



## BEA WebLogic Adapter for Siebel User's Guide

DN3501345.0406

April 20, 2006

EDA, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPPack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks, and iWay and iWay Software are trademarks of Information Builders, Inc.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2006 BEA Systems, Inc. All Rights Reserved.

---

---

## Preface

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

## How This Manual Is Organized

---

This manual includes the following chapters:

Chapter/Appendix		Contents
<b>1</b>	Introducing the BEA WebLogic Adapter for Siebel	Explains how the BEA WebLogic Adapter for Siebel integrates with Siebel. This section also describes the Siebel architecture.
<b>2</b>	Creating XML Schemas or Integration Business Services	Describes how to create XML schemas and business services for Siebel objects using the Integration Business Services Engine (iBSE).
<b>3</b>	Listening for Siebel Events	Describes how to use the BEA WebLogic Adapter for Siebel to listen for events in a Siebel system.
<b>4</b>	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
<b>5</b>	Management and Monitoring	Describes how to configure and use monitoring tools provided by iBSE and JCA to gauge the performance of your run-time environment.
<b>6</b>	Troubleshooting and Error Messages	Explains the limitations and workarounds when connecting to Siebel.
<b>A</b>	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services	Describes how to use Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and Integration Business Services for Siebel.
<b>B</b>	Using Application Explorer in BEA WebLogic Workshop for Event Handling	Describes how to use Java Swing Application Explorer running in BEA WebLogic Workshop to create events for Siebel.

<b>C</b>	Usage Considerations and Sample Files	Provides sample schemas for Siebel Business Components and Siebel Business Services.
<b>D</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 BEA WebLogic 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

---

Delete the items that do not apply to your manual and/or add special conventions that are unique to your manual.

The following conventions apply throughout this manual:

Convention	Description
<b>THIS TYPEFACE</b> or <i>this typeface</i>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term.
<b>this typeface</b>	Highlights a file name or command in a text paragraph that must be lowercase.
<i>this typeface</i>	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.

## Contact Us!

---

Your feedback on the BEA WebLogic Adapter for Siebel documentation is important to us.

Send us e-mail at [docsupport@bea.com](mailto:docsupport@bea.com) if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for Siebel documentation.

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

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

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

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

## 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.

<b>Platform</b>	
<b>Operating System</b>	
<b>OS Version</b>	
<b>Product List</b>	
<b>Adapters</b>	
<b>Adapter Deployment</b>	For example, JCA, Business Services Engine, iWay Adapter Manager
<b>Container Version</b>	

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

Component	Version
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?	
Any <b>change in the application environment:</b> software configuration, EIS/ database configuration, application, and so forth?	

Request/Question	Error/Problem Details or Information
Under what circumstance does the problem <i>not</i> occur?	
Describe the <b>steps</b> to reproduce the problem.	
Describe the <b>problem</b> .	
Specify the <b>error</b> 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

---

---

# Contents

<b>1. Introducing the BEA WebLogic Adapter for Siebel</b> .....	<b>1-1</b>
Features of the BEA WebLogic Adapter for Siebel .....	1-2
The Siebel Application Model .....	1-2
Integrating With Siebel .....	1-3
Siebel EAI Architecture .....	1-4
Using iWay Application Explorer With the BEA WebLogic Adapter for Siebel .....	1-5
Key Features of Application Explorer .....	1-5
Installing and Configuring the Servlet Application Explorer .....	1-5
Deployment Information for the BEA WebLogic Adapter for Siebel .....	1-6
Deployment Information Roadmap .....	1-6
The Integration Business Services Engine .....	1-7
The Enterprise Connector for J2EE Connector Architecture (JCA) .....	1-7
<b>2. Creating XML Schemas or Integration Business Services</b> .....	<b>2-1</b>
Processing Overview .....	2-2
Starting Servlet Application Explorer .....	2-2
Managing a Siebel Connection .....	2-3
Viewing Metadata .....	2-9
Creating a Schema for a Siebel Business Component or Siebel Business Service .....	2-10
Creating an XML Schema for a Siebel Business Component or Siebel Business Service ...	2-11
Creating an XML Schema for a Siebel Integration Object .....	2-14
Creating a Siebel XDR Schema .....	2-15
Creating a Schema for a Siebel Integration Object Using Application Explorer .....	2-18
Schema Location .....	2-23
Returning Fields in a Specified Order .....	2-24
Creating a Web Service .....	2-24
Testing a Web Service for a Business Component or Integration Object .....	2-27
Testing a Web Service for a Business Service .....	2-29
Identity Propagation .....	2-30
<b>3. Listening for Siebel Events</b> .....	<b>3-1</b>
Understanding Event Functionality .....	3-2
Listening for a Siebel Integration Object Event .....	3-2
Creating an Event Port .....	3-2
Creating an Event Port From the Service Adapters Tab .....	3-3
Creating an Event Port From the Event Adapters Tab .....	3-4
<i>Event Adapters</i> Event AdaptersEditing or Deleting an Event Port .....	3-15
Creating a Channel .....	3-16

<b>4. Using Web Services Policy-Based Security</b>	<b>4-1</b>
Integration Business Services Policy-Based Security	4-2
Configuring Integration Business Services Policy-Based Security	4-3
<b>5. Management and Monitoring</b>	<b>5-1</b>
Managing and Monitoring Services and Events Using iBSE	5-2
Managing and Monitoring Services and Events Using the JCA Test Tool	5-16
Setting Engine Log Levels	5-21
Migrating Repositories	5-22
File Repositories	5-22
iBSE Repositories	5-22
JCA Repositories	5-27
Migrating Event Handling Configurations	5-27
Exporting or Importing Targets	5-31
Retrieving or Updating Web Service Method Connection Information	5-35
Starting or Stopping a Channel Programmatically	5-39
<b>6. Troubleshooting and Error Messages</b>	<b>6-1</b>
Troubleshooting	6-2
Application Explorer	6-2
Siebel	6-4
JCA	6-5
iBSE Error Messages	6-6
General Error Handling	6-6
Adapter-Specific Error Handling	6-7
Updating a Siebel Field	6-9
<b>A. Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services</b>	<b>A-1</b>
Starting Application Explorer in BEA WebLogic Workshop	A-2
Creating a New Configuration for iBSE or JCA	A-3
Connecting to Siebel	A-5
Creating and Connecting to a Target	A-5
Managing a Target	A-7
Viewing Metadata	A-9
Creating an XML Schema for a Siebel Business Component or Siebel Business Service	A-10
Creating an XML Schema for a Siebel Business Object or Business Service	A-10
Creating a Schema for a Siebel Integration Object	A-13
Creating a Schema for a Siebel Integration Object Using Application Explorer	A-17
Creating an Integration Business Service	A-22
Testing a Web Service for a Business Object or Integration Object	A-24
Testing a Web Service for a Business Service	A-25
Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows	A-26
Identity Propagation	A-28

Adding a Control for an iWay Resource in BEA WebLogic Workshop .....	A-28
Adding a Web Service Control to a BEA WebLogic Workshop Application .....	A-28
Adding an Extensible CCI Control to a BEA WebLogic Workshop Application .....	A-29
Overview .....	A-29
Using the Extensible CCI Control .....	A-35
<b>B. Using Application Explorer in BEA WebLogic Workshop for Event Handling ...</b>	<b>B-1</b>
Starting Application Explorer in BEA WebLogic Workshop .....	B-2
Understanding Event Functionality .....	B-3
Listening for a Siebel Integration Object Event .....	B-3
Creating an Event Port .....	B-4
Creating an Event Port For a Siebel Integration Object .....	B-4
Creating an Event Port From the Event Adapters Node .....	B-5
Modifying an Event Port .....	B-15
Creating a Channel .....	B-15
Modifying a Channel .....	B-21
Deploying iWay Components in a Clustered BEA WebLogic Environment .....	B-22
<b>C. Usage Considerations and Sample Files .....</b>	<b>C-2</b>
Usage Considerations .....	C-3
Sample Files .....	C-3
Account Business Component .....	C-3
<b>D. Siebel Workflows .....</b>	<b>D-3</b>
Overview .....	D-4
Siebel Workflows .....	D-4
Using a Policy to Invoke a Siebel EAI Workflow .....	D-4
Siebel Workflow - Outbound .....	D-5
Siebel Workflow - Inbound .....	D-6
Creating a Siebel Workflow .....	D-6
Creating a Siebel Workflow for an Event Using MQSeries Transport .....	D-6
Creating a Siebel Workflow for an Event Using File Transport .....	D-11
Creating a Siebel Workflow for an Event Using HTTP Transport .....	D-16
Creating a Siebel Workflow for a Service Using MQSeries Transport .....	D-21
Creating a Siebel Workflow for a Service Using File Transport .....	D-27
Creating a Siebel Workflow for a Service Using HTTP Transport .....	D-32



---

---

## CHAPTER 1

# Introducing the BEA WebLogic Adapter for Siebel

### Topics:

- Features of the BEA WebLogic Adapter for Siebel
- The Siebel Application Model
- Integrating With Siebel
- Siebel EAI Architecture
- Using iWay Application Explorer With the BEA WebLogic Adapter for Siebel
- Deployment Information for the BEA WebLogic Adapter for Siebel

This section explains how the BEA WebLogic Adapter for Siebel integrates with Siebel. It describes the key features and also describes the Siebel architecture.

## Features of the BEA WebLogic Adapter for Siebel

---

The BEA WebLogic 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 BEA WebLogic Adapter for Siebel:

- Supports synchronous and asynchronous, bidirectional message interactions for Siebel Business Services, Business Components, and Integration Objects.
- Includes the iWay Application Explorer (iAE), 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 BEA WebLogic 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 BEA WebLogic 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 BEA WebLogic 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.

<b>Siebel Objects</b>	<b>API or Transport</b>	<b>Process</b>
<b>Business Services</b>	Java Data Bean (Siebel Version 6.3.x - 7.7) Com Data Interface (Siebel Version 6.0.1 - 6.2.x)	Service
<b>Business Components</b>	Java Data Bean (Siebel Version 6.3.x - 7.7) Com Data Interface (Siebel Version 6.0.1 - 6.2.x)	Service
<b>Integration Objects</b>	File	Event, Service
	HTTP	Event, Service
	MQSeries	Event, Service
	MQ Read	Service

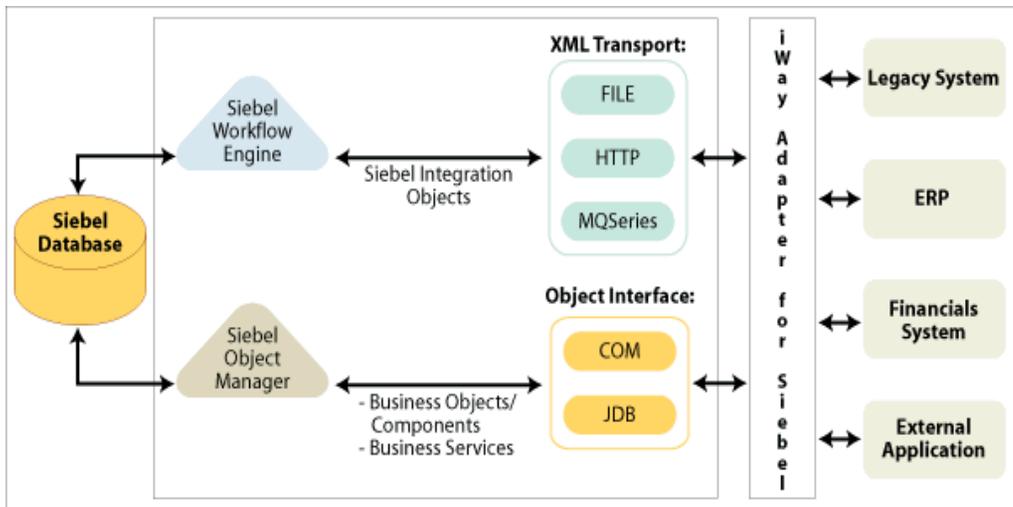
## Siebel EAI Architecture

Siebel provides for integration with other applications and systems using its Siebel EAI framework and its Business Integration Manager facility. The BEA WebLogic 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 BEA WebLogic 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 BEA WebLogic 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 BEA WebLogic 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 BEA WebLogic Adapter for Siebel use either XML schemas or Web services to pass data between the external application and the adapter.

The two versions of Application Explorer that are supported for Siebel are Servlet Application Explorer, a Java Web application running within a servlet container that is accessible through a Web browser, and Application Explorer running in BEA WebLogic Workshop. 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, the iAE can generate an XML schema or Web service to define the object for use in conjunction with the BEA WebLogic Adapter for Siebel.

External applications accessing Siebel via the BEA WebLogic 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 or Integration Business Services*, *Appendix A, Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services* and *Appendix B, Using Application Explorer in BEA WebLogic Workshop for Event Handling*.

### 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. If you are using Application Explorer in BEA WebLogic Workshop, you must also have the BEA WebLogic Workshop installed.

**Note:** To use Application Explorer within the BEA WebLogic Workshop, iBSE must be deployed to the BEA WebLogic server.

In addition, the Siebel Enterprise Information System (EIS) must be installed, configured, and available for client access. The iAE 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 *BEA WebLogic ERP Adapter Installation and Configuration* documentation.

## Deployment Information for the BEA WebLogic Adapter for Siebel

---

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

- Integration 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 Integration 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 BEA WebLogic Adapter for Siebel. A description of the Integration 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 and Appendixes A and B of this guide <i>BEA WebLogic ERP Adapter Installation and Configuration</i>
Integration Business Services Engine (iBSE)	<i>BEA WebLogic ERP Adapter Installation and Configuration</i>
Enterprise Connector for J2EE Connector Architecture (JCA)	<i>BEA WebLogic ERP Adapter Installation and Configuration</i>

## **The Integration Business Services Engine**

The Integration 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 BEA WebLogic 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.



---

---

## CHAPTER 2

# Creating XML Schemas or Integration 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.

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces. For information on running Application Explorer in WebLogic Workshop, see *Appendix A, Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services*.

## Processing Overview

---

The BEA WebLogic 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 How to Start BEA WebLogic Server on Windows

To start BEA WebLogic Server on Windows:

1. Click the Start menu.
2. Select *Programs, BEA WebLogic Platform 8.1, User Projects, your domain for iWay*, and then, click *Start Server*.

### Procedure: How to How to Start BEA WebLogic Server on UNIX

To start BEA WebLogic Server on UNIX or from a command line

1. Enter the following at the prompt:

```
BEA_HOME/user_projects/domains/DOMAIN_NAME/startWebLogic.cmd
```

where:

```
BEA_HOME
```

Is the directory where BEA WebLogic is installed.

```
DOMAIN_NAME
```

Is the domain you are using for iWay.

## 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 BEA. The port for the default domain is 7001.

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 2-7.

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 2-7.

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 2-7.

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 2-8.

## 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.



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.

The dialog box is titled 'Add a new SIEBEL target'. It contains the following text: 'Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.' Below this text are three input fields: 'Target Name:' with a yellow highlighted text box, 'Description:' with an empty text box, and 'Target Type:' with a dropdown menu showing 'Java Data Bean Connection'. At the bottom of the dialog are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

- 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, 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:

- 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 (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.

**Set connection info**

Logon Advanced

Gateway Server:

Enterprise Name:

Siebel Server:

User:

Password:

Help < Back Finish Cancel

- 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*.

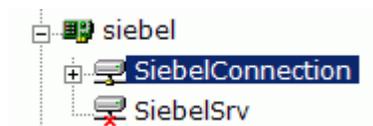
In the left pane, the target you create appears under the Siebel node.

### 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:

Description:

Target Type:

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 2-14.

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 BEA WebLogic Adapter for Siebel for BEA WebLogic for use either in a JCA (Enterprise Connector for J2EE Connector Architecture) environment or a Web services environment.

If you plan to deploy the BEA WebLogic Adapter for Siebel in a Web services environment, see also *Creating a Web Service* on page 2-24.

## **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 2-14.

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.



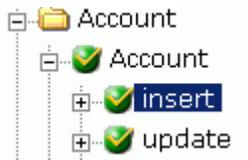
### Procedure: How to Create an XML Schema for a Siebel Business Component or Siebel Business Service

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.



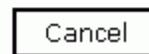
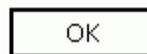
4. In the right pane, move the cursor over *Operations* and select *Generate Schema*.

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 2-23.

## Creating an XML Schema for a Siebel Integration Object

---

The BEA WebLogic 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 D, *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 C, *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.

