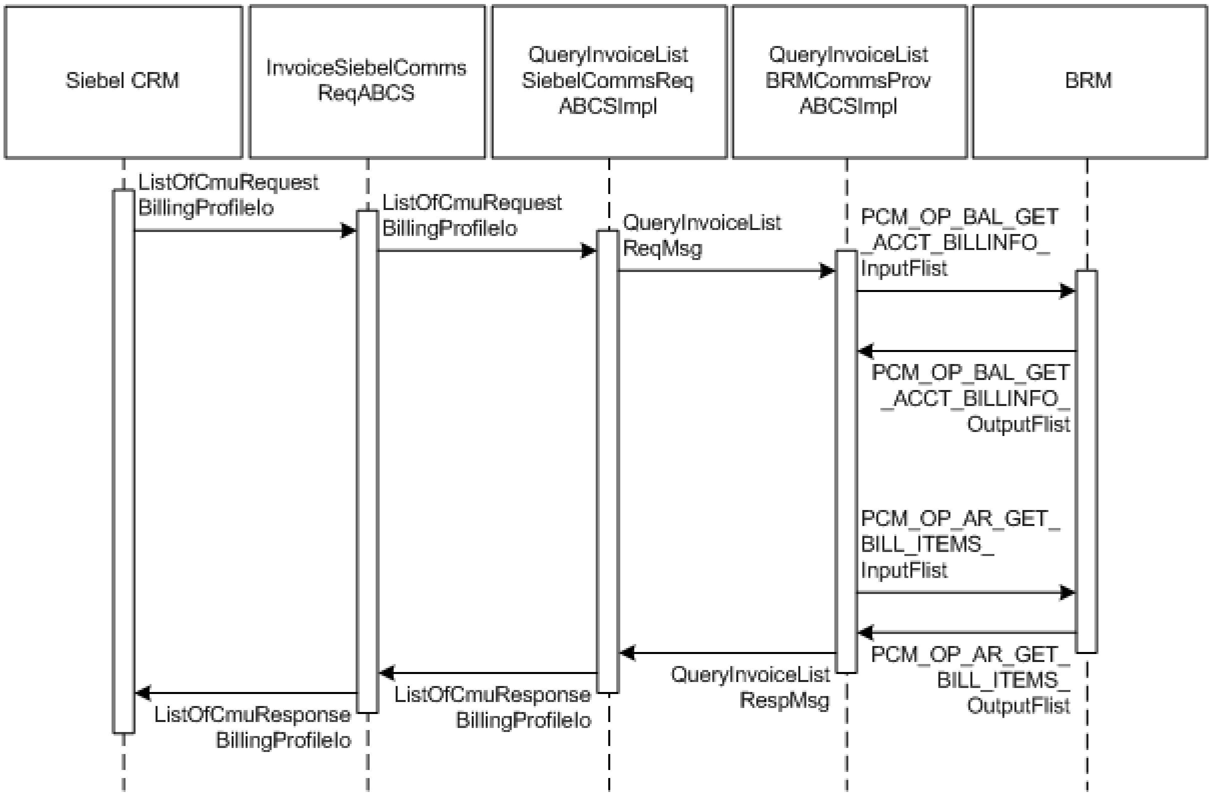
QueryInvoice Integration Flow

This integration flow uses the following interfaces:

- InvoiceSiebelCommsReqABCS with operation QueryInvoice
- QueryInvoiceSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl

Figure 4–4 illustrates the QueryInvoice integration scenario.

Figure 4–4 QueryInvoice Integration Flow Sequence Diagram


Siebel CRM
InvoiceSiebelComms ReqABCS
QueryInvoiceList SiebelCommsReq ABCSImpl
QueryInvoiceList BRMCommsProv ABCSImpl
BRM
ListOfCmuRequest BillingProfileIo
ListOfCmuRequest BillingProfileIo
QueryInvoiceList ReqMsg
PCM_OP_BAL_GET _ACCT_BILLINFO_ InputFlist
PCM_OP_BAL_GET _ACCT_BILLINFO_ OutputFlist
PCM_OP_AR_GET_ BILL_ITEMS_ InputFlist
PCM_OP_AR_GET_ BILL_ITEMS_ OutputFlist
QueryInvoiceList RespMsg
ListOfCmuResponse BillingProfileIo
ListOfCmuResponse BillingProfileIo

When you initiate the QueryInvoice process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and click a Billing Profile for the account.

   This displays the Billing Profile screen. Click the Bills tab and drill-down on the Bill Number to call the query invoice information (header, service charges, and items) from the billing system.

2. The InvoiceSiebelCommsReqABCS with the operation QueryInvoice is called.

   The InvoiceSiebelCommsReqABCS is a generic Siebel invoice interface service with several operations defined on the Invoice EBO.

3. The InvoiceSiebelCommsReqABCS with the operation QueryInvoice routes the QueryInvoiceReqMsg to the QueryInvoiceSiebelCommsReqABCSImpl.

4. The QueryInvoiceSiebelCommsReqABCSImpl transforms the QueryInvoiceReqMsg into the Invoice EBM and routes the QueryInvoiceListReqMsg to the appropriate billing system.

   As delivered, QueryInvoiceListReqMsg is routed to the QueryInvoiceListBRMCommsProvABCSImpl.

5. The QueryInvoiceListBRMCommsProvABCSImpl first checks the Query Criteria code. If it is Query Invoice, it then transforms QueryInvoiceListReqMsg into the input of PCM_OP_BAL_GET_ACCT_BILLINFO_inputFlist and invokes PCM_OP_BAL_GET_ACCT_BILLINFO to get the Bill Info object.

   This information is used to populate PCM_OP_AR_GET_BILL_ITEMS_inputFlist and calls the BRM opcode PCM_OP_AR_GET_BILL_ITEMS. The opcode returns the invoice header, service charges, and items in a flat message to the calling QueryInvoiceListBRMCommsProvABCSImpl.

6. The QueryInvoiceListBRMCommsProvABCSImpl transforms the API output PCM_OP_AR_GET_BILL_ITEMS_outputFlist into the EBM QueryInvoiceListRespMsg and returns it to QueryInvoiceSiebelCommsReqABCSImpl.
7. The QueryInvoiceSiebelCommsReqABCSImpl transforms the QueryInvoiceListRespMsg into the QueryInvoiceRespMsg and returns it to the Siebel Invoice ABC interface service.

   The QueryInvoiceListRespMsg is a flat message from which service charges are calculated and the invoice header, service charges, and items are returned as the QueryInvoiceRespMsg to the calling InvoiceSiebelCommsReqABCS.
8. The InvoiceSiebelCommsReqABCS returns the QueryInvoiceRespMsg to the calling Siebel web service.

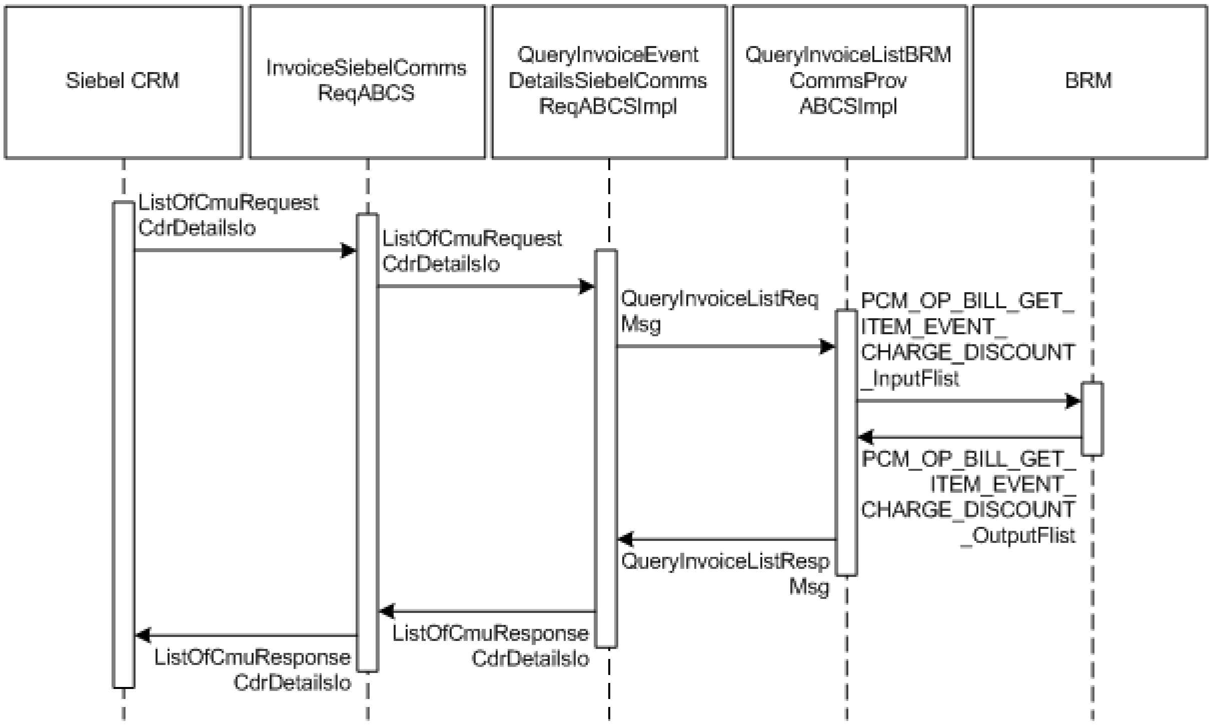
QueryInvoiceUsageAllocation Integration Flow

This integration flow uses the following interfaces:

- InvoiceSiebelCommsReqABCS with operation QueryEventDetails
- QueryInvoiceEventDetailsSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl

Figure 4–5 illustrates the QueryInvoiceUsageAllocation integration scenario.

Figure 4–5 QueryInvoiceUsageAllocation Integration Flow Sequence Diagram


Siebel CRM
InvoiceSiebelComms ReqABCS
QueryInvoiceEvent DetailsSiebelComms ReqABCSImpl
QueryInvoiceListBRM CommsProv ABCSImpl
BRM
ListOfCmuRequest CdrDetailsIo
ListOfCmuRequest CdrDetailsIo
QueryInvoiceListReq Msg
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _InputFlist
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _OutputFlist
QueryInvoiceListResp Msg
ListOfCmuResponse CdrDetailsIo
ListOfCmuResponse CdrDetailsIo

When you initiate the QueryInvoiceUsageAllocation process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and select the billing profile for the account.

   In the Billing Profile screen, click the Bills tab and select an invoice. Select an item charge and click the Net Amount link of the item to view Event Details. This action calls Siebel outbound web service InvoiceSiebelCommsReqABCS to query the event details from the billing system.

2. Navigating to the Invoice Event Details (call detail (CDR) records) applet in the Billing BRM screen invokes the InvoiceSiebelCommsReqABCSService web service with operation QueryEventDetails, which in turn calls the InvoiceSiebelCommsReqABCS with operation QueryEventDetails.

   InvoiceSiebelCommsReqABCS is a generic Siebel Invoice interface service with several operations defined in the Invoice EBO.

3. Invoking InvoiceSiebelCommsReqABCS with operation QueryEventDetails routes the QueryEventDetailsReqMsg to the QueryInvoiceEventDetailsSiebelCommsReqABCSImpl.

4. The QueryInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the QueryEventDetailsReqMsg into Invoice EBM and routes the QueryInvoiceListReqMsg to the appropriate billing system.

   As delivered, QueryInvoiceListReqMsg is routed to the QueryInvoiceListBRMCommsProvABCSImpl.

5. QueryInvoiceListBRMCommsProvABCSImpl checks query criteria code.

   If it is Query Usage Allocation, then it transforms QueryInvoiceListReqMsg into the input of PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT and calls the opcode PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT.

6. QueryInvoiceListBRMCommsProvABCSImpl then transforms the API output PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT_RespMsg into EBM QueryInvoiceListRespMsg and returns it to QueryInvoiceEventDetailsSiebelCommsReqABCSImpl.

7. QueryInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the QueryInvoiceListRespMsg into QueryEventDetailsRespMsg, which is returned to InvoiceSiebelCommsReqABCS.

8. InvoiceSiebelCommsReqABCS returns the QueryEventDetailsRespMsg to the calling Siebel web service.

9. The system then writes the response message to the Siebel Invoice VBC for the user.

SearchInvoiceUsageAllocation Integration Flow

This integration flow uses the following interfaces:

- InvoiceSiebelCommsReqABCS with operation SearchEventDetails
- SearchInvoiceEventDetailsSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl

The SearchInvoiceUsageAllocation supports searching invoice event details (CDR records) on the following columns in addition to the account ID and the item charge ID that are passed to Query Invoice Details:

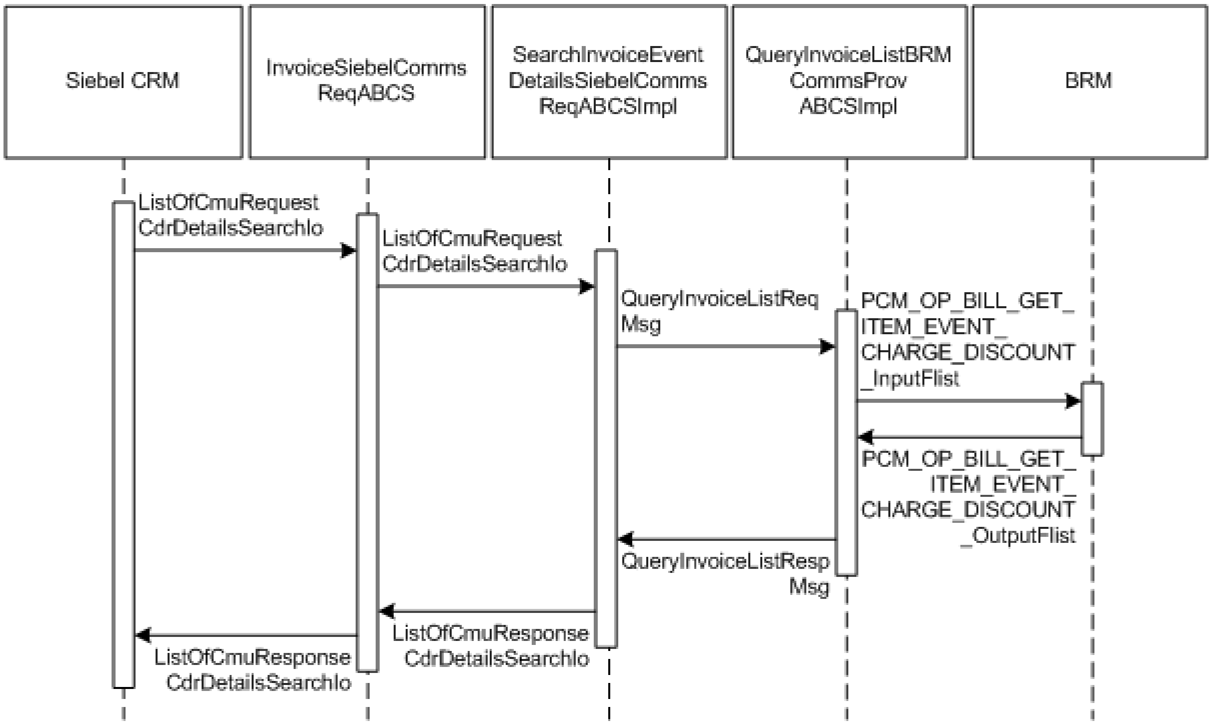
- Minimum Amount

- Maximum Amount
- Start Date
- End Date

The BRM opcode PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT supports only simple queries, so the search supports passing only the data through and not passing complex query criteria, (for example, >, <, between, and so on).

Figure 4–6 illustrates the SearchInvoiceUsageAllocation integration scenario.

Figure 4–6 ServiceinvoiceUsageAllocation Integration Flow Sequence Diagram


Siebel CRM
InvoiceSiebelComms ReqABCS
SearchInvoiceEvent DetailsSiebelComms ReqABCSImpl
QueryInvoiceListBRM CommsProv ABCSImpl
BRM
ListOfCmuRequest CdrDetailsSearchlo
ListOfCmuRequest CdrDetailsSearchlo
QueryInvoiceListReq Msg
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _InputFlist
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _OutputFlist
QueryInvoiceListResp Msg
ListOfCmuResponse CdrDetailsSearchlo
ListOfCmuResponse CdrDetailsSearchlo

When you initiate the SearchInvoiceUsageAllocation process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and select the billing profile for the account.

   In the Billing Profile screen, click the Bills tab and select an invoice. Select an item charge, and click the Net Amount link of the item to view event details of all CDR records for that item charge.

2. Clicking the Search button on the Invoice Event Details (CDR details) applet opens the search applet for query.

3. Entering the search criteria and clicking **Go** invokes an outbound web service InvoiceSiebelReqABCS that in turn calls the InvoiceSiebelCommsReqABCS with operation SearchEventDetails.

   InvoiceSiebelCommsReqABCS is a generic Siebel Invoice interface service with several operations defined on the Invoice EBO.

4. Invoking InvoiceSiebelCommsReqABCS with operation SearchEventDetails routes the SearchEventDetailsReqMsg to the SearchInvoiceEventDetailsSiebelCommsReqABCSImpl.

5. The SearchInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the SearchEventDetailsReqMsg into an Invoice EBM and routes the QueryInvoiceListReqMsg to the appropriate billing systems.

   As delivered, QueryInvoiceListReqMsg is routed to the QueryInvoiceListBRMCommsProvABCSImpl.

6. QueryInvoiceListBRMCommsProvABCSImpl checks the query criteria code. If the query criteria code is Usage Allocation, it then transforms QueryInvoiceListReqMsg into the input of PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT and calls the BRM opcode PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT.

7. QueryInvoiceListBRMCommsProvABCSImpl transforms the API output PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT_RespMsg into EBM QueryInvoiceListRespMsg and returns it to SearchInvoiceEventDetailsSiebelCommsReqABCSImpl.

8. SearchInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the QueryInvoiceListRespMsg into a SearchEventDetailsRespMsg, which is returned to InvoiceSiebelCommsReqABCS.

9. InvoiceSiebelCommsReqABCS returns the SearchEventDetailsRespMsg to the calling Siebel web service.

10. The system writes the search response to the Siebel Invoice VBC for the user.

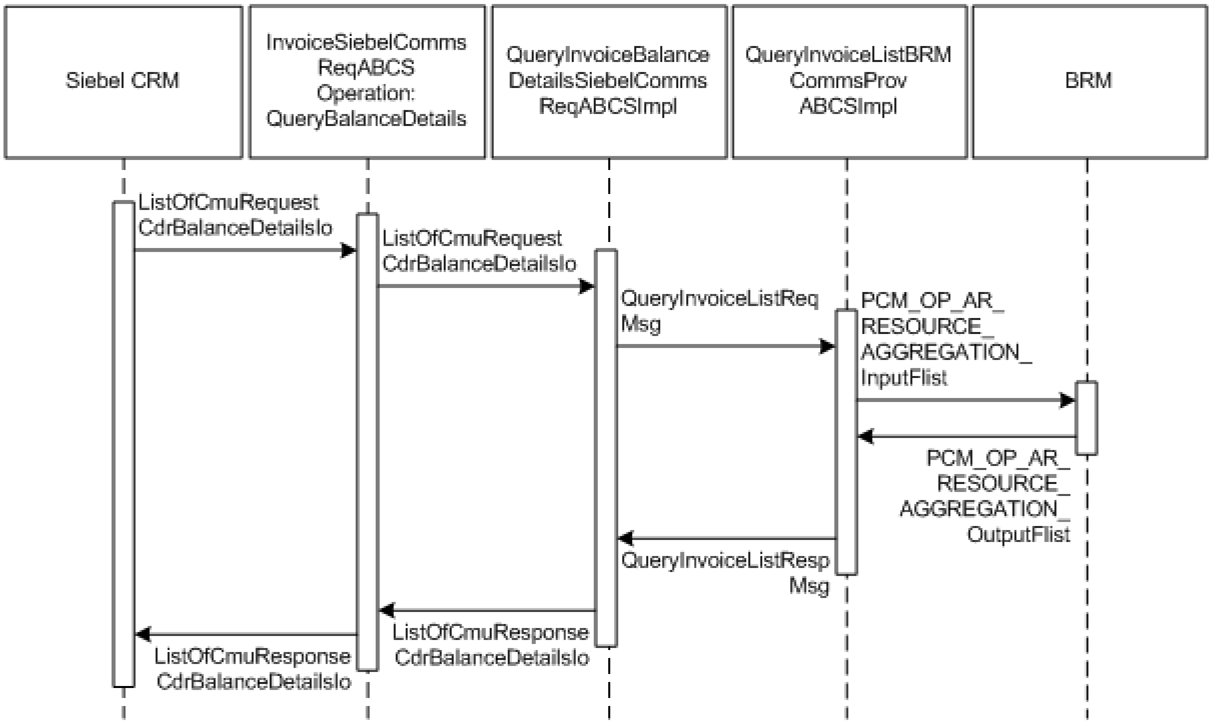
QueryInvoiceUsageAllocationResource Integration Flow

This integration flow uses the following interfaces:

- InvoiceSiebelCommsReqABCS with operation QueryBalanceDetails
- QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl

Figure 4–7 illustrates the QueryInvoiceUsageAllocationResource integration scenario.

Figure 4–7 QueryInvoiceUsageAllocationResource Integration Flow Sequence Diagram


Siebel CRM
InvoiceSiebelComms ReqABCS Operation: QueryBalanceDetails
QueryInvoiceBalance DetailsSiebelComms ReqABCSImpl
QueryInvoiceListBRM CommsProv ABCSImpl
BRM
ListOfCmuRequest CdrBalanceDetailsIo
ListOfCmuRequest CdrBalanceDetailsIo
QueryInvoiceListReq Msg
PCM_OP_AR_ RESOURCE_ AGGREGATION_ InputFlist
PCM_OP_AR_ RESOURCE_ AGGREGATION_ OutputFlist
QueryInvoiceListResp Msg
ListOfCmuResponse CdrBalanceDetailsIo
ListOfCmuResponse CdrBalanceDetailsIo

When you initiate the QueryInvoiceUsageAllocationResource process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and select the billing profile for the account.

   In the Billing Profile screen, click the Bills tab and select an invoice. In Event Details, click View Detail to query the resource impact details for an event from the billing system. A web service call is made to query resource impact details for a particular event from the billing system, which in turn calls the InvoiceSiebelCommsReqABCS with operation QueryResourceBalance.

   InvoiceSiebelCommsReqABCS is a generic Siebel Invoice interface service with several operations defined on the Invoice EBO.

2. Invoking InvoiceSiebelCommsReqABCS with operation QueryBalanceDetails routes the QueryBalanceDetailsReqMsg to the QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl.

3. The QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl transforms the QueryBalanceDetailsReqMsg into Invoice EBM and routes the QueryInvoiceListReqMsg to the appropriate billing system.

   As delivered, QueryInvoiceListReqMsg is routed to the QueryInvoiceListBRMCommsProvABCSImpl.

4. QueryInvoiceListBRMCommsProvABCSImpl checks the query criteria code.

   If the query criteria code is Query Balance Details, it transforms QueryInvoiceListReqMsg into the input of PCM_OP_ AR_RESOURCE_ AGGREGATION and calls the opcode PCM_OP_ AR_RESOURCE_ AGGREGATION.

5. QueryInvoiceListBRMCommsProvABCSImpl transforms the API output PCM_ OP_ AR_RESOURCE_AGGREGATION_RespMsg into EBM

QueryInvoiceListRespMsg and returns it to QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl.

6. QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl transforms the QueryInvoiceListRespMsg into QueryBalanceDetailsRespMsg, which is returned to the InvoiceSiebelCommsReqABCS.
7. InvoiceSiebelCommsReqABCS returns the QueryBalanceDetailsRespMsg back to the calling Siebel web service.
8. The system writes the message to the Siebel UsageAllocationResource VBC for the user.

BRM Interfaces

The integration uses opcodes with the integration flows as shown in Table 4–1.

Table 4–1 Invoice Integration Flow Opcodes

Opcode	**Integration Flow**
PCM_OP_AR_GET_ACCT_BIILS	QueryInvoiceList
PCM_ OP_AR_GET_BILL_ITEMS	QueryInvoice
PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT	QueryInvoiceUsageAllocation
PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT	SearchInvoiceUsageAllocation
PCM_OP_AR_RESOURCE_AGGREGATION_RespMsg	QueryInvoiceUsageAllocationResource

See *BRM Opcode Flist Reference* and *BRM Developer's Guide* for more information about individual opcodes.

Siebel CRM Interfaces

The Invoice integration flow uses the following Siebel CRM interface:

- InvoiceSiebelCommsReqABCS

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

Industry Oracle AIA Components

The Invoice integration flow uses the following delivered enterprise business object (EBO) and enterprise business messages (EBMs):

- InvoiceEBO
- QueryInvoiceEBM
- QueryInvoiceResponseEBM
- QueryInvoiceListEBM
- QueryInvoiceListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_
HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the AIA Reference Doc link.

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

Integration Services

These services are delivered with the Invoice Integration flow:

- InvoiceSiebelCommsReqABCS with operations QueryInvoiceList, QueryInvoice, QueryEventDetails, SearchEventDetails, and QueryBalanceDetails
- QueryInvoiceListSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl
- QueryInvoiceSiebelCommsReqABCSImpl
- QueryInvoiceEventDetailsSiebelCommsReqABCSImpl
- SearchInvoiceEventDetailsSiebelCommsReqABCSImpl
- QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl

InvoiceSiebelCommsReqABCS

InvoiceSiebelCommsReqABCS exposes the following operations related to the Invoice integration on the Siebel ABM:

- QueryInvoiceList:
  - Routes QueryInvoiceListReqMsg to the requester implementation service
  - Routes QueryInvoiceListRespMsg to the requester
- QueryInvoice:
  - Routes QueryInvoiceReqMsg to the requester implementation service
  - Routes QueryInvoiceRespMsg to the requester
- QueryEventDetails:
  - Routes QueryEventDetailsReqMsg to the requester implementation service
  - Routes QueryEventDetailsRespMsg to the requester.
- SearchEventDetails:
  - Routes SearchEventDetailsReqMsg to the requester implementation service

  - Routes SearchEventDetailsRespMsg to the requester
- QueryBalanceDetails:
  - Routes QueryBalanceDetailsReqMsg to the requester implementation service
  - Routes QueryBalanceDetailsRespMsg to the requester
- QueryInvoiceListSiebelCommsReqABCSImpl
- QueryInvoiceListSiebelCommsReqABCSImpl transforms the Siebel message into a QueryInvoiceList EBM and calls the provider to query the invoice list response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryInvoiceListBRMCommsProvABCSImpl

QueryInvoiceListBRMCommsProvABCSImpl transforms:

- QueryInvoiceListRequestEBM into BRM API input format and calls the API to query the invoice list output from the billing system.

  It then transforms the output from the API back to an Invoice EBM message and returns it to the calling requestor.

- QueryInvoice EBM into BRM API input formats and calls the APIs to Query the Invoice output from the billing system.

  It then transforms the output from the APIs back to an Invoice EBM message and returns it to the calling requestor.

- QueryInvoiceList EBM into BRM API input formats and calls the APIs to Query the Invoice Event Details output from the billing system.

  It then transforms the output from the APIs back to an Invoice EBM message and returns it to the calling requestor.

- Invoice EBM into BRM API input formats and calls the APIs to Query the Resource Impact output from the billing system.

  It then transforms the output from the APIs back to an Invoice EBM message and returns it to the calling requestor.

QueryInvoiceSiebelCommsReqABCSImpl

QueryInvoiceSiebelCommsReqABCSImpl transforms the Siebel message into QueryInvoiceEBM and calls the provider to query the invoice from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryInvoiceEventDetailsSiebelCommsReqABCSImpl

QueryInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the Siebel message into QueryInvoiceList EBM and calls the provider to query the invoice event details response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

SearchInvoiceEventDetailsSiebelCommsReqABCSImpl

SearchInvoiceEventDetailsSiebelCommsReqABCSImpl transforms the Siebel message into QueryInvoiceList EBM and calls the provider to query the Invoice Event Details response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

SearchInvoiceCharge supports searching invoice event details (CDR records) on the following columns in addition to the account ID and the item charge ID that are passed to Query Invoice Details:

- Minimum Amount
- Maximum Amount
- Start Date
- End Date

The BRM opcode PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT supports only simple queries, so the search supports passing only the data through and not passing complex query criteria (for example: >, <, between, and so on).

QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl

QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl transforms the Siebel message into an Invoice EBM and calls the provider to query the Resource Impact response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

5

Service Usage Integration Flow

This chapter provides an overview of the Service Usage integration flows and discusses Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Service Usage Integration Overview

The Service Usage integration between Siebel CRM and BRM supports the following integration scenarios:

- QueryServiceUsage enables a customer service representative (CSR) to view account-billing-profile-service-usage-summary in Siebel CRM.
- QueryServiceUsageAllocation enables a CSR to view account-billing-profile-service-usage-charge-details in Siebel CRM.
- SearchServiceUsageAllocation enables a CSR to search account-billing-profile-service-usage-charge-details based on a few columns.
- QueryServiceUsageAllocationResource enables a CSR to view resource balances (for example, nonmonetary) for service usage events in Siebel CRM.

Events that have been captured by BRM but have not been billed are called unbilled usage events. Customer questions about unbilled (service) usage are the most common queries for CSRs. This is because Wireless communications service providers (CSPs) offer service plans that include free usage per bill cycle.

Examples include:

- 1000 Free Local and Long Distance Minutes per month.
- 10 Free SMS per month.
- 1M of data download.

Unlike invoice information, unbilled usage information is constantly changing. The BRM has the most current status of all unbilled usage, so Siebel CRM must retrieve this information, in real time, so that the CSR can respond to customers' questions accurately.

