BEA WebLogic Adapter for Siebel®

User Guide

Version 8.1.0
Document Revised: July 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Introducing the BEA WebLogic Adapter for Siebel

2. Generating Schemas for Siebel Integration Objects

3. Defining Application Views for Siebel

A. Sample Files

B. Creating Siebel Workflows

Index

About This Document

This document describes how to install, configure, and use the BEA WebLogic Adapter for Siebel. This document is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for Siebel,"describes the adapter, how it relates to both Siebel business objects and WebLogic Integration.
- Chapter 2, "Generating Schemas for Siebel Integration Objects," describes how to generate schemas for your Siebel business objects using the BEA Application Explorer.
- Chapter 3, "Defining Application Views for Siebel," describes application views and how to use them to configure events and services.
- Appendix A, "Sample Files," presents sample request and response schema and XML for various Siebel applications.
- Appendix B, "Creating Siebel Workflows," provides examples of Siebel workflows for services and events using each of the three transport types.

Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.

- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.
- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

Additional Information

To learn more about the software components associated with the adapter, see the following BEA documents:

- *BEA WebLogic Adapter for Siebel Release Notes*

  http://edocs.bea.com/wladapters/siebel/docs81/pdf/relnotes.pdf
- *BEA WebLogic Adapter for Siebel Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/siebel/docs81/pdf/install.pdf
- *BEA Application Explorer Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/bae/docs81/pdf/install.pdf
- *Introduction to the BEA WebLogic Adapters for WebLogic Integration*

  http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf
- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  http://dev2dev.bea.com/products/product.jsp?highlight=wla
- Application Integration documentation

  http://edocs.bea.com/wli/docs81/aiover/index.html

  http://edocs.bea.com/wli/docs81/aiuser/index.html
- BEA WebLogic Integration documentation

  http://edocs.bea.com/wli/docs81/index.html

- BEA WebLogic Platform documentation

  http://edocs.bea.com/platform/docs81/index.html

You may also find it helpful to refer to the following Siebel documentation:

- For Siebel eBusiness Bookshelf Version 6.3. or higher, see these topics:
  - Overview: Siebel eBusiness Application Integration Volume I
  - Integration Platform Technologies: Siebel eBusiness Volume II
  - Transports and Interfaces: Siebel eBusiness Application Volume III
  - Business Processes and Rules: Siebel eBusiness Application Integration Volume IV
  - Tools Guide
  - Server Administration Guide
  - Workflow Administration Guide
- For Siebel eBusiness Bookshelf Version 6.0.1, see these topics:
  - Siebel eBusiness Application Integration: Siebel EAI Architecture, Using Siebel Business Services, Using Siebel EAI Adapters & Transports, Creating Business Services in Siebel
  - Siebel Object Interfaces Reference
  - Siebel Tools Guide
  - Server Administration Guide
  - Siebel Workflow Guide

How to Use This Document

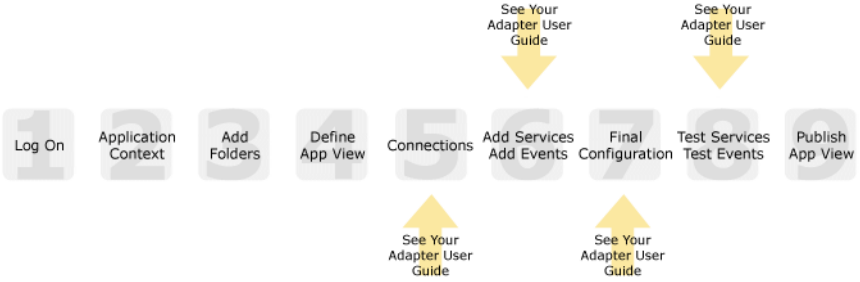
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Siebel system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*



Contact Us!

Your feedback on the BEA WebLogic Adapter for Siebel documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for Siebel documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for Siebel and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for Siebel, or if you have problems using the BEA WebLogic Adapter for Siebel, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER 1

Introducing the BEA WebLogic Adapter for Siebel

This section introduces the BEA WebLogic Adapter for Siebel and describes how the adapter enables integration with Siebel business objects and WebLogic Integration.

It includes the following topics:

About Adapters and BEA WebLogic Integration

The BEA application integration solution uses adapters and application views to help you integrate applications in your enterprise. At its most fundamental level, an *adapter* is software that connects an enterprise information system (EIS) and an integration server. This bi-directional connection consists of *services*—interactions that originate in the adapter (and may require a response from the EIS)—and *events*, interactions that originate in the EIS.

Most EIS systems make selected information and functions available to other applications by way of specialized integration APIs. An adapter connects to the EIS through its integration API, or through database or system calls, and exposes the available functions from the EIS. However, rather than exposing the intricacies of APIs to users, WebLogic Integration incorporates *applications views*—business-oriented interfaces that provide a layer of abstraction between an adapter and the EIS capabilities the adapter exposes.

Figure 1-1 Application View in an Integration Solution

Application views contain definitions for the services and events used by business processes to communicate with an EIS. They also contain connection information and XML schema that define inputs and outputs for services and events. After an adapter is deployed, you can use its Web-based interface to define as many applications views as you need, and other WebLogic Integration components and applications can use that adapter to access data on the EIS.

To learn more about the role of adapters in application integration architecture, see "Key Components of Integration Solutions" on page 1-3.

To learn more about adapters in general, see the *Introduction to the BEA WebLogic Adapters* at the following URL:

`http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf`

Key Components of Integration Solutions

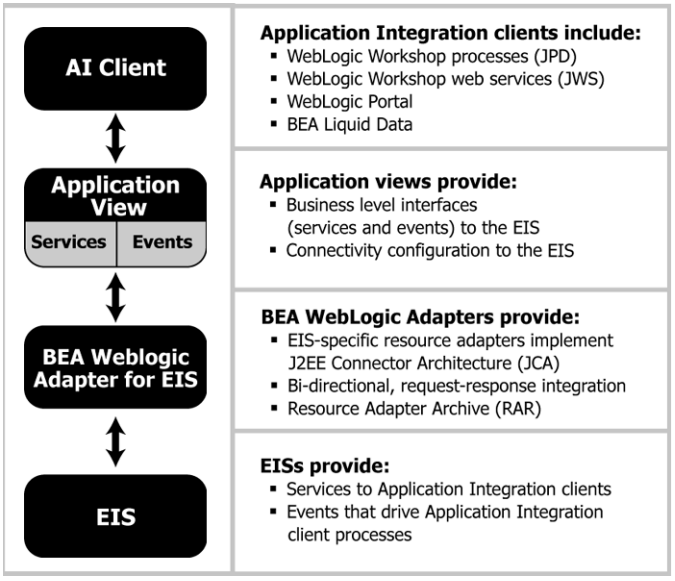
This section describes some of the key concepts you need to be familiar with before you work with an adapter.

- Basic WebLogic Integration Architecture
- Enterprise Information Systems
- Resource Adapters
- Application Views
- Service Clients and Event Consumers
- EIS Metadata, Schemas, and Repositories

Basic WebLogic Integration Architecture

Adapters are used in conjunction with the Application Integration component of BEA WebLogic Integration. This component provides a systematic, standards-based architecture for hosting business-oriented interfaces to enterprise applications.

Figure 1-2 Adapters in the Application Integration Architecture



For general information about Application Integration, see the following documents:

- *Introducing Application Integration* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiover/index.html`

- *Using The Application Integration Design Console* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Enterprise Information Systems

An *enterprise information system* (EIS) is software that provides the information infrastructure for an enterprise. An EIS offers a set of services to its clients, which are made available to clients via local and/or remote interfaces. An integration solution involves integration with one or more EISs.

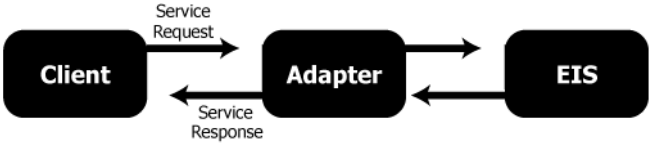
Resource Adapters

A *resource adapter* (or simply *adapter*) is a BEA software component that acts as a connector between an EIS and a J2EE application server (such as BEA WebLogic Server). Each adapter provides bi-directional, request-response integration with a specific application or technology.

Adapters handle two general types of operations:

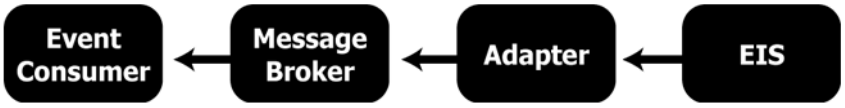
- *Services* are request / response communications with the EIS. Client applications submit service requests to the EIS via the adapter, and the adapter returns the For example, a business process might invoke a SAP BAPI or execute a SELECT statement on a database. EIS response back to the client. Responses are either synchronous or asynchronous.

Figure 1-3 Service Invocations



- *Events* are asynchronous, one-way messages received from an EIS. For example, the adapter can receive an IDOC from a SAP system or a message from an MQSeries system. The adapter routes the EIS message to the appropriate software component via the WebLogic Integration Message Broker and the Application Integration JMS infrastructure.

Figure 1-4 Event Notifications



In effect, a service is a *request for some work to be done* and an event is a *notification that some work has been done*.

For more information about the specific services and events supported by Adapter for Siebel, see "About the BEA WebLogic Adapter for Siebel" on page 1-9.

To learn more about the WebLogic Integration Message Broker and the Application Integration JMS infrastructure, see *Introducing Application Integration* at the following URL:

`http://edocs.bea.com/wli/docs81/aiover/index.html`

Application Views

An *application view* is a business oriented interface to objects and operations within an EIS. Application views include the information needed to communicate with the EIS as well as configurations for services and events.

To learn more about using application views in business processes, see the WebLogic Workshop documentation at the following URL:

`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`

You typically define an application view for a specific business process. Therefore, you might have multiple application views defined for a single adapter, each designed to meet a specific requirement.

An application view defines:

- **Communication with the EIS**, including connection settings, login credentials, and so on.
- **Service invocations**, including the information that the EIS requires for the request, as well as any service request and response schemas associated with the service.
- **Event notifications**, including the information that the EIS publishes and the event schemas for inbound messages.

Use one of the following resources to create application views:

- For detailed information about application views, see "Understanding Application Views" in *Introducing Application Integration* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiover/2intfra.html`

- For detailed information about writing custom code, see "Using Application Views by Writing Custom Code" in *Using The Application Integration Design Console* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiuser/4usrcust.html`

An application view for Adapter for Siebel provides these features:

- Standards-based data representation. All events, requests, and responses are represented as standards-based XML.
- Abstraction from the details of the EIS. Application views offer a level of abstraction from the details of the underlying EIS, freeing the developers to concentrate on the business processes and data and not on the configuration and details of that system.

To learn more about application views, see Chapter 3, "Defining Application Views for Siebel."

Service Clients and Event Consumers

In an integration solution, there are clients that invoke services and consumers for event notifications.

Service Clients

A variety of clients can invoke services on an EIS via an application view. They include BEA WebLogic Workshop business processes, web services, and portals; queries and BEA Liquid Data; and custom Java applications.

For more information, see the following topics in the BEA WebLogic Workshop Help System:

- "Building Integration Applications"
- "Building Web Services"
- "Building Portal Applications"

at the following URL:

`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`

In addition, see "Using Applications With Business Processes" in *Using The Application Integration Design Console* at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/3usruse.html

Event Consumers

Adapters deliver events using the WebLogic Integration Message Broker, which provides business processes with a channels-based publish and subscribe communication mechanism. Consumers can include BEA WebLogic Workshop business processes, web services, and portals, as well as custom Java applications.

For more information, see the following topics in the BEA WebLogic Workshop Help System:

- "Message Broker Subscription Control" in "Message Broker Controls"

  at the following URL:

  http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlsBrokerSubscribe.html

- "Overview: Application Integration"

  at the following URL:

  http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/contolsAppViewOview.html

In addition, see "Receiving Events" in "Using Applications With Business Processes" in *Using The Application Integration Design Console* at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/3usruse.html

EIS Metadata, Schemas, and Repositories

Each EIS uses its own interface to handle service requests and event notifications. Some interfaces are API-based, while others use database or system calls. For example, SAP provides a BAPI interface that defines the parameters and syntax for BAPI requests and responses. For each EIS, the EIS interface defines the *metadata* that applications can use to integrate with the EIS. The EIS publishes data and expects requests in the format dictated by its interface rules and metadata.

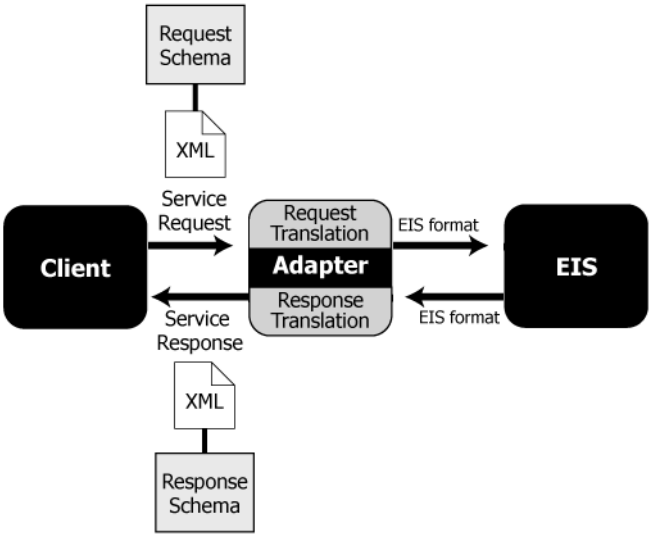
Schemas

At run-time, the EIS and the adapter exchange service requests, service responses, and events via XML documents. The adapter handles the data translation between XML documents and the EIS format, using *schemas* that map the data between XML and the EIS format: Depending on the

adapter, you either create the schema using the BEA Application Explorer, or have it automatically generated by the adapter when you add services and events to an application view.

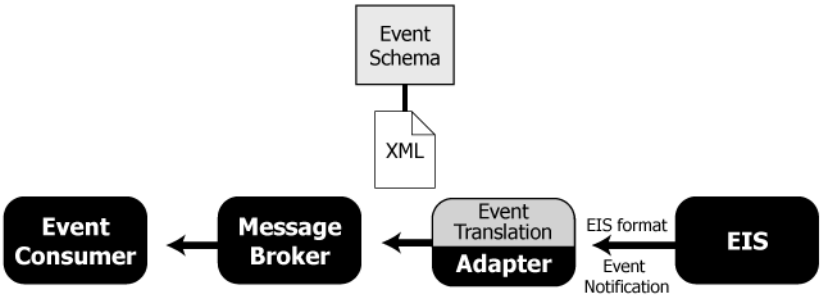
- For service requests, the request arrives at the adapter in the form of an XML document. The adapter uses the *request schema* associated with the service to translate the request to the format that the EIS expects. Similarly, when the adapter receives the response back from the EIS, it uses the *response schema* associated with the service to translate the response to an XML document that the requesting application handles.

Figure 1-5 Adapter Translation for Service Invocations



- For event notifications, the message arrives at the adapter in the format that the EIS uses to publish the event. The adapter uses the *event schema* associated with the event to translate the response to an XML document that the subscribed application handles.

Figure 1-6 Adapter Translation for Event Notifications



To learn more about schemas, see Chapter 2, "Generating Schemas for Siebel Integration Objects."

Repositories

Once you have created the necessary schemas, you save them in a file-based *repository*, along with a manifest file that associates the schemas with events and services. When you configure application views in the Application View Console, you specify the location of the repository so that the application view can find the schemas as needed. For more information, see Chapter 3, "Defining Application Views for Siebel."

About the BEA WebLogic Adapter for Siebel

The BEA WebLogic Adapter for Siebel connects to your Siebel system so that you can easily use your Siebel data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your Siebel system.

This section includes the following topics:

- Supported Siebel Operations for Application Integration
- Supported Services
- Supported Events
- Benefits of the Adapter for Siebel

Supported Siebel Operations for Application Integration

The Adapter for Siebel supports synchronous and asynchronous, bi-directional message interactions for Siebel Business Services, Business Components, and Integration Objects using these Siebel Transports: MQSeries, File, and HTTP.

It provides integration with the following Siebel operations:

- Access to Siebel integration objects using Siebel XML to handle both services and events
- Direct invocation of Siebel Business Services, workflows, and business components using the Siebel Java Data Bean or the Siebel COM Data Interface

Supported Services

The Adapter for Siebel supports three types of services, one for each type of Siebel Transport: MQSeries, File, and HTTP. In each case, the adapter sends a Siebel XML file to Siebel, which uses a Siebel Workflow within the Siebel system to cause a Siebel business event. The Siebel system must have a Siebel Workflow in order to act on the Siebel XML files it receives.

These are the services supported by Adapter for Siebel:

- MQ service, which sends a Siebel XML file to an IBM MQSeries or WebSphere MQ queue.
- File service, which sends a Siebel XML file to a specific directory on disk.
- HTTP service, which sends a Siebel XML file through HTTP.

Supported Events

The Adapter for Siebel supports three types of events, one for each type of Siebel Transports: MQSeries, File, and HTTP. In each case, the adapter picks up a Siebel XML file and passes it to an event variable within a business process. The Siebel system must have a Siebel Workflow in order to send Siebel XML files when a Siebel event occurs.

These are the events supported by Adapter for Siebel.

- MQ event, in which the adapter picks up a Siebel XML file from a specific IBM MQSeries or WebSphere MQ queue.
- File event, in which the adapter picks up a Siebel XML file from a specific directory on disk.

- HTTP event, in which the adapter picks up a Siebel XML file through HTTP.

Benefits of the Adapter for Siebel

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your business processes and enterprise applications with your Siebel system. The Adapter for Siebel provides these benefits:

- Integration can be achieved without custom coding.
- Business processes can be started by events generated by Siebel.
- Business processes can request and receive data from your Siebel system using services.
- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`
- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

  `http://e-docs.bea.com/wls/docs81/cluster/index.html`
- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

  `http://edocs.bea.com/wli/docs81/deploy/index.html`
- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your Siebel system. For more information about security, see the following URL:

  `http://edocs.bea.com/wls/docs81/secintro/index.html`

Getting Started With the Adapter for Siebel

This section gives an overview of how to get started using the BEA WebLogic Adapter for Siebel within the context of an application integration solution. Integration with Siebel involves the following tasks:

- Step 1: Design the Application Integration Solution

- Step 2: Determine the Required Siebel Business Workflows
- Step 3: Generate Schemas for Siebel Integration Objects
- Step 4: Define Application Views and Configure Services and Events
- Step 5: Integrate with Other BEA Software Components
- Step 6: Deploy the Solution to the Production Environment

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or business processes designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as Siebel systems and other EISs.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for Siebel. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including Siebel specialists). Note that an application integration solution can be part of a larger integration solution.

To learn more about designing an application integration solution, see *Designing WebLogic Integration Solutions* at the following URL:

`http://edocs.bea.com/wli/docs81/design/index.html`

Step 2: Determine the Required Siebel Business Workflows

Within the larger context of an application integration project, you must determine which specific Siebel integration objects and workflows are required to handle Siebel XML for services and events to support the business processes in the application integration solution. Or, if you are

invoking Siebel business services or business components directly, rather than through a workflow, you must determine the tasks you need to complete.

Factors to consider include (but are not limited to):

- Type of Siebel integration objects, workflows, and transport used to access the Siebel system.
- Siebel transactions involved in business processes
- Logins required to access Siebel transports and perform the required operations
- Whether operations are, from the adapter point of view:
  - services, which notify the Siebel system, via a Siebel XML document, with a request for action, and, in addition, whether such services should be processed synchronously or asynchronously
  - events, which are notifications from the Siebel system that trigger business processes.

This step involves the expertise of Siebel specialists, including analysts and administrators.

Step 3: Generate Schemas for Siebel Integration Objects

After identifying the Siebel integration objects and workflows required for the application integration solution, you must generate the XML schemas that will be used to exchange data with one or more Siebel systems:

- Services require two XML schemas: one for the Siebel request and another for the Siebel response.
- Events require a single XML schema to handle the data sent by the Siebel system.

You use the BEA Application Explorer tool to generate schemas for Siebel operations. To learn more about schemas, see Chapter 2, "Generating Schemas for Siebel Integration Objects."

Step 4: Define Application Views and Configure Services and Events

After you create the schemas for your Siebel services or events, you create an application view that provides an XML-based interface between WebLogic Server and a particular Siebel system within your enterprise. If you are accessing multiple Siebel systems, you define a separate application view for each Siebel system you want to access. To provide different levels of

security access (such as “guest” and “administrator”), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that you created in “Step 3: Generate Schemas for Siebel Integration Objects” on page 1-13. To learn more about generating schemas, see Chapter 2, “Generating Schemas for Siebel Integration Objects.”

To learn more about defining application views, see Chapter 3, “Defining Application Views for Siebel” in conjunction with*Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 5: Integrate with Other BEA Software Components

Once you have configured and published one or more application views for Siebel integration, you can integrate these application views into other BEA software components, such as business processes or portals built in the BEA WebLogic Workshop environment.

For more information, see *Using the Application Integration Design Console* at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 6: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for Siebel as described in *BEA WebLogic Adapter for Siebel Installation and Configuration Guide*
- Deploy your application views and schemas for Siebel integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

To learn more about designing an application integration solution, see *Designing WebLogic Integration Solutions* at the following URL:

http://edocs.bea.com/wli/docs81/design/index.html

CHAPTER 2

Generating Schemas for Siebel Integration Objects

The BEA WebLogic Adapter for Siebel uses XML documents to communicate with your Siebel system's integration objects for both services and events. The format of these XML documents is determined by XDR schemas you generate using the Siebel Tools Schema Wizard. After you generate the XDR schemas, you can use them as input to the BEA Application Explorer to generate the schemas that the adapter will use.