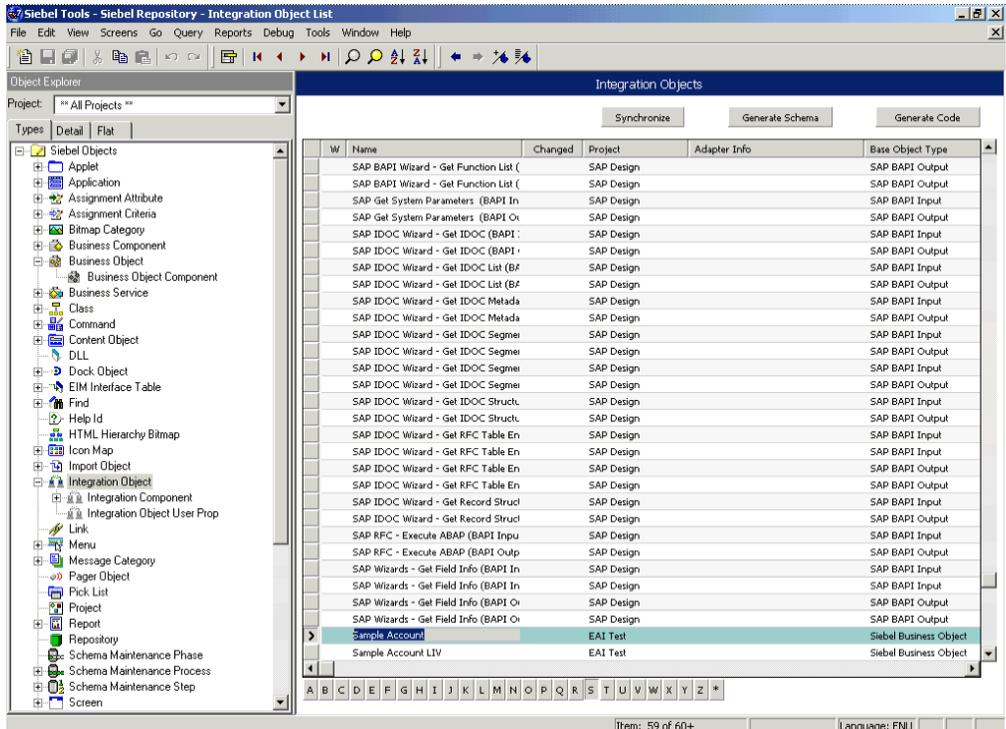


To generate a Siebel XDR schema:

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.

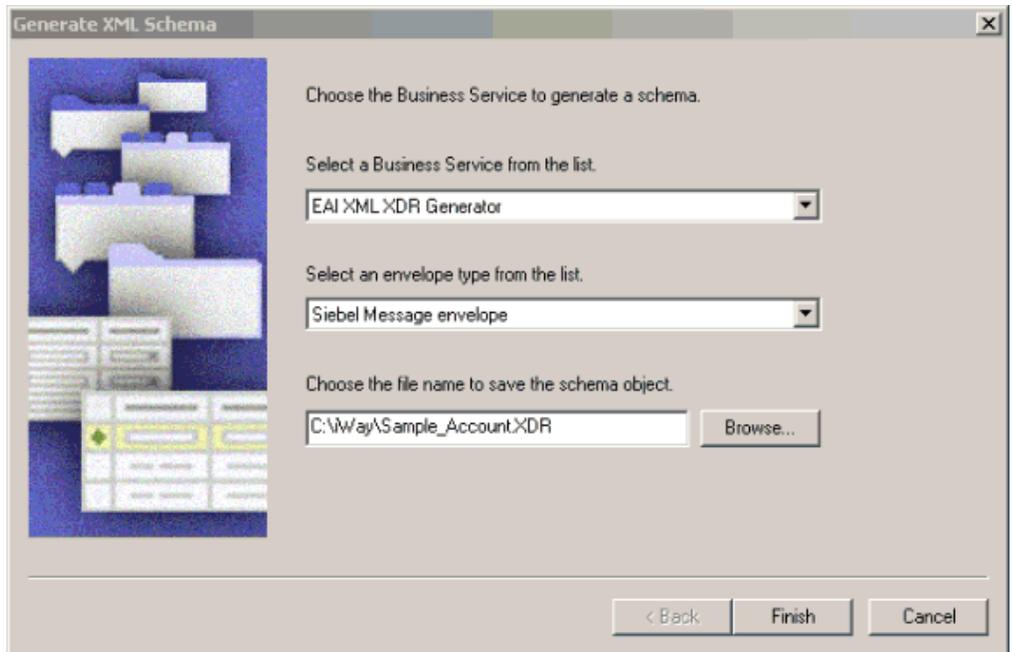
## Creating an XML Schema for a Siebel Integration Object

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.



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.



- 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 2-18.

## 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.

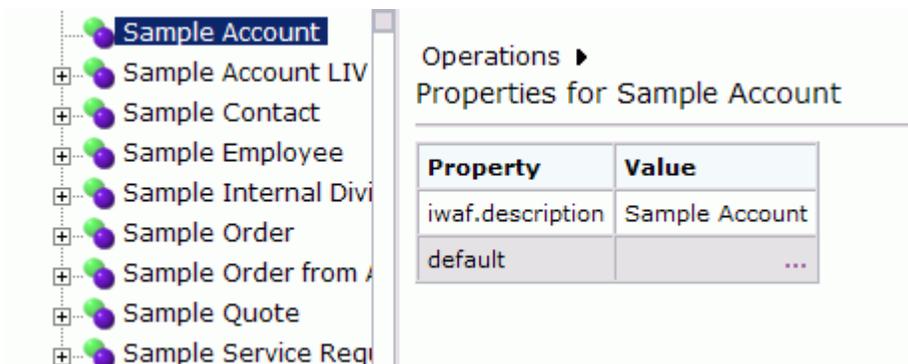
**Note:** The XDR file must be on the same computer as Application Explorer or be available through a mapped connection to another drive or machine.

### 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.



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 :

- 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>&SWEExtCmd=<Execute>&UserName=<UserName>&Password=<Password>](http://my_web_server/eai_enu/start.swe?SWEExtSource=<SourceName>&SWEExtCmd=<Execute>&UserName=<UserName>&Password=<Password>)

The following table lists parameters you supply and provides a definition for each one.

Parameter	Definition
SWE URL	The base SWE URL. For example, <a href="http://web_server/eai_enu/start.swe">http://web_server/eai_enu/start.swe</a> where: <a href="#">web_server</a> Is the name of the Web server hosting Siebel SWE.
SWE External Command	The section within the eai.cfg file to execute, which is the [HTTP Services] section.
SWE External Command	Use Execute.
User Name	The user ID logon to execute.
Password	The logon password to execute.

The following table lists parameters you supply for **MQ Series** and provides a definition for each one.

Parameter	Definition
Queue Manager Name	The name of the queue manager to which the server must connect.
MQ server host for MQClient operation	The host on which the MQ Server is located (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	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 <a href="#">SYSTEM.DEF.SVRCONN</a>

Parameter	Definition
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

Help      OK      Cancel

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 2-23.

You are now ready to configure ports and channels or create an Integration 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 Description. It includes the Disposition Protocol drop-down list from which you select the protocol type, a Help button, and three action buttons.

The image shows a graphical user interface for creating an event port. It features three input fields: 'Event Port Name' (highlighted in yellow), 'Event Port Description', and 'Disposition Protocol' (set to 'FILE'). Below the fields are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

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

## Schema Location

Application Explorer stores the schemas it creates in subdirectories under the iWay home directory of the machine where it is installed. The exact location of the schemas differs depending on whether you deploy Application Explorer with an **iBSE** or a **JCA** configuration.

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

`C:\Program Files\iway55\bea\ibse\wsdl\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 the 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 Integration 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 *BEA WebLogic ERP Adapter Installation and Configuration*.

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 2-14.

**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 *BEA WebLogic ERP Adapter Installation and Configuration*.

### Procedure: How to Generate a Web Service

To generate a Web service:

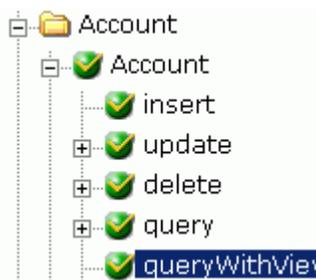
1. If you have not already connected, connect to your Siebel system.
2. Expand the Siebel node.

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 2-14.

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 Integration 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.

The following image shows the Create Web Service pane that opens on the right where you provide information to create a Business Service.

**Create Web Service for queryWithView**

Service Name:

Description:

License:

production  
test

- a. In the Service Name field, type a name to identify the Web service (under the Service node in the left pane of the Integration 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 Integration Business Services tab, and the new business service appears in the left pane.

## Testing a Web Service for a Business Component or Integration Object

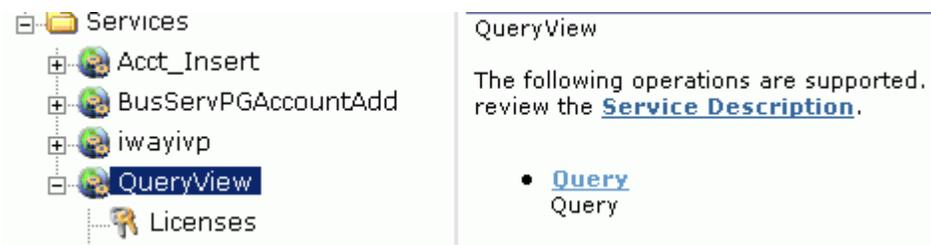
After you create a Web service for the Siebel Business Component, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

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

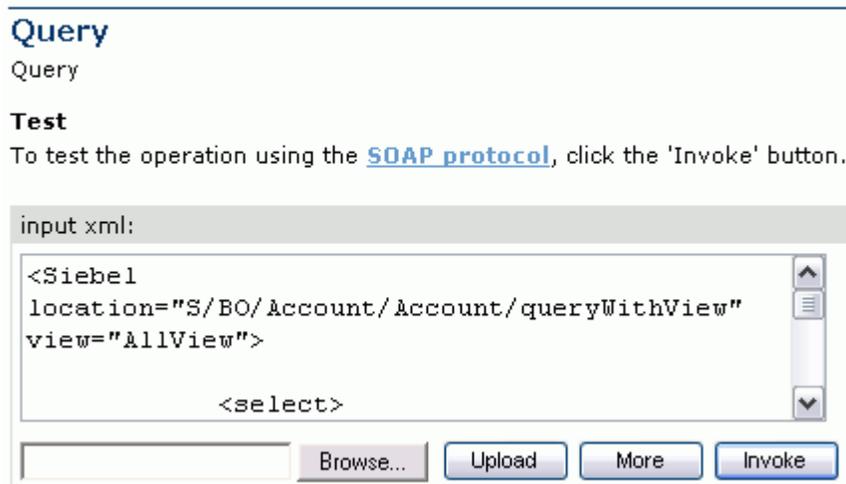
The following image shows a window with a list of services in the left pane and information about the selected service in the right pane.



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.



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:QueryWithView"
    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 Integration Business Services tab of Application Explorer, click the tab to access business services.
2. If it is not expanded, expand the *Integration 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.

Parameter	Value
Account:	<input type="text"/>
Site:	<input type="text"/>

Invoke

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 BEA WebLogic 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. For information on running Application Explorer in WebLogic Workshop, see Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services*.

## 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.

- 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 3-2.

- 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 3-16.

## 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. Information Builders

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 or Integration 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.

## Creating an Event Port From the Service Adapters Tab

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

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

To create an event port from the Service Adapters tab:

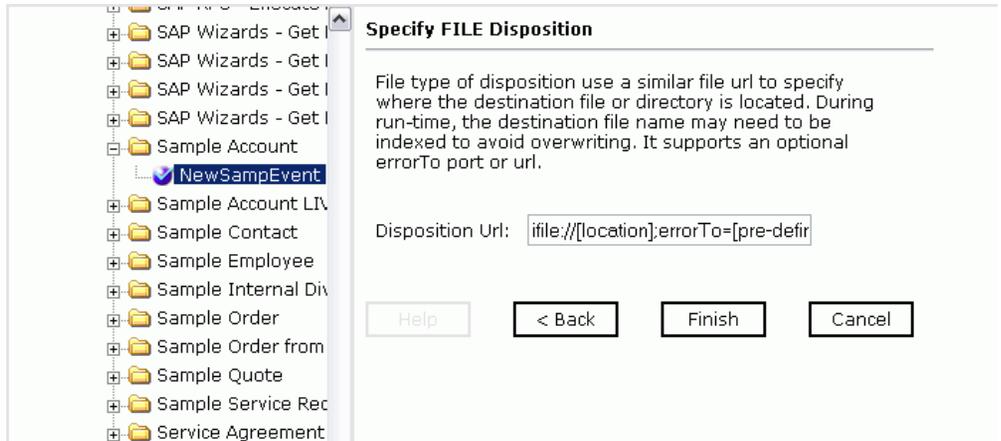
1. Select the Integration Object node you created.
2. Move the pointer over *Operations* and select *Create Event Port*.

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

The screenshot shows a 'Create Event Port' dialog box. It contains three input fields: 'Event Port Name' (highlighted in yellow), 'Event Port Description', and 'Disposition Protocol' (set to 'FILE'). Below the fields are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a black border.

- 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.



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

## Creating an Event Port From the Event Adapters Tab

The following procedures describe how to create an event port from the Event Adapters 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 Service Adapters tab. For more information, see *Creating an Event Port From the Service Adapters Tab* on page 3-3.

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

To create a specific event port for the File disposition:

1. Click the *Events Adapters* tab.  
The Event Adapters window opens.
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:

Description:

Disposition Protocol:  ▼

Disposition:

## Creating an Event Port

- 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.

- 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 3-16.

### Procedure: How to Create an Event Port for iBSE

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

To create an event port for iBSE:

1. Click the *Events Adapters* tab.

The Event Adapters window opens.

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 3-16.

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

To create an event port for a JMS queue:

1. Click the *Events Adapters* tab.  
The Event Adapters window opens.
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=;
```

- 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. The WebLogic connection factory is: <code>javax.jms.QueueConnectionFactory</code>

Parameter	Description
jndiurl	<p>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,</p> <p><code>java.naming.provider.url</code></p> <p>For BEA WebLogic Server this is</p> <p><code>t3://host:port</code></p> <p>where:</p> <p><code>host</code></p> <p>Is the machine name where WebLogic Server is installed.</p> <p><code>port</code></p> <p>Is the port on which WebLogic server is listening. The default port if not changed at installation is 7001.</p>
jndifactory	<p>Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider.</p> <p>For WebLogic Server, the WebLogic factory is</p> <p><code>weblogic.jndi.WLInitialContextFactory</code></p>
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 3-16.

### Procedure: How to Create an Event Port for MSMQ

To create an event port for MSMQ:

1. Click the *Events Adapters* tab.

The Event Adapters window opens.

## Creating an Event Port

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 <i>Private\$</i> .
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 3-16.

## Procedure: How to Create a Port for a SOAP Disposition

To create a port for a SOAP disposition:

1. Click the *Event Adapters* 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	<p>The URL to the WSDL file that is required to create the SOAP message. For example:</p> <pre>http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl</pre> <p>where:</p> <pre>webservice</pre> <p>Is the name of the Web service you created using Application Explorer.</p> <p>This value can be found by navigating to the Integration Business Services tab and opening the Service Description link in a new window. The WSDL URL appears in the Address field.</p> <p>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.</p>

Parameter	Description
soapaction	The method that will be called by the SOAP disposition. Integration Business Services This value can be found in the WSDL file.
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 *Events Adapters* tab.  
The Event Adapters window opens.
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.

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 *Events Adapters* tab.  
The Event Adapters window opens.
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*.

## Creating an Event 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=[channelname];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=
```

- 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.

## Event Adapters Event Adapters **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:

Description:

Disposition Protocol:  ▼

Disposition:

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 *Events Adapters* tab.

The Event Adapters window opens. 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.

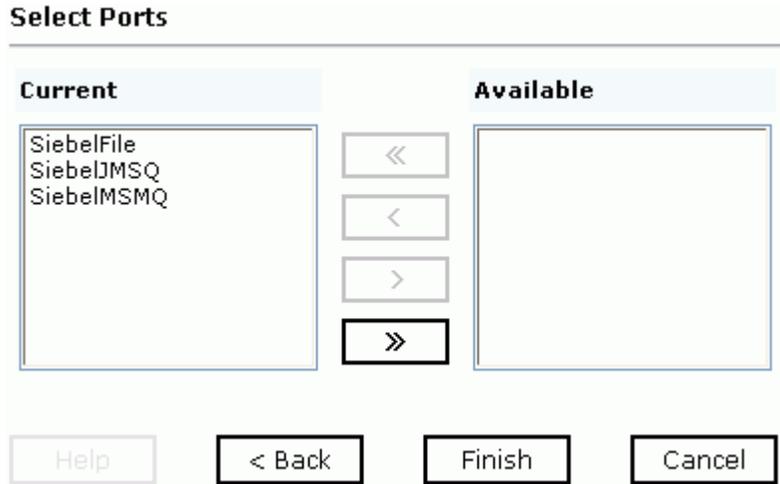
The following table includes the name and description of each parameter.

Parameter	Description
Listener port	Port on which to listen for Siebel event data.

Parameter	Description
Synchronization Type	Choose from three options: <ul style="list-style-type: none"><li>• REQUEST</li><li>• REQUEST_RESPONSE</li><li>• REQUEST_ACK</li></ul>

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*.

## Creating a Channel

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



Operations ▾	
<b>Channel Description</b>	HTTP event
<b>Channel Status</b>	Disconnected
<b>Ports</b>	[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 *Events Adapters* tab.

The Event Adapters window opens. 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: <ul style="list-style-type: none"> <li>• REQUEST</li> <li>• REQUEST_RESPONSE</li> <li>• REQUEST_ACK</li> </ul>

- 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. <ul style="list-style-type: none"> <li>Sequential indicates single processing of requests.</li> <li>Threaded indicates processing of multiple requests simultaneously.</li> </ul>
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 Create an MQSeries Channel

To create an MQSeries channel using Application Explorer:

1. Click the *Events Adapters* tab.

The Event Adapters window opens. 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).

## Creating a Channel

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	<p>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:</p> <p><i>Host\queue type\$qName</i></p> <p>where:</p> <p><i>Host</i></p> <p>Is the machine name where the MQSeries queuing system is running.</p> <p><i>queue type</i></p> <p>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.</p> <p><i>qName</i></p> <p>Is the name of the queue where messages are placed, for example,</p> <p><i>iwaykxcl\Private\$siebel</i></p>

- 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.

