iWay Software
An Information Builders Company

iWay

iWay Application System Adapter for Siebel User's Guide
Version 5 Release 5

Updated for J2EE CA 1.5

DN3501479.0305

Preface

This document is written for system integrators who develop client interfaces between Siebel and other applications.

How This Manual Is Organized

The following table lists the numbers and titles of the chapters and appendixes for this manual with a brief description of the contents of each chapter or appendix.

Chapter/Appendix		Contents
1	Introducing the iWay Application System Adapter for Siebel	Explains how the iWay Application System Adapter for Siebel integrates with Siebel. This section also describes the Siebel architecture.
2	Creating XML Schemas and iWay Business Services	Describes how to create XML schemas and business services for Siebel objects using Servlet Application Explorer.
3	Listening for Siebel Events	Describes how to use the iWay Application System Adapter for Siebel to listen for events in a Siebel system.
4	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
5	Management and Monitoring	Describes how you can use managing and monitoring tools provided by the iWay Business Services Engine (iBSE) and the iWay Connector for JCA to gauge the performance of your run-time environment.
A	Usage Considerations and Sample Files	Provides sample schemas for Siebel Business Components and Siebel Business Services.
B	Siebel Workflows	Describes Siebel topics relating to the processing of Siebel Integration Objects using Siebel XML.

What You Need to Know

This document is written for system integrators who develop client-server interfaces between Siebel and other applications.The iWay Application System Adapter for Siebel provides a means to exchange real-time business data between Siebel systems and other application, database, or external business partner systems. The adapter allows for inbound and outbound processing with Siebel. It is assumed that readers have:

- General knowledge of the Siebel environment, including Siebel Server, Siebel Client, Siebel Tools, and how to configure Siebel Server tasks.
- General knowledge of Siebel EAI concepts including how to use Siebel Tools and Wizards to create and modify Siebel Business Services and Integration Components.
- Specific knowledge of Siebel business applications.
- Knowledge of Siebel processes and data models for the required application area.
- General knowledge of client-server concepts.

Requirements

1. Installation of the proper iWay components.
2. Developer Access to Siebel Interfaces and Components:
    - Siebel Workflow Manager
    - Siebel Integration Object
    - EAI XML XDR Generator
    - EAI Siebel Adapter
    - EAI XML Converter Adapter
    - EAI MQSeries, HTTP or FILE Transports
    - MQSeries Receiver Server Task (for incoming message transport)

Documentation Conventions

The following table lists and describes the conventions that apply throughout this manual.

Convention	Description
`THIS TYPEFACE` or `this typeface`	Denotes syntax that you must enter exactly as shown.
`this typeface`	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
`underscore`	Indicates a default setting.
this typeface	Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term.
this typeface	Highlights a file name or command in a text paragraph that must be lowercase.
this typeface	Indicates a button, menu item, or dialog box option you can click or select.
Key + Key	Indicates keys that you must press simultaneously.
`{ }`	Indicates two or three choices; type one of them, not the braces.
`|`	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
`...`	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (…).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

To view a current listing of our publications and to place an order, visit our World Wide Web site, http://www.iwaysoftware.com. You also can contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about the iWay Application System Adapter for Siebel?

If you bought the product from a vendor other than iWay Software, contact your distributor.

If you bought the product directly from iWay Software, call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your iWay Application System Adapter for Siebel questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (*xxxx.xx*) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, http://www.informationbuilders.com. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, please be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the specifications our consultants require.

Platform	
Operating System	
OS Version	
Product List	
Adapters	
Adapter Deployment	For example, JCA, or iWay Business Services Engine.

Platform	
Container Version	

The following table lists components. Specify the version in the column provided.

Component	**Version**
iWay Adapter	
EIS (DBMS/APP)	
HOTFIX / Service Pack	

The following table lists the types of Application Explorer. Specify the version (and platform, if different than listed previously) in the columns provided.

Application Explorer Type	**Version**	**Platform**
Swing		
Servlet		
ASP		

In the following table, specify the JVM version and vendor in the columns provided.

Version	**Vendor**

The following table lists additional questions to help us serve you better.

Request/Question	**Error/Problem Details or Information**
Provide usage scenarios or summarize the application that produces the problem.	
Did this happen previously?	
Can you reproduce this problem consistently?	

Request/Question	Error/Problem Details or Information
Any **change in the application environment**: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	
Describe the **steps** to reproduce the problem.	
Describe the **problem**.	
Specify the **error** message(s).	

The following table lists error/problem files that might be applicable.

XML schema
XML instances
Other input documents (transformation)
Error screen shots
Error output files
Trace and log files
Log transaction

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to communicate suggestions for improving this publication or to alert us to corrections. You also can go to our Web site, http://www.iwaysoftware.com and use the Documentation Feedback form.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site, http://www.iwaysoftware.com or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our World Wide Web site, http://www.iwaysoftware.com.

Contents

CHAPTER 1

Introducing the iWay Application System Adapter for Siebel

Topics:

This section explains how the iWay Application System Adapter for Siebel integrates with Siebel. It describes the key features and also describes the Siebel architecture.

Features of the iWay Application System Adapter for Siebel

The iWay Application System Adapter for Siebel provides a means to exchange real-time business data between Siebel systems and other application, database, or external business partner systems. The adapter enables external applications for inbound and outbound processing with Siebel.

The adapter uses XML messages to enable non-Siebel applications to communicate and exchange transactions with Siebel using services and events.

- **Services:** Applications use this capability to initiate a Siebel business event.
- **Events:** Applications use this capability to access Siebel data only when a Siebel business event occurs.

The iWay Application System Adapter for Siebel:

- Supports synchronous and asynchronous, bidirectional message interactions for Siebel Business Services, Business Components, and Integration Objects.
- Includes the iWay Application Explorer, a GUI tool that uses the Siebel Object Manager to explore Siebel metadata and build XML schemas or Web services.
- Supports Siebel transports—MQSeries, File, and HTTP. In addition, it also supports MSMQ messaging.

The Siebel Application Model

The Siebel Enterprise application defines a data abstraction layer that removes dependencies on the underlying database. It accomplishes this by using intermediate Business Components and Business Objects that represent database structures. A Business Component typically represents a table in a database. A Business Object is a group of related business components.

From a given business component, you can navigate the relationships defined for that component to another component. The path you use to traverse component relationships is called the navigation path. For example, if you want to obtain all addresses for a particular account, you can traverse the parent/child relationship between Account and Address to obtain those addresses. By using navigation paths, you can traverse nearly all of the business component relationships defined in the Siebel system.

In Siebel, Integration Objects are similar to Siebel Business Components but describe more complex hierarchical data relationships.

Integrating With Siebel

You can use the iWay Application System Adapter for Siebel to invoke a Siebel business process, such as add/update account, or you can use the adapter as part of an integration effort to connect Siebel and non-Siebel systems. The iWay Application System Adapter for Siebel is bidirectional and can detect an event from Siebel by receiving a Siebel XML document emitted by Siebel.

When integrating with Siebel using Siebel XML documents, the adapter application developer must use existing Siebel Integration Objects or create new Siebel Integration Objects to use within a Siebel Workflow. The Workflow processes inbound or outbound Siebel XML and uses various transports such as MQSeries, File, and HTTP to exchange transactions with external systems. The Siebel Workflow is usually created by the Siebel administrator or developer using Siebel Workflow Administration screens.

When integrating with Siebel directly using the Java™ Data Bean or COM Data Interface, the iWay Application System Adapter for Siebel does not require a Siebel Integration Object or Siebel Workflow. Instead, it executes Siebel Business Services and Siebel Business Components directly.

The following table lists Siebel objects and describes the transport methods and processes for each object.

Siebel Objects	API or Transport	Process
Business Services	Java Data Bean (Siebel Version 6.3 or higher) Com Data Interface (Siebel Version 6.2 or lower)	Service
Business Components	Java Data Bean (Siebel Version 6.3 or higher) Com Data Interface (Siebel Version 6.2 or lower)	Service
Integration Objects	File	Event, Service
	HTTP	Event, Service
	MQSeries	Event, Service
	MQ Read	Service

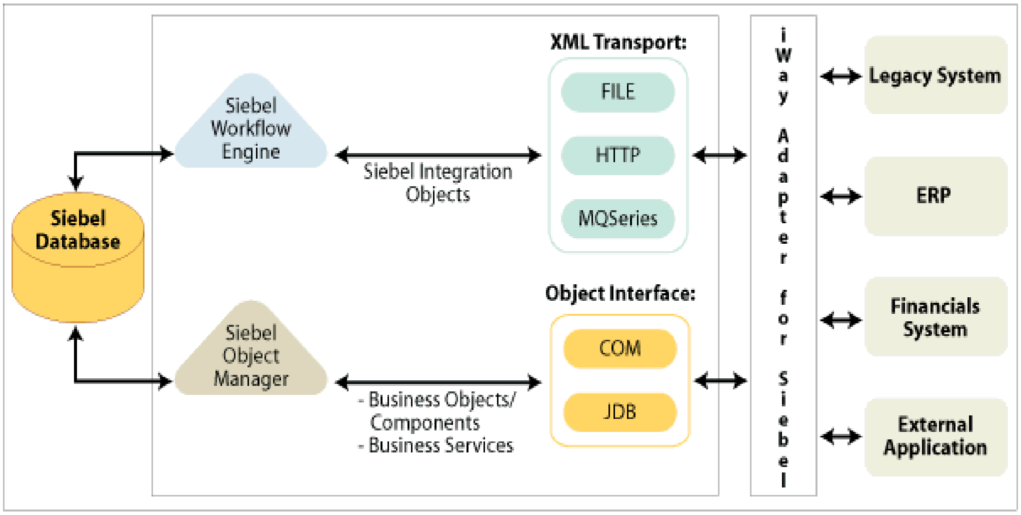
Siebel EAI Architecture

Siebel provides for integration with other applications and systems using its Siebel EAI framework and its Business Integration Manager facility. The iWay Application System Adapter for Siebel uses the Siebel EAI framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality while working within the Siebel framework.

The iWay Application System Adapter for Siebel supports the following integration access methods:

- Siebel Java Data Bean for services involving Siebel Business Components or Siebel Business Services.
- Siebel COM Data Interface for services involving Siebel Business Components or Siebel Business Services.
- Siebel XML for events and services involving Siebel Integration Objects.

The following illustration shows how the iWay Application System Adapter for Siebel helps integrate a Siebel database through either a transport protocol such as File, HTTP, or MQSeries and the Siebel Workflow Engine for Siebel Integration Objects with legacy systems, ERPs, financial systems, and external applications. It also shows how the iWay Adapter for Siebel helps integrate a Siebel database through an object interface such as COM or JDB and the Siebel Object Manager for Business Components and Business Services with legacy systems, ERPs, financial systems, and external applications.



Using iWay Application Explorer With the iWay Application System Adapter for Siebel

iWay Application Explorer uses an explorer metaphor for browsing the Siebel system for Business Services, Business Objects, Business Components, and Integration Objects. The explorer enables you to create XML schemas and Web services for the associated object. External applications that access Siebel through the iWay Application System Adapter for Siebel use either XML schemas or Web services to pass data between the external application and the adapter.

Application Explorer uses interfaces provided by Siebel and in-depth knowledge of the Siebel application systems to access and browse business object metadata. After an object is selected, Application Explorer can generate an XML schema or Web service to define the object for use in conjunction with the iWay Application System Adapter for Siebel.

External applications accessing Siebel via the iWay Application System Adapter for Siebel use either the XML schema or Web service to pass data between the external application and the adapter.

The steps required to create XML schemas for Web services are illustrated in *Chapter 2, Creating XML Schemas and iWay Business Services*.

Key Features of Application Explorer

Key features of Application Explorer include:

- The ability to connect to and explore a variety of application systems.
- Access to application system object metadata.
- A point-and-click process for generating XML schemas and Web services.

Installing and Configuring the Servlet Application Explorer

Application Explorer must be deployed through a servlet container or application server.

In addition, the Siebel Enterprise Information System (EIS) must be installed, configured, and available for client access. Application Explorer need not reside on the same system as Siebel, but network access is required.

For more information on installing and configuring the Java Servlet Application Explorer, see the *iWay Installation and Configuration* documentation.

Deployment Information for the iWay Application System Adapter for Siebel

The iWay Application System Adapter for Siebel works in conjunction with Application Explorer and one of the following components:

- iWay Business Services Engine (iBSE)
- Enterprise Connector for J2EE™ Connector Architecture (JCA)

Application Explorer, used to configure Siebel connections and create Web services and events, can be configured to work in a Web services environment in conjunction with the iWay Business Services Engine or with the Enterprise Connector for J2EE Connector Architecture (JCA). When working in a JCA environment, the connector uses the Common Client Interface (CCI) to provide fast integration services using adapters instead of using Web services.

Both iBSE and the Connector for JCA are deployed to an application server.

Deployment Information Roadmap

The following table lists the location of deployment information for components of the iWay Application System Adapter for Siebel. A description of the iWay Business Services Engine (iBSE) and the Enterprise Connector for J2EE Connector Architecture (JCA) follows the table.

Deployed Component	**For more information, see**
Application Explorer	Chapters 2 and 3 of this guide *iWay Installation and Configuration* *iWay Servlet Application Explorer User's Guide*
iWay Business Services Engine (iBSE)	*iWay Installation and Configuration*
Enterprise Connector for J2EE Connector Architecture (JCA)	*iWay Connector for JCA User's Guide* *iWay Installation and Configuration*

The iWay Business Services Engine

The iWay Business Services Engine (iBSE) exposes—as Web services—enterprise assets that are accessible from adapters regardless of the programming language or the particular operating system.

iBSE simplifies the creation and execution of Web services when running:

- Custom and legacy applications
- Database queries and stored procedures
- Packaged applications
- Terminal emulation and screen-based systems
- Transactional systems

Web services is a distributed programming architecture that solves Enterprise Application Integration (EAI) hurdles that other programming models cannot. It enables programs to communicate with one another using a text-based, platform- and language-independent message format called XML.

Coupled with a platform and language independent messaging protocol called SOAP (Simple Object Access Protocol), XML enables application development and integration by assembling previously built components from multiple Web services.

The Enterprise Connector for J2EE Connector Architecture (JCA)

The Enterprise Connector for J2EE Connector Architecture (JCA) enables developers of JCA-compliant applications to deploy adapters as JCA resources. The connector is supported on J2EE-compliant application servers, such as the Sun Java System Application Server.

The Connector for JCA is distributed as a standard Resource Adapter Archive (RAR) for deployment to the application server. Thus, the connector can be used in systems that are non-compliant, although services such as pooled connections are not available.

Two connectors are distributed in the iWay installation package. One conforms to the JCA 1.0 specification, with extensions that allow for the consumption of events. The other conforms to the JCA 1.5 specification. The JCA 1.0 connector provides for event functionality through the configuration of ports and channels. When using the adapter in conjunction with a JCA 1.5 connector, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities. For more information on event capabilities of the iWay JCA connectors, see Chapter 3, *Listening for Siebel Events*.

For more information on installing and deploying both connectors, see *iWay Installation and Configuration*.

CHAPTER 2

Creating XML Schemas and iWay Business Services

Topics:

- Processing Overview
- Starting Servlet Application Explorer
- Managing a Siebel Connection
- Viewing Metadata
- Creating a Schema for a Siebel Business Component or Siebel Business Service
- Creating an XML Schema for a Siebel Integration Object
- Creating a Web Service

This section provides the information you require to create schemas for Siebel Business Components, Business Services, and Integration Objects. It describes how to use Servlet Application Explorer as deployed to a Sun Java System Application Server.

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces.

Processing Overview

The iWay Application System Adapter for Siebel enables interaction with Siebel Business Services, Business Components, and Integration Objects.

When using the adapter to integrate with Siebel Business Services and Business Components, the adapter uses the Siebel-supplied Java Data Bean or COM EAI interface. You are not required to create Siebel workflows. Also, because the service is accomplished through a TCP connection, you do not require a transport layer such as MQSeries, File, or HTTP.

A request begins with the sending of an XML request document. In most cases, the response is an XML response document that indicates the execution of the Business Service or Business Component.

When using the adapter to integrate with Siebel Integration Objects, the adapter uses Siebel XML, HTTP, File, WebSphere MQ, and MSMQ transports and Siebel workflows. The workflow is defined within Siebel to either emit or receive Siebel XML through one of the supported Siebel transport services for MQSeries, File, or HTTP.

Starting Servlet Application Explorer

Before you can use Servlet Application Explorer, you must start your application server.

Procedure: How to Open Servlet Application Explorer

To open Application Explorer:

1. Ensure that your application server is running.
2. Enter the following URL in your browser:

   `http://`*`hostname`*`:`*`port`*`/iwae/index.html`

   where:

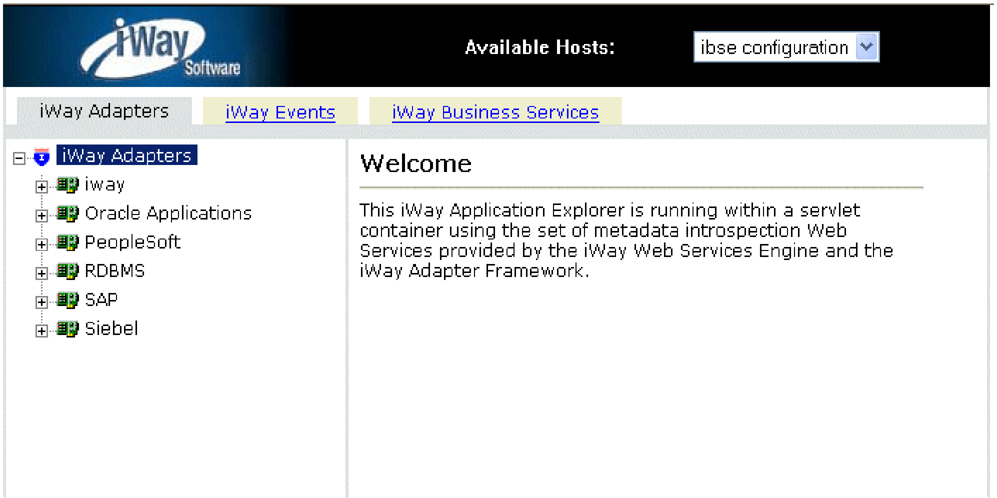
   `hostname`

   Is the name of the machine where your application server is running.

   `port`

   Is the port for the domain you are using for iWay.

The following image shows the Application Explorer window that opens and includes three tabs corresponding to iWay Adapters, iWay Events, and iWay Business Services. The iWay Adapters tab is active, and a list of supported adapters appears in the left navigation pane beneath it. In the upper right, the Available Hosts drop-down list displays the Connector for JCA or Servlet iBSE instance you can access. A Welcome message appears in the right pane.



Welcome

This iWay Application Explorer is running within a servlet container using the set of metadata introspection Web Services provided by the iWay Web Services Engine and the iWay Adapter Framework.

For more information on adding instances, see the *iWay Installation and Configuration* manual.

You are ready to create new targets to the Siebel enterprise information system.

Managing a Siebel Connection

To browse the Siebel Business Services, Business Components, and Integration Objects, you must define a target to Siebel. After you define the target, the parameters are automatically saved. However, you must supply the password to Siebel every time you connect to the target. For more information on connecting to a target, see *How to Connect to a Defined Target* on page 2-8.

You create a new target from the Service Adapters tab of Application Explorer. For information on creating a target, see *How to Define a Target to Siebel* on page 2-4.

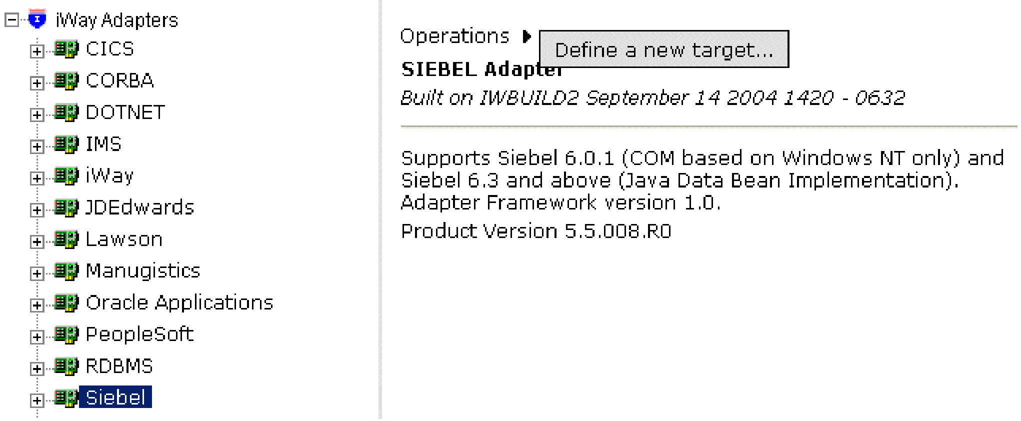
Although you can maintain multiple open connections to different application systems, it is good practice to close connections when not in use. For information on disconnecting from a target, see *How to Disconnect From Siebel* on page 2-8.

After you create a target for Siebel using Application Explorer, you can edit any information that you provided during the creation process. For information on editing a target, see *How to Edit a Target* on page 2-8.

You can delete a target, rather than just disconnecting from a target and closing it. When you delete the target, the node disappears from the list of Siebel targets in the left pane of the explorer. For information on deleting a target, see *How to Delete a Target to Siebel* on page 2-9.

Procedure: How to Define a Target to Siebel

The following image shows a window with a navigation pane that lists supported adapters on the left and information about a selected adapter in the right pane.



SIEBEL Adapter

Built on IWBUILD2 September 14 2004 1420 - 0632

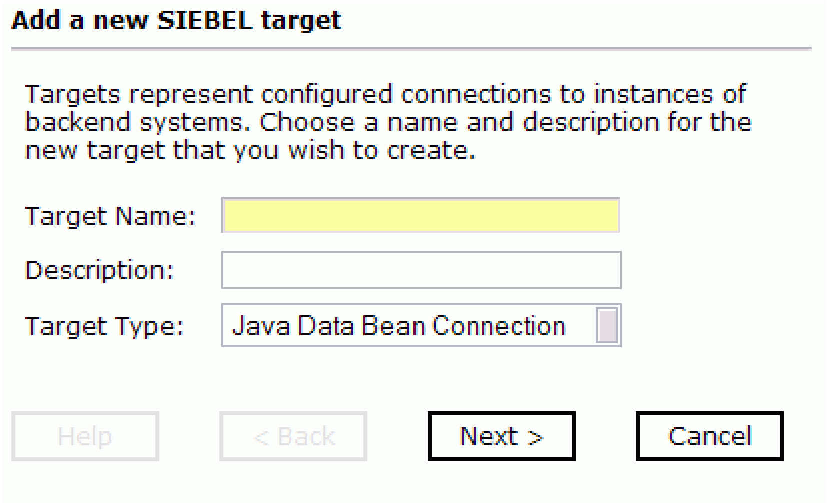
Supports Siebel 6.0.1 (COM based on Windows NT only) and Siebel 6.3 and above (Java Data Bean Implementation). Adapter Framework version 1.0.

Product Version 5.5.008.R0

To define a target to Siebel:

1. In the left pane of Application Explorer under Service Adapters, select the *Siebel* node.
2. In the right pane, move the pointer over *Operations* and select *Define a New Target*.

The following image shows the pane that appears on the right where you can define a new target.



Add a new SIEBEL target

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name:

Description:

Target Type: Java Data Bean Connection

a. In the Target Name field, type a name for the new target, for example, SiebelConnection.

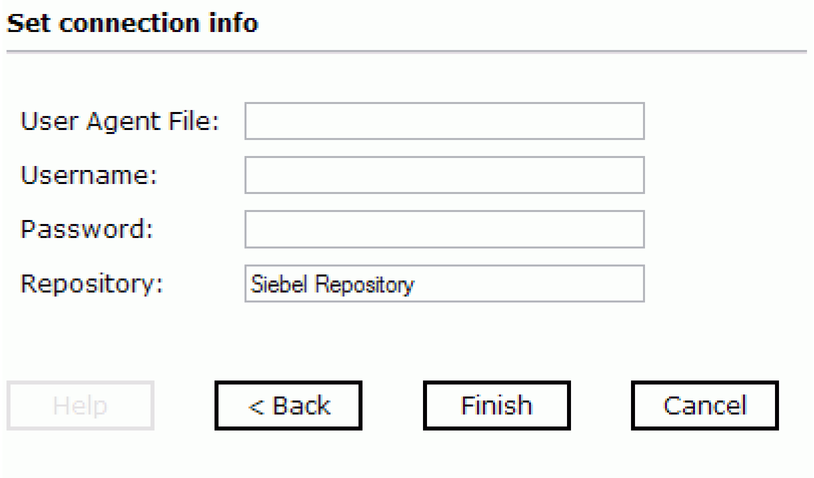
b. In the Description field, type a description (optional).

c. For Siebel 6.0, select *Siebel 6.2* or for Siebel 6.3 or higher, select *Java Bean Data Connection* from the Target Type drop-down list.

3. To continue, click *Next*. To go back or to cancel, click the appropriate button.

When you select 6.2 or lower (COM), the Set connection info dialog box appears where you type connection and logon information.

The following image shows the Set connection info dialog box.



Set connection info

User Agent File:

Username:

Password:

Repository: Siebel Repository

a. In the User Agent File field, type *UAGENT.CFG* or type the name of your configuration file, if the name is different.

UAGENT.CFG is the default value.

This file contains the configuration parameters for connecting to Siebel.

b. In the Username field, type the Siebel user name.

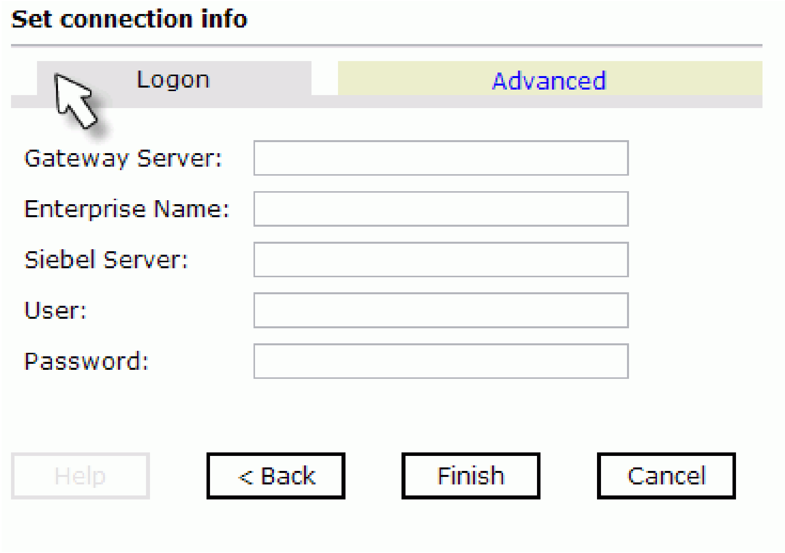
c. In the Password field, type the password associated with the user name.

d. In the Repository field, type the name of the repository where Application Explorer looks for metadata describing Business Services, Business Objects, and Integration Objects.

If no repository is specified, a full list of objects from all available repositories is returned. If a specified repository is not found, an empty list of objects is returned.

When you select 6.3 or higher (JDB), the Set connection info dialog box appears with a Logon tab and an Advanced tab.

The following image shows the active Logon tab where you type configuration information.



a. In the Gateway Server field, type the name of the 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.

b. In the Enterprise Name field, type the appropriate name.

c. In the Siebel Server field, type the name of your Siebel server. You do not have to supply a value in this field when connecting to a Siebel 7.7 system.

d. In the User field, type the user name.

e. In the Password field, type the password associated with the user name.

4. Click the *Advanced* tab and verify the following: Language, Object Manager, and Repository Name.

Object Manager is the name of an active Siebel Object Manager. For example, SCCObjMgr is the name of the Siebel Object Manager used for the Siebel Call Center application. For Siebel 7.0.3, the default value is EAIObjMgr. For Siebel 7.7, the default is EAIObjMgr_enu. Siebel 7.7 requires that you add the language extension (for example, _enu) to the end of the Object Manager name. Check with your Siebel Administrator for the specific names that apply to your system.

If no repository is specified, a full list of objects from all available repositories is returned. If a specified repository is not found, an empty list of objects is returned.

The configuration parameters supplied are used by Siebel client applications to connect to the Siebel system. For more information about these parameters, see your Siebel documentation or ask your Siebel system administrator.

5. Click *OK*.

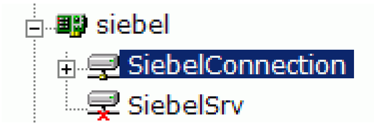
   In the left pane, the target you create appears under the Siebel node.

Procedure: How to Connect to a Defined Target

To connect to a previously defined Siebel target:

1. Click the target name under the Siebel node.
2. In the right pane, move the cursor over *Operations* and select *Connect*.
3. Type the password and click *OK*.

   The following image shows the siebel node with the SiebelConnection target icon selected beneath it. The SiebelSrv target is disconnected from the Siebel system.



   You can browse available Business Objects, Business Services, and Integration Objects in the Siebel system.

Procedure: How to Disconnect From Siebel

To disconnect from Siebel:

1. In the left pane, click the target to which you are connected.
2. In the right pane, move the pointer over *Operations* and select *Disconnect*.

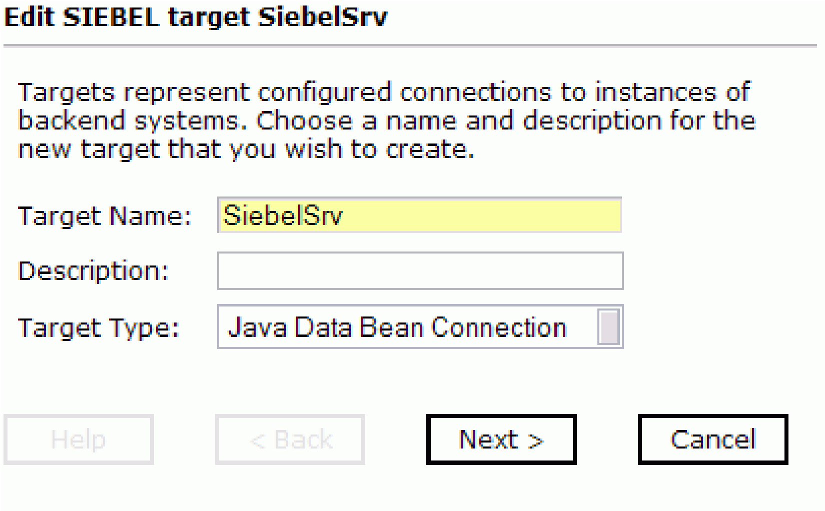
   Disconnecting from the application system drops the target, but the node remains. The SiebelConnection node in the left pane changes to reflect that the target was closed.

Procedure: How to Edit a Target

To edit a target:

1. In the left pane, click the target node.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

The following image shows the Edit pane that opens on the right. There are three fields where you can edit information: Target Name, Description, and Target Type. You can click a button to continue, go back, cancel out of the Edit pane, or view help.



Edit SIEBEL target SiebelSrv

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name: SiebelSrv

Description:

Target Type: Java Data Bean Connection

3. Modify the target information.
4. Click *Next*.

Procedure: How to Delete a Target to Siebel

To delete a target to Siebel:

1. In the left pane, click the target, for example, SiebelConnection.
2. In the right pane, move the cursor over *Operations* and select *Delete*.

   A confirmation dialog box opens.
3. To delete the target you selected, click *OK*.

   The SiebelConnection node disappears from the left pane.

Viewing Metadata

Viewing metadata is useful for understanding the structure of Siebel data. You can review the parameters, data types, and other attributes in the right pane.

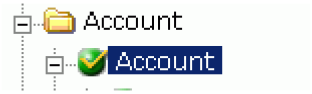
Procedure: How to View Metadata

To view metadata:

1. If you have not started the explorer, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the Business Object or Business Service containing the component for which you want to generate schema.
3. Expand the Business Object or Business Service node.
4. Expand the Business Component or the Business Service node in which you are interested to view the objects under it.

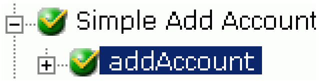
   For a Business Component, select the node in which you are interested.

   The following image shows the Account Business Component expanded with the Account object selected.

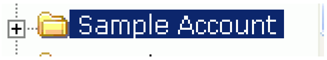
   For a Siebel Business Service, select the object in which you are interested.

   The following image shows the Simple Add Account Business Service expanded with the addAccount object selected.

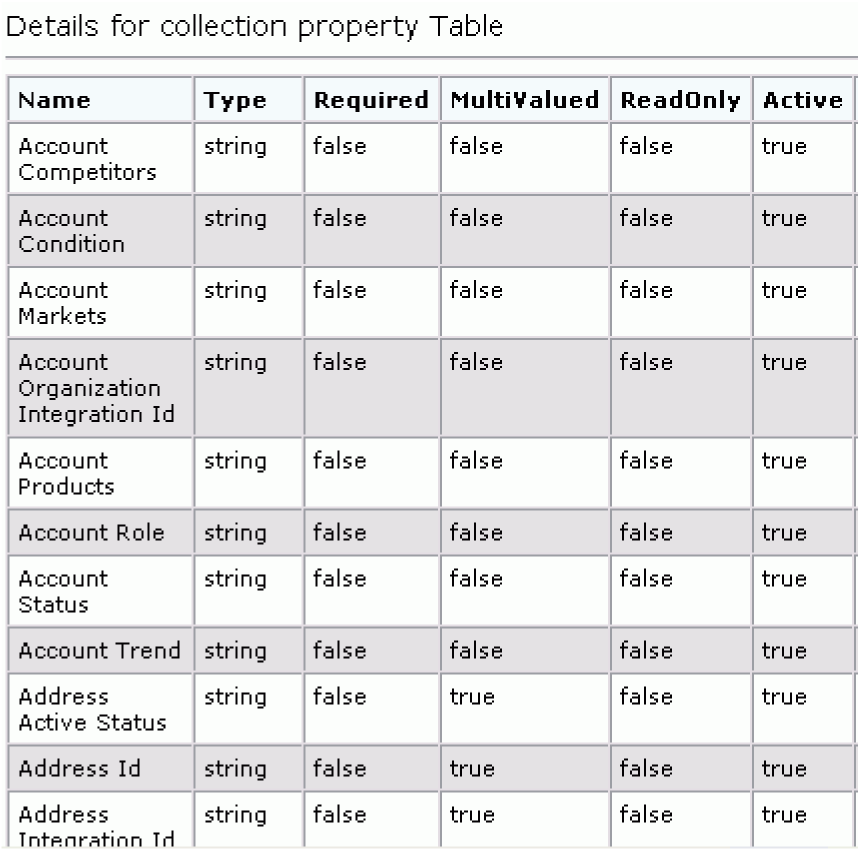
   For an Integration Object, select the Integration Object in which you are interested.

   The following image shows the Sample Account Integration Object selected.

5. In the right pane, click the ellipsis (…) in the Table row of the properties table.

A metadata table appears in the right pane and displays the details of the table you selected. The following image is a sample Details for collection property Table. The table consists of a heading row with column labels that identify each of the six columns: Name, Type, MultiValued, ReadOnly, and Active. Each row represents a different property.



Details for collection property Table

Name	Type	Required	MultiValued	ReadOnly	Active
Account Competitors	string	false	false	false	true
Account Condition	string	false	false	false	true
Account Markets	string	false	false	false	true
Account Organization Integration Id	string	false	false	false	true
Account Products	string	false	false	false	true
Account Role	string	false	false	false	true
Account Status	string	false	false	false	true
Account Trend	string	false	false	false	true
Address Active Status	string	false	true	false	true
Address Id	string	false	true	false	true
Address Integration Id	string	false	true	false	true

Creating a Schema for a Siebel Business Component or Siebel Business Service

You can create service schemas for Business Services, Business Components, and Integration Objects using Application Explorer. For information on creating schemas for Integration Objects, see *Creating an XML Schema for a Siebel Integration Object* on page 2-15.

The following topic, *Creating an XML Schema for a Siebel Business Component or Siebel Business Service*, describes how to create schemas for the adapter when you deploy the iWay Application System Adapter for Siebel for use either in a JCA (iWay Enterprise Connector for J2EE Connector Architecture) environment or a Web services environment.

If you plan to deploy the iWay Application System Adapter for Siebel in a Web services environment, see also *Creating a Web Service* on page 2-25.

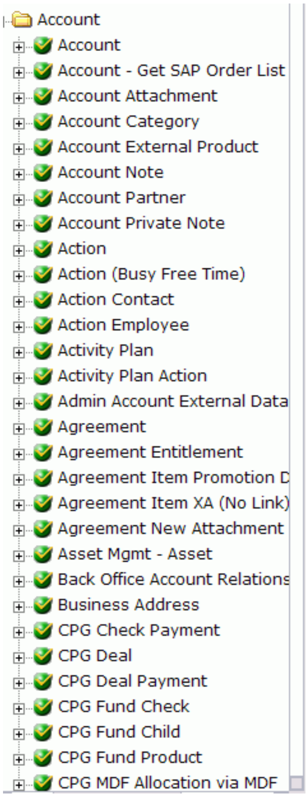
Creating an XML Schema for a Siebel Business Component or Siebel Business Service

You create schemas for Siebel Business Service methods (for example, the Add method) and Business Components using iWay Application Explorer. After you create a schema, you can use it to generate service request and response schemas for the Business Service or Business Component.

For information on creating schemas for Integration Objects, see *Creating an XML Schema for a Siebel Integration Object* on page 2-15.

Siebel Business Objects contain one or more Siebel Business Components. You can view Business Components by clicking the associated Business Object.

The following image shows the Account Business Object node expanded to display all Business Components beneath it.



Account
- Account
- Account - Get SAP Order List
- Account Attachment
- Account Category
- Account External Product
- Account Note
- Account Partner
- Account Private Note
- Action
- Action (Busy Free Time)
- Action Contact
- Action Employee
- Activity Plan
- Activity Plan Action
- Admin Account External Data
- Agreement
- Agreement Entitlement
- Agreement Item Promotion D
- Agreement Item XA (No Link)
- Agreement New Attachment
- Asset Mgmt - Asset
- Back Office Account Relations
- Business Address
- CPG Check Payment
- CPG Deal
- CPG Deal Payment
- CPG Fund Check
- CPG Fund Child
- CPG Fund Product
- CPG MDF Allocation via MDF

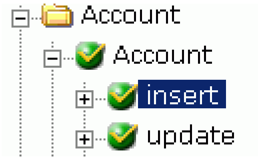
Procedure: How to Create an XML Schema for a Siebel Business Component or Siebel Business Service

To generate service request and response schemas for a Business Component:

1. If you have not started the explorer, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the Business Object or the Business Service node.
3. Expand the Business Component, Business Service, or Integration Object to view the objects under it.

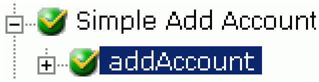
   For a Business Component, expand the component containing the node you wish and then, expand the node and select the method for which you want to create a schema.

The following image shows the Account Business Component expanded to reveal the Account node. This node is expanded with the insert method selected.



For a Siebel Business Service, expand the Business Service node containing the object for which you want to create schema.

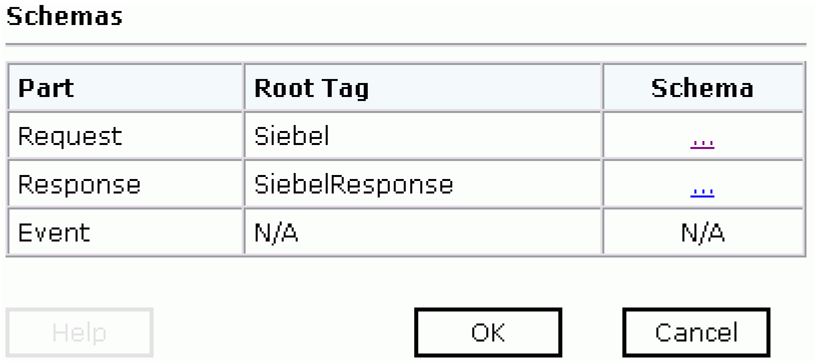
The following image shows the Simple Add Account Business Service expanded with the addAccount object selected.



4. In the right pane, move the cursor over *Operations* and select *Generate Schema*.

   Application Explorer accesses the Siebel repository and builds schemas.

   The following image shows the Schemas table that appears on the right and has three columns, labeled Part, Root Tag, and Schema. The Schema column provides the locations of the schemas. There are three rows: Request, Response, and Event.