This section explains how to use the BEA Application Explorer to generate schemas. It contains the following topics:

Before You Begin

Before you begin to generate schema for the BEA WebLogic Adapter for Siebel, you must:

- Download and install the BEA Application Explorer software. To learn more, see the *BEA Application Explorer Installation and Configuration Guide* at the following URL:

  `http://edocs.bea.com/wladapters/bae/docs81/pdf/install.pdf`

- Obtain the information necessary to connect to your Siebel system. See Table 2-1 for a description of the parameters the BEA Application Explorer requires to connect to Siebel. Contact your Siebel administrator for this information.

- Use the Siebel Tools Schema Wizard to generate an EAI XML XDR schema with a Siebel Message envelope. The XDR file must be located in a directory where the BEA Application Explorer can access it. See "Sample XDR for Sample Account" on page A-75 for an example of an XDR file. See "Generating a Siebel XDR Schema" on page 2-2 for instructions on how to create an XDR schema for Siebel 6.3 or later.

Generating a Siebel XDR Schema

The BEA Application Explorer uses a Siebel XDR file as input for creating the XML schema that the BEA WebLogic Adapter for Siebel uses. For Siebel version 6.3 or later, you create the XDR file using the Siebel Tools Schema Wizard.

For Siebel releases prior to Siebel 6.3, the Siebel Tools Schema Wizard creates DTD schemas instead of the XDR schema the BEA Application Explorer requires. Therefore, you must transform the DTD schema—either manually or using other tools—into XDR schema before you can use them to create the XML schema for the BEA WebLogic Adapter for Siebel. The XDR file must contain the SiebelMessage tag reference. For an example, see "Sample XDR for Sample Account" on page A-75.

To generate a Siebel XDR schema:

1. Log on to Siebel Tools.

   The Siebel Tools Window appears, displaying Integration Objects in the right pane.

2. Select the integration object, Sample Account, in the right pane.
3. Click the Generate Schema button.

   The Generate XML Schema wizard appears.

4. Select EAI XML XDR Generator from the Business Service list.
5. Select Siebel Message envelope from the envelope type list.

   Note: Your Siebel version may not require a message envelope.

6. Enter a file name for the XDR schema and a directory where it can be accessed by the BEA Application Explorer. For example, enter:

   ```
   C:\BEA\BEASCHEMAS\Sample_Account.xdr
   ```

7. Click Finish.

   You can now use the BEA Application Explorer to generate BEA schemas. To learn more about generating schemas, see "Starting the BEA Application Explorer" on page 2-5.

About the BEA Application Explorer

The BEA Application Explorer uses intelligence about Siebel combined with metadata provided by Siebel XDR to generate the schemas required to build application view services and events.

This section contains the following topics:

- About the Process for Defining Schemas
- Types of Schemas Generated by the BEA Application Explorer

About the Process for Defining Schemas

The process for defining XML schemas includes the following steps:

1. Starting the BEA Application Explorer.
2. Setting the Session Path.

   The BEA Application Explorer uses this path to create the directory for the schemas.
3. Creating a New Connection or Using an Existing Connection.
4. Creating Schemas for Services and Creating Schemas for Events.

Types of Schemas Generated by the BEA Application Explorer

Each service or event the BEA WebLogic Adapter for Siebel uses must be defined by a schema. The BEA Application Explorer generates XML schemas for:

- Service Requests
- Service Responses
- Events

Service Requests

Service requests are requests for action that your application makes to your Siebel system. Requests are defined by request schema. As part of the definition, the request schema defines the input parameters required by the Siebel system. The Siebel system responds to the request with a service response.

Service Responses

Service responses are the way the Siebel system responds to a service request. A service response schema defines this service response. Service requests always have corresponding responses.

Events

Events are generated by the Siebel system as a result of activity on that system. You can use these events to trigger an action in your application. For example, the Siebel system may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event. Events are defined by event schema.

Starting the BEA Application Explorer

You use the BEA Application Explorer to generate service request schemas, service response schemas, and event schemas. The schemas you create are published in the WebLogic Integration repository. Before you can create these schemas, however, you must create Siebel XDR schemas for selected Siebel integration objects. If you have not already done so, see your Siebel eBusiness Bookshelf for instructions on using the Siebel Tools Schema Wizard to generate an EAI XML XDR schema.

You must supply the BEA Application Explorer with the location of the Siebel XDR schema for the integration object you are using.

To start the BEA Application Explorer:

1. Open the BEA Application Explorer.
   - In Windows, choose Windows Start → Programs → BEA Application Explorer.
   - On UNIX, run the startup script `beabse.sh` or the Java command `java com.ibi.common.ui.StartPanel`.

   The BEA Application Explorer window appears.

Setting the Session Path

The session path determines the directory where the BEA Application Explorer places your generated XML schemas and connection information. Your schemas are stored here:

- On Windows: *session_path*\siebel*connection_name*\schemas
- On UNIX: *session_path*/siebel/*connection_name*/schemas

Here, *connection_name* is the value you specify when you select a connection. To learn more about selecting a connection, see "Managing Siebel Connections."

To set the session path:

1. From the File menu, choose Session.

   The Enter Session Path window appears, displaying a default path.

2. Do one of the following:
   - To accept the default session path, click OK.
   - To specify a different path, enter the path and click OK.

     Specifying a different path allows you to group your schema according to project, or other logical group.

Managing Siebel Connections

The BEA Application Explorer must connect to your Siebel system before you can generate schemas. Therefore, you must first define a connect to your Siebel system.

This section includes the following topics:

About Siebel Logon Parameters

In order to create a connection, you must supply valid connection information, including a user name and password for your Siebel system. When creating or editing a connection, the BEA Application Explorer prompts you to provide the following Siebel logon parameters, depending on your Siebel version:

Table 2-1 Siebel Logon Parameters

Siebel Version	Parameter	Description
6.2 or higher	User	Siebel user name
	Password	Password for Siebel user name
	Repository Name	Name of the Siebel Repository (if not the default) where the BEA Application Explorer looks for metadata describing Business Services, Business Objects, and Integration Objects.
6.2 or earlier	Config File	Name of configuration file
6.3 or higher	Siebel Server	Name of the machine running Siebel
	Enterprise Name	Name of the Siebel enterprise
	Siebel Gateway Server	Name of the Siebel Gateway Server
	Object Manager	Name of the Siebel Object Manager

These parameters are used by Siebel client applications to connect to the Siebel system. For more information about these parameters, see your Siebel documentation or consult your Siebel system administrator.

Creating a New Connection

If you are creating a new connection, be sure to check that you have the correct information for your Siebel version for the logon parameters in Table 2-1.

To create a new connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click Siebel → New Connection.

The BEA Application Explorer prompts you for a connection name.

2. Enter a name for this connection and click OK.

   The Select Siebel Version dialog box appears.

3. Select your Siebel version.

   The Siebel Logon dialog box for the version you selected appears.

4. Enter the parameters for your system.

   a. For Siebel 6.2 and earlier, enter these values.

Table 2-2 Connection Parameters for Siebel 6.2 or Earlier

Parameter	Description
User	Your Siebel user name.

Table 2-2 Connection Parameters for Siebel 6.2 or Earlier (Continued)

Parameter	Description
Password	Password for the Siebel user name.
Repository Name	Name of the Siebel Repository (if not the default) where the BEA Application Explorer looks for metadata describing Business Services, Business Objects, and Integration Objects. For example: Devo_1 Repository. If you remove the default without entering another repository name, the BEA Application Explorer interrogates all Siebel repositories and presents the results.
Config file	Uagent configuration file, which contains the configuration parameters for connectivity to Siebel 6.0

b. For Siebel 6.3 and later, enter these values.

Table 2-3 Connection Parameters for Siebel 6.3 or Higher

Parameter	Description
Siebel Server	The name of the Siebel server
Enterprise Name	The name of the enterprise

Table 2-3 Connection Parameters for Siebel 6.3 or Higher (Continued)

Parameter	Description
Gateway Server	The name of your Siebel Gateway Server. To specify a Gateway Server that uses a port other than the default (usually 2320), add a colon and the port number. For example: *gateway_name*:*port_number*
Object Manager	The name of an active Siebel Object Manager (if not the default: EAIObjMgr). For example, SCCObjMgr is the name of the Siebel Object Manager used for the Siebel Call Center application.
Repository Name	Name of the Siebel Repository (if not the default) where the BEA Application Explorer looks for metadata describing Business Services, Business Objects, and Integration Objects. For example: Devo_1 Repository. If you remove the default without entering another repository name, the BEA Application Explorer interrogates all Siebel repositories and presents the results.
User	Your Siebel user name
Password	Password for the Siebel user name

5. Click OK.

   The new connection appears under the Siebel node in the BEA Application Explorer window. You can now view business objects and services, as well as all available integration objects in your Siebel system.

Using an Existing Connection

You can use an existing connection rather than creating a new one.

To use an existing Siebel connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click Siebel → Existing Connection → *your connection*.

   The connection appears below the Siebel node.

2. If the connection parameters do not correspond to your system, edit them in the Siebel Logon Window. For a description of logon parameters, see "About Siebel Logon Parameters" on page 2-7.
3. Click OK.

Disconnecting from Siebel

The BEA Application Explorer allows you to disconnect from Siebel.

To disconnect from Siebel:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Disconnect.

This disconnects from Siebel, and the connection icon change to indicate that it not currently connected. To re-establish the connection, right-click on the connection and choose Connect.

Removing Connections

The BEA Application Explorer allows you to remove connections when you no longer need them.

To remove a connection:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Remove.

Managing Schemas

You need to create a schema for each service and event your application uses. You use the BEA Application Explorer to create these schemas.

This section explains:

- Creating Schemas for Services
- Creating Schemas for Events
- Removing Schemas

Creating Schemas for Services

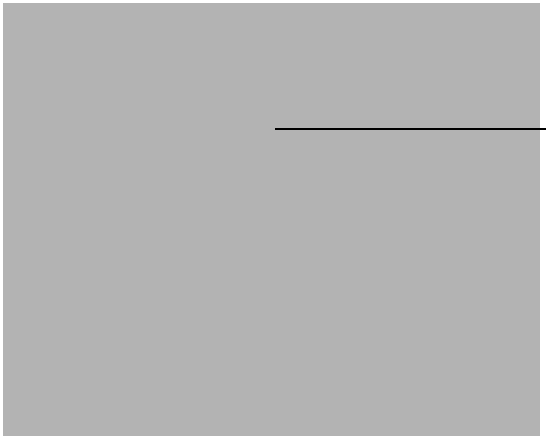
Services require two schemas, one for the request and one for the response. Services always have these two schema, even is the response is not used by your application.

Note: You can create schemas for integration objects, business services, and business components using the steps below.

To create a schema for a service:

1. Start BEA Application Explorer. To learn more, see "Starting the BEA Application Explorer" on page 2-5.
2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see "Setting the Session Path" on page 2-6.
3. Select or create a connection to Siebel. To learn more, see "Managing Siebel Connections" on page 2-6.
4. Expand the tree under Applications → Siebel → *connection name* → Integration Objects to see the items for which you may create a schema. If you cannot expand the tree beneath Siebel, you have not set a connection for Siebel.

5. Select the integration object for this schema.



6. Right-click the item that requires a schema and choose Create Service Schemas.

   The Select XDR window appears.

7. Select the XDR file that corresponds to the service you are creating.

   For example, `ListOfSampleAccount.xml` located in `C:\BEA`. This is the name of the XDR file generated by Siebel for the integration object named Sample Account.

8. Click Open.

   The BEA Application Explorer displays tabs that show the request and response schemas.

The BEA Application Explorer creates a directory structure within the working directory you identified earlier. In this example, the working directory is `C:\BEA\BEASCHEMAS`.

Within this directory, the BEA Application Explorer creates a folder called `Siebel` as well as subfolders to hold the schemas for each configured Siebel connection. In this example, the schemas have been created in the folder called `SiebelConnection`, and the BEA Application Explorer adds the following items to the folder `C:\BEA\BEASCHEMAS\Siebel\SiebelConnection`:

- `manifest.xml`
- `service_Sample Account1-1-FA22.xsd`
- `service_Sample Account1-1-FA22_response.xsd`

You have successfully created service request and response schemas for this integration object.

Creating Schemas for Events

Events only require one schema. There is no response expected when Siebel generates an event.

To create a schema for an event:

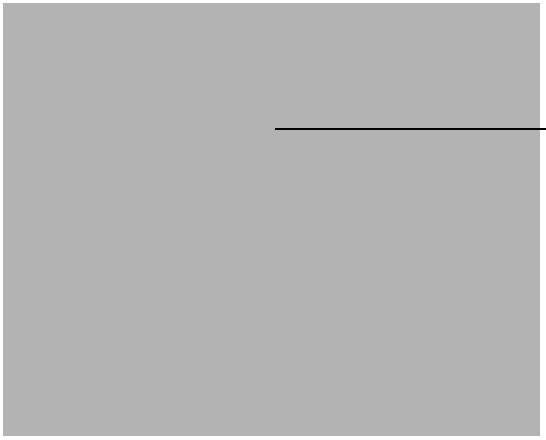
1. Start BEA Application Explorer. To learn more, see "Starting the BEA Application Explorer" on page 2-5.
2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see "Setting the Session Path" on page 2-6.

3. Select or create a connection to your EIS. To learn more, see "Managing Siebel Connections" on page 2-6.

4. Expand the tree under Applications → Siebel → *connection name* → Integration Objects to see the available items. If you cannot expand the tree beneath Siebel, you have not set a connection for Siebel.

Note: You can also select a business service or a business object and create a schema.

5. Select the integration object for this schema.



6. Right-click the item for which you wish to create the schema and choose Create Event Schemas.

   The Select XDR window appears.

7. Select the XDR file that corresponds to the event you are creating.

   For example, `ListOfSampleAccount.xml` located in `C:\BEA`. This is the name of the XDR file generated by Siebel for the integration object named Sample Account.

8. Click Open.

   The BEA Application Explorer displays tabs that show the request and response schemas.

   If it has not already done so, the BEA Application Explorer creates a directory structure within the working directory you identified earlier. In this example, the working directory is `C:\BEA\BEASCHEMAS`.

   Within this directory, the BEA Application Explorer creates a folder called `Siebel` as well as subfolders to hold the schemas for each configured Siebel connection. In this example, the schemas have been created in the folder called `SiebelConnection`, and the BEA Application Explorer adds the following items to the folder `C:\BEA\BEASCHEMAS\Siebel\SiebelConnection`:

- manifest.xml
- event_Sample Account1-1-FA22.xsd

You have successfully created an event schema for this integration object.

Removing Schemas

To remove a schema:

1. Right-click on an integration object for which there is at least one schema.

   If there is an event schema defined for this integration object, the menu has a Remove Event Schemas option.

   If there are service schemas defined for this integration object, the menu has a Remove Event Schema option.

2. Choose the appropriate option.

Next Steps

After you have defined schemas for your events and services, the next step is to create an application view. An application view makes the services and events available to applications. To learn more about application views, see "Defining Application Views for Siebel" on page 3-1.

CHAPTER 3

Defining Application Views for Siebel

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

How to Use This Document

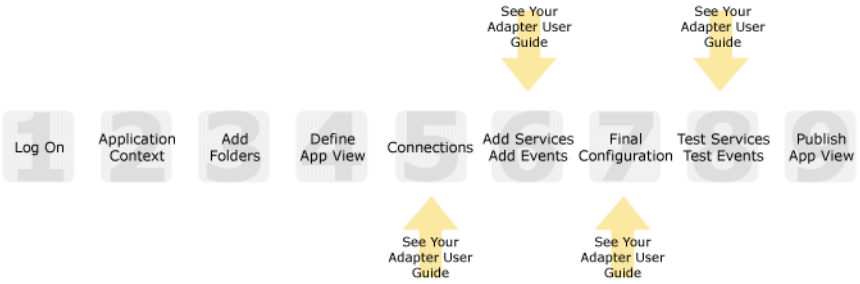
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Siebel system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 3-1 Information Interlock with *Using the Application Integration Design Console*



Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for Siebel Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see "Getting Started With the Adapter for Siebel" on page 1-11.

- Gathered the connection information for your Siebel system. To learn more about the connection information needed by the BEA Application Explorer for your Siebel system, see "Before You Begin" on page 2-2.

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

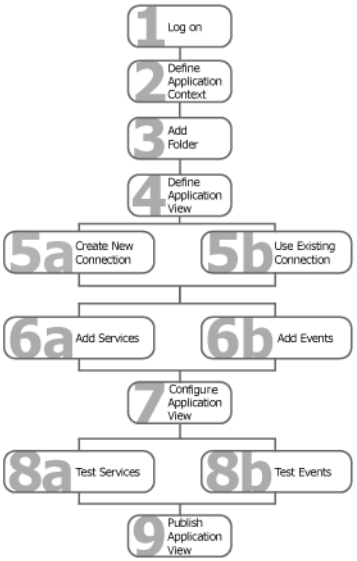
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The information you enter depends on the requirements of your business process and your EIS system configuration. Figure 3-2 summarizes the procedure for defining and configuring an application view.

Figure 3-2 Process for Defining and Configuring an Application View



To define an application view:

1. Log onto the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

   This application will be using the events and services you define in your application view. The application view works within the context of this application.
3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

   If you are adding a new connection service, see "Defining Service Connection Parameters" on page 3-5 for details about Siebel requirements.
6. Add the events and services for this application view.

   See the following sections for details about Siebel requirements:

   – "Setting Service Properties" on page 3-6

– "Setting Event Properties" on page 3-16

7. Perform final configuration tasks.

   If you are adding an event connection, see "Defining Event Connection Parameters" on page 3-21 for details about Siebel requirements.

8. Test all services and events to make sure they can properly interact with the target Siebel system.

   See the following sections for details about Siebel requirements:

   – "Testing Services" on page 3-24

   – "Testing Events Using a Service" on page 3-25

   – "Testing Events Manually" on page 3-26

9. Publish the application view to the target WebLogic Workshop application.

   This is the application you specified in step 2. Publishing the application view allows business process developers within the target application to interact with the newly published application view using an Application View control.

Defining Service Connection Parameters

This information applies to "Step 5A, Create a New Browsing Connection" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

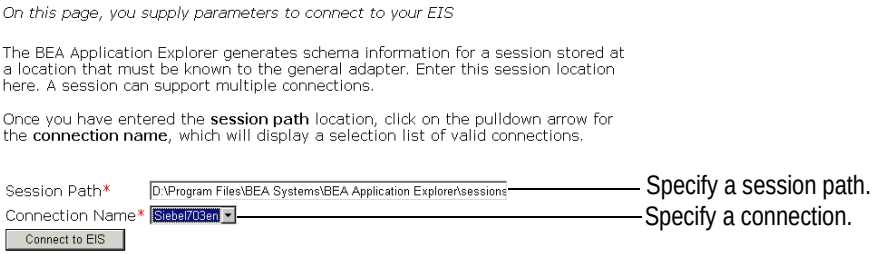
The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.

After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

   The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.



Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

   This information enables the application view to interact with the target Siebel system. You need enter this information only once per application view.

3. Click Connect to EIS.

   You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Service Properties

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Adapter for Siebel uses services to make requests of the Siebel system. A service consists of both a request and a response. Adapter for Siebel supports the following services:

- File Service
- HTTP Service
- MQ Service
- MQRead
- Siebel JavaAPI

- Siebel 6 COMData

File Service

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

A File service sends a Siebel XML file to a specific directory on disk. It requires a Siebel Workflow within the Siebel system. The Siebel Workflow responds to the Siebel XML it receives from the adapter, which triggers a Siebel business event.

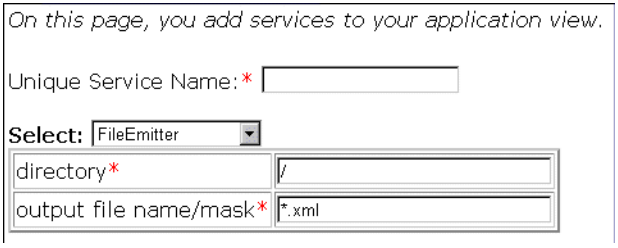
To configure a File service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select File from the Select list.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-1 File Service Parameters

Parameter	Description
directory	The target file system location for the Siebel XML file
output file name/mask	The file name to be used for the output file generated as a result of this operation

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

HTTP Service

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An HTTP service sends a Siebel XML file through HTTP to the Siebel system. It requires a Siebel Workflow within the Siebel system. The Siebel Workflow responds to the Siebel XML it receives from the adapter, which triggers a Siebel business event.

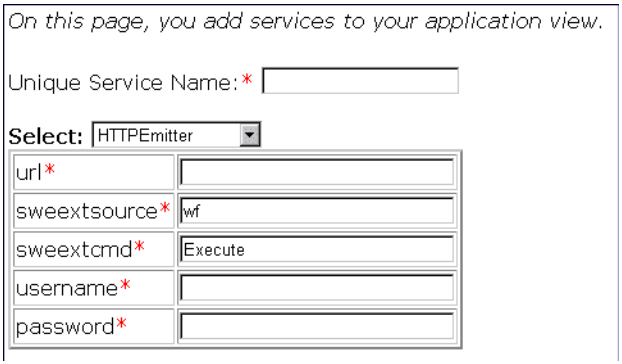
To configure an HTTP Service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select HTTPEmitter from the Select list.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-2 HTTP Service Parameters

Parameter	Description
URL	The URL where Siebel is listening. For example, `http://ariba01/eai/start.swe`
sweextsource	The name of the Siebel Workflow process to be invoked. For example, `Workflow.wf`
sweextcmd	The Siebel command value. For example, `Execute`.
username	Your Siebel system user name
password	The corresponding password for your Siebel system user name

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

MQ Service

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An MQ service sends a Siebel XML file to an IBM MQSeries or WebSphere MQ queue. It requires a Siebel Workflow within the Siebel system. The Siebel Workflow responds to the Siebel XML it receives from the adapter, which triggers a Siebel business event.