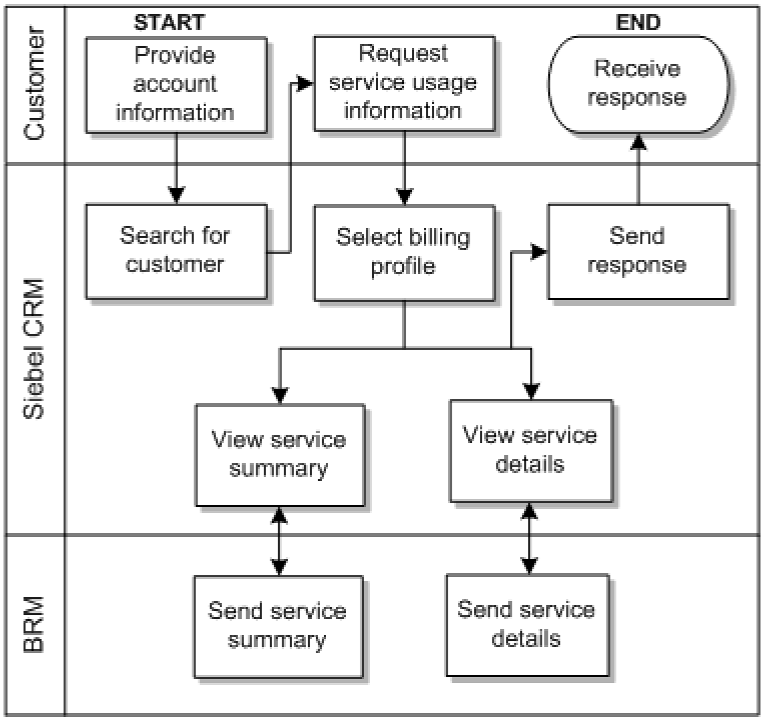
When the CSR clicks the Unbilled Usage tab on the Billing Profile screen, it triggers a request to retrieve item charges from the billing system. The integration process aggregates the item charge information to the services level so that service charges can be displayed in the Service Charges applet.

CSRs can request detailed unbilled data at the event (CDR) level. This is a separate request to BRM. Because this data is generally large, CSRs can limit the data returned by specifying search criteria.

Note: The date and time information of the event details and CDR, which is displayed in the Siebel UI, is specific to the time zone from where the event originated.

Figure 5–1 illustrates the View Service Usage flow:

Figure 5–1 The Service Usage Integration Flow



To view account billing profile service usage in Siebel CRM:

1. Navigate to the Accounts screen and query an account. Scroll down to the Billing Profile applet for this account and click the Billing Profile **Name** link.
2. Click the **Unbilled Usage** tab to view the account's billing profile service usage summary.
3. Click the **Net Amount** link for a particular item charge to view account billing profile service usage allocation (CDRs).
4. Click the **Search** tab to search account billing profile service usage charge details.
5. Click the **View Details** tab to view resource balances for service usage events.

See "Mapping Siebel Billing Management UI Elements to BRM Customer Center" for more information on mapping Siebel CRM elements to BRM elements.

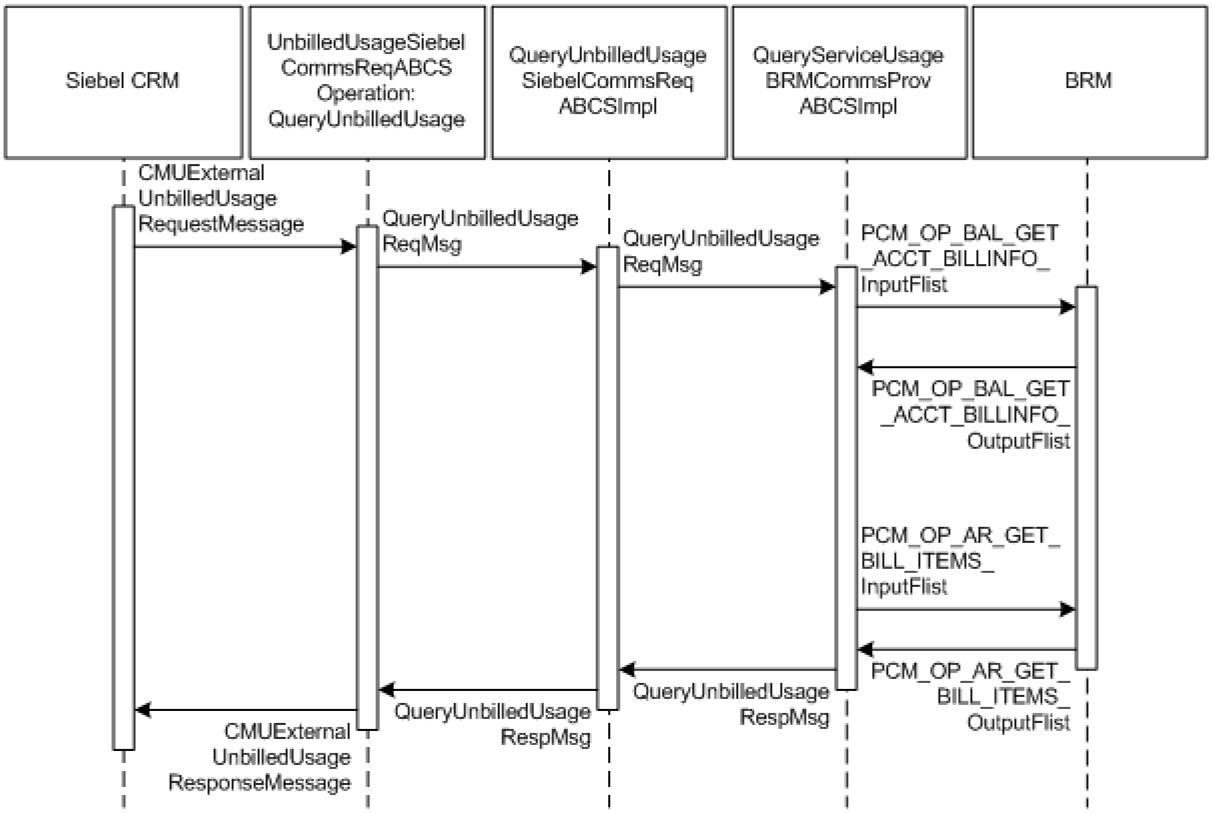
QueryServiceUsage Integration Flow

This integration flow uses the following interfaces:

- UnbilledUsageSiebelCommsReqABCS with operation QueryUnbilledUsage
- QueryUnbilledUsageSiebelCommsReqABCSImpl
- QueryServiceUsageBRMCommsProvABCSImpl

Figure 5–2 illustrates the QueryServiceUsage integration scenario.

Figure 5–2 QueryServiceUsage Integration Flow Sequence Diagram


Siebel CRM
UnbilledUsageSiebel CommsReqABCS Operation: QueryUnbilledUsage
QueryUnbilledUsage SiebelCommsReq ABCSImpl
QueryServiceUsage BRMCommsProv ABCSImpl
BRM
CMUExternal UnbilledUsage RequestMessage
QueryUnbilledUsage ReqMsg
QueryUnbilledUsage ReqMsg
PCM_OP_BAL_GET _ACCT_BILLINFO_ InputFlist
PCM_OP_BAL_GET _ACCT_BILLINFO_ OutputFlist
PCM_OP_AR_GET_ BILL_ITEMS_ InputFlist
PCM_OP_AR_GET_ BILL_ITEMS_ OutputFlist
QueryUnbilledUsage RespMsg
QueryUnbilledUsage RespMsg
CMUExternal UnbilledUsage ResponseMessage

When you initiate the QueryServiceUsage process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and click a billing profile for the account.

   This displays the Billing Profile BRM screen. When you click the Unbilled Usage tab for the billing profile, the Unbilled Usage screen opens and a web service UnbilledUsageSiebelCommsReqABCS call is made to get the unbilled usage details from the billing system.

2. Navigating to the Unbilled Usage tab invokes the UnbilledUsageSiebelCommsReqABCS web service, which in turn calls the UnbilledUsageSiebelCommsReqABCS with operation QueryUnbilledUsage.

   UnbilledUsageSiebelCommsReqABCS is a generic Siebel UnbilledUsage interface service with several operations defined.

3. Invoking UnbilledUsageSiebelCommsReqABCS with operation QueryUnbilledUsage routes the Siebel QueryUnbilledUsageReqMsg to the QueryUnbilledUsageSiebelCommsReqABCSImpl.

4. The QueryUnbilledUsageSiebelCommsReqABCSImpl first transforms the QueryUnbilledUsageReqMsg into QueryServiceUsageEBM and routes the QueryServiceUsageListReqMsg to the appropriate billing systems.

As delivered, QueryServiceUsageListReqMsg is routed to the QueryServiceUsageListBRMCommsProvABCSImpl.

5. QueryServiceUsageBRMCommsProvABCSImpl looks up QueryCode in enterprise business message (EBM) and transforms QueryServiceUsageListReqMsg into the input of PCM_OP_BAL_GET_ACCT_BILLINFO and calls the BRM opcode PCM_OP_BAL_GET_ACCT_BILLINFO.
6. The BRM application programming interface (API) PCM_OP_BAL_GET_ACCT_BILLINFO returns the list of bill infos of that account along with their AR_Billinfo to the calling QueryServiceUsageListBRMCommsProvABCSImpl.
7. This information is used to populate PCM_OP_AR_GET_BILL_ITEMS_inputFlist and calls the BRM opcode PCM_OP_AR_GET_BILL_ITEMS. The opcode returns the invoice header, service charges, and items in a flat message.
8. QueryServiceUsageBRMCommsProvABCSImpl then transforms the BRM API PCM_OP_AR_GET_BILL_ITEMS output into QueryServiceUsageListRespMsg and returns it to QueryUnbilledUsageSiebelCommsReqABCSImpl.
9. QueryUnbilledUsageSiebelCommsReqABCSImpl transforms the QueryUnbilledUsageRespMsg into QueryUnbilledUsageRespMsg, which is returned to UnbilledUsageSiebelCommsReqABCS.
10. UnbilledUsageSiebelCommsReqABCS returns the QueryUnbilledUsageRespMsg to the calling CMUUnbilledUsage as CMUUnbilledUsageResponseMessage.
11. The system writes the CMUUnbilledUsageResponseMessage to the Siebel unbilled usage Details virtual business component (VBC) for the user.

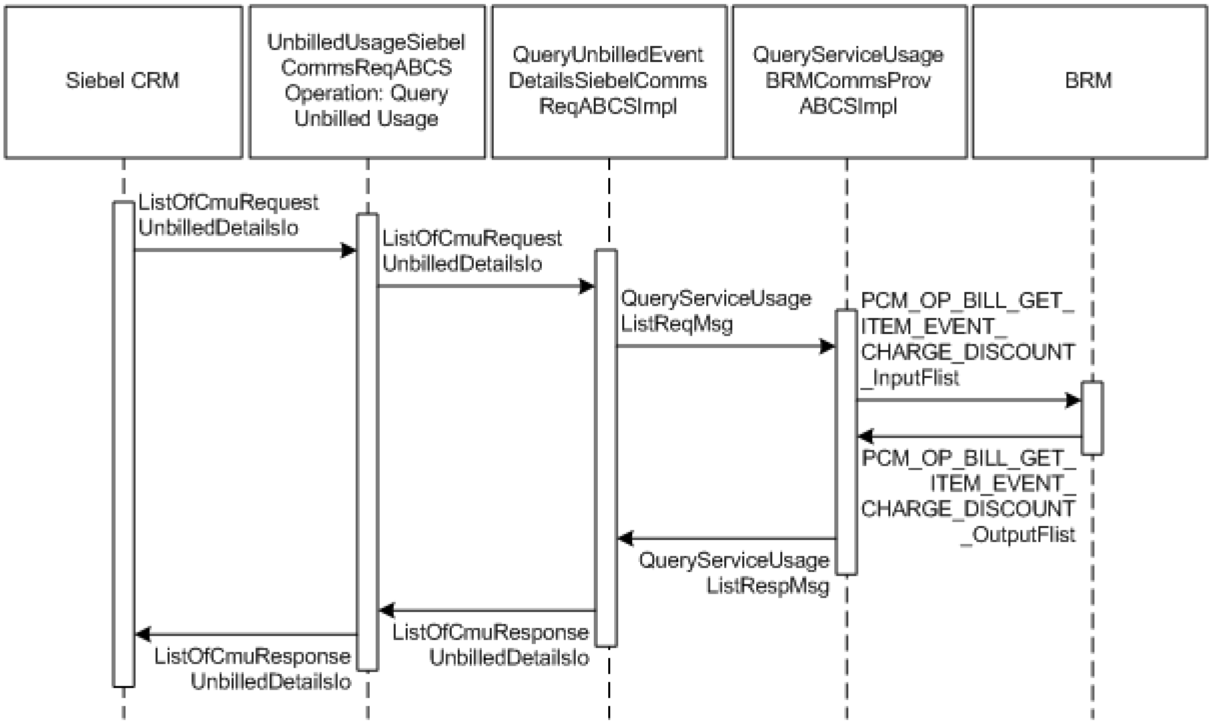
QueryServiceUsageAllocation Integration Flow

This integration flow uses the following interfaces:

- UnbilledUsageSiebelCommsReqABCS with operation QueryEventDetails
- QueryUnbilledEventDetailsSiebelCommsReqABCSImpl
- QueryServiceUsageBRMCommsProvABCSImpl

Figure 5–3 illustrates the QueryServiceUsageAllocation integration scenario.

Figure 5–3 QueryServiceUsageAllocation Integration Flow Sequence Diagram


Siebel CRM
UnbilledUsageSiebel CommsReqABCS Operation: Query Unbilled Usage
QueryUnbilledEvent DetailsSiebelComms ReqABCSImpl
QueryServiceUsage BRMCommsProv ABCSImpl
BRM
ListOfCmuRequest UnbilledDetailsIo
ListOfCmuRequest UnbilledDetailsIo
QueryServiceUsage ListReqMsg
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _InputFlist
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _OutputFlist
QueryServiceUsage ListRespMsg
ListOfCmuResponse UnbilledDetailsIo
ListOfCmuResponse UnbilledDetailsIo

When you initiate the QueryServiceUsageAllocation process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and click the Billing Profile tab.

   This displays the Billing Profile screen. Click the Unbilled Usage tab to open the Unbilled Usage screen.

2. Clicking the Net Amount link for a particular item charge invokes the UnbilledUsageSiebelCommsReqABCS web service, which in turn calls the UnbilledUsageSiebelCommsReqABCS with operation QueryEventDetails.

   UnbilledUsageSiebelCommsReqABCS is a generic Siebel UnbilledUsage interface service with several operations defined.

3. Invoking UnbilledUsageSiebelCommsReqABCS with operation QueryEventDetails routes the Siebel QueryEventDetailsReqMsg to the QueryUnbilledEventDetailsSiebelCommsReqABCSImpl.

4. The QueryUnbilledEventDetailsSiebelCommsReqABCSImpl transforms the QueryEventDetailsReqMsg into QueryServiceUsageListEBM and routes the QueryServiceUsageListReqMsg to the appropriate billing systems.

   As delivered, QueryServiceUsageListReqMsg is routed to the QueryServiceUsageListBRMCommsProvABCSImpl.

5. QueryServiceUsageBRMCommsProvABCSImpl looks up the value of QueryCode and transforms QueryServiceUsageListReqMsg into the input of PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT and calls the opcode PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT.

6. API PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT returns the Item Charge Details or Events output to the calling QueryServiceUsageBRMCommsProvABCSImpl.

7. QueryServiceUsageListBRMCommsProvABCSImpl then transforms the BRM API output into QueryServiceUsageListRespMsg and returns it to QueryUnbilledEventDetailsSiebelCommsReqABCSImpl.
8. QueryUnbilledEventDetailsSiebelCommsReqABCSImpl transforms the QueryServiceUsageListRespMsg into QueryEventDetailsRespMsg, which is returned to UnbilledUsageSiebelCommsReqABCS.
9. UnbilledUsageSiebelCommsReqABCS returns the QueryEventDetailsRespMsg to the calling UnbilledUsageSiebelCommsReqABCS as CMUUnbilledDetailsResponseMessage.
10. The system writes the CMUUnbilledDetailsResponseMessage to the Siebel Unbilled Item Charge Details VBC for the user.

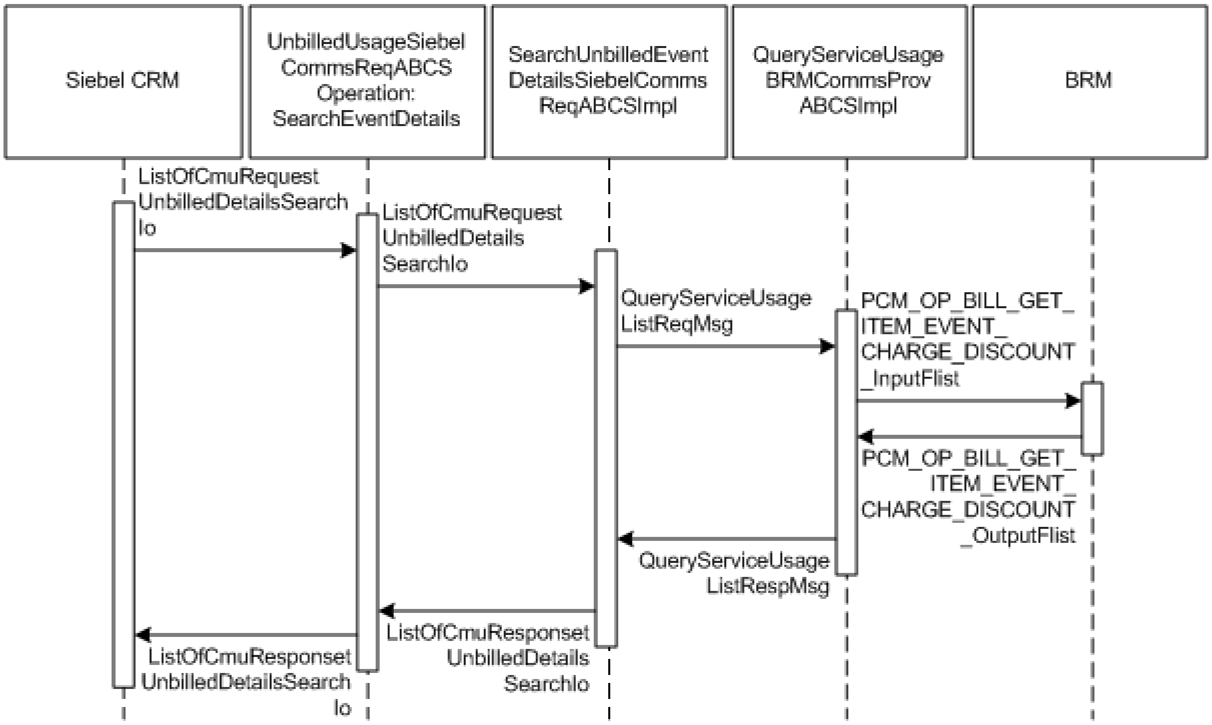
SearchServiceUsageAllocation Integration Flow

This integration flow uses the following interfaces:

- UnbilledUsageSiebelCommsReqABCS with operation SearchEventDetails
- SearchUnbilledEventDetailsSiebelCommsReqABCSImpl
- QueryServiceUsageBRMCommsProvABCSImpl

Figure 5–4 illustrates the SearchServiceUsageAllocation integration scenario.

Figure 5–4 SearchServiceUsageAllocation Integration Flow Sequence Diagram


Siebel CRM
UnbilledUsageSiebel CommsReqABCS Operation: SearchEventDetails
SearchUnbilledEvent DetailsSiebelComms ReqABCSImpl
QueryServiceUsage BRMCommsProv ABCSImpl
BRM
ListOfCmuRequest UnbilledDetailsSearch Io
ListOfCmuRequest UnbilledDetails SearchIo
QueryServiceUsage ListReqMsg
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _InputFlist
PCM_OP_BILL_GET_ ITEM_EVENT_ CHARGE_DISCOUNT _OutputFlist
QueryServiceUsage ListRespMsg
ListOfCmuResponset UnbilledDetails SearchIo
ListOfCmuResponset UnbilledDetailsSearch Io

When you initiate the SearchServiceUsageAllocation process, the following events occur:

1. In Siebel CRM, navigate to the Accounts screen, query an account, and click the Billing Profile tab.

This displays the Billing Profile screen. Click the Unbilled Usage tab to access the Unbilled Usage screen. Click the Net Amount link for an item charge. The event details for that item charge displays on the event details applet. Click the Search button to open a separate applet to specify search criteria. After entering values for these fields, click the Go button.

2. Clicking Go invokes the UnbilledUsageSiebelCommsReqABCS web service, which in turn calls UnbilledUsageSiebelCommsReqABCS with operation SearchEventDetails.

   UnbilledUsageSiebelCommsReqABCS is a generic Siebel UnbilledUsage interface service with several operations defined.

3. Invoking UnbilledUsageSiebelCommsReqABCS with operation SearchEventDetails routes the Siebel SearchEventDetailsReqMsg to SearchUnbilledEventDetailsSiebelCommsReqABCSImpl.

4. SearchUnbilledEventDetailsSiebelCommsReqABCSImpl transforms SearchEventDetailsReqMsg into QueryServiceUsageEBM and routes the QueryServiceUsageListReqMsg to the appropriate billing systems.

   As delivered, QueryServiceUsageListReqMsg is routed to QueryServiceUsageListBRMCommsProvABCSImpl.

5. QueryServiceUsageBRMCommsProvABCSImpl transforms QueryServiceUsageListReqMsg into the input of PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT and calls the BRM opcode PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT.

6. The BRM API PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT returns the Item Charge Details output to the calling QueryServiceUsageBRMCommsProvABCSImpl.

7. QueryServiceUsageBRMCommsProvABCSImpl then transforms the BRM API output into QueryServiceUsageListRespMsg and returns it to SearchUnbilledUsageEventDetailsSiebelCommsReqABCSImpl.

8. SearchUnbilledEventDetailsSiebelCommsReqABCSImpl then transforms the QueryServiceUsageListRespMsg into SearchEventDetailsRespMsg, which is returned to UnbilledUsageSiebelCommsReqABCS.

9. UnbilledUsageSiebelCommsReqABCS returns the SearchEventDetailsRespMsg to the calling Siebel web service UnbilledUsageSiebelCommsReqABCS as CMUUnbilledDetailsSearchResponseMessage.

10. The system writes the CMUUnbilledDetailsSearchResponseMessage to the Siebel Unbilled Item Charge Details VBC for the user.

SearchServiceUsageAllocation supports searching service usage item details (CDR records) on the following columns in addition to the account ID and item charge ID that are passed to SearchEventDetails:

- Minimum Amount
- Maximum Amount
- Start Date
- End Date

The BRM opcode PCM_OP_ BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT supports only simple queries and passes only the data through. It does not pass complex query criteria (for example, >, <, between, and so on).

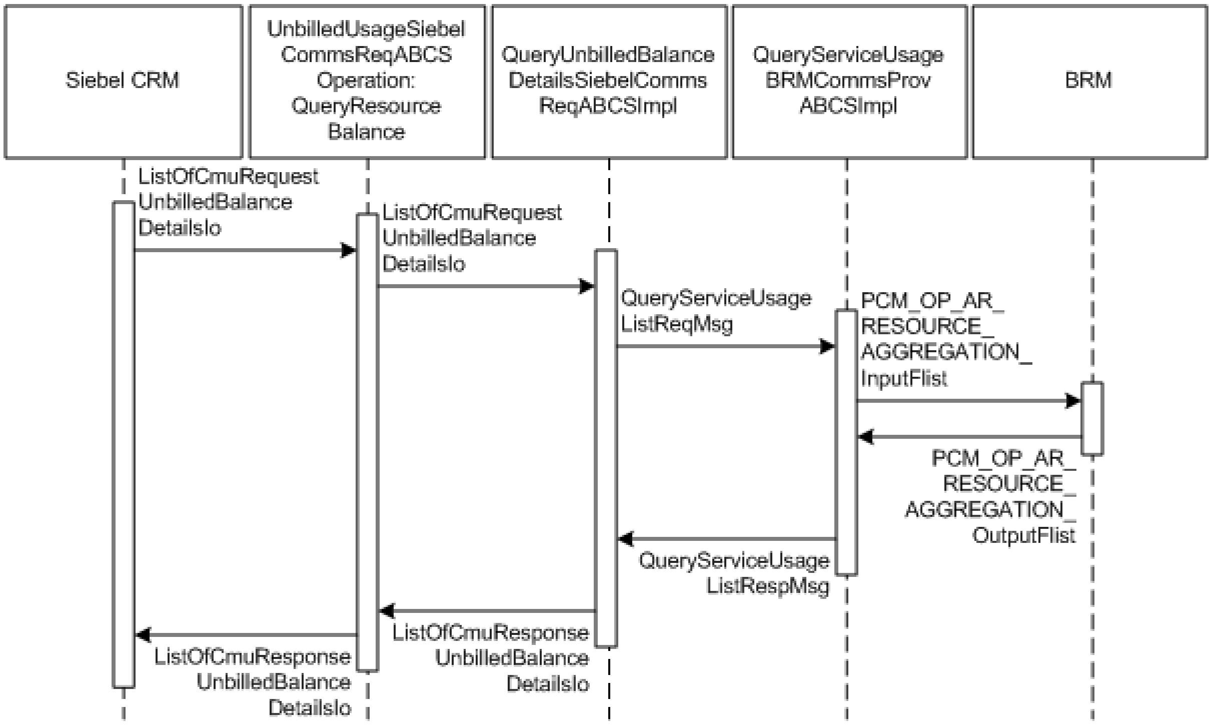
QueryServiceUsageAllocationResource Integration Flow

This integration flow uses the following interfaces:

- UnbilledUsageSiebelCommsReqABCS with operation QueryBalanceDetails
- QueryUnbilledBalanceDetailsSiebelCommsReqABCSImpl
- QueryServiceUsageBRMCommsProvABCSImpl

Figure 5–5 illustrates the QueryServiceUsageAllocationResource integration scenario.

Figure 5–5 QueryServiceUsageAllocationResource Integration Flow Sequence Diagram


Siebel CRM
UnbilledUsageSiebel CommsReqABCS Operation: QueryResource Balance
QueryUnbilledBalance DetailsSiebelComms ReqABCSImpl
QueryServiceUsage BRMCommsProv ABCSImpl
BRM
ListOfCmuRequest UnbilledBalance DetailsIo
ListOfCmuRequest UnbilledBalance DetailsIo
QueryServiceUsage ListReqMsg
PCM_OP_AR_ RESOURCE_ AGGREGATION_ InputFlist
PCM_OP_AR_ RESOURCE_ AGGREGATION_ OutputFlist
QueryServiceUsage ListRespMsg
ListOfCmuResponse UnbilledBalance DetailsIo
ListOfCmuResponse UnbilledBalance DetailsIo

When you initiate the QueryServiceUsageAllocationResource process, the following events occur:

1. In Siebel CRM Query Account, navigate to the Account Summary screen, and drill down on the Billing Profile Name in the Billing Profile applet.

   This displays the Billing Profile screen. Click the Unbilled Usage tab, and then click the Net Amount link for an item charge. This accesses the Event Details applet for that item charge.

2. Clicking View Details for a particular nonmonetary event invokes the outbound web service UnbilledUsageSiebelCommsReqABCS to query resource balance details for a particular event from the billing system, which in turn calls UnbilledUsageSiebelCommsReqABCS with operation QueryResourceBalance.
3. Invoking UnbilledUsageSiebelCommsReqABCS with operation QueryBalanceDetails routes the QueryBalanceDetailsReqMsg to QueryUnbilledBalanceDetailsSiebelCommsReqABCSImpl.
4. QueryUnbilledBalanceDetailsSiebelCommsReqABCSImpl transforms the QueryBalanceDetailsReqMsg into a Service Usage EBM and routes the QueryServiceUsageListReqMsg to the appropriate billing systems.

As delivered, QueryServiceUsageListReqMsg is routed to QueryServiceUsageListBRMCommsProvABCSImpl.

5. QueryServiceUsageBRMCommsProvABCSImpl transforms QueryServiceUsageListReqMsg into the input of PCM_OP_ AR_RESOURCE_ AGGREGATION and calls the BRM API PCM_OP_ AR_RESOURCE_ AGGREGATION.
6. The BRM API PCM_OP_ AR_RESOURCE_AGGREGATION returns the resource balance records for a particular event as part of PCM_OP_AR_RESOURCE_ AGGREGATION_outputFlist to QueryServiceUsageBRMCommsProvABCSImpl.
7. QueryServiceUsageBRMCommsProvABCSImpl then transforms the API output PCM_OP_ AR_RESOURCE_AGGREGATION_RespMsg into QueryServiceUsageListRespMsg and returns it to QueryUnbilledUsageBalanceDetailsSiebelCommsReqABCSImpl.
8. QueryUnbilledBalanceDetailsSiebelCommsReqABCSImpl transforms the QueryServiceUsageListRespMsg into QueryBalanceDetailsRespMsg, which is returned to UnbilledUsageSiebelCommsReqABCS.
9. UnbilledUsageSiebelCommsReqABCS returns the QueryBalanceDetailsRespMsg to the calling Siebel web service.
10. The system writes the message to the Siebel Balance Total VBC for the user.

BRM Interfaces

The QueryServiceUsage integration flow uses this opcode:

- PCM_OP_BAL_GET_ACCT_BILLINFO

The QueryServiceUsageAllocation integration flow uses this opcode:

- PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT

The SearchServiceUsageAllocation integration flow uses this opcode:

- PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT

The QueryServiceUsageAllocationResource integration flow uses this opcode:

- PCM_OP_AR_RESOURCE_AGGREGATION

See *Oracle Communications Billing and Revenue Management Opcode Flist Reference* for more information.

Siebel CRM Interfaces

The service usage integration flow uses this Siebel CRM interface:

- UnbilledUsageSiebelCommsReqABCS

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The Service Usage integration uses the following delivered Industry Oracle AIA components:

- ServiceUsageEBO

- QueryServiceUsageListEBM
- QueryServiceUsageListRequestEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the AIA Reference Doc link.