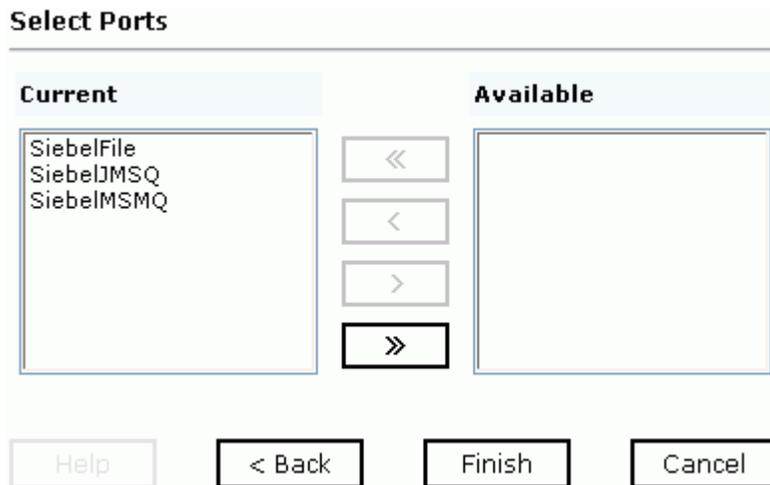
Parameter	Description
Synchronization Type	Choose from three options: <ul style="list-style-type: none"> <li>• REQUEST</li> <li>• REQUEST_RESPONSE</li> <li>• REQUEST_ACK</li> </ul>

- 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. <ul style="list-style-type: none"> <li>• Sequential indicates single processing of requests.</li> <li>• Threaded indicates processing of multiple requests simultaneously.</li> </ul>
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.

## *Creating a Channel*

---

---

## CHAPTER 4

# Using Web Services Policy-Based Security

### Topics:

- Integration Business Services Policy-Based Security
- Configuring Integration Business Services Policy-Based Security

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

**Note:** For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

## Integration Business Services Policy-Based Security

---

Integration 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 Integration 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 Integration Business Services. Instead, you can use one policy for many Integration 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 Integration 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 4-10.

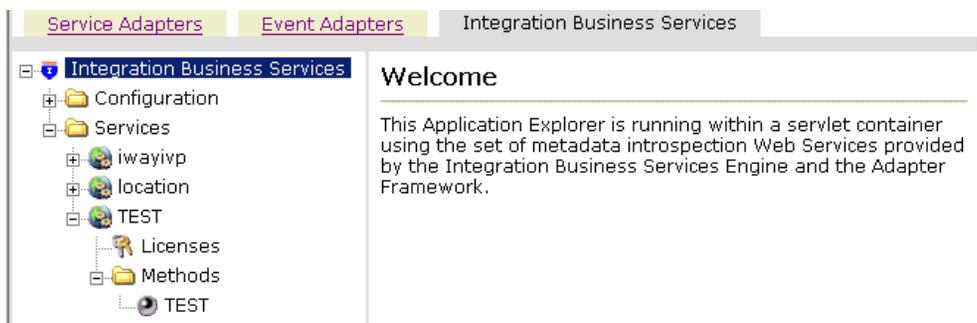
**Note:** For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

### 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 Service Adapters, Event Adapters, and Integration Business Services. The Integration Business Services tab is active and displays a Welcome screen on the right. The image shows the Integration Business Services node expanded in the left pane.



- a. Click the *Integration Business Services* tab.
- b. Expand the *Configuration* node.
- c. Expand the *Security* node.



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.

Operations ▶



## Users

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

User Id	Description
<input type="checkbox"/> 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 *Integration 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.

**Add new group**

Name:

Description:

- a. In the Name field, type 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.

**Modify Group Membership**

**Current**

**Available**

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.

Operations ▶



## Groups

A group is an object that can be granted or denied permissions to run Integration 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
------------	-------------

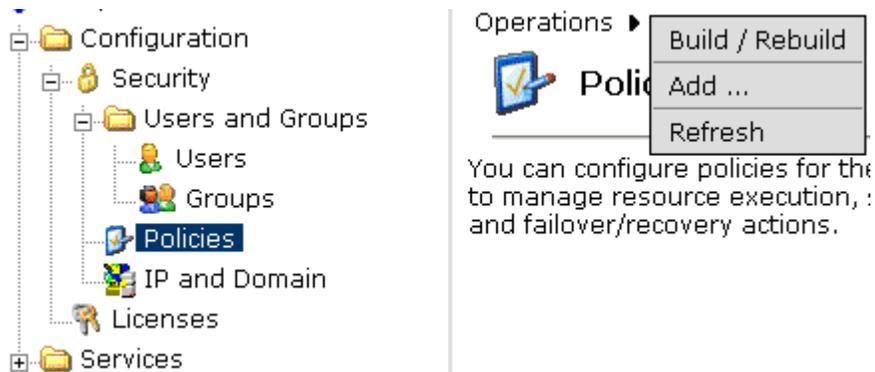
<input type="checkbox"/> newgroup	
-----------------------------------	--

### **Procedure** How to Create an Execution Policy

To create an execution policy:

1. Open Servlet Application Explorer.
  - a. Click the *Integration 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.



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

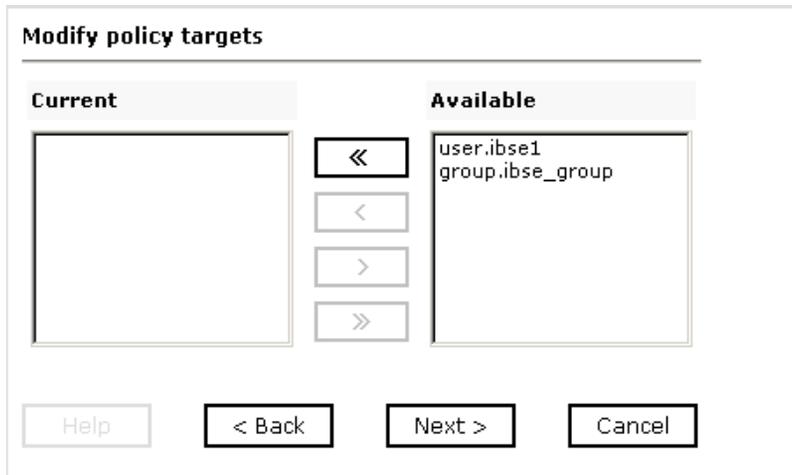
The following image shows the Add a new policy pane that opens with fields for entering the name, type, and description of the policy. To continue, click the *Next* button. The pane includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.

The dialog box is titled 'Add a new policy'. It contains three input fields: 'Name:' with a text box, 'Type:' with a dropdown menu showing 'Execution', and 'Description:' with a text area. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

- 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.



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.

Member Id	Permission
user.ibse1	Deny
group.ibse_group	Deny

Buttons: Help, < Back, Finish, Cancel

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 Integration Business Services Engine to manage resource execution, service routing, data restrictions and failover/recovery actions.

Name	Type	Description
<input type="checkbox"/> 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 *Integration 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:

Access Control:

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).

Operations ▶



## IP and Domain

You can configure the Integration 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
<input type="checkbox"/> 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 Integration 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

---

Integration 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 BEA WebLogic 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.

iBSE Settings:		Save
Property Name	Property Value	
<b>System</b>		
Language	English ▾	
Adapter Lib Directory	C:\Program Files\Way55\lib	
Encoding	UTF-8 ▾	
Debug Level	NONE ▾	
Number of Async. Processors	0 ▾	
<b>Security</b>		
Admin User	iway	
Admin Password	****	
Policy	<input type="checkbox"/>	
<b>Repository</b>		
Repository Type	File System ▾	
Repository Url	file://C:\Program Files\Way55\bea\ibe	
Repository Driver		
Repository User		
Repository Password		
Repository Pooling	<input type="checkbox"/>	
<a href="#">More configuration...</a>		
		Save

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.

Property Name	Property Value
<b>Monitoring</b>	
Repository Type	File System
Repository Url	file://C:\Program Files\Way55\bea
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>
<b>Auditing</b>	
Store Message	<input type="radio"/> yes <input checked="" type="radio"/> no
Max Message Stored	10,000
Save Configuration Save History View Events View Services	
Start Monitoring	

- 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 BEA WebLogic 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.

The screenshot shows a window titled "Service Statistics". It is divided into two main sections: "Web Service Methods" and "Statistics".

**Web Service Methods:** This section has a "Service" label and a "Method" label. Below "Service" is a drop-down menu currently showing "all".

**Statistics:** This section contains a table with the following data:

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

At the bottom right of the window is a button labeled "< home".

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.

<b>Statistic</b>	<b>Description</b>
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).

The screenshot shows a window titled "Service Statistics". It is divided into two main sections: "Web Service Methods" and "Statistics".

**Web Service Methods**

Service:  Method:

**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	<input type="text" value="select a correlation id"/>
Failed Invocations	<input type="text" value="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
Method

B0100033 ▾

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

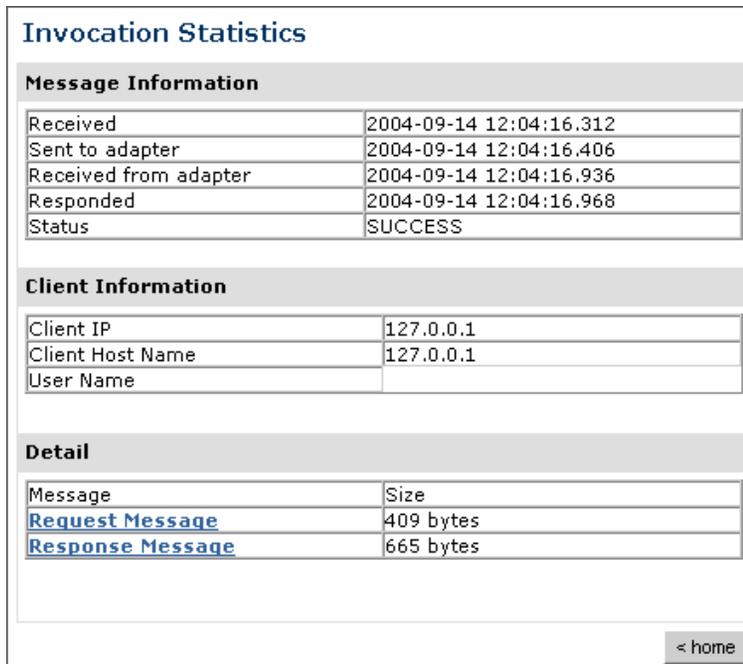
select a correlation id ▾

Suspend Service

< home

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.



The screenshot shows a window titled "Invocation Statistics" with three main sections: "Message Information", "Client Information", and "Detail".

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
<a href="#">Request Message</a>	409 bytes
<a href="#">Response Message</a>	665 bytes

< home

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 BEA WebLogic 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
Ports

all ▼

---

**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	<span style="border: 1px solid gray; padding: 2px;">select a correlation id</span> <span style="border: 1px solid gray; padding: 2px;">▼</span>
Failed Events	<span style="border: 1px solid gray; padding: 2px;">select a correlation id</span> <span style="border: 1px solid gray; padding: 2px;">▼</span>

< home

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.

<b>Statistic</b>	<b>Description</b>
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	<span style="border: 1px solid gray; padding: 2px;">select a correlation id ▾</span>
Failed Events	<span style="border: 1px solid gray; padding: 2px;">select a correlation id ▾</span>

Suspend Channel
Start Channel
< home

- 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.

The screenshot shows a window titled "Channel Statistics". It is divided into two main sections: "Channels" and "Statistics".

**Channels Section:** Contains two drop-down menus. The "Channels" menu is set to "TestChan" and the "Ports" menu is set to "TestPort".

**Statistics Section:** Contains a table with the following data:

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

At the bottom of the window, there are three buttons: "Suspend Channel", "Start Channel", and "< home".

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.

The screenshot shows a window titled "Message Statistics". It contains two main sections: "Message Information" and "Messages".

**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
<a href="#">Event Message</a>	446 bytes
Reply Message	na

In the bottom right corner of the window, there is a button labeled "< home".

- 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`

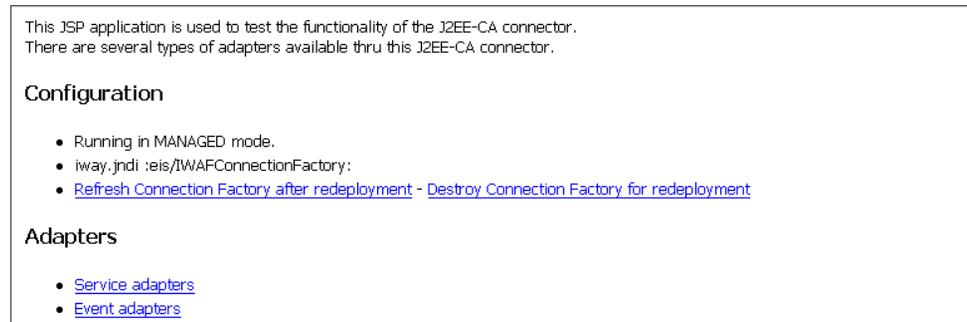
Is the name of the machine where your application server is running.

`port`

Is the port for the domain you are using. The port for the default domain is 7001.,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.



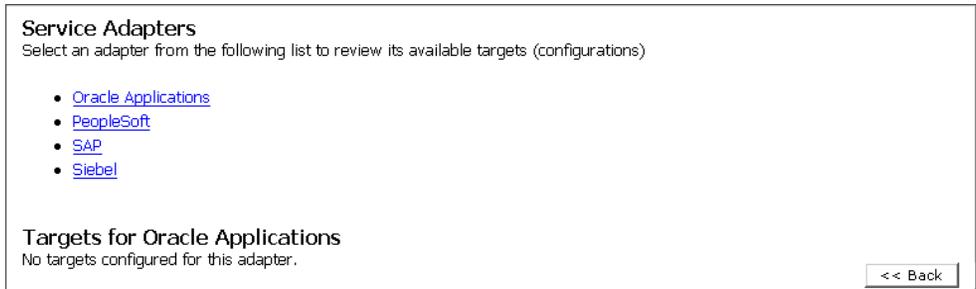
The JCA Test Tool runs in managed mode by default.

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 module using the BEA WebLogic Server console.
  - 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.



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.

The screenshot displays a web-based interface for testing services. It is divided into several sections:

- Service Adapters:** A section titled "Service Adapters" with the instruction "Select an adapter from the following list to review its available targets (configurations)". It contains a bulleted list of links: [Oracle Applications](#), [PeopleSoft](#), [SAP](#), and [Siebel](#).
- Targets for Siebel:** A section titled "Targets for Siebel" with a bulleted list containing [TestService](#).
- Statistics for Siebel target TestService:** A section showing performance metrics: TotalRequestCount : 0, TotalSuccessCount : 0, TotalErrorCount : 0, AverageExecutionTime : 0 msec., and LastExecutionTime : 0 msec.
- Request for Siebel target TestService:** A section with the instruction "Enter the data for this interaction. The configured user/password will be used if the User name is not provided." It includes three input fields: "User:" (text box), "Password:" (text box), and "Input Doc:" (a large text area with a vertical scrollbar).
- Buttons:** At the bottom of the form area are two buttons: "Send" and "Reset". In the bottom right corner of the entire interface is a button labeled "<< Back".

- 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 port for the domain you are using. The port for the default domain is 7001, 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.
- `iwjca.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 module using the BEA WebLogic Server console.
  - 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.

<b>Current channel Statistics</b>	
Commands: <a href="#">start</a> <a href="#">refresh</a>	
Active: false	
TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.
<b>Statistics for port 'fileIN'</b>	
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 *iWay Installation and Configuration for BEA WebLogic* 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 port for the domain you are using. The port for the default domain is 7001, 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*.

The default location for the trace information on Windows is:

`C:\Program Files\bea\ibse\ibselogs`

### 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 Installation and Configuration for BEA WebLogic* 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. The BEA WebLogic Server must be restarted to detect new repository changes.

### 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\bea\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\bea\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 Installation and Configuration for BEA WebLogic* 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

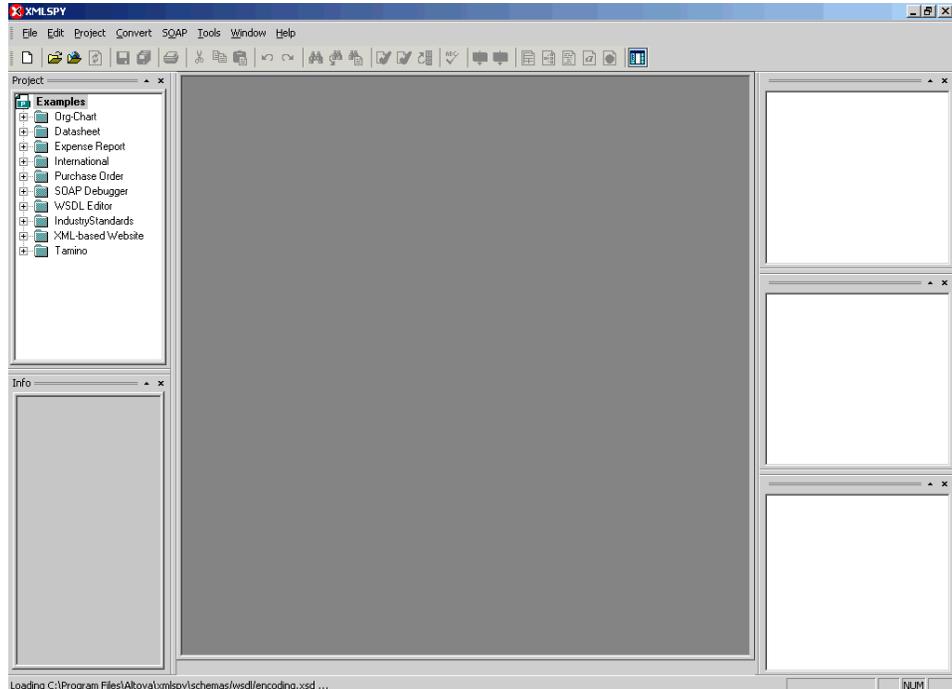
To migrate an iBSE repository that is configured for Oracle:

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

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

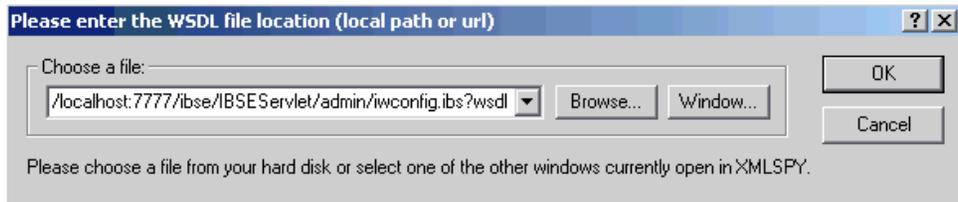
2. Open a third party XML editor, for example, XMLSPY.