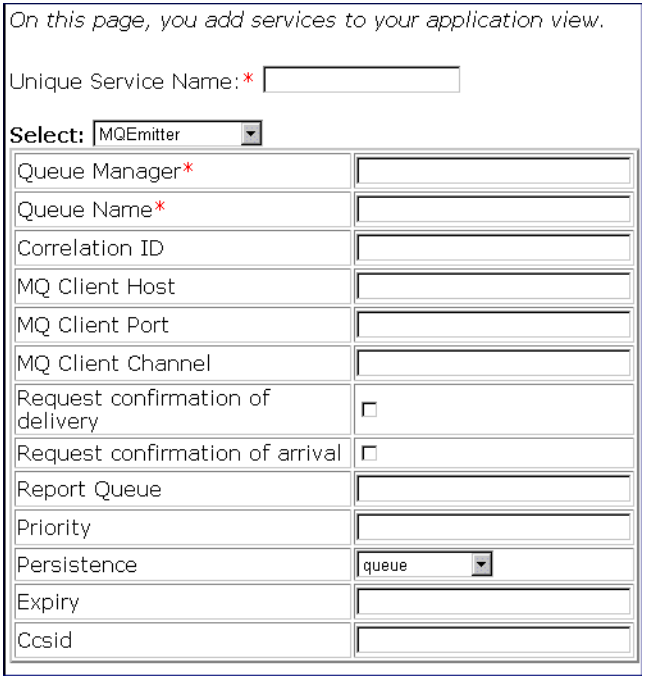
To configure an MQ Service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select MQEmitter from the Select list.

   The Add Services page displays the fields required for this service type.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-3 MQ Service Parameters

Parameter	Description
Queue Manager	The name of the MQ Queue Manager to be used
Queue Name	The name of the MQSeries or WebSphere MQ queue that the Siebel MQSeries Receiver process polls for an XML document
Correlation ID	The correlation ID used in the MQ message header
MQ Client Host	For MQ Client only, the host where the MQ Server is located
MQ Client Port	For MQ Client only, the port number used to connect to an MQ Server
MQ Client Channel	For MQ Client only, the channel between an MQ Client and MQ Server

Table 3-3 MQ Service Parameters (Continued)

Parameter	Description
Delivery confirmation	Sends a report to the report queue confirming message delivery
Arrival confirmation	Sends a report to the report queue confirming message arrival
Report queue	Name of the report queue
Priority	Sets the priority of the message in the message queue, and can be used to determine the order in which messages are retrieved from the queue; 0 is lowest priority and 9 is highest priority
Persistence	Sets the persistence level for messages if the queue manager is restarted. The default is the same as the MQ queue. • If the setting is persistent, the message is recovered and remains on the queue. • If the setting is non-persistent, messages are no longer on the queue.
Expiry	The time that the message remains available in the system awaiting pickup (measured in 1/ 10 seconds). The default is no expiration.
CCSID	The character set encoding for inbound documents. For example, UTF-8.

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

MQRead

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An MQRead service initiates a request to read a single row from a specified IBM MQSeries of WebSphere MQ queue. This is a synchronous service that differs from an MQ event because it does not poll the queue. It requires a Siebel Workflow within the Siebel system. The Siebel Workflow responds to the Siebel XML it receives from the adapter, using an MQ read agent to read the results of a workflow.

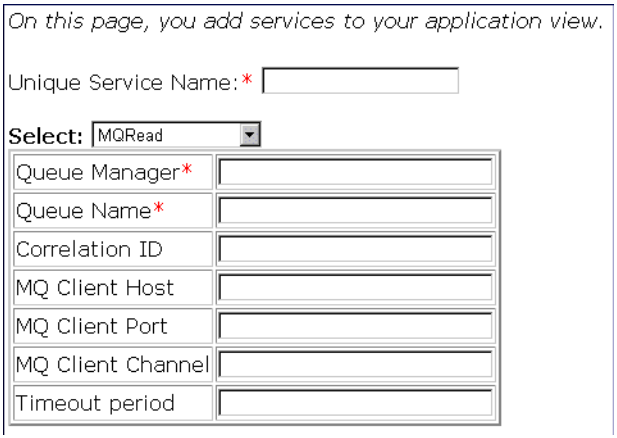
To configure an MQRead Service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select MQRead from the Select list.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-4 MQRead Service Parameters

Parameter	Definition
Queue Manager	The name of the MQ Queue Manager to be used
Queue_Name	The name of the MQSeries or WebSphere MQ queue that the Siebel MQSeries Receiver process polls for an XML document
Correlation_ID	The correlation ID used in the MQ message header
MQ Client Host	For MQ Client only. The host where the MQ Server is located
MQ Client Port	For MQ Client only. The port number used to connect to an MQ Server
MQ Client Channel	For MQ Client only. The channel between an MQ Client and MQ Server
Timeout period	The length of time, in seconds, to stay connected to the MQ Series queue awaiting receipt of a message.

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

Siebel JavaAPI

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

A Siebel JavaAPI Service accesses and integrates directly with Siebel Business Services and Business Components without using Siebel Workflows. This service uses a TCP connection, bypassing a transport layer such as IBM MQSeries, File, or HTTP. The service request begins with by sending a service request document. In most cases, the result is an XML response document indicating that the business service or business component has executed. This service is available for Siebel versions 6.3 or later.

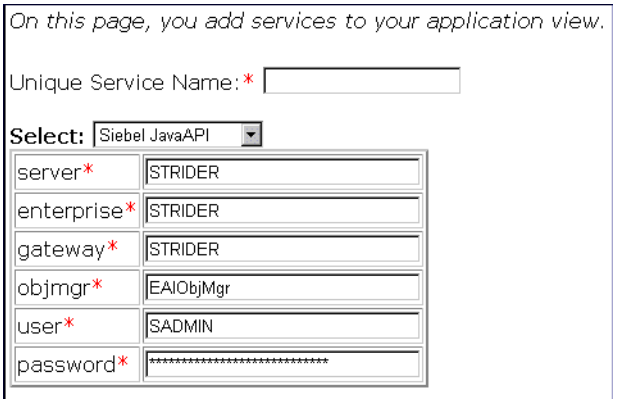
To configure a Siebel JavaAPI Service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select Siebel JavaAPI from the Select list.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-5 Siebel JavaAPI Service Parameters

Parameter	Description
server	The name of the Siebel Application Server
enterprise	The name of the Siebel Enterprise to which this server belongs
gateway	The Siebel gateway name server
user	The Siebel administrator user name
password	The password of the administrative user

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

Siebel 6 COMData

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

A Siebel 6 COMData Service accesses and integrates directly with Siebel Business Services and Business Components without using Siebel Workflows. This service uses a TCP connection, bypassing a transport layer such as IBM MQSeries, File, or HTTP. The service request begins with by sending a service request document. In most cases, the result is an XML response document indicating that the business service or business component has executed. This service is available for Siebel versions 6.2 or earlier.

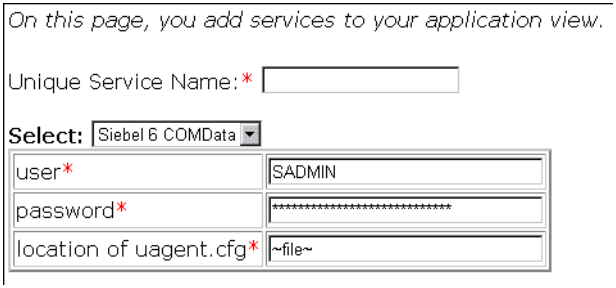
To configure a Siebel 6 COMData Service:

1. Enter a unique service name that describes the function the service performs.

   Valid characters for the service name include a-z, A-Z, 0-9, and _ (underscore).

2. Select Siebel 6 COMData from the Select list.

   The Add Services page displays the fields required for this service type.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-6 Siebel 6 COMData Service Parameters

Parameter	Description
user	The Siebel administrator user name
password	The password of the administrative user
location of uagent.cfg	The file system location of the `uagent.cfg` configuration file

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

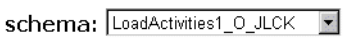
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

You select a schema and select logging options the same way for all services.

To set common service settings:

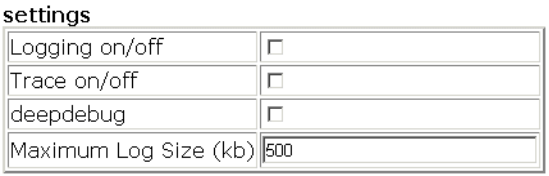
1. In the Schema list, select the schema you want to use with this service.

   For more information, see Chapter 2, "Generating Schemas for Siebel Integration Objects."

2. Configure logging and tracing for this service, as follows:

   Logging captures information from your adapter and writes it in a log file. Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.



   a. Select the Logging on/off check box to enable logging for this service. Logging information is written to a log file (`BEA_FILE_1_0.log`) that appears in the directory from which the application was started.

   b. Select the Trace on/off check box to enable tracing for this service. Trace information appears in the runtime console.

   c. Select the deepdebug check box to enable additional trace information for deeper troubleshooting.

   d. Enter the maximum log file size.

3. Click Add to add the service.

   For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Event Properties

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An event defines how your application responds to events generated by Siebel. The Adapter for Siebel supports the following events:

- MQ Event

- File Event
- HTTP Event

MQ Event

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

In an MQ event, the adapter picks up a Siebel XML file from a specific IBM MQSeries or WebSphere MQ queue and passes it to an event variable that is set in a business process. It requires a Siebel Workflow within the Siebel system, which sends Siebel XML to the adapter when a Siebel event occurs.

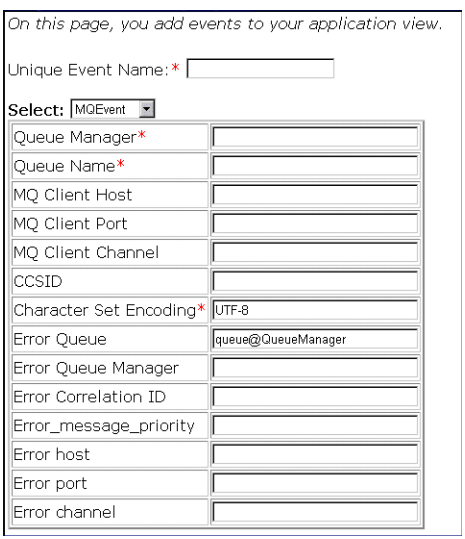
To configure an MQ event:

1. Enter a unique event name that describes the function the event performs.

   Valid characters for the event name include a-z, A-Z, 0-9, and _ (underscore).

2. Select MQEvent from the Select list.

   The Add Events page displays the fields required for this event type.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-7 MQ Event Parameters

Parameter	Description
Queue Manager	The name of the MQ Queue Manager to be used
Input Queue Name	The name of the MQSeries or WebSphere MQ queue that the Siebel MQSeries Receiver process polls for an XML document
MQ Client Host	For MQ Client only. The host where the MQ Server is located
MQ Client Port	For MQ Client only. The port number used to connect to an MQ Server
MQ Client Channel	For MQ Client only. The channel between an MQ Client and MQ Server
Polling Interval	The maximum wait interval (in the format *nn*H:*nn*M:*nn*S) between checks for new documents. The higher this value, the longer the interval, and the fewer system resources that are used. However, with a high value, the worker thread cannot respond to a stop command. If timeout is set to 0, the listener runs once and terminates. Default is 2 seconds.
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.
Error Queue	The name of the error queue to be used
Error Queue Manager	The name of the error queue manager to be used
Error Correlation ID	The correlation ID used in the MQ error message header
Error_message_priority	Sets the priority of the error message, 0 is lowest and 9 is highest
Error host	For MQ Client only, the host where the error queue is located
Error port	For MQ Client only, the port number used to connect to the error queue
Error channel	For MQ Client only, the channel between an MQ Server and error queue

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

File Event

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

In a File event, the adapter picks up a Siebel XML file from a specific location on disk and passes it to an event variable that is set in a business process. It requires a Siebel Workflow within the Siebel system, which sends Siebel XML to the adapter when a Siebel event occurs.

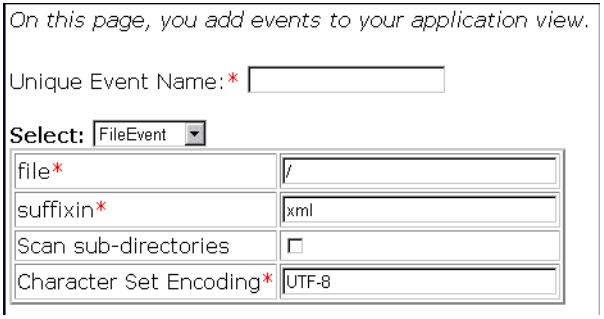
To configure a File event:

1. Enter a unique event name that describes the function the event performs.

   Valid characters for the event name include a-z, A-Z, 0-9, and _ (underscore).

2. Select FileEvent from the Select list.

   The Add Events page displays the fields required for this event type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-8 File Event Parameters

Parameter	Description
file	The file system location to be polled for the file event.
Suffixin	File extension for the file event (typically XML)

Table 3-8 File Event Parameters (Continued)

Parameter	Description
scansubs	Setting for scanning subdirectories for document to be processed. TRUE or FALSE.
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

HTTP Event

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

In an HTTP event, the adapter picks up a Siebel XML file via HTTP and passes it to an event variable that is set in a business process. It requires a Siebel Workflow within the Siebel system, which sends Siebel XML to the adapter when a Siebel event occurs.

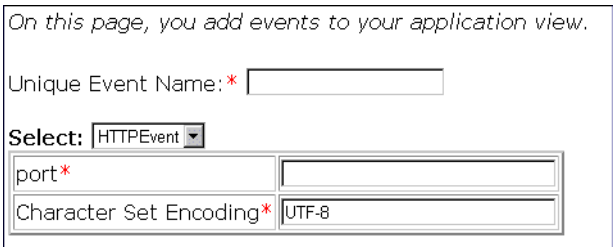
To configure an HTTP event:

1. Enter a unique event name that describes the function the event performs.

   Valid characters for the event name include a-z, A-Z, 0-9, and _ (underscore).

2. Select FileEvent from the Select list.

   The Add Events page displays the fields required for this event type.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-9 HTTP Event Parameters

Parameter	Description
port	The port where the adapter should listen for the HTTP transfer.
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.

4. See "Common Service and Event Settings" on page 3-15 for information about selecting a schema and configuring logging and tracing.

Defining Event Connection Parameters

This information applies to "Step 7, Perform Final Configuration Tasks" in *Using the Application Integration Design Console*, at the following URL:

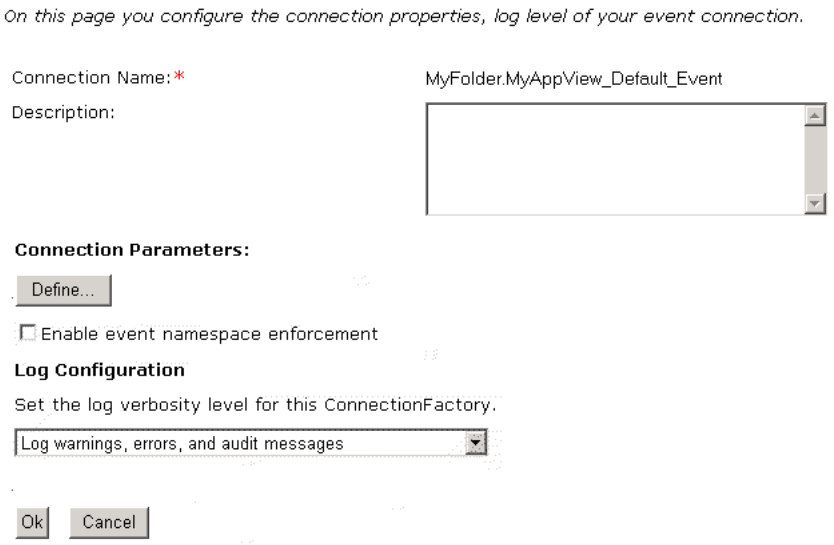
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Connection Information page.

To define event connection parameters:

1. In the Connections area on the Application View Administration page, click Select/Edit to display the Connection Information page.
2. Under Event Connection, click the Select Existing link to display the Select Event Connection page.
3. Click the Event option button to select the default event connection, and then click OK.

The Edit Event Connection page is displayed.



The Edit Event Connection page allows you to define event connection parameters, enable the namespace enforcement option, and specify what information will be logged for the connection factory.

4. Check the Enable event namespace enforcement check box to enable namespace enforcement.

   Enabling the namespace enforcement option indicates that the client requires response documents and event documents to declare the namespace indicated in the response/event definitions, and it forces the proper namespace declaration onto the response/event if needed.

   The BEA WebLogic Adapter for Siebel does not provide responses/events using the namespace declared in the response/event schema This option allows clients that perform schema-based XML checking to use this adapter.

   Note: Because the BEA WebLogic Adapter for Siebel returns raw XML text (not parsed), enabling namespace enforcement has performance implications because it forces a parse of the XML text in order to inject the proper namespace declaration.

5. Select one of the following settings for the log:
   - Log errors and audit messages (recommended)
   - Log warnings, errors, and audit messages
   - Log informational, warning, error, and audit messages
   - Log all messages

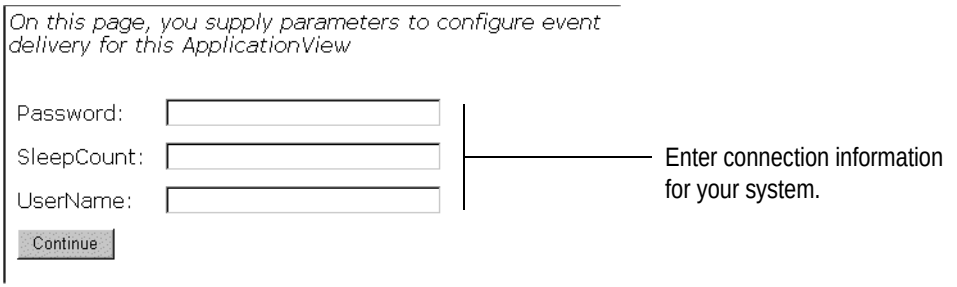
   The table that follows describes the type of information that each logging message contains.

Table 3-10 Logging Message Categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

6. Click Define to enter the event-delivery parameters.

   The parameters you enter on this page enable connection to your Siebel and are used when generating events. The parameters are specific to the associated adapter and are defined in the `weblogic-ra.xml` file within the base adapter.



   Note: A red asterisk (*) indicates that a field is required.

7. Enter the following information:

Table 3-11 Event Connection Parameters

Parameter	Description
Password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events
UserName	Your WebLogic Server Administration Console user name, defined in the `startWebLogic` script

8. Click Continue to save your settings and return to the Edit Event Connection page, and then click OK to return to the Connection Information page.

Testing Services

This information applies to "Step 8A, Test an Application View's Services" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target Siebel system. When you test a service, you supply any inputs required to start the service. For the Adapter for Siebel, the input is in the form of a valid XML string based on the schema you created for your Siebel application. You can create the XML file by running XML Spy, which is installed with WebLogic Integration, against your schema files.

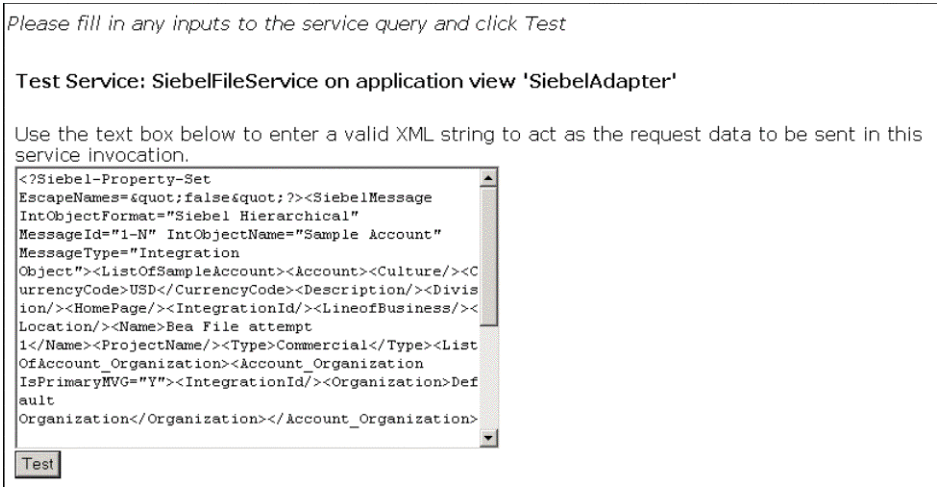
Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test a service:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Services page appears.

2. In the Test Service window, copy the appropriate XML strings from the Siebel XML file from your account for your service type.



3. Click Test.

   The results appear in the Test Results window.

   If you are testing an MQ service, you can also check your MQSeries Explorer for results (choose Queue Managers → *MyQueueManager* → Queues). If the service ran correctly, the queue now contains the XML message.

Testing Events Using a Service

1 2 3 4 5 6 7 **8** 9

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by Siebel. When you test an event, you can trigger the event using a service or manually.

Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test an event:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Events page appears.

2. Click Service and select a service that triggers the event you are testing.

3. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

4. Click Test and enter the XML string needed to trigger the service.

   The service is executed.

   – If the test succeeds, the Test Result page appears, showing the event document, the service input document, and the service output document.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing Events Manually

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Note: Before you test an application view, it must contain at least one event or service. Also, you must place the application view in test mode. To place an application view in test mode, click the Test button at the bottom of the Application View Administration page.

To test an event manually:

1. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

2. Click Test. The test waits for an event to trigger it.

3. In your Siebel system, execute a Siebel Workflow that sends Siebel XML to the adapter.

4. Using the triggering Siebel application, perform an action that executes the service that, in turn, tests the application view event.