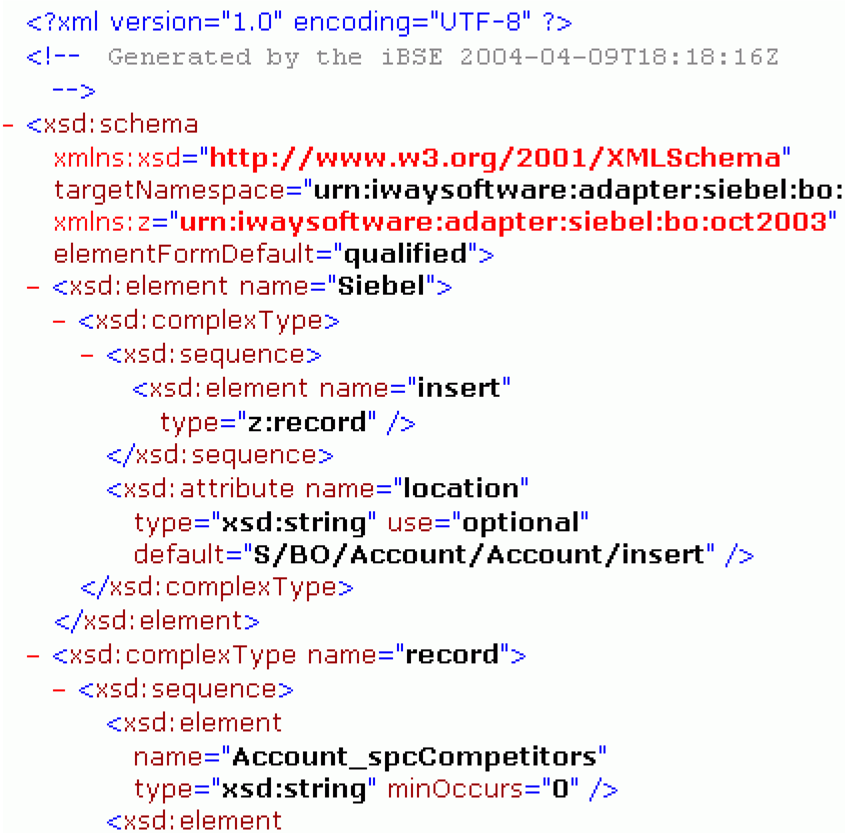


Schemas

Part	Root Tag	Schema
Request	Siebel	...
Response	SiebelResponse	...
Event	N/A	N/A

5. To view a schema, click the ellipsis (…) in the row corresponding to the schema you want to view.

The following image shows the XML schema that appears in the right pane.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  Generated by the iBSE 2004-04-09T18:18:16Z
  -->
- <xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:iwaysoftware:adapter:siebel:bo:
    xmlns:z="urn:iwaysoftware:adapter:siebel:bo:oct2003"
    elementFormDefault="qualified">
  - <xsd:element name="Siebel">
    - <xsd:complexType>
      - <xsd:sequence>
          <xsd:element name="insert"
            type="z:record" />
        </xsd:sequence>
        <xsd:attribute name="location"
          type="xsd:string" use="optional"
          default="S/BO/Account/Account/insert" />
      </xsd:complexType>
    </xsd:element>
  - <xsd:complexType name="record">
    - <xsd:sequence>
        <xsd:element
          name="Account_spcCompetitors"
          type="xsd:string" minOccurs="0" />
        <xsd:element
```

For more information on where the schemas are stored, see *Schema Location on page 2-24*.

Creating an XML Schema for a Siebel Integration Object

The iWay Application System Adapter for Siebel supports access to Siebel Integration Objects by using Siebel XML. Using Siebel Integration Objects through supported transports requires Siebel workflows. For more information, see Appendix B, *Siebel Workflows*.

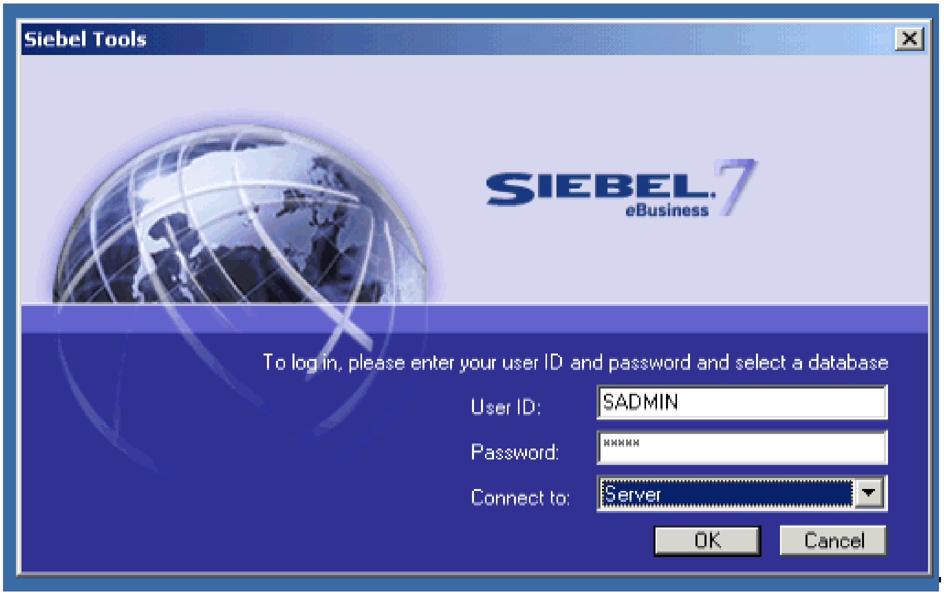
Note: Starting with version 7.5, you can generate XSD schemas using Siebel Tools. You can use these XSD files in Application Explorer.

Creating a Siebel XDR Schema

For releases prior to Siebel 6.3, the Siebel Tools Schema Wizard creates only DTD schemas. You must transform these schemas manually, or by using other tools, into XDR files before Application Explorer can use them as input to create XML schemas. In addition, you must include the SiebelMessage tag reference in your XDR file. For an example, see the Sample Account Integration Object in Appendix A, *Usage Considerations and Sample Files*.

Procedure: How to Create a Siebel XDR Schema for a Siebel Integration Object

The following image shows the Siebel Tools screen where you log on.



To log in, please enter your user ID and password and select a database

User ID: SADMIN

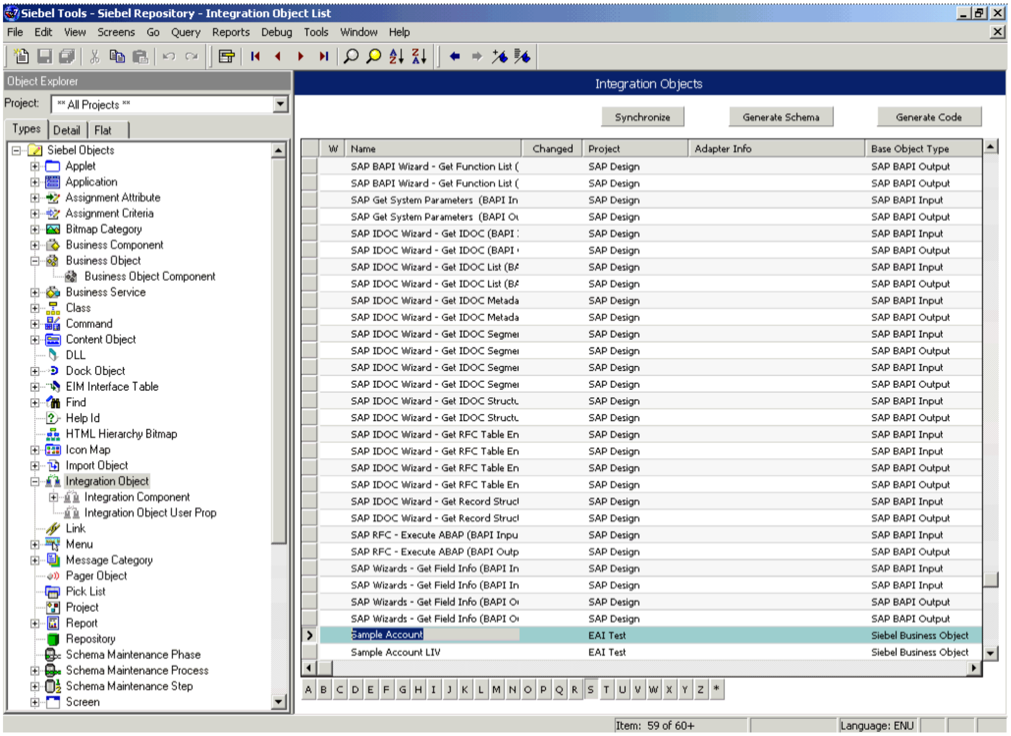
Password: *****

Connect to: Server

To generate a Siebel XDR schema:

1. Log on to Siebel Tools.
    - **a.** Type a user ID and password.
    - **b.** From the Connect to: drop-down list, select a database.
2. Click *OK*.

The following image shows the Siebel Tools window that opens. In the upper left is the Project drop-down list. In the lower left is the Object Explorer navigation pane with three tabs: Types, Detail, and Flat, with the Types tab selected to show a list of Siebel objects. The right pane has three buttons: Synchronize, Generate Schema, and Generate Code, followed by a table of Integration objects with their associated project and base object type. You can navigate by selecting a specific letter of the alphabet from a button beneath the table.

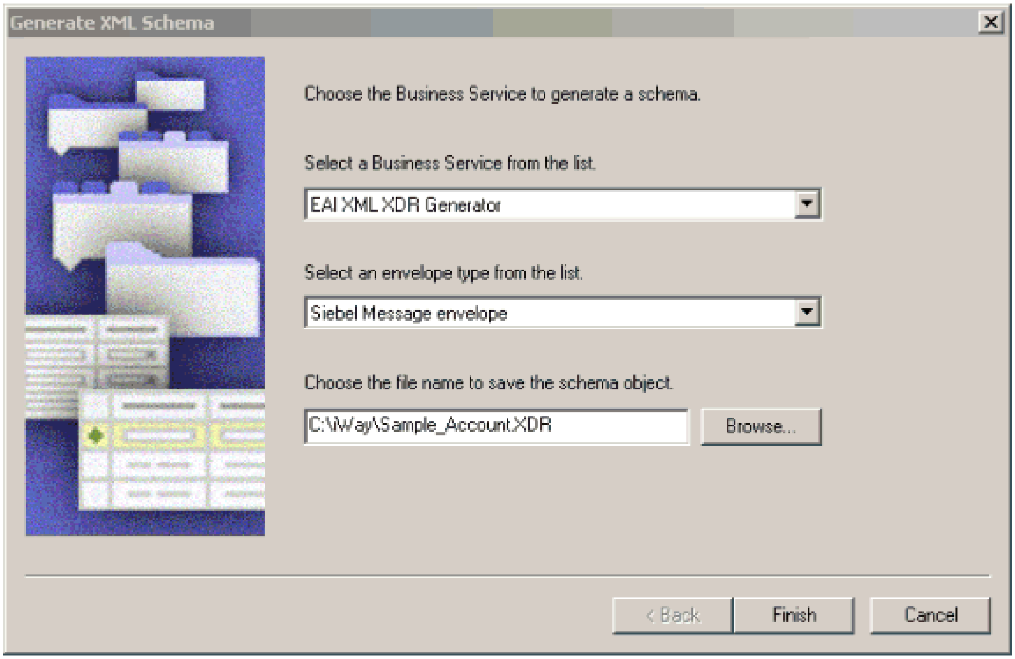


W	Name	Changed	Project	Adapter Info	Base Object Type
	SAP BAPI Wizard - Get Function List (		SAP Design		SAP BAPI Output
	SAP BAPI Wizard - Get Function List (		SAP Design		SAP BAPI Output
	SAP Get System Parameters (BAPI In		SAP Design		SAP BAPI Input
	SAP Get System Parameters (BAPI Ou		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC (BAPI :		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC (BAPI		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC List (BA		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC List (BA		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Metada		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Metada		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Segme		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Segme		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Segme		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Segme		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Structu		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Structu		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get Record Struc		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get Record Struc		SAP Design		SAP BAPI Output
	SAP RFC - Execute ABAP (BAPI Inpu		SAP Design		SAP BAPI Input
	SAP RFC - Execute ABAP (BAPI Outp		SAP Design		SAP BAPI Output
	SAP Wizards - Get Field Info (BAPI In		SAP Design		SAP BAPI Input
	SAP Wizards - Get Field Info (BAPI In		SAP Design		SAP BAPI Input
	SAP Wizards - Get Field Info (BAPI O		SAP Design		SAP BAPI Output
	SAP Wizards - Get Field Info (BAPI O		SAP Design		SAP BAPI Output
>	Sample Account		EAI Test		Siebel Business Object
	Sample Account LIV		EAI Test		Siebel Business Object

3. To create an XML schema, select an integration object, for example, Sample Account.

4. Click the *Generate Schema* button.

The following image shows the Generate XML Schema wizard window that opens where you select a Business Service to generate a schema.



Generate XML Schema

Choose the Business Service to generate a schema.

Select a Business Service from the list.

EAI XML XDR Generator

Select an envelope type from the list.

Siebel Message envelope

Choose the file name to save the schema object.

C:\iWay\Sample_Account.XDR

a. From the Business Service drop-down list, select *EAI XML XDR Generator.*

b. From the envelope type drop-down list, select *Siebel Message envelope*.

c. In the third field, click *Browse* or type to specify a file name for the XDR schema and a directory where it can be accessed by Application Explorer, for example, C:\iWay\Sample_Account.XDR.

   Note: The XDR schema file must be on the same computer as Application Explorer.

5. Click *Finish*.

   Now you can use Application Explorer to generate XML schemas for the Siebel Integration Object.

   For more information, see *How to Create a Schema For a Siebel Integration Object* on page 2-19.

Creating a Schema for a Siebel Integration Object Using Application Explorer

Application Explorer can generate schemas for Integration Objects from Siebel XDR files or it can use Siebel-generated XSD files.

If you created an XDR file using the Siebel Tools Schema Wizard, after you create the Siebel XDR schema for a selected Siebel Integration Object, you can create an XML schema using Application Explorer.

You must supply Application Explorer with the location of the previously created Siebel XDR schema for the particular integration object selected.

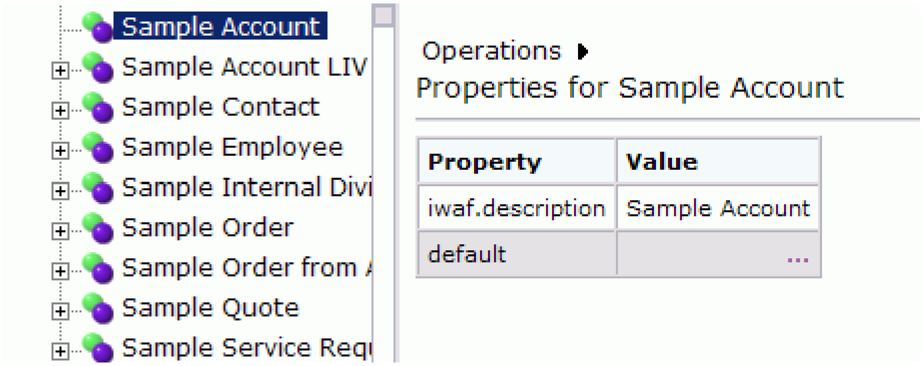
Note: The XDR file must be on the same computer as Application Explorer or be available through a mapped connection to another drive or machine.

Procedure: How to Create a Schema For a Siebel Integration Object

To create an XML schema from a Siebel XDR schema:

1. In Application Explorer, expand the *Integration Objects* node to browse the Integration Objects in the Siebel system.

   The following image shows the Integration Object nodes in the left navigation pane and a table of property details for a selected node in the right pane. Each row of the table lists a property of the object and its value.

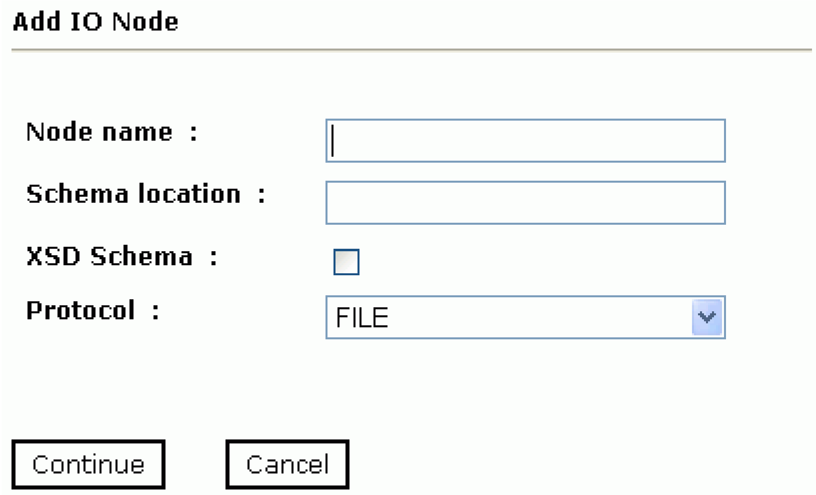


Properties for Sample Account

Property	Value
iwaf.description	Sample Account
default	...

2. Scroll down and select an integration object, for example, Sample Account.
3. To generate a schema, move the pointer over *Operations* and choose *Add IO Node*.

The following image shows the Add IO Node pane that appears on the right.



Add IO Node

Node name :

Schema location :

XSD Schema : □

Protocol : FILE

a. In the Node name field, type a name for the node to create under Sample Account.

b. In the Schema location field, type the location of the XDR or XML schema that was created by Siebel Tools. Include the name of the file in the location, for example:

```
C:\siebelrepo\SAMPLE_ACCOUNT.XDR
```

c. If you are using Siebel-generated XSD schemas, select the *XSD schema* option button.

d. From the drop-down list, choose a protocol used by the Siebel workflow for the Integration Object.

4. Click *Continue*.

For **FILE**, provide the location used by the Siebel workflow.

For **HTTP**, Application Explorer builds the URL that is the key to activating the SWE. The protocol definition has multiple sections:

```
http://my_web_server/eai_enu/start.swe?SWEExtSource=<SourceName>&SWEEx
tCmd=<Execute>&UserName=<UserName>&Password=<Password>
```

The following table lists and defines the parameters you supply for **HTTP**.

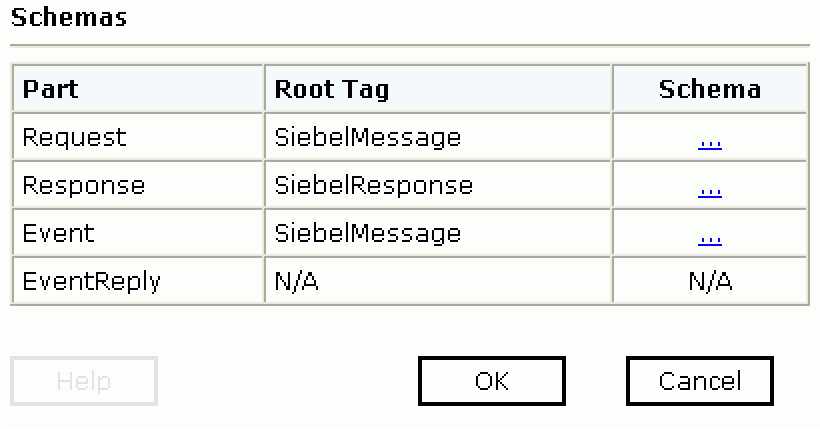
Parameter	**Definition**
SWE URL	Base SWE URL. For example `http://web_server/eai_enu/start.swe` where: *`web_server`* Is the name of the Web server hosting Siebel SWE.
SWE External Command	Section within the eai.cfg file to execute, which is the [HTTP Services] section.
SWE External Command	Use Execute.
User Name	User ID logon to execute.
Password	Logon password to execute.

The following table lists and defines the parameters you supply for **MQ Series**.

Parameter	**Definition**
Queue Manager Name	Name of the queue manager to which the server must connect.
MQ server host for MQClient operation	Host on which the MQ Server resides (MQ Client only).
MQ server port for MQClient operation	The number to connect to an MQ Server queue manager (MQ client only).
MQ server channel for MQClient operation	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is `SYSTEM.DEF.SVRCONN`
Document type XML	Keep the default selection.
respqueue	Name of the queue where messages are placed.

5. Select the node just created and then select *Generate Schema* from the Operations menu in the right pane.

The following image shows the Schemas table that appears on the right and has three columns labeled Part, Root Tag, and Schema. The Schema column provides the locations of the schemas. There are four rows: Request, Response, Event, and EventReply.

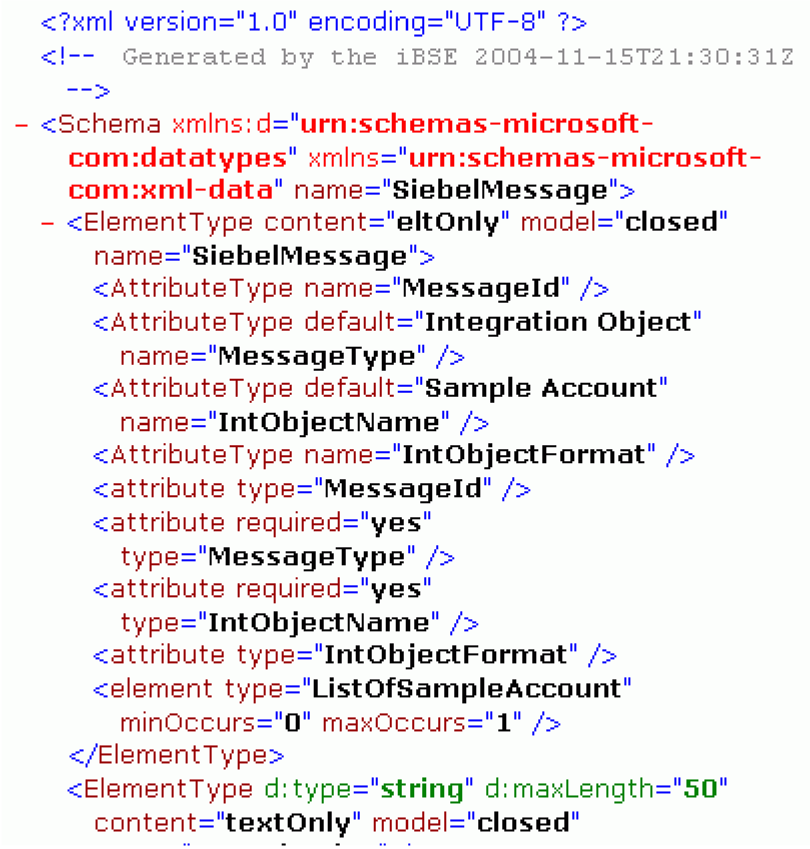


Schemas

Part	Root Tag	Schema
Request	SiebelMessage	...
Response	SiebelResponse	...
Event	SiebelMessage	...
EventReply	N/A	N/A

6. To view the XML for a schema, click the ellipsis (…) in the event row.

The following image shows the XML schema that appears in the right pane.



```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!--  Generated by the iBSE 2004-11-15T21:30:31Z
    -->
- <Schema xmlns:d="urn:schemas-microsoft-
    com:datatypes" xmlns="urn:schemas-microsoft-
    com:xml-data" name="SiebelMessage">
  - <ElementType content="eltOnly" model="closed"
      name="SiebelMessage">
      <AttributeType name="MessageId" />
      <AttributeType default="Integration Object"
        name="MessageType" />
      <AttributeType default="Sample Account"
        name="IntObjectName" />
      <AttributeType name="IntObjectFormat" />
      <attribute type="MessageId" />
      <attribute required="yes"
        type="MessageType" />
      <attribute required="yes"
        type="IntObjectName" />
      <attribute type="IntObjectFormat" />
      <element type="ListOfSampleAccount"
        minOccurs="0" maxOccurs="1" />
    </ElementType>
    <ElementType d:type="string" d:maxLength="50"
      content="textOnly" model="closed"
```

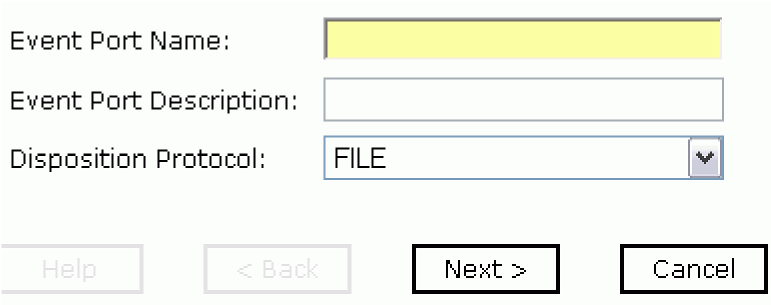
7. Click the browser *Back* button to return to the Schemas table.

A directory structure is created to store the schemas. For more information on where the schemas are stored, see *Schema Location on page 2-24*.

You are now ready to configure ports and channels or create an iWay Business Services for the Siebel Integration Object node you just created.

8. To create an event port, click the IO node name you just created and select *Create Event Port* from the Operations menu in the right pane.

The following image shows the Create Event Port pane that opens on the right. This pane includes two fields you complete to define the port: Event Port Name and Event Port Disposition. It includes the Disposition Protocol drop-down list from which you select the protocol type, a Help button, and three action buttons.



Event Port Name:

Event Port Description:

Disposition Protocol: FILE

For more information on creating event ports, see *Chapter 3, Listening for Siebel Events*.

Schema Location

When deployed to the Sun Java System Application Server and used in conjunction with an iBSE configuration, Application Explorer stores the schemas it creates in subdirectories under the Sun directory structure on the machine where the Sun Java System Application Server is installed. When used in conjunction with an iWay JCA configuration, Application Explorer stores schemas in a subdirectory of the iWay home directory.

When using the adapter with an **iBSE** configuration, the schemas are stored under a \schemas subdirectory in the Sun Java System Application Server installation directory, for example,

```
C:\SUN\AppServer\domains\domain1\applications\j2ee-apps\ibse\ibse_war\wsd
l\schemas\service\Siebel\SiebSrv
```

where:

SiebSrv

Is the name of the connection to the Siebel system as defined in Application Explorer. Under this directory, Application Explorer creates subdirectories containing schemas.

When using the adapter with a **JCA** configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

```
C:\Program Files\iWay55\config\base\schemas\Siebel\SiebServer
```

where:

SiebServer

Is the name of the connection to the Siebel system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

Returning Fields in a Specified Order

When you create a request document from an XML schema to query the Siebel system, you can limit the expected response to specific fields that are specified in the query.

The response will contain the fields in the order in which they were specified. If you do not specify a set of fields, the response document contains the entire set.

For example, the following query will return all fields:

```
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
   <m:select>
     <m:Name>Yelena*</m:Name>
   </m:select>
</m:Siebel>
```

The following query will return a response that only contains the fields Name, Location and Account Status fields:

```
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
   <m:select>
      <m:Name>Yelena*</m:Name>
   </m:select>
   <m:field>Name</m:field>
   <m:field>Location</m:field>
   <m:field>Account Status</m:field>
</m:Siebel>
```

Creating a Web Service

You can generate an iWay Business Services (also known as a Web service) for Siebel objects you wish to use with your adapter.

Ensure you properly configure the servlet iBSE. For more information on installing and deploying iWay components, see the *iWay Installation and Configuration* manual.

Before you create a Web service for an Integration Object, you must first create a Siebel XDR schema that Application Explorer can use to create an XSD schema, unless you are able to generate an XSD schema using Siebel Tools, which you can do with later versions of Siebel. For more information on creating schemas for Integration Objects, see *Creating an XML Schema for a Siebel Integration Object* on page 2-15.

Note: In a J2EE Connector Architecture (JCA) implementation of adapters, Web services are not available. When the adapters are deployed to use the connector for JCA, the Common Client Interface provides integration services using the adapters. For more information, see the *iWay Installation and Configuration* manual *and the iWay Connector for JCA User's Guide*.

Procedure: How to Generate a Web Service

To generate a Web service:

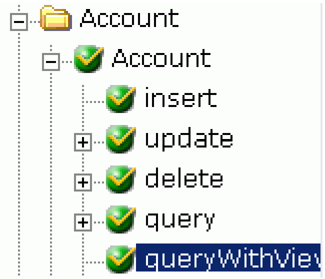
1. If you have not already connected, connect to your Siebel system.
2. Expand the Siebel node.

   For a **Siebel Business Service** or **Integration Object**, expand the node you are interested in and select the node for which you want to create a Web service.

   Note: For an Integration Object, you must first create an IO node before you create a Web service. For more information, see *Creating an XML Schema for a Siebel Integration Object* on page 2-15.

   For a **Business Component**, expand the Business Component for which you want to create a Web service and select a node.

   The following image shows the Account Business Component expanded with the queryWithView method selected.

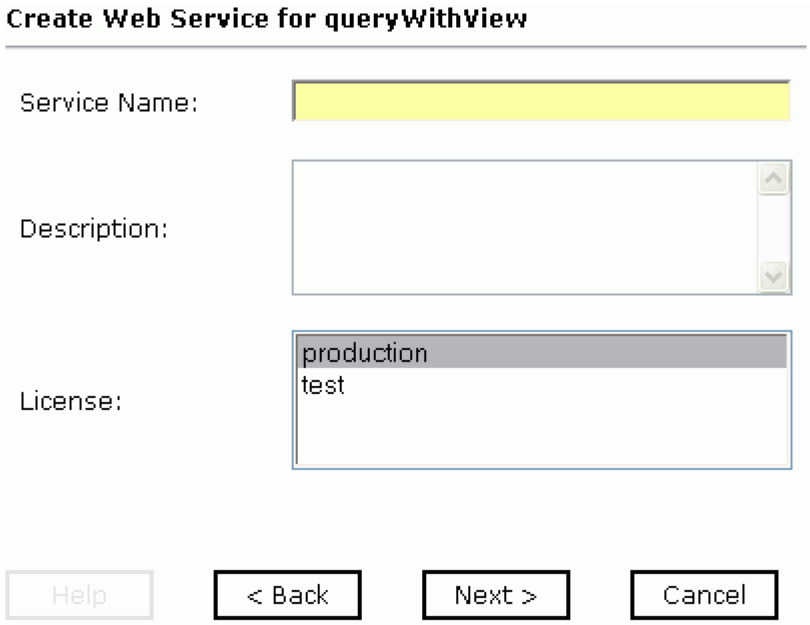
   Expand the object and select a method for creating the Web service, for example, QueryWithView under Account.

3. In the right pane, move the pointer over *Operations* and select *Create iWay Business Services*.

   If this is not the first Web service you have created, you can choose whether to create a new service or use an existing service.

   To use a **previously created service**, select the option to use an existing service and when a drop-down list appears, select the Web service to which you want to add the new service.

If this is the first Web service you are creating or if you select to **create a new service**, the Create Web Service pane appears as shown in the following image.



Create Web Service for queryWithView

Service Name:

Description:

License:

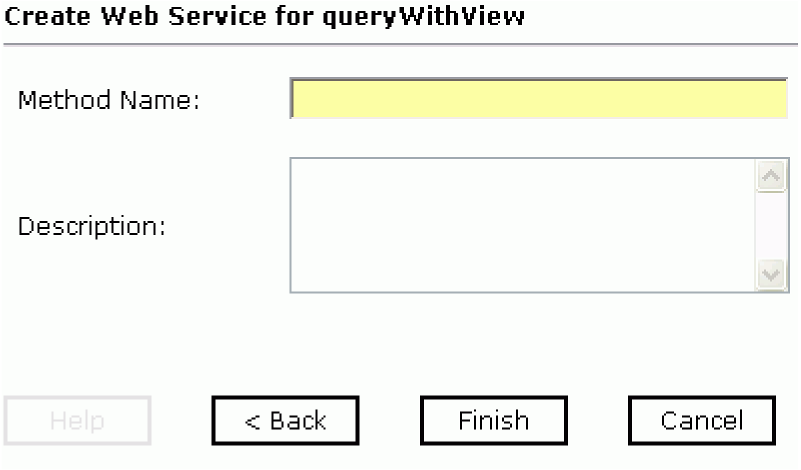
a. In the Service Name field, type a name to identify the Web service (under the Service node in the left pane of the iWay Business Services tab).

b. In the Description field, type a brief description of the Web service.

c. In the License field, select the license(s) with which you want to associate this business service. To select more than one, hold down the *Ctrl* key and click the licenses.

4. Click *Next*.

The following image shows the Create Web Service pane that reappears and prompts you for information about the method of the service.



Create Web Service for queryWithView

Method Name:

Description:

a. In the Method Name field, type a name to specify the name of the SQL statement or stored procedure to add to the business service.

b. In the Description field, type a brief description of the method.

5. Click *Finish*.

   Application Explorer switches the view to the iWay Business Services tab, and the new business service appears in the left pane.

Testing a Web Service for a Business Component or Integration Object

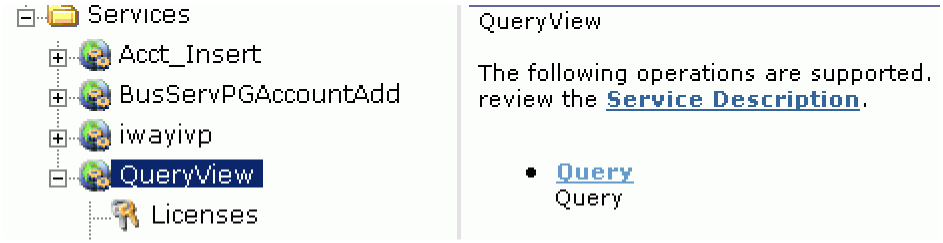
After you create a Web service for the Siebel Business Component, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

Procedure: How to Test a Web Service for a Business Component

To test a Web service for a Business Component:

1. If you are not on the iWay Business Services tab of Application Explorer, click the tab to access business services.
2. If it is not expanded, expand the *iWay Business Services* node.
3. Expand the *Services* node.
4. Select the name of the business service you want to test.

The following image shows a window with a list of services in the left pane and information about the selected service in the right pane.



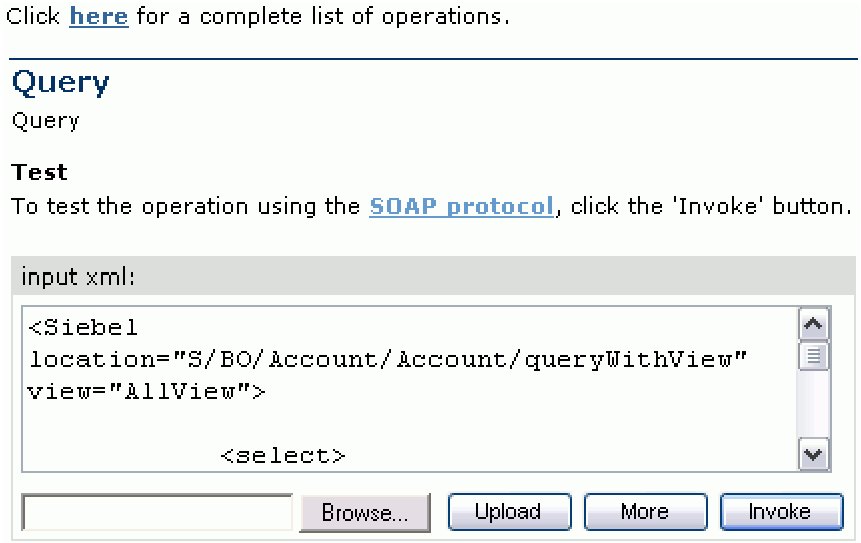
QueryView

The following operations are supported. review the Service Description.

- Query
  Query

5. In the right pane, click the named business service hyperlink, for example, Query.

   The following image shows the test option that appears in the right pane. This pane provides a text field in which to paste the XML input. Beneath it is a Browse field where you can upload a file and three action buttons.



Click here for a complete list of operations.

Query

Query

Test

To test the operation using the SOAP protocol, click the 'Invoke' button.

input xml:

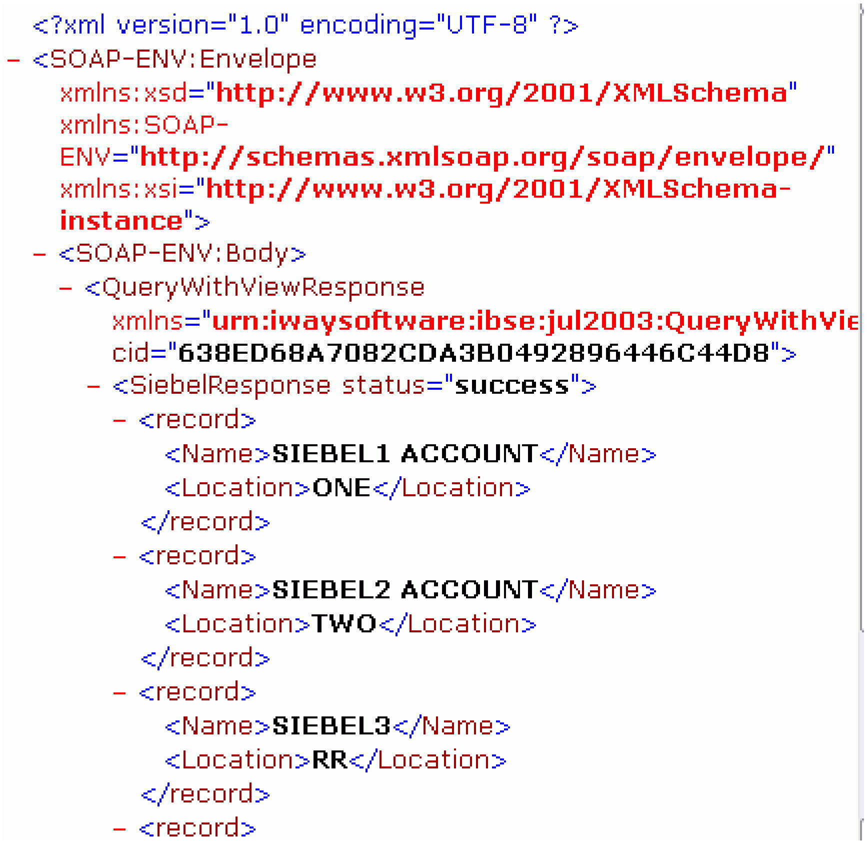
```
<Siebel
location="S/BO/Account/Account/queryWithView"
view="AllView">

            <select>
```

6. Provide the appropriate XML input.

7. Click *Invoke*.

The following image shows the XML test results that appear in the right pane.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:SOAP-
    ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance">
  - <SOAP-ENV:Body>
    - <QueryWithViewResponse
        xmlns="urn:iwaysoftware:ibse:jul2003:QueryWithVie
        cid="638ED68A7082CDA3B0492896446C44D8">
      - <SiebelResponse status="success">
        - <record>
            <Name>SIEBEL1 ACCOUNT</Name>
            <Location>ONE</Location>
          </record>
        - <record>
            <Name>SIEBEL2 ACCOUNT</Name>
            <Location>TWO</Location>
          </record>
        - <record>
            <Name>SIEBEL3</Name>
            <Location>RR</Location>
          </record>
        - <record>
```

Testing a Web Service for a Business Service

After you create a Web service for the Siebel Business Service, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

Procedure: How to Test a Web Service for a Business Service

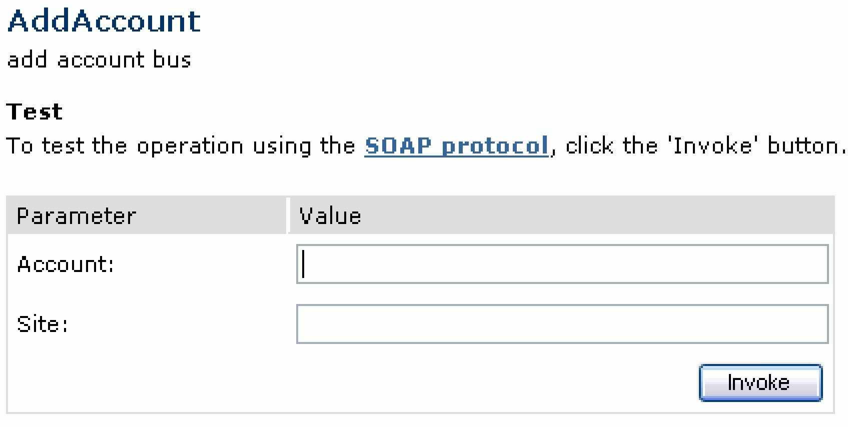
To test a Web service for a Business Service:

1. If you are not on the iWay Business Services tab of Application Explorer, click the tab to access business services.
2. If it is not expanded, expand the*iWay Business Services* node.
3. Expand the *Services* node.
4. Select the name of the business service you want to test.

The business service name appears as a link in the right pane.

5. In the right pane, click the named business services link.

   The following image shows the test option that appears in the right pane. The name of the business service appears in the upper pane. The pane has two fields for adding values for parameters.



AddAccount

add account bus

Test

To test the operation using the SOAP protocol, click the 'Invoke' button.

Parameter	Value
Account:	
Site:	

6. Provide the appropriate input.

7. Click *Invoke*.

   Application Explorer displays the results in the right pane.

Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to Siebel. The user name and password values that you provided for Siebel during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
    <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
        <m:service>String</m:service>
        <m:method>String</m:method>
        <m:license>String</m:license>
        <m:disposition>String</m:disposition>
        <m:Username>String</m:Username>
        <m:Password>String</m:Password>
        <m:language>String</m:language>
    </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, because they are not required:

```
<m:disposition>String</m:disposition>
```

```
<m:language>String</m:language>
```

CHAPTER 3

Listening for Siebel Events

Topics:

- Understanding Event Functionality
- Listening for a Siebel Integration Object Event
- Creating an Event Port
- Creating a Channel

Servlet Application Explorer deployed to Sun Java System Application Server enables you to listen for events in a Siebel system.

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces.

Understanding Event Functionality

Events are generated as a result of a specific business condition being satisfied or triggered in the Siebel system. You can use events to trigger an action in your application. For example, an update to a database can reflect an update to customer information. If your application must perform an action when this happens, your application is a consumer of this event.

After you create a connection to your application system, you can add events using Servlet Application Explorer. To create an event, you must create a port and a channel.

The following is a description of how ports and channels work:

- Port

  A port associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Creating an Event Port on page 3-3*.