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.



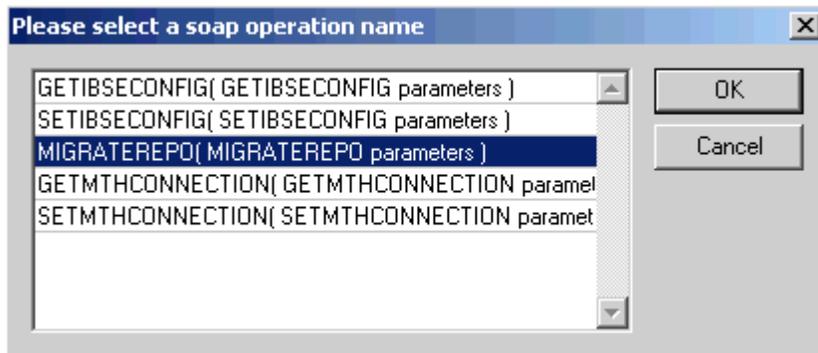
3. From the SOAP menu, select *Create new SOAP request*.

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.



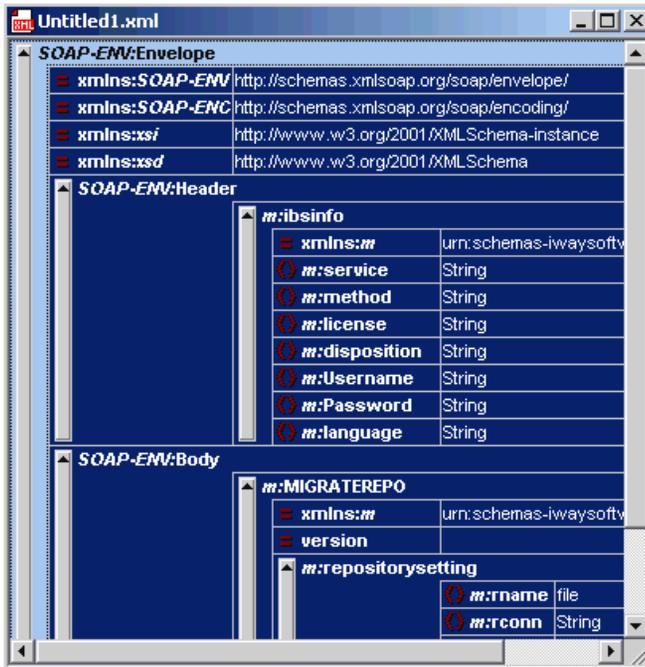
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

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*.



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.



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 Installation and Configuration for BEA WebLogic* 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:

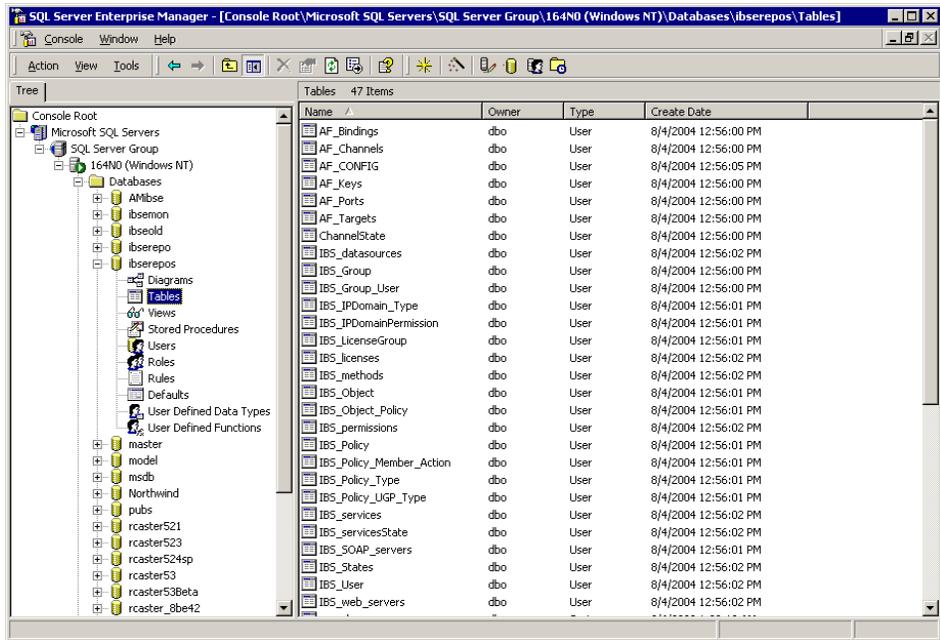
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

`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.



For more information on configuring the Microsoft SQL Server 2000 repository, see the *iWay Installation and Configuration for BEA WebLogic* 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:

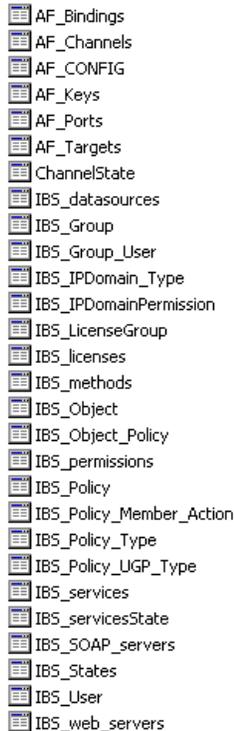
For Oracle 8:

`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 Installation and Configuration for BEA WebLogic* 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.



A screenshot of a database table list showing various tables related to iBSE and AF bindings. The tables listed are:

- 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 Installation and Configuration for BEA WebLogic* 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 Installation and Configuration for BEA WebLogic* 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.

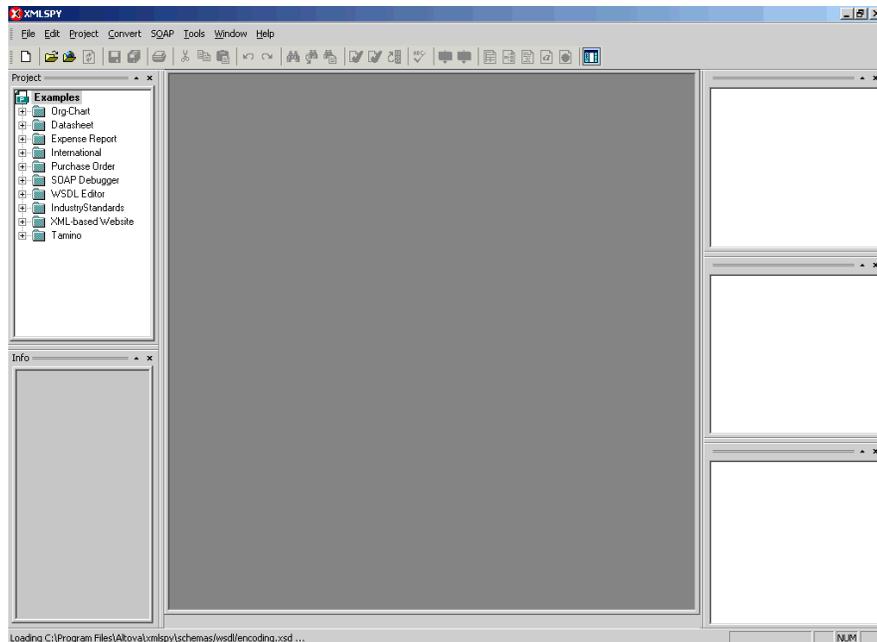
### Procedure: How to Export a Target

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.

## Exporting or Importing Targets

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.



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

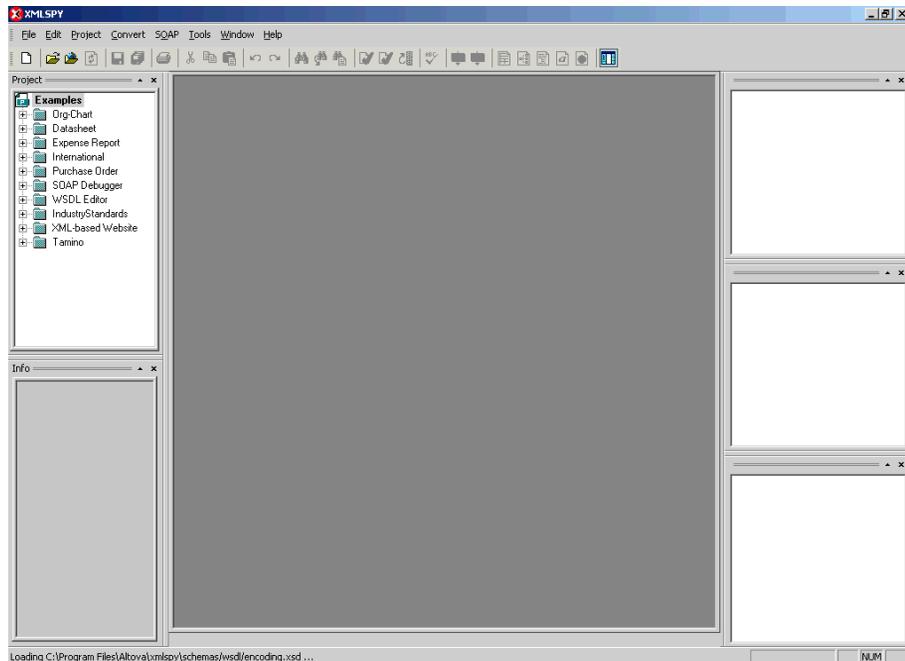
To import a target:

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.

## Exporting or Importing Targets

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.



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>R01GODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</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:

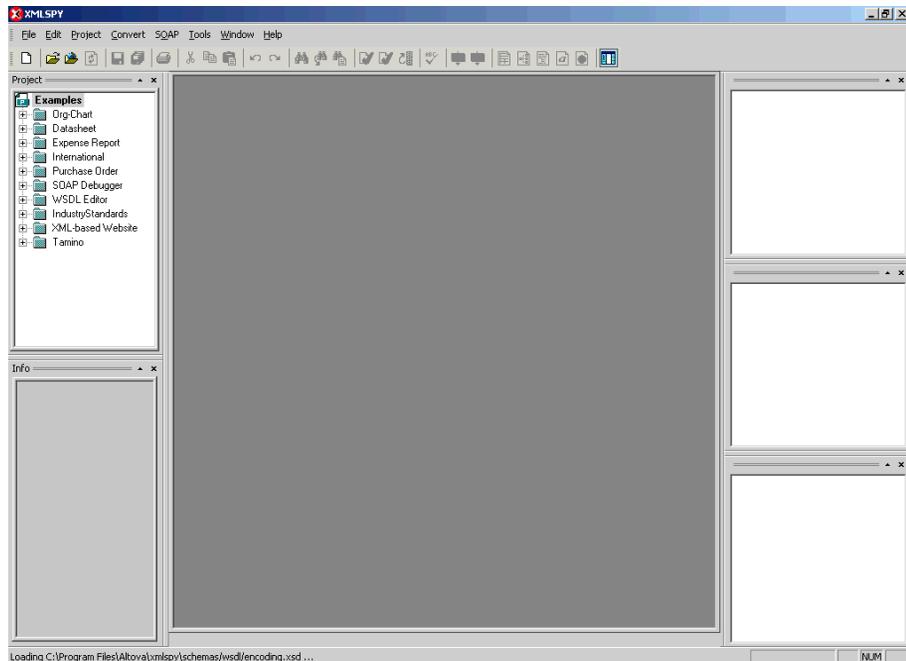
1. Copy the iBSE configuration service URL, for example:

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

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

## Retrieving or Updating Web Service Method Connection Information

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.



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:

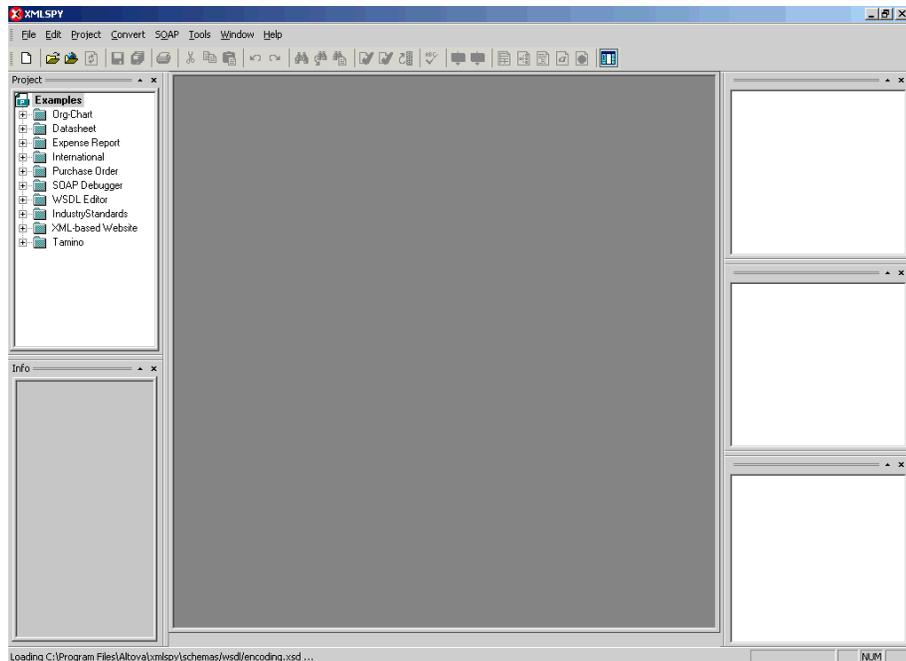
1. Copy the iBSE configuration service URL, for example:

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

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

## Retrieving or Updating Web Service Method Connection Information

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.



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

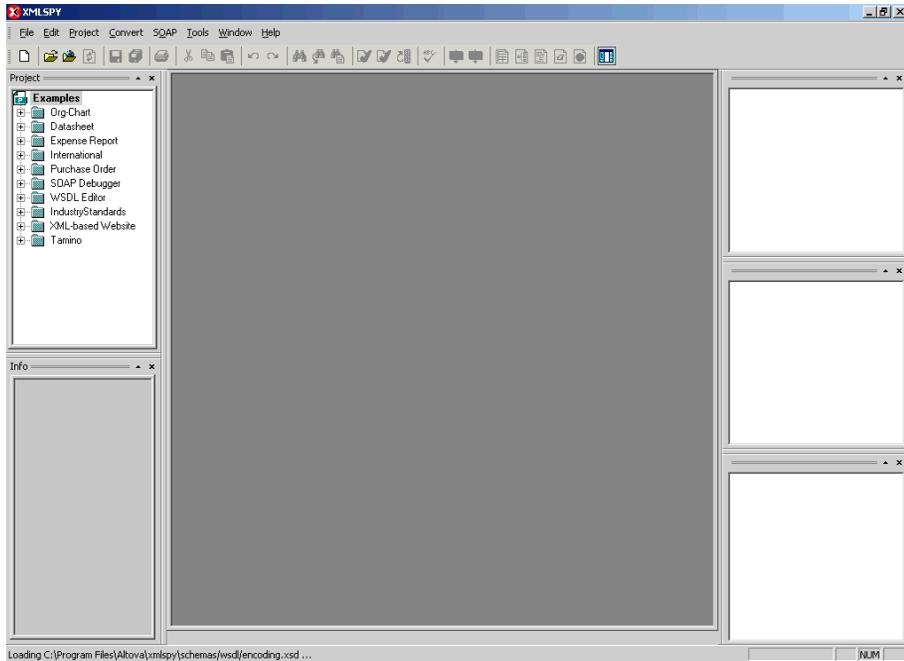
To start a channel programmatically:

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.

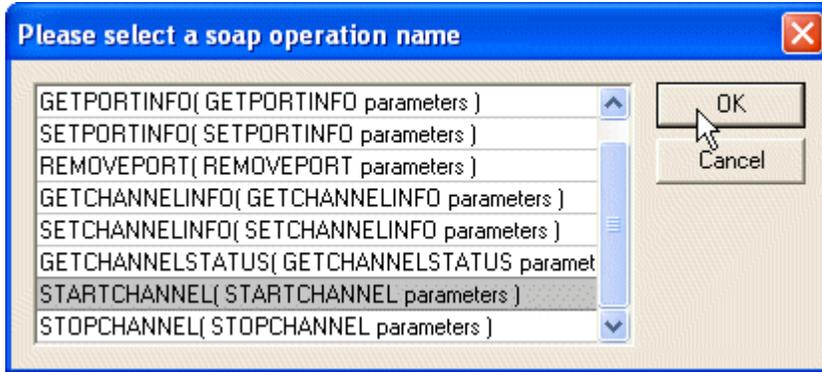
## Starting or Stopping a Channel Programmatically

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.