   – If the test succeeds, the Test Result page displays the event document from the application.

   – If the test fails or takes too long, the Test Result page displays a Timed Out message.

APPENDIX A

Sample Files

This section provides the following sample schemas for Siebel Business Components and Siebel Business Services:

- Account Request Schema
- Account Response Schema
- Sample XML for Account Add Request
- Sample XML for Account Add Response
- Sample XML for Account Delete Request
- Sample XML for Account Delete Response
- Sample XML for Account Query Request
- Sample XML for Account Query Response
- PGMAVV Account Add Request Schema
- PGMAVV Account Add Response Schema
- Sample XML for PGMAVV Account Add Request
- Sample XML for PGMAVV Account Add Response
- Sample Account Service Request Schema
- Sample Account Service Response Schema

- Sample Event Schema
- Sample XDR for Sample Account

Account Request Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <xsd:element name="BusinessComponent">

    <xsd:complexType>

      <xsd:sequence>

<xsd:element name="Select" minOccurs="0" maxOccurs="unbounded" >

    <xsd:complexType>

        <xsd:sequence/>

        <xsd:attribute name="name" use="required">

            <xsd:simpleType>

                <xsd:restriction base="xsd:string">

                    <xsd:enumeration value="Account Competitors"/>

                    <xsd:enumeration value="Account Condition"/>

                    <xsd:enumeration value="Account Markets"/>

                 <xsd:enumeration value="Account Organization Integration Id"/>

                    <xsd:enumeration value="Account Products"/>

                    <xsd:enumeration value="Account Role"/>

                    <xsd:enumeration value="Account Status"/>

                    <xsd:enumeration value="Account Trend"/>

                    <xsd:enumeration value="Address Active Status"/>

                    <xsd:enumeration value="Address Id"/>

                    <xsd:enumeration value="Address Integration Id"/>

                    <xsd:enumeration value="Agreement End Date"/>
```

```
<xsd:enumeration value="Agreement Name"/>
<xsd:enumeration value="Agreement Start Date"/>
<xsd:enumeration value="Agreement Status"/>
<xsd:enumeration value="Algorithm Type"/>
<xsd:enumeration value="Alias"/>
<xsd:enumeration value="Annual Revenue"/>
<xsd:enumeration value="Assignment Area Code"/>
<xsd:enumeration value="Assignment Country Code"/>
<xsd:enumeration value="Assignment Excluded"/>
<xsd:enumeration value="Assignment Manual Flag"/>
<xsd:enumeration value="Back Office Distribution Channel"/>
<xsd:enumeration value="Back Office Order Query End  Dt"/>
<xsd:enumeration value="Back Office Order Query Start  Dt"/>
<xsd:enumeration value="Back Office Sales Area Division Code"/>
<xsd:enumeration value="Back Office Sales Organization"/>
<xsd:enumeration value="Bill Address Flag"/>
<xsd:enumeration value="Bill To City"/>
<xsd:enumeration value="Bill To Country"/>
<xsd:enumeration value="Bill To First Name"/>
<xsd:enumeration value="Bill To Id"/>
<xsd:enumeration value="Bill To Job Title"/>
<xsd:enumeration value="Bill To Last Name"/>
<xsd:enumeration value="Bill To Postal Code"/>
<xsd:enumeration value="Bill To State"/>
<xsd:enumeration value="Bill To Street Address"/>
<xsd:enumeration value="Block Credit Flag"/>
<xsd:enumeration value="Business Profile"/>
```

```
<xsd:enumeration value="CSN"/>
<xsd:enumeration value="City"/>
<xsd:enumeration value="City State"/>
<xsd:enumeration value="Competitor"/>
<xsd:enumeration value="Country"/>
<xsd:enumeration value="County"/>
<xsd:enumeration value="Credit Control Area Code"/>
<xsd:enumeration value="Credit Currency Code"/>
<xsd:enumeration value="Credit Limit Amount"/>
<xsd:enumeration value="Credit Profile Id"/>
<xsd:enumeration value="Culture"/>
<xsd:enumeration value="Currency Code"/>
<xsd:enumeration value="Current Volume"/>
<xsd:enumeration value="Current Volume Currency Code"/>
<xsd:enumeration value="Current Volume Exchange Date"/>
<xsd:enumeration value="Customer Account Group"/>
<xsd:enumeration value="DNBReport"/>
<xsd:enumeration value="DUNS Intcode"/>
<xsd:enumeration value="DUNS Number"/>
<xsd:enumeration value="Date Formed"/>
<xsd:enumeration value="DeDup Key Modification Date"/>
<xsd:enumeration value="DeDup Key Update"/>
<xsd:enumeration value="DeDup Keys"/>
<xsd:enumeration value="DeDup Last Match Date"/>
<xsd:enumeration value="DeDup Token"/>
<xsd:enumeration value="Deduplication Match Score"/>
<xsd:enumeration value="Deduplication Object Id"/>
```

```
<xsd:enumeration value="Description"/>
<xsd:enumeration value="Disable DataCleansing"/>
<xsd:enumeration value="Division"/>
<xsd:enumeration value="Domestic Ultimate DUNS"/>
<xsd:enumeration value="Dummy"/>
<xsd:enumeration value="EAI Sync Date"/>
<xsd:enumeration value="EAI Sync Error Text"/>
<xsd:enumeration value="EAI Sync Status Code"/>
<xsd:enumeration value="Email Address"/>
<xsd:enumeration value="Employee Here"/>
<xsd:enumeration value="Employees"/>
<xsd:enumeration value="Expertise"/>
<xsd:enumeration value="Explorer Label"/>
<xsd:enumeration value="Fax Number"/>
<xsd:enumeration value="Fiscal Year End"/>
<xsd:enumeration value="Freight Terms"/>
<xsd:enumeration value="Freight Terms Info"/>
<xsd:enumeration value="Full Address"/>
<xsd:enumeration value="GSA Flag"/>
<xsd:enumeration value="Global Ultimate DUNS"/>
<xsd:enumeration value="Goals"/>
<xsd:enumeration value="Group Type Code"/>
<xsd:enumeration value="Home Page"/>
<xsd:enumeration value="Industry Condition"/>
<xsd:enumeration value="Industry Trend"/>
<xsd:enumeration value="Integration Id"/>
<xsd:enumeration value="Internal Org Flag"/>
```

```
<xsd:enumeration value="Joined Synonym"/>
<xsd:enumeration value="Key Competitors"/>
<xsd:enumeration value="Language Code"/>
<xsd:enumeration value="Last Clnse Date"/>
<xsd:enumeration value="Last Manager Review Date"/>
<xsd:enumeration value="Last Review Manager Id"/>
<xsd:enumeration value="Line of Business"/>
<xsd:enumeration value="Location"/>
<xsd:enumeration value="Location Level"/>
<xsd:enumeration value="Main Address Flag"/>
<xsd:enumeration value="Main Fax Number"/>
<xsd:enumeration value="Main Phone Number"/>
<xsd:enumeration value="Managers Review"/>
<xsd:enumeration value="Marketing"/>
<xsd:enumeration value="Merge Sequence Number"/>
<xsd:enumeration value="Mission"/>
<xsd:enumeration value="Name"/>
<xsd:enumeration value="Name and Location"/>
<xsd:enumeration value="Not Manager Flag"/>
<xsd:enumeration value="Notes"/>
<xsd:enumeration value="Objectives"/>
<xsd:enumeration value="Organization Id"/>
<xsd:enumeration value="Organization Integration Id"/>
<xsd:enumeration value="Our Position"/>
<xsd:enumeration value="Outline Number"/>
<xsd:enumeration value="PO Approved Flag"/>
<xsd:enumeration value="PO Auto Approval Currency Code"/>
```

```
<xsd:enumeration value="PO Auto Approval Date"/>
<xsd:enumeration value="PO Auto Approval Limit"/>
<xsd:enumeration value="Parent Account Division"/>
<xsd:enumeration value="Parent Account Id"/>
<xsd:enumeration value="Parent Account Integration Id"/>
<xsd:enumeration value="Parent Account Location"/>
<xsd:enumeration value="Parent Account Location Level"/>
<xsd:enumeration value="Parent Account Name"/>
<xsd:enumeration value="Parent Account Region"/>
<xsd:enumeration value="Parent HQ DUNS"/>
<xsd:enumeration value="Partner Flag"/>
<xsd:enumeration value="Partners"/>
<xsd:enumeration value="Party Name"/>
<xsd:enumeration value="Party Type Code"/>
<xsd:enumeration value="Party UId"/>
<xsd:enumeration value="Philosophy"/>
<xsd:enumeration value="Phone Number"/>
<xsd:enumeration value="Position Integration Id"/>
<xsd:enumeration value="Postal Code"/>
<xsd:enumeration value="Price List"/>
<xsd:enumeration value="Price List End Date"/>
<xsd:enumeration value="Price List Id"/>
<xsd:enumeration value="Price List Integration Id"/>
<xsd:enumeration value="Price List Start Date"/>
<xsd:enumeration value="Primary Account City"/>
<xsd:enumeration value="Primary Account Country"/>
<xsd:enumeration value="Primary Account Postal Code"/>
```

```
                <xsd:enumeration value="Primary Account State"/>
                <xsd:enumeration value="Primary Account Street Address"/>
                <xsd:enumeration value="Primary Address Id"/>
                <xsd:enumeration value="Primary Assignment Denorm Flag"/>
                <xsd:enumeration value="Primary Assignment Manual Flag"/>
                <xsd:enumeration value="Primary Assignment System Flag"/>
                <xsd:enumeration value="Primary Assignment Type"/>
                <xsd:enumeration value="Primary Bill To Address Id"/>
                <xsd:enumeration value="Primary Bill To City"/>
                <xsd:enumeration value="Primary Bill To Country"/>
                <xsd:enumeration value="Primary Bill To First Name"/>
                <xsd:enumeration value="Primary Bill To Job Title"/>
                <xsd:enumeration value="Primary Bill To Last Name"/>
                <xsd:enumeration value="Primary Bill To Person Id"/>
                <xsd:enumeration value="Primary Bill To Postal Code"/>
                <xsd:enumeration value="Primary Bill To State"/>
                <xsd:enumeration value="Primary Bill To Street Address"/>
                <xsd:enumeration value="Primary Category Id"/>
            <xsd:enumeration value="Primary Fulfill InvLoc Integration Id"/>
                <xsd:enumeration value="Primary Fulfillment InvLoc ID"/>
                <xsd:enumeration value="Primary Fulfillment Inventory
Location"/>
                <xsd:enumeration value="Primary Industry Id"/>
                <xsd:enumeration value="Primary Organization"/>
                <xsd:enumeration value="Primary Organization Id"/>
                <xsd:enumeration value="Primary Payer Account"/>
                <xsd:enumeration value="Primary Payer Account Id"/>
                <xsd:enumeration value="Primary Position Id"/>
```

```
<xsd:enumeration value="Primary Service Agreement Id"/>
<xsd:enumeration value="Primary Ship To Address Id"/>
<xsd:enumeration value="Primary Ship To City"/>
<xsd:enumeration value="Primary Ship To Country"/>
<xsd:enumeration value="Primary Ship To First Name"/>
<xsd:enumeration value="Primary Ship To Job Title"/>
<xsd:enumeration value="Primary Ship To Last Name"/>
<xsd:enumeration value="Primary Ship To Person Id"/>
<xsd:enumeration value="Primary Ship To Postal Code"/>
<xsd:enumeration value="Primary Ship To State"/>
<xsd:enumeration value="Primary Ship To Street Address"/>
<xsd:enumeration value="Primary Synonym Id"/>
<xsd:enumeration value="Primary Territory Id"/>
<xsd:enumeration value="Primary Type Id"/>
<xsd:enumeration value="Profit"/>
<xsd:enumeration value="Project Bill Type"/>
<xsd:enumeration value="Project Comments"/>
<xsd:enumeration value="Project Fix Fee"/>
<xsd:enumeration value="Project Hour Limit"/>
<xsd:enumeration value="Project Id"/>
<xsd:enumeration value="Project Name"/>
<xsd:enumeration value="Project Percentage of Fee"/>
<xsd:enumeration value="Project Purchase Order"/>
<xsd:enumeration value="Project Relationship Type"/>
<xsd:enumeration value="Project Role"/>
<xsd:enumeration value="Prospect Flag"/>
<xsd:enumeration value="Province"/>
```

```
<xsd:enumeration value="Public"/>
<xsd:enumeration value="Reference Date"/>
<xsd:enumeration value="Reference Flag"/>
<xsd:enumeration value="Reference Stage"/>
<xsd:enumeration value="Region"/>
<xsd:enumeration value="Relationship Level"/>
<xsd:enumeration value="Relationship Type"/>
<xsd:enumeration value="Response Time"/>
<xsd:enumeration value="Revenue"/>
<xsd:enumeration value="Revenue Growth"/>
<xsd:enumeration value="Revision Number"/>
<xsd:enumeration value="Row Status"/>
<xsd:enumeration value="Row Status Asterisk"/>
<xsd:enumeration value="S-S Instance"/>
<xsd:enumeration value="S-S Instance Id"/>
<xsd:enumeration value="S-S Key Id"/>
<xsd:enumeration value="Service Calendar"/>
<xsd:enumeration value="Service Type"/>
<xsd:enumeration value="Ship Address Flag"/>
<xsd:enumeration value="Ship To City"/>
<xsd:enumeration value="Ship To Country"/>
<xsd:enumeration value="Ship To First Name"/>
<xsd:enumeration value="Ship To Job Title"/>
<xsd:enumeration value="Ship To Last Name"/>
<xsd:enumeration value="Ship To Postal Code"/>
<xsd:enumeration value="Ship To State"/>
<xsd:enumeration value="Ship To Street Address"/>
```

```
                    <xsd:enumeration value="Start Date"/>
                    <xsd:enumeration value="State"/>
                    <xsd:enumeration value="Strategies"/>
                    <xsd:enumeration value="Strategy"/>
                    <xsd:enumeration value="Street Address"/>
                    <xsd:enumeration value="Street Address 2"/>
                    <xsd:enumeration value="Success Factors"/>
                    <xsd:enumeration value="Synonym"/>
                    <xsd:enumeration value="Territory"/>
                    <xsd:enumeration value="Territory Id"/>
                    <xsd:enumeration value="Timestamp"/>
                    <xsd:enumeration value="Today"/>
                    <xsd:enumeration value="Total Potential Volume"/>
                <xsd:enumeration value="Total Potential Volume Currency Code"/>
                <xsd:enumeration value="Total Potential Volume Exchange Date"/>
                    <xsd:enumeration value="Type"/>
                    <xsd:enumeration value="Type MVF"/>
                    <xsd:enumeration value="VAT registration number"/>
                    <xsd:enumeration value="Value Proposition"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="Field" minOccurs="0" maxOccurs="unbounded" >
    <xsd:complexType>
```

```
<xsd:sequence/>
<xsd:attribute name="name" use="required">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Account Competitors"/>
            <xsd:enumeration value="Account Condition"/>
            <xsd:enumeration value="Account Markets"/>
          <xsd:enumeration value="Account Organization Integration Id"/>
            <xsd:enumeration value="Account Products"/>
            <xsd:enumeration value="Account Role"/>
            <xsd:enumeration value="Account Status"/>
            <xsd:enumeration value="Account Trend"/>
            <xsd:enumeration value="Address Active Status"/>
            <xsd:enumeration value="Address Id"/>
            <xsd:enumeration value="Address Integration Id"/>
            <xsd:enumeration value="Agreement End Date"/>
            <xsd:enumeration value="Agreement Name"/>
            <xsd:enumeration value="Agreement Start Date"/>
            <xsd:enumeration value="Agreement Status"/>
            <xsd:enumeration value="Algorithm Type"/>
            <xsd:enumeration value="Alias"/>
            <xsd:enumeration value="Annual Revenue"/>
            <xsd:enumeration value="Assignment Area Code"/>
            <xsd:enumeration value="Assignment Country Code"/>
            <xsd:enumeration value="Assignment Excluded"/>
            <xsd:enumeration value="Assignment Manual Flag"/>
           <xsd:enumeration value="Back Office Distribution Channel"/>
```

```
    <xsd:enumeration value="Back Office Order Query End  Dt"/>
   <xsd:enumeration value="Back Office Order Query Start  Dt"/>
<xsd:enumeration value="Back Office Sales Area Division Code"/>
    <xsd:enumeration value="Back Office Sales Organization"/>
    <xsd:enumeration value="Bill Address Flag"/>
    <xsd:enumeration value="Bill To City"/>
    <xsd:enumeration value="Bill To Country"/>
    <xsd:enumeration value="Bill To First Name"/>
    <xsd:enumeration value="Bill To Id"/>
    <xsd:enumeration value="Bill To Job Title"/>
    <xsd:enumeration value="Bill To Last Name"/>
    <xsd:enumeration value="Bill To Postal Code"/>
    <xsd:enumeration value="Bill To State"/>
    <xsd:enumeration value="Bill To Street Address"/>
    <xsd:enumeration value="Block Credit Flag"/>
    <xsd:enumeration value="Business Profile"/>
    <xsd:enumeration value="CSN"/>
    <xsd:enumeration value="City"/>
    <xsd:enumeration value="City State"/>
    <xsd:enumeration value="Competitor"/>
    <xsd:enumeration value="Country"/>
    <xsd:enumeration value="County"/>
    <xsd:enumeration value="Credit Control Area Code"/>
    <xsd:enumeration value="Credit Currency Code"/>
    <xsd:enumeration value="Credit Limit Amount"/>
    <xsd:enumeration value="Credit Profile Id"/>
    <xsd:enumeration value="Culture"/>
```

```
<xsd:enumeration value="Currency Code"/>
<xsd:enumeration value="Current Volume"/>
<xsd:enumeration value="Current Volume Currency Code"/>
<xsd:enumeration value="Current Volume Exchange Date"/>
<xsd:enumeration value="Customer Account Group"/>
<xsd:enumeration value="DNBReport"/>
<xsd:enumeration value="DUNS Intcode"/>
<xsd:enumeration value="DUNS Number"/>
<xsd:enumeration value="Date Formed"/>
<xsd:enumeration value="DeDup Key Modification Date"/>
<xsd:enumeration value="DeDup Key Update"/>
<xsd:enumeration value="DeDup Keys"/>
<xsd:enumeration value="DeDup Last Match Date"/>
<xsd:enumeration value="DeDup Token"/>
<xsd:enumeration value="Deduplication Match Score"/>
<xsd:enumeration value="Deduplication Object Id"/>
<xsd:enumeration value="Description"/>
<xsd:enumeration value="Disable DataCleansing"/>
<xsd:enumeration value="Division"/>
<xsd:enumeration value="Domestic Ultimate DUNS"/>
<xsd:enumeration value="Dummy"/>
<xsd:enumeration value="EAI Sync Date"/>
<xsd:enumeration value="EAI Sync Error Text"/>
<xsd:enumeration value="EAI Sync Status Code"/>
<xsd:enumeration value="Email Address"/>
<xsd:enumeration value="Employee Here"/>
<xsd:enumeration value="Employees"/>
```

```
<xsd:enumeration value="Expertise"/>
<xsd:enumeration value="Explorer Label"/>
<xsd:enumeration value="Fax Number"/>
<xsd:enumeration value="Fiscal Year End"/>
<xsd:enumeration value="Freight Terms"/>
<xsd:enumeration value="Freight Terms Info"/>
<xsd:enumeration value="Full Address"/>
<xsd:enumeration value="GSA Flag"/>
<xsd:enumeration value="Global Ultimate DUNS"/>
<xsd:enumeration value="Goals"/>
<xsd:enumeration value="Group Type Code"/>
<xsd:enumeration value="Home Page"/>
<xsd:enumeration value="Industry Condition"/>
<xsd:enumeration value="Industry Trend"/>
<xsd:enumeration value="Integration Id"/>
<xsd:enumeration value="Internal Org Flag"/>
<xsd:enumeration value="Joined Synonym"/>
<xsd:enumeration value="Key Competitors"/>
<xsd:enumeration value="Language Code"/>
<xsd:enumeration value="Last Clnse Date"/>
<xsd:enumeration value="Last Manager Review Date"/>
<xsd:enumeration value="Last Review Manager Id"/>
<xsd:enumeration value="Line of Business"/>
<xsd:enumeration value="Location"/>
<xsd:enumeration value="Location Level"/>
<xsd:enumeration value="Main Address Flag"/>
<xsd:enumeration value="Main Fax Number"/>
```

```
<xsd:enumeration value="Main Phone Number"/>
<xsd:enumeration value="Managers Review"/>
<xsd:enumeration value="Marketing"/>
<xsd:enumeration value="Merge Sequence Number"/>
<xsd:enumeration value="Mission"/>
<xsd:enumeration value="Name"/>
<xsd:enumeration value="Name and Location"/>
<xsd:enumeration value="Not Manager Flag"/>
<xsd:enumeration value="Notes"/>
<xsd:enumeration value="Objectives"/>
<xsd:enumeration value="Organization Id"/>
<xsd:enumeration value="Organization Integration Id"/>
<xsd:enumeration value="Our Position"/>
<xsd:enumeration value="Outline Number"/>
<xsd:enumeration value="PO Approved Flag"/>
<xsd:enumeration value="PO Auto Approval Currency Code"/>
<xsd:enumeration value="PO Auto Approval Date"/>
<xsd:enumeration value="PO Auto Approval Limit"/>
<xsd:enumeration value="Parent Account Division"/>
<xsd:enumeration value="Parent Account Id"/>
<xsd:enumeration value="Parent Account Integration Id"/>
<xsd:enumeration value="Parent Account Location"/>
<xsd:enumeration value="Parent Account Location Level"/>
<xsd:enumeration value="Parent Account Name"/>
<xsd:enumeration value="Parent Account Region"/>
<xsd:enumeration value="Parent HQ DUNS"/>
<xsd:enumeration value="Partner Flag"/>
```