- Channel

  A channel represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. For more information, see *Creating a Channel* on page 3-17.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Listening for a Siebel Integration Object Event

You can listen for Siebel Integration Object events by configuring ports and channels. Before you create the ports and channels, you must create a schema. To create XML schemas for Siebel Integration Objects, you must first generate XDR schemas using the Siebel Tools Schema Wizard.

The XDR schema is used as input to Application Explorer when generating schemas for integration objects. After you generate the XDR schema, Application Explorer uses the XDR file to generate the XML schema.

For more information on generating XDR schemas using the Siebel Tools Schema Wizard, see Chapter 2, *Creating XML Schemas and iWay Business Services*.

Creating an Event Port

To create an event port for Siebel Integration Objects, you must first indicate the location of the XDR schema for that object. For more information, see *Listening for a Siebel Integration Object Event* on page 3-2.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Creating an Event Port From the iWay Adapters Tab

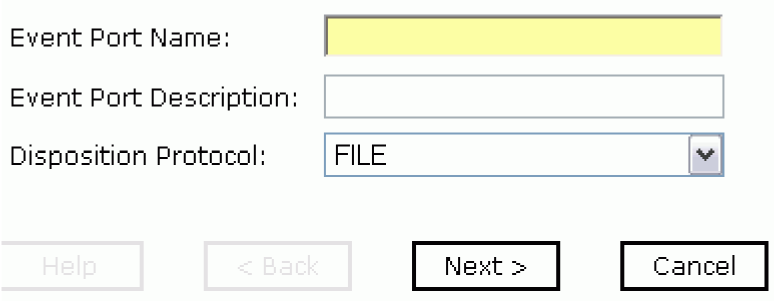
For Siebel Integration Objects, you can bypass the iWay Events tab and create an event port directly from the iWay Adapters tab.

Procedure: How to Create an Event Port From the iWay Adapters Tab

To create an event port from the iWay Adapters tab:

1. Select the Integration Object node you created.
2. Move the pointer over *Operations* and select *Create Event Port*.

   The following image shows the Create Event Port pane that opens on the right where you can create a port.



Event Port Name:

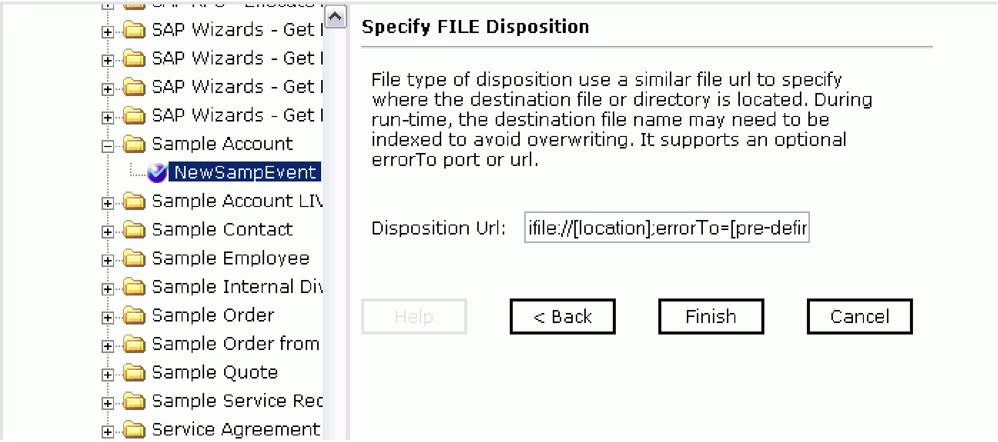
Event Port Description:

Disposition Protocol: FILE

   a. In the Event Port Name field, type a name for the port.
   b. In the Event Port Description field, provide a brief description of the port.
   c. From the Disposition Protocol drop-down list, select the required disposition, for example, FILE.

3. Click *Next*.

The following image shows the navigation pane on the left and the Specify Disposition pane that opens on the right and has information about the disposition type, a field for the disposition URL, a Help button, and three action buttons.



Specify FILE Disposition

File type of disposition use a similar file url to specify where the destination file or directory is located. During run-time, the destination file name may need to be indexed to avoid overwriting. It supports an optional errorTo port or url.

Disposition Url: ifile://[location];errorTo=[pre-defir

4. Type the disposition url and click *Finish*.

Creating an Event Port From the iWay Events Tab

The following procedures describe how to create an event port from the iWay Events window for various dispositions using Application Explorer.

The following dispositions are available when using the servlet Application Explorer in conjunction with an iBSE implementation. You can switch between an iBSE and a JCA implementation by choosing one or the other from the drop-down menu in the upper right of the Application Explorer.

- File
- iBSE
- MSMQ
- JMS queue
- SOAP
- HTTP
- MQ Series
- MAIL

Note: The MAIL disposition option will be supported in a future release.

The following dispositions are available when using Application Explorer in conjunction with a JCA connector implementation.

- File
- HTTP
- JMS queue
- MQ Series

To create an event port for Siebel Integration Objects, you must first indicate the location of the XDR schema for that object. For more information, see *Listening for a Siebel Integration Object Event* on page 3-2.

You also can create an event port directly from the iWay Adapters tab. For more information, see *Creating an Event Port From the iWay Adapters Tab* on page 3-3.

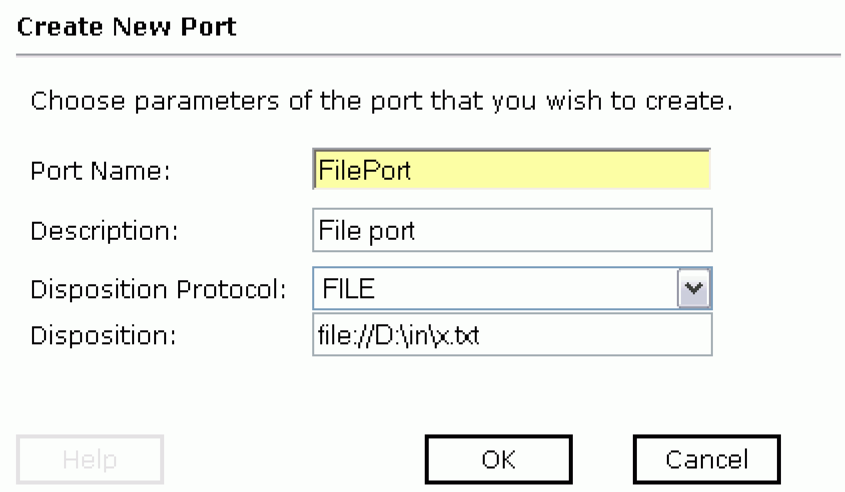
Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Procedure: How to Create an Event Port for the File Disposition

To create a specific event port for the File disposition:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *Siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

The following image shows the Create New Port pane that opens on the right.



Create New Port

Choose parameters of the port that you wish to create.

Port Name: FilePort

Description: File port

Disposition Protocol: FILE

Disposition: file://D:\in\x.txt

a. In the Port Name field, type a name for the event port.

b. In the Description field, provide a brief description of the port.

c. From the Disposition Protocol drop-down list, select *FILE*.

d. In the Disposition field, provide a destination where the event data is written.

When pointing Application Explorer to an **iBSE** deployment, use the following format:

```
ifile://[location];errorTo=[pre-defined port name or another
disposition url]
```

For example:

```
ifile://D:\in\x.txt;errorTo=ifile://D:\error
```

When pointing Application Explorer to a **JCA** deployment, provide the full path to the directory.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

e. Provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	Description
location	The destination and filename of the document where event data is written, for example, D:\in\x.txt.
errorTo	Predefined port name or another disposition URL to which error logs are sent. Optional.

5. Click *OK*.

The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-17.

Procedure: How to Create an Event Port for iBSE

You can call iWay Business Services created through the iWay Business Services Engine (iBSE).

To create an event port for iBSE:

1. Click the *iWay Events* tab.

2. In the left pane, expand the *Siebel* node.

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

The Create Event Port pane opens on the right.

a. In the Port Name field, type a name for the connection.

The name is used to build a repository entry as well as to identify the connection.

b. In the Description field, type a description for the target name you just created.

c. From the Disposition Protocol drop-down list, select *iBSE*.

d. In the Disposition field, enter an iBSE destination in the form of:

```
ibse:svcName.mthName;responseTo=[pre-defined port name or another
disposition url];errorTo=[pre-defined port name or another
disposition url]
```

e. Provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	Description
svcName	Name of the service created with iBSE.
mthName	Name of the method created for the Web service.
responseTo	Location where responses to the Web service are posted. A predefined port name or another full URL. Optional.
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

5. Click *OK*.

   The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-17.

Procedure: How to Create an Event Port for a JMS Queue

To create an event port for a Sun Java System Message Queue:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *Siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create Event Port pane opens on the right.

   a. Type a name for the event port and provide a brief description.

   b. From the Disposition Protocol drop-down list, select *JMSQ*.

   c. In the Disposition field, enter a JMS destination.

      When pointing Application Explorer to an **iBSE** deployment, use the following format:

```
jmsq:myQueueName@myQueueFac;jndiurl=[myurl];jndifactory=[myfactory
];user=[user];password=[xxx];errorTo=[pre-defined port name or
another disposition url]
```

When pointing Application Explorer to a **JCA** deployment, use the following format:

```
jms:jmsqueue@jmsfactory;jndiurl=;jndifactory=;
```

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

d. Provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	**Description**
queue	JNDI name of a queue to which events are emitted.
Connection Factory	A resource that contains information about the JMS Server. You must create the connection factory, for example: `sampleQCF`

jndiurl	The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property,
	`java.naming.provider.url`
	The URL of the Sun Java System Application Server is
	`iiop://localhost:3700`
	where:
	`3700`
	Is a default port.

jndifactory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For Sun Java System Application Server, this is `com.sun.jndi.cosnaming.CNCtxFactory`
user	A valid user name required to access a JMS server.
password	A valid password required to access a JMS server.
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

5. Click *OK*.

   The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

   You are now ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-17.

Procedure: How to Create an Event Port for MSMQ

To create an event port for MSMQ:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create Event Port pane opens on the right.

   a. In the Port Name field, type a name for the connection, for example, Queue1_on_NTK.

      The name is used to build a repository entry as well as to identify the connection.

   b. In the Description field, type a description for the target name you just created.

   c. From the Disposition Protocol drop-down list, select *MSMQ*.

   d. In the Disposition field, enter a MSMQ destination in the form of:

```
msmq:/host/private$/qName;errorTo=[pre-defined port name or
another disposition url]
```

   e. Provide values for the parameters from the following table.

      The following table includes the name and description of each parameter.

Parameter	Description
host	Machine name where the Microsoft Queuing system is running.
Queue Type	Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue. For private queues, enter *Private$*.

Parameter	Description
qName	Name of the private queue where messages are placed.
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

5. Click *OK*.

   The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-17.

Procedure: How to Create a Port for a SOAP Disposition

To create a port for a SOAP disposition:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *Siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port pane opens on the right.

   a. In the Port Name field, type a name for the event.

   b. In the Description field, type a brief description.

   c. From the Disposition Protocol drop-down list, select *SOAP*.

   d. In the Disposition field, enter a SOAP destination, using the following format:

   ```
   soap:[wsdl-url];soapaction=[myaction];method=[web service
   method];namespace=[namespace];responseTo=[pre-defined port name or
   another disposition URL];errorTo=[pre-defined port name or another
   disposition url]
   ```

   e. Provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	Description
wsdl-url	The URL to the WSDL file that is required to create the SOAP message. For example: http://localhost:7001/ibse/IBSEServlet/test/*webservice*.ibs?wsdl where: *webservice* Is the name of the Web service you created using Application Explorer. This value can be found by navigating to the iWay Business Services tab and opening the Service Description link in a new window. The WSDL URL appears in the Address field. You also can open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.
soapaction	The method that will be called by the SOAP disposition. For example: *webservice.method@test@@* where: *webservice* Is the name of the Web service you created using Application Explorer. *method* Is the method being used. *test* Is the license that is being used by the Web service. This value can be found by navigating to the iWay Business Services tab, opening the Service Description link in a new window, and performing a search for soapAction. You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.

Parameter	**Description**
method	The Web service method you are using. This value can be found in the WSDL file.
namespace	The XML namespace you are using. This value can be found in the WSDL file.
responseTo	The location to which responses are posted, which can be a predefined port name or another URL. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

   The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

Procedure: How to Create an Event Port for an HTTP Disposition

To create an event port for an HTTP disposition:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create Event Port pane opens on the right.

   a. Type an event port name and a brief description.

   b. From the disposition protocol drop-down list, select *HTTP*.

   c. From the Disposition field, enter an HTTP destination.

      When pointing Application Explorer to an **iBSE** deployment, use the following format:

      ```
      ihttp://[myurl];responseTo=[pre-defined port name or another
      disposition url];
      ```

where:

`url`

Is the URL target for the post operation, for example `http://myhost:1234/docroot`

`responseTo`

Is the location where responses are posted (optional).

When pointing Application Explorer to a **JCA** deployment, use the following format:

```
http://host:port/uri
```

where:

`host:port`

Is the combination of the name of the host on which the Web server resides and the port on which the server is listening for the post operation.

`uri`

Is the universal resource identifier that completes the url specification.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

5. Click *OK*.

   The port appears under the ports node in the left pane.

Procedure: How to Create an Event Port for an MQSeries Disposition

To create an event port for an MQSeries disposition using Application Explorer:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *siebel* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create Event Port pane opens on the right.

a. Type an event port name and a brief description.

b. From the disposition protocol drop-down list, select *MQSeries*.

c. In the Disposition field, enter an MQSeries destination.

   When pointing Application Explorer to an **iBSE** deployment, use the following format:

   ```
   mqseries:/qManager/qName;host=[hostname];port=[port];channel=[chan
   nnelname];errorTo=[pre-defined port name or another disposition
   url]
   ```

   When pointing Application Explorer to a **JCA** deployment, use the following format:

   ```
   mq:qmanager@respqueue;host=;port=;channel=
   ```

   Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

d. Provide values for the parameters from the following table.

   The following table includes the name and description of each parameter.

Parameter	Description
qManager	Is the name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	The host on which the MQ Server is located (MQ Client only).
port	The number to connect to an MQ Server queue manager (MQ client only).
channel	The case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. This can be a predefined port name or another full URL. Optional.

5. Click *OK*.

The newly created event port appears under the port section of the event adapter in the left pane.

Editing or Deleting an Event Port

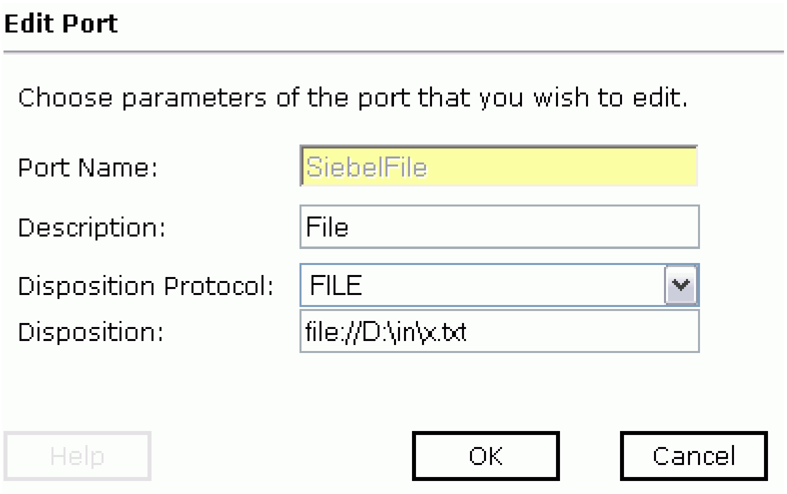
The following procedures provide information on how to modify or delete an event port.

Procedure: How to Edit an Event Port

To edit an existing event port:

1. In the left pane, select the event port you want to edit.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

   The following image shows the Edit Port pane that opens on the right. It includes a field for typing the port name, a field for typing a description, a drop-down list for selecting the disposition protocol, and a field for specifying the disposition. It also includes a Help button and two action buttons.

   Edit Port

   Choose parameters of the port that you wish to edit.

   Port Name: SiebelFile

   Description: File

   Disposition Protocol: FILE

   Disposition: file://D:\in\x.txt

3. Make the required changes to the Description, Disposition Protocol, or Disposition fields, and click *OK*.

   Note: The Edit Port pane does not allow you to change the name of the port, only the parameters.

Procedure: How to Delete an Event Port

To delete an existing event port:

1. Select the event port you want to delete.

2. In the right pane, move the pointer over *Operations* and select *Delete*.

   A confirmation dialog box opens.

3. To delete the event port you selected, click *OK*.

   The event port disappears from the list in the left pane.

Creating a Channel

The following procedures describe how to create a channel for your event. All defined event ports must be associated with a channel. You can create three types of channels:

- HTTP
- File
- MQ Series

Procedure: How to Create an HTTP Channel

To create an HTTP channel using Application Explorer:

1. Click the *iWay Events* tab.

   The adapters in the left pane support events.

2. Expand the *siebel* node.

   The ports and channels nodes appear in the left pane.

3. Click the *channels* node.

4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

   a. When the Add a new channel window opens, type a name for the channel, for example, NewChannel.

   b. Type a brief description.

   c. From the drop-down list, select *HTTP Listener*.

5. Click *Next*.

   The Edit Channels window opens in the right pane.

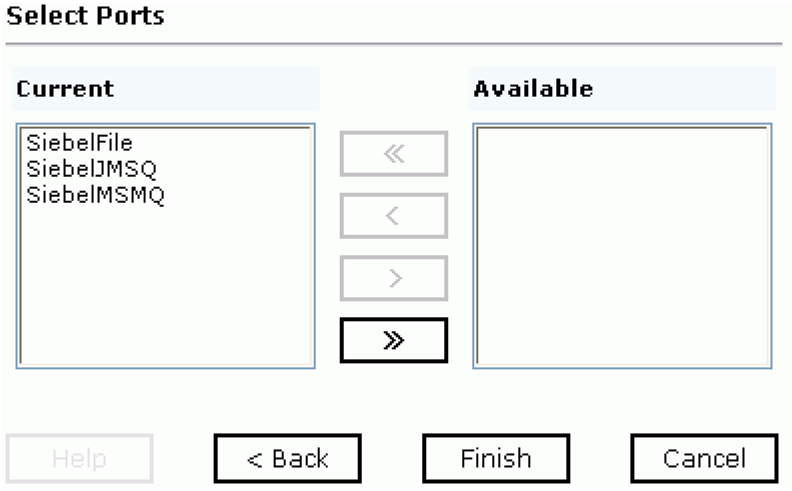
6. Provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	Description
Listener port	Port on which to listen for Siebel event data.
Synchronization Type	Choose from three options: • REQUEST • REQUEST_RESPONSE • REQUEST_ACK

7. Click *Next*.

The following image shows the Select Ports pane that opens with a list of the ports that are currently associated in the Current field on the left. On the right is a list of available ports in the Available field. The pane also includes arrow buttons, a Help button, and three action buttons.

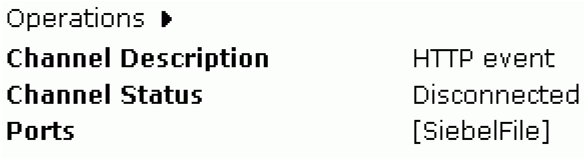


a. Select an event port from the list of current ports.

b. Click the single right arrow button to transfer the port to the list of available ports. To associate all the event ports, click the double right arrow button.

8. Click *Finish*.

The following image is an example of the summary window that opens, showing a description of the channel, its status, and available ports.

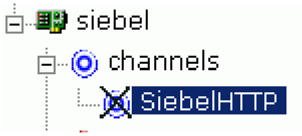


Operations ▶

Channel Description	HTTP event
Channel Status	Disconnected
Ports	[SiebelFile]

All the information in the summary is associated with the channel you created.

The following image shows a channel that appears beneath the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.



9. In the right pane, move the pointer over *Operations* and select *Start the channel*.

   The channel you created becomes active.

   The X that was over the icon disappears.

10. To stop the channel, move the pointer over *Operations* and select *Stop the channel*.

Procedure: How to Create a File Channel

To create a channel using Application Explorer:

1. Click the *iWay Events* tab.

   The adapters in the left pane support events.

2. Expand the *siebel* node.

   The ports and channels nodes appear in the left pane.

3. Click the *channels* node.

4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

   The Add a new channel window opens.

   a. Type a name for the channel, for example, NewChannel.

   b. Type a brief description.

   c. From the drop-down list, select *File Listener*.

5. Click *Next*.

The Edit Channels window opens with three tabs in the right pane.

a. In the Request tab, provide values for the parameters from t.

The following table includes the name and description of each parameter.

Parameter	**Description**
Polling Location	The target file system location for the Siebel XML file.
File Mask	The file name to be used for the output file generated as a result of this operation.

b. In the Response tab, provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	**Description**
Synchronization Type	The target file system location for the Siebel XML file.
Response/Ack Directory	Choose from three options: • REQUEST • REQUEST_RESPONSE • REQUEST_ACK

c. In the Advanced tab, provide values for the parameters from the following table.

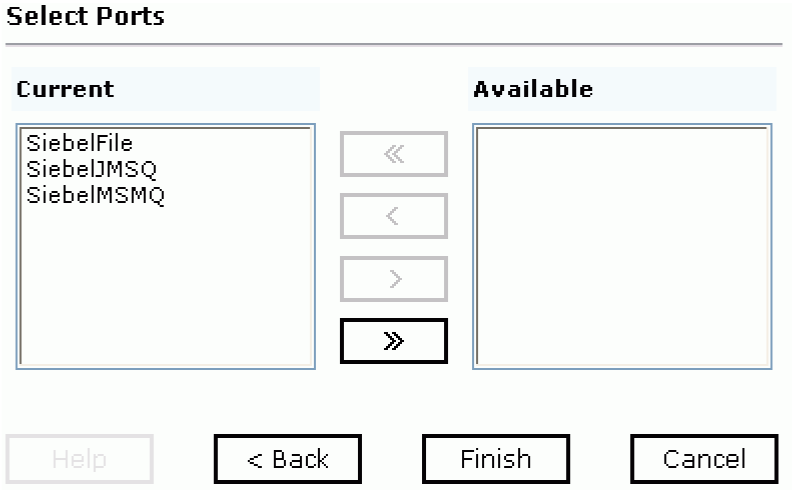
The following table includes the name and description of each parameter.

Parameter	**Description**
Error Directory	Directory to which documents with errors are written.
Poll interval (msec):	The interval (in milliseconds) when to check for new input. Optional. The default is 3 seconds.

Parameter	**Description**
Processing Mode	Choose Sequential or Threaded. • Sequential indicates single processing of requests. • Threaded indicates processing of multiple requests simultaneously.
Thread limit	If you selected threaded processing, indicate the maximum number of requests that can be processed simultaneously.

6. Click *Next*.

   The following image shows the Select Ports pane that opens with a list of the ports that are currently associated in the Current field on the left. On the right is a list of available ports in the Available field. The pane also includes arrow buttons, a Help button, and three action buttons.



Select Ports

Current

SiebelFile
SiebelJMSQ
SiebelMSMQ

Available

   a. Select an event port from the list of current ports.

   b. Click the single right arrow button to transfer the port to the list of available ports. To associate all the event ports, click the double right arrow button.

7. Click *Finish*.

   The summary window opens. A summary provides the channel description, channel status, and available ports. All the information is associated with the channel you created.

The channel appears under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

8. In the right pane, move the pointer over *Operations* and select *Start the channel*.

The channel you created becomes active.

The X that was over the icon disappears.

9. To stop the channel, move the pointer over *Operations* and select *Stop the channel*.

Procedure: How to Create an MQSeries Channel

To create an MQSeries channel using Application Explorer:

1. Click the *iWay Events* tab.

The adapters in the left pane support events.

2. Expand the *siebel* node.

The ports and channels nodes appear in the left pane.

3. Click the *channels* node.

4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

The Add a new channel window opens.

a. Type a name for the channel, for example, NewChannel.

b. Type a brief description.

c. From the drop-down list, select *MQ Series Listener*.

5. Click *Next*.

The Edit Channels panes opens on the right and has three tabs.

a. In the Request tab, provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	Description
Queue manager name	Name of the queue manager to which the server must connect.
MQ server host for MQClient operation	Host on which the MQ Server is located (MQ Client only).

Parameter	**Description**
MQ server port for MQClient operation	The number to connect to an MQ Server queue manager (MQ client only).
MQ server channel for MQClient operation	The case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
Document type XML	Keep the default selection.

Request queue name	Queue where the message is routed and where request documents are received. The name of the queue is case-sensitive and conforms to the following format:
	`Host\queue type$\qName`
	where:
	`Host`
	Is the machine name where the MQSeries queuing system is running.
	`queue type`
	Private queues are queues that are not published in Active Directory and appear only on the local computer where they reside. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
	`qName`
	Is the name of the queue where messages are placed, for example,
	`iwaykxc1\Private$\siebel`

b. In the Response tab, provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	**Description**
Synchronization Type	Choose from three options: • REQUEST • REQUEST_RESPONSE • REQUEST_ACK

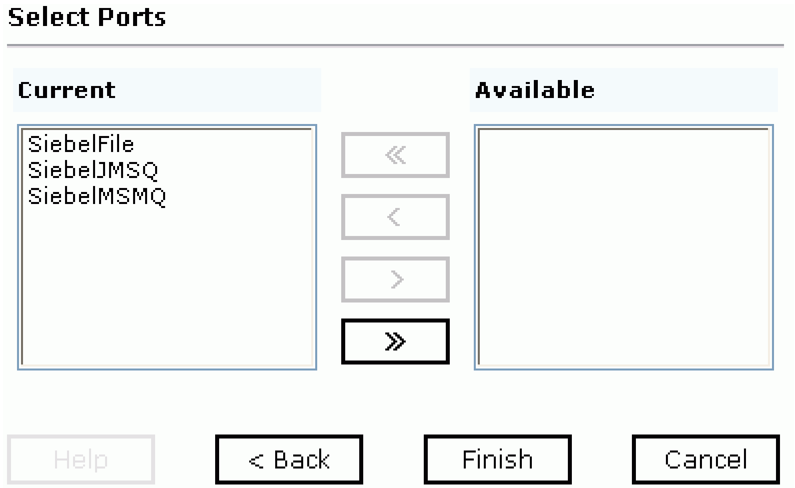
c. In the Advanced tab, provide values for the parameters from the following table.

The following table includes the name and description of each parameter.

Parameter	**Description**
Error Directory	Directory to which documents with errors are written.
Message wait interval (msec):	The interval (in milliseconds) when to check for new input Optional. The default is 3 seconds.
Mode of operation	Choose Sequential or Threaded. • Sequential indicates single processing of requests. • Threaded indicates processing of multiple requests simultaneously.
Thread limit	If you selected threaded processing, indicate the maximum number of requests that can be processed simultaneously.

6. Click *Next*.

The following image shows the Select Ports pane that opens with a list of the ports that are currently associated in the Current field on the left. On the right is a list of available ports in the Available field. The pane also includes arrow buttons, a Help button, and three action buttons.



a. Select an event port from the list of current ports.

b. Click the single right arrow button to transfer the port to the list of available ports. To associate all the event ports, click the double right arrow button.

7. Click *Finish*.

   The summary window opens. A summary provides the channel description, channel status, and available ports. All the information is associated with the channel you created.

   The channel appears under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

8. In the right pane, move the pointer over *Operations* and select *Start the channel*.

   The channel you created becomes active.

   The X that was over the icon disappears.

9. To stop the channel, move the pointer over *Operations* and select *Stop the channel*.

Procedure: How to Edit a Channel

To edit an existing channel:

1. In the left pane, select the channel you want to edit.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

   The Edit channels window opens.
3. Make the required changes to the channel configuration and click *Finish*.

Procedure: How to Delete a Channel

To delete an existing channel:

1. In the left pane, select the channel you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

   A confirmation dialog box opens.
3. To delete the channel you selected, click *OK*.

   The channel disappears from the list in the left pane.

CHAPTER 4

Using Web Services Policy-Based Security

Topics:

- iWay Business Services Policy-Based Security
- Configuring iWay Business Services Policy-Based Security

Servlet Application Explorer provides a security feature called iWay Business Services policy-based security. The following topics describe how this feature works and how to configure it.

iWay Business Services Policy-Based Security

iWay Business Services provide a layer of abstraction between the back-end business logic they invoke and the user or application running the business service. This enables easy application integration but raises the issue of controlling the use and execution of critical and sensitive business logic that is run as a business service.

Servlet Application Explorer controls the use of business services that use adapters with a feature called policy-based security. This feature enables an administrator to apply *policies* to iWay Business Services (iBS) to deny or permit their execution.

A *policy* is a set of privileges associated with the execution of a business service that can be applied to an existing or new iBS. When you assign specific rights or privileges inside a policy, you need not recreate privileges for every iBS that has security issues in common with other iWay Business Services. Instead, you can use one policy for many iWay Business Services.

The goal is to secure requests at both the transport and the SOAP request level that is transmitted on the wire. Some policies do not deal with security issues directly but affect the run-time behavior of the business services to which they are applied.

The iBSE administrator creates an instance of a policy type, names it, associates individual users and/or groups (a collection of users), and then applies the policy to one or more business services.

You can assign a policy to an iBS or to a method within an iBS. If a policy is applied only to a method, other methods in that iBS are not governed by it. However, if a policy is applied to the iBS, all methods are governed by it. At run time, the user ID and password that are sent to iBSE in the SOAP request message are checked against the list of users for all policies applied to the specific iBS. The Resource Execution policy type is supported and dictates who can or cannot execute the iBS.

When a policy is not applied, the default value for an iBS is to “grant all.” For example, anyone can execute the iBS until the Resource Execution policy is associated to the iBS. At that time, only users granted execution permission, or those who do not belong to a group that was denied execution permissions, have access to the iBS.

Configuring iWay Business Services Policy-Based Security

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Servlet Application Explorer. For more information, see *How to Create a User to Associate With a Policy* on page 4-3 or *How to Create a Group to Associate With a Policy* on page 4-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *How to Create an Execution Policy* on page 4-7.

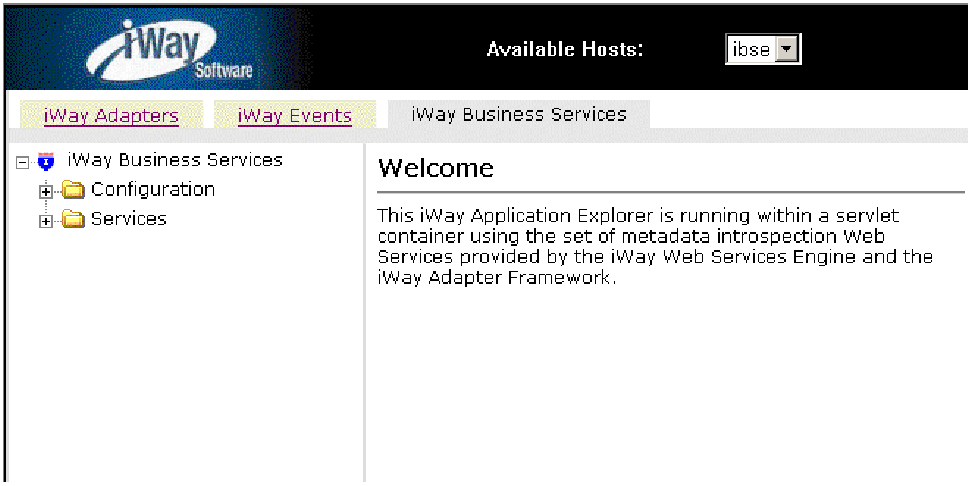
You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 4-11.

Procedure How to Create a User to Associate With a Policy

To create a user to associate with a policy:

1. Open Servlet Application Explorer.

   The following image shows the window that opens and includes three tabs corresponding to iWay Adapters, iWay Events, and iWay Business Services. The iWay Business Services tab is active and displays a Welcome screen on the right. The image shows the iWay Business Services node expanded in the left pane.



Welcome

This iWay Application Explorer is running within a servlet container using the set of metadata introspection Web Services provided by the iWay Web Services Engine and the iWay Adapter Framework.

   a. Click the *iWay Business Services* tab.

   b. Expand the *Configuration* node.

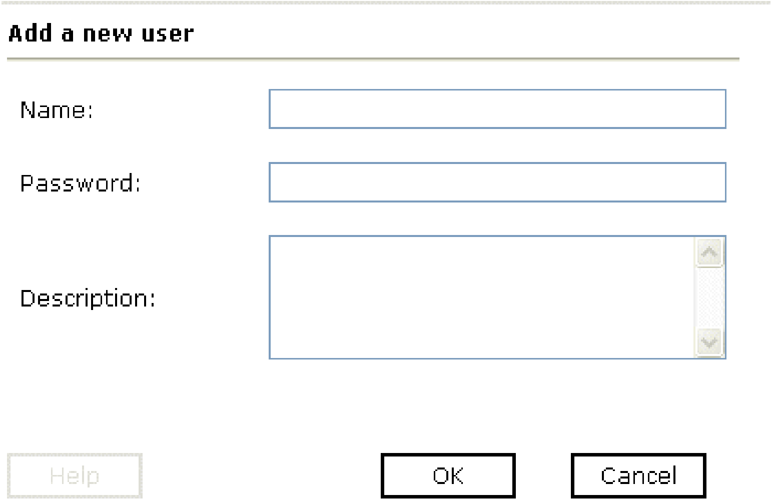
   c. Expand the *Security* node.

   d. Expand the *Users and Groups* node.

   e. Select *Users*.

2. In the right pane, move the pointer over *Operations* and select *Add*.

   The following image shows the Add a new user pane that opens and includes fields where you enter a user name, a password, and a description of the user. The pane includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.



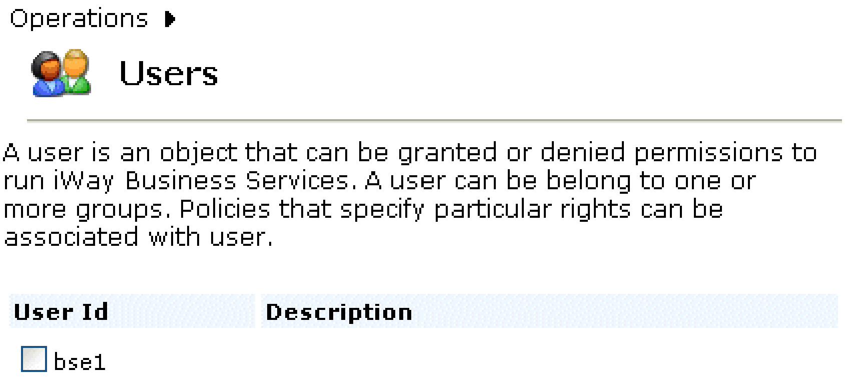
   a. In the Name field, type a user ID.

   b. In the Password field, type the password associated with the user ID.

   c. In the Description field, type a description of the user (optional).

3. Click *OK*.

The following image opens and shows a new user added to the configuration. It includes a definition of a user and a user ID and description.



A user is an object that can be granted or denied permissions to run iWay Business Services. A user can be belong to one or more groups. Policies that specify particular rights can be associated with user.

User Id	Description
□ bse1	

Procedure How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

1. Open Servlet Application Explorer.

   a. Click the *iWay Business Services* tab.

   b. Expand the *Configuration* node.

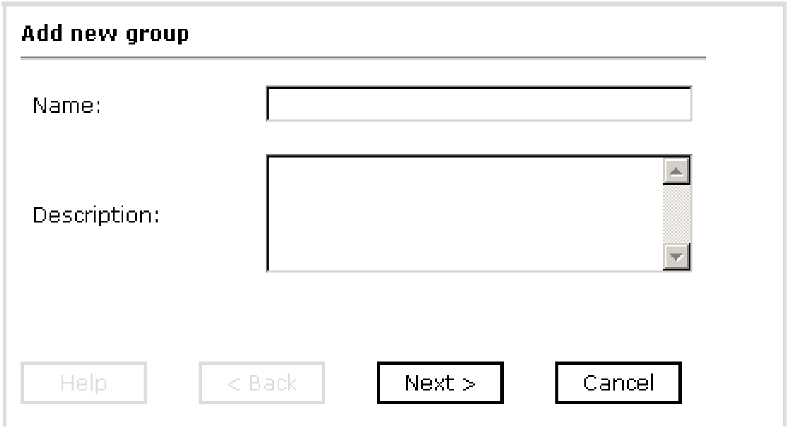
   c. Expand the *Security* node.

   d. Expand the *Users and Groups* node.

   e. Select *Groups*.

2. In the right pane, move the pointer over *Operations* and click *Add*.

The following image shows the Add new group pane that opens with fields where you enter a name and a description for the group. To continue after typing inputs, click the *Next* button. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.

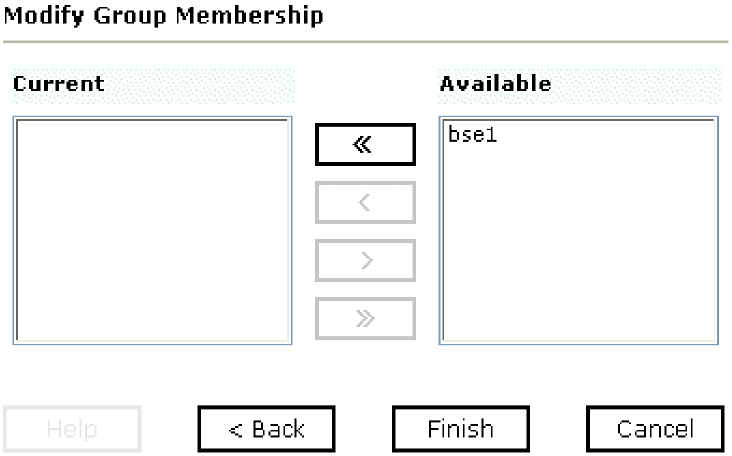


a. In the Name field, type a a name for the group.

b. In the Description field, type a description for the group (optional).

3. Click *Next*.

The following image shows the Modify Group Membership pane where you can move users to or from a group using the arrow keys to move them between the Current and Available lists and then clicking the Finish button. The pane includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.

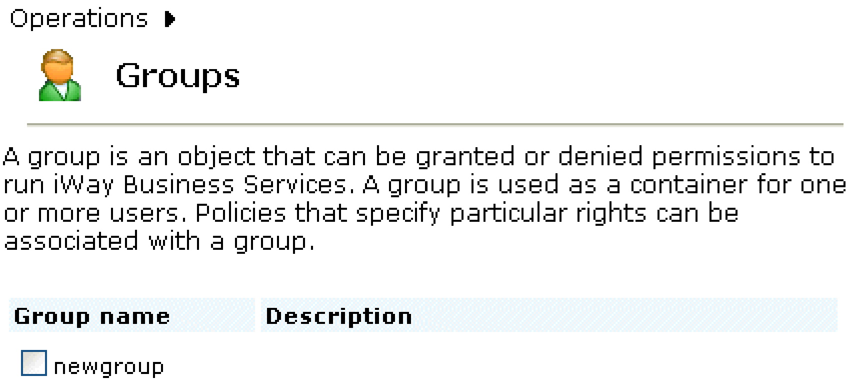


You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

4. After you select a minimum of one user, click *Finish*.

   The new group is added.

   The following image shows a pane with a new group added to the configuration. It includes a definition of a group and the group name and description.



Groups

A group is an object that can be granted or denied permissions to run iWay Business Services. A group is used as a container for one or more users. Policies that specify particular rights can be associated with a group.

Group name	Description
□ newgroup	

Procedure How to Create an Execution Policy

To create an execution policy:

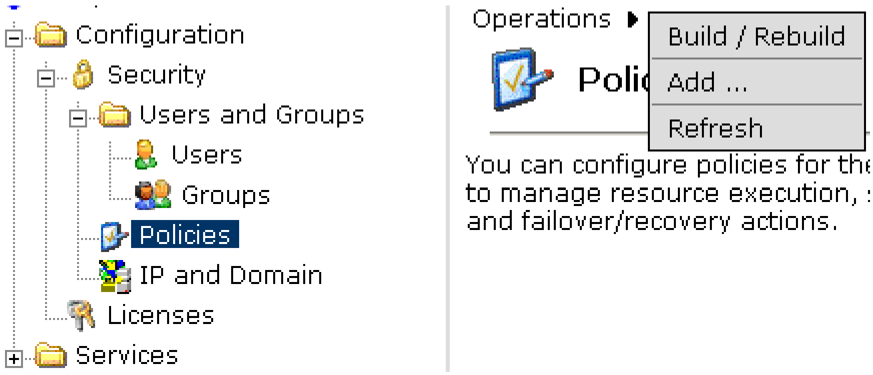
1. Open Servlet Application Explorer.

   a. Click the *iWay Business Services* tab.

   b. Expand the *Configuration* node.

   c. Select *Policies*.