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

Integration Services

These services are delivered with the Service Usage integration flow:

- UnbilledUsageSiebelCommsReqABCS with operations QueryUnbilledUsage, QueryEventDetails, SearchEventDetails, and QueryBalanceDetails
- QueryUnbilledUsageSiebelCommsReqABCSImpl
- QueryServiceUsageBRMCommsProvABCSImpl
- QueryUnbilledEventDetailsSiebelCommsReqABCSImpl
- SearchUnbilledEventDetailsSiebelCommsReqABCSImpl
- QueryUnbilled7yBalanceDetailsSiebelCommsReqABCSImpl

UnbilledUsageSiebelCommsReqABCS

UnbilledUsageSiebelCommsReqABCS exposes the following operations:

- QueryUnbilledUsage:
  - Routes QueryUnbilledUsageReqMsg to the requester implementation service
  - Routes QueryUnbilledUsageRespMsg to the requester
- QueryEventDetails:
  - Routes QueryEventDetailsReqMsg to the requester implementation service
  - Routes QueryEventDetailsRespMsg to the requester
- SearchEventDetails:

  - Routes SearchEventDetailsReqMsg to the requester implementation service
  - Routes SearchEventDetailsRespMsg to the requester
- QueryBalanceDetails:
  - Routes QueryBalanceDetailsReqMsg to the requester implementation service
  - Routes QueryBalanceDetailsRespMsg to the requester

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

QueryUnbilledUsageSiebelCommsReqABCSImpl

This service transforms the Siebel message into a QueryServiceUsageListRequest EBM and calls the provider to get the QueryServiceUsage response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

QueryServiceUsageListBRMCommsProvABCSImpl

QueryServiceUsageListBRMCommsProvABCSImpl transforms:

- ServiceUsageEBM into BRM API input format and calls the API to get the service items output from the billing system.

  It then transforms the output from the API back to a ServiceUsage EBM message and returns it to the calling requestor.

- QueryServiceUsageListEBM into BRM API input formats and calls the APIs to query the resource balance output from the billing system.

  It then transforms the output from the APIs back to a ServiceUsage EBM message and returns it to the calling requestor.

QueryUnbilledUsageEventDetailsSiebelCommsReqABCSImpl

QueryUnbilledUsageEventDetailsSiebelCommsReqABCSImpl transforms the QueryEventDetailsReqMsg into QueryServiceUsageListEBM.

SearchUnbilledUsageEventDetailsSiebelCommsReqABCSImpl

SearchUnbilledUsageEventDetailsSiebelCommsReqABCSImpl transforms SearchEventDetailsReqMsg into QueryServiceUsageEBM.

QueryUnbilledUsageBalanceDetailsSiebelCommsReqABCSImpl

QueryUnbilledUsageBalanceDetailsSiebelCommsReqABCSImpl transforms the Siebel message into a QueryServiceUsageListEBM and calls the provider to query the resource balance response from the billing system. It then transforms the EBM response back to a Siebel message and returns it to the calling Siebel web service.

6 Payment Integration Flow

This chapter provides an overview of the payment integration flows and discusses Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM) interfaces, communications industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Payment Integration Overview

The payment integration between Siebel CRM and BRM supports the following integration scenarios:

- CreateReceivedPayment lets you capture a payment in Siebel CRM either for an account at the billing profile level or at the invoice level and to post the payment in BRM.
- QueryReceivedPaymentList enables lets you view the history of payments in Siebel CRM at both the billing profile-level and invoice-level by retrieving payment records from BRM.
- SearchPayment lets you search for payment records in BRM to display in Siebel CRM for an account at the billing profile-level or at the invoice-level.

Capturing Payments

Your customers can make payments in many ways: using a credit card or debit card over the phone, using a credit or debit card at a self-service location, using a check by mail, or using an electronic payment from a bank account.

The integration accepts payments from these channels in two ways:

- From Siebel CRM: a customer service representative captures payment information in Siebel CRM
- From the cross-channel system directly: payment information is captured the cross-channel system such as customer self-service, retail locations, web commerce, and partner systems

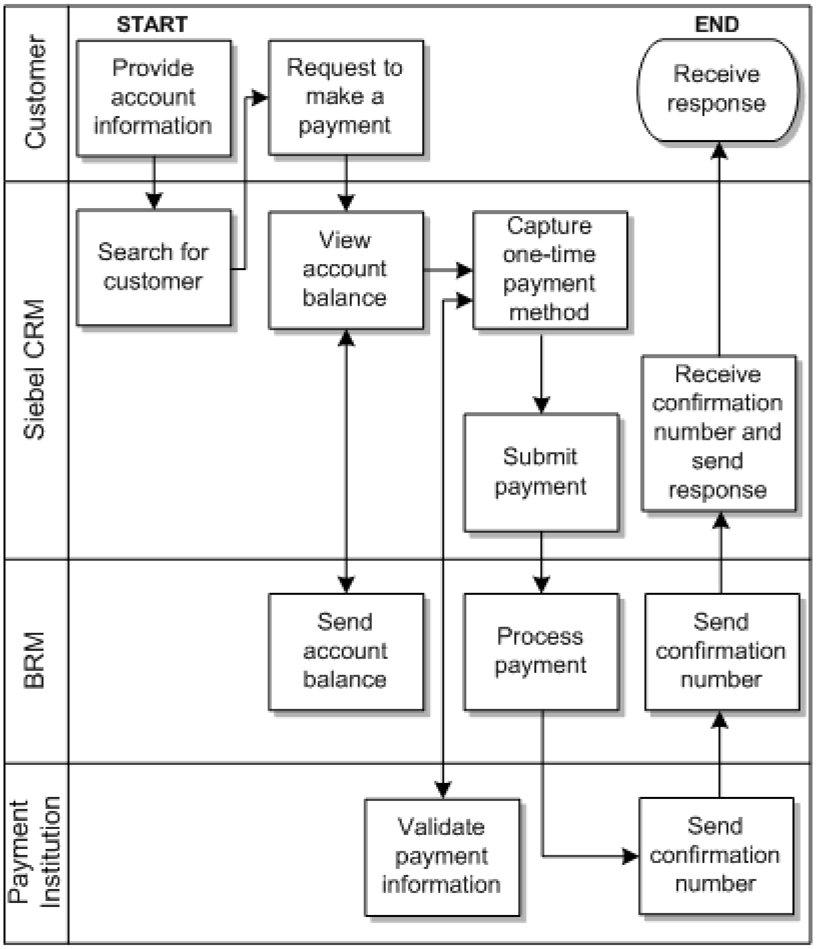
Capturing Payments in Siebel CRM

You capture one-time payments in Siebel CRM and send them to BRM for processing. BRM processes the payments and returns confirmation to Siebel CRM, which your customers can keep for their records.

You can accept one-time payments from self-paying accounts and non-paying subordinate accounts. All payments for the non-paying account are paid by a parent account. You capture one-time payments for non-paying accounts on the parent account's billing profile screen.

Figure 6–1 illustrates the flow for capturing payments in Siebel CRM.

Figure 6–1 The Capturing Payments Integration Flow



To view the history of payments and to capture new payments in Siebel CRM:

1. Navigate to the Accounts screen and query an account. Scroll down to the billing profile applet for this account and click the billing profile **Name** link.
2. Click the **Payments** tab to view the history of payments.
3. Click **New** to capture a new payment for this account.
4. From the Payments tab, click **Search** to open the Search applet and search for specific payment records.

See "Mapping Siebel Billing Management UI Elements to BRM Customer Center" for more information on mapping Siebel CRM elements to BRM elements.

Capturing Cross-Channel Payments

You must customize the integration to accept one-time payments from cross-channel systems other than Siebel CRM. See "Configuring Cross-Channel Payments" for more information.

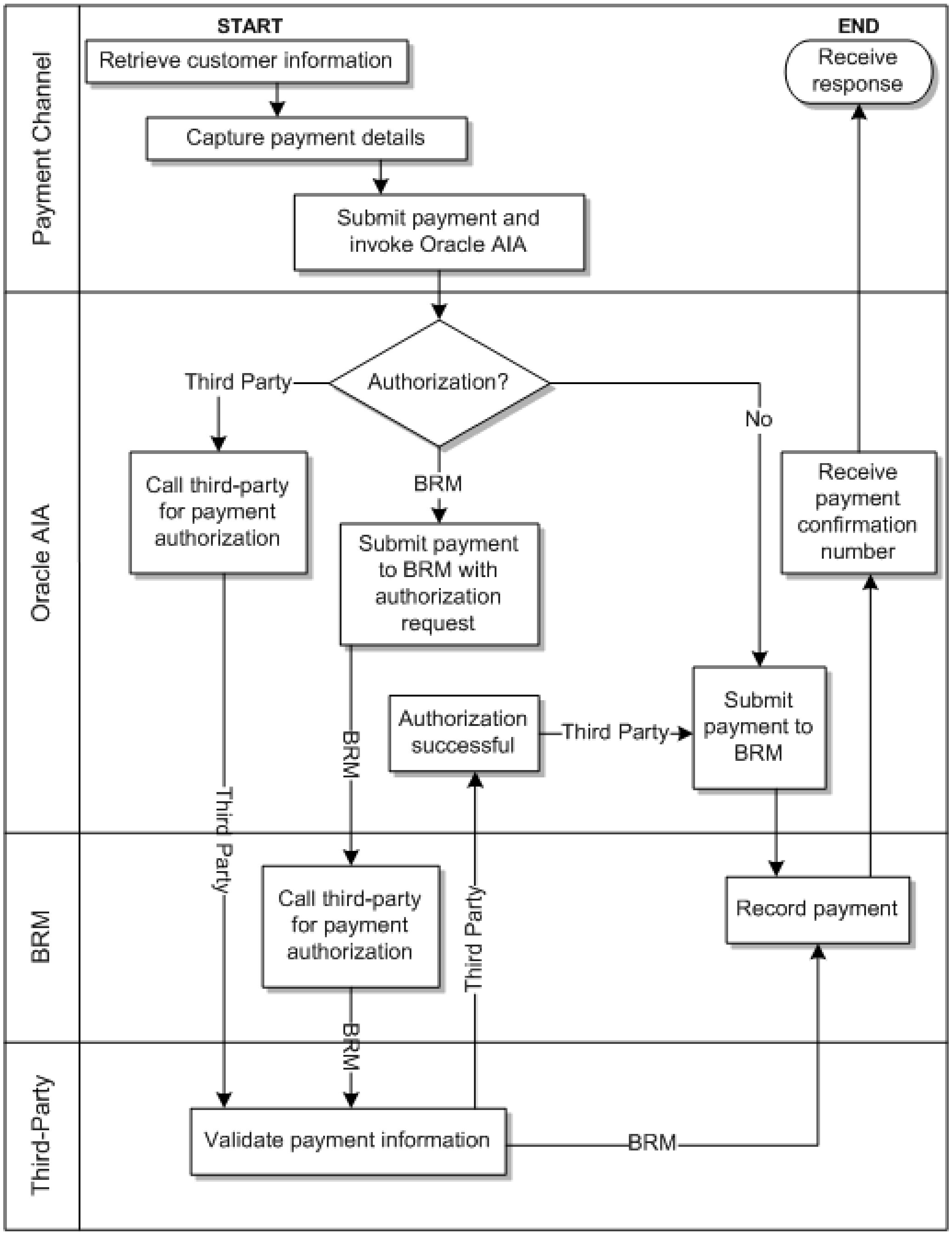
When you have customized the integration, you can capture one-time payments directly in the cross-channel system, such as a retail point of sale or online self-service, and send the payments through the integration to BRM for processing. You have three options for authorizing payments:

- BRM: BRM communicates with a third-party authorization system.
- Third-party: the integration communicates with a third-party authorization system.
- External: an external system, such as a retail point of sale system, authorizes payments and BRM records the payment immediately.

When payments are successfully authorized and recorded in BRM, the integration sends a confirmation message to the cross-channel system. When authorization or BRM processing fails, the integration sends an error message to the cross-channel system. You can customize the integration to send the confirmation or error message to Siebel CRM as well.

Figure 6–2 illustrates the flow for capturing payments from cross-channel systems.

Figure 6–2 Capturing Payments from Cross-Channel Systems


Payment Channel
Oracle AIA
BRM
Third-Party
START
Retrieve customer information
Capture payment details
Submit payment and invoke Oracle AIA
END
Receive response
Authorization?
Third Party
No
BRM
Call third-party for payment authorization
Submit payment to BRM with authorization request
Receive payment confirmation number
Authorization successful
Third Party
Submit payment to BRM
BRM
Third Party
Call third-party for payment authorization
Third Party
Record payment
BRM
Validate payment information
BRM

Viewing Payment History

Regardless of how a payment is made, it must be displayed in the payment history retrieved from the billing system into Siebel CRM. BRM maintains payment information for a customer for a fixed period which varies by your legal requirements and business needs.

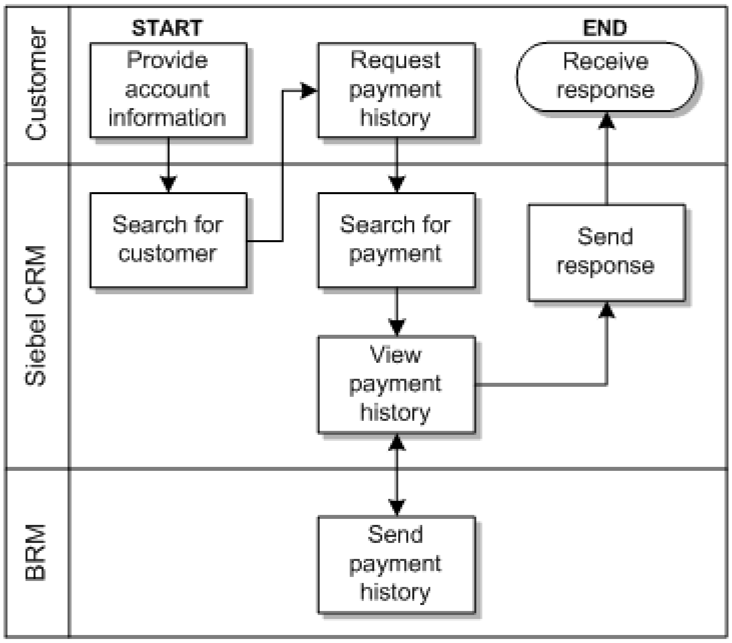
During a sales or service process, a customer may want to know his payment history. You can query a customer's past payments to respond to queries from the Payments tab of the Siebel CRM Billing Profile screen, triggering a request to the billing system to return the latest n payments. You can also specify search criteria to find the correct payment record if the initial list of payments is not the right one. The Payment History view also indicates the user of any payment reversals that might have been made on a payment that has been previously processed and allocated.

Searching for Payments

The Search Payment functionality lets you search for specific payment records in BRM based on the customer request and displays these in Siebel CRM. The search is performed for an account at the billing profile level or at the invoice level. You search for payments on either a date range or a paid amount.

Figure 6–3 illustrates the flow for Viewing Payment History.

Figure 6–3 The View Payment History Integration Flow



CreateReceivedPayment Integration Flow

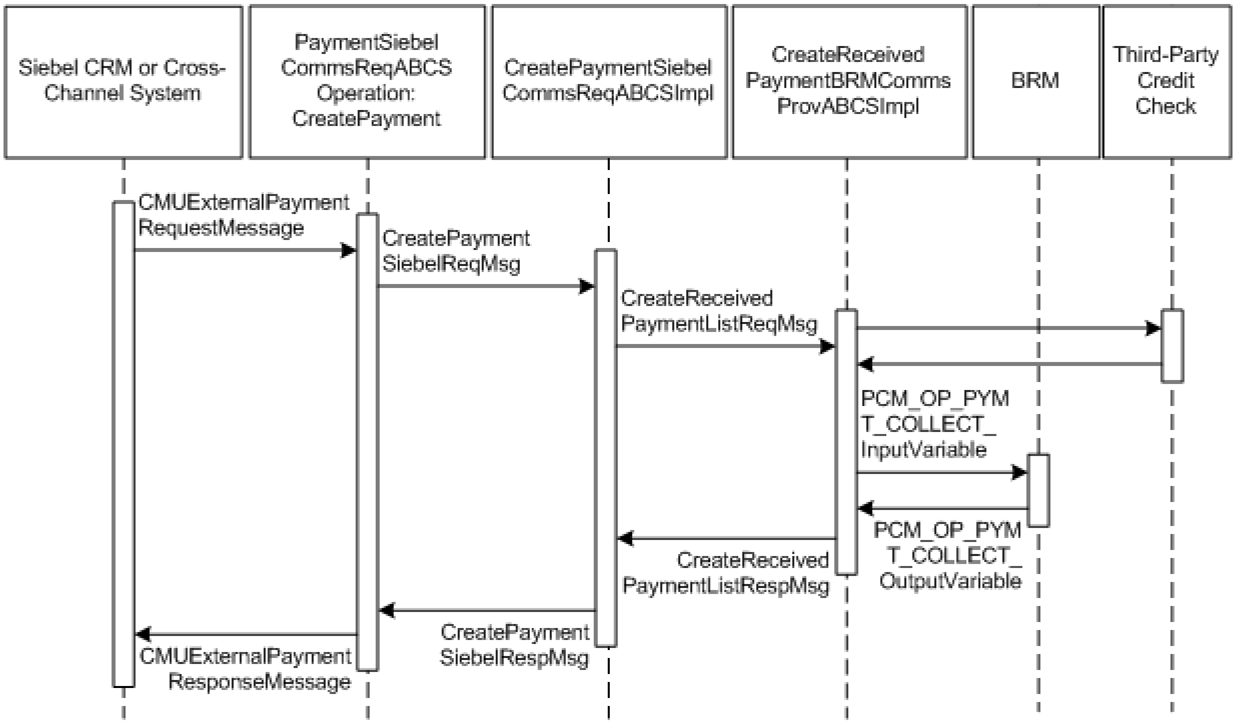
This integration flow uses the following interfaces:

- PaymentSiebelCommsReqABCS with operations CreatePayment and CreateInvoicePayment
- CreatePaymentSiebelCommsReqABCSImpl
- CreateInvoicePaymentSiebelCommsReqABCSImpl
- CreateReceivedPaymentBRMCommsProvABCSImpl

Note: The integration flow for CreateInvoicePayment is similar to the integration flow for CreatePayment.

Figure 6–4 illustrates the CreateReceivedPayment integration flow.

Figure 6–4 CreateReceivedPayment Integration Flow Sequence Diagram


Siebel CRM or Cross-Channel System
PaymentSiebel CommsReqABCS Operation: CreatePayment
CreatePaymentSiebel CommsReqABCSImpl
CreateReceived PaymentBRMComms ProvABCSImpl
BRM
Third-Party Credit Check
CMUExternalPayment RequestMessage
CreatePayment SiebelReqMsg
CreateReceived PaymentListReqMsg
PCM_OP_PYM T_COLLECT_ InputVariable
PCM_OP_PYM T_COLLECT_ OutputVariable
CreateReceived PaymentListRespMsg
CreatePayment SiebelRespMsg
CMUExternalPayment ResponseMessage

The CreateReceivedPayment integration flow occurs as follows:

1. You initiate the CreateReceivedPayment process by submitting a one-time payment for a billing profile from Siebel CRM or a payment channel. See the discussion of recording profile-level payments in *Siebel Communications Guide* for more information.
2. Siebel CRM or the channel invokes the PaymentSiebelCommsReqABCS web service with the operation **CreatePayment**.
3. The integration routes the CreatePaymentSiebelReqMsg message to the CreatePaymentSiebelCommsReqABCSImpl service.
4. The CreatePaymentSiebelCommsReqABCSImpl service transforms the CreatePaymentReqMsg into the CreateReceivedPaymentListReqMsg enterprise business message (EBM) and routes it to the appropriate billing system.

   By default, the CreateReceivedPaymentListReqMsg EBM is sent to the CreateReceivedPaymentBRMCommsProvABCSImpl service to be routed to BRM.
5. If the CreateReceivedPaymentListReqMsg EBM indicates that a third-party credit check is required (the **OneTimePayment** property is set to **T**), the CreateReceivedPaymentBRMCommsProvABCSImpl service initiates the third-party credit check and receives the authorization response.
6. If the third-party credit check is successful or was not required, the CreateReceivedPaymentBRMCommsProvABCSImpl service transforms the CreateReceivedPaymentListReqMsg EBM into the input of PCM_OP_PYMT_COLLECT and calls the BRMPymtServices web service with the PCM_OP_PYMT_COLLECT opcode.

   When calling the PCM_OP_PYMT_COLLECT opcode, the CreateReceivedPaymentBRMCommsProvABCSImpl service sets the value of the **PIN_FLD_COMMAND** field as follows:

- If the **OneTimePayment** property in the EBM is set to **Y**, the **PIN_FLD_COMMAND** field is set to **4**.
- If the **OneTimePayment** property in the EBM is set to **O** or **T**, the **PIN_FLD_COMMAND** field is set to **0**.

  See "Configuring the Message from Cross-Channel Systems" for more information about the **OneTimePayment** property.

If the third-party credit check is not successful, a BPEL exception is thrown and a failure response message is returned to Siebel CRM or the cross-channel system.

7. If the **PIN_FLD_COMMAND** field is set to **4**, the BRM initiates the payment authorization with a third-party system and receives the response.
8. If the BRM authorization is successful or was not required, the BRMPymtServices web service calls the PCM_OP_PYMT_COLLECT opcode and returns the payment object output to the CreateReceivedPaymentBRMCommsProvABCSImpl service.

   If the BRM authorization is not successful, a BPEL exception is thrown and a failure response message is returned to Siebel CRM or the cross-channel system
9. The CreateReceivedPaymentBRMCommsProvABCSImpl service transforms the BRM API output into a CreateReceivedPaymentListRespMsg EBM and returns it to the CreatePaymentSiebelCommsReqABCSImpl service.
10. The CreatePaymentSiebelCommsReqABCSImpl service transforms the CreatePaymentListRespMsg EBM into a CreatePaymentSiebelRespMsg message, and returns it to the PaymentSiebelCommsReqABCS service.
11. The PaymentSiebelCommsReqABCS service returns the CreatePaymentSiebelRespMsg message to the CMUCreatePayment service as a CMUCreatePaymentResponseMessage message.
12. The system writes the CMUCreatePaymentResponseMessage to Siebel CRM or the payment channel and displays the payment confirmation number.

QueryReceivedPaymentList Integration Flow

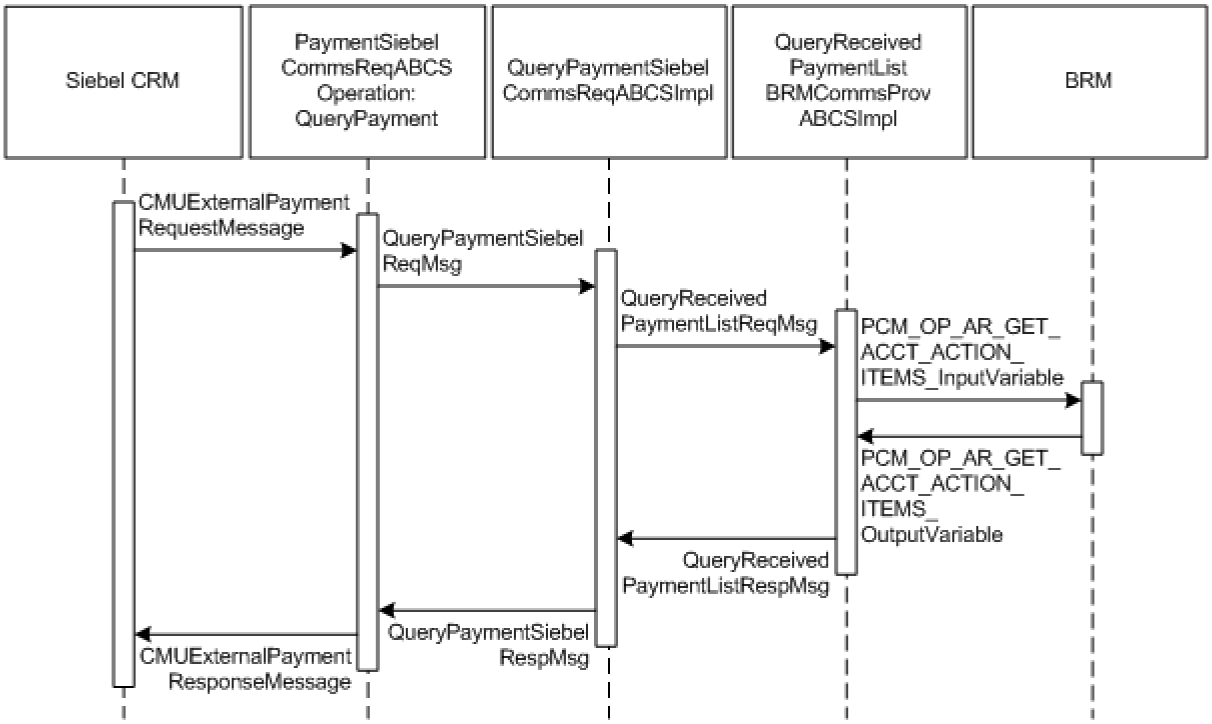
The QueryReceivedPaymentList integration flow uses the following interfaces:

- ReceivedPaymentSiebelCommsReqABCS Interface with operations QueryPayment, SearchPayment, and QueryInvoicePayment
- QueryPaymentSiebelCommsReqABCSImpl
- QueryInvoicePaymentSiebelCommsReqABCSImpl
- SearchPaymentSiebelCommsReqABCSImpl
- QueryReceivedPaymentListBRMCommsProvABCSImpl

> **Note:** The QueryInvoicePayment integration flow is similar to the QueryPayment integration flow except that a different ABCSImpl is used.

Figure 6–5 illustrates the QueryReceivedPaymentList integration scenario.

Figure 6–5 QueryReceivedPaymentList Integration Flow Sequence Diagram


Siebel CRM
PaymentSiebel CommsReqABCS Operation: QueryPayment
QueryPaymentSiebel CommsReqABCSImpl
QueryReceived PaymentList BRMCommsProv ABCSImpl
BRM
CMUExternalPayment RequestMessage
QueryPaymentSiebel ReqMsg
QueryReceived PaymentListReqMsg
PCM_OP_AR_GET_ ACCT_ACTION_ ITEMS_InputVariable
PCM_OP_AR_GET_ ACCT_ACTION_ ITEMS_ OutputVariable
QueryReceived PaymentListRespMsg
QueryPaymentSiebel RespMsg
CMUExternalPayment ResponseMessage

When you initiate the QueryReceivedPaymentList process, the following events occur:

1. In Siebel CRM, navigate to the Billing Profile screen.
2. Navigate to the Accounts screen, query an account, and click a billing profile for the account.

   On the Billing Profile screen, click the Payments tab for the billing profile. The Payments screen has two buttons, one to create a payment and one to search for payments. Navigating to the Payments tab invokes the PaymentSiebelCommsReqABCS web service, which in turn calls PaymentSiebelCommsReqABCS with operation QueryPayment.
3. Invoking PaymentSiebelCommsReqABCS with operation QueryPayment routes the Siebel QueryPaymentReqMsg to the QueryPaymentSiebelCommsReqABCSImpl.
4. The QueryPaymentSiebelCommsReqABCSImpl transforms the QueryPaymentReqMsg into QueryReceivedPaymentListEBM and routes the QueryReceivedPaymentListEBM to the appropriate billing system.

   As delivered, QueryReceivedPaymentListEBM is routed to QueryReceivedPaymentListBRMCommsProvABCSImpl.
5. QueryReceivedPaymentListBRMCommsProvABCSImpl transforms QueryReceivedPaymentListReqMsg into the input of PCM_OP_AR_GET_ACCT_ACTION_ITEMS and calls BRMARService with operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS.
6. Invoking BRMARService with operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS invokes the API PCM_OP_AR_GET_ACCT_ACTION_ITEMS and returns the payment object output to QueryReceivedPaymentListBRMCommsProvABCSImpl.

7. QueryReceivedPaymentListBRMCommsProvABCSImpl then transforms the BRM API output into QueryReceivedPaymentListResponseEBM and returns it to QueryReceivedPaymentListSiebelABCSImpl.
8. QueryReceivedPaymentListSiebelABCSImpl transforms the QueryReceivedPaymentListResponseEBM into QueryPaymentRespMsg, which is returned to PaymentSiebelCommsReqABCS.
9. ReceivedPaymentSiebelCommsReqABCS returns the QueryPaymentRespMsg to CMUQueryPayment as CMUQueryPaymentResponseMessage.
10. The system writes the CMUQueryPaymentResponseMessage to the Siebel Unbilled Details VBC for the user.

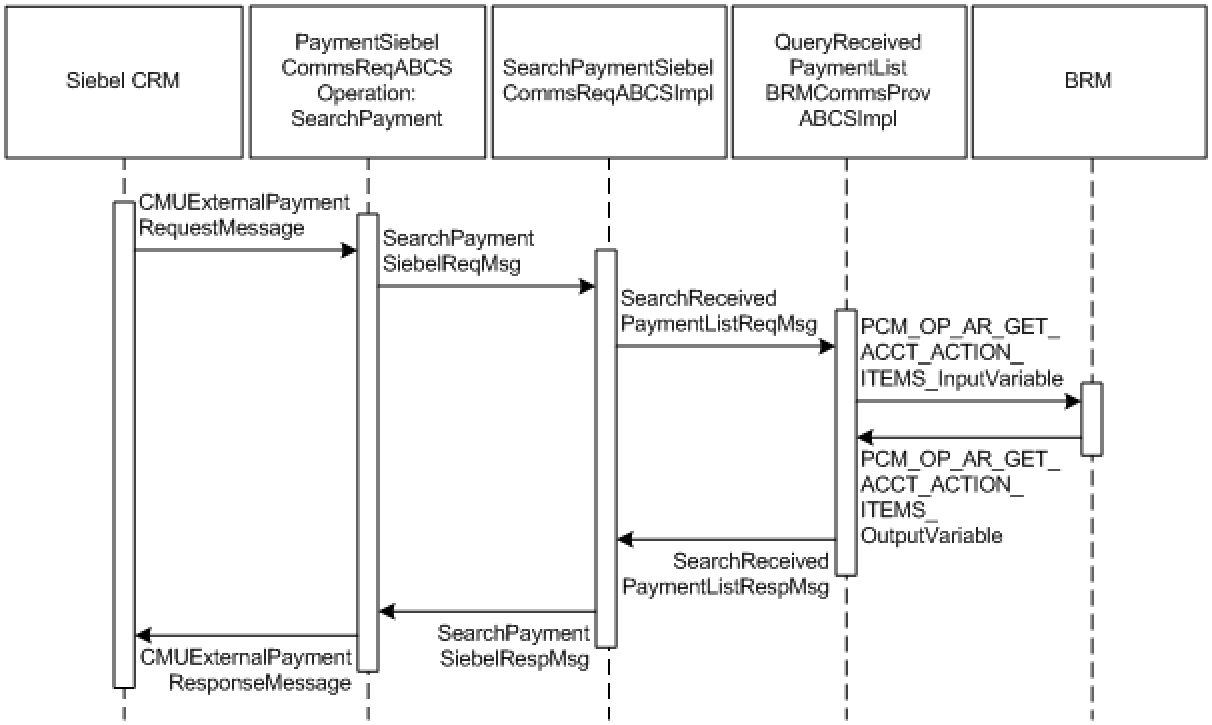
SearchPayment Integration Flow

This integration flow uses the following interfaces:

- ReceivedPaymentSiebelCommsReqABCSInterface with the following operations:
  - QueryPayment
  - SearchPayment
  - QueryInvoicePayment
- SearchPaymentSiebelCommsReqABCSImpl
- QueryReceivedPaymentListBRMCommsProvABCSImpl

Figure 6–6 illustrates the SearchPayment integration scenario.