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.



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

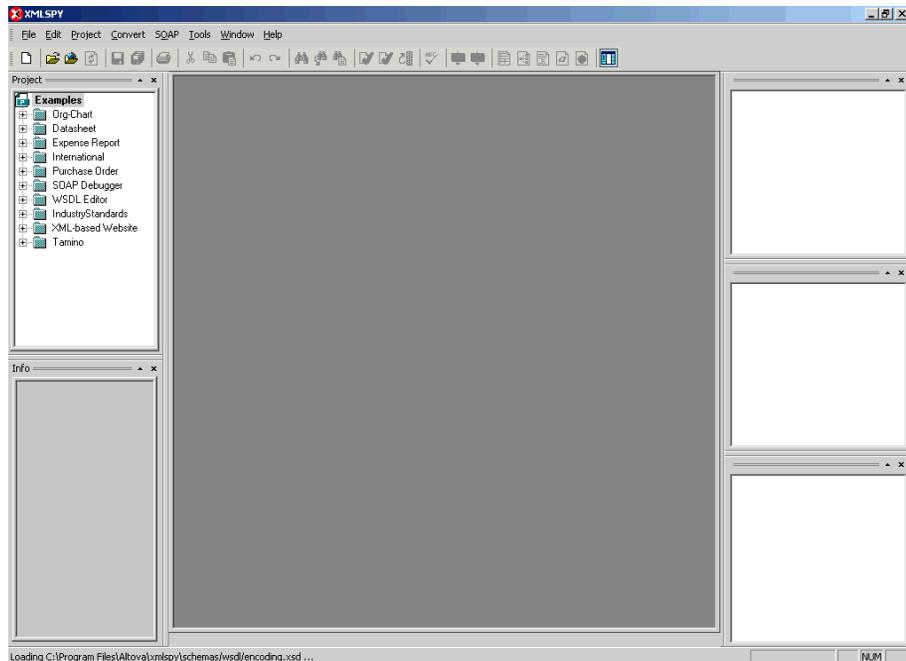
To stop a channel programmatically:

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.

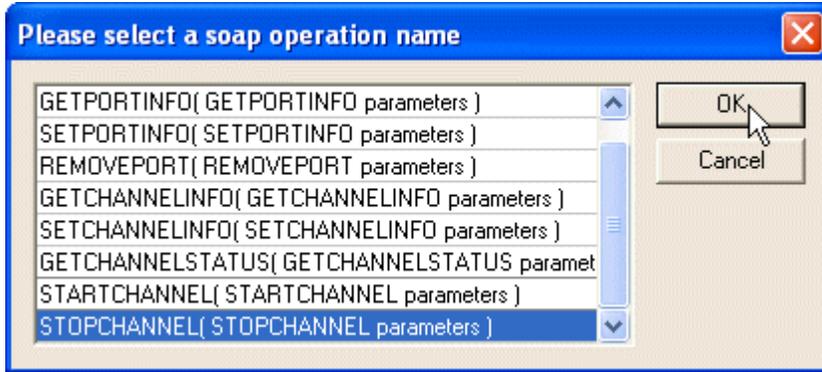


3. From the SOAP menu, select *Create new SOAP request*.

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.



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*.

## *Starting or Stopping a Channel Programmatically*

---

---

## CHAPTER 6

# Troubleshooting and Error Messages

### Topics:

- Troubleshooting
- iBSE Error Messages
- Updating a Siebel Field

This section explains the limitations and workarounds when connecting to Siebel.

The adapter-specific errors listed in this appendix can arise whether using the adapter with a JCA or with an iBSE configuration.

## Troubleshooting

---

This topic provides troubleshooting information for Siebel, separated into four categories:

- Application Explorer
- Siebel
- JCA
- iBSE

**Note:** Log file information that can be relevant in troubleshooting can be found in the following locations:

- The JCA trace information can be found under the C:\ProgramFiles\iWay55\config\base\log directory.
- iBSE trace information can be found under the C:\Program Files\iWay55\bea\ibse\ibselogs directory.
- The log file for Application Explorer can be found under the C:\Program File\iWay55\tools\iwae\bin directory.

### Application Explorer

The following table lists errors that occur in Application Explorer and provides a solution to each error.

Error	Solution
Siebel does not appear in the Application Explorer Adapter node list.	Ensure that the Siebel JAR files supplied with your Siebel distribution media were placed in the <i>iway_home/lib</i> directory. For example, for Siebel 7.03 environments, the SiebelJI_Common.jar and SiebelJI_enu.jar should be placed in the <i>iway_home/lib</i> directory.
Target Type drop-down list contains only Java Data Bean Connection, and COM connection type is required.	Ensure that the Siebel thin client is installed correctly on the machine hosting Application Explorer so that appropriate COM environment is available.

Error	Solution
<p>An error message that includes the name of the Siebel Gateway server appears when you try to connect to a Siebel target, for example,</p> <pre>Problem activating adapter (ariba0x). Check logs for more information.</pre>	<p>Ensure that the name of the Siebel Gateway server is correctly defined for the target you want to connect to.</p>
<p>You receive the following error when trying to connect to a Siebel target:</p> <pre>Problem activating adapter. (You have entered an invalid set of logon parameters. Please type in your logon parameters again.). Check logs for more information.</pre>	<p>Ensure that the User ID and password parameter values to connect to your Siebel system are correct.</p>
<p>You receive the following error when trying to connect to a Siebel target:</p> <pre>Problem activating adapter. (Couldn't get nameserver connection). Check logs for more information.</pre>	<p>Check the network connectivity to the Siebel environment. Correct networking problem and retry connection.</p> <p>Also, if Siebel was started recently, it might not be fully functional yet. If so, wait until Siebel starts completely.</p> <p><b>Note:</b> If Siebel Server was restarted after servlet iBSE connected, then servlet iBSE also must be recycled. This is due to a known Siebel issue. For more information, see Siebel Alert 984.</p>
<p>You receive the following error when attempting to connect to a Siebel target:</p> <pre>Problem activating adapter. (NSReadKey request failed (no error information)...). Check logs for more information.</pre>	<p>Ensure that the values defined for Siebel Server, Enterprise Name, and Object Manager for the target to which you want to connect are correct and retry the connection.</p>

Error	Solution
<p>You receive the following error when attempting to connect to a Siebel target:</p> <pre data-bbox="267 378 701 546">Problem activating adapter. (Error loading translatable messages: com.siebel.locale.enux.message s.SSAMessages_enux). Check logs for more information</pre>	<p>Ensure that the value of the Language parameter on the Advanced tab is defined correctly for the target you are using to connect to your Siebel system (for example, enu for English).</p>
<p>A successful connection is made to the Siebel environment, but no values are available in Business Object, Business Service, and Integration Object nodes in the Application Explorer tree.</p>	<p>The Repository Name specified on the Advanced tab in the Siebel target configuration is either void or empty of any components in the targeted Siebel environment, or the Repository Name is not valid for the targeted Siebel environment. Verify that the Repository Name is valid and contains components for interrogation, and then reconnect.</p>

## Siebel

The following table lists errors that occur when using the adapter with either an iBSE or JCA repository project and provides a solution to each error.

Error	Solution
<p>A successful connection is made to Siebel environment, but no values are available in the Business Object, Business Service, and Integration Object nodes in the Application Explorer tree.</p>	<p>The Repository Name specified on the Advanced tab in the Siebel Target configuration is either void or empty of any components in the targeted Siebel environment, or the Repository Name is not valid for the targeted Siebel environment. Verify that the Repository Name is valid and contains components for interrogation and then, reconnect.</p>
<p>When executing a request, the following error message appears:</p> <pre data-bbox="267 1447 687 1501">AdapterException: Unsupported Action: {0} Tquery</pre>	<p>Verify that the method is available for the specific request by verifying schema.</p>

Error	Solution
<p>When executing a request, the following error message appears:</p> <pre>AdapterException: Field 'NFame' does not exist in definition for business component 'Account'. Please ask your systems administrator to check your application configuration.</pre>	<p>Ensure that the field names are valid within the request document by referring to the schema for that specific object and then, resubmit the request.</p>
<p>When connecting to releases prior to Siebel 7.7 using the Java Data Bean Interface, you cannot reconnect after initial connection loss. This might occur when Application Explorer experiences a brief loss of network connection or if the Siebel Server or Gateway Service is restarted while Application Explorer is logged into the Siebel application.</p>	<p>Restart your application server and Application Explorer in order to log in successfully to the Siebel application. This is a known Siebel API issue. For more information, see Siebel Alert 984.</p>

## JCA

The following table describes an error that occurs when connecting to a JCA configuration and provides a solution to the error.

Error	Solution
<p>In Application Explorer, the following error message appears when you attempt to connect to a JCA configuration:</p> <pre>Could not initialize JCA</pre>	<p>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example,</p> <pre>iway_home/lib</pre>

## iBSE Error Messages

---

This topic discusses the different types of errors that can occur when processing Integration Business Services through the Integration Business Services Engine (iBSE).

### General Error Handling

iBSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and execution time, various conditions can cause errors in iBSE when Integration Business Services that use adapters are running. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis.

Usually, the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the iBSE. If an error occurs, the way it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside iBSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when iBSE receives an invalid SOAP request:

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Parameter node is missing</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In this example, iBSE did not receive an element in the SOAP request message that is mandatory for the WSDL for Integration Business Services.

## Adapter-Specific Error Handling

When an adapter raises an exception during execution, the SOAP agent in iBSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether or not an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in iBSE, and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault.

While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Integration Business Service consumer application.

### BEA WebLogic Adapter for Siebel Invalid SOAP Request

When the adapter receives a SOAP request message that does not conform to the WSDL for the Integration Business Services being executed, the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>XD[FAIL] Parse failure (IS) 3:
org.xml.sax.SAXParseException: Premature end of file.</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Empty Result From Siebel Request

When the adapter cannot connect to Siebel when executing a Web service, the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
```

```
<faultstring><Exception> - major:4096 minor: -1 message:NSReadKey
request 11 was abandoned after 37846ms connection:12a due to Connection
shutdown request Connection reset by peer:JVM_recv in socket input stream
stream read DetailedMessage:Unknown</Exception></faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Invalid SOAP Request

When the adapter receives a SOAP request message that does not conform to the WSDL for the Integration Business Services being executed, the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1"
?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>SOAP-ENV:Server</faultcode>
<faultstring>RPC server connection failed: Connection refused:
connect</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Empty Result From BEA WebLogic Adapter for Siebel Request

When the adapter executes a SOAP request using input parameters passed that do not match records in the target system, the following SOAP response is generated.

**Note:** The condition for this adapter does not yield a SOAP fault.

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
xmlns="urn:schemas-iwaysoftware-com:iwse"
cid="2A3CB42703EB20203F91951B89F3C5AF">
<RunDBQueryResult run="1" />
</m:RunDBQueryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Updating a Siebel Field

---

If you have problems updating a Siebel field, you might need to perform configuration changes on the Siebel system.

### **Procedure** How to Verify Whether a Siebel Field is Activated for Updates

This procedure describes how to verify whether a Siebel field is activated for updates.

1. Obtain the siebel.srf file from the Siebel server with which you are having difficulties.

The .srf file is typically located in the \tools\OBJECTS\ENU folder of the Siebel server install, for example:

```
D:\sea752\siebsrvr\OBJECTS\ENU
```

2. Under the Siebel\tools directory, locate the siebel.srf file.

The file is typically located under a similar folder, for example:

```
D:\sea752\tools\OBJECTS\ENU
```

3. Rename the siebel.srf file in the Siebel\tools directory, for example, siebel.srf.original.
4. Place a copy of the siebel.srf from the Siebel server into the Siebel\tools directory and then, make a copy of it in the same directory and call it siebeltest.srf.
5. Start Siebel Tools and connect to the Siebel server.
6. Identify the Business Component for the Siebel field that is not being updated, for example, the Business Component Account for the Home Page field.
7. With the Account Business Component highlighted, click *Field* in the Object Explorer.
8. Identify the field which is not activated for updates or not subject to adapter update request, for example, Home Page or Email Address.
9. Check the properties for the Siebel field, specifically for the Force Active property.

If the property is false, that is, it has no check box selected or it has the value FALSE (this depends on the Siebel Tools GUI configuration that you are using), then you must change it to true, as explained in the following steps.

**Note:** This procedure refers to the check box as the method for making the property TRUE.

- a. If the Force Active property is false, select *Lock Project* from the Tools menu.

Note the name of the project being locked. It will be specified in the Project property of the Business Component for the field being made Force Active.

- b. Click *Force Active* to set it to true.

This places a check mark in the property field.

- c. After setting the Force Active property to true for the Account Business Component, select *Unlock Project* from the Tools menu.

You must also check the Multivalue Link property of the field. If there is no value in the Multivalue Link property, then proceed to the next step. If there is a value in this property, it is a reference to another Business Component. This indicates that the field in question is linked to a field in the Business Component identified by this property. Ensure that the linked field in the Business Component specified by the Multivalue Link property of the first field has a Force Active property of true.

For example, if the Email Address field has a Multivalue link property with a value of Business Address, you must go to the Business Address Business Component and ensure that the linked field in the Business Address Business Component also has a value of true in the Force Active field. Repeat Step 6 through Step 9, including substeps, for the field in the Business Component specified in the Multivalue link property.

You must compile the project so that the changes take effect.

10. Select *Compile Projects* from the Tools menu.

The Object Compiler appears.

- a. Select the *Selected projects* option button.
- b. Select the Project(s) to compile. To select multiple projects, hold down the Ctrl key as you select each project.

If the field had a Business Component defined in the Multivalue Link property, you also must select the project(s) for the additional Business Component(s) you updated.

- c. Click *Browse* and navigate to the location of the copied siebel.srf file in the Siebel\tools directory.

In this procedure, the copied file is siebeltest.srf.

11. Click *Compile*.

Compiling should take several minutes.

12. Transfer the newly compiled siebeltest.srf file to the Siebel Server system and replace the siebel.srf located there under \siebsrvr\OBJECTS\ENU with this new one. You must:
  - a. First stop the Siebel Server.
  - b. Make a backup copy of the current .srf file.

- c.** Replace the .srf file with siebeltest.srf by placing the .srf file in the \siebsrvr\OBJECTS\ENU directory and renaming it to siebel.srf (the original file name).

**13.** Restart the Siebel server and wait until it starts up completely.

This may take up to 5-10 minutes.

- a.** Open Task Manager and wait to see that CPU usage stops hovering at 100% and returns to a more normal range for activity on your machine.

- b.** Retest.

It now should be possible to update the fields through the adapter or Siebel application.



---

---

## APPENDIX A

# Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration Business Services

### Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Creating a New Configuration for iBSE or JCA
- Connecting to Siebel
- Viewing Metadata
- Creating an XML Schema for a Siebel Business Component or Siebel Business Service
- Creating an Integration Business Service
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an Extensible CCI Control to a BEA WebLogic Workshop Application

This section describes how to use the Java Swing implementation of Application Explorer as deployed in BEA WebLogic Workshop. Application Explorer deployed in WebLogic Workshop is functionally similar to the Servlet Application Explorer.

## Starting Application Explorer in BEA WebLogic Workshop

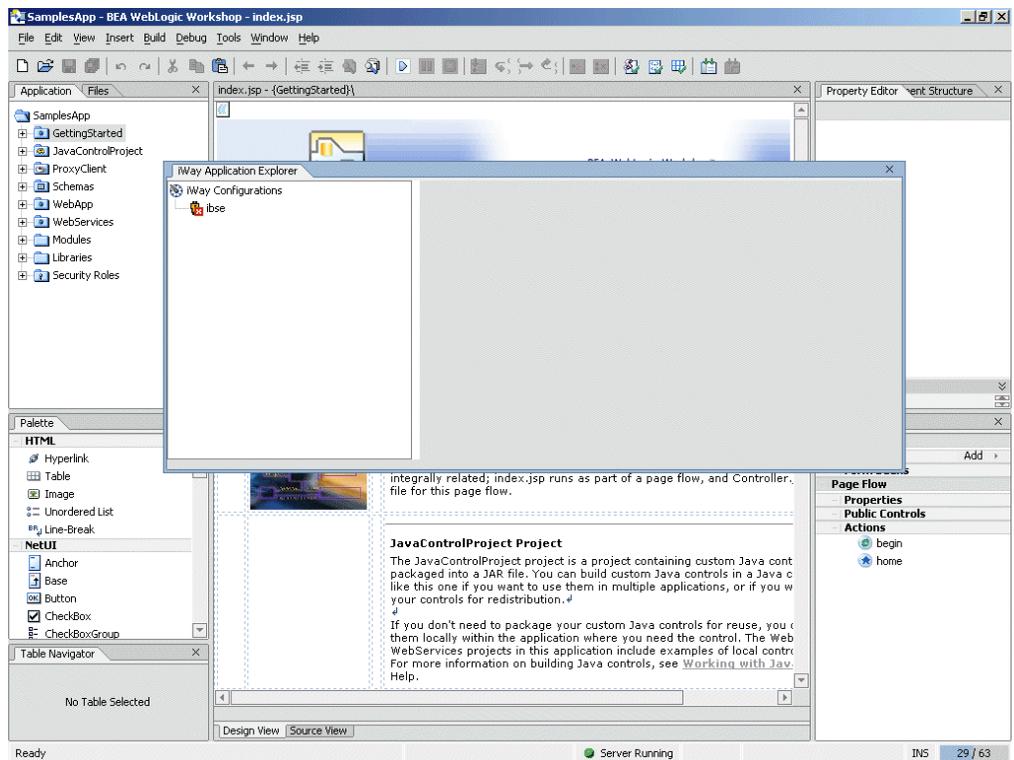
The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

### Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer:

1. Ensure the server on which Application Explorer is deployed is started.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *iWay Application Explorer*.