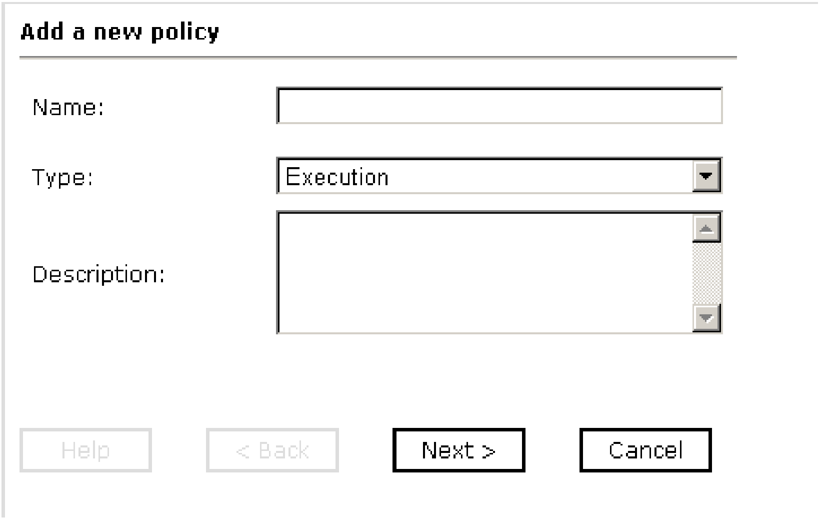
The following image shows the Policies pane on the right where you apply a policy. The Operations menu becomes available with three options, Build/Rebuild, Add, and Refresh.



You can configure policies for the
to manage resource execution,
and failover/recovery actions.

2. Move the pointer over *Operations* and click *Add*.

The following image shows the Add a new policy pane that opens with fields for entering the name, type, and description of the policy. To continue, click the *Next* button. The pane includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.



Add a new policy

Name:

Type: Execution

Description:

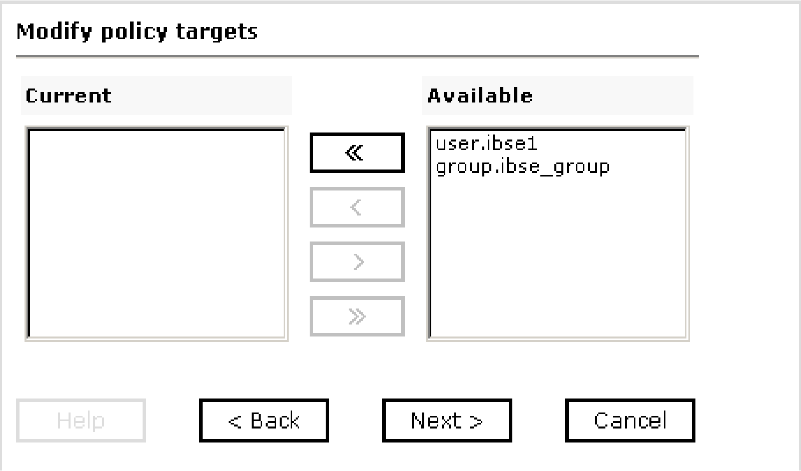
a. In the Name field, type a a name for the policy.

b. From the Type drop-down list, select *Execution*.

c. In the Description field, type a description for the policy (optional).

3. Click *Next*.

The following image shows the Modify policy targets pane that opens and includes a list of current and available targets and arrow buttons to move targets from one list to the other. The pane also includes a Help button, a Back button to return to the previous screen, a Next button to continue to the next screen, and a Cancel button to escape from the pane.



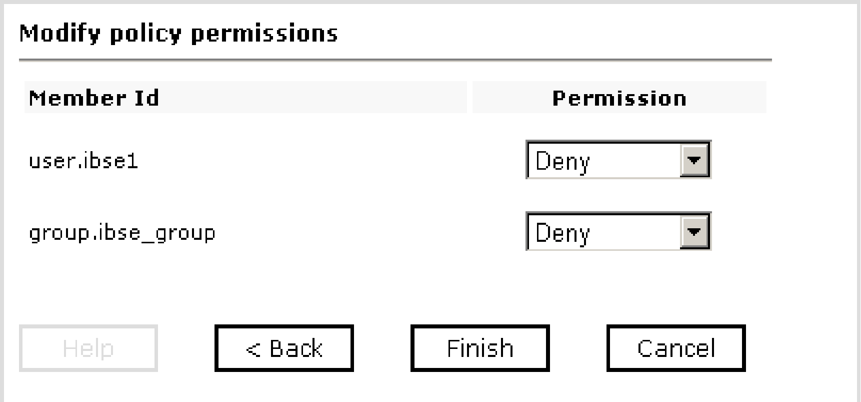
user.ibse1
group.ibse_group

4. Select a minimum of one user or group from the Available pane.

Note: This user ID is checked against the value in the user ID element of the SOAP header sent to iBSE in a SOAP request.

5. Click *Next*.

The following image shows the Modify policy permissions pane that opens and includes drop-down lists where you can select to grant or deny permission to members and then click a button to finish. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.



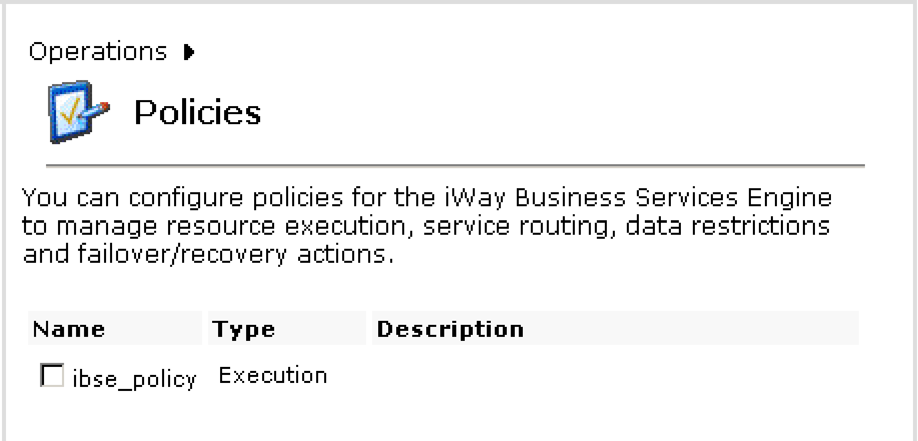
Modify policy permissions

Member Id	Permission
user.ibse1	Deny
group.ibse_group	Deny

6. To assign whether users or groups may execute the iBSE, select *Grant* to permit execution or *Deny* to restrict execution from a Permission drop-down list.

7. Click *Finish*.

   The following image shows the pane that summarizes your configuration. It includes a definition of policies and the name, type, and description of the policies.



Operations ▸

Policies

You can configure policies for the iWay Business Services Engine to manage resource execution, service routing, data restrictions and failover/recovery actions.

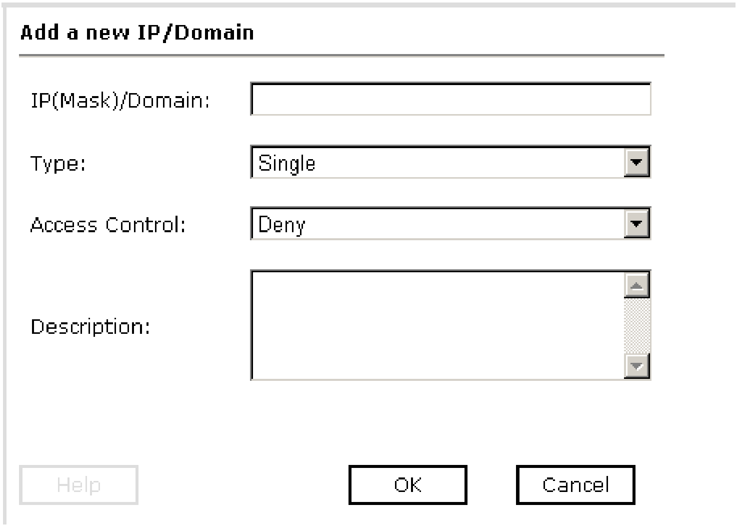
Name	Type	Description
□ ibse_policy	Execution	

Procedure How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

1. Open Servlet Application Explorer.
   - a. Select the *iWay Business Services* tab.
   - b. Expand the *Configuration* node.
   - c. Expand the *Security* node.
   - d. Select *IP and Domain*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

   The following image shows the Add a new IP/Domain pane that opens where you enter information for the IP/Domain in four fields. You must select a type of restriction from a drop-down list before you can enter information in the IP(Mask)/Domain field. The pane also includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.



Add a new IP/Domain

IP(Mask)/Domain:

Type: Single

Access Control: Deny

Description:

   - a. From the Type drop-down list, select the type of restriction.
   - b. In the IP(Mask)/Domain field, type the IP or domain name using the following guidelines.

     If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

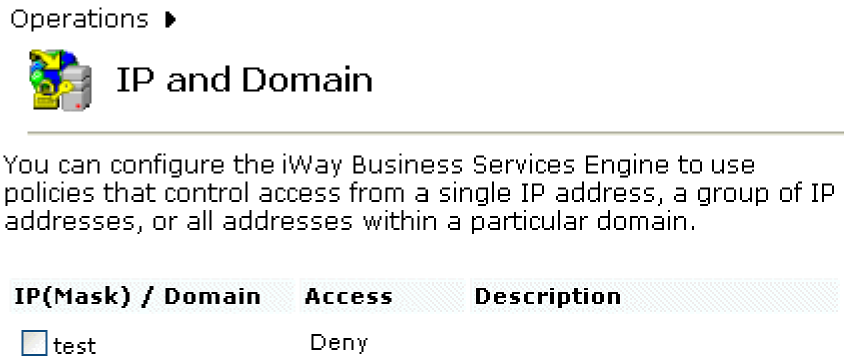
If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.

If you select Domain, you must provide the domain name, for example, yahoo.com.

3. From the Access Control drop-down list, select *Grant* to permit access or *Deny* to restrict access for the IP addresses and domain names you are adding.

4. Click *OK*.

   The following image shows the pane that opens and summarizes your configuration including the domain name, whether access is granted or denied, and a description (optional).



IP and Domain

You can configure the iWay Business Services Engine to use policies that control access from a single IP address, a group of IP addresses, or all addresses within a particular domain.

IP(Mask) / Domain	Access	Description
□ test	Deny	

CHAPTER 5

Management and Monitoring

Topics:

- Managing and Monitoring Services and Events Using iBSE
- Managing and Monitoring Services and Events Using the JCA Test Tool
- Setting Engine Log Levels
- Migrating Repositories
- Exporting or Importing Targets
- Retrieving or Updating Web Service Method Connection Information
- Starting or Stopping a Channel Programmatically

After you create services and events using Servlet Application Explorer, you can use managing and monitoring tools provided by the iWay Business Services Engine (iBSE) and the iWay Connector for JCA to measure the performance of your run-time environment. This section describes how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

iWay Business Services Engine (iBSE) provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that your application server is started.
2. To access the monitoring console, enter the following URL in your Web browser:

   `http://localhost:port/ibse/IBSEConfig`

   where:

   `localhost`

   Is the machine where the application server is running.

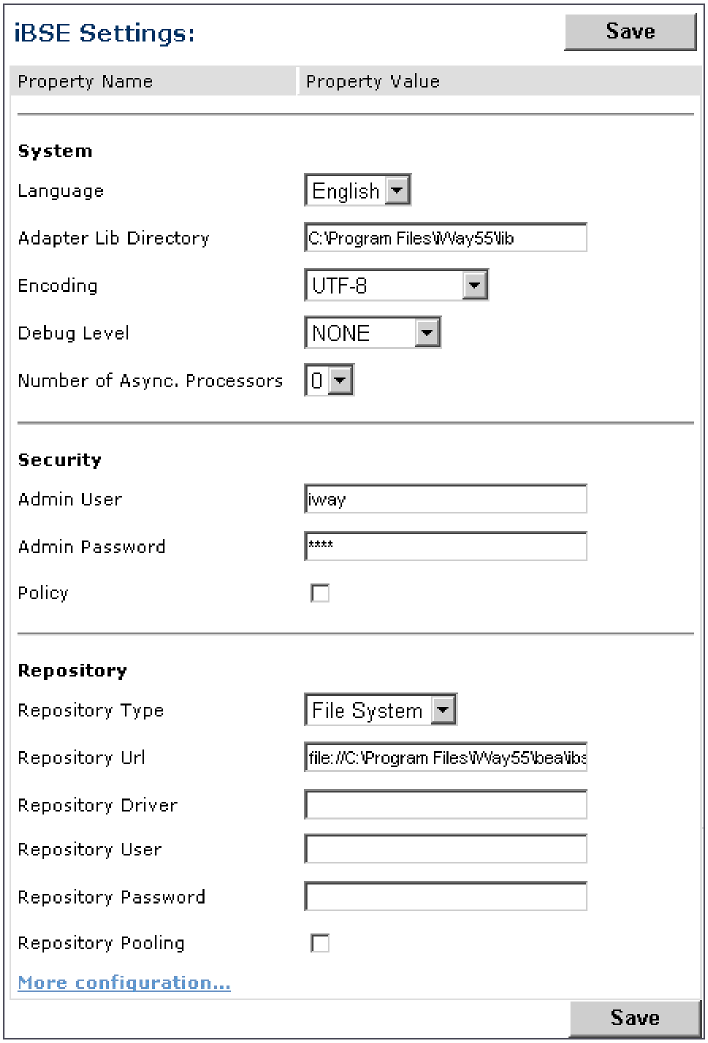
   `port`

   Is the HTTP port for the application server.

   The following image shows the iBSE Settings window that opens. It lists property names and includes fields where you can enter values for each property. To configure system settings, the System pane contains drop-down lists for selecting language, encoding, the debug level, and the number of asynchronous processors. It also contains a field where you can enter a path to the adapters lib directory.

   To configure security settings, the Security pane contains fields for typing the Admin User name and the associated password and a check box for specifying policy.

To configure repository settings, the Repository pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. In the upper and lower right of the window is a Save button. In the lower left of the window is an option to click to access more configuration settings.



Property Name	Property Value
System	
Language	English
Adapter Lib Directory	C:\Program Files\iWay55\lib
Encoding	UTF-8
Debug Level	NONE
Number of Async. Processors	0
Security	
Admin User	iway
Admin Password	****
Policy	□
Repository	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bea\ibs
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	□

3. Click *More configuration.*

Tip: To access the monitoring console directly, enter the following URL in your Web browser:

`http://localhost:port/ibse/IBSEStatus`

where:

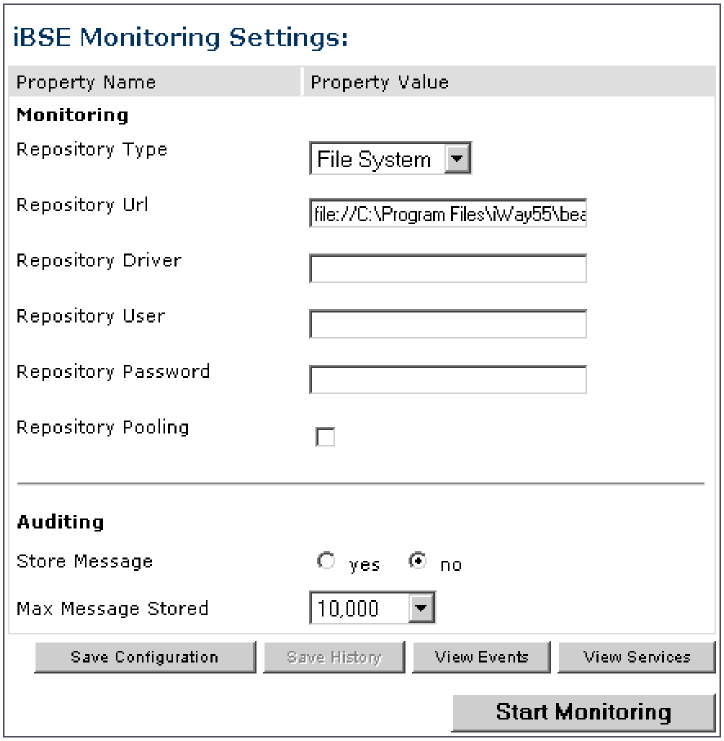
`localhost`

Is the machine where the application server is running.

`port`

Is the HTTP port for the application server.

The following image shows the iBSE Monitoring Settings window that opens. It lists property names and includes a corresponding field where you can enter values for each property. The Monitoring pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. The Auditing pane contains an option button to click to specify whether to store a message and a drop-down list where you can select the maximum messages to store. At the bottom of the window is a row of buttons that you can click to save your configuration, view events, or view services. The Save History button is inactive. After you enter properties and choose whether to save or view, you can click the Start Monitoring button.



iBSE Monitoring Settings:

Property Name	Property Value
Monitoring	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bea
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	☐
Auditing	
Store Message	○ yes ● no
Max Message Stored	10,000

   a. In the Monitoring pane, from the Repository Type drop-down list, select the type of repository you are using.
   b. To connect to the database in the Repository Url field, type a JDBC URL.
   c. To connect to the database in the Repository Driver field, type a JDBC Class.
   d. To access the monitoring repository database, type a user ID and password.
   e. To enable pooling, click the *Repository Pooling* check box.
   f. In the Auditing pane, select *yes* if you want to store messages.

      This option is disabled by default.

      Note: You must start and then, stop monitoring to enable this option.
   g. Select the maximum number of messages you want to store.

      By default, 10,000 is selected.

      Note: Depending on your environment and the number of messages that are exchanged, storing a large number of messages may affect system performance. If you need more information about your system resources, consult your system administrator.
   h. Click *Save Configuration*.

4. Click *Start Monitoring*.

   iBSE begins to monitor all services and events currently in use. If you selected the option to store messages, iBSE stores messages.

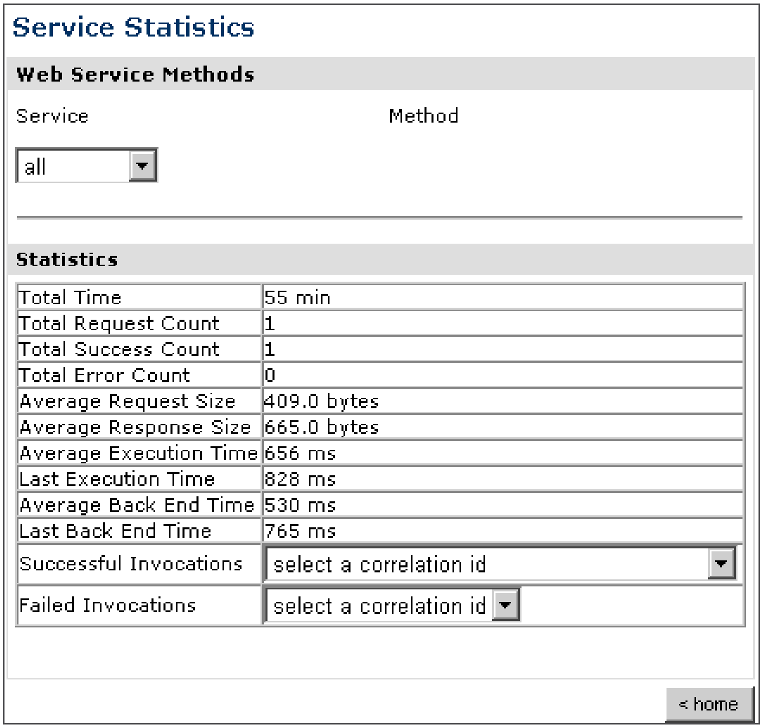
5. To stop monitoring, click *Stop Monitoring*.

Procedure: How to Monitor Services

To monitor services:

1. Ensure that your application server is started.
2. From the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Services*.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list where you select a service. On the right, space is reserved for a drop-down list of methods that will appear. The Statistics pane contains a table with a summary of service statistics and two drop-down lists where you can select a successful or failed invocation to view more information about that service. At the bottom of the window is a home button to click to return to the iBSE Monitoring Settings window.



Service Statistics

Web Service Methods

Service: all

Method

Statistics

Total Time	55 min
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	828 ms
Average Back End Time	530 ms
Last Back End Time	765 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

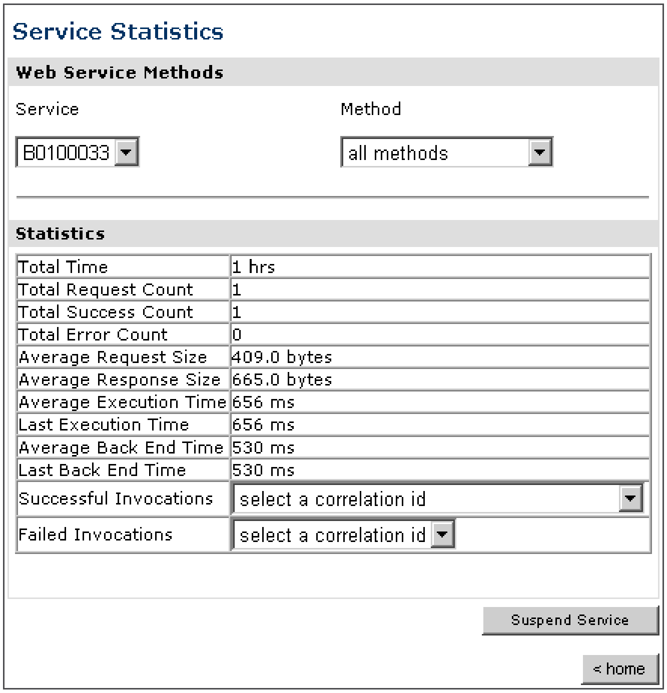
The system level summary provides services statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding service statistic.

Statistic	**Description**
Total Time	Total amount of time iBSE monitors services. The time starts after you click Start Monitoring in the iBSE Monitoring Settings window.
Total Request Count	Total number of services requests that were made during the monitoring session.
Total Success Count	Total number of successful service executions.
Total Error Count	Total number of errors that were encountered.
Average Request Size	Average size of an available service request.
Average Response Size	Average size of an available service response size.
Average Execution Time	Average execution time for a service.
Last Execution Time	Last execution time for a service.
Average Back End Time	Average back end time for a service.
Last Back End Time	Last back end time for a service.
Successful Invocations	A list of successful services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.
Failed Invocations	A list of failed services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.

4. Select a service from the drop-down list.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).



Service Statistics

Web Service Methods

Service: B0100033

Method: all methods

Statistics

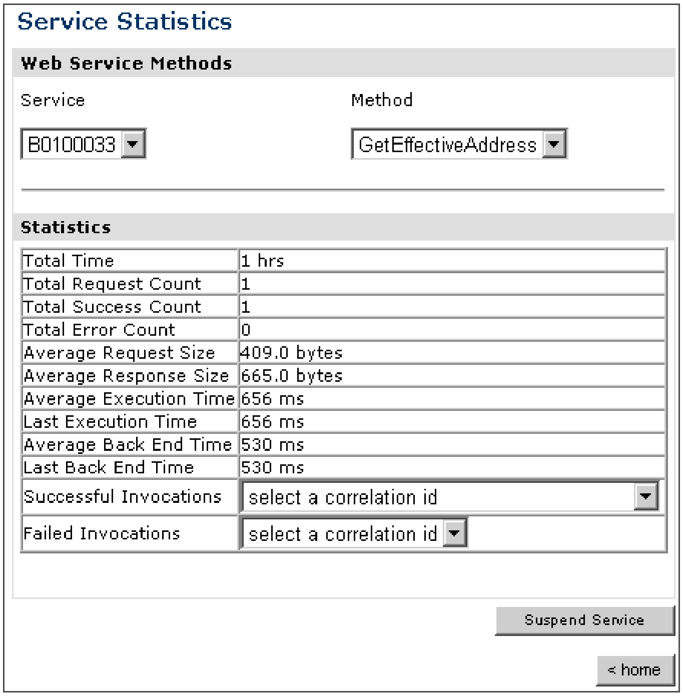
Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

a. To stop a service at any time, click *Suspend Service*.

b. To restart the service, click *Resume Service*.

5. Select a method for the service from the Method drop-down list.

The following image shows the Method Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).



Service Statistics

Web Service Methods

Service: B0100033

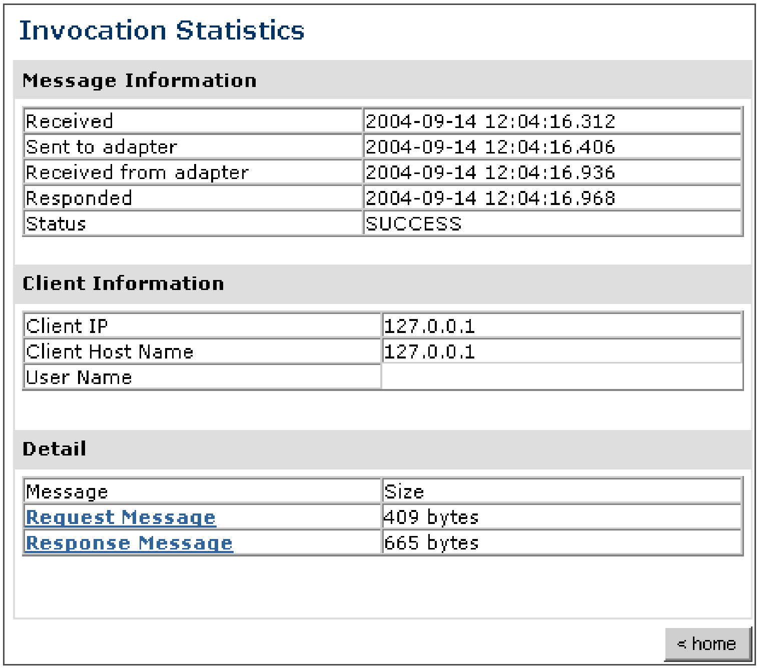
Method: GetEffectiveAddress

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

6. For additional information about a successful service and its method, select a service based on its correlation ID from the Successful Invocation drop-down list.

The following image shows the Invocation Level Statistics window that opens. The Message Information pane contains a table of information about the message. The Client Information pane contains a table of information about the client. The Detail pane contains a table that shows the size of the request and response messages, with options to click to view the respective XML documents. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.



Invocation Statistics

Message Information

Received	2004-09-14 12:04:16.312
Sent to adapter	2004-09-14 12:04:16.406
Received from adapter	2004-09-14 12:04:16.936
Responded	2004-09-14 12:04:16.968
Status	SUCCESS

Client Information

Client IP	127.0.0.1
Client Host Name	127.0.0.1
User Name	

Detail

Message	Size
Request Message	409 bytes
Response Message	665 bytes

7. To view the XML request document in your Web browser, click *Request Message*.

   You can also view the XML response document for the service.

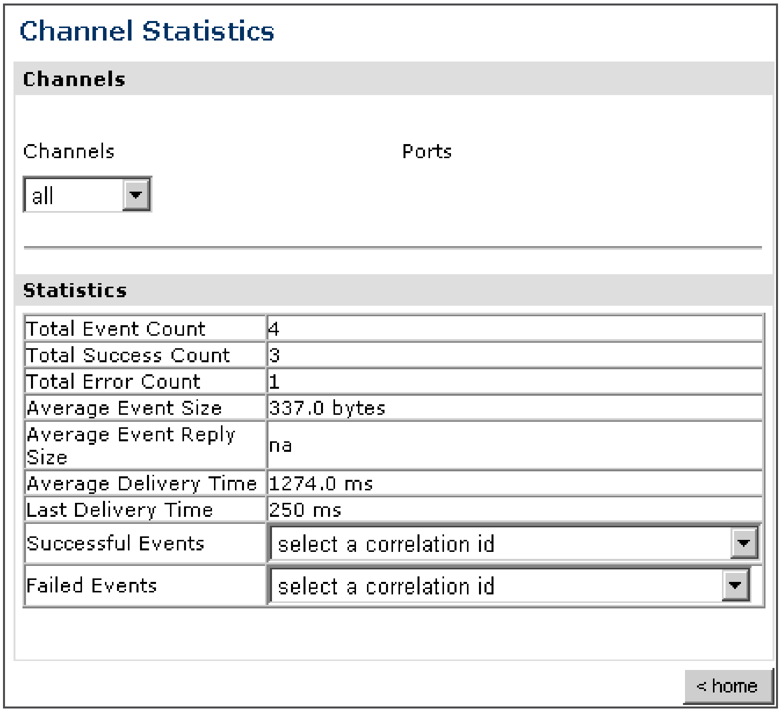
8. To return to the iBSE Monitoring Settings window, click *home*.

Procedure: How to Monitor Events

To monitor events:

1. Ensure that your application server is started.
2. In the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Events*.

The following image shows the System Level Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel. On the right, space is reserved for a drop-down list of ports that will appear. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.



Channel Statistics

Channels

Channels: all

Ports

Statistics

Total Event Count	4
Total Success Count	3
Total Error Count	1
Average Event Size	337.0 bytes
Average Event Reply Size	na
Average Delivery Time	1274.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id
Failed Events	select a correlation id

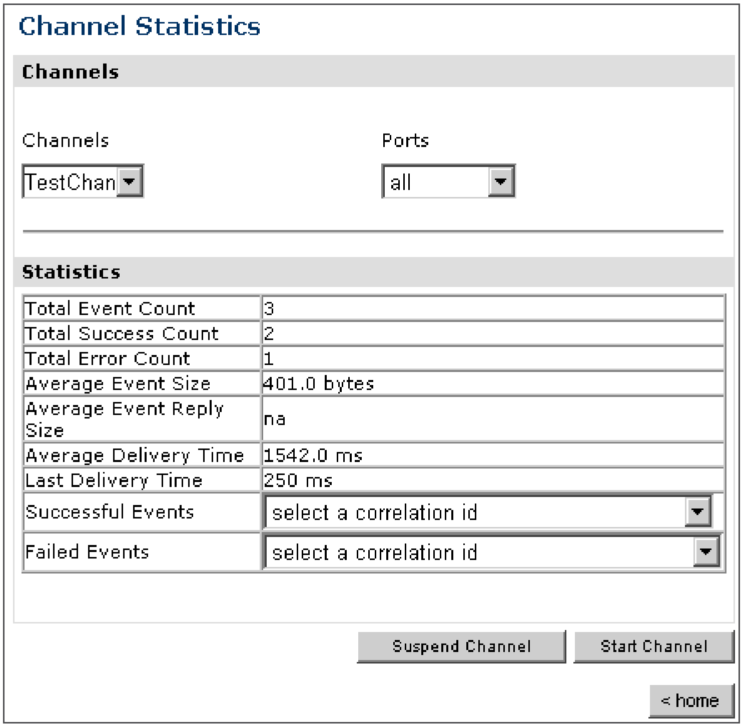
The system level summary provides event statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding event statistic.

Statistic	**Description**
Total Event Count	Total number of events.
Total Success Count	Total number of successful event executions.
Total Error Count	Total number of errors that were encountered.
Average Event Size	Average size of an available event request.
Average Event Reply Size	Average size of an available event response.
Average Delivery Time	Average delivery time for an event.
Last Delivery Time	Last delivery time for an event.
Successful Events	List of successful events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.
Failed Events	List of failed events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.

4. Select a channel from the drop-down list.

The following image shows the Channel Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.



Channel Statistics

Channels

Channels: TestChan

Ports: all

Statistics

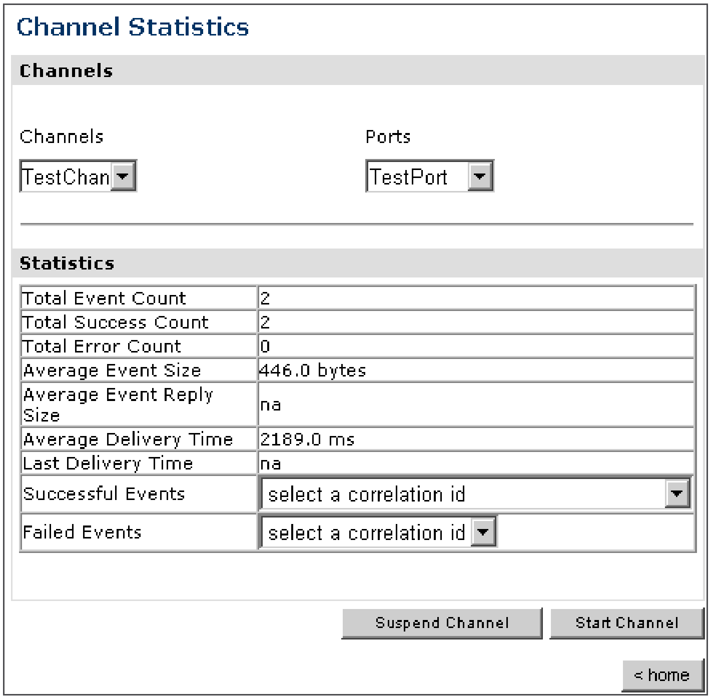
Total Event Count	3
Total Success Count	2
Total Error Count	1
Average Event Size	401.0 bytes
Average Event Reply Size	na
Average Delivery Time	1542.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id
Failed Events	select a correlation id

a. To stop a channel at any time, click *Suspend Channel*.

b. To start the channel, click *Start Channel*.

5. From the Ports drop-down list, select a port for the channel.

The following image shows the Port Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.



Channel Statistics

Channels

Channels: TestChan

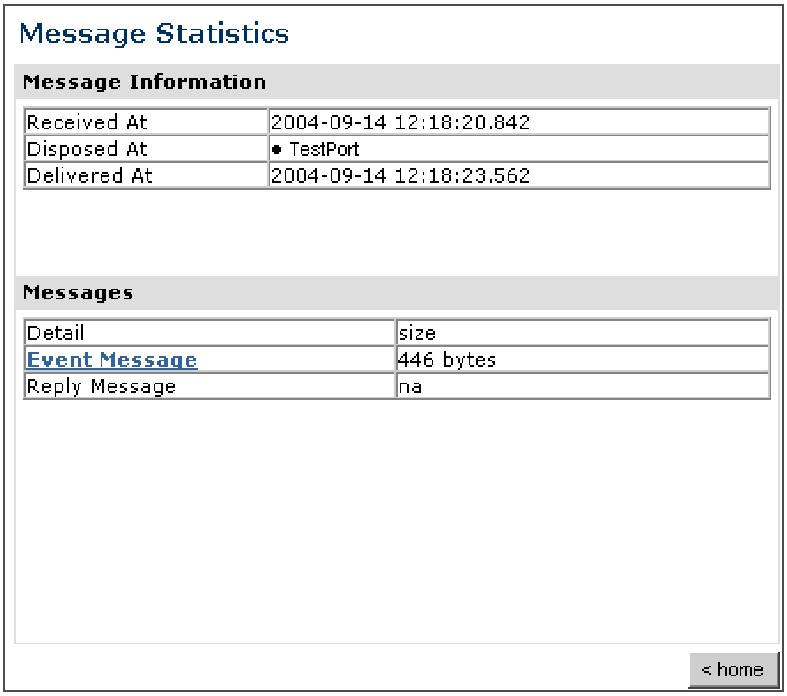
Ports: TestPort

Statistics

Total Event Count	2
Total Success Count	2
Total Error Count	0
Average Event Size	446.0 bytes
Average Event Reply Size	na
Average Delivery Time	2189.0 ms
Last Delivery Time	na
Successful Events	select a correlation id
Failed Events	select a correlation id

6. For more information about a successful event and its port, select an event based on its correlation ID from the Successful Events drop-down list.

The following image shows the Event Level Statistics (Message Statistics) window that opens. The Message Information pane contains a table of information pertaining to the event message. The Messages pane contains a table that shows the size of the event and reply messages, with an option to view an XML document of the event message. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.



Message Statistics

Message Information

Received At	2004-09-14 12:18:20.842
Disposed At	● TestPort
Delivered At	2004-09-14 12:18:23.562

Messages

Detail	size
Event Message	446 bytes
Reply Message	na

a. To view the XML event document in your Web browser, click *Event Message*.

b. To return to the iBSE Monitoring Settings window, click *home*.

Managing and Monitoring Services and Events Using the JCA Test Tool

The JCA Test Tool, which is also known as the JCA Installation Verification Program (IVP), provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

Procedure: How to Manage and Monitor Services Using the JCA Test Tool

To manage and monitor services using the JCA Test Tool:

1. Open a Web browser to:

   `http://localhost:port/iwjcaivp`

   where:

   `localhost`

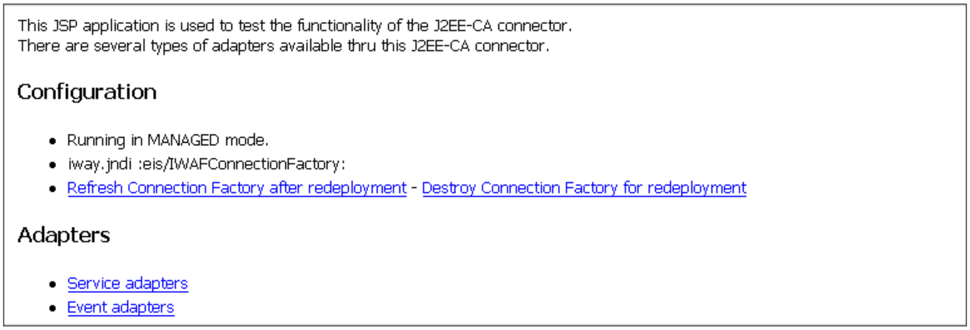
   Is the name of the machine where your application server is running.

   `port`

   Is the HTTP port for the application server, for example:

   `http://localhost:7001/iwjcaivp`

   The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.

   This JSP application is used to test the functionality of the J2EE-CA connector.
   There are several types of adapters available thru this J2EE-CA connector.

   Configuration

   - Running in MANAGED mode.
   - iway.jndi :eis/IWAFConnectionFactory:
   - Refresh Connection Factory after redeployment - Destroy Connection Factory for redeployment

   Adapters

   - Service adapters
   - Event adapters

   The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest service adapter configuration.

   Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you also must perform these steps for every new JCA configuration that is created using Application Explorer.

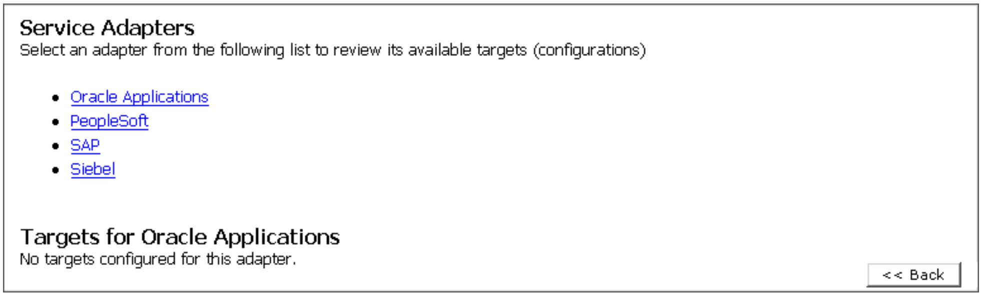
   a. Click *Destroy Connection Factory for redeployment*.

   b. Redeploy the JCA connector.

   c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.

3. Click *Service adapters*.

   The following image shows the Service Adapters page that opens. The page provides a live list of available service adapters and a list of targets configured for a specific adapter. In the lower right is a Back button to click to return to the previous page.



Service Adapters

Select an adapter from the following list to review its available targets (configurations)

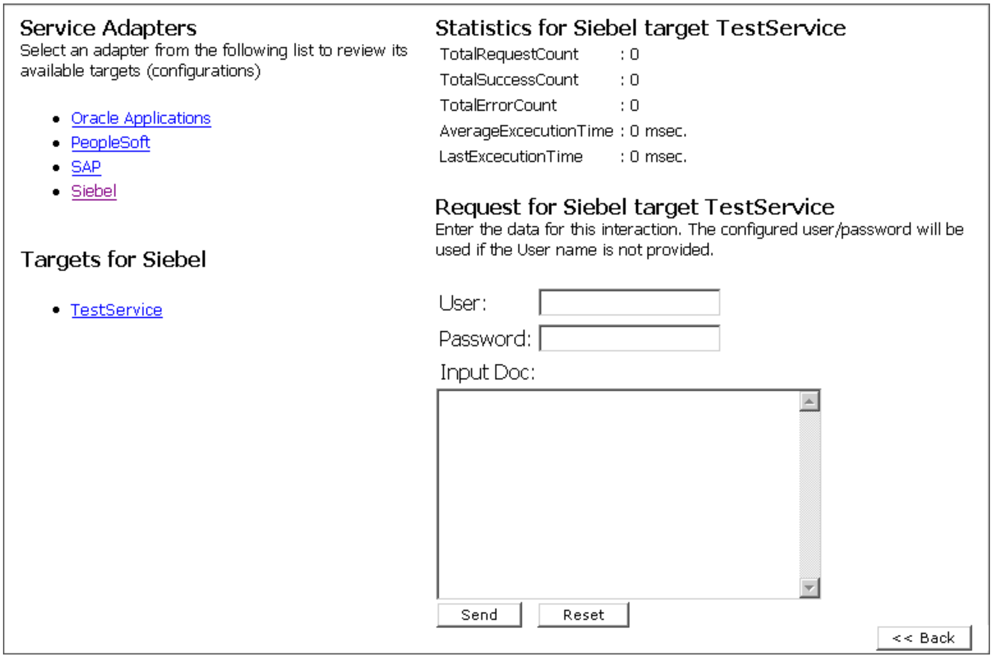
- Oracle Applications
- PeopleSoft
- SAP
- Siebel

Targets for Oracle Applications

No targets configured for this adapter.

4. Select a service adapter to monitor.

The following image shows the page that opens. The left side provides a live list of available service adapters and a list of any targets configured for a specific adapter. The upper right side shows statistics for a selected target. The middle right has a User field and a Password field. The lower right contains a box where you type or paste an input document. Below the input box is a Send button to click to send a request for a test service and a Reset button to click to reset the fields. In the lower right is a Back button to click to return to the previous page.