Figure 6–6 SearchPayment Integration Flow Sequence Diagram


Siebel CRM
PaymentSiebel CommsReqABCS Operation: SearchPayment
SearchPaymentSiebel CommsReqABCSImpl
QueryReceived PaymentList BRMCommsProv ABCSImpl
BRM
CMUExternalPayment RequestMessage
SearchPayment SiebelReqMsg
SearchReceived PaymentListReqMsg
PCM_OP_AR_GET_ ACCT_ACTION_ ITEMS_InputVariable
PCM_OP_AR_GET_ ACCT_ACTION_ ITEMS_ OutputVariable
SearchReceived PaymentListRespMsg
SearchPayment SiebelRespMsg
CMUExternalPayment ResponseMessage

When you initiate the SearchPayment process, the following events occur:

1. In Siebel CRM, the user navigates to the Accounts screen, queries an account, and clicks a billing profile for the account. This displays the Billing Profile BRM screen. Click the Payments tab for the billing profile to access the payments screen. The Payments screen has two tabs, one to create a payment and one to search for payments. When you click Search Payment, a search applet is opened. After entering the search criteria click Go. This invokes the outbound web service PaymentSiebelCommsReqABCS with operation SearchPayment to fetch the payment records in the billing system.

   The following scenarios exist in which the same functionality is required in Siebel CRM. The integration flow is similar in both of these cases, but they have a separate SiebelABCSImpl.

   Once the CMUGetPayments is invoked, it fetches all records. You can search based on certain search criteria such as dates and amount range. After entering inputs, click Search to initiate this flow.

   From the Invoice tab, you can search for payments that are made against a specific invoice.

2. Navigating to the Payments tab invokes the PaymentSiebelCommsReqABCS web service, which calls PaymentSiebelCommsReqABCS with the operation SearchPayment.

   PaymentSiebelCommsReqABCS is a generic Siebel Payments interface service with several operations defined on the ReceivedPayment enterprise business object (EBO).

3. Invoking the PaymentSiebelCommsReqABCS with the operation SearchPayment routes the Siebel SearchPaymentReqMsg to SearchPaymentSiebelCommsReqABCSImpl.

4. The SearchPaymentSiebelCommsReqABCSImpl first transforms the SearchPaymentReqMsg into the SearchReceivedPaymentEBM and routes the SearchReceivedPaymentEBM to the appropriate billing system.

   As delivered, the SearchReceivedPaymentEBM is routed to the QueryReceivedPaymentListBRMCommsProvABCSImpl.

5. The QueryReceivedPaymentListBRMCommsProvABCSImpl first transforms the SearchReceivedPaymentListReqMsg into the input of PCM_OP_AR_GET_ACCT_ACTION_ITEMS and calls the BRM web service BRMARServices with the operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS.

6. Invoking the BRMARServices with the operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS invokes the BRM API PCM_OP_AR_GET_ACCT_ACTION_ITEMS and returns the payment object output to QueryReceivedPaymentListBRMCommsProvABCSImpl.

7. The QueryReceivedPaymentListBRMCommsProvABCSImpl then transforms the BRM API output into the SearchReceivedPaymentResponseEBM and returns it to SearchReceivedPaymentListSiebelABCSImpl.

8. The SearchReceivedPaymentListSiebelABCSImpl transforms the SearchReceivedPaymentResponseEBM into the SearchPaymentRespMsg, which is returned to the PaymentSiebelCommsReqABCS.

9. The ReceivedPaymentSiebelCommsReqABCS returns the SearchPaymentRespMsg to the calling Siebel web service CMUSearchPayment as CMUSearchPaymentResponseMessage.

10. The system writes the CMUSearchPaymentResponseMessage to the Siebel Unbilled Details VBC for the user.

BRM Interfaces

The CreateReceivedPayment integration flow uses this opcode:

- PCM_OP_PYMT_COLLECT

The QueryReceivedPaymentList integration flow uses this opcode:

- PCM_OP_AR_GET_ACCT_ACTION_ITEMS

See *Oracle Communications Billing and Revenue Management Opcode Flist Reference* for more information.

Siebel CRM Interfaces

The payment integration flow uses these Siebel CRM interfaces:

- For the CreateReceivedPayment flow: PaymentSiebelCommsReqABCS operation CreatePayment
- For the QueryReceivedPaymentList flow: PaymentSiebelCommsReqABCS operation QueryPayment

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The Payment integration uses the following delivered EBOs and EBMs:

- ReceivedPaymentEBO
- CreateReceivedPaymentEBM
- CreateReceivedPaymentResponseEBM
- QueryReceivedPaymentListEBM
- QueryReceivedPaymentListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the AIA Reference Doc link.

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

Integration Services

These services are delivered with the Payment integration flow:

- PaymentSiebelCommsReqABCS
- CreatePaymentSiebelCommsReqABCSImpl
- CreateInvoicePaymentSiebelCommsReqABCSImpl
- CreateReceivedPaymentBRMCommsProvABCSImpl
- QueryPaymentSiebelCommsReqABCSImpl
- QueryInvoicePaymentSiebelCommsReqABCSImpl
- SearchPaymentSiebelCommsReqABCSImpl
- QueryReceivedPaymentListBRMCommsProvABCSImpl

PaymentSiebelCommsReqABCS

PaymentSiebelCommsReqABCS exposes the following operations:

- CreatePayment:
  - Routes CreatePaymentReqMsg to the requester implementation service
  - Routes CreatePaymentSiebelRespMsg to the requester
- CreateInvoicePayment:
  - Routes CreateInvoicePaymentReqMsg to the requester implementation service
  - Routes CreateInvoicePaymentRespMsg to the requester
- QueryPayment:
  - Routes QueryPaymentReqMsg to the requester implementation service
  - Routes QueryPaymentRespMsg to the requester
- SearchPayment:
  - Routes SearchPaymentReqMsg to the requester implementation service
  - Routes SearchPaymentRespMsg to the requester
- QueryInvoicePayment:
  - Routes QueryInvoicePaymentReqMsg to the requester implementation service.
  - Routes QueryInvoicePaymentRespMsg to the requester

CreatePaymentSiebelCommsReqABCSImpl

The CreatePaymentSiebelCommsReqABCSImpl transforms the CreatePaymentReqMsg into a Payment EBM.

CreateInvoicePaymentSiebelCommsReqABCSImpl

CreateInvoicePaymentSiebelCommsReqABCSImpl transforms the CreateReceivedPaymentResponseEBM into CreateInvoicePaymentSiebelRespMsg, which is returned to the Siebel Account ABC interface service.

CreateReceivedPaymentBRMCommsProvABCSImpl

CreateReceivedPaymentBRMCommsProvABCSImpl transforms the ReceivedPayment EBM into BRM API input format and calls the API to Create Payment output from the billing system. It then transforms the output from the API back to a ReceivedPayment EBM message and returns it to the calling requestor.

QueryPaymentSiebelCommsReqABCSImpl

The QueryPaymentSiebelCommsReqABCSImpl transforms the QueryPaymentReqMsg into QueryReceivedPaymentListEBM.

QueryInvoicePaymentSiebelCommsReqABCSImpl

The QueryInvoicePaymentSiebelCommsReqABCSImpl transforms the QueryInvoicePaymentReqMsg into ReceivedPaymentEBM.

SearchPaymentSiebelCommsReqABCSImpl

The SearchPaymentSiebelCommsReqABCSImpl transforms the SearchPaymentReqMsg into ReceivedPaymentEBM.

QueryReceivedPaymentListBRMCommsProvABCSImpl

QueryReceivedPaymentListBRMCommsProvABCSImpl transforms the BRM API output into QueryReceivedPaymentListResponseEBM.

7

Adjustment Integration Flow

This chapter provides an overview of the adjustment integration flows and discusses Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM) interfaces, industry Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Adjustment Integration Overview

The Adjustment integration between Siebel CRM and BRM supports the following integration scenarios:

- QueryAdjustment enables a customer service representative (CSR) to view the adjustments for an invoice in Siebel CRM.
- CreateAdjustment enables a CSR to create different levels of adjustments (invoice, item, and event) for an invoice in Siebel CRM.

The invoice adjustment feature enables CSRs to make adjustments at three levels in the invoice (header, item, and event) in Siebel CRM. When a customer calls to dispute an item or multiple items on a bill, CSRs first identify the bill, and then find the disputed items on the bill.

The integration supports adjustment requests that are both amount-based and percentage-based for adjustments at the bill/header and event-level. Therefore, a CSR either captures an adjustment request for an absolute value (such as USD 2) or percentage value (such as. 2%).

In addition to invoice adjustments, a CSR can use this integration feature to make unbilled service usage adjustments at the event (call detail (CDR) record) level.

Caution: In the billing system, taxes can only be applied on AR items after billing. Therefore, for Unbilled Event adjustments, the tax flag that is set in the Siebel CRM UI must be Exclude Tax.

CSRs request adjustments on a variety of levels, as appropriate to the situation. For instance, if the customer made a 10-minute call that was mistakenly billed as a 30-minute call, the CSR requests an adjustment for that specific call at the call detail-level. If, however, the customers plan provided 100 free minutes a month, but charges started accruing after only 30 minutes, the CSR requests an adjustment at the summary-level instead. CSRs can create an adjustment request for one or more lines on a single invoice. The adjustment request can include the following details per line:

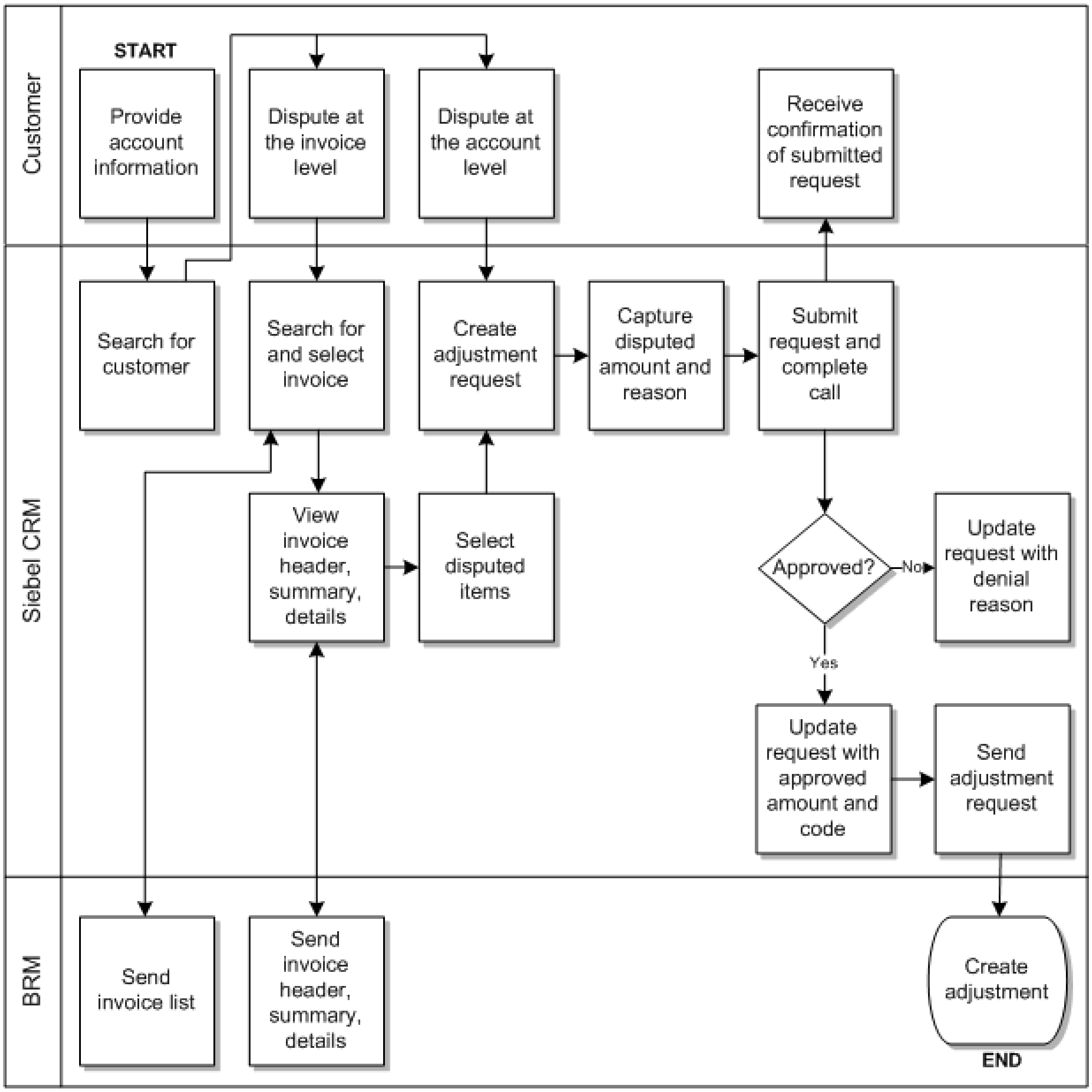
- Account #
- Invoice #

- Request ID
- Requested Date
- Adjustment Amount Requested
- Adjustment Type (for example, credit, debit)
- Reason for Request
- Comments
- Status
- Amount Approved
- Approval Code
- Date Approved

When the CSR clicks the Adjustments tab of the Siebel Billing Profile screen, the adjustment history information appears. The adjustment records that appear in this view are adjustment requests that have originated from Siebel CRM and stored within Siebel the database. The CSR must navigate to the Invoice Detail view to make adjustments at the header and item levels. For adjustments at the event-level, the CSR navigates to the event details view. Each adjustment request triggers a separate web services call in the billing system. The adjustment, if approved, is created in BRM and is reflected in the customers' next bill. If the adjustment is not approved, the adjustment request is updated with the reason for denial.

Figure 7–1 illustrates the flow for Adjustments:

Figure 7–1 The Adjustment Integration Flow


Customer
Siebel CRM
BRM
START
Provide account information
Dispute at the invoice level
Dispute at the account level
Receive confirmation of submitted request
Search for customer
Search for and select invoice
Create adjustment request
Capture disputed amount and reason
Submit request and complete call
View invoice header, summary, details
Select disputed items
Approved?
No
Yes
Update request with denial reason
Update request with approved amount and code
Send adjustment request
Send invoice list
Send invoice header, summary, details
Create adjustment
END

To view the adjustments for an invoice and create different levels of adjustments in Siebel CRM:

1. Navigate to the Accounts screen and query an account. Scroll down to the Billing profile applet and click the Billing profile **Name** link.
2. Click the **Bills** tab to view the list of bills under the account. Click the **Bill Number** link to open the Bill Details screen.
3. Click the **A/R Items** tab to view the adjustments for an invoice.
4. In the Bills Detail view, an adjustment request can be captured at the header, item, or event level. Each of these sections in the view has an **Adjust** button.
5. After creating a new adjustment for a particular level and getting the approval, click the **Submit** button.

See "Mapping Siebel Billing Management UI Elements to BRM Customer Center" for more information on mapping Siebel CRM elements to BRM elements.

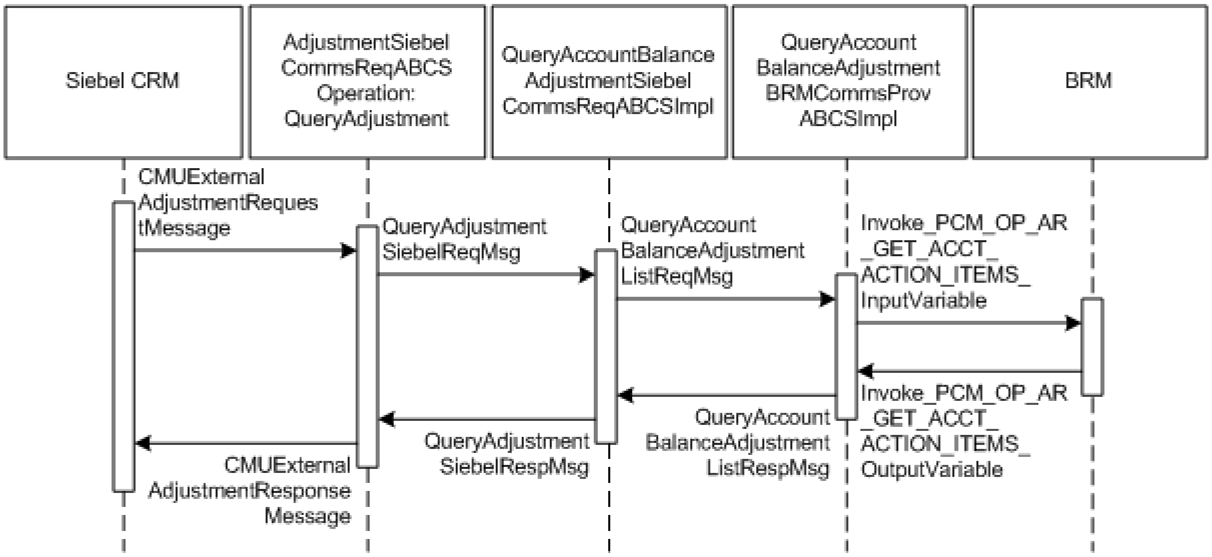
QueryAccountBalanceAdjustment Integration Flow

This integration flow uses the following interfaces:

- AdjustmentSiebelCommsReqABCS with operation QueryAdjustment
- QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl
- QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl

Figure 7–2 illustrates the QueryAccountBalanceAdjustment integration scenario.

Figure 7–2 QueryAccountBalanceAdjustment Integration Flow Sequence Diagram


Siebel CRM
AdjustmentSiebel CommsReqABCS Operation: QueryAdjustment
QueryAccountBalance AdjustmentSiebel CommsReqABCSImpl
QueryAccount BalanceAdjustment BRMCommsProv ABCSImpl
BRM
CMUExternal AdjustmentReques tMessage
QueryAdjustment SiebelReqMsg
QueryAccount BalanceAdjustment ListReqMsg
Invoke_PCM_OP_AR _GET_ACCT_ ACTION_ITEMS_ InputVariable
Invoke_PCM_OP_AR _GET_ACCT_ ACTION_ITEMS_ OutputVariable
QueryAccount BalanceAdjustment ListRespMsg
QueryAdjustment SiebelRespMsg
CMUExternal AdjustmentResponse Message

When you initiate the QueryAccountBalanceAdjustment process, the following events occur:

1. In Siebel CRM, navigate to the Billing Profile screen.
2. Navigate to Accounts, query an account, and click the billing profile for the account. On the Billing Profile screen, click the Bill tab to view the list of bills under the account. To open the Bill Detail View screen, click the Bill No link. This opens the Bill details view with the following information: bill summary, service charges, and item charges. Click the A/R Items tab and a web service call is made to get the adjustments specific to this bill for the account.
3. Open an invoice and select the A/R Items tab. A web service call is made to get the adjustment for that invoice.
4. Navigate to the Bill tab and open the Bill Detail View.
5. Select the A/R Items tab. This invokes the AdjustmentSiebelCommsReqABCS web service, which in turn calls AdjustmentSiebelCommsReqABCS with operation QueryAdjustment.

   AdjustmentSiebelCommsReqABCS is a generic Siebel adjustment interface service with several operations defined on the AccountBalanceAdjustmentEBO.

6. Invoking AdjustmentSiebelCommsReqABCS with operation QueryAdjustment routes the QueryAdjustmentReqMsg to QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl.
7. The QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl transforms the QueryAdjustmentReqMsg into QueryAccountBalanceAdjustmentList_InputVariable and routes the QueryAccountBalanceAdjustmentListReqMsg to the appropriate billing system.

   As delivered, QueryAccountBalanceAdjustmentListReqMsg is routed to QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl.
8. QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl first transforms QueryAccountBalanceAdjustmentListReqMsg into the Invoke_PCM_OP_AR_GET_ACCT_ACTION_ITEMS_InputVariable as input of PCM_OP_AR_GET_ACCT_ACTION_ITEMS and calls BRMARServices with operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS.
9. Invoking BRMARServices with operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS invokes the BRM application programming interface (API) PCM_OP_AR_GET_ACCT_ACTION_ITEMS and returns the adjustment outputs Invoke_PCM_OP_AR_GET_ACCT_ACTION_ITEMS_OutputVariable to QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl.
10. QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl transforms the BRM API output Invoke_PCM_OP_AR_GET_ACCT_ACTION_ITEMS_OutputVariable into enterprise business message (EBM) output QueryAccountBalanceAdjustmentListRespMsg and returns it to QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl.
11. QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl then transforms the QueryAccountBalanceAdjustmentListRespMsg into QueryAdjustmentRespMsg, which is returned to AdjustmentSiebelCommsReqABCS.
12. AdjustmentSiebelCommsReqABCS returns the QueryAdjustmentRespMsg to the calling CMUExternalAdjustments as QueryAdjustmentRespMsg.
13. CMUAdjustmentResponseMessage is then written to the Siebel Balance Summary virtual business component (VBC) for the users.

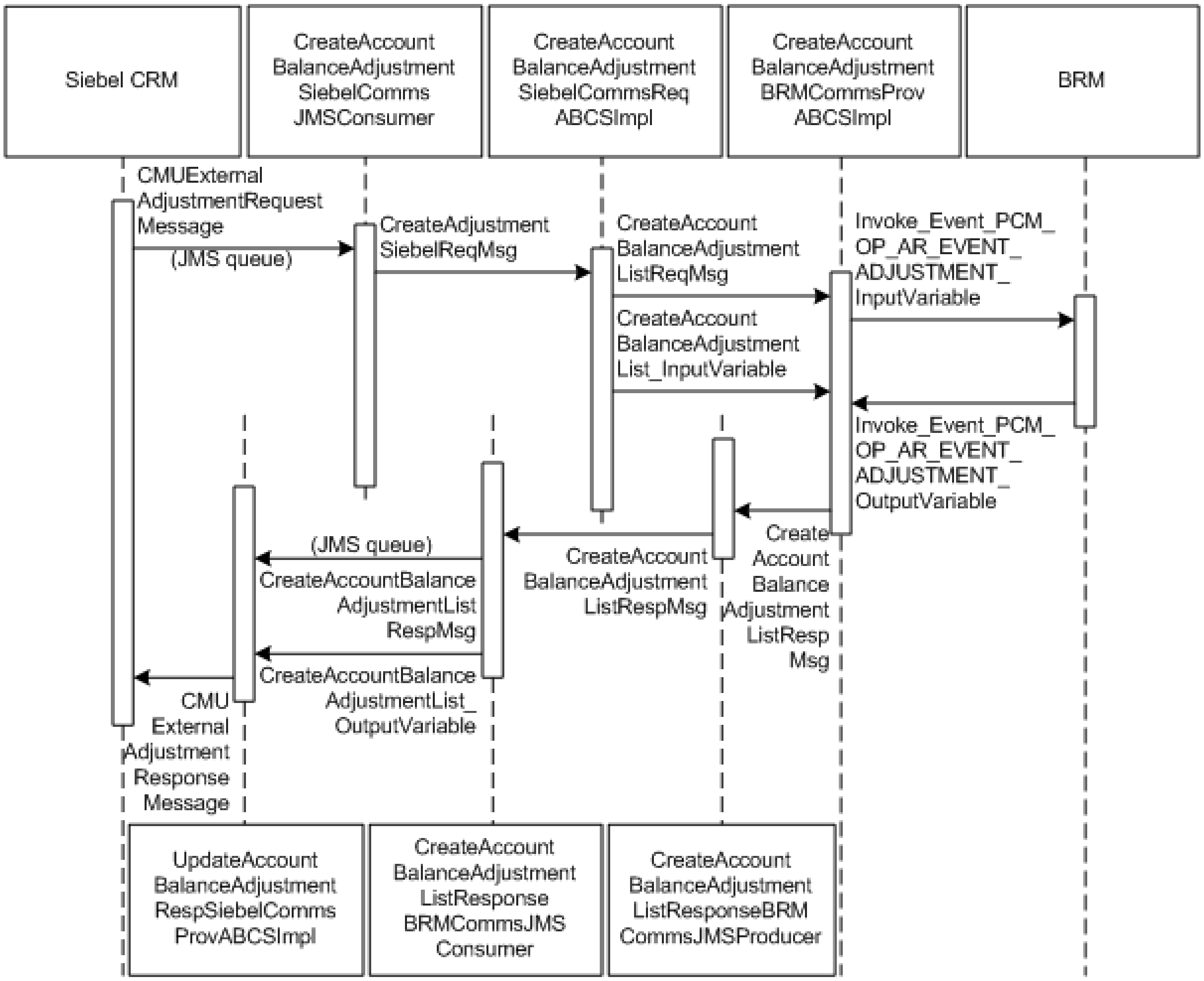
CreateAccountBalanceAdjustment Integration Flow

This integration flow uses the following interfaces:

- CreateAccountBalanceAdjustmentSiebelCommsReqABCSImpl
- CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl
- UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl
- CreateAccountBalanceAdjustmentListResponseBRMCommsJMSConsumer
- CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer
- CreateAccountBalanceAdjustmentSiebelCommsJMSConsumer

Figure 7–3 illustrates the CreateAccountBalanceAdjustment integration scenario.

Figure 7–3 CreateAccountBalanceAdjustment Integration Flow Sequence Diagram


Siebel CRM
CreateAccount BalanceAdjustment SiebelComms JMSConsumer
CreateAccount BalanceAdjustment SiebelCommsReq ABCSImpl
CreateAccount BalanceAdjustment BRMCommsProv ABCSImpl
BRM
CMUExternal AdjustmentRequest Message
(JMS queue)
CreateAdjustment SiebelReqMsg
CreateAccount BalanceAdjustment ListReqMsg
CreateAccount BalanceAdjustment List_InputVariable
Invoke_Event_PCM_ OP_AR_EVENT_ ADJUSTMENT_ InputVariable
Invoke_Event_PCM_ OP_AR_EVENT_ ADJUSTMENT_ OutputVariable
Create Account Balance Adjustment ListResp Msg
CreateAccount BalanceAdjustment ListRespMsg
(JMS queue)
CreateAccountBalance AdjustmentList RespMsg
CreateAccountBalance AdjustmentList_ OutputVariable
CMU External Adjustment Response Message
UpdateAccount BalanceAdjustment RespSiebelComms ProvABCSImpl
CreateAccount BalanceAdjustment ListResponse BRMCommsJMS Consumer
CreateAccount BalanceAdjustment ListResponseBRM CommsJMSProducer

When you initiate the CreateAccountBalanceAdjustment process, the following events occur:

1. In Siebel CRM, navigate to the Billing Profile screen.
2. Navigate to the Accounts screen, query an account, and click a billing profile for the account. On the Billing Profile screen, the Adjustment tab displays all the adjustment requests.
3. To create an adjustment for an invoice, click the Bills tab.

   Select the bill against which an adjustment request must be created.
4. In the Bill Details view, an adjustment request can be captured at the header, item, or event level. Each of these sections in the view has an Adjust button.

   Clicking this button creates a new adjustment request at that level.
5. To create an adjustment at the event-level, click the Net Amount link for the required item charge.

   This opens the Event Details view to create the adjustment.
6. At the event-level, adjustments can be created for both monetary and nonmonetary resources such as free minutes.
7. You can also create adjustments for unbilled usage.

Unbilled adjustments are applicable only at the event-level for both monetary and nonmonetary resources.

8. The Adjust button on different screens calls the same web service to create the adjustment.

   But on the BRM side, it is based on the adjustment level. The adjustment type is set by Siebel CRM. Based on the value in this column, the correct opcode is called.

9. After creating the new adjustment for a particular level and getting the approval, click the Submit button.

   A web service call is made to SWICreateAdjustment, which in turn puts the message into the Queue AIA_CMUREQADJIOJMSQUEUE along with a Simple Object Access Protocol (SOAP) envelope. After submit, the adjustment Status changes to Submitted and the adjustment record becomes read-only. Adjustments are persisted in Siebel CRM and when the adjustments are accepted, the status of the record changes to Posted. If the changes are not approved, the status changes to Not Posted.

10. CreateAccountBalanceAdjustmentSiebelCommsJMSConsumer dequeues the message and transforms it into the Siebel request application business message (ABM) and routes the CreateAdjustmentReqMsg to CreateAccountBalanceAdjustmentSiebelCommsABCSImpl

11. The CreateAccountBalanceAdjustmentSiebelCommsABCSImpl transforms the CreateAdjustmentReqMsg into CreateAccountBalanceAdjustmentList_InputVariable and routes the CreateAccountBalanceAdjustmentList_InputVariable to the appropriate billing system.

    As delivered, CreateAccountBalanceAdjustmentList_InputVariable is routed to CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl.

12. CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl first checks the parameters of the Adjustment type coming from the request (Billed or Unbilled) and based on the parameter, PCM_OP_AR_EVENT_ADJUSTMENT is invoked with the appropriate data.

13. After checking the parameters of a particular service from request, the CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl service transforms the CreateAccountBalanceAdjustmentListReqMsg into the BRM input flist message and invokes the opcode.