The following image shows the Application Explorer window that opens, with the BEA WebLogic Workshop window in the background. The left pane of the window contains the active Application Explorer tab that lists configurations. The right pane is empty.



You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

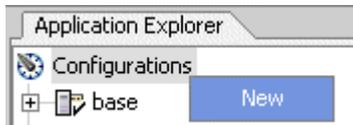
## Creating a New Configuration for iBSE or JCA

---

Before you can start using Application Explorer, you must define a new configuration for iBSE or JCA.

### Procedure: How to Create a New Configuration for iBSE or JCA

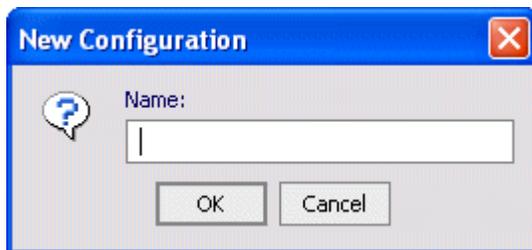
The following image shows the Application Explorer tab in the left pane with the Configurations node selected.



To create a new configuration:

1. Right-click *Configurations* and select *New*.

The following image shows the New Configuration pane that opens where you create a new configuration.

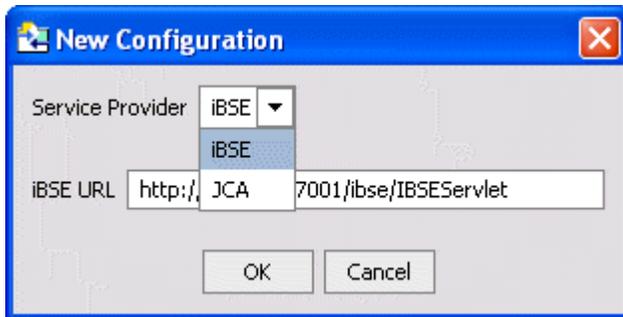


2. Type the name of the new configuration and click *OK*.

**Note:** If you are creating a new JCA configuration, type *base* in the name field. You must use this value if you are pointing to the default iWay configuration.

## Creating a New Configuration for iBSE or JCA

The following image shows the drop-down list and iBSE URL field that appear in the New Configuration pane. There are also two action buttons.



- From the Service Provider drop-down list, select *iBSE* or *JCA*.
  - If you select *iBSE*, type the URL for *iBSE*, for example,  
<http://localhost:7001/ibse/IBSEServlet>  
where:  
[localhost](#)  
Is where your application server is running.
  - If you select *JCA*, enter the full path to the directory where *iWay 5.5* is installed, for example,

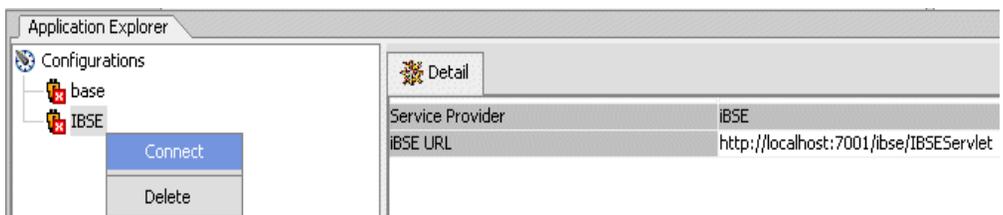
[C:\Program Files\iWay55](#)

where:

[iWay55](#)

Is the full path to your *iWay* installation.

The following image shows the Application Explorer tab active in the left pane. A node representing the new configuration appears beneath the Configurations node. The right pane provides details of the configuration you created.



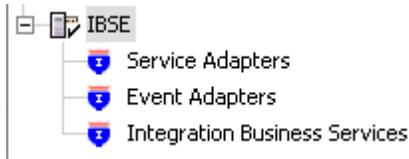
After you add your configuration, you must connect to it.

- Right-click the configuration to which you want to connect, and select *Connect*.

The following image shows the Service Adapters and Event Adapters nodes that appear beneath the configuration when you connect to **JCA**.



The following image shows the Service Adapters, Event Adapters, and Integration Business Services nodes that appear beneath the configuration when you connect to **iBSE**.



5. To display the service and event adapters that are installed, expand each node.

The Service Adapters list includes a Siebel node that enables you to connect to Siebel metadata and create XML request and response schemas to use to listen for events or create Web services. For more information, see *Creating an Integration Business Service* on page A-22.

The Event Adapters list includes a Siebel node that enables you to create ports and channels for Siebel event handling. For more information, see *Adding a Control for an iWay Resource in BEA WebLogic Workshop* on page A-28.

## Connecting to Siebel

To browse the Siebel Business Services, Business Components, and Integration Objects, you must define a target to Siebel and connect to it. The target serves as your connection point. The left pane displays the application systems supported by Application Explorer based on the iWay adapters you installed and are licensed to use.

### Creating and Connecting to a Target

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.

#### Procedure: How to Create a New Target

To create a new target:

1. In the left pane, expand *iWay Service Adapters* and click the *Siebel* node.
2. Right-click the *Siebel* node and select *Add Target*.

The Add Target dialog box appears.

- a. In the Name field, type a name for the new target, for example, SiebelConnection.
- b. In the Description field, type a brief description of the target (optional).
- c. From the Target Type drop-down list, select the type of target.

For Siebel 6.0, choose *Siebel 6.2*.

For Siebel 6.3, choose *Java Bean Data Connection*.

**3.** Click OK.

When you select 6.2 or lower (COM), provide the following Siebel information:

- User Agent File (UAGENT.CFG)
- Username
- Password
- Repository

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

When you select 6.3 (JDB), provide the following Siebel information:

- Gateway Server
- Enterprise Name
- Siebel Server
- User
- Password

In the Advanced tab, verify the following:

- Object Manager
- Repository Name
- Language

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. The default value is EAIObjMgr. For Siebel 7.0.3, the default Object Manager is EAIObjMgr. For Siebel 7.7, the default is EAIObjMgr\_enu. 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 will be returned. If a specified repository is not found, an empty list of objects will be 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.

4. Click *OK*.

The new target appears under the Siebel node.

## Procedure: How to Connect to a Target

To connect to a target:

1. In the left pane, expand the *Siebel* node and select the target to which you want to connect.

The following image show the Siebel node selected with two disconnected targets beneath it.



2. In the right pane, enter the password for that target.
3. In the left pane, right-click the target name and select *Connect*.

The following image shows the Siebel node with the SiebelConnection target icon selected beneath it. The icon changes to reflect that you are connected to the Siebel system and a new disconnected target icon appears beneath it.



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

## Managing a Target

Although you can maintain multiple open connections to different application systems, iWay Software recommends that you close connections that are not in use. After you disconnect, you can modify an existing target. You can modify the connection parameters when your system properties change.

You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of targets in the left pane of the explorer.

The following procedures describe how to disconnect from a target, edit a target, and delete a target.

### **Procedure: How to Disconnect From a Target**

To disconnect from a target:

1. In the left pane, right-click the target to which you are connected.
2. 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 you disconnected from the target.

### **Procedure: How to Edit a Target**

To edit a target:

1. Ensure the target you want to edit is disconnected.
2. In the left pane, right-click the target and select *Edit*.  
The Edit pane opens on the right.
3. Modify the target information.
4. To continue editing, click *Next*.
5. When you have finished making all your edits, click *Finish*.

### **Procedure: How to Delete a Target**

To delete a target:

1. In the left pane, right-click the target, for example, SiebelConnection.
2. Select *Delete*.

A confirmation dialog box opens.

3. To confirm that you want to delete the target, click *OK*.

The SiebelConnection node disappears from the left pane.

## Viewing Metadata

---

Viewing metadata is useful in 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 are not connected, 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 view metadata.
3. Expand the Business Object or Business Service node.
4. To view the objects under a Business Component or Business Service, expand 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 *Table* tab.

The following image shows the metadata table that appears in the right pane. The table consists of a heading row with column labels that identify each of the seven columns: Name, Type, Required, MultiValued, ReadOnly, Active, and Hidden. Each row represents a new data item.

Name	Type	Required	MultiValued	ReadOnly	Active	Hidden
Account Co...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Co...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Mar...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Org...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Pro...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Role	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Sta...	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Trend	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Address Acti...	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Address Id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Address Int...	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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

---

You can create service schemas for Business Services and Business Components using iWay Application Explorer.

The following topic describes how to create schemas for the adapter when you use Application Explorer running within the BEA WebLogic Workshop.

For information on creating schemas for Integration Objects, see *Creating a Schema for a Siebel Integration Object* on page A-13.

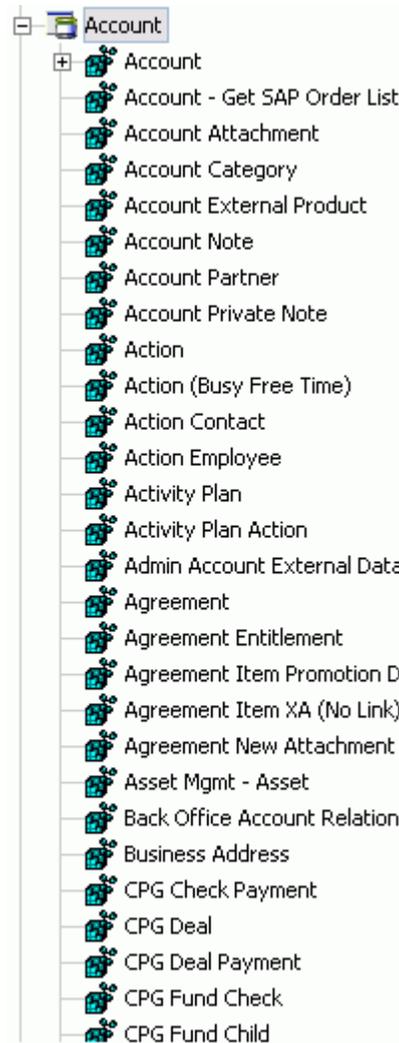
For more information on creating Web services, see *Creating an Integration Business Service* on page A-22.

## Creating an XML Schema for a Siebel Business Object or Business Service

You can create schemas for a Siebel Business Component method (for example, the Add method) or a Siebel Business Service using iWay Application Explorer.

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

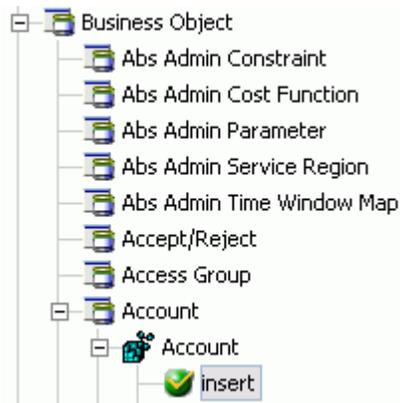
The following image shows the Account Business Object expanded to display all Business Components.



## 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 are not connected, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the Business Object or Business Service node.
3. Expand the Business Object or Business Service to view the objects under it.
  - The following image shows the Account **Business Component** expanded to show the Account node beneath it. The Account node is expanded to show the insert node, which is selected.



- a. Expand the component containing the node you want.
  - b. Expand the node and select the method for which you want to create a schema, for example, insert.
- The following image shows the Simple Add Account **Siebel Business Service** expanded to show the addAccount node beneath it. The addAccount node is selected and expanded to show the objects beneath it.



- a. Expand the Business Service node containing the object for which you want to create schema, for example, Simple Add Account.
- b. Select the object, for example, addAccount.

Application Explorer accesses the Siebel repository and builds schemas. The Table, Request Schema, and Response Schema tabs appear in the right pane.

4. To view a schema, click the tab 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" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:z="urn:iwaysoftware:adapte...
  <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/...
    </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...
      <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:sequence>
  </xsd:complexType>
</xsd:schema>
```

For more information on where the schemas are stored, see *Schema Location* on page A-21.

## Creating a Schema for a Siebel Integration Object

The BEA WebLogic Adapter for Siebel supports access to Siebel Integration Objects by using Siebel XML to handle events. Using Siebel Integration Objects through supported transports requires Siebel Workflows. For more information, see Appendix D, *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 Schema for a Siebel Integration Object

For releases prior to Siebel 6.3, the Siebel Tools Schema Wizard only creates 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 C, *Usage Considerations and Sample Files*.

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

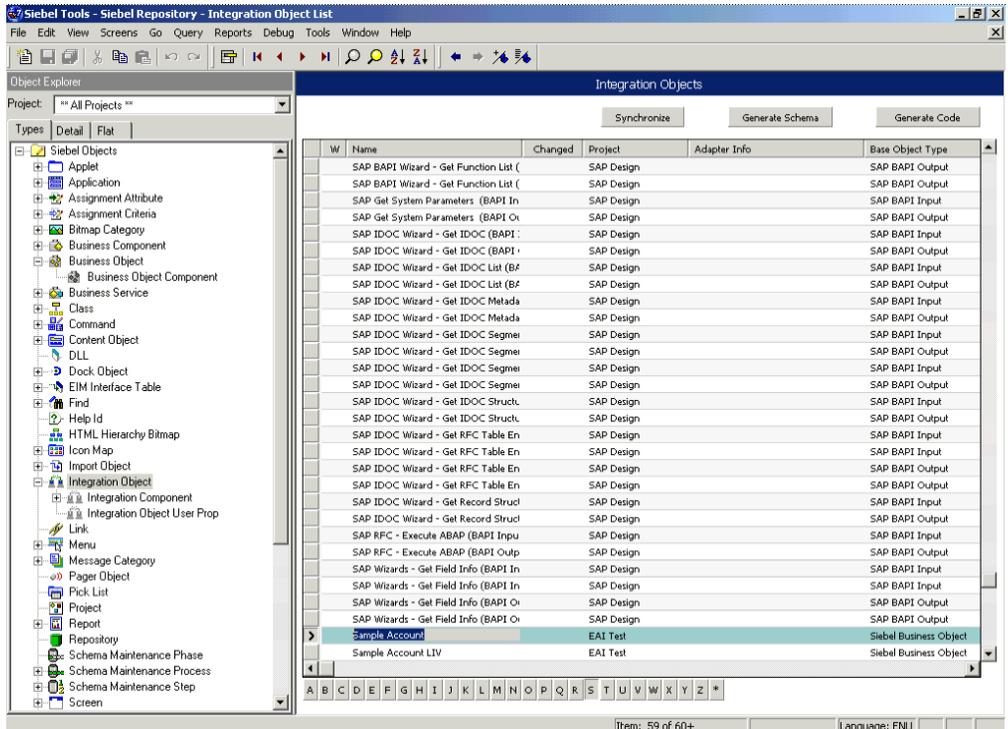
The following image shows the Siebel Tools screen where you log on.



To 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.

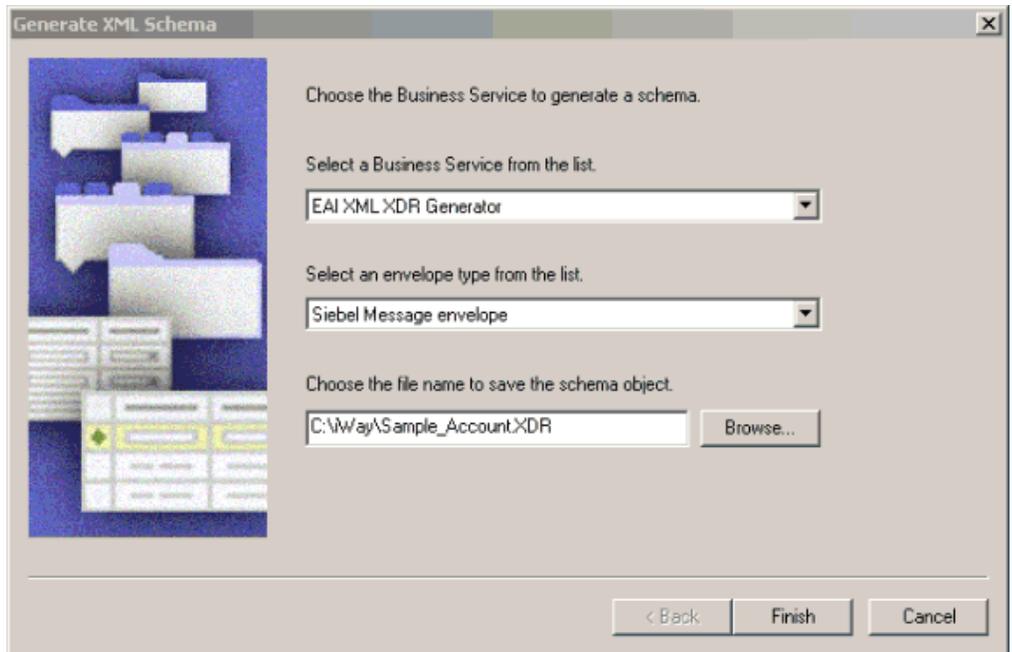
The following image shows the Siebel Tools window that opens. In the upper left is the Project drop-down list. On the lower left is the Object Explorer navigation pane with three tabs: Types, Detail, and Flat, with the Types tab selected to show a list of Siebel objects. The right pane has three buttons: 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.



3. To create an XML schema, select an integration object, for example, Sample Account.
4. Click the *Generate Schema* button.

## Creating a Schema for a Siebel Integration Object

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



- 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, specify a file name for the XDR schema and a directory where it can be accessed by Application Explorer.

For example, type the following sample name and directory or click *Browse* to select *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 *Create a Schema For a Siebel Integration Object* on page A-17.

## 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.

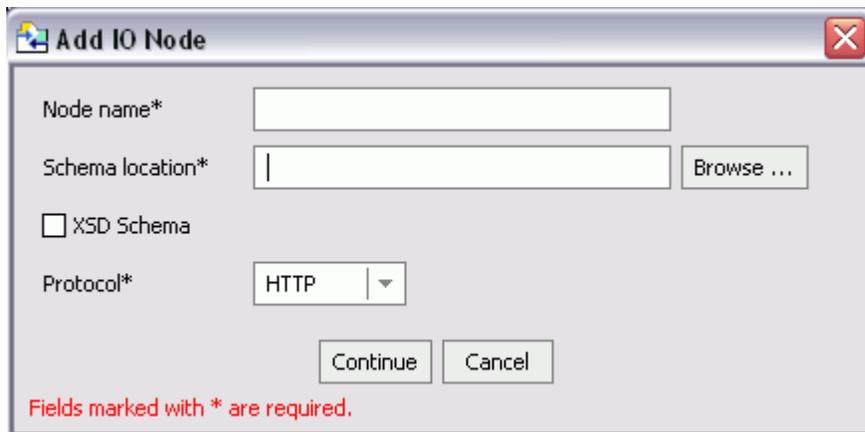
**Note:** The XDR file must be on the same computer as Application Explorer or be available through a mapped connection to another drive or machine.

### 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.
2. Scroll down and select an integration object, for example, Sample Account.
3. To generate a schema, right click the node and choose *Add IO Node*.

The following image shows the Add IO Node dialog box that appears. The dialog box has two action buttons, Continue and Cancel.



- 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.`

## Creating a Schema for a Siebel Integration Object

- c. If you are using Siebel-generated XSD schemas, select *XSD schema*.
- d. Choose a protocol used by the Siebel workflow for the Integration Object.
  - 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?SWEEExtSource=<SourceName>&SWEExtCmd=<Execute>&UserName=<UserName>&Password=<Password>
```

Provide values for the parameters from the following table, which includes a name and description of each parameter.