Service Adapters

Select an adapter from the following list to review its available targets (configurations)

- Oracle Applications
- PeopleSoft
- SAP
- Siebel

Targets for Siebel

- TestService

Statistics for Siebel target TestService

```
TotalRequestCount      : 0
TotalSuccessCount      : 0
TotalErrorCount        : 0
AverageExcecutionTime : 0 msec.
LastExcecutionTime     : 0 msec.
```

Request for Siebel target TestService

Enter the data for this interaction. The configured user/password will be used if the User name is not provided.

User:

Password:

Input Doc:

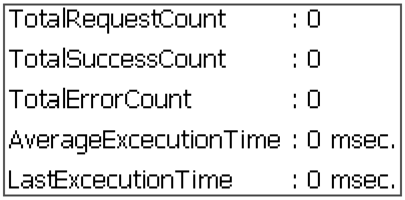
a. Click the desired target for your service adapter.

b. In the Request area, enter a user name and password.

c. In the Input Doc area, enter a request document that was created from the request schema for your service.

5. Click *Send*.

The following image shows the updated statistics that appear for your service if the request is successful. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds.



```
TotalRequestCount      : 0
TotalSuccessCount      : 0
TotalErrorCount        : 0
AverageExcecutionTime  : 0 msec.
LastExcecutionTime     : 0 msec.
```

Procedure: How to Manage and Monitor Events Using the JCA Test Tool

To manage and monitor events using the JCA Test Tool:

1. Open a Web browser to:

   `http://localhost:port/iwjcaivp`

   where:

   `localhost`

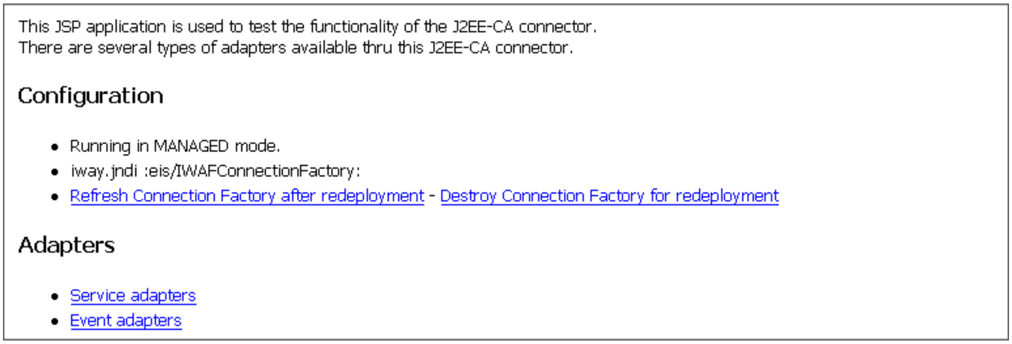
   Is the name of the machine where your application server is running.

   `port`

   Is the HTTP port for the application server, for example:

   `http://localhost:7001/iwjcaivp`

   The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.



This JSP application is used to test the functionality of the J2EE-CA connector.
There are several types of adapters available thru this J2EE-CA connector.

Configuration

- Running in MANAGED mode.
- iway.jndi :eis/IWAFConnectionFactory:
- Refresh Connection Factory after redeployment - Destroy Connection Factory for redeployment

Adapters

- Service adapters
- Event adapters

   The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest event adapter configuration.

Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you must also perform these steps for every new JCA configuration that is created using Application Explorer.

- **a.** Click *Destroy Connection Factory for redeployment*.
- **b.** Redeploy the JCA connector.
- **c.** In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.

3. Click *Event adapters*.

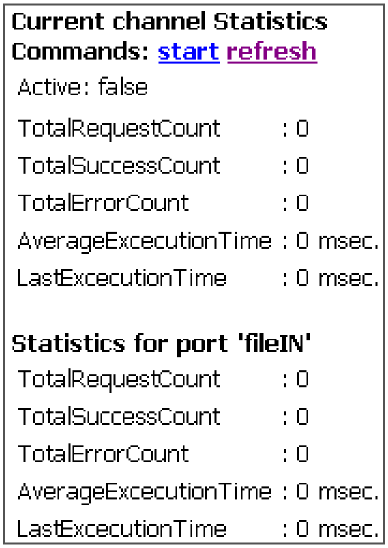
The Event Adapters page opens.

4. Select the event adapter to monitor.

5. Click the desired channel for your event adapter.

6. Click *start*.

The following image shows the updated statistics for your channel and the port. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds. There are options to click in the upper right of the page to start or refresh the channel.



Current channel Statistics
Commands: start refresh

```
Active: false
TotalRequestCount       : 0
TotalSuccessCount       : 0
TotalErrorCount         : 0
AverageExcecutionTime : 0 msec.
LastExcecutionTime      : 0 msec.
```

Statistics for port 'fileIN'

```
TotalRequestCount       : 0
TotalSuccessCount       : 0
TotalErrorCount         : 0
AverageExcecutionTime : 0 msec.
LastExcecutionTime      : 0 msec.
```

Setting Engine Log Levels

The following section describes how to set engine log levels for Servlet iBSE and JCA. For more information, see the documentation.

Procedure: How to Enable Tracing for Servlet iBSE

To enable tracing for Servlet iBSE:

1. Open the Servlet iBSE configuration page at:

   ```
   http://localhost:port/ibse/IBSEConfig
   ```

   where:

   localhost

   Is the name of the machine where your application server is running.

   port

   Is the HTTP port for the application server, for example:

   ```
   http://localhost:7001/ibse/IBSEConfig
   ```

2. In the System pane, from the Debug drop-down list, select the level of tracing.
3. Click *Save*.

Tracing information is written to the ibselogs directory where your application server accesses or has expanded Servlet iBSE.

Procedure: How to Enable Tracing for JCA

To enable tracing for JCA:

1. Open the extracted ra.xml file in a text editor.
2. Locate and change the following setting:

   LogLevel. This setting can be set to DEBUG, INFO, or ERROR.

   ```
   <context-param>
   <config-property>
       <config-property-name>LogLevel</config-property-name>
       <config-property-type>java.lang.String</config-property-type>
       <config-property-value></config-property-value>
   </config-property>
   ```

   For example:

   ```
   <config-property-value>DEBUG</config-property-value>
   ```

   A directory in the configuration directory contains the logs.

   a. Review the logs generated by your application server.

   b. Leave the remainder of the previous file unchanged.

3. Save the file and exit the editor.
4. Redeploy the connector.

Migrating Repositories

During design time, a repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. For more information on configuring repositories, see the *iWay 5.5 Installation and Configuration* documentation.

The information in the repository also is referenced at run time. For management purposes, you can migrate iBSE and JCA repositories to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a development environment to a production environment.

File Repositories

If you want to migrate a File repository to another destination, copy the ibserepo.xml file from the following path:

```
drive:\Program Files\iWay55\ibse\ibserepo.xml
```

where:

`drive`

Is the location of your iWay 5.5 installation.

You can place the ibserepo.xml file in a new location that is a root directory of the iBSE Web application, for example:

```
drive:\ProductionConfig\ibse\ibserepo.xml
```

iBSE Repositories

The following topic describes how to migrate an iBSE repository that is configured for Oracle. You can follow the same procedure if you want to migrate an iBSE repository that is configured for Microsoft SQL Server 2000, Sybase, or DB2. However, when you are configuring a new environment, you must execute the script that creates the repository tables for your database. In addition, verify that all required files and drivers for your database are in the class path. For more information on configuring repositories, see the *iWay 5.5 Installation and Configuration* documentation.

Note: The following procedure allows you to migrate only Web services. If migrating event handling information is one of your requirements, you must migrate at the database level. For more information, see *Migrating Event Handling Configurations* on page 5-27.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

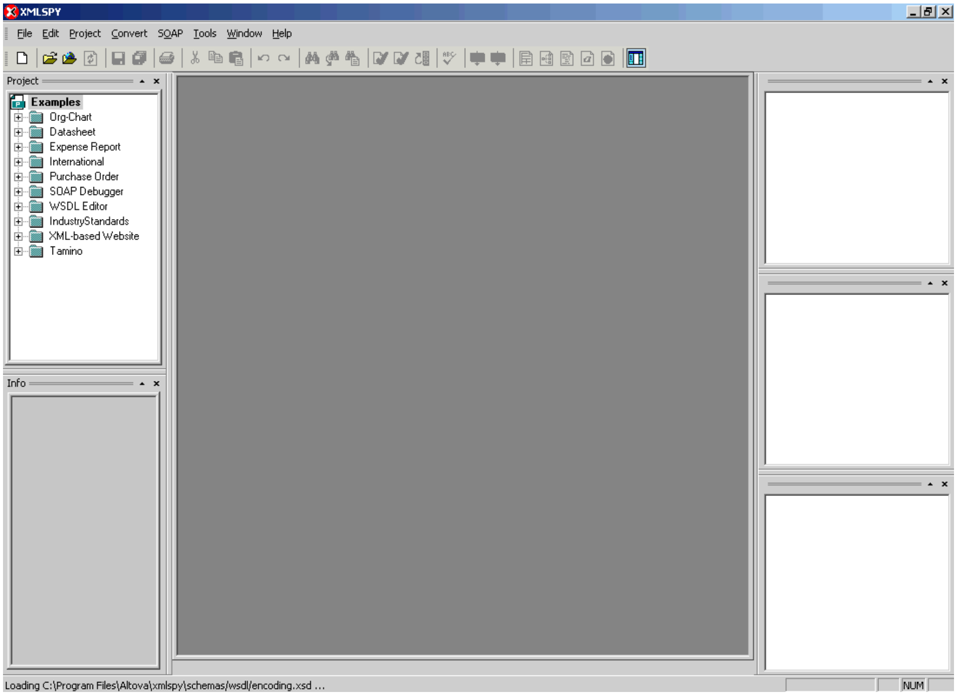
To migrate an iBSE repository that is configured for Oracle:

1. Copy the iBSE configuration service URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

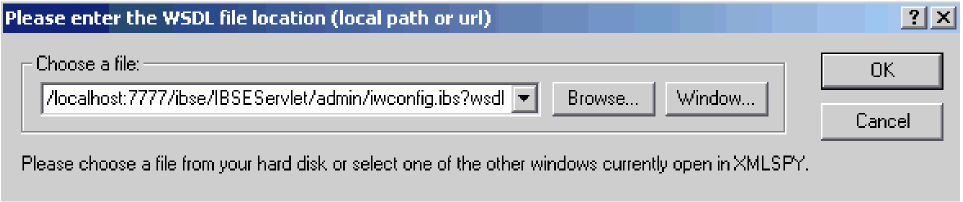
   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

The following image shows the WSDL file location dialog box that opens, where you enter a local path or URL. The dialog includes Browse, Window, OK, and Cancel buttons.



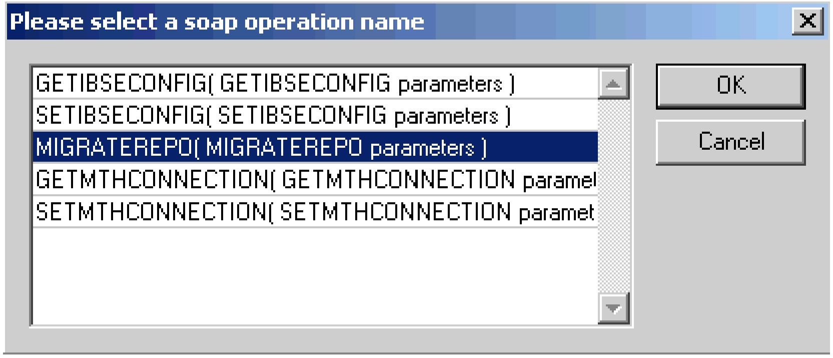
/localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl

Please choose a file from your hard disk or select one of the other windows currently open in XMLSPY.

4. In the Choose a file field, paste the iBSE configuration service URL.

5. Click *OK*.

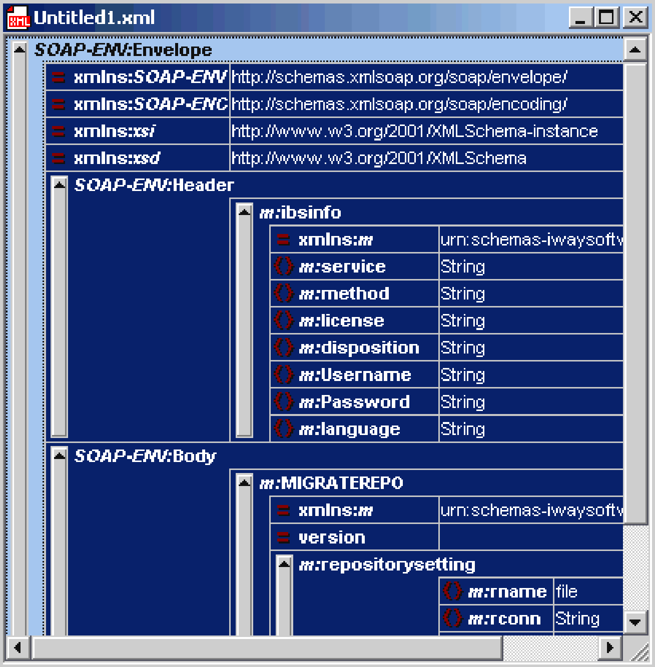
   The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select from the list and click OK or to escape from the dialog box, you can click Cancel.



GETIBSECONFIG(GETIBSECONFIG parameters)
SETIBSECONFIG(SETIBSECONFIG parameters)
MIGRATEREPO(MIGRATEREPO parameters)
GETMTHCONNECTION(GETMTHCONNECTION paramet
SETMTHCONNECTION(SETMTHCONNECTION paramet

6. Select the *MIGRATEREPO(MIGRATEREPO parameters)* control method and click *OK*.

The following image shows a portion of the window that opens with the structure of the SOAP envelope. It includes information about location and schemas.



Untitled1.xml

SOAP-ENV:Envelope

xmlns:SOAP-ENV	http://schemas.xmlsoap.org/soap/envelope/
xmlns:SOAP-ENC	http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns:xsd	http://www.w3.org/2001/XMLSchema

SOAP-ENV:Header

m:ibsinfo

xmlns:m	urn:schemas-iwaysoftw
m:service	String
m:method	String
m:license	String
m:disposition	String
m:Username	String
m:Password	String
m:language	String

SOAP-ENV:Body

m:MIGRATEREPO

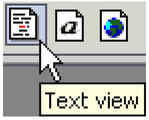
xmlns:m	urn:schemas-iwaysoftw
version	

m:repositorysetting

m:rname	file
m:rconn	String

7. Locate the *Text view* icon in the tool bar.

   In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

   ```
   <m:MIGRATEREPO
   xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version="">
   <m:repositorysetting>
   <m:rname>oracle</m:rname>
   <m:rconn>String</m:rconn>
   <m:rdriver>String</m:rdriver>
   <m:ruser>String</m:ruser>
   <m:rpwd>String</m:rpwd>
   </m:repositorysetting>
   <m:servicename>String</m:servicename>
   </m:MIGRATEREPO>
   ```

   a. For the <m:rconn> tag, replace the String placeholder with the repository URL where you want to migrate your existing iBSE repository.

   For example, the Oracle repository URL has the following format:

   ```
   jdbc:oracle:thin:@[host]:[port]:[sid]
   ```

   b. For the <m:rdriver> tag, replace the String placeholder with the location of your Oracle driver.

   Note: This is an optional tag. If you do not specify a value, the default Oracle JDBC driver is used.

   c. For the <m:ruser> tag, replace the String placeholder with a valid user name to access the Oracle repository.

   d. For the <m:rpwd> tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

    If you want to migrate a **single** Web service from the current iBSE repository, enter the Web service name in the <m:servicename> tag, for example:

    ```
    <m:servicename>Service1</m:servicename>
    ```

    If you want to migrate **multiple** Web services from the current iBSE repository, duplicate the <m:servicename> tag for each Web service, for example:

    ```
    <m:servicename>Service1</m:servicename>
    <m:servicename>Service2</m:servicename>
    ```

    If you want to migrate **all** Web services from the current iBSE repository, remove the <m:servicename> tag.

11. From the SOAP menu, select *Send request to server.*

    Your iBSE repository and the Web services you specified migrate to the new Oracle repository URL that you specified.

JCA Repositories

The following procedure describes how to migrate a JCA repository. For more information on configuring JCA repositories, see the *iWay 5.5 Installation and Configuration* documentation.

Procedure: How to Migrate a JCA Repository

To migrate a JCA repository:

1. Navigate to the location of your JCA configuration directory where the repository schemas and other information is stored, for example:

   ```
   C:\Program Files\iway55\config\base
   ```

2. Locate and copy the *repository.xml* file.
3. Place this file in a new JCA configuration directory to migrate the existing repository.

   Your JCA repository migrates to the new JCA configuration directory.

Migrating Event Handling Configurations

This topic describes how to migrate your iBSE repositories at a database level for Microsoft SQL Server 2000, Oracle, Sybase, or DB2. You can use this information to migrate event handling information, for example, port or channel configurations.

Procedure How to Migrate a Microsoft SQL Server 2000 Repository

To migrate a Microsoft SQL Server 2000 repository:

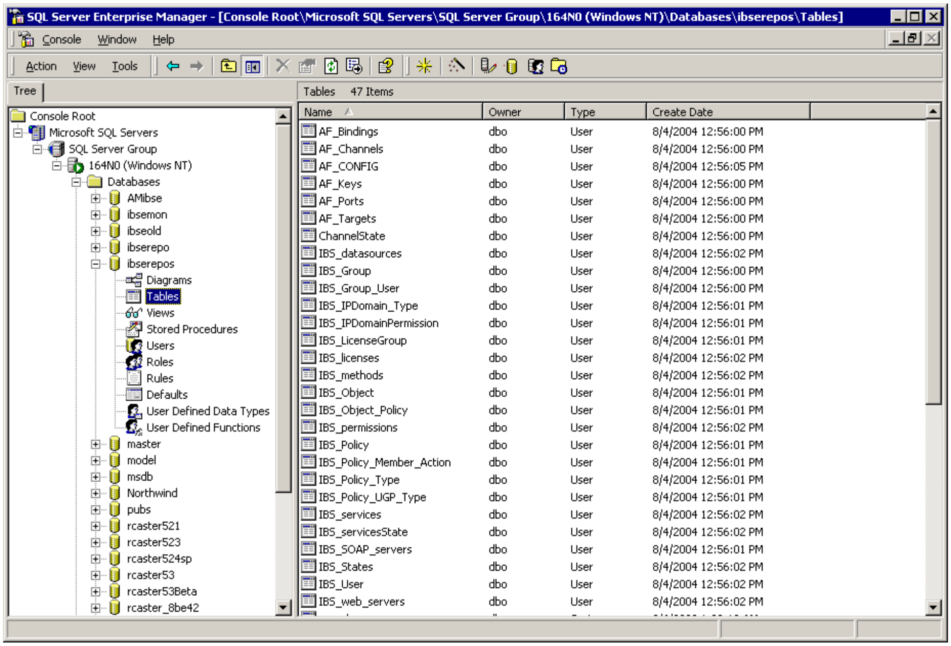
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   C:\Program Files\iWay55\etc\setup
   ```

   This directory contains SQL to create the repository tables in the following file:

   ```
   iwse.sql
   ```

You can use iwse.sql to create the database tables that are used by iBSE. For example, the following image shows the tree in the left pane and tables in the right pane. The tables are listed by name in one column with corresponding columns for information about owner, type, and the date the table was created.



Name	Owner	Type	Create Date
AF_Bindings	dbo	User	8/4/2004 12:56:00 PM
AF_Channels	dbo	User	8/4/2004 12:56:00 PM
AF_CONFIG	dbo	User	8/4/2004 12:56:05 PM
AF_Keys	dbo	User	8/4/2004 12:56:00 PM
AF_Ports	dbo	User	8/4/2004 12:56:00 PM
AF_Targets	dbo	User	8/4/2004 12:56:00 PM
ChannelState	dbo	User	8/4/2004 12:56:00 PM
IBS_datasources	dbo	User	8/4/2004 12:56:02 PM
IBS_Group	dbo	User	8/4/2004 12:56:00 PM
IBS_Group_User	dbo	User	8/4/2004 12:56:00 PM
IBS_IPDomain_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_IPDomainPermission	dbo	User	8/4/2004 12:56:01 PM
IBS_LicenseGroup	dbo	User	8/4/2004 12:56:01 PM
IBS_licenses	dbo	User	8/4/2004 12:56:02 PM
IBS_methods	dbo	User	8/4/2004 12:56:02 PM
IBS_Object	dbo	User	8/4/2004 12:56:01 PM
IBS_Object_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_permissions	dbo	User	8/4/2004 12:56:02 PM
IBS_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Member_Action	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_UGP_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_services	dbo	User	8/4/2004 12:56:02 PM
IBS_servicesState	dbo	User	8/4/2004 12:56:02 PM
IBS_SOAP_servers	dbo	User	8/4/2004 12:56:01 PM
IBS_States	dbo	User	8/4/2004 12:56:02 PM
IBS_User	dbo	User	8/4/2004 12:56:02 PM
IBS_web_servers	dbo	User	8/4/2004 12:56:02 PM

For more information on configuring the Microsoft SQL Server 2000 repository, see the *iWay 5.5 Installation and Configuration* documentation.

2. To migrate the tables that were created by the iwse.sql script for iBSE, use your Microsoft SQL Server 2000 database tool set. For more information, consult your database administrator.

Procedure How to Migrate an Oracle Repository

To migrate an Oracle repository:

1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   C:\Program Files\iWay55\etc\setup
   ```

   This directory contains SQL to create the repository tables in the following files:

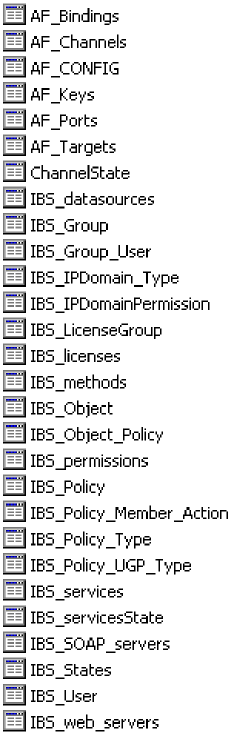
   For Oracle 8:

   ```
   iwse.ora
   ```

For Oracle 9:

```
iwse.ora9
```

2. To create the Oracle database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IBS_datasources
IBS_Group
IBS_Group_User
IBS_IPDomain_Type
IBS_IPDomainPermission
IBS_LicenseGroup
IBS_licenses
IBS_methods
IBS_Object
IBS_Object_Policy
IBS_permissions
IBS_Policy
IBS_Policy_Member_Action
IBS_Policy_Type
IBS_Policy_UGP_Type
IBS_services
IBS_servicesState
IBS_SOAP_servers
IBS_States
IBS_User
IBS_web_servers

For more information on configuring the Oracle repository, see the *iWay 5.5 Installation and Configuration* documentation.

3. To migrate the tables that were created by the SQL script for iBSE, use your Oracle database tool set. For more information, consult your database administrator.

Procedure How to Migrate a Sybase Repository

To migrate a Sybase repository:

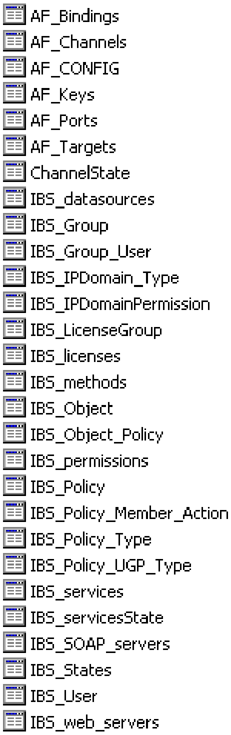
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   C:\Program Files\iWay55\etc\setup
   ```

   This directory contains SQL to create the repository tables in the following file:

   ```
   sybase-iwse.sql
   ```

2. To create the Sybase database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IBS_datasources
IBS_Group
IBS_Group_User
IBS_IPDomain_Type
IBS_IPDomainPermission
IBS_LicenseGroup
IBS_licenses
IBS_methods
IBS_Object
IBS_Object_Policy
IBS_permissions
IBS_Policy
IBS_Policy_Member_Action
IBS_Policy_Type
IBS_Policy_UGP_Type
IBS_services
IBS_servicesState
IBS_SOAP_servers
IBS_States
IBS_User
IBS_web_servers

   For more information on configuring the Sybase repository, see the *iWay 5.5 Installation and Configuration* documentation.

3. To migrate the tables that were created by the SQL script for iBSE, use your Sybase database tool set. For more information, consult your database administrator.

Procedure How to Migrate a DB2 Repository

To migrate a DB2 repository:

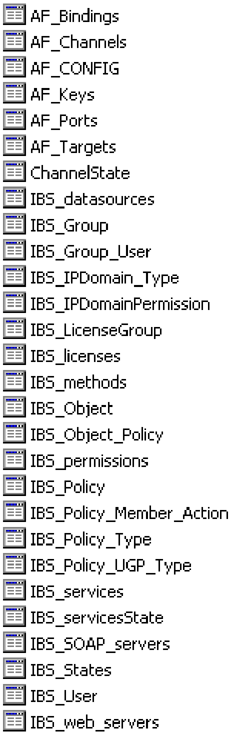
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   C:\Program Files\iWay55\etc\setup
   ```

   This directory contains SQL to create the repository tables in the following file:

   ```
   db2-iwse.sql
   ```

2. To create the DB2 database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



- AF_Bindings
- AF_Channels
- AF_CONFIG
- AF_Keys
- AF_Ports
- AF_Targets
- ChannelState
- IBS_datasources
- IBS_Group
- IBS_Group_User
- IBS_IPDomain_Type
- IBS_IPDomainPermission
- IBS_LicenseGroup
- IBS_licenses
- IBS_methods
- IBS_Object
- IBS_Object_Policy
- IBS_permissions
- IBS_Policy
- IBS_Policy_Member_Action
- IBS_Policy_Type
- IBS_Policy_UGP_Type
- IBS_services
- IBS_servicesState
- IBS_SOAP_servers
- IBS_States
- IBS_User
- IBS_web_servers

For more information on configuring the DB2 repository, see the *iWay 5.5 Installation and Configuration* documentation.

You can migrate the tables that were created by the SQL script for iBSE using your DB2 database toolset. For more information, consult your database administrator.

Exporting or Importing Targets

After you migrate your repository, you can export or import targets with their connection information and persistent data between repositories.

Procedure: How to Export a Target

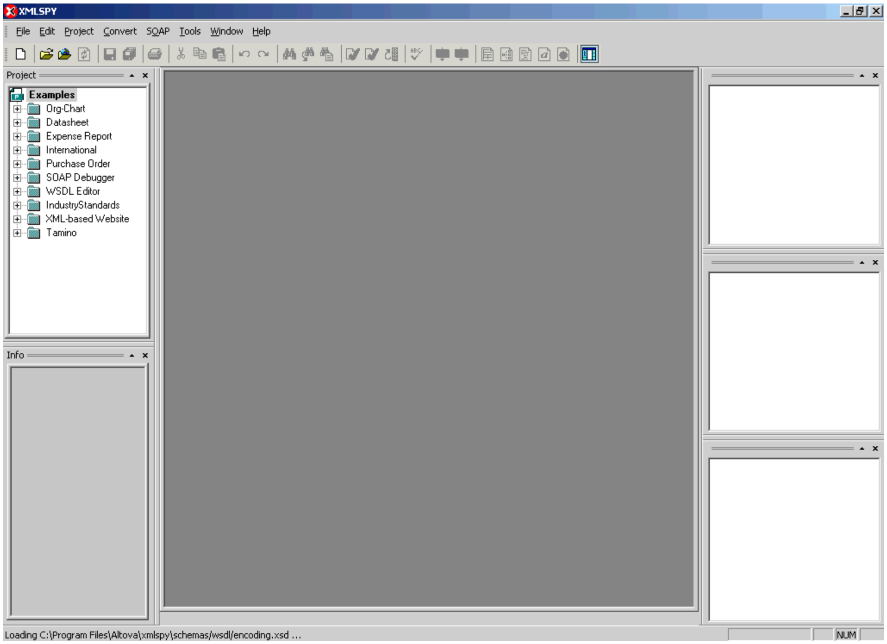
To export a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl\encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL.

5. Click *OK*.

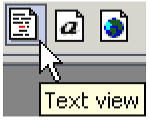
The soap operation name dialog box opens and lists the available control methods.

6. Select the *EXPORTTARGET(EXPORTTARGET parameters)* control method and click *OK*.

   A window opens that shows the structure of the SOAP envelope.

7. Locate the *Text view* icon in the tool bar.

   In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:EXPORTTARGET
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
<m:target>String</m:target>
<m:name>String</m:name>
</m:EXPORTTARGET>
```

   a. For the <m:target> tag, replace the String placeholder with the EIS target system name as it appears in Application Explorer and verify whether this value is case sensitive.

   b. For the <m:name> tag, replace the String placeholder with the name of the target you want to export.

10. From the SOAP menu, select *Send request to server.*

    A response is returned that contains the <m: exporttime> and <m: contents> elements. You must use these elements when importing your target.

Procedure: How to Import a Target

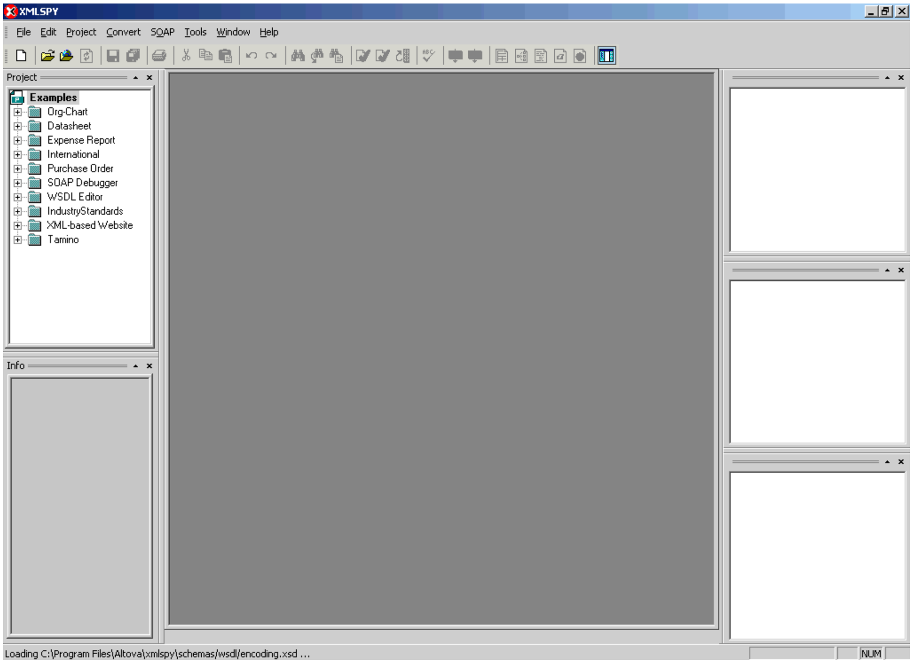
To import a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.

   Loading C:\Program Files\Altova\xmlspy\schemas\wsdl\encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL and click *OK*.

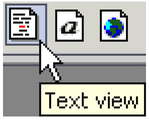
   The soap operation name dialog box opens and lists the available control methods.

5. Select the *IMPORTTARGET(IMPORTTARGET parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:IMPORTTARGET
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
<m:targetinstance>
<m:target>String</m:target>
<m:name>String</m:name>
<m:description>String</m:description>
<m:repositoryid>String</m:repositoryid>
<m:exporttime>2001-12-17T09:30:47-05:00</m:exporttime>
<m:contents>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</m:contents>
</m:targetinstance>
</m:IMPORTTARGET>
```

a. For the <m:target> tag, replace the String placeholder with the EIS target system name.

b. For the <m:name> tag, replace the String placeholder with the new name of the target you want to import.

c. For the <m:description> tag, replace the String placeholder with a description of the target.

d. For the <m:repositoryid> tag, copy and paste the contents of the <m:repositoryid> tag that was returned when you exported your target.

e. For the <m: exporttime> tag, copy and paste the contents of the <m: exporttime> tag that was returned when you exported your target.

f. For the <m: contents> tag, copy and paste the contents of the <m: contents> tag that was returned when you exported your target.

9. From the SOAP menu, select *Send request to server.*

Retrieving or Updating Web Service Method Connection Information

After you migrate your repository, you can retrieve or update connection information for your Web service methods.

Procedure: How to Retrieve Web Service Method Connection Information

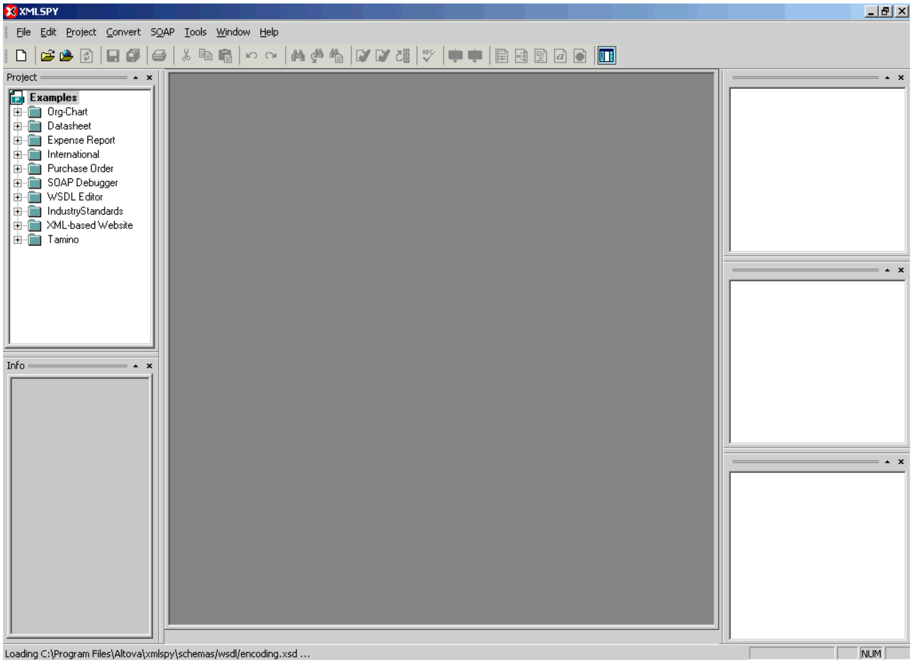
To retrieve Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl\encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

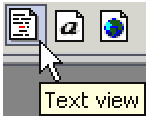
   The soap operation name dialog box opens and lists the available control methods.

5. Select the *GETMTHCONNECTION(GETMTHCONNECTION parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

   In the following image, the pointer points to the Text view icon.

7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:GETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:servicename>String</m:servicename>
<m:methodname>String</m:methodname>
</m:GETMTHCONNECTION>
```

   a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.

   b. For the <m:methodname> tag, replace the String placeholder with name of the Web service method.

9. From the SOAP menu, select *Send request to server.*

   A response is returned that contains the <m: descriptor> element. You must use this element when updating your Web service method.

Procedure: How to Update Web Service Method Connection Information

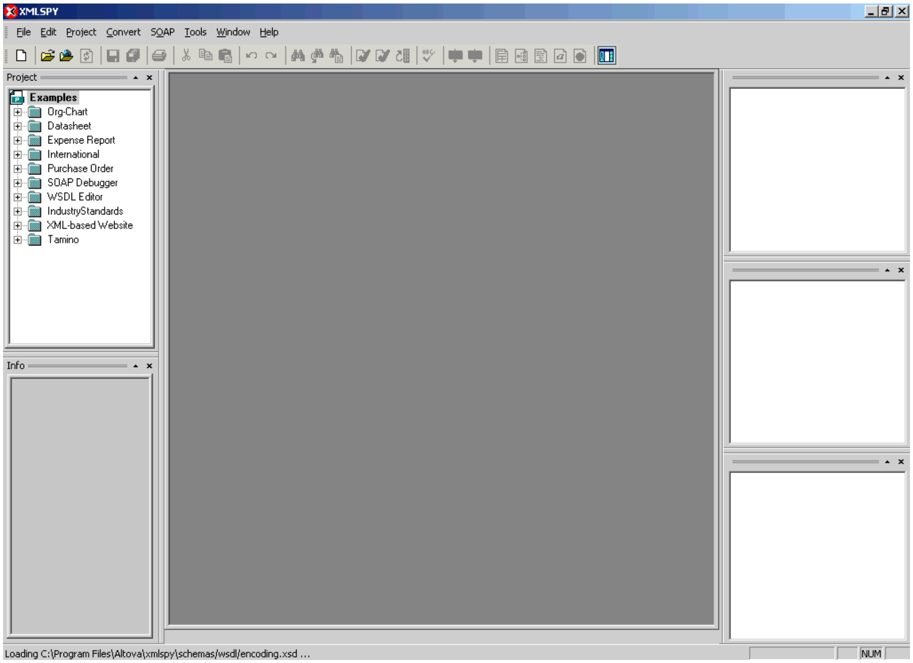
To update Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas/wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

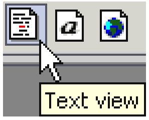
   The soap operation name dialog box opens and lists the available control methods.

5. Select the *SETMTHCONNECTION(SETMTHCONNECTION parameters)* control method and click *OK*.

   A window opens that shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

   In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:SETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:servicename>String</m:servicename>
<m:methodname>String</m:methodname>
<m:descriptor format="" channel="">
     <m:option title="">
       <m:group title="">
          <m:param/>
       </m:group>
     </m:option>
</m:descriptor>
</m:SETMTHCONNECTION>
```

   a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.

   b. For the <m:methodname> tag, replace the String placeholder with the name of the Web service method.

   c. For the <m: descriptor> tag, copy and paste the contents of the <m: descriptor> tag that was returned when you retrieved Web Service method connection information.

9. Modify the contents of the <m: descriptor> tag to change the existing Web Service method connection information.

10. From the SOAP menu, select *Send request to server.*

Starting or Stopping a Channel Programmatically

The following topic describes how to start or stop a channel programmatically.

Procedure: How to Start a Channel Programmatically

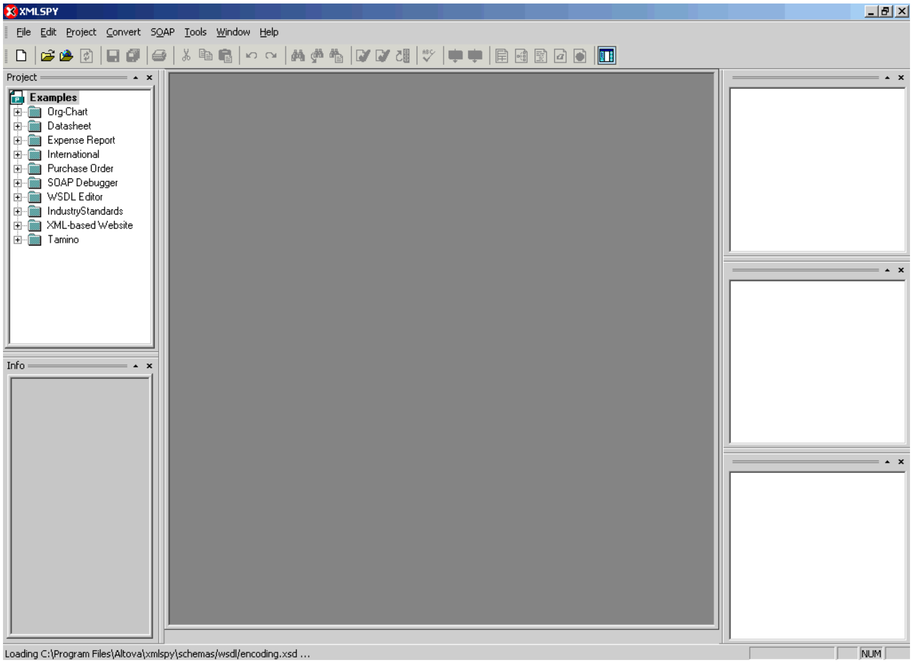
To start a channel programmatically:

1. Copy the iBSE control event URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



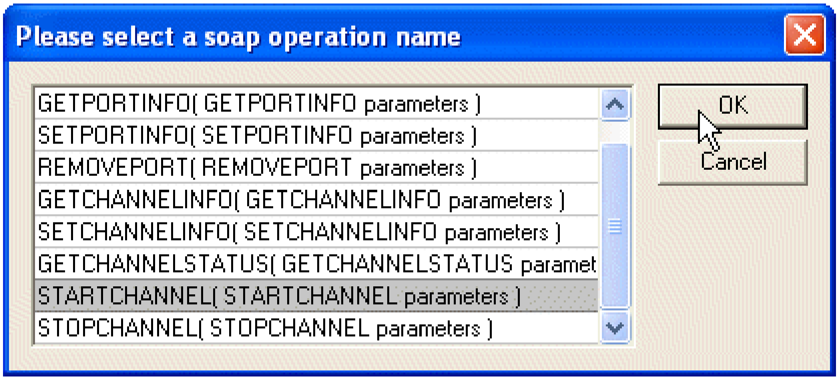
Loading C:\Program Files\Altova\xmlspy\schemas/wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



Please select a soap operation name

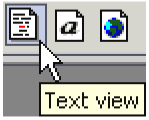
GETPORTINFO(GETPORTINFO parameters)
SETPORTINFO(SETPORTINFO parameters)
REMOVEPORT(REMOVEPORT parameters)
GETCHANNELINFO(GETCHANNELINFO parameters)
SETCHANNELINFO(SETCHANNELINFO parameters)
GETCHANNELSTATUS(GETCHANNELSTATUS paramet
STARTCHANNEL(STARTCHANNEL parameters)
STOPCHANNEL(STOPCHANNEL parameters)

5. Select the *STARTCHANNEL(STARTCHANNEL parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

   In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STARTCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STARTCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the Channel you want to start.