14. Invoking BRM API PCM_OP_AR_EVENT_ADJUSTMENT with account ID, billing profile ID, and event ID returns the list of events associated with items.

15. BRMARServices sends the response back to CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl service with the list of adjustments and status for opcode calls.

    The Status field is mapped to the response and goes back to the Siebel CRM user interface (UI).

16. After getting the response back from BRMARService, the status of the adjustment in CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl is checked.

    If the status value returns as *Fail*, an error-handling framework service is called. This error-handling framework service calls the different error-handling services and logs the error for that particular failed adjustment request in the Admin Console.

17. Administrators can see the status of the particular failed adjustment request, adjustment ID, and integration ID of that particular request.

Administrators can also get the BRM description for a failed request.

18. CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl then takes the response from the service, transforms Invoke_Event_PCM_OP_AR_EVENT_ADJUSTMENT_OutputVariable to the CreateAccountBalanceAdjustmentListRespMsg, and routes it to the CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer.
19. CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer then puts the message into the queue AIA_CRTADJLSTRSPJMSQUEUE.
20. CreateAccountBalanceAdjustmentListResponseBRMCommsJMSConsumer picks the message from AIA_CRTADJLSTRSPJMSQUEUE and routes the CreateAccountBalanceAdjustmentListResponseMsg to UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl.
21. UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl transforms the CreateAccountBalanceAdjustmentListRespMsg into SWISIAAdjustmentIO and invokes the Siebel Update web service with this message.

Note: In the case of billed adjustments at the Bill and Item levels, the BRM provider calls PCM_OP_AR_BILL_ADJUSTMENT and PCM_OP_AR_ITEM_ADJUSTMENT opcodes.

BRM Interfaces

The QueryAdjustment integration flow uses:

- BRMARServices with operation PCM_OP_AR_GET_ACCT_ACTION_ITEMS

The CreateAdjustment integration flow uses:

- BRMARServices with operation PCM_OP_AR_EVENT_ADJUSTMENT

See *Oracle Communications Billing and Revenue Management Opcode Flist Reference* for more information.

Siebel CRM Interfaces

The Adjustment integration flow uses these Siebel CRM interfaces:

- SWICreateAdjustment: To submit the adjustment
- SWIAdjustmentStatusUpdate: To update the adjustment

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The Adjustment integration uses the following delivered enterprise business objects (EBOs) and enterprise business messages (EBMs):

- AccountBalanceAdjustmentEBO
- QueryAccountBalanceAdjustmentListEBM
- QueryAccountBalanceAdjutsmentListResponseEBM
- CreateAccountBalanceAdjustmentListEBM

- CreateAccountBalanceAdjustmentListResponseEBM

The industry enterprise business object (EBO) and enterprise business message XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the AIA Reference Doc link.

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

Integration Services

These services are delivered with the Adjustment Integration flow:

- AdjustmentSiebelCommsReqABCS
- QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl
- QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl
- CreateAccountBalanceAdjustmentSiebelCommsABCSImpl
- CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl
- UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl
- CreateAccountBalanceAdjustmentListResponseBRMCommsJMSConsumer
- CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer
- CreateAccountBalanceAdjustmentSiebelCommsJMSConsumer

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

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

AdjustmentSiebelCommsReqABCS

AdjustmentSiebelCommsReqABCS exposes the following operation related to Account Balance Adjustment on the Siebel ABM:

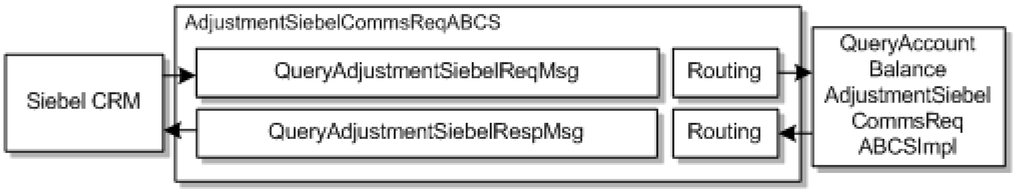
QueryAdjustment:

Routes QueryAdjustmentReqMsg to the provider implementation service

Routes QueryAdjustmentRespMsg to the requester

Figure 7–4 illustrates the relationship of AdjustmentSiebelCommsReqABCS with the other services in the integration flow.

Figure 7–4 AdjustmentSiebelCommsReqABCS

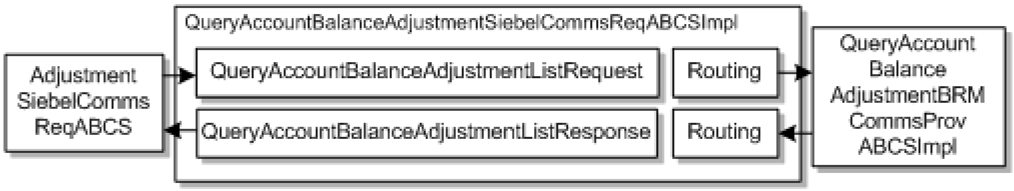

AdjustmentSiebelCommsReqABCS
Siebel CRM
QueryAdjustmentSiebelReqMsg
Routing
QueryAdjustmentSiebelRespMsg
Routing
QueryAccount Balance AdjustmentSiebel CommsReq ABCSImpl

QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl

QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl transforms the Siebel message into the AccountBalanceAdjustmentEBM and calls the provider to get the Adjustment response from the billing system. It then transforms the AccountBalanceAdjustmentEBM response back to a Siebel message and returns it to the calling Siebel web service.

Figure 7–5 illustrates the relationship of QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl with the other services in the integration flow.

Figure 7–5 QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl

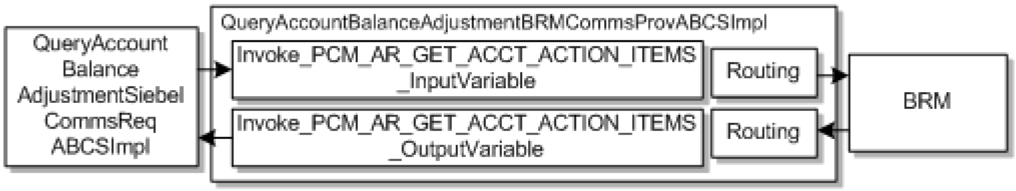

QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl
Adjustment SiebelComms ReqABCS
QueryAccountBalanceAdjustmentListRequest
Routing
QueryAccountBalanceAdjustmentListResponse
Routing
QueryAccount Balance AdjustmentBRM CommsProv ABCSImpl

QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl

QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl transforms the AccountBalanceAdjustmentEBM message into an BRM API input format message and calls the API to get the adjustment details from the billing system. It then transforms the output from the API back to an AccountBalanceAdjustmentEBM message and returns it to the calling service.

Figure 7–6 illustrates the relationship of QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl with the other services in the integration flow:

Figure 7–6 QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl


QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl
QueryAccount Balance AdjustmentSiebel CommsReq ABCSImpl
Invoke_PCM_AR_GET_ACCT_ACTION_ITEMS _InputVariable
Routing
Invoke_PCM_AR_GET_ACCT_ACTION_ITEMS _OutputVariable
Routing
BRM

CreateAccountBalanceAdjustmentSiebelCommsABCSImpl

CreateAccountBalanceAdjustmentSiebelCommsABCSImpl transforms the Siebel message into an AccountBalanceAdjustmentEBM message format and calls the provider to create the adjustment and get the response from the billing system. It then transforms the EBM response message to a Siebel message and routes it back to the AdjustmentSiebelCommsReqABCS service WSDL.

CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl

CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl transforms the AccountBalanceAdjustmentListEBM into an BRM API input format and calls the APIs to get the bill, item charge, and event details or Resource Impact Event adjustments from the billing system. It then transforms the output from the APIs back to an AccountBalanceAdjustmentListResponseEBM format that returns it to the calling service.

UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl

UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl transforms the AccountBalanceAdjustmentListResponseEBM into Siebel web service input formats and calls the web service to update the adjustment status.

This service is SPM enabled.

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

CreateAccountBalanceAdjustmentListResponseBRMCommsJMSConsumer

This service picks the message from the queue AIA_CRTADJLSTRSPJMSQUEUE and routes the CreateAccountBalanceAdjustmentListResponseMsg to UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl.

CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer

This process gets the CreateAccountBalanceAdjustmentListResponseEBM message from CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl and puts the message into the Queue AIA_CMUREQADJIOJMSQUEUE.

CreateAccountBalanceAdjustmentSiebelCommsJMSConsumer

This process picks the message with a SOAP envelope from the queue AIA_CMUREQADJIOJMSQUEUE, transforms the message into a Siebel Requestor ABM by opening the SOAP envelope, and calls the CreateAccountBalanceAdjustmentSiebelCommsABCSImpl with the ListOfCmuRequestAdjustmentIO message.

8

Understanding the Process Integration for Collection Management

This chapter provides an overview of the collections integration flows and discusses Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM) interfaces, Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

Process Integration for Collection Management Overview

Collection management is a process to collect money from subscribers after the grace period provided to pay the dues is over. If subscribers do not make a payment after the grace period, service providers may choose to remind the subscribers at first with a letter or a phone call. If these contact methods fail, service providers may decide to take actions such as inactivating the service.

The process integration for collection management provides synchronization of collections actions based on specified collections scenarios between BRM and Siebel CRM, and administration of these collections actions and credit alerts.

The collection management business process covers the entire collections life cycle across BRM and Siebel CRM to define activities typically performed, such as:

1. In BRM, a collections scenario is defined, created, and associated with a sequence of actions that must be performed.
2. Any bill units that move into collections and require an actionable event are passed to Siebel CRM as an action notification event in the form of a credit alert.
3. Siebel CRM capabilities can be accessed by the collections agent to create additional activities which can be associated with a credit alert.

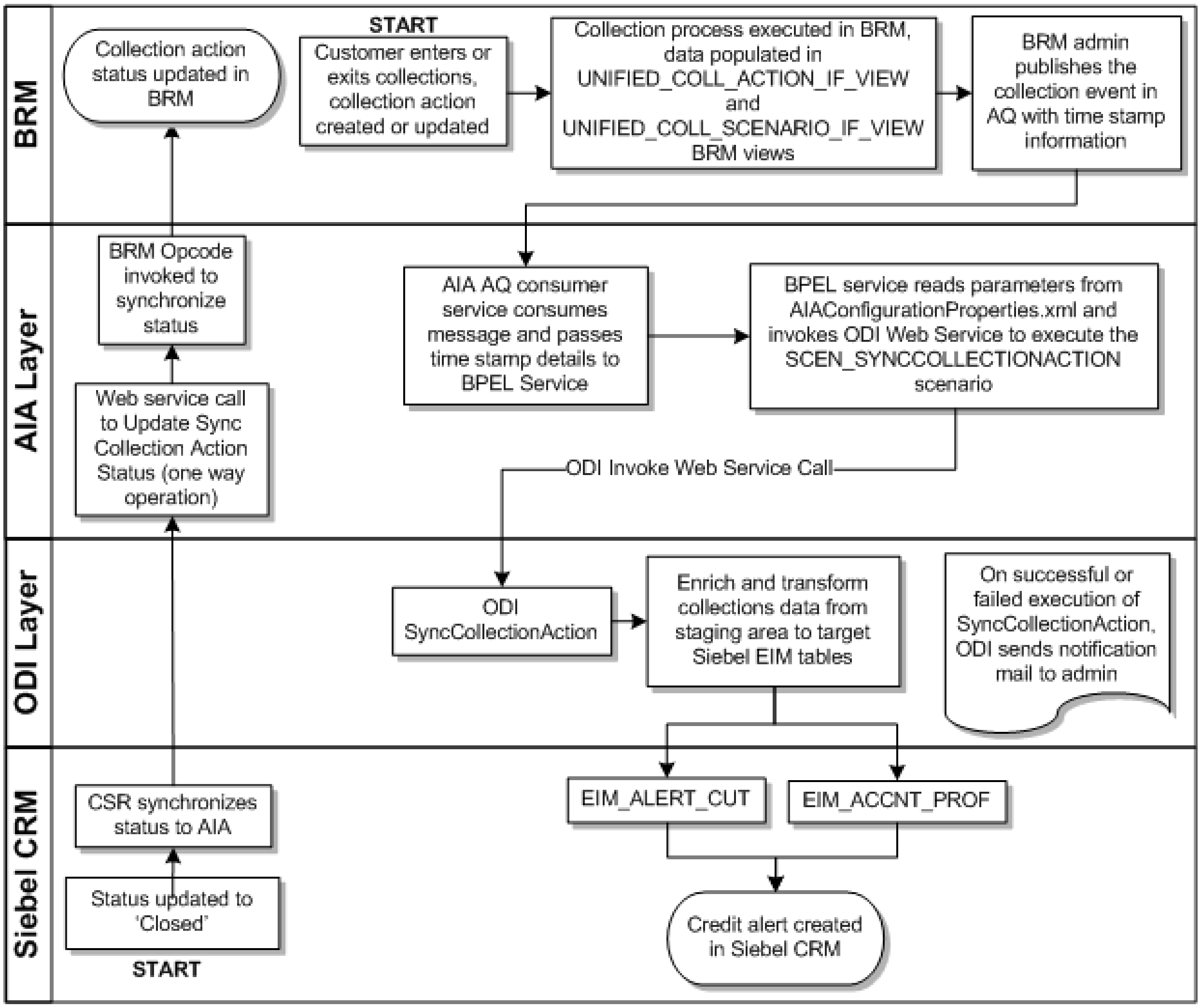
The process integration for collection management consists of these integration flows:

- Synchronizing Collection Scenarios and Actions integration flow
- Synchronizing Credit Alerts integration flow

Collection Management Business Process Flow

Figure 8–1 illustrates the overall flow for the process integration for collection management.

Figure 8–1 Collection Management Business Process Flow


BRM
AIA Layer
ODI Layer
Siebel CRM
START
Collection action status updated in BRM
Customer enters or exits collections, collection action created or updated
Collection process executed in BRM, data populated in UNIFIED_COLL_ACTION_IF_VIEW and UNIFIED_COLL_SCENARIO_IF_VIEW BRM views
BRM admin publishes the collection event in AQ with time stamp information
BRM Opcode invoked to synchronize status
AIA AQ consumer service consumes message and passes time stamp details to BPEL Service
BPEL service reads parameters from AIAConfigurationProperties.xml and invokes ODI Web Service to execute the SCEN_SYNCCOLLECTIONACTION scenario
Web service call to Update Sync Collection Action Status (one way operation)
ODI Invoke Web Service Call
ODI SyncCollectionAction
Enrich and transform collections data from staging area to target Siebel EIM tables
On successful or failed execution of SyncCollectionAction, ODI sends notification mail to admin
EIM_ALERT_CUT
EIM_ACCNT_PROF
CSR synchronizes status to AIA
Status updated to 'Closed'
START
Credit alert created in Siebel CRM

Assumptions and Constraints for Working with Collection Management

These are the assumptions and constraints for the process integration for collection management:

- Cross reference data and domain value map (DVM) setup should be verified for a successful collections batch load run.
- To achieve and maintain high performance, the database memory area must be large enough to hold the frequently accessed data in the cache.
- If an amount due alert has not been sent by BRM a default 0 (zero) amount is set in the integration process.
- When creating a credit alert, a default agreement association is provided at the account-level and not the credit alert-level.
- The integration supports multiple billing systems, with the assumption that a given Siebel CRM billing profile is mapped or synchronized to a single billing system. With this assumption, the ready-to-use credit alert status update flow from Siebel CRM to billing routes updates back to the appropriate billing system. However, the inbound Oracle Data Integrator (ODI) based flow that synchronizes

credit alerts from BRM to Siebel CRM requires some manual adjustments (cloning of ODI artifacts and other design-time changes) for each additional billing system that gets added.

See "Solution Assumptions and Constraints" for other dependencies.

See the discussion of configuring multiple BRM instances in *Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management* for more information about configuring multiple billing systems.

- BRM collections messages are stored in BRM tables, which can be queried from BRM database views. In a multischema scenario, a unified view is available on BRM to query the information of collections actions and scenarios.
- No cross-reference for the collections action ID exists. The BRM Portal Object (POID) for the collections action ID is sent to Siebel CRM as the integration ID. This POID is used for all references to the action and when an Update Collection Action and Update Collection Action status is selected.
- In case of failure executions, there are two options available for the administrator:
  - Re-execute the ODI scenario manually.
  - Start the SyncCollectionHeaderInfoBRMCommsReqImpl service using the timestamp details. This will start the ODI web service to begin the scenario.
- ODI uses SMTP port 25 to send notification mails. To use a different port you must configure the SMTP Port in ODI. See the discussion of defining Java options in ODI on the Oracle Support Web site:

  `https://support.oracle.com/epmos/faces/DocumentDisplay?id=1317507.1`

Synchronizing Collection Scenarios and Actions Integration Flow - BRM to Siebel CRM

This section provides an overview of the Synchronizing Collection Scenarios and Actions integration flow and discusses how to:

- Configure the **AIAConfigurationProperties.xml** file in an Oracle AIA layer
- Receive and transform the data

For information about configuring BRM for integrated collections management, see:

- Setting Up BRM
- *Oracle Communications Billing and Revenue Management Concepts*
- *Oracle Communications Billing and Revenue Management Collections Manager*

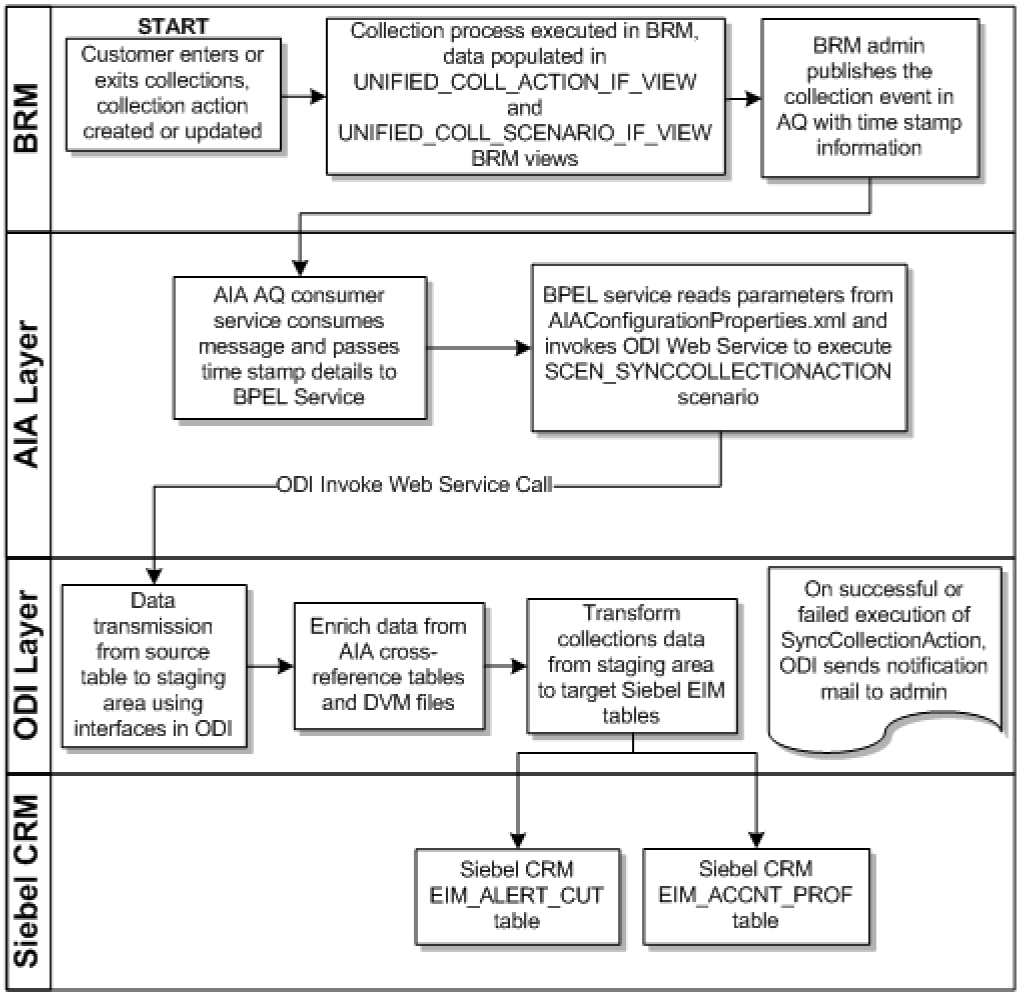
For information about managing collections in Siebel CRM and configuring Siebel CRM for integration, see *Siebel Communications Guide*.

Understanding Synchronizing Collection Scenarios and Actions Integration Flow

The synchronizing collection scenarios and actions integration enables the transfer of collections data from BRM database views to Siebel Enterprise Integration Manager (EIM) tables. This integration uses ODI to transfer the data from BRM to Siebel CRM.

Figure 8–2 illustrates the synchronizing collection scenarios and actions integration flow.

Figure 8–2 Collections Data Synchronizing Activity Flow Diagram


BRM
AIA Layer
ODI Layer
Siebel CRM
START
Customer enters or exits collections, collection action created or updated
Collection process executed in BRM, data populated in UNIFIED_COLL_ACTION_IF_VIEW and UNIFIED_COLL_SCENARIO_IF_VIEW BRM views
BRM admin publishes the collection event in AQ with time stamp information
AIA AQ consumer service consumes message and passes time stamp details to BPEL Service
BPEL service reads parameters from AIAConfigurationProperties.xml and invokes ODI Web Service to execute SCEN_SYNCCOLLECTIONACTION scenario
ODI Invoke Web Service Call
Data transmission from source table to staging area using interfaces in ODI
Enrich data from AIA cross-reference tables and DVM files
Transform collections data from staging area to target Siebel EIM tables
On successful or failed execution of SyncCollectionAction, ODI sends notification mail to admin
Siebel CRM EIM_ALERT_CUT table
Siebel CRM EIM_ACCNT_PROF table

When you initiate this process by creating a new collections action or updating an existing collections action, the following events occur:

1. A collections process is started in BRM and the actions data is populated into two database views:
   - UNIFIED_COLL_ACTION_IF_VIEW which contains all the collections action data (for example, to send dunning letter)
   - UNIFIED_COLL_SCENARIO_IF_VIEW which contains the status of billing profiles (for example, Entered Collections or Exited Collections).
2. When the collections process is finished, BRM creates a CollectionsInfoChange business event and publishes it to an Advanced Queue (AQ) database queue with the start and end timestamp for the collection event. The timestamp information is in Unix format. For example, 1319357706.
3. The SyncCollectionsInfoChangeBRMAQ service receives an AQ queue business event notification message and passes the timestamp details to SyncCollectionHeaderInfoBRMCommsReqImpl BPEL service.
4. The SyncCollectionHeaderInfoBRMCommsReqImpl BPEL service reads the required parameters from **AIAConfigurationProperties.xml** file and calls the ODI Web Service to start the SCEN_SYNCCOLLECTIONACTION scenario.

5. SCEN_SYNCCOLLECTIONACTION process starts ODI interfaces to perform the following:
   - Reads data from UNIFIED_COLL_ACTION_IF_VIEW and updates this data by performing a cross-reference and DVM lookup, and querying Siebel CRM base tables.
   - Reads data from the UNIFIED_COLL_SCENARIO_IF_VIEW view and updates this data by performing a cross-reference and DVM lookup, and querying Siebel CRM base tables.
   - Updates the AccountID and BillingProfileID columns using cross-reference data.
   - Updates the Currency Code, Action Name, Priority, and similar columns using DVMs.
6. The updated data is mapped to the EIM_ALERT_CUT table. For every row inserted into the EIM tables, the value of the IF_ROW_STAT column is set to FOR_IMPORT.
7. The Entered Collections and Exited Collections rows are extracted from the EIM_ALERT_CUT table and written to the EIM_ACCNT_PROF table. This data is responsible for changing the status of Billing Profile.
8. A success or failure notification mail is sent to the administrator. If a failure message is received the administrator must take the appropriate action. See "Handling Errors" for more information.
9. Siebel CRM runs a batch job and moves the data from the EIM table to the base table and creates credit alerts.

Table 8–1 Synchronizing Collection Scenarios and Actions Integration Flow

Work Location	Step
BRM	1. Collections actions are generated and stored in a database view. 2. Publishes an event to the AQ queue with timestamp details.
Integration Process (ODI)	3. AQ Consumer Service receives the message and passes on the details to a BPEL Service. 4. BPEL Service calls the ODI Web service with a timestamp value as input parameters. 5. Integration flow reads the messages from the BRM database views. 6. Data is collected from BRM, updated by cross-references and DVM lookup, and inserted into the Siebel EIM tables. 7. After the ODI Scenario is processed, a success or failure notification message is sent to the Administrator.
Siebel CRM	8. Siebel CRM runs a batch job to move data from the EIM table to the Base table and creates credit alerts. 9. If required, updates the billing profile.

Time Zone Handling

Oracle AIA does not do a time-zone conversion when synchronizing credit alerts from BRM to Siebel CRM.

BRM publishes collection StartTime and EndTime timestamp details in BRM local server time in epoch format.

Configuring and Generating Data in BRM

You must configure BRM for integrated collections management as described in:

- Setting Up BRM
- The discussion of using integrating collections with your external CRM application in *Oracle Communications Billing and Revenue Management Concepts*
- The discussion of integrating Collections Manager with custom client applications in *Oracle Communications Billing and Revenue Management Collections Manager*

When you have configured BRM for integrated collections management, BRM generates collections data as follows:

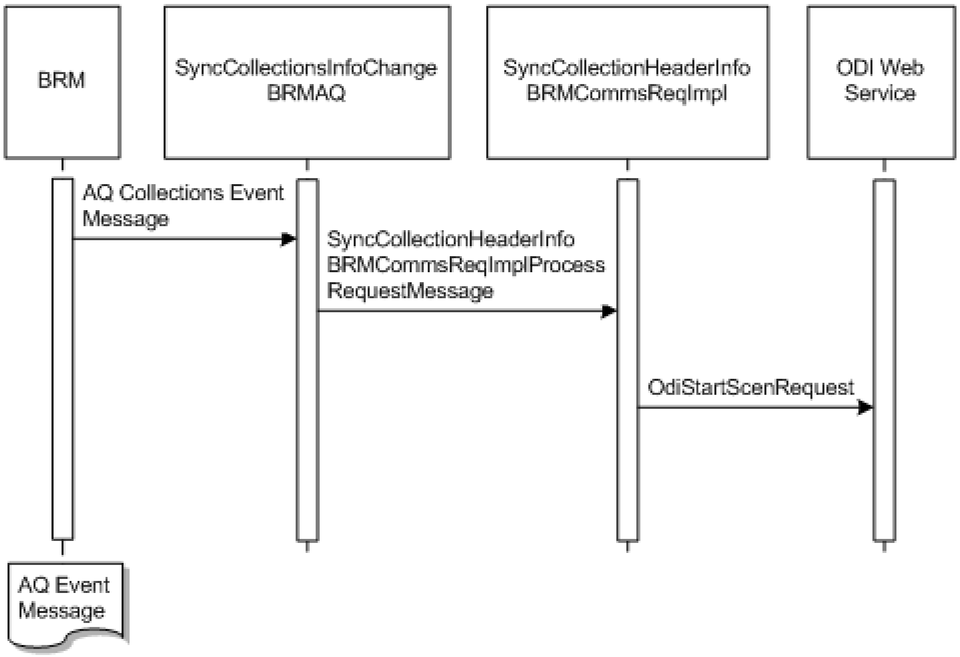
1. A customer enters collections and one of the following collections activities occurs:
   - An account enters or exits collections.
   - A collections action object is created or updated.
   - A /schedule object is created.
2. A collections process is started in BRM and data is populated to the UNIFIED_COLL_ACTION_IF_VIEW and UNIFIED_COLL_SCENARIO_IF_VIEW views.

See *Oracle Communications Billing and Revenue Management Collections Manager* for more information about how BRM handles collections.

Receiving and Transforming the Data

The synchronizing collection scenarios and actions process integration uses ODI to receive, transform, and load the BRM data into the Siebel CRM EIM tables.

Figure 8–3 Synchronizing Collection Scenarios and Actions from BRM to Siebel CRM Sequence Diagram


BRM
SyncCollectionsInfoChange BRMAQ
SyncCollectionHeaderInfo BRMCommsReqImpl
ODI Web Service
AQ Collections Event Message
SyncCollectionHeaderInfo BRMCommsReqImplProcess RequestMessage
OdiStartScenRequest
AQ Event Message

These tasks are performed as a part of the data load process:

1. In BRM the **pin_collections_process** utility generates the **/event/notification/collections/info_change** BRM business event.
2. The Data Manager AQ queue receives the payload data and publishes a new **CollectionsInfoChange** business event to the AQ queue. This includes the start and end timestamps, and the Account POID from the BRM event.
3. An Oracle AIA BPEL service SyncCollectionsInfoChangeBRMAQ receives a message from the AQ queue that new data is available and starts the SyncCollectionHeaderInfoBRMCommsReqImpl BPEL service.
4. The SyncCollectionHeaderInfoBRMCommsReqImpl BPEL service receives the timestamp parameters and reads the required parameters from the **AIAConfigurationProperties** file. Using the parameters it calls the ODI web service to start the SCEN_SYNCCOLLECTIONACTION process.
5. SCEN_SYNCCOLLECTIONACTION process starts ODI to perform the following:
   - Reads data from the UNIFIED_COLL_ACTION_IF_VIEW view and updates this data by performing a cross-reference and DVM lookup, and querying Siebel CRM base tables.
   - Reads data from the UNIFIED_COLL_SCENARIO_IF_VIEW view and updates this data by performing a cross-reference and DVM lookup, and querying Siebel CRM base tables.
   - Updates the AccountID and BillingProfileID columns using cross-reference data.
   - Updates the Currency Code, Action Name, Priority, and similar columns using DVMs.
6. The updated data is mapped to the Siebel EIM_ALERT_CUT table.
7. The Entered Collections and Exited Collections rows are extracted from the EIM_ALERT_CUT table and written to the EIM_ACCNT_PROF table. This data is responsible for changing the status of Billing Profile.
8. For every row inserted in the EIM tables, the value of the IF_ROW_STAT column is set to FOR_IMPORT.