```
<xsd:enumeration value="Partners"/>
<xsd:enumeration value="Party Name"/>
<xsd:enumeration value="Party Type Code"/>
<xsd:enumeration value="Party UId"/>
<xsd:enumeration value="Philosophy"/>
<xsd:enumeration value="Phone Number"/>
<xsd:enumeration value="Position Integration Id"/>
<xsd:enumeration value="Postal Code"/>
<xsd:enumeration value="Price List"/>
<xsd:enumeration value="Price List End Date"/>
<xsd:enumeration value="Price List Id"/>
<xsd:enumeration value="Price List Integration Id"/>
<xsd:enumeration value="Price List Start Date"/>
<xsd:enumeration value="Primary Account City"/>
<xsd:enumeration value="Primary Account Country"/>
<xsd:enumeration value="Primary Account Postal Code"/>
<xsd:enumeration value="Primary Account State"/>
<xsd:enumeration value="Primary Account Street Address"/>
<xsd:enumeration value="Primary Address Id"/>
<xsd:enumeration value="Primary Assignment Denorm Flag"/>
<xsd:enumeration value="Primary Assignment Manual Flag"/>
<xsd:enumeration value="Primary Assignment System Flag"/>
<xsd:enumeration value="Primary Assignment Type"/>
<xsd:enumeration value="Primary Bill To Address Id"/>
<xsd:enumeration value="Primary Bill To City"/>
<xsd:enumeration value="Primary Bill To Country"/>
<xsd:enumeration value="Primary Bill To First Name"/>
```

```
                    <xsd:enumeration value="Primary Bill To Job Title"/>
                    <xsd:enumeration value="Primary Bill To Last Name"/>
                    <xsd:enumeration value="Primary Bill To Person Id"/>
                    <xsd:enumeration value="Primary Bill To Postal Code"/>
                    <xsd:enumeration value="Primary Bill To State"/>
                    <xsd:enumeration value="Primary Bill To Street Address"/>
                    <xsd:enumeration value="Primary Category Id"/>
                <xsd:enumeration value="Primary Fulfill InvLoc Integration Id"/>
                    <xsd:enumeration value="Primary Fulfillment InvLoc ID"/>
                    <xsd:enumeration value="Primary Fulfillment Inventory
Location"/>
                    <xsd:enumeration value="Primary Industry Id"/>
                    <xsd:enumeration value="Primary Organization"/>
                    <xsd:enumeration value="Primary Organization Id"/>
                    <xsd:enumeration value="Primary Payer Account"/>
                    <xsd:enumeration value="Primary Payer Account Id"/>
                    <xsd:enumeration value="Primary Position Id"/>
                    <xsd:enumeration value="Primary Service Agreement Id"/>
                    <xsd:enumeration value="Primary Ship To Address Id"/>
                    <xsd:enumeration value="Primary Ship To City"/>
                    <xsd:enumeration value="Primary Ship To Country"/>
                    <xsd:enumeration value="Primary Ship To First Name"/>
                    <xsd:enumeration value="Primary Ship To Job Title"/>
                    <xsd:enumeration value="Primary Ship To Last Name"/>
                    <xsd:enumeration value="Primary Ship To Person Id"/>
                    <xsd:enumeration value="Primary Ship To Postal Code"/>
                    <xsd:enumeration value="Primary Ship To State"/>
                    <xsd:enumeration value="Primary Ship To Street Address"/>
```

```
<xsd:enumeration value="Primary Synonym Id"/>
<xsd:enumeration value="Primary Territory Id"/>
<xsd:enumeration value="Primary Type Id"/>
<xsd:enumeration value="Profit"/>
<xsd:enumeration value="Project Bill Type"/>
<xsd:enumeration value="Project Comments"/>
<xsd:enumeration value="Project Fix Fee"/>
<xsd:enumeration value="Project Hour Limit"/>
<xsd:enumeration value="Project Id"/>
<xsd:enumeration value="Project Name"/>
<xsd:enumeration value="Project Percentage of Fee"/>
<xsd:enumeration value="Project Purchase Order"/>
<xsd:enumeration value="Project Relationship Type"/>
<xsd:enumeration value="Project Role"/>
<xsd:enumeration value="Prospect Flag"/>
<xsd:enumeration value="Province"/>
<xsd:enumeration value="Public"/>
<xsd:enumeration value="Reference Date"/>
<xsd:enumeration value="Reference Flag"/>
<xsd:enumeration value="Reference Stage"/>
<xsd:enumeration value="Region"/>
<xsd:enumeration value="Relationship Level"/>
<xsd:enumeration value="Relationship Type"/>
<xsd:enumeration value="Response Time"/>
<xsd:enumeration value="Revenue"/>
<xsd:enumeration value="Revenue Growth"/>
<xsd:enumeration value="Revision Number"/>
```

```
<xsd:enumeration value="Row Status"/>
<xsd:enumeration value="Row Status Asterisk"/>
<xsd:enumeration value="S-S Instance"/>
<xsd:enumeration value="S-S Instance Id"/>
<xsd:enumeration value="S-S Key Id"/>
<xsd:enumeration value="Service Calendar"/>
<xsd:enumeration value="Service Type"/>
<xsd:enumeration value="Ship Address Flag"/>
<xsd:enumeration value="Ship To City"/>
<xsd:enumeration value="Ship To Country"/>
<xsd:enumeration value="Ship To First Name"/>
<xsd:enumeration value="Ship To Job Title"/>
<xsd:enumeration value="Ship To Last Name"/>
<xsd:enumeration value="Ship To Postal Code"/>
<xsd:enumeration value="Ship To State"/>
<xsd:enumeration value="Ship To Street Address"/>
<xsd:enumeration value="Start Date"/>
<xsd:enumeration value="State"/>
<xsd:enumeration value="Strategies"/>
<xsd:enumeration value="Strategy"/>
<xsd:enumeration value="Street Address"/>
<xsd:enumeration value="Street Address 2"/>
<xsd:enumeration value="Success Factors"/>
<xsd:enumeration value="Synonym"/>
<xsd:enumeration value="Territory"/>
<xsd:enumeration value="Territory Id"/>
<xsd:enumeration value="Timestamp"/>
```

```
                    <xsd:enumeration value="Today"/>
                    <xsd:enumeration value="Total Potential Volume"/>
                <xsd:enumeration value="Total Potential Volume Currency Code"/>
                <xsd:enumeration value="Total Potential Volume Exchange Date"/>
                    <xsd:enumeration value="Type"/>
                    <xsd:enumeration value="Type MVF"/>
                    <xsd:enumeration value="VAT registration number"/>
                    <xsd:enumeration value="Value Proposition"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
      </xsd:sequence>
      <xsd:attribute name="boname" type="xsd:string" use="required"
fixed="Account"/>
      <xsd:attribute name="bcname" type="xsd:string" use="required"
fixed="Account"/>
     <xsd:attribute name="operation" use="required">
     <xsd:simpleType>
     <xsd:restriction base="xsd:string">
     <xsd:enumeration value="insert"/>
     <xsd:enumeration value="update"/>
     <xsd:enumeration value="delete"/>
     <xsd:enumeration value="query"/>
 </xsd:restriction>
</xsd:simpleType>
```

```
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Account Response Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xsd:element name="BusinessComponent">
    <xsd:complexType>
      <xsd:sequence>
<xsd:element name="Record" minOccurs="0" maxOccurs="unbounded" >
    <xsd:complexType>
        <xsd:sequence>
<xsd:element name="Field" minOccurs="0" maxOccurs="unbounded" >
    <xsd:complexType>
        <xsd:sequence/>
        <xsd:attribute name="name" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Account Competitors"/>
                    <xsd:enumeration value="Account Condition"/>
                    <xsd:enumeration value="Account Markets"/>
                 <xsd:enumeration value="Account Organization Integration Id"/>
                    <xsd:enumeration value="Account Products"/>
                    <xsd:enumeration value="Account Role"/>
                    <xsd:enumeration value="Account Status"/>
```

```
<xsd:enumeration value="Account Trend"/>
<xsd:enumeration value="Address Active Status"/>
<xsd:enumeration value="Address Id"/>
<xsd:enumeration value="Address Integration Id"/>
<xsd:enumeration value="Agreement End Date"/>
<xsd:enumeration value="Agreement Name"/>
<xsd:enumeration value="Agreement Start Date"/>
<xsd:enumeration value="Agreement Status"/>
<xsd:enumeration value="Algorithm Type"/>
<xsd:enumeration value="Alias"/>
<xsd:enumeration value="Annual Revenue"/>
<xsd:enumeration value="Assignment Area Code"/>
<xsd:enumeration value="Assignment Country Code"/>
<xsd:enumeration value="Assignment Excluded"/>
<xsd:enumeration value="Assignment Manual Flag"/>
<xsd:enumeration value="Back Office Distribution Channel"/>
<xsd:enumeration value="Back Office Order Query End  Dt"/>
<xsd:enumeration value="Back Office Order Query Start  Dt"/>
<xsd:enumeration value="Back Office Sales Area Division Code"/>
<xsd:enumeration value="Back Office Sales Organization"/>
<xsd:enumeration value="Bill Address Flag"/>
<xsd:enumeration value="Bill To City"/>
<xsd:enumeration value="Bill To Country"/>
<xsd:enumeration value="Bill To First Name"/>
<xsd:enumeration value="Bill To Id"/>
<xsd:enumeration value="Bill To Job Title"/>
<xsd:enumeration value="Bill To Last Name"/>
```

```
<xsd:enumeration value="Bill To Postal Code"/>
<xsd:enumeration value="Bill To State"/>
<xsd:enumeration value="Bill To Street Address"/>
<xsd:enumeration value="Block Credit Flag"/>
<xsd:enumeration value="Business Profile"/>
<xsd:enumeration value="CSN"/>
<xsd:enumeration value="City"/>
<xsd:enumeration value="City State"/>
<xsd:enumeration value="Competitor"/>
<xsd:enumeration value="Country"/>
<xsd:enumeration value="County"/>
<xsd:enumeration value="Credit Control Area Code"/>
<xsd:enumeration value="Credit Currency Code"/>
<xsd:enumeration value="Credit Limit Amount"/>
<xsd:enumeration value="Credit Profile Id"/>
<xsd:enumeration value="Culture"/>
<xsd:enumeration value="Currency Code"/>
<xsd:enumeration value="Current Volume"/>
<xsd:enumeration value="Current Volume Currency Code"/>
<xsd:enumeration value="Current Volume Exchange Date"/>
<xsd:enumeration value="Customer Account Group"/>
<xsd:enumeration value="DNBReport"/>
<xsd:enumeration value="DUNS Intcode"/>
<xsd:enumeration value="DUNS Number"/>
<xsd:enumeration value="Date Formed"/>
<xsd:enumeration value="DeDup Key Modification Date"/>
<xsd:enumeration value="DeDup Key Update"/>
```

```
<xsd:enumeration value="DeDup Keys"/>
<xsd:enumeration value="DeDup Last Match Date"/>
<xsd:enumeration value="DeDup Token"/>
<xsd:enumeration value="Deduplication Match Score"/>
<xsd:enumeration value="Deduplication Object Id"/>
<xsd:enumeration value="Description"/>
<xsd:enumeration value="Disable DataCleansing"/>
<xsd:enumeration value="Division"/>
<xsd:enumeration value="Domestic Ultimate DUNS"/>
<xsd:enumeration value="Dummy"/>
<xsd:enumeration value="EAI Sync Date"/>
<xsd:enumeration value="EAI Sync Error Text"/>
<xsd:enumeration value="EAI Sync Status Code"/>
<xsd:enumeration value="Email Address"/>
<xsd:enumeration value="Employee Here"/>
<xsd:enumeration value="Employees"/>
<xsd:enumeration value="Expertise"/>
<xsd:enumeration value="Explorer Label"/>
<xsd:enumeration value="Fax Number"/>
<xsd:enumeration value="Fiscal Year End"/>
<xsd:enumeration value="Freight Terms"/>
<xsd:enumeration value="Freight Terms Info"/>
<xsd:enumeration value="Full Address"/>
<xsd:enumeration value="GSA Flag"/>
<xsd:enumeration value="Global Ultimate DUNS"/>
<xsd:enumeration value="Goals"/>
<xsd:enumeration value="Group Type Code"/>
```

```
<xsd:enumeration value="Home Page"/>
<xsd:enumeration value="Industry Condition"/>
<xsd:enumeration value="Industry Trend"/>
<xsd:enumeration value="Integration Id"/>
<xsd:enumeration value="Internal Org Flag"/>
<xsd:enumeration value="Joined Synonym"/>
<xsd:enumeration value="Key Competitors"/>
<xsd:enumeration value="Language Code"/>
<xsd:enumeration value="Last Clnse Date"/>
<xsd:enumeration value="Last Manager Review Date"/>
<xsd:enumeration value="Last Review Manager Id"/>
<xsd:enumeration value="Line of Business"/>
<xsd:enumeration value="Location"/>
<xsd:enumeration value="Location Level"/>
<xsd:enumeration value="Main Address Flag"/>
<xsd:enumeration value="Main Fax Number"/>
<xsd:enumeration value="Main Phone Number"/>
<xsd:enumeration value="Managers Review"/>
<xsd:enumeration value="Marketing"/>
<xsd:enumeration value="Merge Sequence Number"/>
<xsd:enumeration value="Mission"/>
<xsd:enumeration value="Name"/>
<xsd:enumeration value="Name and Location"/>
<xsd:enumeration value="Not Manager Flag"/>
<xsd:enumeration value="Notes"/>
<xsd:enumeration value="Objectives"/>
<xsd:enumeration value="Organization Id"/>
```

```
<xsd:enumeration value="Organization Integration Id"/>
<xsd:enumeration value="Our Position"/>
<xsd:enumeration value="Outline Number"/>
<xsd:enumeration value="PO Approved Flag"/>
<xsd:enumeration value="PO Auto Approval Currency Code"/>
<xsd:enumeration value="PO Auto Approval Date"/>
<xsd:enumeration value="PO Auto Approval Limit"/>
<xsd:enumeration value="Parent Account Division"/>
<xsd:enumeration value="Parent Account Id"/>
<xsd:enumeration value="Parent Account Integration Id"/>
<xsd:enumeration value="Parent Account Location"/>
<xsd:enumeration value="Parent Account Location Level"/>
<xsd:enumeration value="Parent Account Name"/>
<xsd:enumeration value="Parent Account Region"/>
<xsd:enumeration value="Parent HQ DUNS"/>
<xsd:enumeration value="Partner Flag"/>
<xsd:enumeration value="Partners"/>
<xsd:enumeration value="Party Name"/>
<xsd:enumeration value="Party Type Code"/>
<xsd:enumeration value="Party UId"/>
<xsd:enumeration value="Philosophy"/>
<xsd:enumeration value="Phone Number"/>
<xsd:enumeration value="Position Integration Id"/>
<xsd:enumeration value="Postal Code"/>
<xsd:enumeration value="Price List"/>
<xsd:enumeration value="Price List End Date"/>
<xsd:enumeration value="Price List Id"/>
```

```
                    <xsd:enumeration value="Price List Integration Id"/>
                    <xsd:enumeration value="Price List Start Date"/>
                    <xsd:enumeration value="Primary Account City"/>
                    <xsd:enumeration value="Primary Account Country"/>
                    <xsd:enumeration value="Primary Account Postal Code"/>
                    <xsd:enumeration value="Primary Account State"/>
                    <xsd:enumeration value="Primary Account Street Address"/>
                    <xsd:enumeration value="Primary Address Id"/>
                    <xsd:enumeration value="Primary Assignment Denorm Flag"/>
                    <xsd:enumeration value="Primary Assignment Manual Flag"/>
                    <xsd:enumeration value="Primary Assignment System Flag"/>
                    <xsd:enumeration value="Primary Assignment Type"/>
                    <xsd:enumeration value="Primary Bill To Address Id"/>
                    <xsd:enumeration value="Primary Bill To City"/>
                    <xsd:enumeration value="Primary Bill To Country"/>
                    <xsd:enumeration value="Primary Bill To First Name"/>
                    <xsd:enumeration value="Primary Bill To Job Title"/>
                    <xsd:enumeration value="Primary Bill To Last Name"/>
                    <xsd:enumeration value="Primary Bill To Person Id"/>
                    <xsd:enumeration value="Primary Bill To Postal Code"/>
                    <xsd:enumeration value="Primary Bill To State"/>
                    <xsd:enumeration value="Primary Bill To Street Address"/>
                    <xsd:enumeration value="Primary Category Id"/>
                <xsd:enumeration value="Primary Fulfill InvLoc Integration Id"/>
                    <xsd:enumeration value="Primary Fulfillment InvLoc ID"/>
                    <xsd:enumeration value="Primary Fulfillment Inventory
Location"/>
                    <xsd:enumeration value="Primary Industry Id"/>
```

```
<xsd:enumeration value="Primary Organization"/>
<xsd:enumeration value="Primary Organization Id"/>
<xsd:enumeration value="Primary Payer Account"/>
<xsd:enumeration value="Primary Payer Account Id"/>
<xsd:enumeration value="Primary Position Id"/>
<xsd:enumeration value="Primary Service Agreement Id"/>
<xsd:enumeration value="Primary Ship To Address Id"/>
<xsd:enumeration value="Primary Ship To City"/>
<xsd:enumeration value="Primary Ship To Country"/>
<xsd:enumeration value="Primary Ship To First Name"/>
<xsd:enumeration value="Primary Ship To Job Title"/>
<xsd:enumeration value="Primary Ship To Last Name"/>
<xsd:enumeration value="Primary Ship To Person Id"/>
<xsd:enumeration value="Primary Ship To Postal Code"/>
<xsd:enumeration value="Primary Ship To State"/>
<xsd:enumeration value="Primary Ship To Street Address"/>
<xsd:enumeration value="Primary Synonym Id"/>
<xsd:enumeration value="Primary Territory Id"/>
<xsd:enumeration value="Primary Type Id"/>
<xsd:enumeration value="Profit"/>
<xsd:enumeration value="Project Bill Type"/>
<xsd:enumeration value="Project Comments"/>
<xsd:enumeration value="Project Fix Fee"/>
<xsd:enumeration value="Project Hour Limit"/>
<xsd:enumeration value="Project Id"/>
<xsd:enumeration value="Project Name"/>
<xsd:enumeration value="Project Percentage of Fee"/>
```

```
<xsd:enumeration value="Project Purchase Order"/>
<xsd:enumeration value="Project Relationship Type"/>
<xsd:enumeration value="Project Role"/>
<xsd:enumeration value="Prospect Flag"/>
<xsd:enumeration value="Province"/>
<xsd:enumeration value="Public"/>
<xsd:enumeration value="Reference Date"/>
<xsd:enumeration value="Reference Flag"/>
<xsd:enumeration value="Reference Stage"/>
<xsd:enumeration value="Region"/>
<xsd:enumeration value="Relationship Level"/>
<xsd:enumeration value="Relationship Type"/>
<xsd:enumeration value="Response Time"/>
<xsd:enumeration value="Revenue"/>
<xsd:enumeration value="Revenue Growth"/>
<xsd:enumeration value="Revision Number"/>
<xsd:enumeration value="Row Status"/>
<xsd:enumeration value="Row Status Asterisk"/>
<xsd:enumeration value="S-S Instance"/>
<xsd:enumeration value="S-S Instance Id"/>
<xsd:enumeration value="S-S Key Id"/>
<xsd:enumeration value="Service Calendar"/>
<xsd:enumeration value="Service Type"/>
<xsd:enumeration value="Ship Address Flag"/>
<xsd:enumeration value="Ship To City"/>
<xsd:enumeration value="Ship To Country"/>
<xsd:enumeration value="Ship To First Name"/>
```

```
                <xsd:enumeration value="Ship To Job Title"/>
                <xsd:enumeration value="Ship To Last Name"/>
                <xsd:enumeration value="Ship To Postal Code"/>
                <xsd:enumeration value="Ship To State"/>
                <xsd:enumeration value="Ship To Street Address"/>
                <xsd:enumeration value="Start Date"/>
                <xsd:enumeration value="State"/>
                <xsd:enumeration value="Strategies"/>
                <xsd:enumeration value="Strategy"/>
                <xsd:enumeration value="Street Address"/>
                <xsd:enumeration value="Street Address 2"/>
                <xsd:enumeration value="Success Factors"/>
                <xsd:enumeration value="Synonym"/>
                <xsd:enumeration value="Territory"/>
                <xsd:enumeration value="Territory Id"/>
                <xsd:enumeration value="Timestamp"/>
                <xsd:enumeration value="Today"/>
                <xsd:enumeration value="Total Potential Volume"/>
            <xsd:enumeration value="Total Potential Volume Currency Code"/>
            <xsd:enumeration value="Total Potential Volume Exchange Date"/>
                <xsd:enumeration value="Type"/>
                <xsd:enumeration value="Type MVF"/>
                <xsd:enumeration value="VAT registration number"/>
                <xsd:enumeration value="Value Proposition"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
```

```
        <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
      </xsd:sequence>
      <xsd:attribute name="status" use="required">
      <xsd:simpleType>
      <xsd:restriction base="xsd:string">
      <xsd:enumeration value="success"/>
      <xsd:enumeration value="failure"/>
      <xsd:restriction>
      </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="reason" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Sample XML for Account Add Request

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
 edited with XML Spy v4.4 U (http://www.xmlspy.com) by jerry (Information
Builders Inc.)
-->
- <!--
Sample XML file generated by XML Spy v4.4 U (http://www.xmlspy.com)
```

```
-->
<BusinessComponent  boname="Account" bcname="Account" operation="insert">
<Field name="Currency Code" value="USD" />
<Field name="Name" value="JDB 5" />
</BusinessComponent>
```

Sample XML for Account Add Response