10. From the SOAP menu, select *Send request to server.*

Procedure: How to Stop a Channel Programmatically

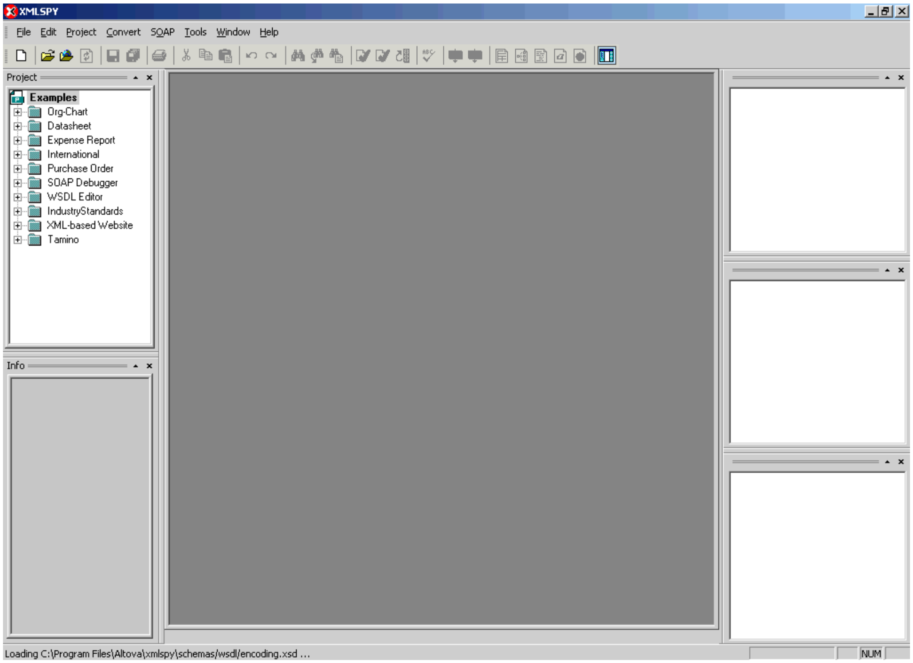
To stop a channel programmatically:

1. Copy the iBSE control event URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



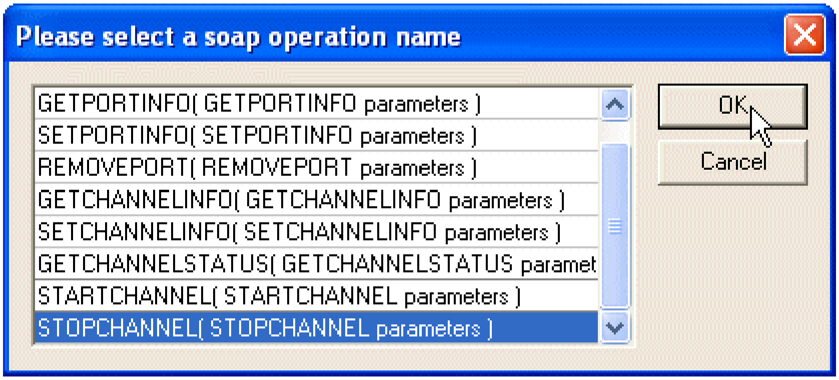
Loading C:\Program Files\Altova\xmlspy\schemas\wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



Please select a soap operation name

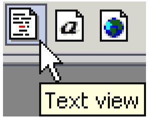
GETPORTINFO(GETPORTINFO parameters)
SETPORTINFO(SETPORTINFO parameters)
REMOVEPORT(REMOVEPORT parameters)
GETCHANNELINFO(GETCHANNELINFO parameters)
SETCHANNELINFO(SETCHANNELINFO parameters)
GETCHANNELSTATUS(GETCHANNELSTATUS paramet
STARTCHANNEL(STARTCHANNEL parameters)
STOPCHANNEL(STOPCHANNEL parameters)

5. Select the *STOPCHANNEL(STOPCHANNEL parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

   In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STOPCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STOPCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the Channel you want to stop.

10. From the SOAP menu, select *Send request to server.*

APPENDIX A

Usage Considerations and Sample Files

Topics:

- Usage Considerations
- Sample Files

This section briefly describes usage considerations for Siebel. It also provides sample XML schemas for Siebel Business Components, Siebel Business Services, and Integration Objects.

Usage Considerations

For Business Components, the iWay Application System Adapter for Siebel enables Insert, Update, Delete, and Query. It also enables a method called QueryWithView. The View modes are a visibility feature provided by Siebel.

By using QueryWithView, you can specify a Siebel View mode as a parameter. The API parameters allow different presentations of data depending on the Siebel environment that you configured.

You can use Query except when you want to enable a user to retrieve records based on different view modes. In this case, use QueryWithView. For more information on QueryWithView mode or Siebel “Visibility” concepts, see your Siebel Administrator.

The following levels are available:

- Sales Rep View
- Manager View
- Personal View
- All View
- Organization View
- Group View
- Catalog View
- SubOrganization View

Sample Files

The following topics includes samples of schemas for Account Business Component.

Account Business Component

The following examples include samples of:

- QueryWithView request schema
- QueryWithView response schema

Example Account QueryWithView Request Schema

```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!--  Generated by the iBSE 2004-04-09T19:36:01Z
  -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:adapter:siebel:bo:oct2003"
xmlns:z="urn:iwaysoftware:adapter:siebel:bo:oct2003" elementFormDefault="qualified">
   <xsd:element name="Siebel">
  <xsd:complexType>
  <xsd:sequence>
  <xsd:element name="select" type="z:record" />
  <xsd:element name="field" minOccurs="0" maxOccurs="unbounded">
  <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 Denorm Flag" />
  <xsd:enumeration value="Assignment Excluded" />
  <xsd:enumeration value="Assignment Manual Flag" />
  <xsd:enumeration value="Assignment System Flag" />
  <xsd:enumeration value="Assignment Type" />
  <xsd:enumeration value="BO Export Status" />
  <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" />
  <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="Category" />
  <xsd:enumeration value="Category Value" />
  <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="Disable Mailings" />
  <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="First Name" />
<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="Full Name" />
<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" />
<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 Name" />
<xsd:enumeration value="Last Review Manager Id" />
<xsd:enumeration value="Last Update - SDQ" />
<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" />
<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" />
  <xsd:enumeration value="Position Id" />
  <xsd:enumeration value="Position Integration Id" />
  <xsd:enumeration value="Position Territory" />
  <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 Contact Address Id" />
<xsd:enumeration value="Primary Employee Id" />
<xsd:enumeration value="Primary Employee Login" />
<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="Sales Rep" />
<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="Territory Item Type" />
<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:element>
</xsd:sequence>
```

```
  <xsd:attribute name="location" type="xsd:string" use="optional"
default="S/BO/Account/Account/queryWithView" />
  <xsd:attribute name="view" use="required">
  <xsd:simpleType>
  <xsd:restriction base="xsd:string">
  <xsd:enumeration value="SalesRepView" />
  <xsd:enumeration value="ManagerView" />
  <xsd:enumeration value="PersonalView" />
  <xsd:enumeration value="AllView" />
  <xsd:enumeration value="NoneSetView" />
  <xsd:enumeration value="OrganizationView" />
  <xsd:enumeration value="ContactView" />
  <xsd:enumeration value="GroupView" />
  <xsd:enumeration value="CatalogView" />
  <xsd:enumeration value="SubOrganizationView" />
  </xsd:restriction>
  </xsd:simpleType>
  </xsd:attribute>
  </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="record">
  <xsd:sequence>
  <xsd:element name="Account_spcCompetitors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcCondition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcMarkets" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcOrganization_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Account_spcProducts" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcRole" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcTrend" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcActive_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcIntegration_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Agreement_spcEnd_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcStart_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Algorithm_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Alias" type="xsd:string" minOccurs="0" />
  <xsd:element name="Annual_spcRevenue" type="xsd:string" minOccurs="0" />
  <xsd:element name="Assignment_spcArea_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Assignment_spcCountry_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcDenorm_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcExcluded" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Assignment_spcManual_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcSystem_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="BO_spcExport_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcDistribution_spcChannel" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcOrder_spcQuery_spcEnd_spc_spcDt"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcOrder_spcQuery_spcStart_spc_spcDt"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcArea" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcArea_spcDivision_spcCode"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcOrganization" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Bill_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcCountry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcFirst_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcJob_spcTitle" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcLast_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcPostal_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Bill_spcTo_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Block_spcCredit_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Business_spcProfile" type="xsd:string" minOccurs="0" />
  <xsd:element name="CSN" type="xsd:string" minOccurs="0" />
  <xsd:element name="Category" type="xsd:string" minOccurs="0" />
  <xsd:element name="Category_spcValue" type="xsd:string" minOccurs="0" />
  <xsd:element name="City" type="xsd:string" minOccurs="0" />
  <xsd:element name="City_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Competitor" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Country" type="xsd:string" minOccurs="0" />
  <xsd:element name="County" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcControl_spcArea_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Credit_spcCurrency_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcLimit_spcAmount" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcProfile_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Culture" type="xsd:string" minOccurs="0" />
  <xsd:element name="Currency_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Current_spcVolume" type="xsd:string" minOccurs="0" />
  <xsd:element name="Current_spcVolume_spcCurrency_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Current_spcVolume_spcExchange_spcDate" type="xsd:date"
minOccurs="0" />
  <xsd:element name="Customer_spcAccount_spcGroup" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="DNBReport" type="xsd:string" minOccurs="0" />
  <xsd:element name="DUNS_spcIntcode" type="xsd:string" minOccurs="0" />
  <xsd:element name="DUNS_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Date_spcFormed" type="xsd:date" minOccurs="0" />
  <xsd:element name="DeDup_spcKey_spcModification_spcDate" type="xsd:string"
minOccurs="0" />
  <xsd:element name="DeDup_spcKey_spcUpdate" type="xsd:string" minOccurs="0" />
  <xsd:element name="DeDup_spcKeys" type="xsd:string" minOccurs="0" />
  <xsd:element name="DeDup_spcLast_spcMatch_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="DeDup_spcToken" type="xsd:string" minOccurs="0" />
  <xsd:element name="Deduplication_spcMatch_spcScore" type="xsd:int" minOccurs="0"
/>
  <xsd:element name="Deduplication_spcObject_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Description" type="xsd:string" minOccurs="0" />
  <xsd:element name="Disable_spcDataCleansing" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Disable_spcMailings" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Division" type="xsd:string" minOccurs="0" />
  <xsd:element name="Domestic_spcUltimate_spcDUNS" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Dummy" type="xsd:string" minOccurs="0" />
  <xsd:element name="EAI_spcSync_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="EAI_spcSync_spcError_spcText" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="EAI_spcSync_spcStatus_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Email_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Employee_spcHere" type="xsd:int" minOccurs="0" />
  <xsd:element name="Employees" type="xsd:int" minOccurs="0" />
  <xsd:element name="Expertise" type="xsd:string" minOccurs="0" />
  <xsd:element name="Explorer_spcLabel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Fax_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="First_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Fiscal_spcYear_spcEnd" type="xsd:date" minOccurs="0" />
  <xsd:element name="Freight_spcTerms" type="xsd:string" minOccurs="0" />
  <xsd:element name="Freight_spcTerms_spcInfo" type="xsd:string" minOccurs="0" />
  <xsd:element name="Full_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Full_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="GSA_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Global_spcUltimate_spcDUNS" type="xsd:string" minOccurs="0" />
  <xsd:element name="Goals" type="xsd:string" minOccurs="0" />
  <xsd:element name="Group_spcType_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Home_spcPage" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry_spcCondition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry_spcTrend" type="xsd:string" minOccurs="0" />
  <xsd:element name="Integration_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Internal_spcOrg_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Joined_spcSynonym" type="xsd:string" minOccurs="0" />
  <xsd:element name="Key_spcCompetitors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Language_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcClnse_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcManager_spcReview_spcDate" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Last_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcReview_spcManager_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Last_spcUpdate_spc-_spcSDQ" type="xsd:string" minOccurs="0" />
  <xsd:element name="Line_spcof_spcBusiness" type="xsd:string" minOccurs="0" />
  <xsd:element name="Location" type="xsd:string" minOccurs="0" />
  <xsd:element name="Location_spcLevel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcFax_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcPhone_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Managers_spcReview" type="xsd:string" minOccurs="0" />
  <xsd:element name="Marketing" type="xsd:string" minOccurs="0" />
  <xsd:element name="Merge_spcSequence_spcNumber" type="xsd:int" minOccurs="0" />
  <xsd:element name="Mission" type="xsd:string" minOccurs="0" />
  <xsd:element name="Name" type="xsd:string" minOccurs="0" />
  <xsd:element name="Name_spcand_spcLocation" type="xsd:string" minOccurs="0" />
  <xsd:element name="Not_spcManager_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Notes" type="xsd:string" minOccurs="0" />
  <xsd:element name="Objectives" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Our_spcPosition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Outline_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="PO_spcApproved_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="PO_spcAuto_spcApproval_spcCurrency_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="PO_spcAuto_spcApproval_spcDate" type="xsd:date" minOccurs="0"
/>
  <xsd:element name="PO_spcAuto_spcApproval_spcLimit" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcDivision" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Parent_spcAccount_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcLocation" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Parent_spcAccount_spcLocation_spcLevel" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcRegion" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcHQ_spcDUNS" type="xsd:string" minOccurs="0" />
  <xsd:element name="Partner_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Partners" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcType_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcUId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Philosophy" type="xsd:string" minOccurs="0" />
  <xsd:element name="Phone_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position_spcIntegration_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Position_spcTerritory" type="xsd:string" minOccurs="0" />
  <xsd:element name="Postal_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList_spcEnd_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Price_spcList_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Price_spcList_spcStart_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcAccount_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcCountry" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcAccount_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcAddress_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcDenorm_spcFlag" type="xsd:boolean"
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcManual_spcFlag" type="xsd:boolean"
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcSystem_spcFlag" type="xsd:boolean"
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcType" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcAddress_spcId" type="xsd:string"
```

```
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcCity" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcCountry" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcFirst_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcJob_spcTitle" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcLast_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcPerson_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcState" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcCategory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcContact_spcAddress_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcEmployee_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcEmployee_spcLogin" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcFulfill_spcInvLoc_spcIntegration_spcId"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcFulfillment_spcInvLoc_spcID" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcFulfillment_spcInventory_spcLocation"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcIndustry_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcOrganization" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcOrganization_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcPayer_spcAccount" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcPayer_spcAccount_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcPosition_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcService_spcAgreement_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcAddress_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcCity" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcShip_spcTo_spcCountry" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcFirst_spcName" type="xsd:string"
```

```
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcJob_spcTitle" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcLast_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcPerson_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcState" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcShip_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcSynonym_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcTerritory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcType_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Profit" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcBill_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcComments" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcFix_spcFee" type="xsd:int" minOccurs="0" />
  <xsd:element name="Project_spcHour_spcLimit" type="xsd:int" minOccurs="0" />
  <xsd:element name="Project_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcPercentage_spcof_spcFee" type="xsd:int"
minOccurs="0" />
  <xsd:element name="Project_spcPurchase_spcOrder" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Project_spcRelationship_spcType" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Project_spcRole" type="xsd:string" minOccurs="0" />
  <xsd:element name="Prospect_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Province" type="xsd:string" minOccurs="0" />
  <xsd:element name="Public" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Reference_spcDate" type="xsd:date" minOccurs="0" />
  <xsd:element name="Reference_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Reference_spcStage" type="xsd:string" minOccurs="0" />
  <xsd:element name="Region" type="xsd:string" minOccurs="0" />
  <xsd:element name="Relationship_spcLevel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Relationship_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Response_spcTime" type="xsd:string" minOccurs="0" />
  <xsd:element name="Revenue" type="xsd:string" minOccurs="0" />
  <xsd:element name="Revenue_spcGrowth" type="xsd:int" minOccurs="0" />
  <xsd:element name="Revision_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Row_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Row_spcStatus_spcAsterisk" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcInstance" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcInstance_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcKey_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Sales_spcRep" type="xsd:string" minOccurs="0" />
  <xsd:element name="Service_spcCalendar" type="xsd:string" minOccurs="0" />
  <xsd:element name="Service_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcCountry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcFirst_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcJob_spcTitle" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcLast_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcPostal_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Ship_spcTo_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Start_spcDate" type="xsd:date" minOccurs="0" />
  <xsd:element name="State" type="xsd:string" minOccurs="0" />
  <xsd:element name="Strategies" type="xsd:string" minOccurs="0" />
  <xsd:element name="Strategy" type="xsd:string" minOccurs="0" />
  <xsd:element name="Street_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Street_spcAddress_spc2" type="xsd:string" minOccurs="0" />
  <xsd:element name="Success_spcFactors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Synonym" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory_spcItem_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Timestamp" type="xsd:string" minOccurs="0" />
  <xsd:element name="Today" type="xsd:date" minOccurs="0" />
  <xsd:element name="Total_spcPotential_spcVolume" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Total_spcPotential_spcVolume_spcCurrency_spcCode"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Total_spcPotential_spcVolume_spcExchange_spcDate"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Type" type="xsd:string" minOccurs="0" />
  <xsd:element name="Type_spcMVF" type="xsd:string" minOccurs="0" />
  <xsd:element name="VAT_spcregistration_spcnumber" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Value_spcProposition" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
  </xsd:complexType>
  </xsd:schema>
```

Example Account QueryWithView Response Schema

```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!--  Generated by the iBSE 2004-04-09T19:44:34Z
  -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:adapter:siebel:bo:oct2003"
xmlns:z="urn:iwaysoftware:adapter:siebel:bo:oct2003" elementFormDefault="qualified">
  <xsd:element name="SiebelResponse">
  <xsd:complexType>
  <xsd:sequence>
  <xsd:element name="record" minOccurs="0" maxOccurs="unbounded" type="z:record" />
  </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="optional" />
  </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="record">
  <xsd:sequence>
  <xsd:element name="Account_spcCompetitors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcCondition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcMarkets" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcOrganization_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Account_spcProducts" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcRole" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Account_spcTrend" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcActive_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Address_spcIntegration_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Agreement_spcEnd_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcStart_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Agreement_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Algorithm_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Alias" type="xsd:string" minOccurs="0" />
  <xsd:element name="Annual_spcRevenue" type="xsd:string" minOccurs="0" />
  <xsd:element name="Assignment_spcArea_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Assignment_spcCountry_spcCode" type="xsd:string" minOccurs="0"
```

```
/>
  <xsd:element name="Assignment_spcDenorm_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcExcluded" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Assignment_spcManual_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcSystem_spcFlag" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Assignment_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="BO_spcExport_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcDistribution_spcChannel" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcOrder_spcQuery_spcEnd_spc_spcDt"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcOrder_spcQuery_spcStart_spc_spcDt"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcArea" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcArea_spcDivision_spcCode"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Back_spcOffice_spcSales_spcOrganization" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Bill_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcCountry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcFirst_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcJob_spcTitle" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcLast_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcPostal_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Bill_spcTo_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Bill_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Block_spcCredit_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Business_spcProfile" type="xsd:string" minOccurs="0" />
  <xsd:element name="CSN" type="xsd:string" minOccurs="0" />
  <xsd:element name="Category" type="xsd:string" minOccurs="0" />
  <xsd:element name="Category_spcValue" type="xsd:string" minOccurs="0" />
  <xsd:element name="City" type="xsd:string" minOccurs="0" />
  <xsd:element name="City_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Competitor" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Country" type="xsd:string" minOccurs="0" />
  <xsd:element name="County" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcControl_spcArea_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Credit_spcCurrency_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcLimit_spcAmount" type="xsd:string" minOccurs="0" />
  <xsd:element name="Credit_spcProfile_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Culture" type="xsd:string" minOccurs="0" />
  <xsd:element name="Currency_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Current_spcVolume" type="xsd:string" minOccurs="0" />
  <xsd:element name="Current_spcVolume_spcCurrency_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Current_spcVolume_spcExchange_spcDate" type="xsd:date"
minOccurs="0" />
  <xsd:element name="Customer_spcAccount_spcGroup" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="DNBReport" type="xsd:string" minOccurs="0" />
  <xsd:element name="DUNS_spcIntcode" type="xsd:string" minOccurs="0" />
  <xsd:element name="DUNS_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Date_spcFormed" type="xsd:date" minOccurs="0" />
  <xsd:element name="DeDup_spcKey_spcModification_spcDate" type="xsd:string"
minOccurs="0" />
  <xsd:element name="DeDup_spcKey_spcUpdate" type="xsd:string" minOccurs="0" />
  <xsd:element name="DeDup_spcKeys" type="xsd:string" minOccurs="0" />
  <xsd:element name="DeDup_spcLast_spcMatch_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="DeDup_spcToken" type="xsd:string" minOccurs="0" />
  <xsd:element name="Deduplication_spcMatch_spcScore" type="xsd:int" minOccurs="0"
/>
  <xsd:element name="Deduplication_spcObject_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Description" type="xsd:string" minOccurs="0" />
  <xsd:element name="Disable_spcDataCleansing" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Disable_spcMailings" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Division" type="xsd:string" minOccurs="0" />
  <xsd:element name="Domestic_spcUltimate_spcDUNS" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Dummy" type="xsd:string" minOccurs="0" />
  <xsd:element name="EAI_spcSync_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="EAI_spcSync_spcError_spcText" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="EAI_spcSync_spcStatus_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Email_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Employee_spcHere" type="xsd:int" minOccurs="0" />
  <xsd:element name="Employees" type="xsd:int" minOccurs="0" />
  <xsd:element name="Expertise" type="xsd:string" minOccurs="0" />
  <xsd:element name="Explorer_spcLabel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Fax_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="First_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Fiscal_spcYear_spcEnd" type="xsd:date" minOccurs="0" />
  <xsd:element name="Freight_spcTerms" type="xsd:string" minOccurs="0" />
  <xsd:element name="Freight_spcTerms_spcInfo" type="xsd:string" minOccurs="0" />
  <xsd:element name="Full_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Full_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="GSA_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Global_spcUltimate_spcDUNS" type="xsd:string" minOccurs="0" />
  <xsd:element name="Goals" type="xsd:string" minOccurs="0" />
  <xsd:element name="Group_spcType_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Home_spcPage" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry_spcCondition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Industry_spcTrend" type="xsd:string" minOccurs="0" />
  <xsd:element name="Integration_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Internal_spcOrg_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Joined_spcSynonym" type="xsd:string" minOccurs="0" />
  <xsd:element name="Key_spcCompetitors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Language_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcClnse_spcDate" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcManager_spcReview_spcDate" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Last_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Last_spcReview_spcManager_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Last_spcUpdate_spc-_spcSDQ" type="xsd:string" minOccurs="0" />
  <xsd:element name="Line_spcof_spcBusiness" type="xsd:string" minOccurs="0" />
  <xsd:element name="Location" type="xsd:string" minOccurs="0" />
  <xsd:element name="Location_spcLevel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcFax_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Main_spcPhone_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Managers_spcReview" type="xsd:string" minOccurs="0" />
  <xsd:element name="Marketing" type="xsd:string" minOccurs="0" />
  <xsd:element name="Merge_spcSequence_spcNumber" type="xsd:int" minOccurs="0" />
  <xsd:element name="Mission" type="xsd:string" minOccurs="0" />
  <xsd:element name="Name" type="xsd:string" minOccurs="0" />
  <xsd:element name="Name_spcand_spcLocation" type="xsd:string" minOccurs="0" />
  <xsd:element name="Not_spcManager_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Notes" type="xsd:string" minOccurs="0" />
  <xsd:element name="Objectives" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Organization_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Our_spcPosition" type="xsd:string" minOccurs="0" />
  <xsd:element name="Outline_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="PO_spcApproved_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="PO_spcAuto_spcApproval_spcCurrency_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="PO_spcAuto_spcApproval_spcDate" type="xsd:date" minOccurs="0"
/>
  <xsd:element name="PO_spcAuto_spcApproval_spcLimit" type="xsd:string"
```

```
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcDivision" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Parent_spcAccount_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcLocation" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Parent_spcAccount_spcLocation_spcLevel" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcAccount_spcRegion" type="xsd:string" minOccurs="0" />
  <xsd:element name="Parent_spcHQ_spcDUNS" type="xsd:string" minOccurs="0" />
  <xsd:element name="Partner_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Partners" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcType_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Party_spcUId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Philosophy" type="xsd:string" minOccurs="0" />
  <xsd:element name="Phone_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position_spcIntegration_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Position_spcTerritory" type="xsd:string" minOccurs="0" />
  <xsd:element name="Postal_spcCode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList_spcEnd_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Price_spcList_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Price_spcList_spcIntegration_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Price_spcList_spcStart_spcDate" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcAccount_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcCountry" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcAccount_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAccount_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcAddress_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcDenorm_spcFlag" type="xsd:boolean"
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcManual_spcFlag" type="xsd:boolean"
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcSystem_spcFlag" type="xsd:boolean"
```

```
minOccurs="0" />
  <xsd:element name="Primary_spcAssignment_spcType" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcAddress_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcCity" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcCountry" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcFirst_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcJob_spcTitle" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcLast_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcPerson_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcBill_spcTo_spcState" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcBill_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcCategory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcContact_spcAddress_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcEmployee_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcEmployee_spcLogin" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcFulfill_spcInvLoc_spcIntegration_spcId"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcFulfillment_spcInvLoc_spcID" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcFulfillment_spcInventory_spcLocation"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcIndustry_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcOrganization" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcOrganization_spcId" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcPayer_spcAccount" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcPayer_spcAccount_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcPosition_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcService_spcAgreement_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcAddress_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcCity" type="xsd:string" minOccurs="0"
```

```
/>
  <xsd:element name="Primary_spcShip_spcTo_spcCountry" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcFirst_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcJob_spcTitle" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcLast_spcName" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcPerson_spcId" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcPostal_spcCode" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcShip_spcTo_spcState" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Primary_spcShip_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Primary_spcSynonym_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcTerritory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Primary_spcType_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Profit" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcBill_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcComments" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcFix_spcFee" type="xsd:int" minOccurs="0" />
  <xsd:element name="Project_spcHour_spcLimit" type="xsd:int" minOccurs="0" />
  <xsd:element name="Project_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Project_spcPercentage_spcof_spcFee" type="xsd:int"
minOccurs="0" />
  <xsd:element name="Project_spcPurchase_spcOrder" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Project_spcRelationship_spcType" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Project_spcRole" type="xsd:string" minOccurs="0" />
  <xsd:element name="Prospect_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Province" type="xsd:string" minOccurs="0" />
  <xsd:element name="Public" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Reference_spcDate" type="xsd:date" minOccurs="0" />
  <xsd:element name="Reference_spcFlag" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Reference_spcStage" type="xsd:string" minOccurs="0" />
  <xsd:element name="Region" type="xsd:string" minOccurs="0" />
  <xsd:element name="Relationship_spcLevel" type="xsd:string" minOccurs="0" />
  <xsd:element name="Relationship_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Response_spcTime" type="xsd:string" minOccurs="0" />
  <xsd:element name="Revenue" type="xsd:string" minOccurs="0" />
  <xsd:element name="Revenue_spcGrowth" type="xsd:int" minOccurs="0" />
  <xsd:element name="Revision_spcNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="Row_spcStatus" type="xsd:string" minOccurs="0" />
  <xsd:element name="Row_spcStatus_spcAsterisk" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcInstance" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcInstance_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="S-S_spcKey_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Sales_spcRep" type="xsd:string" minOccurs="0" />
  <xsd:element name="Service_spcCalendar" type="xsd:string" minOccurs="0" />
  <xsd:element name="Service_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcAddress_spcFlag" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcCity" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcCountry" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcFirst_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcJob_spcTitle" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcLast_spcName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcPostal_spcCode" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Ship_spcTo_spcState" type="xsd:string" minOccurs="0" />
  <xsd:element name="Ship_spcTo_spcStreet_spcAddress" type="xsd:string"
minOccurs="0" />
  <xsd:element name="Start_spcDate" type="xsd:date" minOccurs="0" />
  <xsd:element name="State" type="xsd:string" minOccurs="0" />
  <xsd:element name="Strategies" type="xsd:string" minOccurs="0" />
  <xsd:element name="Strategy" type="xsd:string" minOccurs="0" />
  <xsd:element name="Street_spcAddress" type="xsd:string" minOccurs="0" />
  <xsd:element name="Street_spcAddress_spc2" type="xsd:string" minOccurs="0" />
  <xsd:element name="Success_spcFactors" type="xsd:string" minOccurs="0" />
  <xsd:element name="Synonym" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory_spcId" type="xsd:string" minOccurs="0" />
  <xsd:element name="Territory_spcItem_spcType" type="xsd:string" minOccurs="0" />
  <xsd:element name="Timestamp" type="xsd:string" minOccurs="0" />
  <xsd:element name="Today" type="xsd:date" minOccurs="0" />
  <xsd:element name="Total_spcPotential_spcVolume" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Total_spcPotential_spcVolume_spcCurrency_spcCode"
type="xsd:string" minOccurs="0" />
  <xsd:element name="Total_spcPotential_spcVolume_spcExchange_spcDate"
type="xsd:date" minOccurs="0" />
  <xsd:element name="Type" type="xsd:string" minOccurs="0" />
  <xsd:element name="Type_spcMVF" type="xsd:string" minOccurs="0" />
  <xsd:element name="VAT_spcregistration_spcnumber" type="xsd:string" minOccurs="0"
/>
  <xsd:element name="Value_spcProposition" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
  </xsd:complexType>
  </xsd:schema>
```

Example XML Input for Account QueryWithView

```
<Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
            <select>
                        <Name>SIEBEL*</Name>
            </select>
            <field>Name</field>
            <field>Location</field>
  </Siebel>
Sample XML for Account Add Service Response
```

Example XML for Account QueryWithView Response

```
<?xml version="1.0" encoding="UTF-8" ?>
  <SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
  <NewAccountQueryResponse
xmlns="urn:iwaysoftware:ibse:jul2003:NewAccountQuery:response"
cid="B24B35E1A39476D1D9EFB8F2A0F4818A">
  <SiebelResponse status="success">
  <record>
  <Name>SIEBEL1 ACCOUNT</Name>
  <Location>ONE</Location>
  </record>
  <record>
  <Name>SIEBEL2 ACCOUNT</Name>
  <Location>TWO</Location>
  </record>
  <record>
  <Name>SIEBEL3</Name>
  <Location>RR</Location>
  </record>
  <record>
  <Name>SIEBEL4</Name>
  <Location>FOUR</Location>
  </record>
  </SiebelResponse>
  </NewAccountQueryResponse>
  </SOAP-ENV:Body>

  </SOAP-ENV:Envelope>
```

APPENDIX B

Siebel Workflows

Topics:

- Overview
- Creating a Siebel Workflow

This section describes Siebel Workflows relating to the processing of Siebel Integration Objects using Siebel XML.

Overview

When using Siebel XML to integrate with Siebel Integration Objects, the interface uses a Siebel Workflow. A Siebel Workflow is defined within Siebel to emit or to receive Siebel XML. In either case, emitting or receiving is handled by Siebel transport services for MQSeries, File, or HTTP. The following topics discuss the use and creation of workflows that employ the supported transport services.

Note: This section is intended as a supplement to the documentation designed for the iWay Application System Adapter for Siebel user and is not intended as a substitute for Siebel documentation. For complete and up-to-date information on Siebel Workflow and policy topics, see the *Siebel Bookshelf* for your Siebel system.

Siebel Workflows

A Siebel Workflow is a series of Siebel Business Services linked together to accomplish a business task. You create workflows using the Siebel Client Workflow Administration screens. Workflows are invoked through one of the following methods:

- Using a workflow policy
- Using a run-time event (Siebel Event)
- Using a script (eScript or Siebel VB)

The following topic briefly describes how to invoke the workflow through a policy condition. For more information on policy and other methods, see the documentation on the *Siebel Bookshelf*.

Using a Policy to Invoke a Siebel EAI Workflow

A workflow policy is defined by a set of conditions that executes a set of defined actions. A Siebel workflow policy consists of:

- Conditions that define circumstances, based on changes in the state of a Siebel database.
- Actions that define steps taken when conditions are fulfilled.

Creating a policy to invoke a workflow as an action involves the following steps:

1. Define an action to be executed after a policy is triggered. Use the Run Integration Process program.
2. Create a policy by setting conditions and selecting appropriate policy groups and actions.
3. Activate the policy by choosing an activation date.

4. Run the Generate Triggers server task from Server Administration windows to set the conditions to be monitored.
5. Start the Workflow Monitor agent after editing with the appropriate policy group (to which your policy belongs) to evaluate whether to perform an action.
6. Start the Workflow Action Agent server task from Server Administration windows to perform the action.

For more information on the previous steps, see the documentation on the *Siebel Bookshelf*.

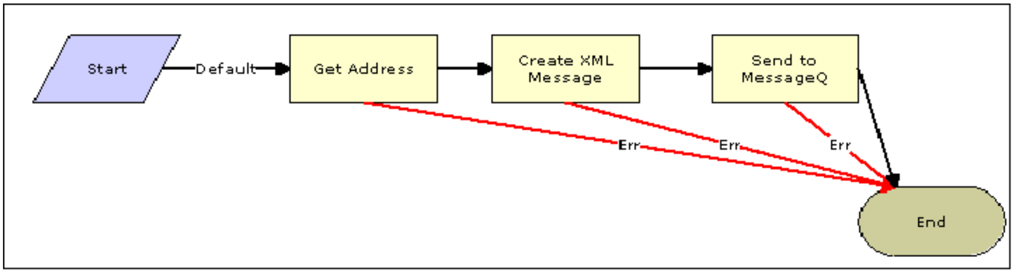
Siebel Workflow - Outbound

When a Siebel Workflow is triggered based on a Siebel policy, run-time, or script (eScript or Siebel VB) event, the result is the generation of a Siebel XML document that is placed on one of the Siebel transports. For example, when you add a new account in the Siebel Call Center application, you can design and configure a workflow to be triggered on the account transaction. You can design the workflow to extract the data for the new record, convert it to Siebel XML, and then, place it on an MQSeries message queue.

In this example, the Siebel Workflow process executes the following series of Siebel Business Services:

1. Calls the Siebel EAI Siebel Adapter, which queries for the newly updated account record, and places the data in its original internal structure into memory.
2. Calls the Siebel EAI XML Converter, which converts the data into an XML message.
3. Calls the Siebel EAI MQSeries Transport, which places the newly created XML message into the appropriate MQSeries message queue.

   After the message is placed in the message queue, it is retrieved by an iWay Application System Adapter for Siebel 6.3 and higher. The following illustration shows the Workflow sequence described in the previous steps. The flow boxes are from left to right: Start, Get Address, Create XML Message, Send to MessageQ, and End.



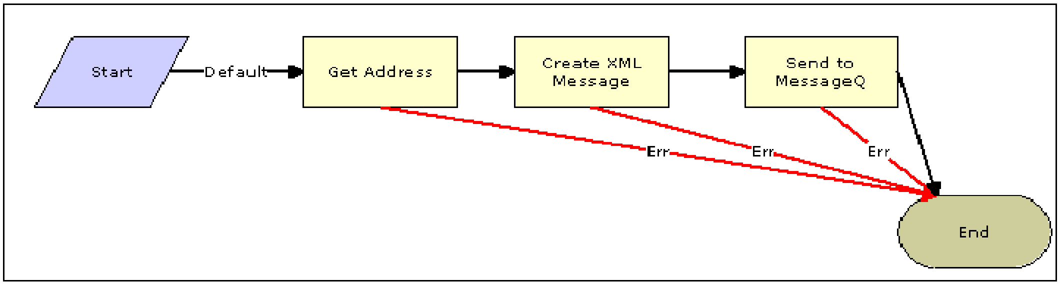
Siebel Workflow - Inbound

A Siebel Workflow that is triggered by an external event begins by receiving a Siebel XML document placed on one of its transports. The result might be the update of a Siebel record using the XML as input, for example, when a new account is added in another CRM system but also must be updated in the Siebel Call Center application. You can design and configure a Workflow to receive or listen on an MQSeries message queue. Upon receipt of the XML message, the Workflow processes the transaction into the Siebel system to update the record.

In this example, upon receipt of the Siebel XML message in the message queue, the Siebel MQSeries Receiver server task initiates a Siebel Workflow process, which in turn executes a series of Siebel Business Services as follows:

1. Calls the Siebel EAI XML Converter, which converts the XML message into Siebel internal format.
2. Calls the Siebel EAI Siebel Adapter, which applies the newly updated account record based on the methods defined in its service.

The following illustration shows the inbound Workflow process based on the previous description. The flow boxes are from left to right: Start, Get Address, Create XML Message, Send to MessageQ, and End.



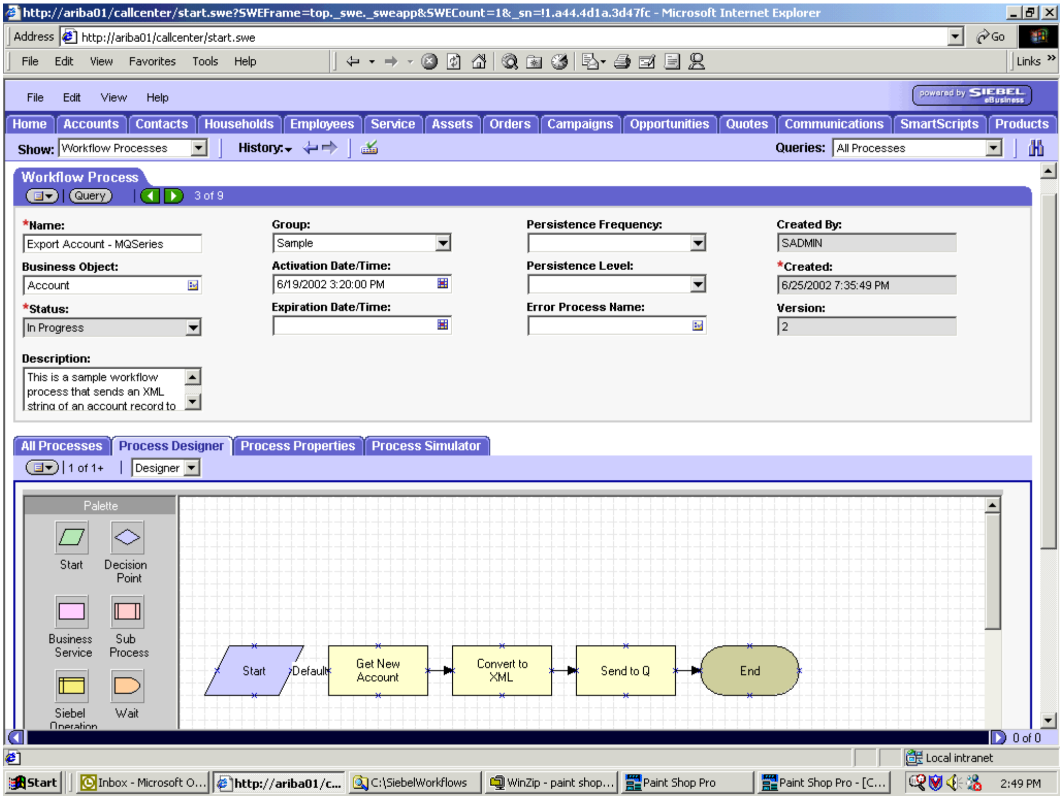
Creating a Siebel Workflow

The following topics include procedures for creating Siebel Workflows in the Siebel Workflow Administration window.

Creating a Siebel Workflow for an Event Using MQSeries Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the MQSeries transport.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Designer tab is active and displays an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.



Show: Workflow Processes

Queries: All Processes

*Name: Export Account - MQSeries

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that sends an XML string of an account record to

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 6/25/2002 7:35:49 PM

Version: 2

Procedure: How to Create a Siebel Workflow for an Event Using MQSeriesTransport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application. The Workflow is then placed on an MQSeries message queue.