Note: Before ODI loads the enriched collections data into the Siebel EIM tables, it performs data cleansing. All existing records that were successfully imported into the Siebel base table are deleted from the EIM tables.

Configuring AIAConfigurationProperties file to work with ODI

This section discusses how to configure the **AIAConfigurationProperties.xml** file to work with ODI.

Based on your requirements, after you have installed the process integration for collection management, you need to change the default values of the collections parameters to suit your implementation.

To configure the **AIAConfigurationProperties.xml** file to change the default values, do the following:

1. Open *$AIA_INSTANCES*/**config/AIAConfigurationProperties.xml** file in a text editor.

See "Building AIA Integration Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to reload updates to **AIAConfigurationProperties.xml** file.

2. Locate <ModuleConfiguration moduleName="CollectionsParameters"> and update the following default values:

   - FromMailAddress
   - ToMailAddress
   - MailServer
   - ODI.USERNAME
   - ODI.PASSWORD
   - ODI.WORKREPNAME
   - BatchSize
   - Default.SystemID
   - AIAHome
   - IF_ROW_BATCH_NUM_CreatedStart
   - IF_ROW_BATCH_NUM_CreatedEnd
   - IF_ROW_BATCH_NUM_UpdateStart
   - IF_ROW_BATCH_NUM_UpdateEnd
   - IF_ROW_BATCH_NUM_Error
   - AL_SOURCE_CD
   - AL_TYPE_CD

   See "Configuring the Process Integration for Collections Management" for more information about these properties.

Synchronizing Credit Alerts Integration Flow - Siebel CRM to BRM

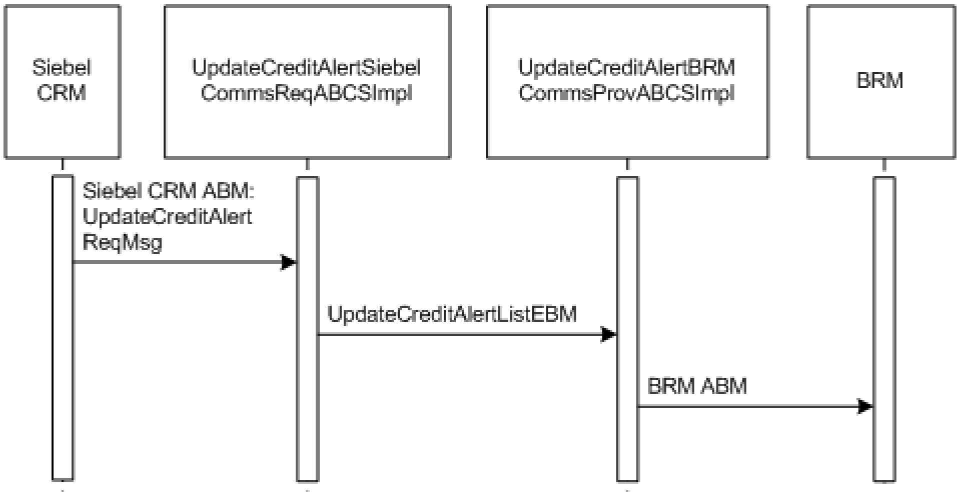
This integration flow synchronizes the credit alerts from Siebel CRM to BRM.

This integration flow uses the following services:

- UpdateCreditAlertSiebelCommsReqABCSImpl with operation UpdateCreditAlert
- UpdateCreditAlertBRMCommsProvABCSImpl with operation UpdateCreditAlert

Figure 8–4 illustrates the synchronizing credit alerts integration scenario.

Figure 8–4 Synchronizing Credit Alerts from Siebel CRM to BRM Sequence Diagram


Siebel CRM
UpdateCreditAlertSiebelCommsReqABCSImpl
UpdateCreditAlertBRMCommsProvABCSImpl
BRM
Siebel CRM ABM: UpdateCreditAlertReqMsg
UpdateCreditAlertListEBM
BRM ABM

When you initiate the synchronizing credit alert process, the following events occur:

1. Siebel CRM starts Oracle AIA UpdateCreditAlertSiebelCommsReqABCSImpl service.
2. The UpdateCreditAlertSiebelCommsReqABCSImpl service transforms Credit Alert Status Siebel Application Business Message (ABM) ListOfSWICUTCreditManagementIO to the UpdateCreditAlertListEBM message. Based on the Billing Profile ID it sets the Target System ID in the EBM and calls the UpdateCreditAlertBRMCommsProvABCSImpl service.
3. The UpdateCreditAlertBRMCommsProvABCSImpl service searches for the Billing Profile ID, and sets the Target System ID and URL of target BRM instance.

   The UpdateCreditAlertBRMCommsProvABCSImpl service then transforms the UpdateCreditAlertListEBM message to an ABM and starts the PCM_OP_COLLECTIONS_SET_ACTION_STATUS BRM opcode.
4. The credit alert or action status is updated to **Closed** in BRM.

Table 8–2 Synchronizing Credit Alerts Integration Flow

Work Location	Step
Siebel CRM	1. The action status for a billing profile is updated.
Integration Process (Oracle AIA)	2. Requester Application Business Connector Service (ABCS) receives the request from Siebel and transforms the application business specific message (ABM) to application neutral message (EBM). 3. Provider Application Business Connector Service (ABCS) transforms EBM to ABM and starts a BRM op-code.
BRM	4. The BRM op-code updates the status of the given collections action in BRM.

Handling Errors

The process integration for collection management handles BPEL services and ODI errors.

Handling BPEL Service Errors

For BPEL service errors, do the following:

SyncCollectionHeaderInfoBRMCommsReqImp

This service uses the default error handling capability of Oracle AIA.

- Faults: On Error the collections Application Business Message (ABM) is transformed to a fault message and returned to the calling process. The fault message contains the collections run timestamp details. The Error BPEL process is started. An Exception is raised.
- Based on your Error Handling Notification configuration, an email is sent to the user.

  According to the message received the user must take the appropriate action.

UpdateCreditAlertSiebelCommsReqABCSImpl

This service uses the default error handling capability of Oracle AIA.

- Faults: On Error the Credit Alert ABM is transformed to a fault message and returned to the calling process. The fault message contains the Credit Alert details. The Error BPEL process is started. An Exception is raised.
- Based on your Error Handling Notification configuration, an email is sent to the user.

  According to the message received the user must take the appropriate action.

UpdateCreditAlertBRMCommsProvABCSImpl

This service uses the default error handling capability of Oracle AIA.

- Faults: On Error the Credit Alert ABM is transformed to a fault message and returned to the calling process. The fault message contains the Credit Alert details. The Error BPEL process is started. An Exception is raised.
- Based on your Error Handling Notification configuration, an email is sent to the user.

  According to the message received the user must take the appropriate action.

See *Oracle Application Integration Architecture Foundation Pack* for more information.

Handling ODI Error Messages

After each scenario run a notification email is sent to the user's email address.

> **Note:** The FromMailAddress and ToMailAddress must be set in the **AIAConfiguraionProperties.xml** file. See "Configuring AIAConfigurationProperties file to work with ODI" for more information.

On receiving a notification email, for failed jobs do the following:

1. Take the appropriate action based on the message received.
2. Re-start the scenario by doing one of the following:
   a. Manually restart the scenario. See "Restarting Your Job Manually" for more information.

b. Re run the job by passing the timestamp details to the SyncCollectionHeaderInfoBRMCommsReqImpl service.

Restarting Your Job Manually

The notification mail contains the start and end timestamps, for example:

```
Coll_StartTime: 1299794311
Coll_EndTime: 1299794315
```

To restart your job manually:

1. On the Oracle AIA system, navigate to the following file path:

   ODI_HOME/**oracledi/agent/bin**

   where *ODI_HOME* is the path to ODI home

2. Run the following command, where *Start_Time_Stamp* and *End_Time_Stamp* are the timestamp values shown in the notification email:

   - In a Linux environment:

     ```
     ./startscen.sh SYNCCOLLECTIONACTION 001 GLOBAL
     "GLOBAL.Coll_StartTime=Start_Time_Stamp"
     "GLOBAL.Coll_EndTime=End_Time_Stamp"
     ```

   - In a Windows environment:

     ```
     startscen.bat SYNCCOLLECTIONACTION 001 GLOBAL
     "GLOBAL.Coll_StartTime=Start_Time_Stamp"
     "GLOBAL.Coll_EndTime=End_Time_Stamp"
     ```

   For this command, ODI_HOME is the path of ODI home.

   > **Note:** Make sure that Java home is set in the path before you run this command by running the **source <aia.home>/bin/aiaenv.sh** command.

BRM Interfaces

The synchronizing credit alerts integration uses the following BRM interface:

- PCM_OP_COLLECTIONS_SET_ACTION_STATUS

See *Oracle Communications Billing and Revenue Management Opcode Flist Reference* for more information.

Siebel CRM Interfaces

The synchronizing credit alerts integration uses this Siebel CRM interface:

- Outbound web service UpdateCreditAlertSiebelCommsReqABCSImplServicePort operation UpdateCreditAlert

See "Web Services Reference" in *Siebel Order Management Guide Addendum for Communications* for more information.

Industry Oracle AIA Components

The synchronizing credit alerts integration uses the following delivered enterprise business objects (EBOs) and enterprise business messages (EBMs):

- CreditAlertEBO
- UpdateCreditAlertListEBM

The industry EBO and EBM XML schema (EBM XSD) files are located under the Metadata Services (MDS) repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/Communications/EBO/

The industry enterprise business service (EBS) WSDL files are located under the MDS repository at: $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Communications/EBO/

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

See "Configuring and Using Oracle Enterprise Repository as the Oracle SOA Repository" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on how to use and configure OER to provide the Oracle AIA Reference Doc link.

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

Integration Services

These services are delivered as part of the process integration for collection management:

- UpdateCreditAlertSiebelCommsReqABCSImpl
- UpdateCreditAlertBRMCommsProvABCSImpl
- SyncCollectionsInfoChangeBRMAQ
- SyncCollectionHeaderInfoBRMCommsReqImpl

UpdateCreditAlertSiebelCommsReqABCSImpl

UpdateCreditAlertSiebelCommsReqABCSImpl is a Business Process Execution Language (BPEL) process. It performs the following actions:

- Receives Siebel Credit Alert ABM containing billing profile ID, credit alert ID, integration ID, and status.
- Transforms the ABM to UpdateCreditAlertListEBM and starts the UpdateCreditAlertBRMCommsProvABCSImpl service.

UpdateCreditAlertBRMCommsProvABCSImpl

UpdateCreditAlertBRMCommsProvABCSImpl is a BPEL process. It performs the following actions:

- Receives UpdateCreditAlertBRMCommsProvABCSImpl from the requestor service.
- Transforms the UpdateCreditAlertListEBM to BRM opcode PCM_OP_COLLECTIONS_SET_ACTION_STATUS flist.
- Starts BRM PCM_OP_COLLECTIONS_SET_ACTION_STATUS and updates the status of the credit alert or action to **Closed**.

SyncCollectionsInfoChangeBRMAQ

SyncCollectionsInfoChangeBRMAQ is a mediator based service. It performs the following actions:

- Reads the message from AQ event message published by BRM.
- Starts SyncCollectionHeaderInfoBRMCommsReqImpl

SyncCollectionHeaderInfoBRMCommsReqImpl

SyncCollectionHeaderInfoBRMCommsReqImpl a BPEL process. It performs the following actions:

- Receives the timestamp values.
- Reads the properties from the **AIAConfigurationProperties.xml** file
- Transforms the input message to an ODI web service message type.
- Starts the operation of the ODI web service.

Integrating Collections in a Multischema Environment

In BRM, you can distribute your data amongst multiple schemas in the same database to increase scalability and support load balancing. To distribute BRM accounts and associated objects amongst multiple schemas, you can migrate them between schemas in batches, with each batch consisting of a list of accounts identified by their BRM Portal Objects (POID), source and destination schema numbers.

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.

To communicate the correct account information between BRM and Siebel CRM, a common Oracle AIA identifier and an Oracle AIA cross-reference table is used to map accounts between BRM and Siebel CRM. The Oracle AIA cross-reference table uses BRM POIDs that include the BRM schema number. The BRM POID is made of a schema number, an object type, a unique object ID, and a revision number. For example:

```
0.0.0.2 /account 11599 4
```

where:

- **0.0.0.2** is the BRM schema number
- **/account** is the object type

- **11599** is the unique object ID
- **4** is the revision number

To integrate collections with Siebel CRM in a multischema environment, BRM populates the custom views UNIFIED_COLL_ACTION_IF_VIEW and UNIFIED_COLL_SCENARIO_IF_VIEW in BRM collections tables with the POID schema number, hard-coded as 0.0.0.1 for **/collections_action** regardless of the schema where the account resides. For example, for the account in the secondary schema shown in the example above, the value for a late fee would be **0.0.0.1/collections_action/late_fee 100845 0** in the UNIFIED_COLL_ACTION_IF_VIEW.

By including the hard-coded POID schema number, the schema numbers of the action objects being published to Siebel CRM do not change in the custom views, even if the action objects are migrated from one schema to another. Siebel CRM always reads the objects as residing in schema 0.0.0.1, whether it is a single-schema or a multischema environment. During the Synchronizing Credit Alerts integration flow, when synchronizing credit alerts from Siebel CRM to BRM, the integration reads the schema number for the account object POID to get the appropriate schema number for the /collections_action POIDs.

Part II

Configuring the Delivered Integrations

Part II includes the following chapters and appendixes:

- Configuring the Process Integration for Billing Management
- Configuring the Process Integration for Collections Management
- Mapping Siebel Billing Management UI Elements to BRM Customer Center
- Composite Application Validation System Changes
- Reintroducing Enterprise Business Services

9

Configuring the Process Integration for Billing Management

This chapter discusses how to set up Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM) for integrated billing management.

It also discusses how to set up BRM, Oracle Business Intelligence Publisher (BI Publisher), and Oracle WebCenter Content for PDF invoice display.

In addition, it discusses how to work with domain value maps (DVMs) and cross-references, how to handle errors, and how to configure the process integration for billing management.

Setting Up BRM for Integrated Billing Management

To set up BRM for integrated billing management:

- Configure the BRM JCA adapter as described in *Oracle Communications Billing and Revenue Management JCA Resource Adapter*.
- To ensure that resource balances with infinite effectivity show a null date (instead of 31-Dec-1969/01-Jan-1970), set the ZeroEpochAsNull BRM JCA parameter (in JCA Resource Adapter connection factory) to **True**.
- To display billing dates in Siebel CRM in the same time zone as the billing system server time, set the InteractionTimeZone parameter to the time zone of the BRM server. The InteractionTimeZone parameter in the JCA Adapter controls the time zone conversion for dates that are returned by BRM for billing queries.

See *Oracle Communications Billing and Revenue Management JCA Resource Adapter* for more information about the ZeroEpochAsNull and InteractionTimeZone JCA parameters.

Setting Up Siebel CRM for Integrated Billing Management

To set up Siebel CRM for integrated billing management:

- Set the UTCCanonical process property to *Y* for the Siebel CRM interfaces described in the instructions for ACR 474 and ACR 508 in *Siebel Maintenance Release Guide*.
- Configure the SWICreateAdjustment Siebel CRM outbound workflow to enqueue the Siebel messages in the AIA_CMUREQADJIOJMSQUEUE queue for the CreateAdjustment flow.

See *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* and *Siebel Application Integration for Oracle Fusion Middleware Guide* for more information about configuring queueing in Siebel CRM and Oracle AIA.

Setting Up PDF Invoice Display

The process integration for billing management lets you display PDF invoices directly in Siebel CRM using BRM, Oracle Business Intelligence Publisher (BI Publisher), and Oracle WebCenter Content. For more information about the applications and processes involved in displaying PDF invoices, see "QueryPDFInvoice Integration Flow."

This section describes how to set up the integration between the applications. It assumes that BRM, the BRM third-party software, Siebel CRM, BI Publisher, and WebCenter Content are all installed.

For information about installing and using BRM, BI Publisher, and WebCenter Content to generate and store invoices, including setting up notifications and scheduling invoice generation, see the documentation for the individual applications.

Setting Up the Integration Between BRM and BI Publisher

To set up the integration between BRM and BI Publisher to display PDF invoices, you perform tasks in BRM and in BI Publisher as described in this section.

BRM Setup Tasks

Setting up BRM for integration with BI Publisher involves the following high-level tasks, which are described in detail in the chapter about designing and generating invoices using BI Publisher in *BRM Designing and Generating Invoices*:

1. Specifying BI Publisher invoice report and template names in BRM.
2. Installing BRM-BI Publisher invoicing integration package.
3. Configuring the **pin_inv_doc_gen** utility.

   When performing this task, while creating the XMPL_INVOICE_JOB table in the scheduler database, after creating the invoiceXREF data source as a generic data source, create a new generic data source for each database schema, giving it the following details:

   - Name: **0.0.0.***n*
   - JNDI name: **jdbc/OracleBRM/0.0.0.***n*

   where *n* is the schema number.
4. Configuring the BRM-BI Publisher invoicing integration to support multischema systems.

   When performing this task, after creating data sources for each BRM schema in the BI Publisher Enterprise client, create an additional data source, giving it the following details:

   - Data Source Name: **invoiceXREF**
   - JNDI Name: **jdbc/OracleBRM/invoiceXREF**

> **Note:** You can create the schema data sources and the invoiceXREF data source as either JDBC connections or JNDI connections, depending on your WebLogic Server configuration. See the discussion of setting up data sources in *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher* for more information about setting up JDBC and JNDI connections.

5. Specifying the default format in which to store invoices in BRM. Use XML format.

BI Publisher Setup Tasks

To set up BI Publisher for integration with BRM, you perform the following tasks:

- Create JNDI connections. See "Creating BI Publisher JNDI Connections."
- Enable Remote JDBC Connection. See "Enabling Remote JDBC Connection."
- Add BRM Reports to the BI Publisher catalog and set up a shared folder for clustered environments. See "Configuring the BI Publisher Server and Scheduler."
- Set up the data model. See "Setting Up the BRM Sample Data Model."
- Customize the default SQL queries. See "Customizing SQL Queries."

Creating BI Publisher JNDI Connections

To create the BI Publisher JNDI connections that connect BI Publisher to BRM:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, select the **Administration** link.

   The Administration page appears.
3. Under Data Sources, select the **JNDI Connection** link.
4. Click **Add Data Source**.
5. Enter values in the following fields:
   - In the **Data Source Name** field, enter the name of the BRM schema to which BI Publisher should connect. For example, **0.0.0.1**.
   - In the **JNDI Name** field, enter **jdbc/OracleBRM/***schema_name*.

   These are the same values that you entered when creating the additional generic data source for BRM in the WebLogic Server Administration Console, as described in "BRM Setup Tasks".
6. Click **Test Connection**.

   A message confirming that the test was successful appears at the top of the page.

   If an error appears, verify that you entered the correct schema and JNDI name and try again.
7. Click **Apply**.
8. Click **Add Data Source**.
9. Enter values in the following fields:
   - In the **Data Source Name** field, enter **invoiceXREF**.
   - In the **JNDI Name** field, enter **jdbc/OracleBRM/invoiceXREF**.

These are the same values that you entered when creating the invoiceXREF generic data source for BRM in the WebLogic Server Administration Console.

10. Click **Test Connection**.

    A message confirming that the test was successful appears at the top of the page.

    If an error appears, verify that you entered the correct schema and JNDI name and try again.

Enabling Remote JDBC Connection

To enable remote JDBC connection:

1. Open the **setDomainEnv.sh** file located in the following directory on the system where BI Publisher is installed:

   Middleware_home/**user_projects/domains/***bip_domain***/bin**

2. Search for the following line:

   ```
   WLS_JDBC_REMOTE_ENABLED="-Dweblogic.jdbc.remoteEnabled=
   ```

3. Set the value to **true**. For example:

   ```
   WLS_JDBC_REMOTE_ENABLED="-Dweblogic.jdbc.remoteEnabled=true"
   ```

4. Save and close the file.
5. Restart the BI Publisher server.

Configuring the BI Publisher Server and Scheduler

This procedure assumes you have enabled the BRM-BI Publisher invoicing integration package application as described in *BRM Designing and Generating Invoices*.

To configure the BI Publisher server and scheduler:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, select the **Administration** link.

   The Administration page appears.

3. Under System Maintenance, select the **Server Configuration** link.
4. In the Catalog area, do the following:

   a. From the **Catalog Type** menu, select **Oracle BI EE Catalog**.

   b. Click **Test Connection**.

      A confirmation that the test was successful appears at the top of the page.

   c. Click **Upload to BI Presentation Catalog**.

      BI Publisher adds the BRM reports to the catalog.

5. For BI Publisher deployments in clustered environments, do the following:

   a. In the General Properties area, in the **System Temporary Directory** field, enter the path to a directory to be shared by all the servers in the cluster. You must have write permission for this directory.

   b. Click the **Scheduler Configuration** tab.

   c. In the JMS Configuration area, in the **Shared Directory** field, enter the path to the same directory that you entered in the **System Temporary Directory** field.

d. Click **Test JMS**.

   If the test is successful, a confirmation message appears at the top of the page.

   If the test fails, ensure that you have write permission for the folder and try again.

e. Restart the managed servers and the administration server.

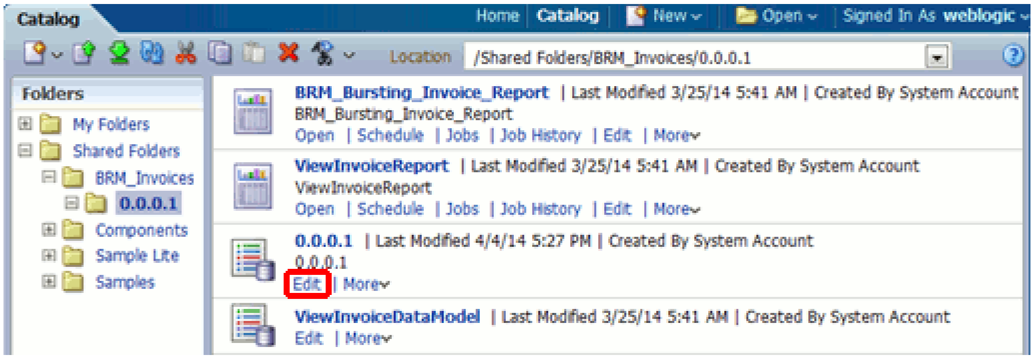
Setting Up the BRM Sample Data Model

To set up the BRM sample data model:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, select the **Catalog** link.

   The Catalog page appears.

3. In the Folders section, expand **Shared Folders**, then **BRM_Invoices**, and then select the folder for the BRM schema for which you are setting up the integration.
4. Under the object representing the BRM schema's data model, click. **Edit**:



BRM_Bursting_Invoice_Report | Last Modified 3/25/14 5:41 AM | Created By System Account
BRM_Bursting_Invoice_Report
Open | Schedule | Jobs | Job History | Edit | More

ViewInvoiceReport | Last Modified 3/25/14 5:41 AM | Created By System Account
ViewInvoiceReport
Open | Schedule | Jobs | Job History | Edit | More

0.0.0.1 | Last Modified 4/4/14 5:27 PM | Created By System Account
0.0.0.1
Edit | More

ViewInvoiceDataModel | Last Modified 3/25/14 5:41 AM | Created By System Account
Edit | More

   The data model editor page appears.

5. In the component section, expand **Data Sets** and select **BRM_Bursting_Invoice_Report_Data_Set**.
6. Click the **Data** tab.
7. Enter values in the **Minimum Invoice POID ID** field and **Maximum Invoice POID ID** field.

   You can enter any value here. The BRM request that invokes BI Publisher includes minimum and maximum values that override the values entered here.

8. Click **View** to preview the changes.
9. Click **Save As Sample Data**.

   A confirmation dialog box appears.

10. Click **OK**.

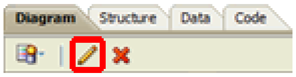
Customizing SQL Queries

To customize the SQL queries:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, select the **Catalog** link.

3. In the Folders section, expand **Shared Folders**, then **BRM_Invoices**, and then select the folder for the BRM schema for which you are setting up the integration.
4. Under the object representing the BRM schema's data model, select **Edit**.

   The data model editor page appears.
5. In the components section, expand **Data Sets** and select **BRM_Bursting_Invoice_Report_Data_Set**.
6. In the **Diagram** tab, click the edit icon:



7. Replace the default SQL query with the following query. The bold elements are metadata files and tables required for the integration.

```
select inv.to_clob("INVOICE_FORMATS_BUF_T"."BUFFER_BUF") as "BUFFER_BUF",
       "INVOICE_FORMATS_BUF_T"."OBJ_ID0" as "OBJ_ID0",
       "INVOICE_T"."STATUS" as "STATUS",
       "INVOICE_T"."CREATED_T" as "CREATED_T",
       "BILLINFO_T"."PARENT_FLAGS" as "PARENT_FLAGS",
       "INVOICE_T"."BILL_NO" as "BILL_NO",
       "BILL_T"."TOTAL_DUE" "TOTAL_DUE",       To_Char(To_
Date('19700101000000','YYYYMMDDHH24MISS') +"BILL_T"."DUE_T"/86400, 'YYYY-MM-DD
HH:MI:SS') "DUE_DATE",       To_Char(To_
Date('19700101000000','YYYYMMDDHH24MISS') +"INVOICE_T"."BILL_DATE_T"/86400,
'YYYY-MM-DD HH:MI:SS') "BILL_DATE",       "ACCOUNT_NAMEINFO_T"."FIRST_NAME"|| '
' || "ACCOUNT_NAMEINFO_T"."MIDDLE_NAME" || ' ' ||"ACCOUNT_NAMEINFO_T"."LAST_
NAME" "CONTACT_NAME",       "ACCOUNT_T"."ACCOUNT_NO" "ACCOUNT_NAME",
       To_Char(To_Date('19700101000000','YYYYMMDDHH24MISS') + "BILL_T"."START_
T"/86400, 'YYYY-MM-DD - ') || To_Char(To_
Date('19700101000000','YYYYMMDDHH24MISS') + "BILL_T"."END_
T"/86400,'YYYY-MM-DD') "BILL_PERIOD"
from "BILLINFO_T" "BILLINFO_T",
       "INVOICE_T" "INVOICE_T",
       "INVOICE_FORMATS_BUF_T" "INVOICE_FORMATS_BUF_T",
       "ACCOUNT_T" "ACCOUNT_T",
       "BILL_T" "BILL_T",       "ACCOUNT_NAMEINFO_T" "ACCOUNT_NAMEINFO_T"
where "INVOICE_FORMATS_BUF_T"."OBJ_ID0"="INVOICE_T"."POID_ID0"
   and "INVOICE_T"."BILLINFO_OBJ_ID0"="BILLINFO_T"."POID_ID0"
   and "BILL_T"."BILLINFO_OBJ_ID0" = "BILLINFO_T"."POID_ID0"
   and "BILL_T"."ACCOUNT_OBJ_ID0" = "ACCOUNT_T"."POID_ID0"
   and "ACCOUNT_NAMEINFO_T"."OBJ_ID0" = "ACCOUNT_T"."POID_ID0"
   and "BILL_T"."BILL_NO" = "INVOICE_T"."BILL_NO"
   and "INVOICE_T"."STATUS" = :p_Inv_Status
   and "INVOICE_T"."CREATED_T" >= :p_Inv_Start_t
   and "INVOICE_T"."CREATED_T" < :p_Inv_End_t
   and "INVOICE_FORMATS_BUF_T"."OBJ_ID0" >= :p_Inv_MIN_POID
   and "INVOICE_FORMATS_BUF_T"."OBJ_ID0" <= :p_Inv_MAX_POID
   and  ((:p_Inv_Type = 0 and "BILLINFO_T"."PARENT_FLAGS" = 0)
             or(:p_Inv_Type = 1 and "BILLINFO_T"."PARENT_FLAGS" > 0)
             or(:p_Inv_type = 2))
```

8. Click **OK**.
9. Click the **Save** icon.
10. In the component section, expand **Bursting** and select **BRM_Bursting_Invoice_Report_Bursting_Query**.

11. From the **Data Source** menu, select the data source for the BRM schema created in step 5 of "Creating BI Publisher JNDI Connections".

12. In the SQL Query field, replace the default SQL query with the following query. The bold parameters are required for the integration.

```
select  "INVOICE_T"."POID_ID0" as "KEY",
        "INVOICE_T"."TEMPLATE_NAME" as "TEMPLATE",
        'en-US' as "LOCALE",
        'PDF' as "OUTPUT_FORMAT",
        'WCC' as "DEL_CHANNEL",
        'America/Los_Angeles' as "TIMEZONE",
        'GREGORIAN' as "CALENDAR",
        "INVOICE_T"."BILL_NO" as "OUTPUT_NAME",
        'true' as "SAVE_OUTPUT",
        'WCC' as "PARAMETER1",
        'custInvGroup' as "PARAMETER2",
        'BRM_BIP' as "PARAMETER3",
        '' as "PARAMETER4",
        'Invoice_' ||"INVOICE_T"."BILL_NO" as "PARAMETER5",
        'Invoice_'||"INVOICE_T"."BILL_NO"||'.pdf' as "PARAMETER6",
        '' as "PARAMETER7",
        '' as "PARAMETER8",
        'true' as "PARAMETER9"
 from   "INVOICE_T","PAYINFO_T","PAYINFO_INV_T"
where   "INVOICE_T"."ACCOUNT_OBJ_ID0"="PAYINFO_T"."ACCOUNT_OBJ_ID0"
  and   "PAYINFO_T"."POID_ID0"="PAYINFO_INV_T"."OBJ_ID0"
  and   "PAYINFO_INV_T"."DELIVERY_PREFER"=0
  and   "INVOICE_T"."STATUS"    =  :p_Inv_Status
  and   "INVOICE_T"."CREATED_T" >= :p_Inv_Start_t
  and   "INVOICE_T"."CREATED_T" <  :p_Inv_End_t
  and   "INVOICE_T"."POID_ID0"  >= :p_Inv_MIN_POID
  and   "INVOICE_T"."POID_ID0"  <= :p_Inv_MAX_POID
```

13. Click the **Save** icon.

Table 9–1 describes the parameters in the SQL query that are relevant to the integration. You can use the values of these parameters to locate the invoices in WebCenter Content.