```
<?xml version="1.0" encoding="UTF-8" ?>
<BusinessComponent status="success" reason="" />
```

Sample XML for Account Delete Request

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
 edited with XML Spy v4.4 U (http://www.xmlspy.com)
-->
- <!--
Sample XML file generated by XML Spy v4.4 U (http://www.xmlspy.com)
-->
 <BusinessComponent boname="Account" bcname="Account" operation="delete">
<Select name="Name" value="JD*" />
</BusinessComponent>
```

Sample XML for Account Delete Response

```
<?xml version="1.0" encoding="UTF-8" ?>
<BusinessComponent status="success" reason="" />
```

Sample XML for Account Query Request

```
<BusinessComponent boname="Account" bcname="Account" operation="query">
<Select name="Name" value="Ja*" />
```

```
<Field name="Name" value="" />
<Field name="City" value="" />
<Field name="Street Address" value="" />
<Field name="Type" value="" />
<Field name="Account Status" value="Active" />
</BusinessComponent>
```

Sample XML for Account Query Response

```
<?xml version="1.0" encoding="UTF-8" ?>
 <BusinessComponent status="success" reason="">
 <Record>
<Field name="Name" value="Java Data Bean Account3" />
<Field name="City" value="Norwood City3" />
<Field name="Street Address" value="201 Wickham Way3" />
<Field name="Type" value="" />
<Field name="Account Status" value="" />
</Record>
 <Record>
<Field name="Name" value="Java Data Bean Account4" />
<Field name="City" value="Norwood City4" />
<Field name="Street Address" value="201 Wickham Way4" />
<Field name="Type" value="" />
<Field name="Account Status" value="" />
</Record>
 <Record>
<Field name="Name" value="Java Data Bean Account5" />
<Field name="City" value="Norwood City5" />
<Field name="Street Address" value="201 Wickham Way5" />
```

```
<Field name="Type" value="" />
<Field name="Account Status" value="" />
</Record>
 <Record>
<Field name="Name" value="Java Data Bean Account6" />
<Field name="City" value="Norwood City6" />
<Field name="Street Address" value="201 Wickham Way6" />
<Field name="Type" value="OEM" />
<Field name="Account Status" value="Active" />
</Record>
</BusinessComponent>
```

PGMAVV Account Add Request Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xsd:element name="BusinessService">
    <xsd:complexType>
      <xsd:sequence>
<xsd:element name="Param" minOccurs="0" maxOccurs="unbounded" >
    <xsd:complexType>
        <xsd:sequence/>
        <xsd:attribute name="name" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Param1"/>
                    <xsd:enumeration value="Param2"/>
                </xsd:restriction>
```

```
            </xsd:simpleType>

        </xsd:attribute>

        <xsd:attribute name="value" type="xsd:string" use="required" />

    </xsd:complexType>

</xsd:element>

      </xsd:sequence>

      <xsd:attribute name="servicename" type="xsd:string" use="required"
fixed="PGMAVV Calculator"/>

      <xsd:attribute name="methodname" type="xsd:string" use="required"
fixed="Add"/>

    </xsd:complexType>

  </xsd:element>

</xsd:schema>
```

PGMAVV Account Add Response Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <xsd:element name="BusinessService">

    <xsd:complexType>

      <xsd:sequence>

<xsd:element name="Result" minOccurs="0" maxOccurs="unbounded" >

    <xsd:complexType>

        <xsd:sequence/>

        <xsd:attribute name="name" use="required">

            <xsd:simpleType>

                <xsd:restriction base="xsd:string">

                    <xsd:enumeration value="Value"/>

                    <xsd:enumeration value="Operator"/>
```

```
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
      </xsd:sequence>
      <xsd:attribute name="status" use="required">
      <xsd:simpleType>
      <xsd:restriction base="xsd:string">
      <xsd:enumeration value="success"/>
      <xsd:enumeration value="failure"/>
      </xsd:restriction>
      </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="reason" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Sample XML for PGMAVV Account Add Request

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Sample XML file generated by XML Spy v4.4 U (http://www.xmlspy.com)
-->
 <BusinessService servicename="PGMAVV Calculator" methodname="Add">
<Param name="Param1" value="1" />
<Param name="Param2" value="3" />
```

```
</BusinessService>
```

Sample XML for PGMAVV Account Add Response

```
<?xml version="1.0" encoding="UTF-8" ?>

<BusinessService status="success" reason="">

<Result name="Value" value="4" />

<Result name="Operator" value="+" />

</BusinessService>
```

Sample Account Service Request Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<!--

  [XDR-XSD] This schema automatically updated from an IE5-compatible XDR schema
to W3C

  XML Schema by xdr-xsd-converter.xslt 1.2, available from Microsoft.

  Contact jmarsh@microsoft.com.


  Note: if a local copy of XMLSchema.dtd and datatypes.dtd are not

  available, use the official location of
"http://www.w3.org/2000/10/XMLSchema.dtd"

  for the system id.

-->

<!-- Copyright (C) 2001, Siebel Systems, L.P., All rights reserved. -->

<!-- Siebel XDR Generation -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>Schema name: SiebelMessage</xsd:documentation>
  </xsd:annotation>

<!-- [XDR-XSD] "SiebelMessage" element  -->
<xsd:element name="SiebelMessage">
  <xsd:complexType>
```

```
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="ListOfSampleAccount" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="MessageId"/>
      <xsd:attribute name="MessageType" use="required" fixed="Integration
Object"/>
     <xsd:attribute name="IntObjectName" use="required" fixed="Sample Account"/>
      <xsd:attribute name="IntObjectFormat"/>
  </xsd:complexType>
</xsd:element>

<!-- Shared Element List.  These elements are guarenteed -->
<!-- to have the same datatype, length, precision, and scale.-->
<!-- [XDR-XSD] "Type" element  -->
<xsd:element name="Type">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Name" element  -->
<xsd:element name="Name">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Location" element  -->
<xsd:element name="Location">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Division" element  -->
<xsd:element name="Division">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "Description" element  -->
<xsd:element name="Description">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "CurrencyCode" element  -->
<xsd:element name="CurrencyCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="15"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "StreetAddress" element  -->
<xsd:element name="StreetAddress">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="200"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "State" element  -->
<xsd:element name="State">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PostalCode" element  -->
<xsd:element name="PostalCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Country" element  -->
<xsd:element name="Country">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "City" element  -->
<xsd:element name="City">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AddressName" element  -->
<xsd:element name="AddressName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Organization" element  -->
<xsd:element name="Organization">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "IntegrationId" element  -->
<xsd:element name="IntegrationId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ListOfSampleAccount" element  -->
<xsd:element name="ListOfSampleAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Account" maxOccurs="unbounded" minOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

```
<!-- [XDR-XSD] "Account" element  -->
<xsd:element name="Account">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="System" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Culture" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CurrencyCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Description" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Division" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="HomePage" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LineofBusiness" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Location" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Name" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ProjectName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Type" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfAccount_Organization" maxOccurs="1"
minOccurs="0"/>
      <xsd:element ref="ListOfAccountAttachment" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfBusinessAddress" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfContact" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfSubAccount" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "System" element  -->
<xsd:element name="System" type="xsd:string"/>
<!-- [XDR-XSD] "Culture" element  -->
<xsd:element name="Culture">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "HomePage" element  -->
<xsd:element name="HomePage">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "LineofBusiness" element  -->
<xsd:element name="LineofBusiness">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ProjectName" element  -->
<xsd:element name="ProjectName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ListOfAccount_Organization" element  -->
<xsd:element name="ListOfAccount_Organization">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Account_Organization" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Account_Organization" element  -->
<xsd:element name="Account_Organization">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Organization" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="IsPrimaryMVG">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
```

```
     <xsd:maxLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfAccountAttachment" element  -->
<xsd:element name="ListOfAccountAttachment">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
     <xsd:element ref="AccountAttachment" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "AccountAttachment" element  -->
<xsd:element name="AccountAttachment">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="AccntFileExt" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AccntFileName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Comment" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AttachmentId" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
```

```
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfBusinessAddress" element  -->
<xsd:element name="ListOfBusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="BusinessAddress" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "BusinessAddress" element  -->
<xsd:element name="BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="AddressName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="City" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Country" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="FaxNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PhoneNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PostalCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Province" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="State" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

```
<!-- [XDR-XSD] "ListOfContact" element  -->
<xsd:element name="ListOfContact">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Contact" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Contact" element  -->
<xsd:element name="Contact">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="CellularPhone" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Comment2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="FirstName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="HomePhone" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="JobTitle" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LastName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MiddleName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Organization" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PersonalContact" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfContact_BusinessAddress" maxOccurs="1"
minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfSubAccount" element  -->
<xsd:element name="ListOfSubAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="SubAccount" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
```

```
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "SubAccount" element  -->
<xsd:element name="SubAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Id" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Created" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Updated" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ConflictId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ModId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AccountStatus2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Alias2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AssignmentAreaCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AssignmentCountryCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CSN2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Competitor" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CurrencyCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Description" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Division" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="GroupTypeCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="InternalOrgFlag" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Location" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LocationLevel" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MainFaxNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MainPhoneNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Name" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountDivision" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountLocation" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountLocationLevel" maxOccurs="1"
minOccurs="0"/>
      <xsd:element ref="ParentAccountName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountRegion" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyTypeCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyUId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryIndustryId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryOrganizationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryPositionId" maxOccurs="1" minOccurs="0"/>
     <xsd:element ref="PrimaryServiceAgreementId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimarySynonymId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Region" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SInstance" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SInstanceId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SKeyId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Type" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
```

```
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "AccntFileExt" element  -->
<xsd:element name="AccntFileExt">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AccntFileName" element  -->
<xsd:element name="AccntFileName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="220"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Comment" element  -->
<xsd:element name="Comment" type="xsd:string"/>
<!-- [XDR-XSD] "AttachmentId" element  -->
<xsd:element name="AttachmentId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "FaxNumber" element  -->
<xsd:element name="FaxNumber" type="xsd:string"/>
<!-- [XDR-XSD] "PhoneNumber" element  -->
<xsd:element name="PhoneNumber" type="xsd:string"/>
```

```
<!-- [XDR-XSD] "Province" element  -->
<xsd:element name="Province">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "CellularPhone" element  -->
<xsd:element name="CellularPhone" type="xsd:string"/>
<!-- [XDR-XSD] "Comment2" element  -->
<xsd:element name="Comment2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "FirstName" element  -->
<xsd:element name="FirstName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "HomePhone" element  -->
<xsd:element name="HomePhone" type="xsd:string"/>
<!-- [XDR-XSD] "JobTitle" element  -->
<xsd:element name="JobTitle">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="75"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "LastName" element  -->
<xsd:element name="LastName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "MiddleName" element  -->
<xsd:element name="MiddleName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PersonalContact" element  -->
<xsd:element name="PersonalContact" type="xsd:string"/>
<!-- [XDR-XSD] "ListOfContact_BusinessAddress" element  -->
<xsd:element name="ListOfContact_BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Contact_BusinessAddress" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Contact_BusinessAddress" element  -->
<xsd:element name="Contact_BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="City" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Country" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PostalCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="State" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AddressName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
      <xsd:attribute name="IsPrimaryMVG">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
      <xsd:attribute name="searchspec">
```

```
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Id" element  -->
<xsd:element name="Id">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Created" element  -->
<xsd:element name="Created">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Updated" element  -->
<xsd:element name="Updated">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ConflictId" element  -->
<xsd:element name="ConflictId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ModId" element  -->
<xsd:element name="ModId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AccountStatus2" element  -->
<xsd:element name="AccountStatus2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Alias2" element  -->
<xsd:element name="Alias2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AssignmentAreaCode" element  -->
<xsd:element name="AssignmentAreaCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="5"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AssignmentCountryCode" element  -->
<xsd:element name="AssignmentCountryCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="5"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "CSN2" element  -->
<xsd:element name="CSN2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "Competitor" element  -->
<xsd:element name="Competitor" type="xsd:string"/>
<!-- [XDR-XSD] "GroupTypeCode" element  -->
<xsd:element name="GroupTypeCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "InternalOrgFlag" element  -->
<xsd:element name="InternalOrgFlag" type="xsd:string"/>
<!-- [XDR-XSD] "LocationLevel" element  -->
<xsd:element name="LocationLevel">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "MainFaxNumber" element  -->
<xsd:element name="MainFaxNumber" type="xsd:string"/>
<!-- [XDR-XSD] "MainPhoneNumber" element  -->
<xsd:element name="MainPhoneNumber" type="xsd:string"/>
<!-- [XDR-XSD] "ParentAccountDivision" element  -->
<xsd:element name="ParentAccountDivision">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountId" element  -->
<xsd:element name="ParentAccountId" type="xsd:string"/>
<!-- [XDR-XSD] "ParentAccountLocation" element  -->
<xsd:element name="ParentAccountLocation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountLocationLevel" element  -->
<xsd:element name="ParentAccountLocationLevel">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
```

```
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountName" element  -->
<xsd:element name="ParentAccountName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountRegion" element  -->
<xsd:element name="ParentAccountRegion">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyName" element  -->
<xsd:element name="PartyName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyTypeCode" element  -->
<xsd:element name="PartyTypeCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyUId" element  -->
<xsd:element name="PartyUId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "PrimaryIndustryId" element  -->
<xsd:element name="PrimaryIndustryId" type="xsd:string"/>
<!-- [XDR-XSD] "PrimaryOrganizationId" element  -->
<xsd:element name="PrimaryOrganizationId" type="xsd:string"/>
<!-- [XDR-XSD] "PrimaryPositionId" element  -->
<xsd:element name="PrimaryPositionId" type="xsd:string"/>
<!-- [XDR-XSD] "PrimaryServiceAgreementId" element  -->
<xsd:element name="PrimaryServiceAgreementId" type="xsd:string"/>
<!-- [XDR-XSD] "PrimarySynonymId" element  -->
<xsd:element name="PrimarySynonymId" type="xsd:string"/>
<!-- [XDR-XSD] "Region" element  -->
<xsd:element name="Region">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "S-SInstance" element  -->
<xsd:element name="S-SInstance">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="75"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "S-SInstanceId" element  -->
<xsd:element name="S-SInstanceId" type="xsd:string"/>
<!-- [XDR-XSD] "S-SKeyId" element  -->
<xsd:element name="S-SKeyId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "StreetAddress2" element  -->
<xsd:element name="StreetAddress2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] XDR datatype derivations -->
</xsd:schema>
```

Sample Account Service Response Schema

```
<?xml version="1.0" encoding="UTF-8"?><xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"><xsd:element
name="emitStatus"><xsd:complexType><xsd:all><xsd:element name="protocol"
type="xsd:string" minOccurs="0"/><xsd:element name="status" type="xsd:string"
minOccurs="0"/><xsd:element name="parms" type="xsd:string"
minOccurs="0"/><xsd:element name="native" type="xsd:string"
minOccurs="0"/><xsd:element name="text" type="xsd:string"
minOccurs="0"/><xsd:element name="name" type="xsd:string"
minOccurs="0"/><xsd:element name="retries" type="xsd:string"
minOccurs="0"/><xsd:element name="timestamp" type="xsd:string"
minOccurs="0"/></xsd:all></xsd:complexType></xsd:element></xsd:schema>
```

Sample Event Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  [XDR-XSD] This schema automatically updated from an IE5-compatible XDR
  schema to W3C XML Schema by xdr-xsd-converter.xslt 1.2, available from
  Microsoft.
  Contact jmarsh@microsoft.com.

  Note: if a local copy of XMLSchema.dtd and datatypes.dtd are not
  available, use the official location of
"http://www.w3.org/2000/10/XMLSchema.dtd"
  for the system id.
-->
<!-- Copyright (C) 2001, Siebel Systems, L.P., All rights reserved. -->
<!-- Siebel XDR Generation -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xsd:annotation>
<xsd:documentation>Schema name: SiebelMessage</xsd:documentation>
</xsd:annotation>

<!-- [XDR-XSD] "SiebelMessage" element  -->
<xsd:element name="SiebelMessage">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="ListOfSampleAccount" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="MessageId"/>
        <xsd:attribute name="MessageType" use="required" fixed="Integration
Object"/>
        <xsd:attribute name="IntObjectName" use="required" fixed="Sample
Account"/>
```

```
        <xsd:attribute name="IntObjectFormat"/>
  </xsd:complexType>
</xsd:element>
<!-- Shared Element List.  These elements are guarenteed -->
<!-- to have the same datatype, length, precision, and scale.-->

<!-- [XDR-XSD] "Type" element  -->
<xsd:element name="Type">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Name" element  -->
<xsd:element name="Name">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Location" element  -->
<xsd:element name="Location">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Division" element  -->
<xsd:element name="Division">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Description" element  -->
<xsd:element name="Description">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "CurrencyCode" element  -->
<xsd:element name="CurrencyCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="15"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "StreetAddress" element  -->
<xsd:element name="StreetAddress">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="200"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "State" element  -->
<xsd:element name="State">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PostalCode" element  -->
<xsd:element name="PostalCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Country" element  -->
<xsd:element name="Country">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "City" element  -->
<xsd:element name="City">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AddressName" element  -->
<xsd:element name="AddressName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Organization" element  -->
<xsd:element name="Organization">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "IntegrationId" element  -->
<xsd:element name="IntegrationId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ListOfSampleAccount" element  -->
<xsd:element name="ListOfSampleAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Account" maxOccurs="unbounded" minOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Account" element  -->
<xsd:element name="Account">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="System" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Culture" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CurrencyCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Description" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Division" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="HomePage" maxOccurs="1" minOccurs="0"/>
```

```
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LineofBusiness" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Location" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Name" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ProjectName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Type" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfAccount_Organization" maxOccurs="1"
minOccurs="0"/>
      <xsd:element ref="ListOfAccountAttachment" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfBusinessAddress" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfContact" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfSubAccount" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "System" element  -->
<xsd:element name="System" type="xsd:string"/>

<!-- [XDR-XSD] "Culture" element  -->
<xsd:element name="Culture">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "HomePage" element  -->
<xsd:element name="HomePage">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "LineofBusiness" element  -->
<xsd:element name="LineofBusiness">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ProjectName" element  -->
<xsd:element name="ProjectName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ListOfAccount_Organization" element  -->
<xsd:element name="ListOfAccount_Organization">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Account_Organization" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Account_Organization" element  -->
<xsd:element name="Account_Organization">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Organization" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="IsPrimaryMVG">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfAccountAttachment" element  -->
<xsd:element name="ListOfAccountAttachment">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
     <xsd:element ref="AccountAttachment" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "AccountAttachment" element  -->
<xsd:element name="AccountAttachment">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="AccntFileExt" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AccntFileName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Comment" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AttachmentId" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfBusinessAddress" element  -->
<xsd:element name="ListOfBusinessAddress">
```

```
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="BusinessAddress" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "BusinessAddress" element  -->
<xsd:element name="BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="AddressName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="City" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Country" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="FaxNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PhoneNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PostalCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Province" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="State" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfContact" element  -->
<xsd:element name="ListOfContact">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Contact" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

```
<!-- [XDR-XSD] "Contact" element  -->
<xsd:element name="Contact">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="CellularPhone" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Comment2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="FirstName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="HomePhone" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="JobTitle" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LastName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MiddleName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Organization" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PersonalContact" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ListOfContact_BusinessAddress" maxOccurs="1"
minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "ListOfSubAccount" element  -->
<xsd:element name="ListOfSubAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="SubAccount" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "SubAccount" element  -->
<xsd:element name="SubAccount">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Id" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Created" maxOccurs="1" minOccurs="0"/>
```

```
      <xsd:element ref="Updated" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ConflictId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ModId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AccountStatus2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Alias2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AssignmentAreaCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AssignmentCountryCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CSN2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Competitor" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="CurrencyCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Description" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Division" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="GroupTypeCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="InternalOrgFlag" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Location" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="LocationLevel" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MainFaxNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="MainPhoneNumber" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Name" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountDivision" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountLocation" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountLocationLevel" maxOccurs="1"
minOccurs="0"/>
      <xsd:element ref="ParentAccountName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="ParentAccountRegion" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyTypeCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PartyUId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryIndustryId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryOrganizationId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimaryPositionId" maxOccurs="1" minOccurs="0"/>
     <xsd:element ref="PrimaryServiceAgreementId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PrimarySynonymId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Region" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SInstance" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SInstanceId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="S-SKeyId" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Type" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
```

```
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "AccntFileExt" element  -->
<xsd:element name="AccntFileExt">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AccntFileName" element  -->
<xsd:element name="AccntFileName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="220"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Comment" element  -->
<xsd:element name="Comment" type="xsd:string"/>

<!-- [XDR-XSD] "AttachmentId" element  -->
<xsd:element name="AttachmentId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "FaxNumber" element  -->
<xsd:element name="FaxNumber" type="xsd:string"/>

<!-- [XDR-XSD] "PhoneNumber" element  -->
<xsd:element name="PhoneNumber" type="xsd:string"/>

<!-- [XDR-XSD] "Province" element  -->
<xsd:element name="Province">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "CellularPhone" element  -->
<xsd:element name="CellularPhone" type="xsd:string"/>

<!-- [XDR-XSD] "Comment2" element  -->
<xsd:element name="Comment2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "FirstName" element  -->
<xsd:element name="FirstName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "HomePhone" element  -->
<xsd:element name="HomePhone" type="xsd:string"/>

<!-- [XDR-XSD] "JobTitle" element  -->
<xsd:element name="JobTitle">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="75"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "LastName" element  -->
<xsd:element name="LastName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "MiddleName" element  -->
<xsd:element name="MiddleName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PersonalContact" element  -->
<xsd:element name="PersonalContact" type="xsd:string"/>

<!-- [XDR-XSD] "ListOfContact_BusinessAddress" element  -->
<xsd:element name="ListOfContact_BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="Contact_BusinessAddress" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Contact_BusinessAddress" element  -->
<xsd:element name="Contact_BusinessAddress">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element ref="City" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="Country" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="PostalCode" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="State" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="StreetAddress2" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="AddressName" maxOccurs="1" minOccurs="0"/>
      <xsd:element ref="IntegrationId" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
        <xsd:attribute name="IsPrimaryMVG">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="operation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
        <xsd:attribute name="searchspec">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="250"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:attribute>
  </xsd:complexType>
</xsd:element>