Parameter	Definition
SWE URL	The base SWE URL. For example, <a href="http://web_server/eai_enu/start.swe">http://web_server/eai_enu/start.swe</a> where: <a href="#">web_server</a> Is the name of the Web server hosting Siebel SWE.
SWE External Command	The section within the eai.cfg file to execute, which is the [HTTP Services] section.
SWE External Command	Use <i>Execute</i> .
User Name	The user ID logon to execute.
Password	The logon password to execute.

For **MQ Series**, provide values for the parameters from the following table, which includes a name and description of each parameter.

Parameter	Definition
Queue Manager Name	The name of the queue manager to which the server must connect.
MQ server host for MQClient operation	The host on which the MQ Server is located (for the MQ Client only).
MQ server port for MQClient operation	The number to connect to an MQ Server queue manager (for the 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 (for the MQ client only). The default channel name for MQSeries is <a href="#">SYSTEM.DEF.SVRCONN</a>
Document type XML	Keep the default selection.
respqueue	Name of the queue where messages are placed.

**4.** Click *Continue*.

The Schemas pane opens.

5. To view a schema, click the corresponding tab.

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

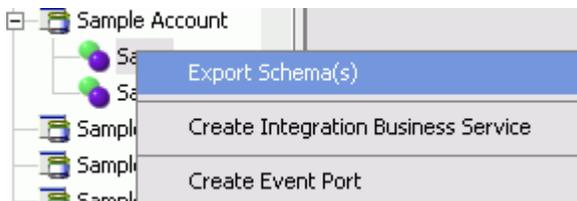
```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>Schema name: SiebelMessage</xsd:documentation>
  </xsd:annotation>
  <xsd:element name="SiebelMessage">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="ListOfSampleAccount" maxOccurs="1" minOccurs="0"/>
      </xsd:choice>
      <xsd:attribute name="MessageId"/>
      <xsd:attribute name="MessageType" use="required" fixed="Integration Object"/>
      <xsd:attribute name="IntObjectName" use="required" fixed="Sample Account"/>
      <xsd:attribute name="IntObjectFormat"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Organization">

```

A directory structure is created to store the schemas. For more information on where the schemas are stored, see *Schema Location* on page A-21.

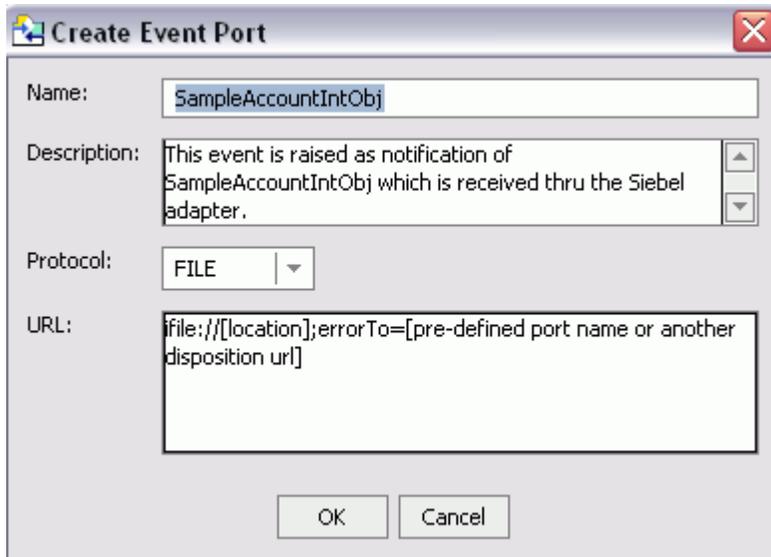
You are now ready to configure ports and channels for Siebel Integration Objects.

The following image shows a menu that appears when you right-click an object under the Sample Account node in the left pane. The following menu options appear: Export Schema(s), Create Integration Business Service, and Create Event Port.



6. To create an event port, right-click the event node name you just created and select *Create Event Port*.

The following image shows the Create Event Port dialog box that opens. There are three fields and one drop-down list where you add information to create an event port, and two action buttons.



For more information on creating event ports, see *Appendix B, Using Application Explorer in BEA WebLogic Workshop for Event Handling*.

## Reference: Schema Location

Application Explorer stores the schemas it creates in subdirectories under the iWay home directory of the machine where it is installed. The exact location of the schemas differs depending on whether you deploy Application Explorer with an iBSE or a JCA configuration.

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

`C:\Program Files\iway55\bea\ibse\wsdl\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. Request and response schema files are added to the folder.

- 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.

## Creating an Integration Business Service

---

You can create a business service (also known as a Web service) for Siebel objects you want to use with your adapter. After creating the Web service, you can export the WSDL file for the Web service to a directory accessible by BEA WebLogic Workshop, making it easy to incorporate iWay Web services into BEA WebLogic Workshop Workflows. For more information, see *Export iWay WSDL* on page A-26.

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

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

### Procedure: How to Create an Integration Business Service

**Note:** 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 a Schema for a Siebel Integration Object* on page A-13.

To create a business service for a Siebel Business Object:

1. If you are not connected, connect to your Siebel system.
2. Expand the Siebel node in which you are interested.

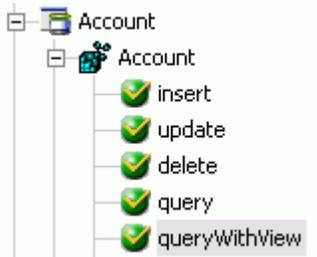
For a **Siebel Business Service** or **Integration Object**, expand the node in which you are interested and select the node for which you want to create a Integration Business Service.

**Note:** For an Integration Object, you must first create an IO node before you create a Integration Business Service. For more information, see *Creating a Schema for a Siebel Integration Object* on page A-13.

For a **Business Component**, expand the Business Component for which you want to create an Integration Business Service and select a node.

3. Expand the Business Component for which you want to create a Web service.

The following image shows the `queryWithView` method selected beneath the Account object listed beneath the Account node in the left pane.

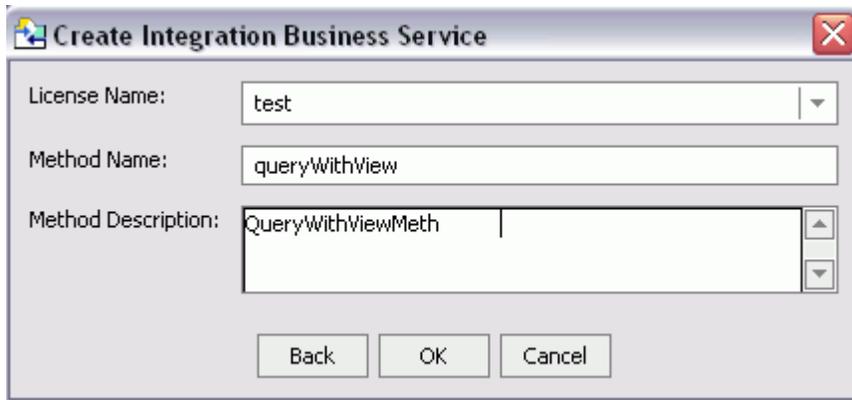


4. Expand the object, for example, Account.
  - a. Under Account, select a method for creating the business service, for example, `queryWithView`.
  - b. Right-click the method, and select *Create Integration Business Service*.

The Create Integration Business Service dialog box appears in the right pane.
  - c. From the Existing Service Names drop-down list, select either a new service or an existing service.
  - d. If you are creating a new service, specify a service name.

This name identifies the Web service in the list of services under the Integration Business Services node.
  - e. In the Method Description field, type a brief description of the service.
5. Click *Next*.

The following image shows the Create Integration Business Service dialog box that opens where you add information to create a Business Service.



- a. In the License Name drop-down list, 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.
  - b. 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.
  - c. In the Method Description field, type a brief description of the method.
6. Click *OK*.

Application Explorer expands the Integration Business Services node in the left pane to show the new Web service and presents a test input area in the right pane.

## Testing a Web Service for a Business Object or Integration Object

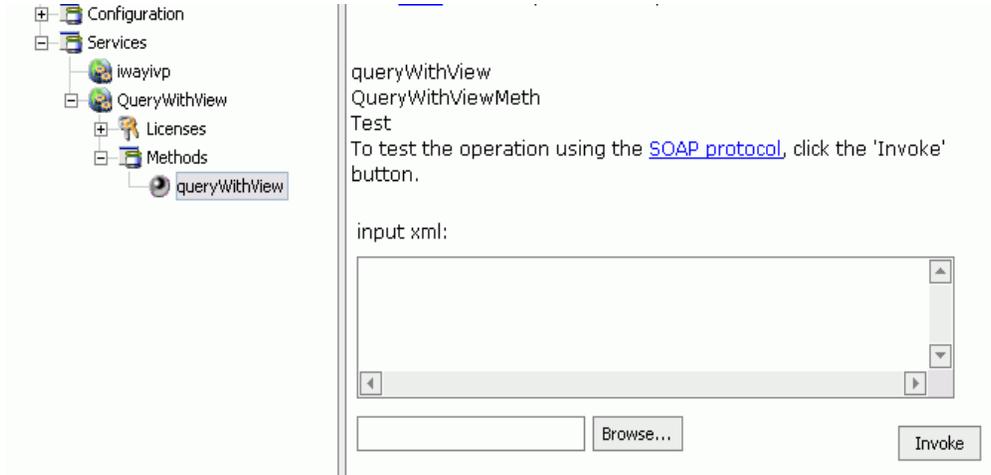
After you create a Web service for the Siebel Business Object, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

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

To test a Web service for a Business Object:

1. In the left pane of Application Explorer, expand the *Integration Business Services* node.
2. Expand the *Services* node.
3. Select the name of the business service you want to test.
4. Expand the *Methods* node under the service and select the method you want to test.

The following image shows the navigation pane on the left with the queryWithView method selected. In the right pane is a description of the method and a field where you can input XML to test the service.



5. Provide the appropriate XML input in the input xml field.
6. Click *Invoke*.

The following image shows the XML results that Application Explorer displays in the right pane.

```
?xml version="1.0" encoding="UTF-8" ?> SIEBEL1 ACCOUNTONE SIEBEL2 ACCOUNTTWO
SIEBEL3RR SIEBEL4FOUR
```

## Testing a Web Service for a Business Service

After a Web service is created 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 it is not expanded, expand the *Integration Business Services* node.
2. Expand the *Services* node.
3. Expand the name of the Business Service you want to test.
4. Expand the *Methods* node and select the name of the method you want to test.

The following image shows the test option that appears in the right pane with a description of how to test the method. Beneath the description is a table with fields where you provide values for the Account and Site parameters.

Parameter	Value
Account:	<input type="text"/>
Site:	<input type="text"/>

Invoke

5. Provide the appropriate input.
6. Click *Invoke*.

Application Explorer displays the results in the results window in the right pane.

## Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows

Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate iWay Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use iWay Web services, you simply export the WSDL to a directory accessible to BEA WebLogic Workshop.

### Procedure: How to Export iWay WSDL

To export iWay WSDL:

1. After you create a Web service, right-click the Web service name and select *Export WSDL*.

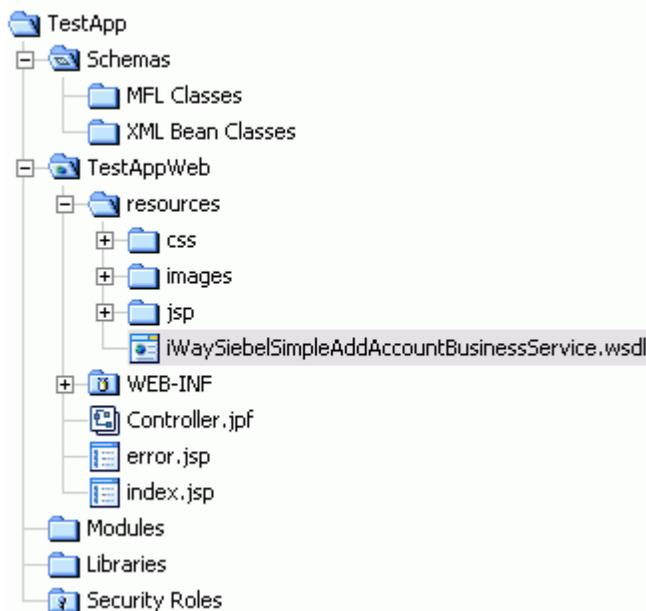
The following image shows the menu that appears when you right-click the service name in the left pane.



The Save dialog box appears.

2. Save the WSDL to a directory accessible to BEA WebLogic Workshop, for example, the \resources directory in your BEA WebLogic Workshop Web application directory structure.

The following image shows a WSDL file as it appears beneath the resources folder of your Web application.



## 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, since they are not required:

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

## Adding a Control for an iWay Resource in BEA WebLogic Workshop

---

Java controls provide a convenient way to incorporate access to iWay resources. You can add controls in BEA WebLogic Workshop to use Web services created by Application Explorer, or you can add controls that enable you to take advantage of the JCA resources of Application Explorer.

### Adding a Web Service Control to a BEA WebLogic Workshop Application

After you create an iWay Web service using Application Explorer and export the WSDL file, you can create a control for the Web service.

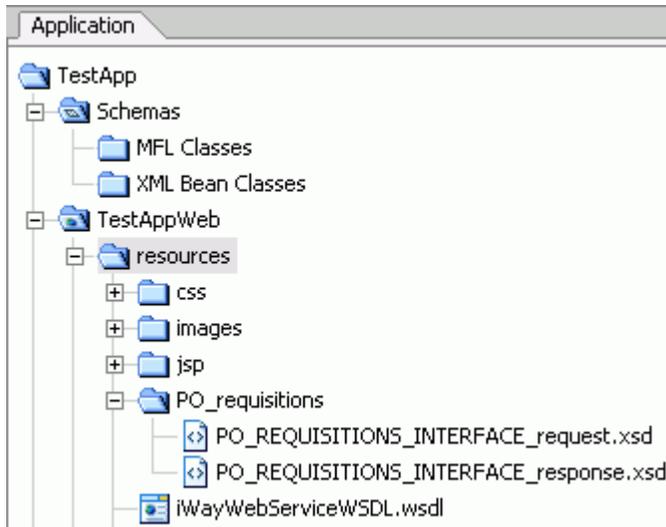
For more information on exporting a WSDL file, see *Export iWay WSDL* on page A-26.

#### Procedure: How to Add a Web Service Control

To add a Web service control:

1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

The following image shows a WSDL file as it appears in the Application tab beneath the resources folder of your Web application.



2. Right-click the *WSDL* file and select *Generate Service Control*.

The following image shows the control for the WSDL that appears beneath the selected WSDL file in the resources folder.



## Adding an Extensible CCI Control to a BEA WebLogic Workshop Application

The following section describes the enhanced CCI control, which is extensible and provides JCX with typed inputs and outputs for JCA in BEA WebLogic Workshop.

### Overview

The extensible iWay CCI control provides:

- **Method and tag validation.** BEA WebLogic Workshop provides warnings regarding invalid methods and tags.
- **Improved error handling.**

You can define new methods that rely on the generic *service* and *authService* methods. For example, you can define a JCX with a new method without writing casting code or explicit transformations such as the following:

## Adding an Extensible CCI Control to a BEA WebLogic Workshop Application

```
public SiebelResponseDocument SiebJCA(SiebelDocument aRequest) throws  
Exception;
```

In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control* on page A-30.