Table 9–1 BRM Bursting Query SQL Parameters

Parameter	Status	Description
DEL_CHANNEL	Required	The delivery channel. Use **WCC**.
SAVE_OUTPUT	Required	Determines if BI Publisher saves the document internally. Oracle recommends setting this to **true** to enable you to re-send invoices if the upload to WebCenter Content fails.
PARAMETER1	Required	The name of the WebCenter Content server. Use **WCC**.
PARAMETER2	Required	The WebCenter Content security group that has permission to write to the WebCenter Content server. Oracle recommends using the custInvGroup group, which is created when you perform the procedure in "Creating WebCenter Content Metadata, Content Type, and Security Elements." Alternatively, you can use any custom security group that you have set up.
PARAMETER3	Required	The author of the PDF invoice that will be generated. Any string is valid.

Table 9–1 (Cont.) BRM Bursting Query SQL Parameters

Parameter	Status	Description
PARAMETER4	Optional	An account name to associate with the PDF invoice that will be generated. Any string is valid.
PARAMETER5	Required	The title of the PDF invoice that will be generated. Any string is valid.
PARAMETER6	Required	The file name of the PDF invoice that will be generated and saved to the WebCenter Content server. Any string ending in **.pdf** is valid.
PARAMETER7	Optional	Comments to associate with the PDF invoice that will be generated . Any string is valid.
PARAMETER8	Optional	A customer ID to associate with the PDF invoice that will be generated. Any string is valid.
PARAMETER9	Required	Determines if custom data is enabled. Use **true**.

Setting Up the Integration Between BI Publisher and WebCenter Content

To set up the integration between BI Publisher and WebCenter Content to display PDF invoices, you perform tasks in BI Publisher and in WebCenter Content as described in this section.

This section assumes you have installed both BI Publisher and WebCenter Content.

WebCenter Content Setup Tasks

To set up WebCenter Content to integrate with BI Publisher:

- Add the BI Publisher Server. See "Adding the BI Publisher Server to the WebCenter Content Security Filter."
- Add metadata, content type, and security information. See "Creating WebCenter Content Metadata, Content Type, and Security Elements."
- Add the Oracle AIA component that connects to WebCenter Content. "Adding the AIA_GETFILE Component to WebCenter Content."
- Add a rule and content profile. See "Adding the Rule and Content Profile for Integration with BI Publisher."

Adding the BI Publisher Server to the WebCenter Content Security Filter

To add the BI Publisher server to the WebCenter Content security filter, do one of the following:

- If you have not completed the initial configuration of Content Server as described in *Oracle WebCenter Content Installation Guide*, do the following when you configure the Content Server instance:

  Add the IP address of the BI Publisher server in the **Incoming Socket Connection Address Security Filter** field.

- If you have already completed the initial configuration, add the BI Publisher server using Oracle Enterprise Manager Fusion Middleware Control:

  a. Log in to Oracle Enterprise Manager Fusion Middleware Control for the server where WebCenter Content is running.

  b. In the Target Navigation tree, expand **Web Center**, then **Content**, and then **Content Server**.

c. Click the link for the Web Center Content server.

d. From the **Content Server** menu, select **Configuration Pages**, and then **Internet Configuration**.

e. In the **IP Address Filter** field, add the IP address of the BI Publisher server to the pipe-delimited list. Ensure that there are no spaces between the numbers and the pipes.

   For example:

   ```
   127.0.0.1|0:0:0:0:0:0:0:1|192.0.2.1|192.0.2.254
   ```

f. Click **Apply**.

g. Restart the WebCenter Content administration server and managed servers.

Creating WebCenter Content Metadata, Content Type, and Security Elements

This procedure creates the metadata, security group, role, and document type required for WebCenter Content custom metadata. This includes the custInvGroup security group and the custInvReadWriteRole security role, which BI Publisher uses when mapping BRM data to WebCenter Content metadata.

To create the WebCenter Content metadata, content type, and security elements:

1. From the directory where you installed the Pre-Built Integration for Agent Assisted Billing Care, unzip the following file:

   ```
   AIA_home/services/industry/Communications/JavaServices/ebp/WCC_Customization.zip
   ```

   The **WCC_Customization.zip** file contains the **AIA_GETFILE.zip** file and the **WCCProvisioning** directory.

2. Copy the **WCCProvisioning** directory to the directory for the WebCenter Content server.

3. From the **WCCProvisioning** directory located in the WebCenter Content server directory, open one of the following files in a text editor:

   - (Linux) **WCCProvisioning.sh**
   - (Windows) **WCCProvisioning.bat**

4. Search for the following line:

   ```
   WCCURL="idc://WCC_SERVER_NAME:4444"
   ```

5. Replace **WCC_SERVER_NAME:4444** with the name and port of your WebCenter Content server.

6. Search for the following line:

   ```
   WCCUSER=
   ```

7. Ensure that the user listed for this parameter has administrator privileges.

8. Save and close the file.

9. Copy the following files to the **WCCProvisioning** directory located in the WebCenter Content server directory:

   - *WCC_home*/**ucm/Distribution/RIDC/oracle.ucm.ridc-11.1.1.jar**
   - *WCC_home*/**ucm/Distribution/FAProv/ucmscript.jar**

10. Run one of the following scripts:

    - (Linux) **WCCProvisioning.sh**
    - (Windows) **WCCProvisioning.bat**

    The metadata, security group, role, and document type required for WebCenter Content custom metadata are created.

Adding the AIA_GETFILE Component to WebCenter Content

To add the AIA_GETFILE component to WebCenter Content:

1. Log in to your WebCenter Content Server instance.
2. Expand the **Administration** tray and then the **Admin Server** element.
3. Select the **Component Manager** link.
4. In the first paragraph under Component Manager, select the **advanced component manager** link.

   The advanced Component Manager page appears.
5. In the **Install New Component** field, click **Browse**.
6. Browse to the **AIA_GETFILE.zip** file in the directory for the WebCenter Content server.
7. Click **Install**.
8. Click **Continue**.

   A message appears confirming that the AIA_GETFILE component was uploaded and installed.
9. Click the **Click here to enable the component and go to the page to restart the server** link.

   The Component Manager screen appears.
10. Select the **AIA_GETFILE** option and click **Update**.
11. Restart WebCenter Content Server as described in the chapter about managing system processes in *Oracle Fusion Middleware Administering Oracle WebCenter Content*.

Adding the Rule and Content Profile for Integration with BI Publisher

To add the rule and content profile that get the custom metadata from BI Publisher:

1. Log in to the WebCenter Content Server console.
2. Expand the **Administration** tray and select **Admin Applets**.
3. Select **Configuration Manager**.
4. Click the **Rules** tab.
5. Click **Add**.
6. In the **Name** field, enter **custInvRule**.
7. In the **Description** field, enter a description of the rule. For example:

   ```
   Rule created to add metadata for BRM invoices.
   ```
8. Click the **Fields** tab.
9. Select the **Display information fields** option.

10. For each item in the **Field Name** menu that you want to include as metadata on invoices, do the following:

    a. Select the item and click **OK**.

       The Add Rule Field page appears.

    b. Leave all fields blank and click **OK**.

11. Click **OK**.

    The rule is saved.

12. Click the **Profiles** tab.

13. Click **Add**.

14. In the **Name** field, enter **custInvProfile**.

15. Click **OK**.

16. In the **Display Label** field, enter **Customer Invoice Profile**.

17. In the **Description** field, enter a description of the profile. For example:

    ```
    Profile created to add custom metadata for BRM invoices.
    ```

18. In the **Trigger** field, enter **custInvTrigger**.

19. Click **Add**.

20. From the **Name** menu, select **custInvRule**.

21. Click **OK**.

    The rule is added to the profile.

22. Click **OK**.

    If an error message indicates that the profile could not be added because of an invalid trigger value, click **OK** to ignore it.

    The profile is saved.

BI Publisher Setup Tasks

To set up BI Publisher to integrate with WebCenter Content:

- Enable the integration from BI Publisher to WebCenter Content in the **xmlp-server-config.xml** file. See "Enabling the Integration Between BI Publisher and WebCenter Content."
- Add WebCenter Content as a delivery destination. See "Adding WebCenter Content as a Delivery Destination."
- Set up custom invoice metadata. See "Setting Up Custom Metadata."

Enabling the Integration Between BI Publisher and WebCenter Content

To enable integration from BI Publisher to WebCenter Content:

1. Open the **xmlp-server-config.xml** file located in the following directory:

   Middleware_home/**user_projects/domains/***BIP_domain*/**config/bipublisher/repository/Admin/Configuration**

   where:

   - *Middleware_home* is the location where Oracle Fusion Middleware is installed.

- *BIP_domain* is the directory that contains the configuration for the domain onto which BI Publisher is installed.

2. Add the following entry to the list of properties:

```
<property name="WCC_ENABLED" value="true"/>
```

3. Save and close the file.
4. Restart the BI Publisher administration server and managed servers.

Adding WebCenter Content as a Delivery Destination

To add WebCenter Content as a delivery destination:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, click the **Administration** link.

   The Administration page appears.
3. Under **Delivery**, click the **Content Server** link.
4. Click **Add Server**.
5. In the **Server Name** field, enter **WCC**.
6. In the **URI** field, enter the Uniform Resource Identifier for the WebCenter Content server.
7. In the security area, in the **Username** and **Password** fields, enter the user name and password for the WebCenter Content user that has privileges to write PDFs to WebCenter Content.
8. Select the **Enable Custom Metadata** option. This enables BI Publisher to update metadata in WebCenter Content.
9. Click **Test Connection**.

   If the test is successful, a confirmation message appears at the top of the page.

   If the test fails, ensure that you entered correct values and try again.
10. Click **Apply**.

Setting Up Custom Metadata

To set up custom invoice metadata in BI Publisher:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, click the **Catalog** link.

   The Catalog page appears.
3. In the Folders section, expand **Shared Folders**, then **BRM_Invoices**, and then select the folder for the BRM schema for which you are setting up the integration.
4. Select the name of the object representing the data model for the BRM schema.

   The data model editor appears.
5. In the component section, select **Custom Metadata**.

   The metadata details appear in the work section.
6. From the **Server** menu, select **WCC**.
7. From the **Content Profile** menu, select **custInvProfile**.

8. From the **Rules** menu, select **custInvRule**.
9. Click **Load Metadata**.

   The metadata fields appear in the Meta Data table.
10. From the menu for each row in the Data Mapping column, select the XPath to the corresponding BRM data.

    For example, for the xBillNo row, select **/ROW_SET/LIST_ROW/ROW/BILL_NO**.
11. Click the **Save** icon.

Setting Up the Integration Between WebCenter Content and Siebel CRM

To set up the integration between WebCenter Content and Siebel CRM to display PDF invoices, set the symbolic URL in Siebel CRM to the WebCenter Content server.

See the discussion of using a WebCenter Content host while setting up the invoice image feature in *Siebel Communications Guide for Siebel Innovation Pack 2014* for information about how to set the symbolic URL.

When performing this task, when you change the values for the records in the Symbolic URL Arguments list to accommodate your WebCenter Content host, use the values shown in Table 9–2.

Table 9–2 WebCenter Content Symbolic URL Arguments

Name	Required	Type	Value	Append as Argument	Sequence #
ReplaceResponseWithForm	Yes	Constant	ReplaceResponseWithForm	Yes	1
IdcService	Yes	Constant	**AIA_GETFILE**	Yes	2
allowInterrupt	Yes	Constant	1	Yes	3
RevisionSelectionMethod	Yes	Constant	latest	Yes	4
noSaveAs	Yes	Constant	1	Yes	5
BILLNO	Yes	Field	Bill Number	Yes	6
j_username	Yes	Command	UserLoginId or UserSiebelLoginId	Yes	7
j_password	Yes	Command	UserLoginPassword or UserSiebelLoginPassword	Yes	8
refresh	**Yes**	**Command**	**true**	**Yes**	**9**
FreePopup		Command	TRUE	Yes	**10**
FullWindow		Command	TRUE	Yes	**11**
PostRequest	Yes	Command	PostRequest	Yes	**12**

The fields in bold in the table represent fields you must change from the default. You must create the **refresh** argument by selecting **New Record** from the Symbolic URL Arguments menu.

Testing the Integration

You can test the integration between the applications by running on-demand jobs. You can test the integration between BI Publisher and WebCenter Content by running the job from BI Publisher and you can test the integration between BRM, BI Publisher, and WebCenter Content by running the job from BRM.

Testing the Integration Between BI Publisher and WebCenter Content

To test the integration between BI Publisher and WebCenter Content:

1. Log in to the BI Publisher Enterprise client.
2. From the global header, click the **New** menu and select **Report Job**.
3. On the **General** tab, in the **Report** field, click the magnifying glass icon.

   The Open dialog box appears.
4. Under Catalog, expand **Shared Folders**, then **BRM_Invoices**, and then select the folder for the BRM schema.
5. Select **BRM_Bursting_Invoice_Report**.
6. Click **Open**.
7. Click the **Output** tab.
8. Select the **Use Bursting Definition to Determine Output & Delivery Destination** option.
9. Click the **Schedule** tab.
10. From the **Frequency** menu, select **Once**.
11. Select the **Run Now** option.
12. Click **Submit**.

    The Submit dialog box appears, displaying summary information about the job.
13. In the **Report Job Name** field, enter a name for the job.
14. Click **OK**.

    A notification that the job was successfully submitted appears.
15. From the global header, click the **Open** menu and select **Report Job History**.
16. In the Report Job Histories table, select the name of your test report job.
17. Review the Output & Delivery table. The status column should contain a green check mark icon for each output row.

    If the Status column contains an error icon:

    a. Click the icon and review the error.

    b. Check that your WebCenter Content connection details are correct.

    c. Ensure that the WebCenter Content server is running.

    d. Re-send a row to WebCenter Content by clicking the **Send** icon for that row.

    > **Note:** The **Republish** button regenerates the report job using the same data but does not automatically republish the job to WebCenter Content. You must use the **Send** icon to re-send each row individually.
18. Log in to the WebCenter Content Server console.
19. Expand the **Search** tray.

20. In the **Release Date: From** field, enter the current date in *mm/dd/yy* format and click **Search**.

    All content added on the current date appears in the Search Results area.

21. Confirm that the invoice reports triggered from BI Publisher appear.

Testing the Integration Between BRM, BI Publisher, and WebCenter Content

To test the integration between BRM, BI Publisher, and WebCenter Content, generate the invoices using the BRM **pin_inv_doc_gen** utility and confirm that they appear in WebCenter Content.

For more information about using **pin_inv_doc_gen**, see the discussion of generating BI Publisher invoice documents in *BRM Designing and Generating Invoices*.

Security Recommendations for PDF Invoice Display

Because customer invoices contain sensitive information, you must ensure that the data is generated, sent, and stored securely in each application.

For information about securing each application, see:

- *Oracle Communications Billing and Revenue Management Security Guide*
- *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- The discussion of managing security and user access in *Oracle WebCenter Content System Administrator's Guide for Content Server*

High Availability Recommendations for PDF Invoice Display

To help your system generate and store large numbers of invoices with minimal failures and low performance impact, Oracle recommends deploying BI Publisher and WebCenter Content in highly-available environments using a hardware load balancer.

For information about high availability for BI Publisher and WebCenter Content, see:

- The discussion of configuring high availability for WebCenter Content in *Oracle Fusion Middleware High Availability Guide*
- The discussion of configuring high availability for BI Publisher in *Oracle Fusion Middleware High Availability Guide*
- The discussion of deploying BI Publisher for high availability in *Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*

Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle service-oriented architecture (SOA) Suite that enable you to equate lookup codes and other static values across applications, for example, FOOT and FT or US and USA.

DVMs are static in nature, though administrators can add maps as required. Transactional business processes never update DVMs-they only read from them. They are stored in XML files and cached in memory at run time.

DVM types are seeded for the Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care flows. Administrators can extend the list of mapped values by adding more maps.

Table 9–3 lists the DVMs for the process integration for billing management.

Table 9–3 Billing Management Integration - DVMs

DVM	Description
CURRENCY_CODE	Currency codes.
RESOURCE	Nonmonetary resources (Free Minutes, Text Messages, and so on).
ACCOUNTBALANCEADJUSTMENT_REASON	Reason for adjustment.
ACCOUNTBALANCEADJUSTMENT_STATUS	Status of adjustment request (Posted, Not-Posted).
ACCOUNTBALANCEADJUSTMENT_TYPE	Type of adjustment (Credit, debit, and so on).
ACCOUNTBALANCEADJUSTMENT_TAXTREATMENT	Tax treatment on adjustment amount (Include, Exclude).
ACCOUNTBALANCEADJUSTMENT_USAGEALLOCATION_TAXTREATMENT	Tax treatment on CDR adjustment amount (Include, Exclude).
INSTALLEDPRODUCT_STATUS	Status of installed product (Active, Canceled, and so on).
RECIEVEDPAYMENT_TYPE	Type of payment (Credit, Direct Debit).
ACCOUNTBALANCEADJUSTMENT_SUBSTATUS	Sub-status of adjustment request.

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

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 9–4 contains the billing management integration cross-references.

Table 9–4 Billing Management Integration - Cross-References

Name	Columns	Mapping Details	Description
CUSTOMERPARTY_ACCOUNTID	SEBL_01,COMMON,BRM_01	Set up as part of customer sync	Query
CUSTOMERPARTY_BILLPROFILEID	SEBL_01,COMMON,BRM_01	Set up as part of customer sync	Query

Handling Errors

Based on the roles defined for the services, e-mail notifications are sent if a service ends due to an error.

Table 9–5 lists the error messages provided by the process integration for billing management.

Table 9–5 Billing Management Integration - Error Messages

Integration/Service Name	Error Code	Message Text
Account Balance / QueryBalanceSummarySiebel ReqABCSImpl Query Invoice List / QueryInvoiceListSiebelCommsReqABCSImpl	AIA_ERR_AIACOMBMPI_0003	Billing Profile BPName for the account does not exist in the billing system. 1) To correct the error, submit a sales order with this billing profile. 2) Ensure that the sales order created with this billing profile is successfully submitted to the billing system.
Create Payment / CreateReceivedPaymentBRMCommsProvABCSImpl	AIA_ERR_AIACOMBMPI_0005	BRM Error Message (For example, *Service Unavailable*)

Describing Delivered Error Notification Roles and Users

The following roles and users are delivered as default values for issuing error notifications for the process integration for billing management:

- **Role:** *AIAIntegrationAdmin*
- **User:** *AIAIntegrationAdminUser*

The default password for all users is *welcome1*.

See *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for information about how to set up error notifications and trace and error logs using these values.

Configuring the Process Integration for Billing Management

Configure these properties in the **AIAConfigurationProperties.xml** file. The file is located in <AIA_INSTANCES>/config/. Entries in the **AIAConfigurationProperties.xml** file are case-sensitive.

See the discussion of building AIA integration flows in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about reloading updates to **AIAConfigurationProperties.xml** file.

The following Business Process Execution Language (BPEL) processes have entries listed in Table 9–6.

- QueryBalanceSummarySiebelCommsReqABCSImpl
- QueryCustomerPartyListBRMCommsProvABCSImpl
- QueryBalanceGroupListSiebelCommsReqABCSImpl
- QueryBalanceDetailsSiebelCommsReqABCSImpl
- QueryBalanceGroupServicesSiebelCommsReqABCSImpl
- QueryInstalledProductListBRMCommsProvABCSImpl
- QueryInvoiceListSiebelCommsReqABCSImpl
- QueryInvoiceListBRMCommsProvABCSImpl
- QueryInvoiceSiebelCommsReqABCSImpl
- QueryInvoiceEventDetailsSiebelCommsReqABCSImpl
- SearchInvoiceEventDetailsSiebelCommsReqABCSImpl

- QueryInvoiceBalanceDetailsSiebelCommsReqABCSImpl
- QueryUnbilledUsageSiebelCommsReqABCSImpl
- QueryServiceUsageListBRMCommsProvABCSImpl
- QueryUnbilledUsageEventDetailsSiebelCommsReqABCSImpl
- SearchUnbilledUsageEventDetailsSiebelCommsReqABCSImpl
- QueryUnbilledUsageBalanceDetailsSiebelCommsReqABCSImpl
- CreatePaymentSiebelCommsReqABCSImpl
- CreateInvoicePaymentSiebelCommsReqABCSImpl
- CreateReceivedPaymentBRMCommsProvABCSImpl
- QueryPaymentSiebelCommsReqABCSImpl
- QueryInvoicePaymentSiebelCommsReqABCSImpl
- SearchPaymentSiebelCommsReqABCSImpl
- QueryReceivedPaymentListBRMCommsProvABCSImpl
- QueryAccountBalanceAdjustmentSiebelCommsReqABCSImpl
- QueryAccountBalanceAdjustmentBRMCommsProvABCSImpl
- CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl

Table 9–6 BPEL Process Property Values - 1

Property Name	**Value/Default Value**	**Description**
ABCSExtension.PreXform*ABM/EBM_name*TO*ABM/EBM_name*	true/false	Controls whether the extension point before transformation of application business message (ABM) to enterprise business message (EBM) is invoked during processing.
ABCSExtension.PreInvoke*PartnerLink_name*	true/false	Controls whether the extension point before invocation to enterprise business service (EBS) is invoked during processing.
ABCSExtension.PostXform*ABM/EBM_name*to*ABM/EBM_name*	true/false	Controls whether the extension point before transformation of EBM to ABM is invoked during processing.
ABCSExtension.PostInvokePartnerLink_name	true/false	Controls whether the extension point before invocation of callback service or response return is invoked during processing.
Routing.PartnerLink_Name.RouteToCAVS	true/false	Controls whether the Composite Application Validation System (CAVS) is used to handle the request.
Default.SystemID	Valid string	Specifies the name of the default systemID of the requester application.
Routing.PartnerLink_name.BRM_01.EnpointURI	eis/BRM	Specifies the JNDI entry for the partner link.
EBSOverride.*EBS_name.operation_name*.PortType	Valid string	PortType of the webservice that needs to be invoked dynamically. This value should be consistent with the EBSOverride.*EBS_name.operation_name*.Address property.

Table 9–6 (Cont.) BPEL Process Property Values - 1

Property Name	Value/Default Value	Description
EBSOverride.*EBS_name.operation_name*.ServiceName	Valid string	ServiceName of the webservice that needs to be invoked dynamically. This value should be consistent with the EBSOverride.*EBS_name.operation_name*.Address property.
EBSOverride.*EBS_name.operation_name*.Address	Valid string	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
BRM.Payment.Command	0	This property is specific to CreateReceivedPaymentBRMCommsProvABCSImpl.
Routing.CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer.EndpointURI	Valid string	Endpoint URL of the CreateAccountBalanceAdjustmentListResponseBRMCommsJMSProducer. This property is specific to CreateAccountBalanceAdjustmentBRMCommsProvABCSImpl.

These BPEL processes have entries listed in Table 9–7.

- CreateAccountBalanceAdjustmentSiebelCommsReqABCSImpl
- UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl

Table 9–7 BPEL Processes Property Values - 2

Property Name	Value/Default Value	Description
ABCSExtension.PreXform*ABM/EBM_name*TO*ABM/EBM_name*	true/false	Controls whether the extension point before transformation of ABM to EBM is invoked during processing.
ABCSExtension.PreInvoke*PartnerLink_name*	true/false	Controls whether the extension point before invocation to enterprise business service (EBS) is invoked during processing.
Routing.*PartnerLink_name*.RouteToCAVS	true/false	Controls whether the CAVS is used to handle the request
Default.SystemID	Valid string	Specifies the name of the default systemID of the requester application
EBSOverride.*EBS_name.operation_name*.PortType	Valid string	PortType of the webservice that needs to be invoked dynamically. This value should be consistent with the EBSOverride.*EBS_name.operation_name*.Address property.

Table 9–7 (Cont.) BPEL Processes Property Values - 2

Property Name	Value/Default Value	Description
EBSOverride.*EBS_name.operation_name*.ServiceName	Valid string	ServiceName of the webservice that needs to be invoked dynamically. This value should be consistent with the EBSOverride.*EBS_name.operation_name*.Address property.
EBSOverride.*EBS_name.operation_name*.Address	Valid string	This property is used to dynamically invoke any webservice from this service. This holds the address.endpoint URI of the webservice that needs to be invoked dynamically. To invoke CAVS or any other provider ABCS, this property needs to be updated accordingly.
Routing.SWIAdjustmentStatusUpdate.SEBL_01.EndpointURI	Valid string	Siebel endpoint URIL. This property is specific to UpdateAccountBalanceAdjustmentRespSiebelCommsProvABCSImpl.

Configuring Oracle HTTP Server for Billing Management

To integrate invoice header-level adjustments, you must configure Oracle HTTP Server (OHS) so that it recognizes the Oracle AIA mirror servlet that comes with the installation.

For more information about servlets, see *Oracle Fusion Middleware Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.

To configure OHS so that it recognizes the mirror servlet:

1. Open the following file:

   ```
   WebTier_HOME/instances/instance_name/config/OHS/component_name/mod_wl_ohs.conf
   ```

   where:

   - *WebTier_HOME* is the directory in which OHS Web Tier is installed.
   - *instance_name* is the OHS Web Tier instance defined during cluster set up.
   - *component_name* is the name for the OHS component defined during cluster set up. By default, this is **ohs1**.

2. Add the following code:

   ```
   <Location  /AIA>
       SetHandler weblogic-handler
       WebLogicCluster hostname1:port,hostname2:port
       WLLogFile  /tmp/web_log.log
   </Location>
   ```

   where:

   - *hostname1* and *hostname2* are the host names of the servers in the Oracle WebLogic server cluster.
   - *port* is the port where the host is listening for HTTP requests.

3. Save and close the file.
4. Restart OHS. See *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server* for more information.

Configuring Cross-Channel Payments

To accept payments from cross-channel systems other than Siebel CRM, you must customize the integration. The artifacts in the CreateReceivedPayments integration flow support messages from different cross-channel systems, but you must configure the following components:

- The message sent from the channel to the PaymentSiebelCommsReqABCS service
- The scope of the CreateReceivedBRMCommsProvABCSImpl BPEL process
- The response message to Siebel CRM

Configuring the Message from Cross-Channel Systems

Configure the message sent from cross-channel systems to invoke the PaymentSiebelCommsReqABCS service with the CreatePayment operation.