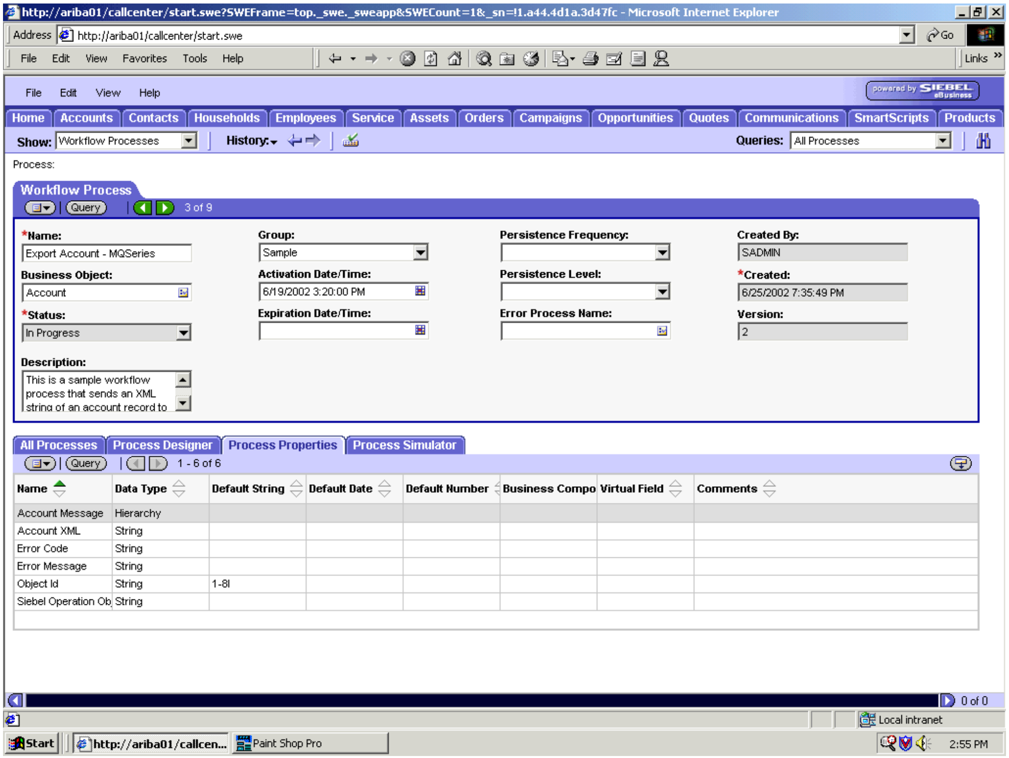
To create a Siebel Workflow:

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.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account information for a new Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Process:
Workflow Process
Query
3 of 9
All Processes
Process Designer
Process Properties
Process Simulator
Query
1 - 6 of 6
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
2:55 PM

Show: Workflow Processes

Queries: All Processes

*Name: Export Account - MQSeries

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that sends an XML string of an account record to

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 6/25/2002 7:35:49 PM

Version: 2

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

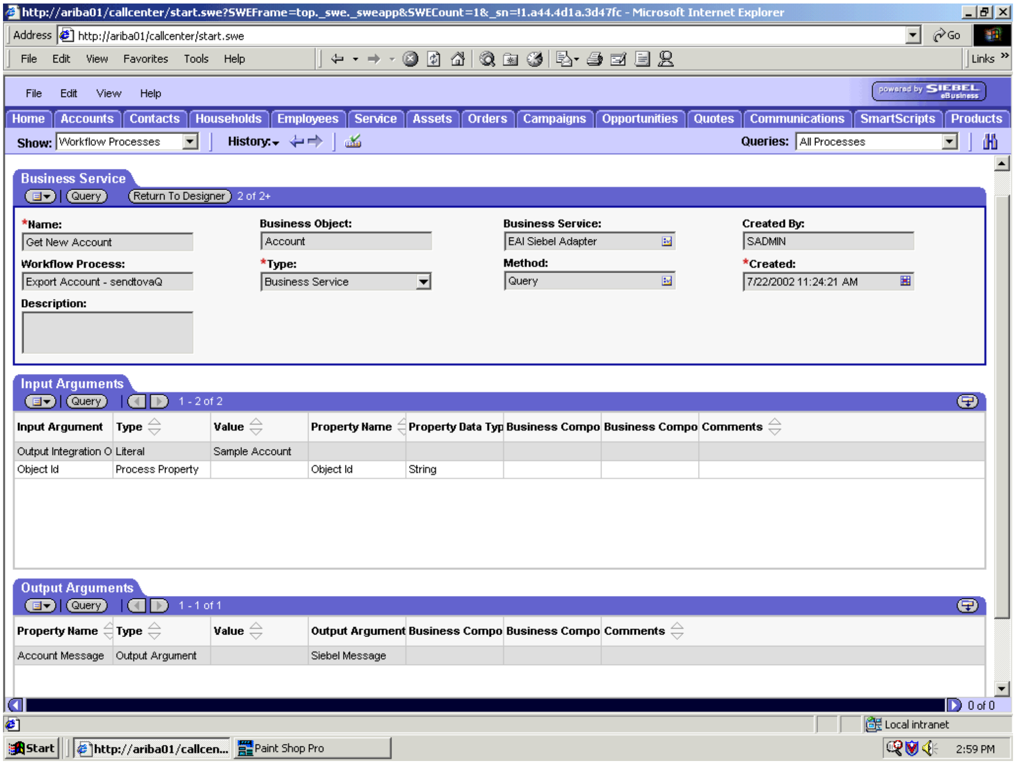
2. Use the Siebel Workflow Administration windows to create a Workflow.

3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

   Using the Query method, the Business Service obtains the Account information from Siebel.

   Output from this Business Service is generated in hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining a new Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
2 of 2+
1 - 2 of 2
1 - 1 of 1
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
2:59 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account - sendtovaQ	*Type: Business Service	Method: Query	*Created: 7/22/2002 11:24:21 AM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

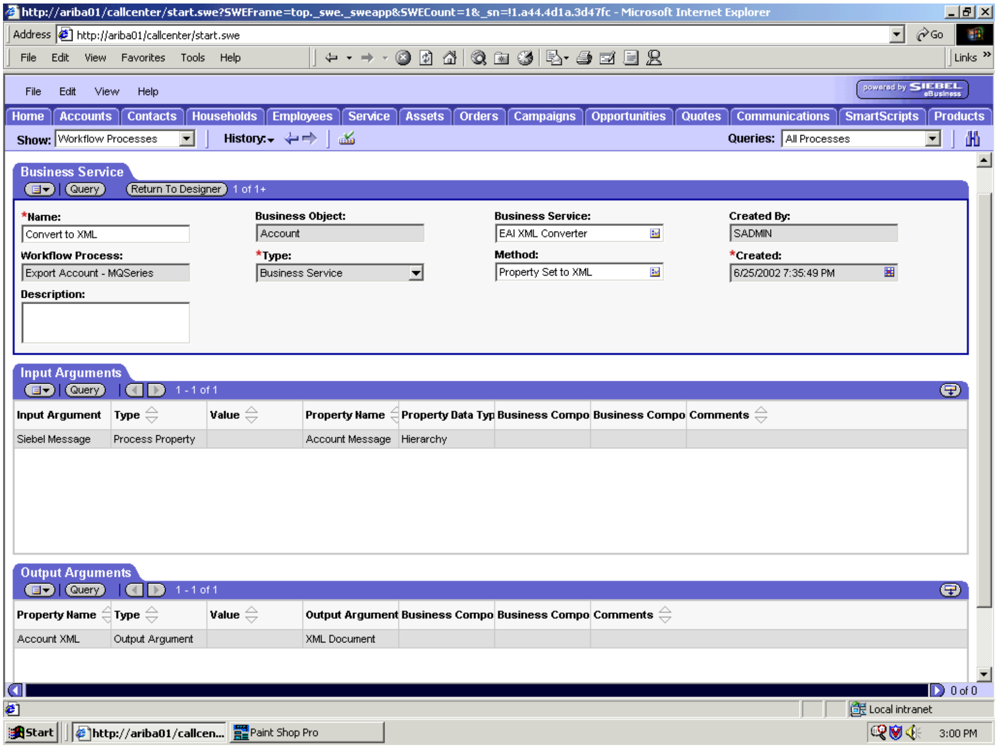
Output Arguments

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

4. Define an EAI XML Converter Business Service step and call it *Convert to XML*.

   It is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Convert to XML Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
1 of 1+
1 - 1 of 1
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:00 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Convert to XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Export Account - MQSeries	*Type: Business Service	Method: Property Set to XML	*Created: 6/25/2002 7:35:49 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

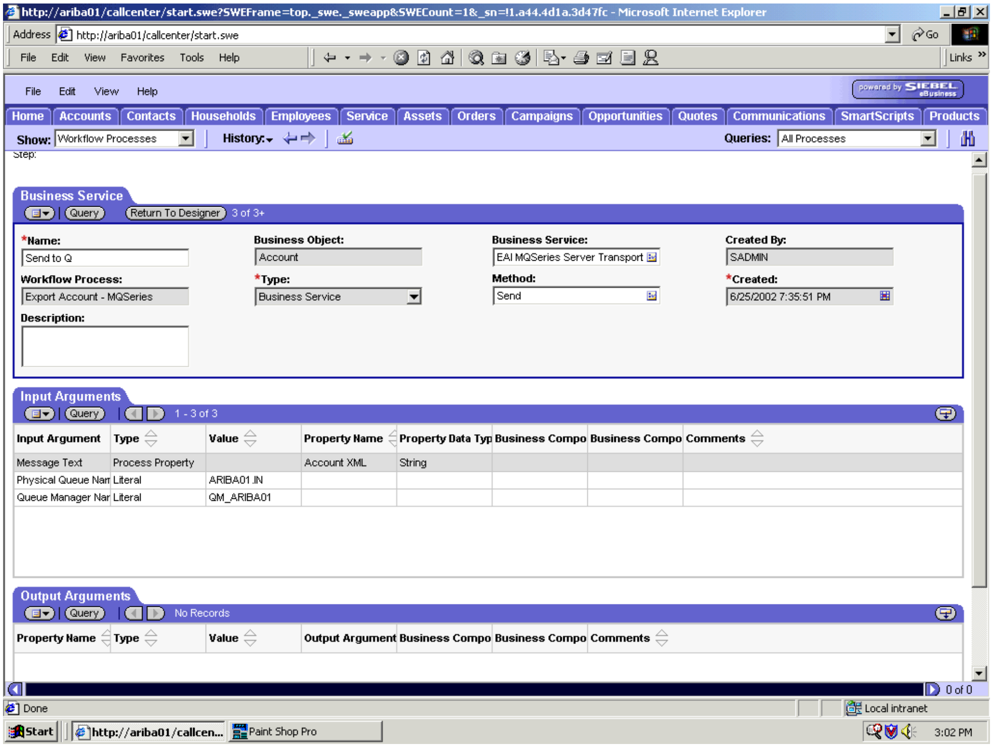
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		XML Document			

5. Define an EAI MQSeries server transport Business Service step and call it *Send to Q*.

   It is 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.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Send to Q Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Step:
Query
Return To Designer
3 of 3+
Query
1 - 3 of 3
Query
No Records
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:02 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Send to Q	Business Object: Account	Business Service: EAI MQSeries Server Transport	Created By: SADMIN
Workflow Process: Export Account - MQSeries	*Type: Business Service	Method: Send	*Created: 6/25/2002 7:35:51 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
Physical Queue Nam	Literal	ARIBA01.IN					
Queue Manager Nar	Literal	QM_ARIBA01					

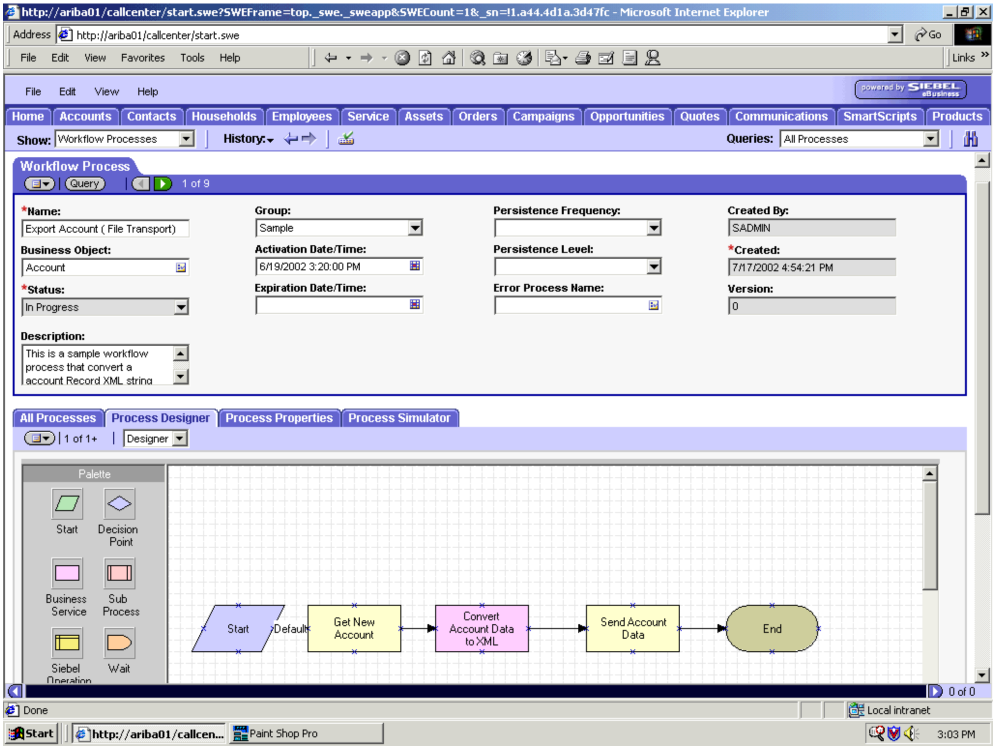
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

Creating a Siebel Workflow for an Event Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow is designed for exporting Siebel Account record information using the File transport.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Designer tab is active and shows an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Go
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Workflow Process
Query
1 of 9
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Siebel Operation
Wait
Start
Default
Get New Account
Convert Account Data to XML
Send Account Data
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:03 PM

*Name: Export Account (File Transport)

Group: Sample

Persistence Frequency:

Created By: SADMIN

Business Object: Account

Activation Date/Time: 6/19/2002 3:20:00 PM

Persistence Level:

*Created: 7/17/2002 4:54:21 PM

*Status: In Progress

Expiration Date/Time:

Error Process Name:

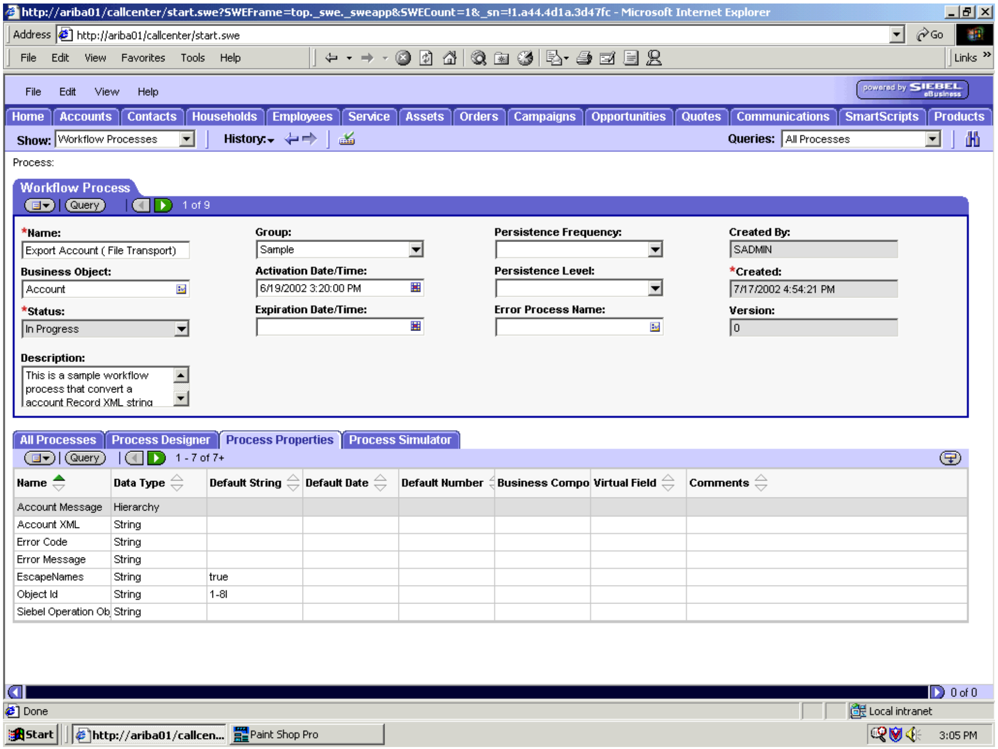
Version: 0

Description: This is a sample workflow process that convert a account Record XML string

Procedure: How to Create a Siebel Workflow for an Event Using File Transport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Process:
Workflow Process
Query
1 of 9
All Processes Process Designer Process Properties Process Simulator
Query
1 - 7 of 7+
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:05 PM

Show: Workflow Processes

Queries: All Processes

*Name: Export Account (File Transport)

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that convert a account Record XML string

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 7/17/2002 4:54:21 PM

Version: 0

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
EscapeNames	String	true					
Object Id	String	1-8I					
Siebel Operation Ob	String						

To create a Siebel Workflow:

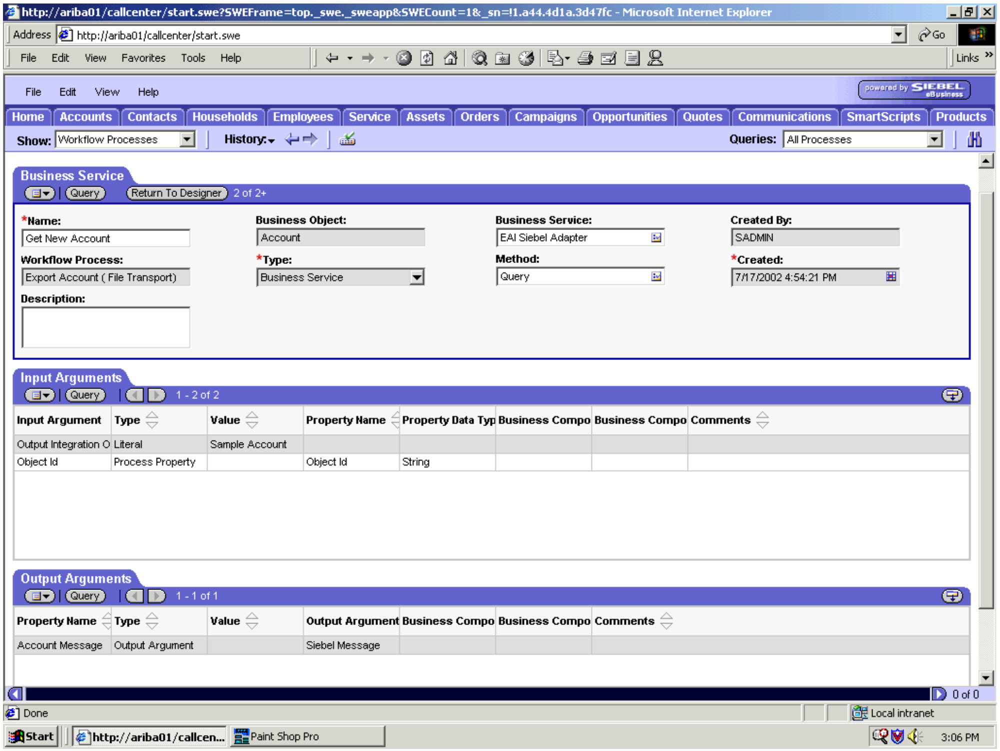
1. On the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies which Siebel Account data the Workflow converted to XML.

2. Use the Siebel Workflow Administration windows to create a Workflow.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Get New Account Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
2 of 2+
1 - 2 of 2
1 - 1 of 1
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:06 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account (File Transport)	*Type: Business Service	Method: Query	*Created: 7/17/2002 4:54:21 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

Output Arguments

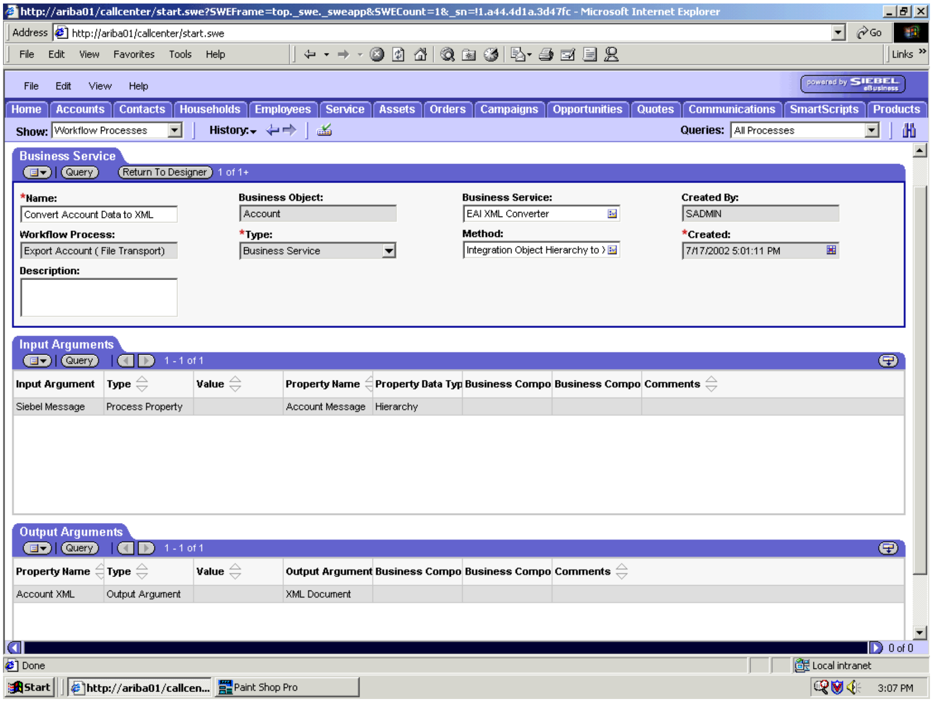
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

   Using the Query method, the Business Service obtains the Account information from Siebel.

   Output from this Business Service is generated in hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Convert Account Data to XML Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:07 PM

Show: Workflow Processes

Queries: All Processes

Business Service

1 of 1+

*Name: Convert Account Data to XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Export Account (File Transport)	*Type: Business Service	Method: Integration Object Hierarchy to ›	*Created: 7/17/2002 5:01:11 PM
Description:			

Input Arguments

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

1 - 1 of 1

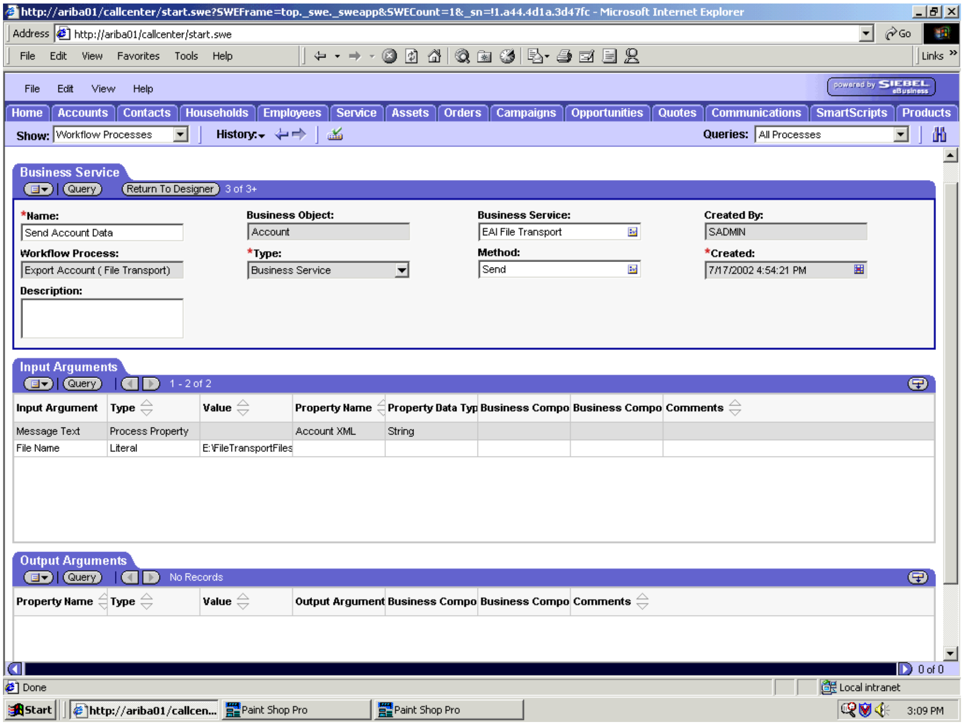
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		XML Document			

0 of 0

4. Define an EAI XML Converter Business Service step and call it *Convert Account Data to XML*.

   This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Send Account Data Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Business Service
Query
Return To Designer
3 of 3+
Input Arguments
Query
1 - 2 of 2
Output Arguments
Query
No Records
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
Paint Shop Pro
3:09 PM

Show: Workflow Processes

Queries: All Processes

*Name: Send Account Data

Workflow Process: Export Account (File Transport)

Description:

Business Object: Account

*Type: Business Service

Business Service: EAI File Transport

Method: Send

Created By: SADMIN

*Created: 7/17/2002 4:54:21 PM

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
File Name	Literal	E:\FileTransportFiles					

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

5. Define an EAI File transport Business Service step and call it *Send Account Data*.

   This Business Service is defined 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.

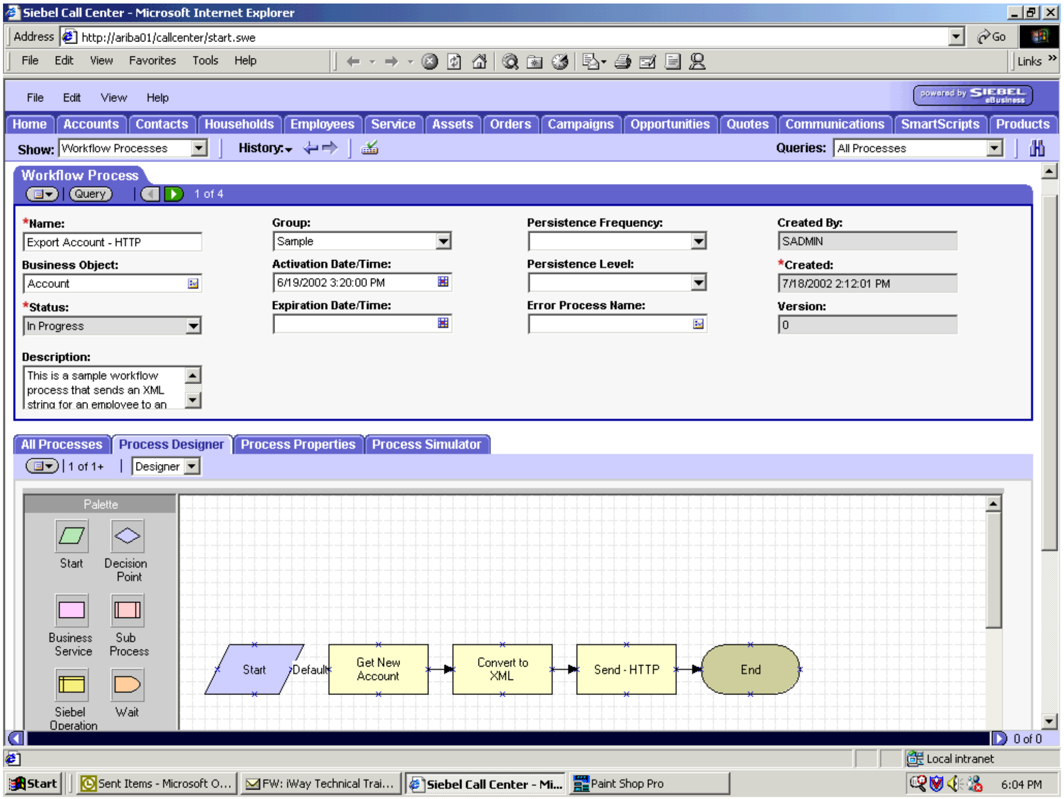
Creating a Siebel Workflow for an Event Using HTTP Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the HTTP transport.

Procedure: How to Create a Siebel Workflow for an Event Using HTTP Transport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Designer tab is active and displays an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
1 of 4
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Siebel Operation
Wait
Start
Default
Get New Account
Convert to XML
Send - HTTP
End
0 of 0
Local intranet
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
6:04 PM

Show: Workflow Processes

Queries: All Processes

*Name: Export Account - HTTP

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that sends an XML string for an employee to an

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

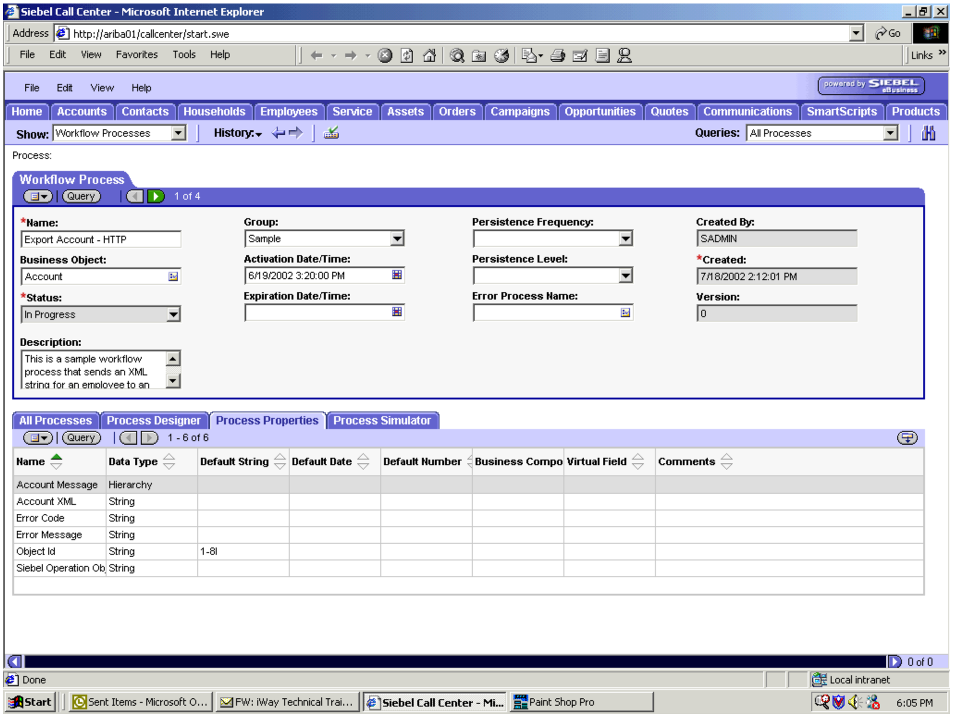
Error Process Name:

Created By: SADMIN

*Created: 7/18/2002 2:12:01 PM

Version: 0

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Go
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
1 of 4
All Processes
Process Designer
Process Properties
Process Simulator
Query
1 - 6 of 6
0 of 0
Done
Local intranet
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
6:05 PM

*Name: Export Account - HTTP	Group: Sample	Persistence Frequency:	Created By: SADMIN
Business Object: Account	Activation Date/Time: 6/19/2002 3:20:00 PM	Persistence Level:	*Created: 7/18/2002 2:12:01 PM
*Status: In Progress	Expiration Date/Time:	Error Process Name:	Version: 0
Description: This is a sample workflow process that sends an XML string for an employee to an			

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

To create a Siebel Workflow:

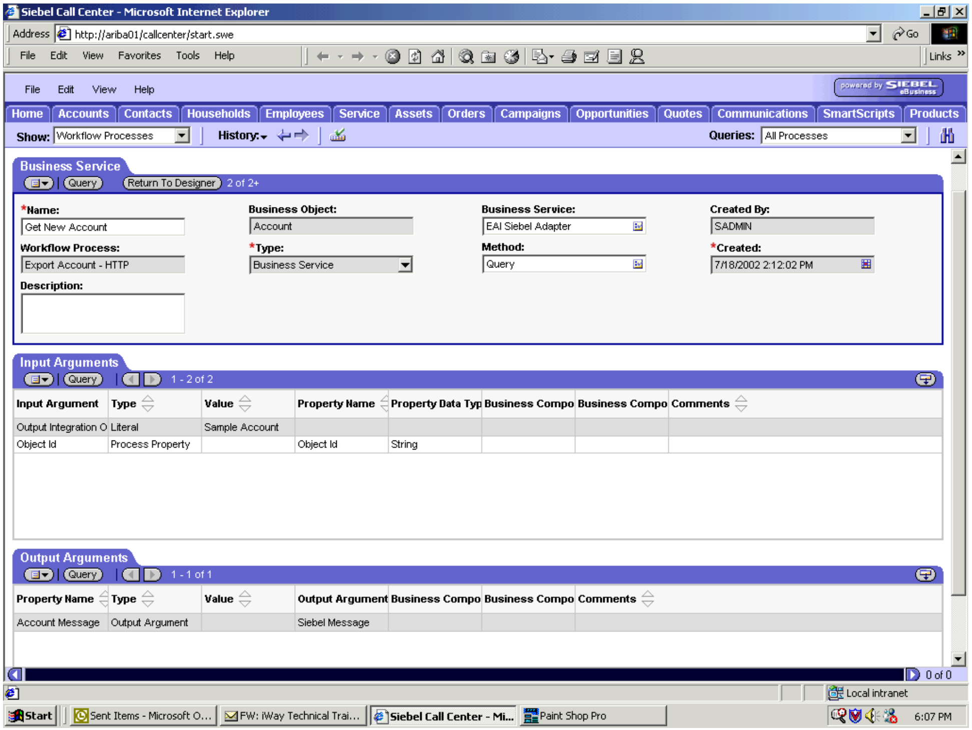
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   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. Use the Siebel Workflow Administration windows to create a Workflow.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Get New Account Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
Local intranet
6:07 PM

Show: Workflow Processes

Queries: All Processes

Business Service 2 of 2+

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account - HTTP	*Type: Business Service	Method: Query	*Created: 7/18/2002 2:12:02 PM
Description:			

Input Arguments 1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

Output Arguments 1 - 1 of 1

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

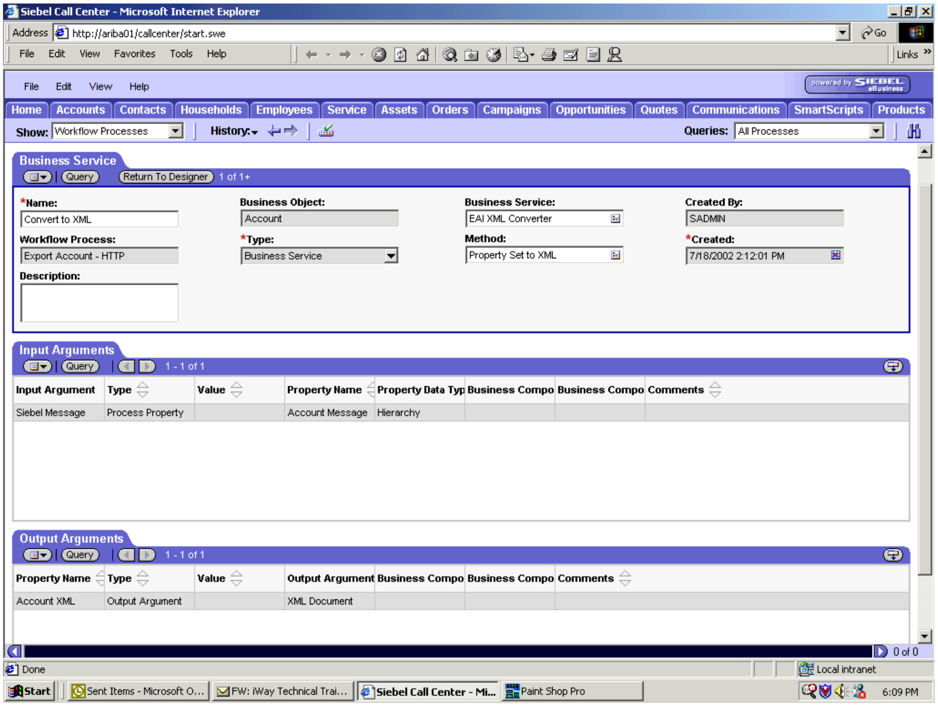
0 of 0

3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

   Using the Query method, the Business Service obtains the Account information from Siebel.

   Output from this Business Service is generated in hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Convert to XML Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.



Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Convert to XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Export Account - HTTP	*Type: Business Service	Method: Property Set to XML	*Created: 7/18/2002 2:12:01 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

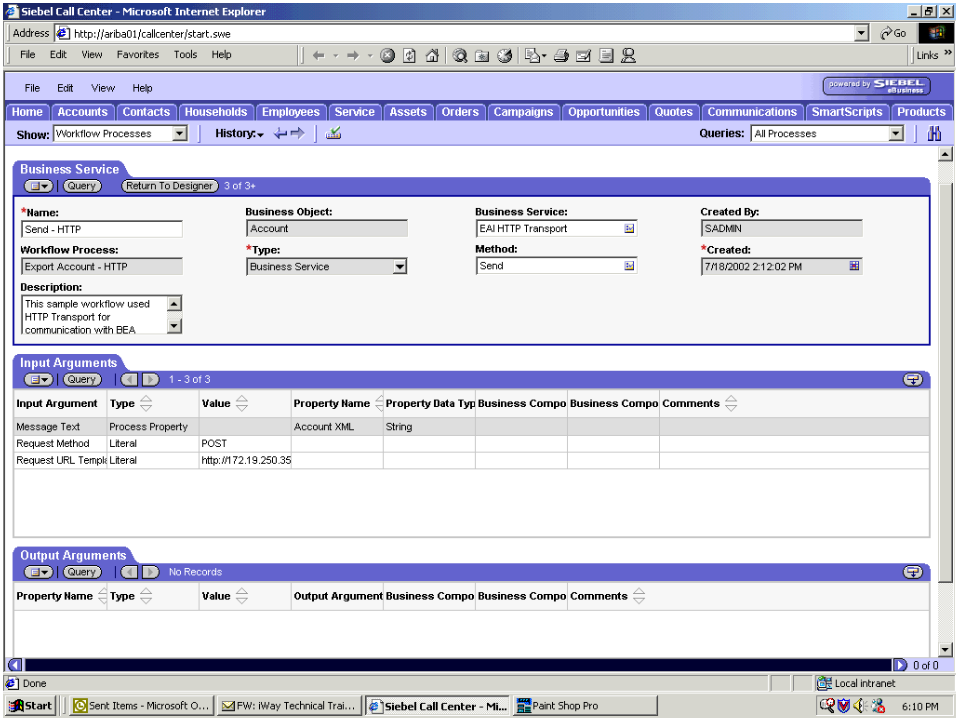
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		XML Document			

4. Define an EAI XML Converter Business Service step and call it *Convert to XML*.

   This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Send - HTTP Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.



Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Send - HTTP	Business Object: Account	Business Service: EAI HTTP Transport	Created By: SADMIN
Workflow Process: Export Account - HTTP	*Type: Business Service	Method: Send	*Created: 7/18/2002 2:12:02 PM
Description: This sample workflow used HTTP Transport for communication with BEA			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
Request Method	Literal	POST					
Request URL Templ	Literal	http://172.19.250.35					

Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

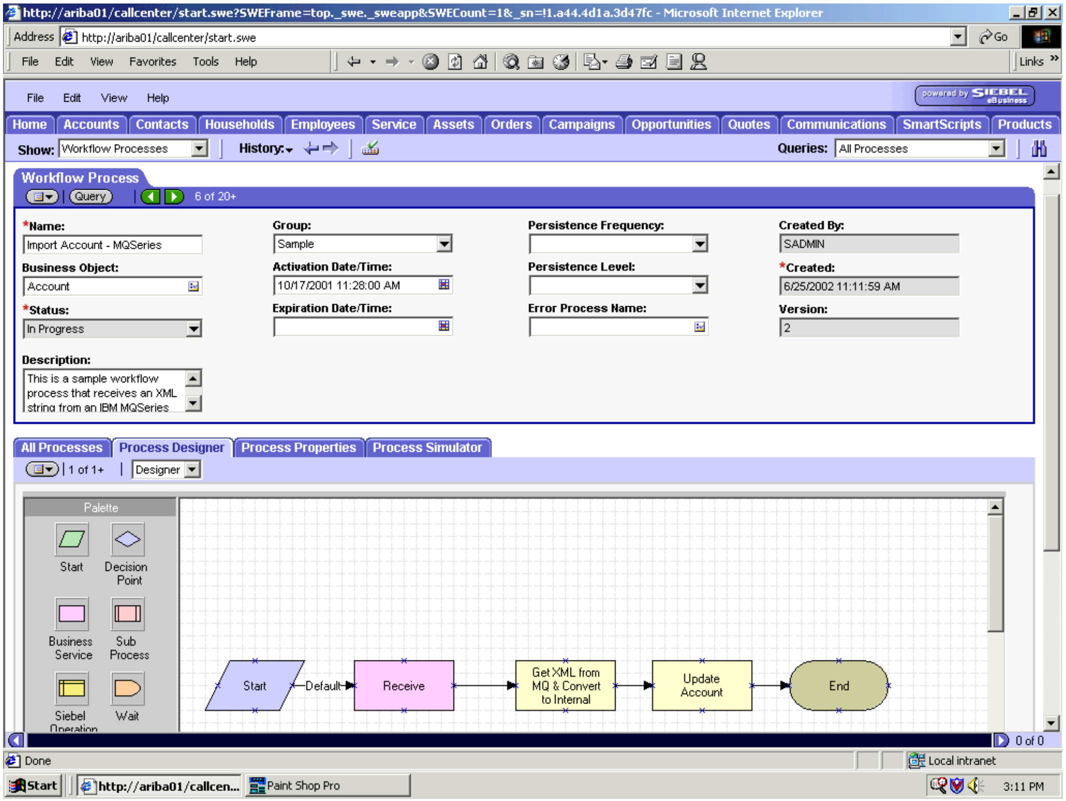
5. Define an EAI HTTP Transport Business Service step and call it *Send - HTTP*.