<!-- [XDR-XSD] "Id" element  -->
<xsd:element name="Id">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Created" element  -->
<xsd:element name="Created">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Updated" element  -->
<xsd:element name="Updated">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ConflictId" element  -->
<xsd:element name="ConflictId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ModId" element  -->
<xsd:element name="ModId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "AccountStatus2" element  -->
<xsd:element name="AccountStatus2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Alias2" element  -->
<xsd:element name="Alias2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AssignmentAreaCode" element  -->
<xsd:element name="AssignmentAreaCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="5"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "AssignmentCountryCode" element  -->
<xsd:element name="AssignmentCountryCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="5"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "CSN2" element  -->
<xsd:element name="CSN2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "Competitor" element  -->
<xsd:element name="Competitor" type="xsd:string"/>

<!-- [XDR-XSD] "GroupTypeCode" element  -->
<xsd:element name="GroupTypeCode">
```

```
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "InternalOrgFlag" element  -->
<xsd:element name="InternalOrgFlag" type="xsd:string"/>

<!-- [XDR-XSD] "LocationLevel" element  -->
<xsd:element name="LocationLevel">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "MainFaxNumber" element  -->
<xsd:element name="MainFaxNumber" type="xsd:string"/>

<!-- [XDR-XSD] "MainPhoneNumber" element  -->
<xsd:element name="MainPhoneNumber" type="xsd:string"/>

<!-- [XDR-XSD] "ParentAccountDivision" element  -->
<xsd:element name="ParentAccountDivision">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountId" element  -->
<xsd:element name="ParentAccountId" type="xsd:string"/>

<!-- [XDR-XSD] "ParentAccountLocation" element  -->
<xsd:element name="ParentAccountLocation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountLocationLevel" element  -->
<xsd:element name="ParentAccountLocationLevel">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
```

```
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountName" element  -->
<xsd:element name="ParentAccountName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "ParentAccountRegion" element  -->
<xsd:element name="ParentAccountRegion">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyName" element  -->
<xsd:element name="PartyName">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyTypeCode" element  -->
<xsd:element name="PartyTypeCode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "PartyUId" element  -->
<xsd:element name="PartyUId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<!-- [XDR-XSD] "PrimaryIndustryId" element  -->
<xsd:element name="PrimaryIndustryId" type="xsd:string"/>

<!-- [XDR-XSD] "PrimaryOrganizationId" element  -->
<xsd:element name="PrimaryOrganizationId" type="xsd:string"/>

<!-- [XDR-XSD] "PrimaryPositionId" element  -->
<xsd:element name="PrimaryPositionId" type="xsd:string"/>

<!-- [XDR-XSD] "PrimaryServiceAgreementId" element  -->
<xsd:element name="PrimaryServiceAgreementId" type="xsd:string"/>

<!-- [XDR-XSD] "PrimarySynonymId" element  -->
<xsd:element name="PrimarySynonymId" type="xsd:string"/>

<!-- [XDR-XSD] "Region" element  -->
<xsd:element name="Region">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "S-SInstance" element  -->
<xsd:element name="S-SInstance">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="75"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "S-SInstanceId" element  -->
<xsd:element name="S-SInstanceId" type="xsd:string"/>

<!-- [XDR-XSD] "S-SKeyId" element  -->
<xsd:element name="S-SKeyId">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] "StreetAddress2" element  -->
<xsd:element name="StreetAddress2">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
```

```
  </xsd:simpleType>
</xsd:element>

<!-- [XDR-XSD] XDR datatype derivations -->
</xsd:schema>

Sample XML for Account Service Request

<?xml version="1.0" encoding="UTF-8"?>

<?Siebel-Property-Set EscapeNames="false"?>

<SiebelMessage MessageId="1-1N" IntObjectFormat="Siebel Hierarchical"
MessageType="Integration Object" IntObjectName="Sample Account">

   <ListOfSampleAccount>
      <Account>
         <Culture/>
         <CurrencyCode>USD</CurrencyCode>
         <Description/>
         <Division/>
         <HomePage/>
         <IntegrationId/>
         <LineofBusiness/>
         <Location/>
         <Name>New Sample Company</Name>
         <ProjectName/>
         <Type>Competitor</Type>
         <ListOfAccount_Organization>
            <Account_Organization IsPrimaryMVG="Y">
               <IntegrationId/>
               <Organization>Default Organization</Organization>
            </Account_Organization>
         </ListOfAccount_Organization>
         <ListOfAccountAttachment/>
         <ListOfBusinessAddress>
            <BusinessAddress>
               <AddressName>P Plaza, NY, NY</AddressName>
               <City>NY</City>
               <Country>USA</Country>
               <FaxNumber/>
               <IntegrationId/>
               <PhoneNumber/>
               <PostalCode/>
               <Province/>
               <State>NY</State>
               <StreetAddress>P Plaza</StreetAddress>
            </BusinessAddress>
         </ListOfBusinessAddress>
         <ListOfContact/>
```

```
      </Account>
   </ListOfSampleAccount>
</SiebelMessage>
```

Sample XDR for Sample Account

```
<?xml version="1.0" encoding="UTF-8"?>

<?Siebel-Property-Set EscapeNames="false"?>

<SiebelMessage MessageId="1-1N" IntObjectFormat="Siebel Hierarchical"
MessageType="Integration Object" IntObjectName="Sample Account">

   <ListOfSampleAccount>
      <Account>
         <Culture/>
         <CurrencyCode>USD</CurrencyCode>
         <Description/>
         <Division/>
         <HomePage/>
         <IntegrationId/>
         <LineofBusiness/>
         <Location/>
         <Name>New Sample Company</Name>
         <ProjectName/>
         <Type>Competitor</Type>
         <ListOfAccount_Organization>
            <Account_Organization IsPrimaryMVG="Y">
               <IntegrationId/>
               <Organization>Default Organization</Organization>
            </Account_Organization>
         </ListOfAccount_Organization>
         <ListOfAccountAttachment/>
         <ListOfBusinessAddress>
            <BusinessAddress>
               <AddressName>P Plaza, NY, NY</AddressName>
               <City>NY</City>
               <Country>USA</Country>
               <FaxNumber/>
               <IntegrationId/>
               <PhoneNumber/>
               <PostalCode/>
               <Province/>
               <State>NY</State>
               <StreetAddress>P Plaza</StreetAddress>
            </BusinessAddress>
         </ListOfBusinessAddress>
         <ListOfContact/>
      </Account>
```