This message must conform to the existing **PaymentSiebelCommsReqABCSImpl.wsdl** file, as shown in the following sample message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cmu="http://www.siebel.com/xml/CMU%20Request%20New%20Payment%20Capture%20IO"
>
<soapenv:Header>
  <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-sec
ext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-util
ity-1.0.xsd">
    <wsse:UsernameToken>
      <wsse:Username>weblogic</wsse:Username>
      <wsse:Password Type="wsse:PasswordText">weblogic1</wsse:Password>
      <wsu:Created>2012-11-19T08:44:51Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
   <soapenv:Body>
      <cmu:ListOfCmuRequestNewPaymentCaptureIo Language="ENU" Locale="English -
United States" MessageId="" EnterpriseServerName="siebel">
         <!--Zero or more repetitions:-->
         <cmu:CmuCaptureNewPaymentVbc>
            <cmu:AccountId>88-5E8DX</cmu:AccountId>
            <cmu:AccountName>JKNBP124858_Account1</cmu:AccountName>
            <cmu:BankAccountHoldersName></cmu:BankAccountHoldersName>
            <cmu:BankAccountHoldersNumber></cmu:BankAccountHoldersNumber>
            <cmu:BillingProfileId>88-5EBX3</cmu:BillingProfileId>
            <cmu:BillingProfileName>88-5EBX3</cmu:BillingProfileName>
            <cmu:Comments></cmu:Comments>
            <cmu:CreditCardExpirationDate>11/13/2013
            </cmu:CreditCardExpirationDate>
            <cmu:CreditCardHoldersName>aaa</cmu:CreditCardHoldersName>
            <cmu:CreditCardNumber>1111222233334444</cmu:CreditCardNumber>
            <cmu:CreditCardSecurityCode>159</cmu:CreditCardSecurityCode>
            <cmu:CurrencyCode>USD</cmu:CurrencyCode>
            <cmu:OneTimePayment>T</cmu:OneTimePayment>
            <cmu:PaymentAmount>25</cmu:PaymentAmount>
            <cmu:PaymentDate>11/12/2012</cmu:PaymentDate>
            <cmu:PaymentMethod>Credit</cmu:PaymentMethod>
            <cmu:RoutingNumber></cmu:RoutingNumber>
            <cmu:ServiceAccountId>88-5E8DX</cmu:ServiceAccountId>
         </cmu:CmuCaptureNewPaymentVbc>
      </cmu:ListOfCmuRequestNewPaymentCaptureIo>
   </soapenv:Body>
</soapenv:Envelope>
```

The **OneTimePayment** property determines how the integration handles payment authorization. The possible values for the **OneTimePayment** property are:

- **Y**: BRM communicates with a third-party gateway to authorize the payment.
- **T**: The integration invokes a third-party system to authorize the payment
- **O**: No authorization is required by the integration. For example, an external retail point-of-sale system could authorize the payment.

Creating Extension Points to Third-Party Payment Authorization Systems:

To create an extension point to a third-party payment authorization system:

1. Using JDeveloper, customize the scope of the CreateReceivedCommsProvABCSImpl process by creating the following elements:
   - The partner link to the third-party system
   - A new local input variable for the Invoke activity
   - An Assign activity called **RequestAssign_CreditCheck** to set the values in the input variable
   - The Invoke activity to call the third-party partner link
   - An Assign activity called **ResponseAssign_CreditCheck** to set the values in the output variable
   - A transformation to update the PCM_OP_PYMT_COLLECT payload to either authorize and record the payment or just record the payment, depending on the incoming message
2. Configure the scope to raise a BPEL exception to return an error to the cross-channel system without invoking BRM if authorization fails.

See the discussion of designing extension points in the ABCS BPEL process in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about creating the partner link, activities, and variables.

Configuring the Response Message to Siebel CRM

You can optionally configure the integration to track the payment failure or success response in Siebel CRM as an activity on the customer account.

To configure the response message to Siebel CRM:

1. In Siebel CRM, create an activity on the customer account to track the payment response. See *Siebel Customer Relationship Console User Guide* for more information about adding activities.
2. In JDeveloper, in the scope of the existing PaymentSiebelCommsReqABCS service, create the call to the Siebel CRM API.
3. Set the EnterpriseServerName property as follows:

   ```
   <EnterpriseServerName>"SiebelAPI"</EnterpriseServerName>
   ```

10

Configuring the Process Integration for Collections Management

This chapter discusses how to set up Oracle Communications Billing and Revenue Management (BRM) and Siebel customer relationship management (Siebel CRM). In addition, it discusses how to work with domain value maps (DVMs) and cross references, how to handle errors, and how to configure the process integration for collections management.

Setting Up BRM

To set up BRM:

1. Specify pay types to process.
2. Set the minimum overdue balance to process.
3. Set the number of bill units retrieved during step searches.
4. Set up invoice reminders.
5. Define collections features.
6. Configure how Collections Manager determines dates.

   See the discussion of setting up Collections Manager in *BRM Collections Manager Guide* for more information.
7. Create views on BRM and configure pin_collections_process to record start and end time.

Setting up Collection Action Names in BRM

As stated in "Process Integration for Collection Management Overview" the process integration for collection management synchronizes collection actions between BRM and Siebel CRM. The collection actions are mapped to Siebel CRM as credit alerts using the **DVM COLLECTION_ACTIONNAME.xml** file.

As delivered, DVM mappings exist for the following collections actions:

- Impose Late Fee
- Courtesy Phone Call
- Courtesy Email or SMS Reminder
- Courtesy Dunning Letter
- Demanding Phone Call

- Inactivate Services of Billinfo
- Harsh Dunning Letter
- Refer to outside agency
- Writeoff Billinfo
- Close Services of Billinfo
- Inactivate the Account

Of the 11 collections actions, these 4 collections actions are seeded in BRM and are available as part of the product installation.

- Inactivate Services of Billinfo
- Close Services of Billinfo
- Writeoff Billinfo
- Refer to outside agency

Before you use the process integration, the implementer has the option of either adding these remaining 7 collection actions in BRM or run the collection integration process with the 4 seeded values listed previously. If the implementer's intent is to add these additional collection actions, then they must be entered manually in Collection Manager:

- Impose Late Fee
- Courtesy Phone Call
- Courtesy Email or SMS Reminder
- Courtesy Dunning Letter
- Demanding Phone Call
- Harsh Dunning Letter
- Inactivate the Account

Caution: Use caution when manually entering these action names in BRM because any change in spelling or case breaks the Sync Collection Action integration flow. The DVM lookup fails due to a mismatch of strings.

Note: If the intent of the implementer is to use new or different collection action names, then explicit changes must be made in the COLLECTION_ACTIONNAME DVM before you use the process integration.

Setting Up Siebel CRM

For some Siebel CRM interfaces, in Siebel, you must set the process property UTCCanonical to **Y**.

See instructions for ACR 474 and ACR 508 in *Siebel Maintenance Release Guide* for more information about which Siebel CRM interfaces require you to enable the UTCCanonical process property.

The credit alert appears in the customer service supervisor's queue by default, and the supervisor assigns the credit alert to a customer service representative (CSR). Siebel CRM can also be customized to automatically assign these credit alerts to a CSR based on a set of criteria as defined by the service provider.

To set up Siebel CRM:

1. Extract and copy the .ifb files from the ACR 463/EIM folder.
2. Place the **ProcessAlertsAndBillingProf.ifb** file on the server path: Siebsrvr/Admin

To configure Siebel CRM:

1. Make sure that the Siebel server is running.
2. Make sure that the Siebel Enterprise Integration Manager (EIM) component is online.
3. For creation of the EIM job:
    a. Go to **SiteMap, Administration - Server Management, Jobs**
    b. Click the **New** button and select *Enterprise Integration Manager* in the **Component/Job** tab.
    c. Complete the following parameters on the Job Parameters applet: **Configuration file = ProcessAlertsAndBillingProf.ifb**. Enter appropriate settings for **Error Flag, Trace Flag**, and **SQL Trace Flag**
4. Click the **Submit Job** button.
5. Make sure that the status changes to **Success** for the job that you have run.
6. After the EIM job is done, query the IF_ROW_STAT column of EIM_ALERT_CUT table to ensure that all the records have been loaded in the Siebel base tables correctly.
7. Check the Siebel CRM user interface (UI) to ensure that the data appears in corresponding views of the Credit Management screen.

    See the discussion of configuring EIM interfaces in *Configuring Siebel Business Applications* for more information.

For Oracle Data Integrator (ODI)-based collection flows, during the transfer of data from BRM to Siebel CRM, ODI creates some temporary tables in the Siebel database. Once the data is successfully written to the Siebel EIM table, these temporary tables are dropped. Therefore, the Siebel database administer must grant *create table* privilege so that ODI can create the temporary tables.

See *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator* for more information on using ODI.

Setting Up Split Cross Reference Tables

To maintain performance levels when looking up values in large cross reference tables, Oracle recommends splitting the cross reference tables into multiple tables, with one table for each cross-reference object. You split cross reference tables using the XREF Migration Utility and add the custom table names to the **AIAConfigurationProperties.xml** file.

To set up split cross-reference tables:

1. Split the collections cross-reference tables using the XREF Migration Utility as described at the following Oracle Technology Network web site:

   http://www.oracle.com/technetwork/middleware/foundation-pack/learnmore/aiaxref-524690.html

2. Open **AIAConfigurationProperties.xml** in a text editor.
3. Add a comma-separated list of the names of your custom cross reference tables.
4. Save and close the file.

Working with DVMs

Domain value maps (DVMs) are a standard feature of the Oracle service-oriented architecture (SOA) Suite that enables 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 them. They are stored in XML files and cached in memory at run time.

DVM types are seeded for the Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care flows. Administrators can extend the list of mapped values by adding more maps.

Table 10–1 lists the DVMs for the process integration for collections management:

Table 10–1 Collections Management DVMs

DVM	Description
COLLECTION_ACTIONNAME.xml	DVM mapping for action name
COLLECTION_PRIORITY.xml	DVM mapping for priority
COLLECTION_STATUS.xml	DVM mapping for status
COLLECTION_SUBSTATUS.xml	DVM mapping for sub-status
CURRENCY_CODE.xml	DVM mapping for currency code

> **Caution:** DVMs are stored in the Metadata Services (MDS) repository, which uses the database persistence, and are managed using tools provided by JDeveloper or Foundation Pack.

See the discussions of working with message transformations, 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

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 10–2 lists the collections management cross references:

Table 10–2 Collections Management Cross References

Cross Reference Table Name	Column	Description
CUSTOMERPARTY_ ACCOUNTID	BRM/ SIEBEL	Used for cross reference lookup for account ID.
CUSTOMERPARTY_ BILLPROFILEID	BRM/ SIEBEL	Used for cross reference lookup for billing profile ID.

Handling Errors

Handling BRM errors:

Error details are logged in the **CollectionsErrorLog.txt** file. This log file is available at <User_Home>/DISHome/CollectionsHome

Here <User_Home> is the UNIX user home path. For example, /slot/ems2880/oracle/DISHome/CollectionsHome

Handling Siebel CRM errors:

After every EIM load, check the value of the IF_ROW_STAT column in the EIM_ ALERT_CUT table. The status is imported for successful loading. If the status is not imported, errors occurred during data load from the Siebel EIM tables to the Siebel base tables. Perform the following actions for this type of error:

1. From the Jobs view in the Administration - Server Management screen, navigate to the Tasks View in the same screen.
2. Get the task number from the Task field.
3. Navigate to the Siebsrvr\Log directory in the server and locate the log file by querying with the same task number.
4. The log file contains details of every level of EIM processing and errors if any.
5. Run the EIM job with appropriate .ifb property values after modifying the values for failed records.

See the discussion of resolving import processing problems in *EIM Administration Guide* for more information about EIM Error handling.

For more information about the errors generated by BRM and Siebel CRM applications, see the documentation for that product.

See the discussions of using error notifications and using trace and error logs in *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* for information on how to set up error notifications using these values.

Configuring the Process Integration for Collections Management

Configure these properties in the **AIAConfigurationProperties.xml** file. The file is located in *<AIA_INSTANCE_HOME>***/AIAMetaData/config/**. Entries in the **AIAConfigurationProperties.xml** file are case-sensitive. See Table 10–3.

See "Building AIA Integration Flows" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information on reloading updates to **AIAConfigurationProperties.xml** file.

Table 10–3 AIAConfigurationProperties.xml - Property Values

Property Name	Value/Default Values	Description
FromMailAddress	user@oracle.com After installation, administrators or users can change this value.	This property is required to send a notification mail to user/admin.
ToMailAddress	user@oracle.com After installation, administrators or users can change this value	This property is required to send a notification mail to user/admin
MailServer	mail.oracle.com After installation, administrators or users can change this value.	This property is required to send a notification mail to user/admin.
ODI.USERNAME	odi.username After installation, administrators or users can change this value.	This property is read by SyncCollectionHeaderInfoBRMCommsReqImpl process to invoke ODI Web Service.
ODI.PASSWORD	odi.password	This property is read by SyncCollectionHeaderInfoBRMCommsReqImpl process to invoke ODI Web Service. The odi.password value for this property should be retained as is because at runtime an encrypted ODI password is read from **AIAInstallProperties.xml** file.
ODI.WORKREPNAME	odi.workrep.name After installation, administrators or users can change this value.	This property is read by SyncCollectionHeaderInfoBRMCommsReqImpl process to invoke ODI Web Service.
AIAHome	No default value. A value is entered during installation.	This property contains the absolute path to AIA Home.
BatchSize	Default value = 5000 After installation, administrators or users can change this value.	This property specifies the number of messages to dequeue in one batch. See "Changing the BatchSize Default Value" for information about changing the batch size default value.
Default.SystemID	SEBL_01	This property gives the system ID of Siebel CRM.
IF_ROW_BATCH_NUM_CreatedStart	Default value = 100. After installation, administrators or users can change this value.	This property indicates the starting value of the batch number (IF_ROW_BATCH_NUM) for the rows for new collection actions created in the Siebel EIM table.
IF_ROW_BATCH_NUM_CreatedEnd	Default value = 199 After installation, administrators or users can change this value.	This property indicates the end value of the batch number (IF_ROW_BATCH_NUM) for the rows for new collection actions created in the Siebel EIM table.
IF_ROW_BATCH_NUM_UpdateStart	Default value = 200 After installation, administrators or users can change this value.	This property indicates the starting value of the batch number (IF_ROW_BATCH_NUM) for the updated collection actions in the Siebel EIM table.
IF_ROW_BATCH_NUM_UpdateEnd	Default value = 299 After installation, administrators or users can change this value	This property indicates the ending value of the batch number (IF_ROW_BATCH_NUM) for the updated collection actions in the Siebel EIM table.

Table 10–3 (Cont.) AIAConfigurationProperties.xml - Property Values

Property Name	Value/Default Values	Description
IF_ROW_BATCH_NUM_Error	Default value = 50 After installation, administrators or users can change this value.	This property defines the value for the IF_ROW_BATCH_NUM for the unsuccessful collections actions during importing from the Siebel EIM table.
AL_SOURCE_CD	Customer	This property is used to update the AL_SOURCE_CD value in the Siebel EIM table.
AL_TYPE_CD	Credit	This property is used to update the AL_TYPE_CD value in the Siebel EIM table.

Changing the BatchSize Default Value

Changing the BatchSize default value has dependencies on the batch size that is defined in Siebel. Before you make any changes to this property, check Siebel documentation to find out the optimal batch size and accordingly make changes to this property.

For example, if the BatchSize property is changed to 5000 and the IF_ROW_BATCH_NUM_CreatedStart = 100 and IF_ROW_BATCH_NUM_CreatedEnd = 199, then ODI can simultaneously dequeue 5000*100 = 500000 records into Siebel EIM table. Siebel batch size for the EIM table load must match so that it can manage this data upload from ODI.

A

Mapping Siebel Billing Management UI Elements to BRM Customer Center

This appendix provides a mapping of fields on the Siebel customer relationship management (Siebel CRM) Billing Management UI to fields in Oracle Communications Billing and Revenue Management (BRM) Customer Center. This appendix can be used as a reference to explain the Agent Assisted Billing Care integration.

> **Note:** Siebel UI screenshots are for reference only and may differ from actual screens.

Billing Profile and Account Balance

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link.

Siebel Screen: Billing Profile Portal screen

Siebel View: Billing Invoice

UI Component: Billing Profile, as described in Table A–1.

Table A–1 Billing Profile Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Billing Profile Form	Account	Summary	Contact Information	Company or First & Last Name	Account/Customer name
--	Primary	NA	NA	NA	--
--	Profile Name	Payments	Billing Payment Method	Bill Unit	Account/Customer Billing Profile name
--	Bill Type	NA	NA	NA	Bill/Invoice type requested by customer (summary or detail)
--	Service Account	Summary	Contact Information	Company or First & Last Name	In cases where the billing account and service account are different. In such scenarios it results in a parent-child hierarchy in BRM. The service account is the nonpaying child account.

Table A–1 (Cont.) Billing Profile Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
--	Bill Media	Payments	Billing Payment Method - Payment Options	Delivery Method	Delivery method for invoice. For example, delivery of invoices by *email*, *paper*, and so on.
--	Payment Method	Payments	Billing Payment Method	Payment Method	How customers pay their bills. Payment methods include *credit card*, *invoice*, *debit card*, and so on.
--	Billing Profile Status	NA	NA	NA	Current status of the customer's billing profile (*active* or *inactive*).

UI Component: Balance Summary, as described in Table A–2.

Table A–2 Balance Summary Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Balance Summary Form	Balance	Balance	Balance Summary	Amount due for all bills	Total billed amount that has not been paid. This includes the balance impact of any unresolved dispute.
--	Due Now	Balance	Balance Summary	Due Now	Billed amount the customer currently owes. This is calculated as *Amount due for all bills* minus *Adjustments/Payments* that are not yet applied.
--	Pending Payments/Adjustments	Balance	Balance Summary	Adjustments/Payments not applied	Total of unallocated payments and unallocated account adjustments.
--	Currency	NA	NA	NA	--
--	Unresolved Disputes	Balance	Balance Summary	Unresolved Disputes	Total of disputed amounts that have been removed from the *Due Now* amount before the settlement of the dispute.
--	Total	Balance	Balance Summary	Total	Sum of the *Due Now* amount and the *Bill in Progress* (estimate) amount.
--	Unbilled Usage	Balance	Balance Summary	Bills in Progress	The *Bill in Progress* shows the current balance of the upcoming bill, including unbilled item charges, cycle forward arrears fees, and A/R actions on those charges and fees.

Bills

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Bills** tab

Siebel View: Billing Invoice

UI Component: Bills, as described in Table A–3.

Table A–3 Bills Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Invoice List Applet	Bill Number	Balance	Bills	Number	A unique number that identifies a specific bill. Each invoice contains a bill number. A bill is an object in the Oracle Communications BRM database that stores the balance impacts in the bill items of a customer's account during one billing cycle. Bills contain information about the customer's account, the account's billing cycle, and the amount billed.
--	Bill Period	Balance	Bills	Billing Cycle	The time period during which charges accumulate in an account before a bill is finalized. One billing cycle can contain one or more accounting cycles.
--	Amount Due	Balance	Bills	Balance	The original bill amount minus *Payments and A/R Actions*.
--	Due Date	Balance	Bills	Due Date	The date on which the bills payment is due.
--	Previous Balance	NA	NA	NA	--
--	Payments and A/R Actions	Balance	Bills	Payments and A/R Actions	The total sum of all payments made for a bill plus the A/R actions such as *adjustments* or *refunds*.
--	Bill Payment	Payments	Payments Received	Paid	The total payment made against a bill.

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Bills** tab, **Bill Number** link

Siebel View: Billed Usage Detail

UI Component: Bill Details, as described in Table A–4, Service Charges, as described in Table A–5, and Item Charges as described in Table A–6.

Table A–4 Bill Details Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Invoice Form Applet	Account	Summary	Contact Information	Company or First & Last Name	Account/Customer name
--	Profile Name	Payments	Billing Payment Method	Bill Unit	Account/Customer Billing profile name
--	Bill Number	Balance	Bill Details	Number	A unique number that identifies a specific bill. Each invoice contains a bill number. A bill is an object in the Oracle Communications BRM database that stores the balance impacts in the bill items of a customer's account during one billing cycle. Bills contain information about the customer's account, the account's billing cycle, and the amount billed.
--	Bill Period	Balance	Bill Details	Billing Cycle	The time period during which charges accumulate in an account before a bill is finalized. One billing cycle can contain one or more accounting cycles.
--	Amount Due	Balance	Bill Details	Balance	The original bill amount minus the *Payments and A/R Actions*.
--	Due Date	Balance	Bill Details	Due Date	The date on which the bills payment is due.
--	Previous Balance	NA	NA	NA	--
--	Payments and A/R Actions	Balance	Bill Details	Payments and A/R Actions	The total sum of all payments made for the bill plus the A/R actions such as *adjustments* or *refunds*.
--	Bill Payment	Balance	Bill Details	Total Payments	The total payment made against a bill.

Table A–5 Service Charges Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Invoice Service Charge List	Description	Balance	Bill Details, Item Charges	Description	Name of the service
--	Charge	Balance	Bill Details, Item Charges	Charge	Holds charges of all items under the service.
--	Discount	Balance	Bill Details, Item Charges	Discount	Holds the total of all discounts given under the service or item.
--	Net Amount	Balance	Bill Details, Item Charges	Net	The *Net Amount* is the amount after discounts, payments, and A/R actions have been applied to the service.

Table A–6 shows the mappings for Item Charges.

Table A–6 Item Charges Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Invoice Item Charge List	Description	Balance	Bill Details, Item Charges	Description	Item is an entity that represents a group of charges. For example, a *Cycle Forward* charge for the service VoIP instance.
--	Charge	Balance	Bill Details, Item Charges	Charge	Charge indicates the total amount for the item.
--	Discount	Balance	Bill Details, Item Charges	Discount	The Discount column shows any discount that is applicable to the item.
--	Net Amount	Balance	Bill Details, Item Charges	Net Amount	The *Net Amount* is the amount after discounts, payments, and A/R actions have been applied to the item.

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Bills** tab, **Bill, Item Charges, Net Amount** link

Siebel View: CDR Details

UI Component: Event Details as described in Table A–7.

Table A–7 Event Details Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
CDR Details List	Date	Balance	Bill Details, Item Charges, Description	Date & Time	Date and time when the call was made. The time zone of the call is in the time zone in which the call was originally made.
--	Charge	Balance	Bill Details, Item Charges, Description	Charge	*Charge* indicates the total amount for the call (CDR).
--	Discount	Balance	Bill Details, Item Charges, Description	Discount	The *Discount* column shows any discount that is applicable.
--	Net Amount	Balance	Bill Details, Item Charges, Description	Net	Actual amount due after any discounts are applied to the charge.
--	Duration	Balance	Bill Details, Item Charges, Description	Quantity	Total time of the call.
--	Number Called	Balance	Bill Details, Item Charges, Description	Called No	Telephone number to which the call was made.
--	Non-Currency	Balance	Bill Details, Item Charges, Description	Non-Currency	If the event is of type *nonmonetary*, the column is checked.

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Bills** tab, **Bill Number** link, **Payments** tab

Siebel View: Billed Usage Payments

UI Component: Bill Payments as described in Table A–8.

Table A–8 Bill Payments Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Billed Usage Payments List	Payment Number	Balance	Bill Details, Payment Details	Payment Number	The payment item number to identify a payment.
--	Payment Date	Balance	Bill Details, Payment Details	Posted	The date on which the payment was posted.
--	Payment Method	Balance	Bill Details, Payment Details	Payment Type	The *Payment Method* identifies how customers paid their bill; for example, by credit card or direct deposit.
--	Payment Amount	Balance	Bill Details, Payment Details	Amount	The total amount that was paid by the subscriber as part of the payment.
--	Confirmation Number	Balance	Bill Details, Payment Details	Payment Number	The payment item number to identify a payment.
--	Allocated	Balance	Bill Details, Payment Details	Allocated	The payment amount that has been allocated to the bill.
--	Unallocated	Balance	Bill Details, Payment Details	Unallocated	The total amount from the payment made that is not yet applied.
--	Reversed	Balance	Bill Details, Payment Details	Reversed	If a particular payment has been reversed, then this column displays the value *Y*.

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Bills** tab, **Bill Number** link, **A/R Items** tab

Siebel View: A/R items Details

UI Component: Bills A/R items as described in Table A–9.

Table A–9 Bills A/R Items Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
A/R Items List	Type	Balance	Bill Details, A/R Items	Type	The type of A/R actions, such as *Adjustment*, *Dispute*, *Refund*, *Write-off*, and so on.
--	Date	Balance	Bill Details, A/R Items	Date	The date on which the adjustment was made.
--	Resource Name	Balance	Bill Details, A/R Items	Resource	This column indicates to which resource the adjustment was made. For example, *currency* resource or *noncurrency* resource such as free seconds.
--	Amount	Balance	Bill Details, A/R Items	Amount	The adjustment amount that was made against a resource.

Balance Group

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Balance Group** tab

Siebel View: Balance Group

UI Component: Balance Group Details (Balance Group, Balance, Balance Details and Services) as described in Table A–10.

Table A–10 Balance Group Details Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Balance Group	Name	Balance	All Credit Limit, Currency Credit Limit	Balance Group	The name of the balance group. For example, *Account Level Balance Group*.
Balance	Balance	--	All Credit Limit, Currency Credit Limit	Outstanding	Total balance under the balance group.
--	Unit of Measure	NA	NA	NA	In case of monetary resource, this column in Siebel CRM displays *Currency* and for nonmonetary resource this column is blank.
--	Ceiling Credit Limit	--	All Credit Limit, Currency Credit Limit	Amount or Unlimited	**1.** If there is a value under the Account in BRM, this indicates the credit limit of the balance group. **2.** If the *Unlimited* column is selected, this indicates there is no credit limit for the balance group.
--	Floor Credit Limit	NA	NA	NA	--
--	Threshold Credit Limit	NA	NA	NA	--
Balance Details (for monetary resource)	Available	Plan	Product Detail	Outstanding	Total balance under the balance group.
--	Valid From	Plan	Product Detail	Purchase Start Date	Purchase start or valid from date.
--	Valid To	Plan	Product Detail	Purchase End Date	Purchase end or valid to date.
--	No End	Plan	Product Detail	Check box	This is used when the product has unlimited validity.
Balance Details for non-monetary resource)	Available	Non-Currency	Non-Currency Details	Available	Total balance under the balance group.
--	Valid From	Non-Currency	Non-Currency Details	Valid From	Resource start or valid from date
--	Valid To	Non-Currency	Non-Currency Details	Valid To	Resource end or valid to date.

Table A–10 (Cont.) Balance Group Details Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
--	No End	Non-Currency	Non-Currency Details	Check box	This is used when the resource has unlimited validity.
Balance Group Services	BRM Service ID	Balance	All Credit Limit, Currency Credit Limit	Balance Group	If the product is part of a service bundle, then this column displays the *BRM Service ID* of the product under the balance group.
--	Service Type	Plans	Plans	Service	Billing service type of the product.
--	Product Name	Plans	Plans	Product/Discount	Product name.
--	Effective Date	Plans	Plans	Purchased	Product purchased date.
--	Status	Plans	Plans	Status	Current status of the product (*active* or *canceled*).

Unbilled Usage

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Unbilled** tab

Siebel View: Unbilled Usage

UI Component: Bill Details, Service Charges, Item Charges as described in Table A–11.

Table A–11 Bill Details, Service Charges, and Item Charges Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Unbilled Usage	Description	Balance	Bills in Progress, Item Charges	Description	Name of the service.
	Charge	Balance	Bills in Progress, Item Charges	Charge	Holds unbilled charges of all items under the service.
--	Discount	Balance	Bills in Progress, Item Charges	Discount	Holds the total of all discounts given under the service or item.
--	Net Amount	Balance	Bills in Progress, Item Charges	Net	The *Net Amount* is the amount after discounts, payments, and A/R actions have been applied to the service.
Unbilled Item Charges	Description	Balance	Bills in Progress, Item Charges	Description	Item is an entity that represents a group of charges. For example, a *Cycle Forward* charge for the service VoIP instance.

Table A–11 (Cont.) Bill Details, Service Charges, and Item Charges Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
--	Charge	Balance	Bills in Progress, Item Charges	Charge	*Charge* indicates the total amount for the item.
--	Discount	Balance	Bills in Progress, Item Charges	Discount	The *Discount* column shows any discount that is applicable.
--	Net Amount	Balance	Bills in Progress, Item Charges	Net Amount	The *Net Amount* is the amount after discounts, payments, and A/R actions have been applied to the item.

Payments

Navigate to the **Account Summary, Billing Profile** applet, **Billing Profile Name** link, **Payments** tab

Siebel View: Billing Profile Payment

UI Component: Payments as described in Table A–12.

Table A–12 Payments Mapping

Siebel Applet	Siebel Field	BRM Tab	BRM Form	BRM Field	Comments
Billing Profile Payments List	Payment Number	Payments	Payments Received	Number	The payment item number to identify a payment.
--	Payment Date	Payments	Payments Received	Date	The date on which the payment was posted.
--	Payment Method	Payments	Payments Received	Payment Method	The *Payment Method* identifies how customers paid their bill; for example, by credit card or direct deposit.
--	Payment Amount	Payments	Payments Received	Paid	The total amount that was paid by the subscriber as part of the payment.
--	Comments	Payments	Payments Received	NA	--
--	Confirmation Number	Payments	Payments Received	Number	The payment item number to identify a payment.
--	Allocated	Payments	Payments Received	Allocated	The payment amount that has been allocated to the bill.
--	Unallocated	Payments	Payments Received	Unallocated	The total amount from the payment made that is not yet applied.
--	Reversed	Payments	Payments Received	Reversed	Any payments that were reversed.

B

Composite Application Validation System Changes

This appendix discusses how the Composite Application Validation System (CAVS) has changed from the Oracle Application Integration Architecture (Oracle AIA) Communications 11.1 release to the Oracle AIA Communications 11.2 release and provides details on Requestor application business connector services (ABCSs) and Provider ABCSs.

Configuration Properties for CAVS Enablement in 11.1

In the 11.1 Oracle AIA CAVS implementation, every service has a number of configuration properties.

See the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack* for more information about these configuration properties.

For the RouteToCAVS property, the ready-to-use value is False. Oracle AIA provides a user interface (UI), which allows the user to toggle this property value between True and False for each service listed.

To navigate to this UI:

1. Log in to the AIA Console (http://<host>:<port>/AIA).
2. Go to **Setup**.
3. Select the **AIA Configuration** tab.

Configuration Properties for CAVS Enablement in 11.2

CAVS enablement has been reorganized. As a result, the UI can no longer be used to toggle the value of the `RouteToCAVS` property for the Communications Order to Cash services, which are part of the 11.2 release.

The following instructions describe how to modify the configuration properties for Requestor ABCS and Provider ABCS to enable CAVS.

Note: Any change in the System Configuration screen does not enable CAVS for a service. You must make changes manually in the Oracle AIA configuration file to make the service CAVS enabled.

Requestor ABCS

For CAVS enablement of Requestor ABCS, a single configuration property is maintained.

For example, `EBSOverride.CommunicationsSalesOrderEBSV2.ProcessSalesOrderFulfillment.Address`

In order to enable CAVS, you must manually edit the **AIAConfigurationProperties.xml** file located in: *$AIA_HOME*/**aia_instances/$INSTANCE_NAME/AIAMetaData/config**.

> **Note:** Entries in the **AIAConfigurationProperties.xml** file are case sensitive.

To enable CAVS for Requestor ABCS:

1. Open the **AIAConfigurationProperties.xml** file.
2. Set the Address property to the CAVS URI for each service that you want to be CAVS enabled.

   For example, http://<soa_server>:<soa_port>/AIAValidationSystemServlet/asyncrequestrecipient
3. Save and close the file.
4. To make your changes effective login to the AIA Console (http://<host>:<port>/AIA).
5. Go to **Setup** and select the **AIA Configuration** tab.
6. Click **Reload** to reload the configuration file.

Provider ABCS

For CAVS enablement of a Provider ABCS, two configuration properties are maintained. For example:

- "Routing.SWI_spcOrder_spcUpsert.RouteToCAVS"
- "Routing.SWI_spcOrder_spcUpsert.SEBL_01.EndpointURI"

In order to enable CAVS, you must manually edit the **AIAConfigurationProperties.xml** file, which is located in: *$AIA_HOME*/**aia_instances/$INSTANCE_NAME/AIAMetaData/config**.

> **Note:** Entries in the **AIAConfigurationProperties.xml** file are case sensitive.

To enable CAVS for Provider ABCS:

1. Open the **AIAConfigurationProperties.xml** file.
2. Set the `RouteToCAVS` property value to *True* and set the EndpointURI property value to the CAVS URL for each service that you want to be CAVS enabled.
3. Save and close the file after you have set this property for all desired Provider ABCSs.

4. To make your changes effective login to the AIA Console (http://<host>:<port>/AIA).
5. Go to **Setup** and select the **AIA Configuration** tab.
6. Click **Reload** to reload the configuration file.

C

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 EBSs:

1. Go to JDevloper and create a new composite for the EBS with an Oracle Mediator service. Use the EBS WSDL provided by Fusion Middleware Foundation Pack.
2. Create routing rules in Oracle Mediator to route to appropriate Provider connectors.
3. Save your changes.
4. Open the **AIAConfigurationProperties.xml** file, which is located in: *$AIA_HOME*/**aia_instances/$INSTANCE_NAME/AIAMetaData/config**.

   Note: Entries in the **AIAConfigurationProperties.xml** file are case sensitive.

5. To invoke new EBS connectors you need to replace the Provider connector's name and address with the EBS name and address.

   This action tells the Requestor to invoke EBS instead of the Provider application business connector service (ABCS).

6. Save and close the file.
7. To make your changes effective. Login to the AIA Console (*http://<host>:<port>/AIA*).
8. Go to **Setup**, and select the **AIA Configuration** tab.

9. Click **Reload** to reload the configuration file.