   This Business Service is defined 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 a Service Using MQSeries Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the MQSeries Transport.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The Process Designer tab is active and shows an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
6 of 20+
All Processes
Process Designer
Process Properties
Process Simulator
1 of 1+
Palette
Start
Decision Point
Business Service
Sub Process
Siebel Operation
Wait
Start
Default
Receive
Get XML from MQ & Convert to Internal
Update Account
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:11 PM

Show: Workflow Processes

Queries: All Processes

*Name: Import Account - MQSeries

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that receives an XML string from an IBM MQSeries

Group: Sample

Activation Date/Time: 10/17/2001 11:28:00 AM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 6/25/2002 11:11:59 AM

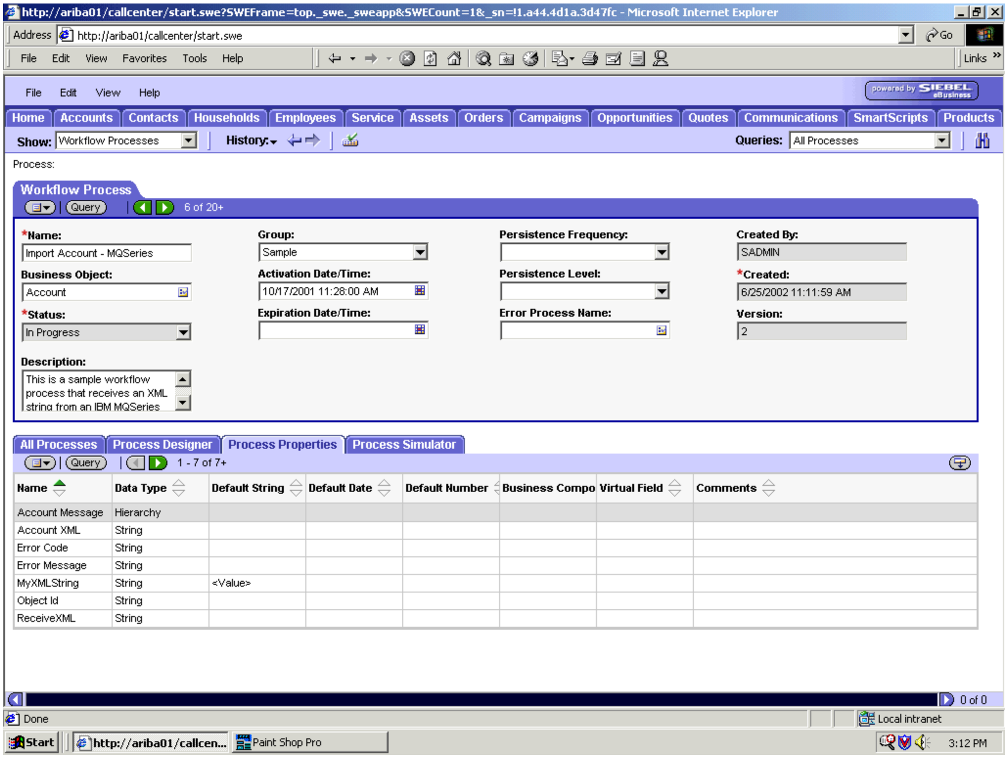
Version: 2

Designer

Procedure: How to Create a Siebel Workflow for a Service Using MQSeries Transport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
6 of 20+
All Processes Process Designer Process Properties Process Simulator
Query
1 - 7 of 7+
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:12 PM

*Name:	Import Account - MQSeries
Business Object:	Account
*Status:	In Progress
Description:	This is a sample workflow process that receives an XML string from an IBM MQSeries
Group:	Sample
Activation Date/Time:	10/17/2001 11:28:00 AM
Expiration Date/Time:	
Persistence Frequency:	
Persistence Level:	
Error Process Name:	
Created By:	SADMIN
*Created:	6/25/2002 11:11:59 AM
Version:	2

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
MyXMLString	String	<Value>					
Object Id	String						
ReceiveXML	String						

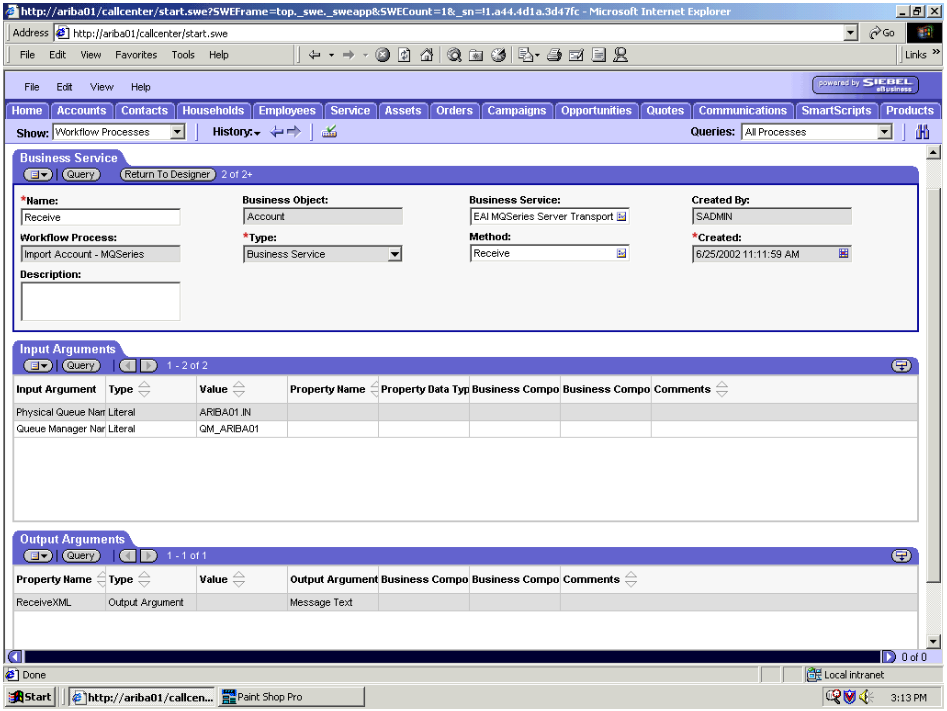
To create a Siebel Workflow:

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 converted to XML.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Receive Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:13 PM

Show: Workflow Processes

Queries: All Processes

Business Service

Query | Return To Designer | 2 of 2+

*Name: Receive	Business Object: Account	Business Service: EAI MQSeries Server Transport	Created By: SADMIN
Workflow Process: Import Account - MQSeries	*Type: Business Service	Method: Receive	*Created: 6/25/2002 11:11:59 AM
Description:			

Input Arguments

Query | 1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Physical Queue Nan	Literal	ARIBA01.IN					
Queue Manager Nar	Literal	QM_ARIBA01					

Output Arguments

Query | 1 - 1 of 1

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
ReceiveXML	Output Argument		Message Text			

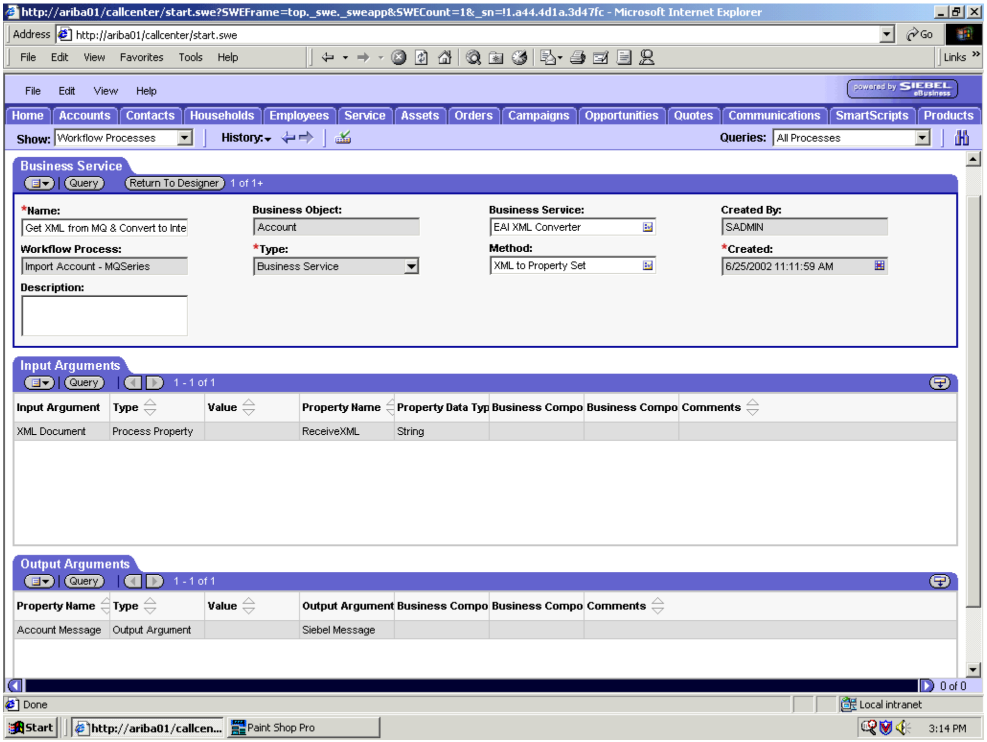
0 of 0

2. Define an EAI MQSeries Server Transport Business Service step and call it *Receive*.

   The Business Service is defined to receive the Account data from the 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.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Get XML from MQ & Convert to XML Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
1 of 1+
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:14 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Get XML from MQ & Convert to Inte

Business Object: Account

Business Service: EAI XML Converter

Created By: SADMIN

Workflow Process: Import Account - MQSeries

*Type: Business Service

Method: XML to Property Set

*Created: 6/25/2002 11:11:59 AM

Description:

Input Arguments

1 - 1 of 1

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

Output Arguments

1 - 1 of 1

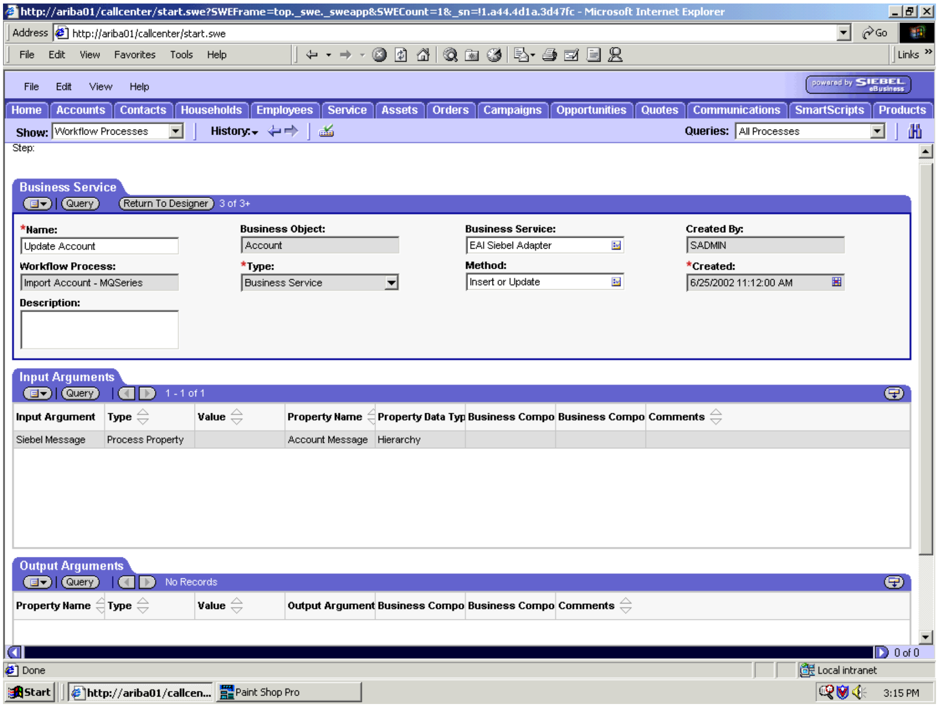
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

0 of 0

3. Define an EAI XML Converter Business Service step and call it *Get XML from MQ & Convert to XML*.

   This Business Service is defined to receive the Account data from the EAI MQSeries Server Transport Business Service in XML format and convert it to hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Update Account Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Step:
Business Service
Query
Return To Designer
3 of 3+
Input Arguments
Query
1 - 1 of 1
Output Arguments
Query
No Records
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:15 PM

*Name: Update Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Import Account - MQSeries	*Type: Business Service	Method: Insert or Update	*Created: 6/25/2002 11:12:00 AM
Description:			

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

4. Define an EAI Siebel Adapter Business Service step and call it *Update Account*.

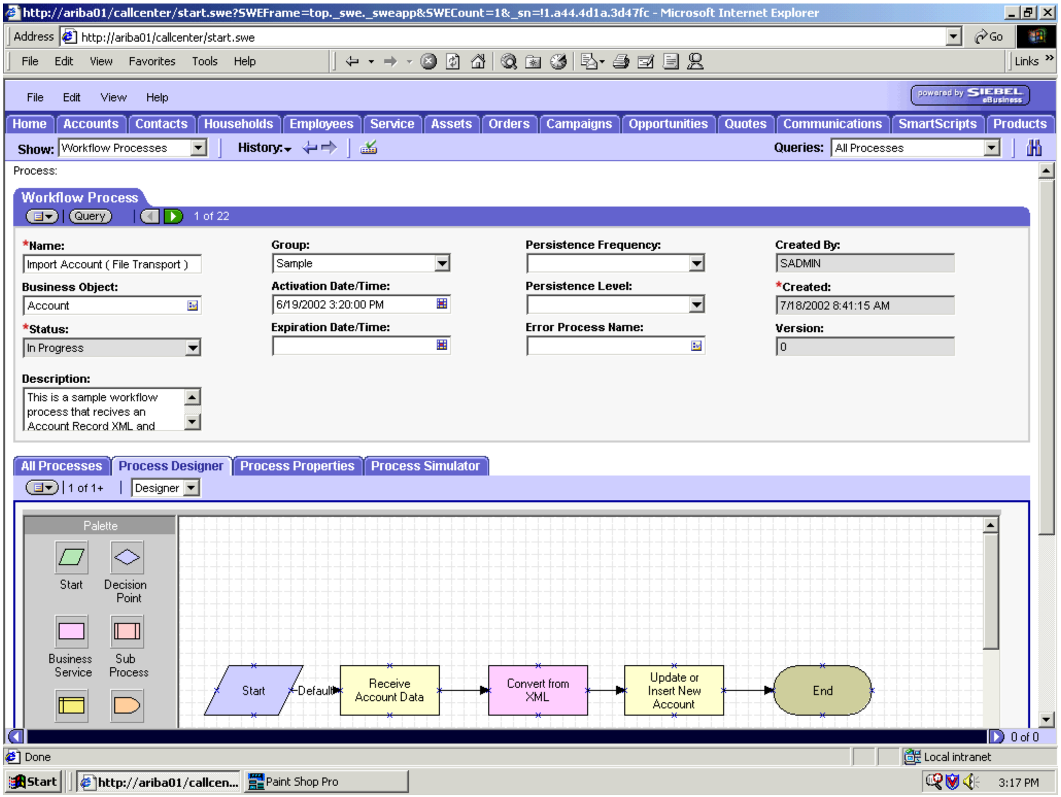
   This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

   The Business Service applies the Account information into Siebel using the Insert or Update method.

Creating a Siebel Workflow for a Service Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The workflow was designed for importing Siebel Account record information through the File transport.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Designer tab is active and displays an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
1 of 22
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Start
Default
Receive Account Data
Convert from XML
Update or Insert New Account
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:17 PM

*Name: Import Account (File Transport)

Group: Sample

Persistence Frequency:

Created By: SADMIN

Business Object: Account

Activation Date/Time: 6/19/2002 3:20:00 PM

Persistence Level:

*Created: 7/18/2002 8:41:15 AM

*Status: In Progress

Expiration Date/Time:

Error Process Name:

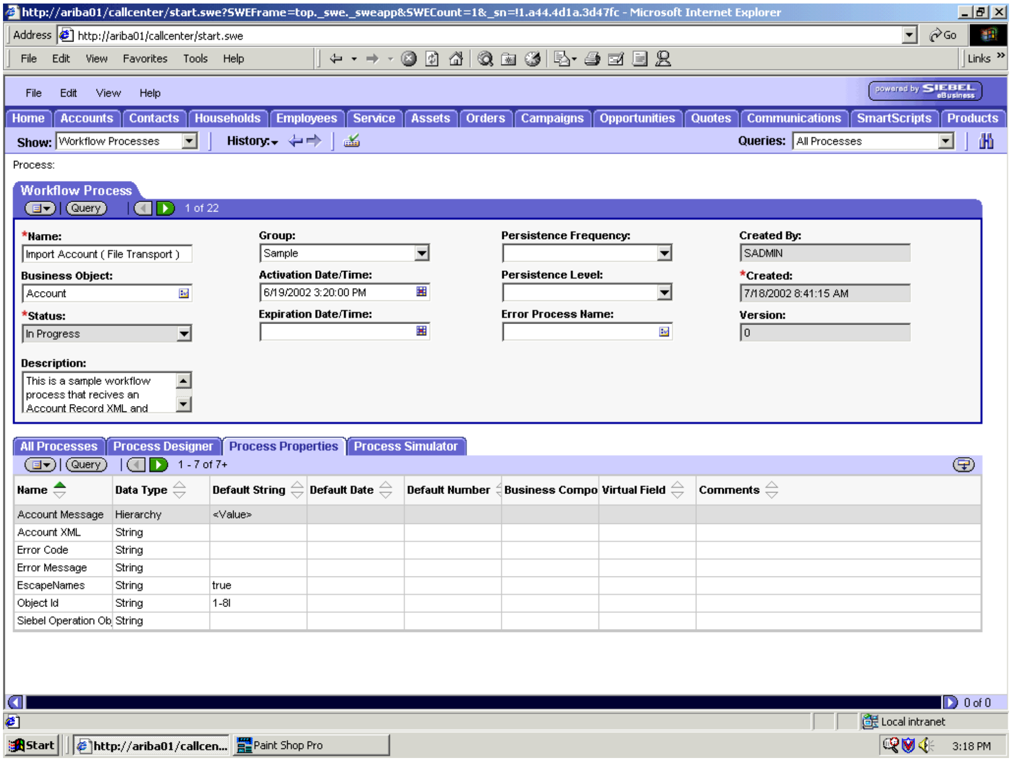
Version: 0

Description: This is a sample workflow process that recives an Account Record XML and

Procedure: How to Create a Siebel Workflow for a Service Using File Transport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
1 of 22
All Processes Process Designer Process Properties Process Simulator
Query
1 - 7 of 7+
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:18 PM

*Name: Import Account (File Transport)	Group: Sample	Persistence Frequency:	Created By: SADMIN
Business Object: Account	Activation Date/Time: 6/19/2002 3:20:00 PM	Persistence Level:	*Created: 7/18/2002 8:41:15 AM
*Status: In Progress	Expiration Date/Time:	Error Process Name:	Version: 0

Description: This is a sample workflow process that recives an Account Record XML and

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
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						

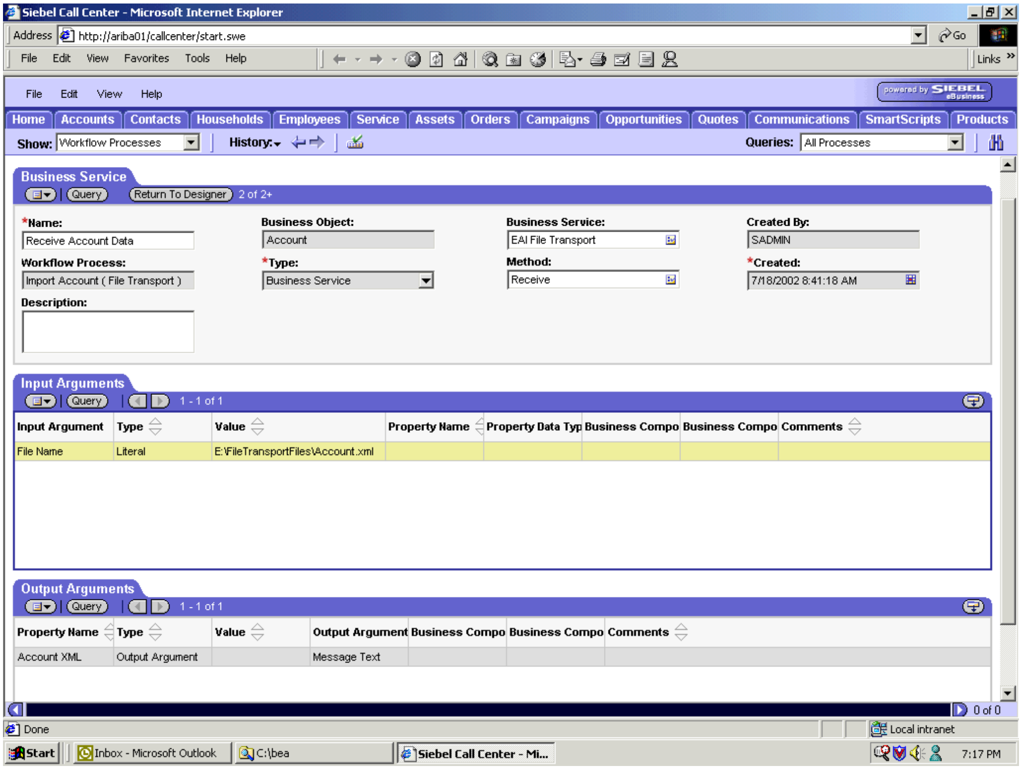
To create a Siebel Workflow:

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 converted to XML.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Receive Account Data Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
2 of 2+
Input Arguments
Query
1 - 1 of 1
Output Arguments
Query
1 - 1 of 1
0 of 0
Done
Local intranet
Start
Inbox - Microsoft Outlook
C:\bea
Siebel Call Center - Mi...
7:17 PM

*Name: Receive Account Data	Business Object: Account	Business Service: EAI File Transport	Created By: SADMIN
Workflow Process: Import Account (File Transport)	*Type: Business Service	Method: Receive	*Created: 7/18/2002 8:41:18 AM
Description:			

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
File Name	Literal	E:\FileTransportFiles\Account.xml					

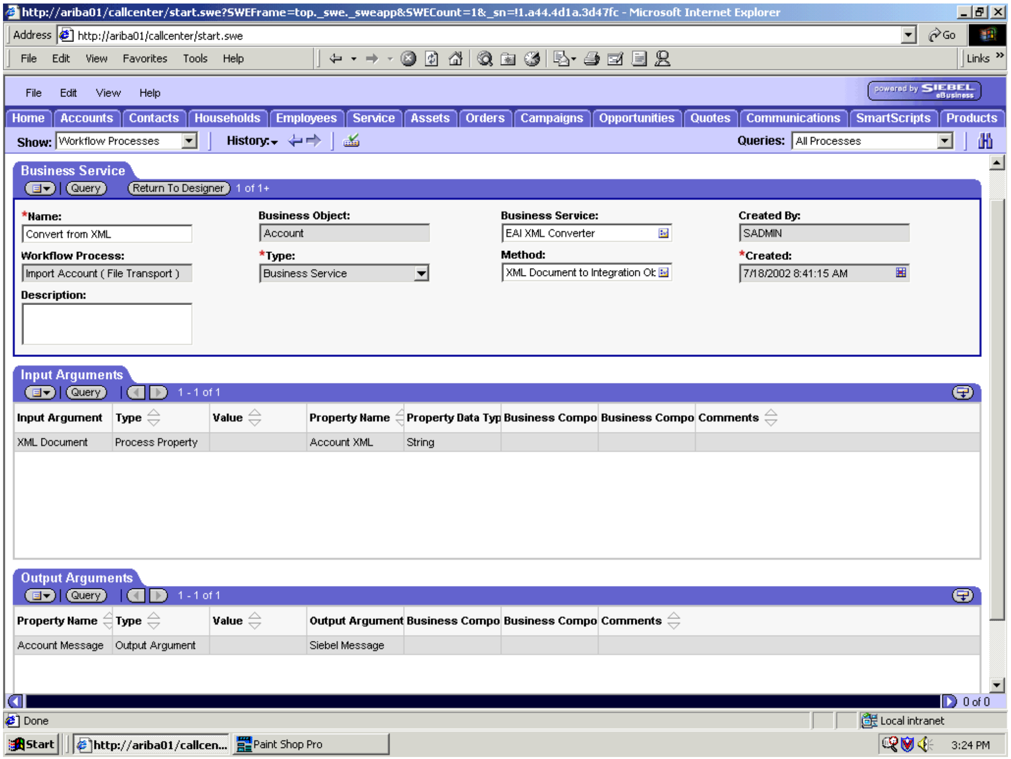
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		Message Text			

2. Define an EAI FileTransport Business Service step and call it *Receive Account Data*.

The Business Service is defined 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.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Convert from XML Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Business Service
Query
Return To Designer
1 of 1+
Input Arguments
Query
1 - 1 of 1
Output Arguments
Query
1 - 1 of 1
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:24 PM

Show: Workflow Processes

Queries: All Processes

*Name: Convert from XML

Workflow Process: Import Account (File Transport)

Description:

Business Object: Account

*Type: Business Service

Business Service: EAI XML Converter

Method: XML Document to Integration Ob

Created By: SADMIN

*Created: 7/18/2002 8:41:15 AM

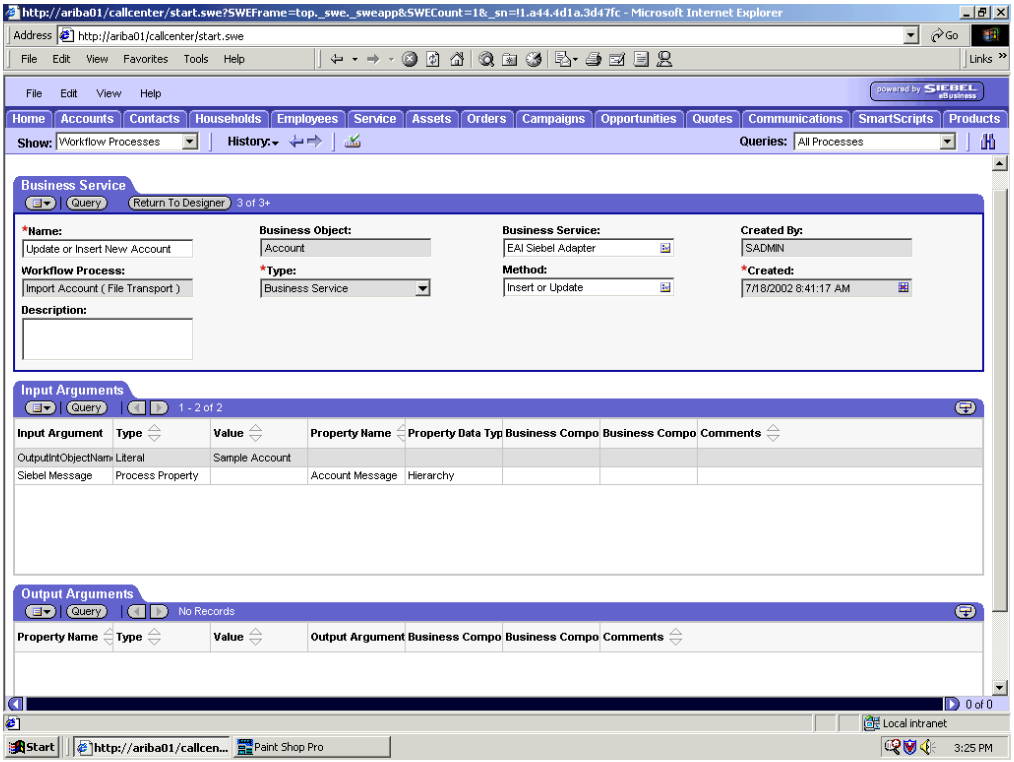
Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
XML Document	Process Property		Account XML	String			

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

3. Define an EAI XML Converter Business Service step and call it *Convert from XML*.

   This Business Service is defined to receive the Account data from the EAI File Transport Business Service in XML format and convert it to hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Update or Insert New Account Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Query
Return To Designer
3 of 3+
Query
1 - 2 of 2
Query
No Records
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:25 PM

Business Service

*Name: Update or Insert New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Import Account (File Transport)	*Type: Business Service	Method: Insert or Update	*Created: 7/18/2002 8:41:17 AM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
OutputIntObjectNam	Literal	Sample Account					
Siebel Message	Process Property		Account Message	Hierarchy			

Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

4. Define an EAI Siebel Adapter Business Service step and call it *Update or Insert New Account*.

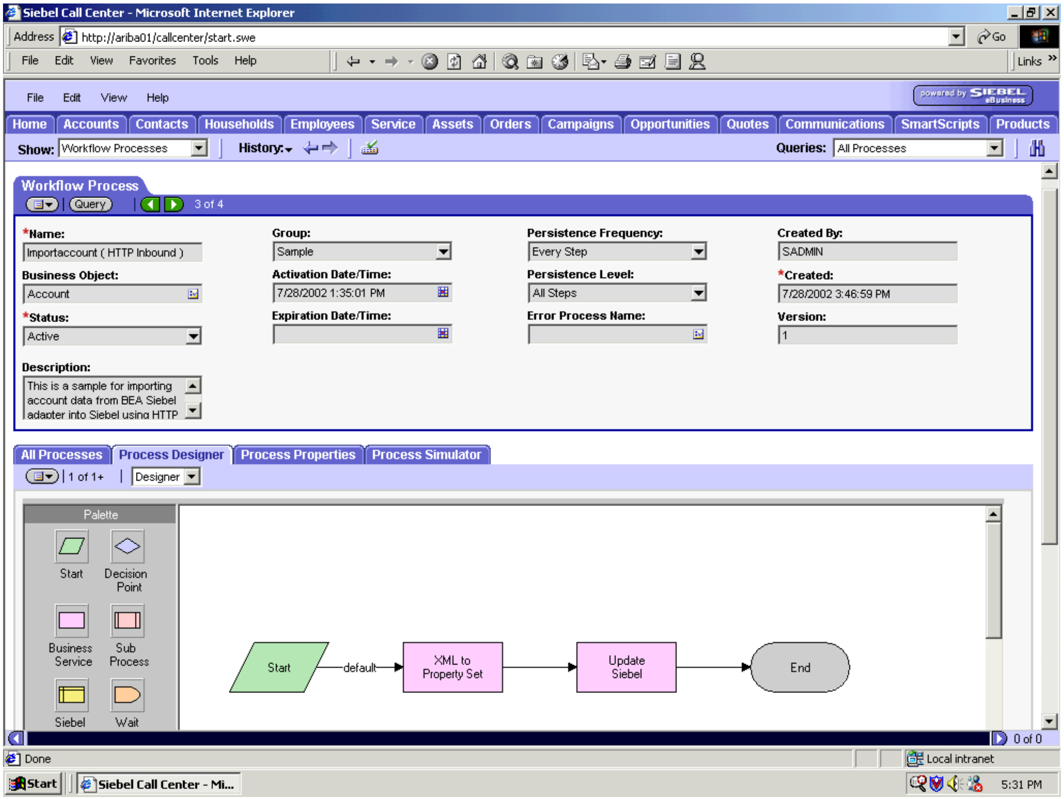
   This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

   The Business Service applies the Account information into Siebel using the Insert or Update method.

Creating a Siebel Workflow for a Service Using HTTP Transport

The following procedure is an example of a Siebel workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the HTTP transport.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Designer tab is active and displays an illustration of a Workflow process. The Workflow process can be modified using the palette to the left of the diagram.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Workflow Process
Query
3 of 4

*Name: Importaccount (HTTP Inbound)

Business Object: Account

*Status: Active

Description: This is a sample for importing account data from BEA Siebel adapter into Siebel using HTTP

Group: Sample

Activation Date/Time: 7/28/2002 1:35:01 PM

Expiration Date/Time:

Persistence Frequency: Every Step

Persistence Level: All Steps

Error Process Name:

Created By: SADMIN

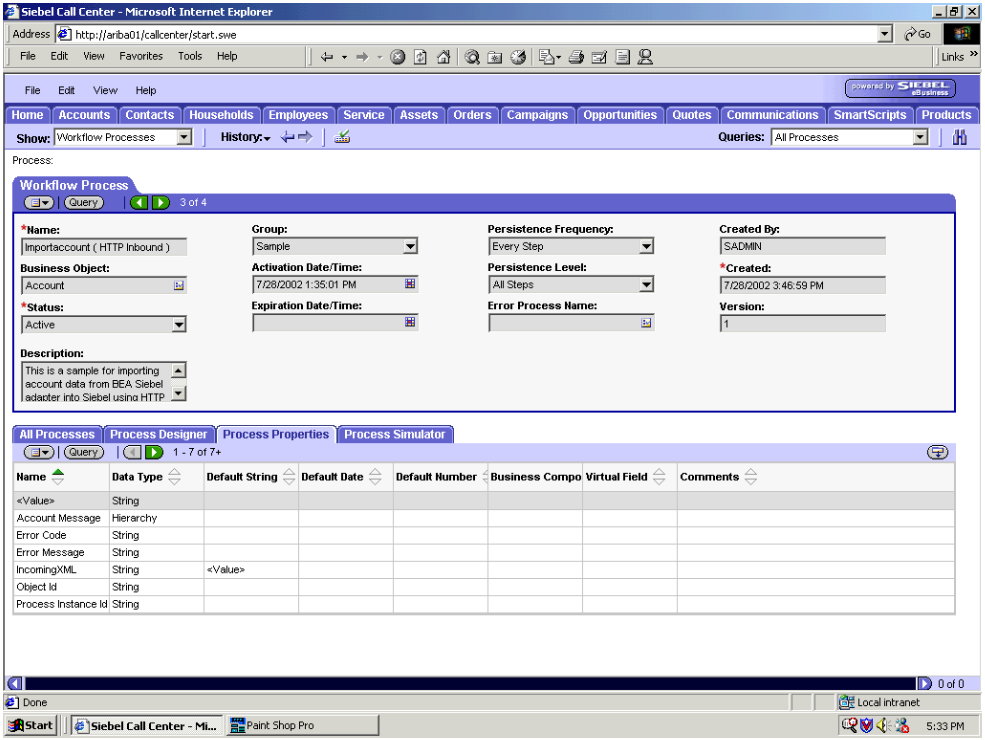
*Created: 7/28/2002 3:46:59 PM

Version: 1

Procedure: How to Create a Siebel Workflow for a Service Using HTTP Transport

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists for defining Account record information for each Workflow. The lower pane includes four tabs. The Process Properties tab is active and includes a chart of Siebel Account data properties.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
3 of 4
All Processes Process Designer Process Properties Process Simulator
Query
1 - 7 of 7+
0 of 0
Done
Local intranet
Start
Siebel Call Center - Mi...
Paint Shop Pro
5:33 PM

Show: Workflow Processes

Queries: All Processes

Process:

*Name: Importaccount (HTTP Inbound)

Business Object: Account

*Status: Active

Description: This is a sample for importing account data from BEA Siebel adapter into Siebel using HTTP

Group: Sample

Activation Date/Time: 7/28/2002 1:35:01 PM

Expiration Date/Time:

Persistence Frequency: Every Step

Persistence Level: All Steps

Error Process Name:

Created By: SADMIN

*Created: 7/28/2002 3:46:59 PM

Version: 1

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

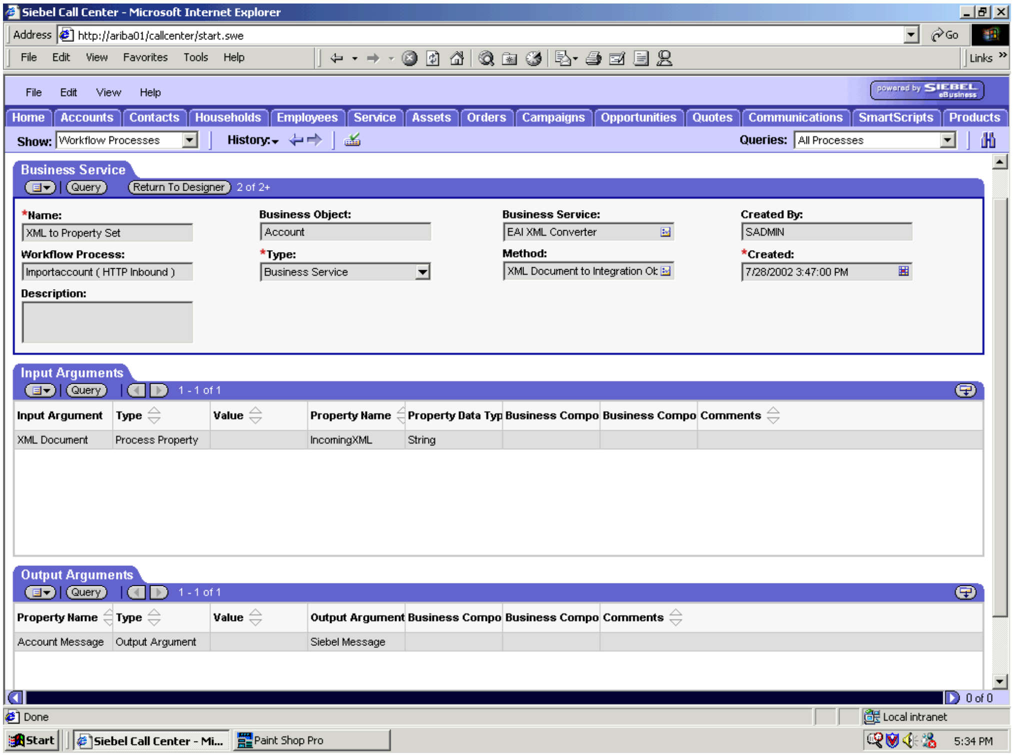
To create a Siebel Workflow:

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 converted to XML.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the XML to Property Set Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
2 of 2+
1 - 1 of 1
0 of 0
Done
Local intranet
Start
Siebel Call Center - Mi...
Paint Shop Pro
5:34 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*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

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

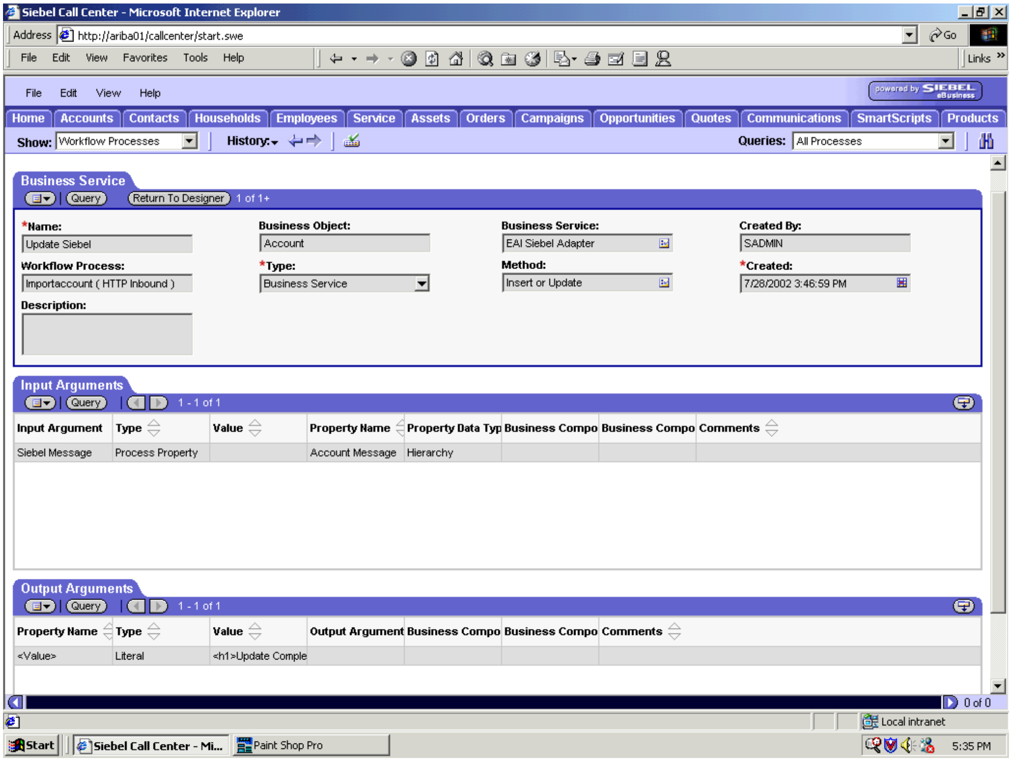
Output Arguments

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

2. Define an EAI XML Converter Business Service step and call it *XML to Property Set*.

   The Business Service is defined to receive the Account data from the EAI HTTP Transport Business Service in XML format and convert it to hierarchical format.

The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the Update Siebel Business Service step. The middle pane shows the Input Arguments tab, which includes a chart of Input Arguments. The lower pane shows the Output Arguments tab, which includes a chart of Output Argument properties.



Show: Workflow Processes

Queries: All Processes

Business Service

*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 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 1 - 1 of 1

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
<Value>	Literal	<h1>Update Comple				

0 of 0

3. Define an EAI Siebel Adapter Business Service step and call it *Update Siebel*.

   The Business Service is defined 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.

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: http://www.informationbuilders.com/bookstore/derf.html

Name:____

Company:____

Address:____

Telephone:____________________Date:____________________

E-mail:____

Comments:

Reader Comments

Information Builders, Two Penn Plaza, New York, NY 10121-2898 (212) 736-4433