```
   </ListOfSampleAccount>
</SiebelMessage>
```

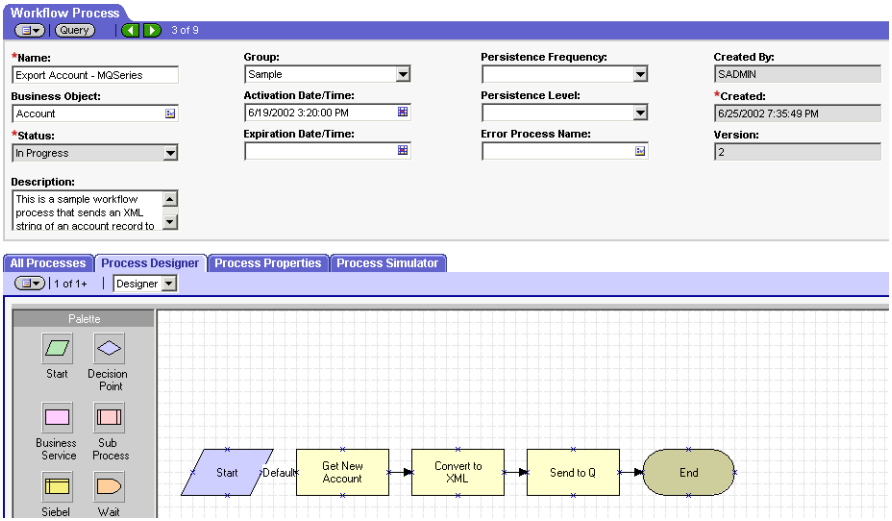
APPENDIX B

Creating Siebel Workflows

This section provides sample Siebel Workflows. It includes the following topics:

Creating a Siebel Workflow for Event Using MQSeries Transport

This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow exports Siebel Account record information using the MQSeries transport.

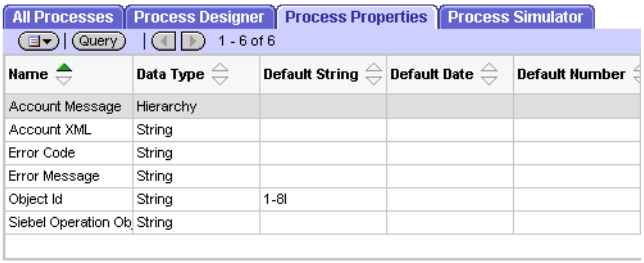


The steps that follow demonstrate how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center. It is then placed on an MQSeries message queue.

To create a Siebel Workflow for an event using MQSeries transport:

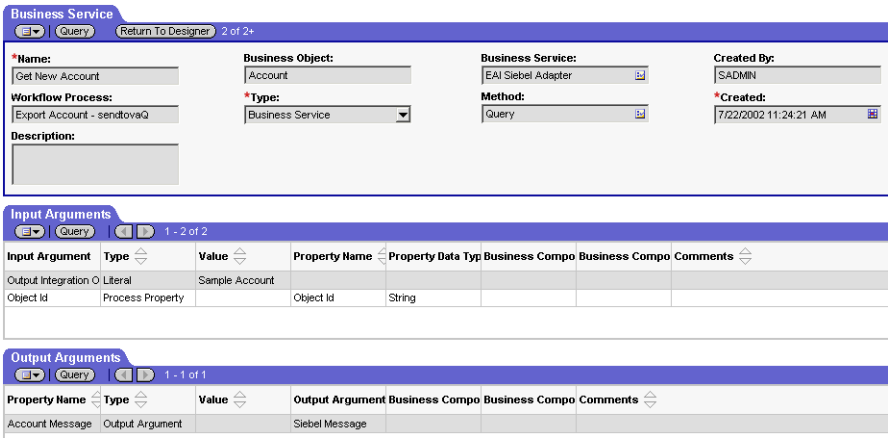
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   The Account message contains Siebel Account data in hierarchical format. Account XML specifies the Siebel Account data that the workflow has converted to XML.

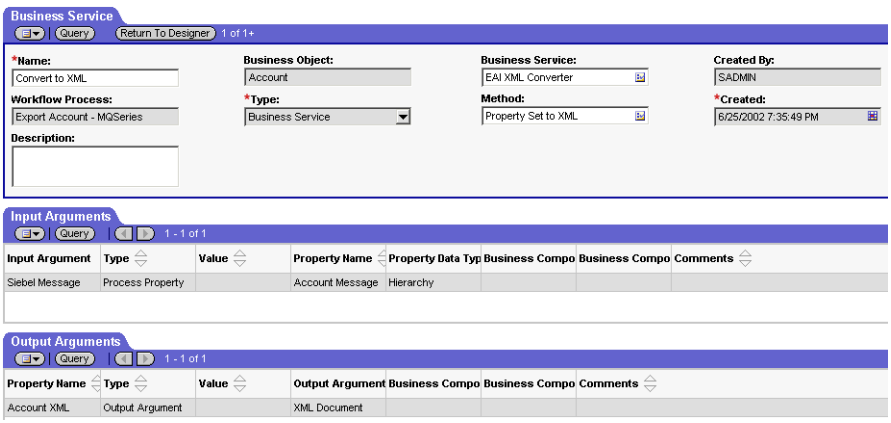


2. Use the Siebel Workflow Administration windows to create a workflow. Define an EAI Siebel Adapter business service step to receive an instance of Account data and call it Get New Account.

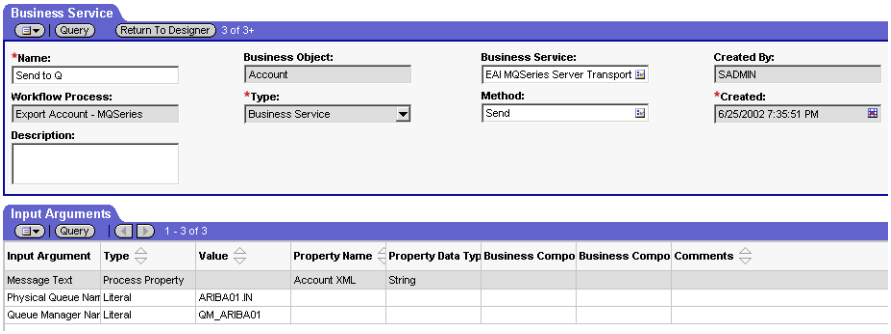
   The business service obtains the Account information from Siebel using the Query method. Output from this business service is generated in hierarchical format.



3. Define an EAI XML Converter business service step and call it Convert to XML. Define it to receive the Account data from the EAI Siebel Adapter business service in hierarchical format and convert it to XML format.

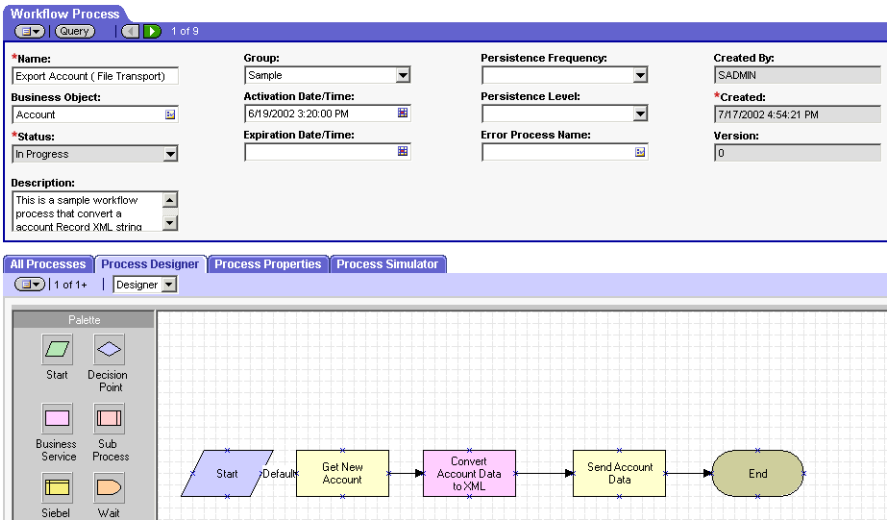


4. Define an EAI MQSeries server transport business service step and call it Send to Q. It should be defined to receive the Account data from the EAI XML Converter business service in Siebel XML format and send the Account XML to MQSeries using the Send method.



Creating a Siebel Workflow for Event Using File Transport

This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow exports Siebel Account record information using the File transport.

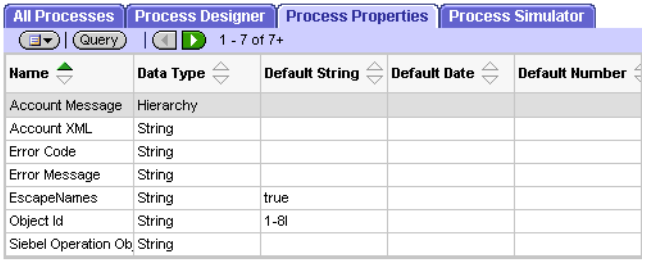


The steps that follow demonstrate to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center. It is then placed on the file system.

To create a Siebel workflow for an event using file transport:

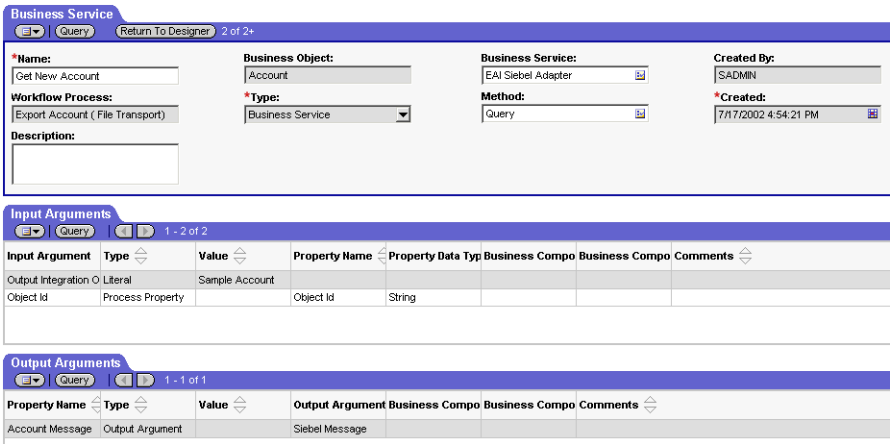
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

Account message contains the Siebel Account data in hierarchical format. Account XML specifies which Siebel Account data the workflow has converted to XML.

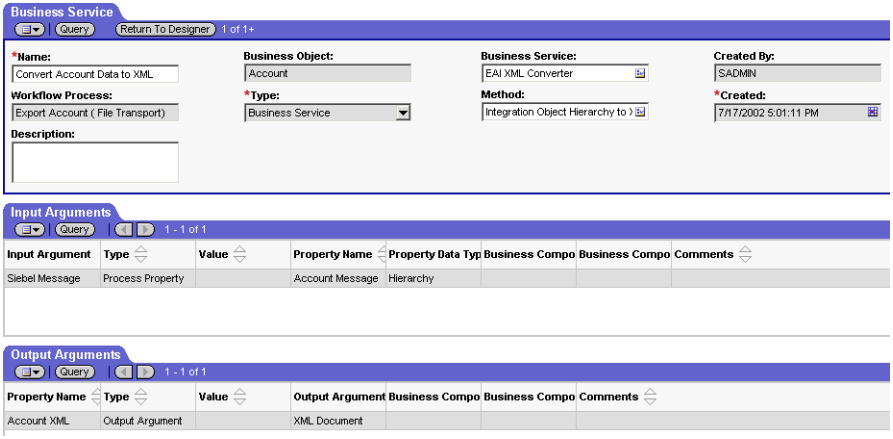


2. Use the Siebel Workflow Administration windows to create a workflow. Define an EAI Siebel Adapter business service step to receive an instance of Account data and call it Get New Account.

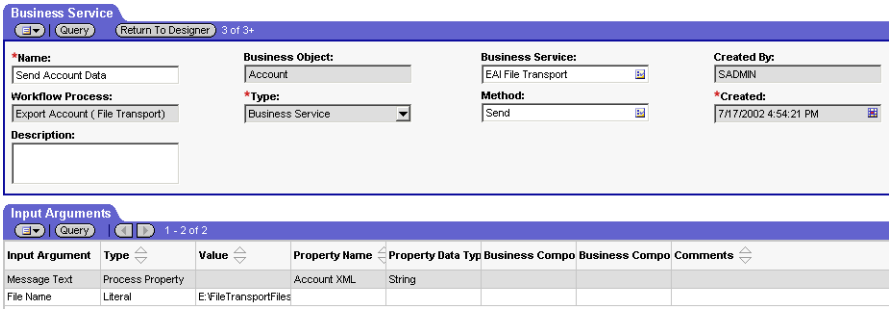
   The business service obtains the Account information from Siebel using the Query method. Output from this business service is generated in hierarchical format.



3. Define an EAI XML Converter business service step and call it Convert Account Data to XML. Define it to receive the Account data from the EAI Siebel Adapter business service in hierarchical format and convert it to XML format.

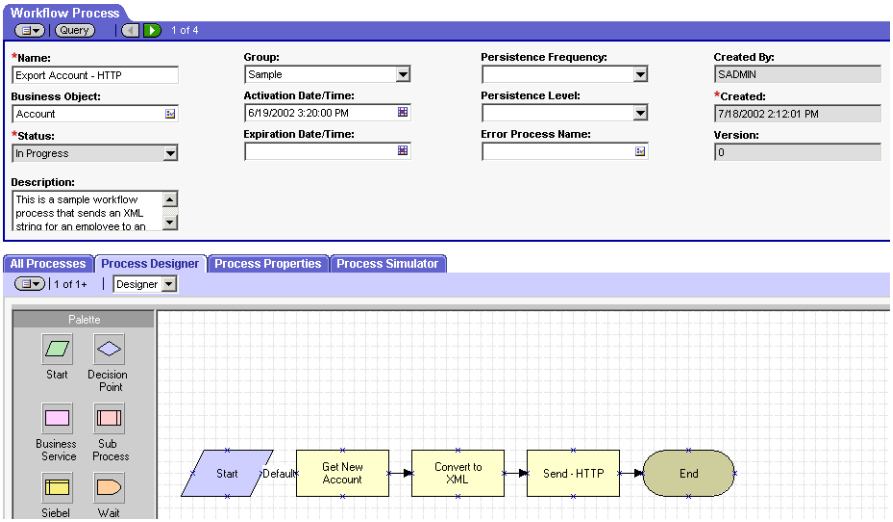


4. Define an EAI File transport business service step and call it Send Account Data. Define it to receive the Account data from the EAI XML Converter business service in Siebel XML format and Send the Account XML to the file system in a specified directory using the Send method.



Creating a Siebel Workflow for Event Using HTTP Transport

This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow exports Siebel Account record information using the HTTP transport.

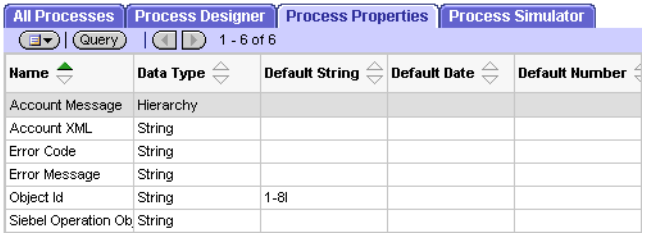


The steps that follow demonstrate how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center.

To create a Siebel workflow for an event using HTTP transport:

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

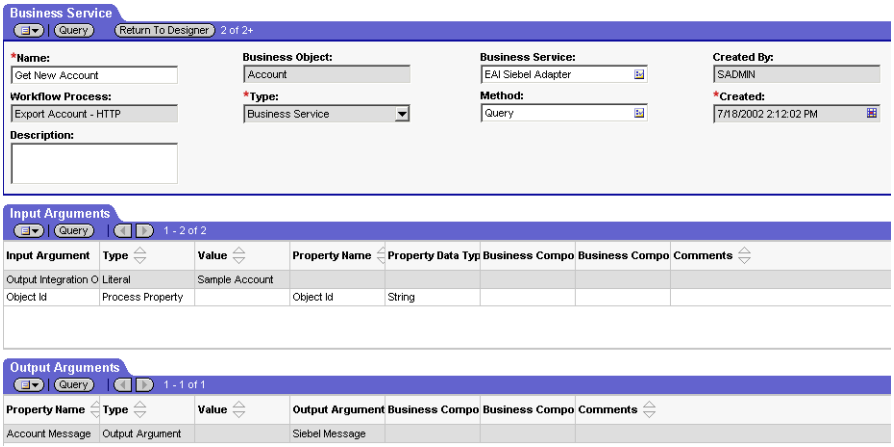
   Account message contains the Siebel Account data in hierarchical format. Account XML specifies the Siebel Account data that the workflow has converted to XML.



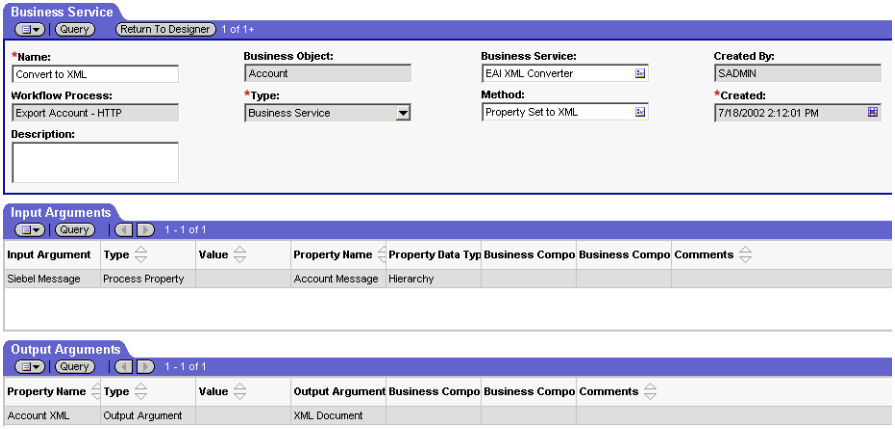
Name	Data Type	Default String	Default Date	Default Number
Account Message	Hierarchy			
Account XML	String			
Error Code	String			
Error Message	String			
Object Id	String	1-8I		
Siebel Operation Ob	String			

2. Use the Siebel Workflow Administration windows to create a workflow. Define an EAI Siebel Adapter business service step to receive an instance of Account data and call it Get New Account.

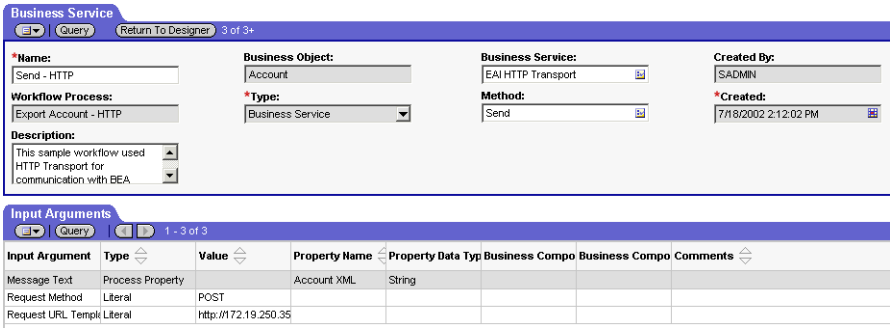
The business service obtains the Account information from Siebel using the Query method. Output from this business service is generated in hierarchical format.



3. Define an EAI XML Converter business service step and call it Convert to XML. Define it to receive the Account data from the EAI Siebel Adapter business service in hierarchical format and convert it to XML format.

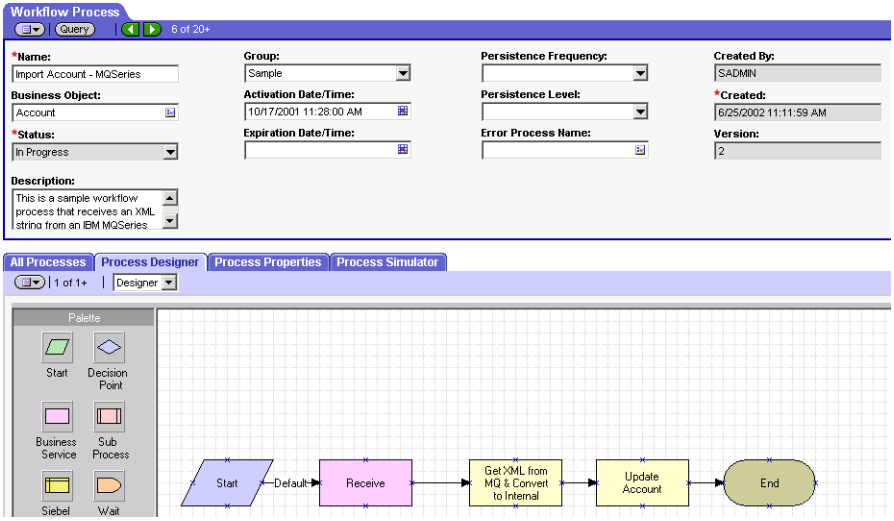


4. Define an EAI HTTP Transport business service step and call it Send - HTTP. Define it to receive the Account data from the EAI XML Converter business service in Siebel XML format and send the Account XML to HTTP using the Send method.



Creating a Siebel Workflow for Service Using MQSeries Transport

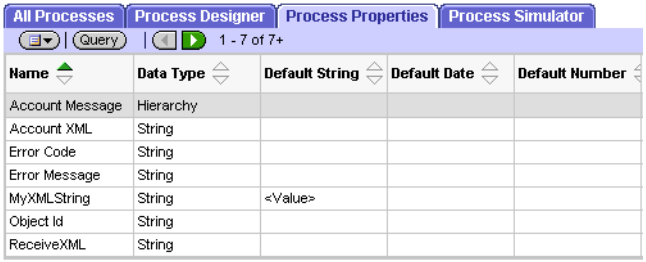
This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow imports Siebel Account record information using the MQSeries transport.



To create a Siebel workflow for a service using MQSeries transport:

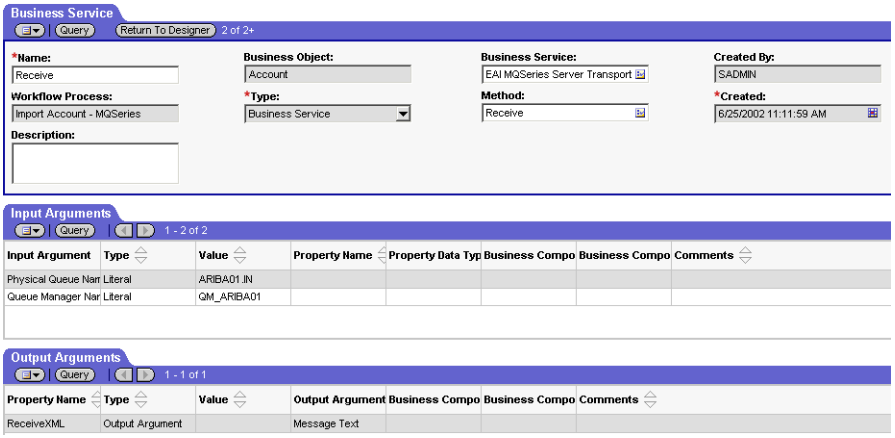
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

Account message contains the Siebel Account data in hierarchical format. Account XML specifies the Siebel Account data that the workflow has converted to XML.

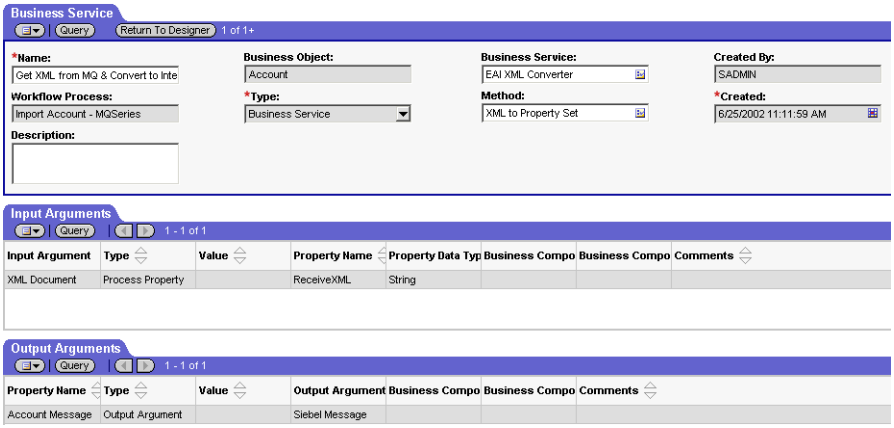


2. Define an EAI MQSeries Server Transport business service step and call it Receive.Define it to receive the Account data from the IBM MQSeries message queue.

   The EAI MQSeries Server Transport business service receives the Account data in Siebel XML format and sends it to the EAI XML Converter business service.

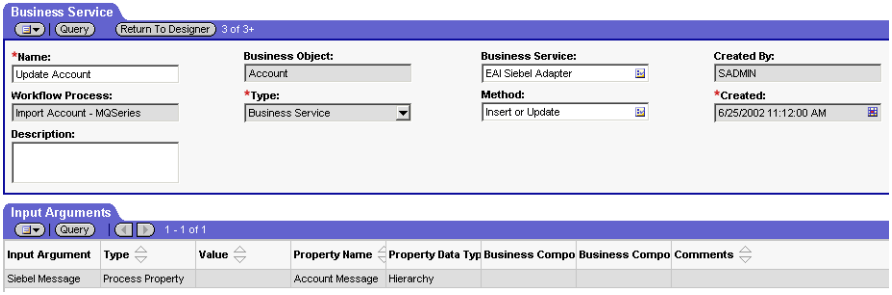


3. Define an EAI XML Converter business service step and call it Get XML from MQ & Convert to XML.Define it to receive the Account data from the EAI MQSeries Server Transport business service in XML format and convert it to hierarchical format.



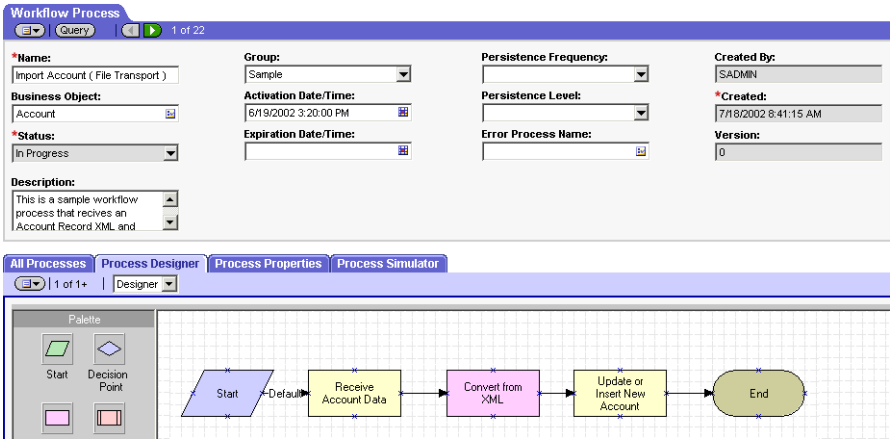
4. Define an EAI Siebel Adapter business service step and call it Update Account. Define it to receive from the EAI XML Converter business service the instance of Account data in hierarchical format.

   The business service applies the Account information into Siebel using the Insert or Update method.



Creating a Siebel Workflow for Service Using File Transport

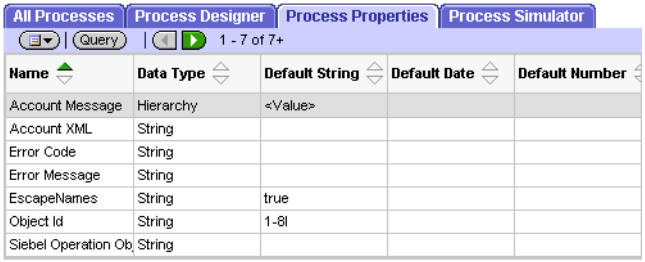
This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow imports Siebel Account record information using the File transport.



To create a Siebel workflow for a service using file transport:

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

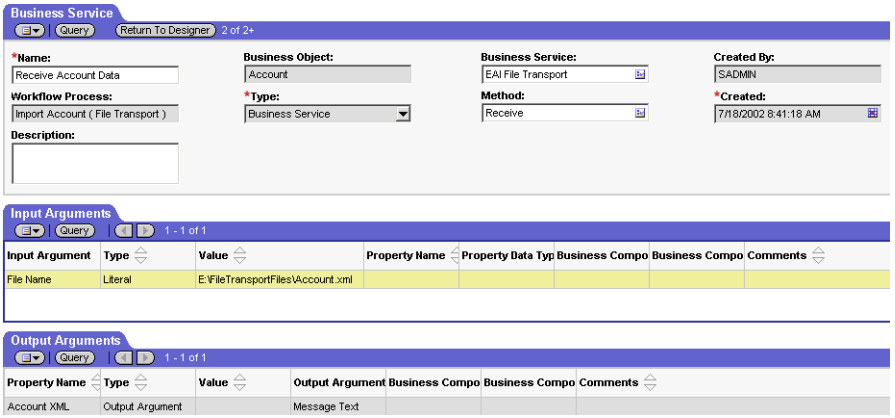
   Account message contains the Siebel Account data in hierarchical format. Account XML specifies the Siebel Account data that the workflow has converted to XML.



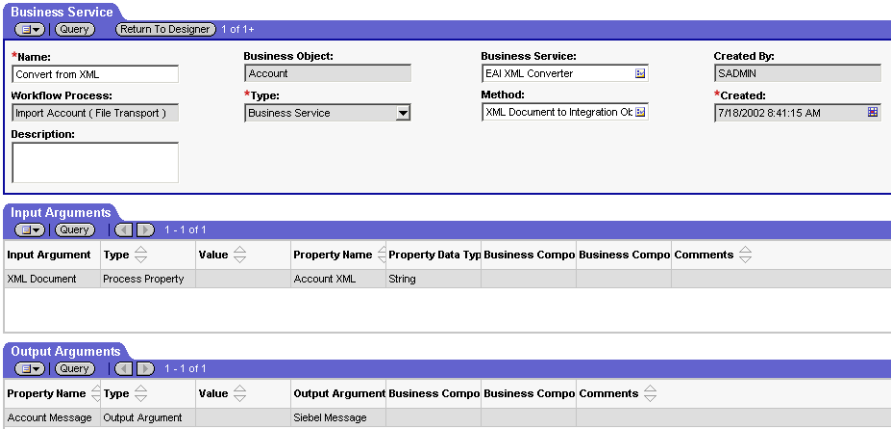
Name	Data Type	Default String	Default Date	Default Number
Account Message	Hierarchy	<Value>		
Account XML	String			
Error Code	String			
Error Message	String			
EscapeNames	String	true		
Object Id	String	1-8I		
Siebel Operation Ob	String			

2. Define an EAI File Transport business service step and call it Receive Account Data. Define it to receive the Account data from the file system.

   The EAI File Transport business service receives the Account data in Siebel XML format and sends it to the EAI XML Converter business service.

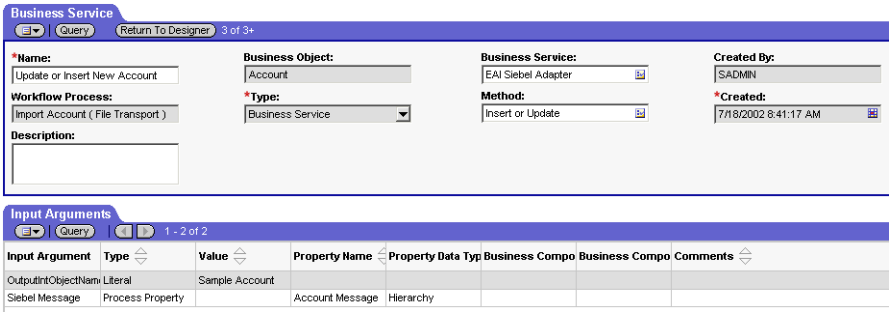


3. Define an EAI XML Converter business service step and call it Convert from XML. Define it to receive the Account data from the EAI File Transport business service in XML format and convert it to hierarchical format.



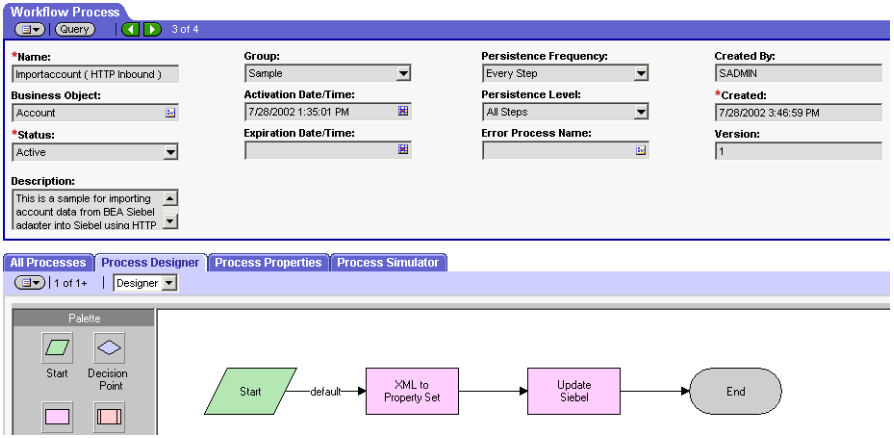
4. Define an EAI Siebel Adapter business service step and call it Update or Insert New Account. Define it to receive from the EAI XML Converter business service the instance of Account data in hierarchical format.

   The business service applies the Account information into Siebel using the Insert or Update method.



Creating a Siebel Workflow for Service Using HTTP Transport

This section describes a Siebel Workflow as seen in the Siebel Workflow Administration window. The workflow imports Siebel Account record information through the HTTP transport.



To create a Siebel workflow for a service using HTTP transport:

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format. Account XML specifies the Siebel Account data that the workflow has converted to XML.

All Processes | Process Designer | Process Properties | Process Simulator

Query | 1 - 7 of 7+

Name	Data Type	Default String	Default Date	Default Number
<Value>	String			
Account Message	Hierarchy			
Error Code	String			
Error Message	String			
IncomingXML	String	<Value>		
Object Id	String			
Process Instance Id	String			

2. Define an EAI XML Converter business service step and call it XML to Property Set. Define it to receive the Account data from the EAI HTTP Transport business service in XML format and convert it to hierarchical format.

Business Service

Query | Return To Designer | 2 of 2+

*Name: XML to Property Set	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Importaccount (HTTP Inbound)	*Type: Business Service	Method: XML Document to Integration Ob	*Created: 7/28/2002 3:47:00 PM
Description:			

Input Arguments

Query | 1 - 1 of 1

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
XML Document	Process Property		IncomingXML	String			

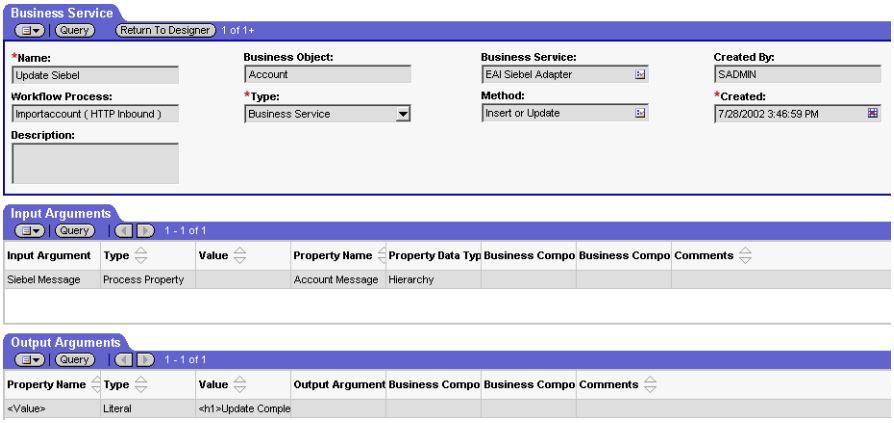
Output Arguments

Query | 1 - 1 of 1

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

3. Define an EAI Siebel Adapter business service step and call it Update Siebel. Define it to receive from the EAI XML Converter business service the instance of Account data in hierarchical format.

   The business service applies the Account information into Siebel using the Insert or Update method.


Business Service
Query
Return To Designer
1 of 1+
*Name:
Update Siebel
Business Object:
Account
Business Service:
EAI Siebel Adapter
Created By:
SADMIN
Workflow Process:
importaccount (HTTP Inbound)
*Type:
Business Service
Method:
Insert or Update
*Created:
7/28/2002 3:46:59 PM
Description:
Input Arguments
Query
1 - 1 of 1
Input Argument
Type
Value
Property Name
Property Data Typ
Business Compo
Business Compo
Comments
Siebel Message
Process Property
Account Message
Hierarchy
Output Arguments
Query
1 - 1 of 1
Property Name
Type
Value
Output Argument
Business Compo
Business Compo
Comments
<Value>
Literal
<h1>Update Comple

Index

N

P

R

S

T

X