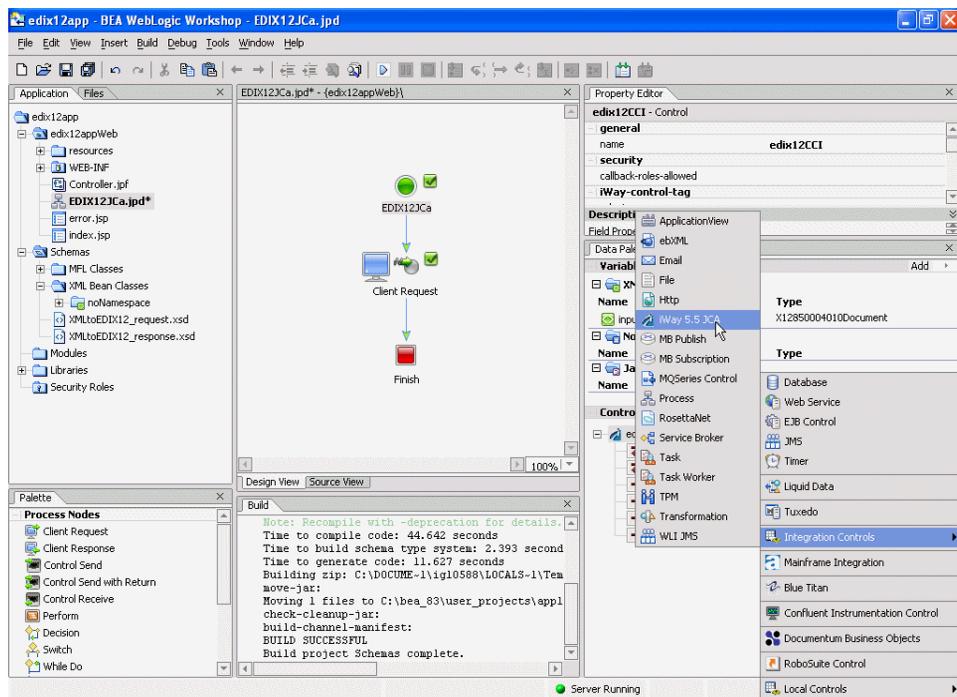
You can also use dynamic class casting to specify schema-based input or output XmlObjects to be casted into a pure XmlObject as a service method, which is expected by the CCI control. For more information, see *Using Dynamic Class Casting* on page A-36.

### Example: Defining a Control Using the Extensible CCI Control

The following sample JCX demonstrates how to define a control for Siebel using the extensible CCI control in BEA WebLogic Workshop.

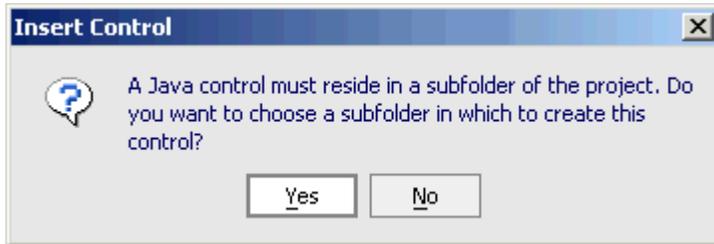
1. Start BEA WebLogic Workshop and create a new project.

The following image shows the BEA WebLogic Workshop window. The Application tab is active in the upper left pane and the Palette tab appears in the lower left with a list of process nodes. The upper middle pane shows a design in the Design View tab and the lower middle pane contains the Build tab. The Property Editor tab is open in the upper right and the Data Palette tab is open in the lower right.



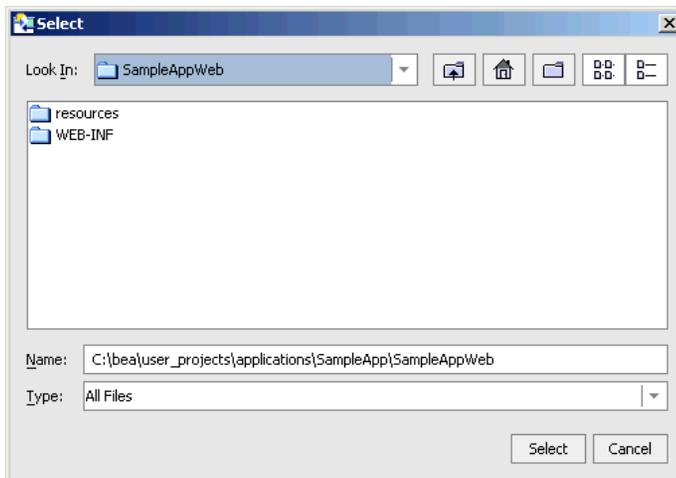
2. Click *Add* from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

The following image shows the Insert Control message box that opens, prompting you to select a subfolder where the control will reside.



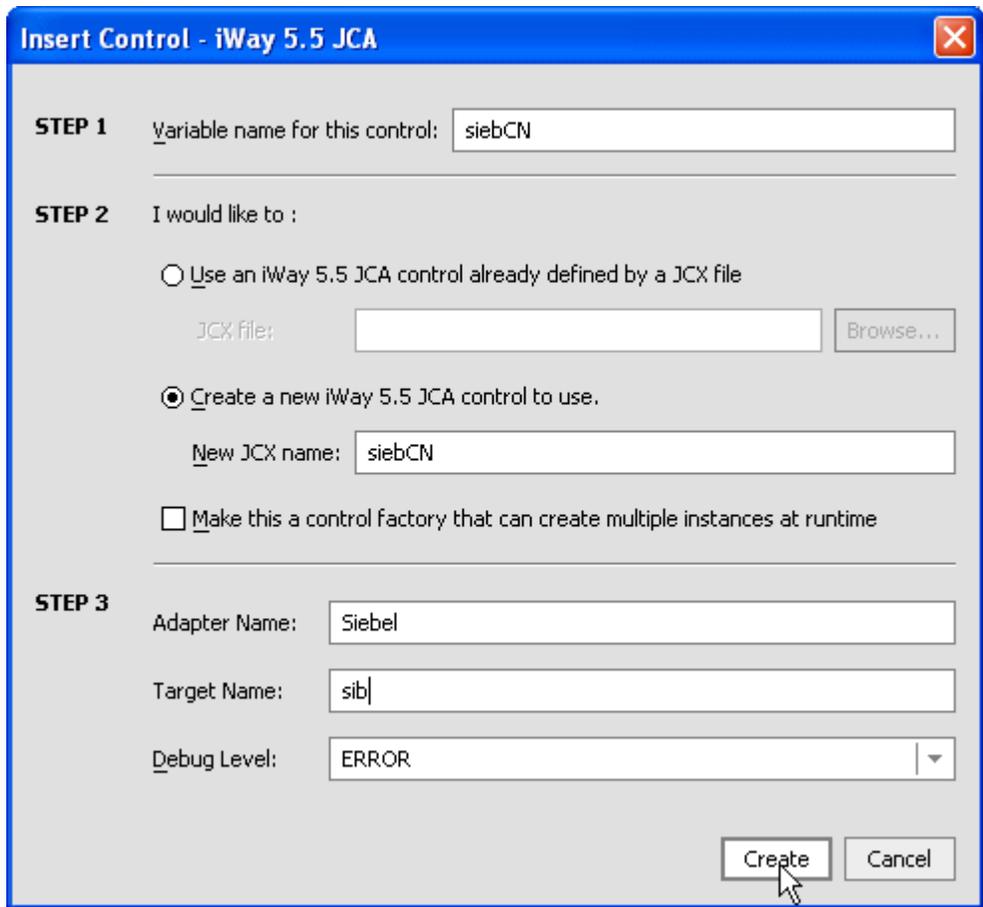
3. Click *Yes*.

The following image shows the Select dialog box that opens where you browse to locate a subfolder or type the name of a directory into a field.



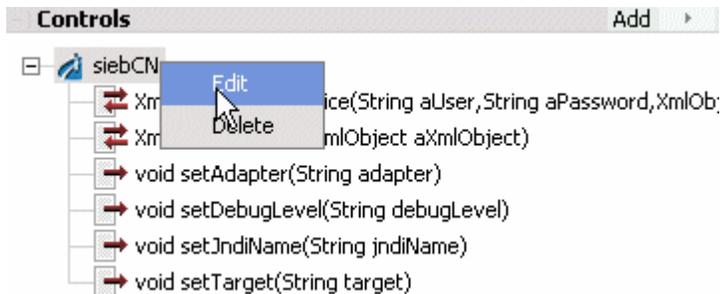
4. Choose a subfolder for the CCI control and click *Select*.

The following image shows the Insert Control - iWay 5.5 JCA dialog box that opens. STEP 1 in the upper pane has a field for providing a variable name for the control. STEP 2 in the middle pane has option buttons where you select a predefined control or create a new one, and fields for selecting a control or naming a new one. There is a check box in the middle pane that you can select to make the control create multiple instances at run time. STEP 3 has two fields and a drop-down list where you provide target information.



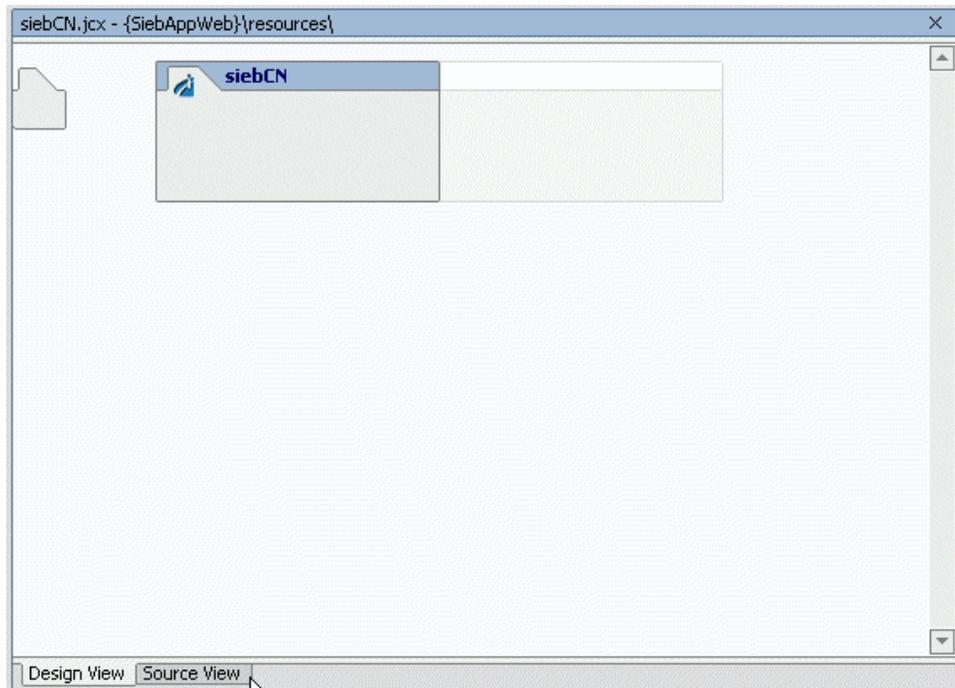
- a. Provide a variable name for the control.
  - b. Click *Create a new iWay 5.5 JCA control to use* and provide a new JCX name.
  - c. Type the adapter name, target name, and select a debug level from the drop-down list.
5. Click *Create*.

The following image shows the new JCX file that is created beneath the Controls section in the Data Palette tab.



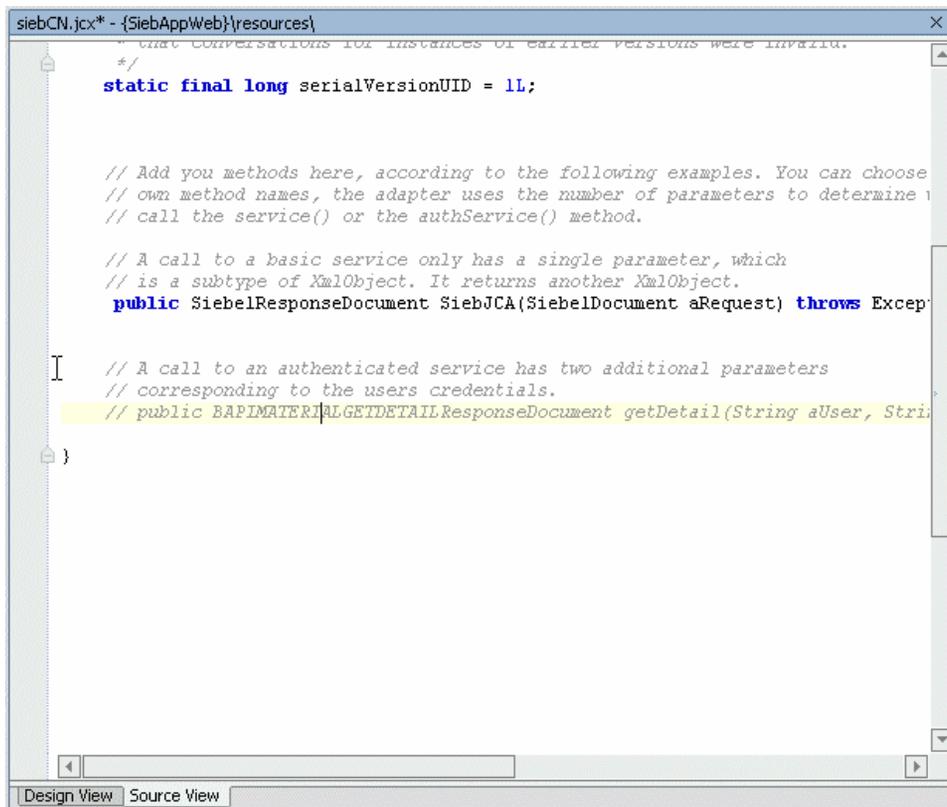
6. Right-click the control, for example, siebCN, and select *Edit*.

The following image shows the design view for the control that opens in the active Design View tab. The Source View tab is next to Design View in the lower left of the pane.



7. Click the *Source View* tab.

The following image shows the XML source view for the control that opens in the active Source View tab. The Design View tab is next to Source View in the lower left of the pane.



```
siebCN.jcx* - {SiebAppWeb}\resources
    * that Computations for instances of earlier versions were invalid.
    */
    static final long serialVersionUID = 1L;

    // Add you methods here, according to the following examples. You can choose
    // own method names, the adapter uses the number of parameters to determine
    // call the service() or the authService() method.

    // A call to a basic service only has a single parameter, which
    // is a subtype of XmlObject. It returns another XmlObject.
    public SiebelResponseDocument SiebJCA(SiebelDocument aRequest) throws Excepti

    // A call to an authenticated service has two additional parameters
    // corresponding to the users credentials.
    // public BAPIMATERIALGETDETAILResponseDocument getDetail(String aUser, Stri
}

Design View Source View
```

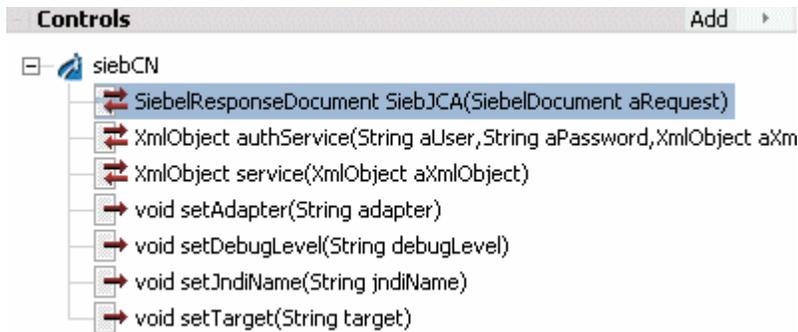
Perform the following steps:

- a. Uncomment the public class definition.
- b. Change the existing response data type to match your response data type according to the Siebel schema, for example, SiebelResponseDocument.
- c. Change the existing method name to match your method, for example, SiebJCA.
- d. Change the existing incoming data type to match your request data type according to the Siebel schema, for example, SiebelDocument.

The following syntax provides a sample Siebel method call, which calls the adapter service:

```
public SiebelResponseDocument SiebJCA(SiebelDocument aRequest) throws
Exception;
```

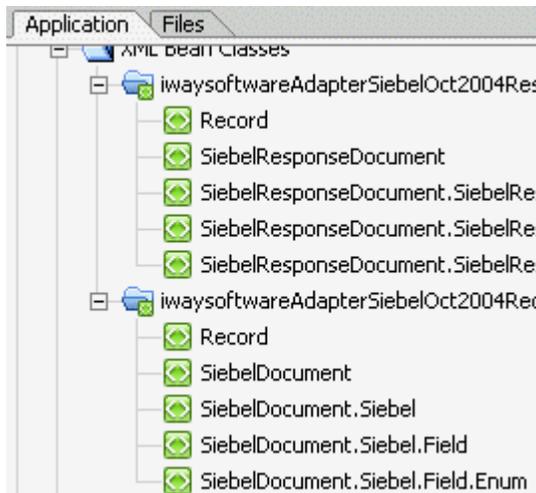
The following image shows the control that is now available beneath the Controls section in the Data Palette tab and can be added to a workflow.



**Note:** You can view available data types beneath the XML Bean Classes folder in the *Application* tab, which are added once you import your XML request or response schemas from Application Explorer.

These data types are case-sensitive and must be entered exactly as shown.

The following image shows data types available beneath the XML Bean Classes folder in the *Application* tab. The *Files* tab is next to the *Application* tab in the upper left of the pane.



## Using the Extensible CCI Control

The extensible CCI control functions much like a database control since it generates JCX files to which you can add your own methods.

Your own methods can use the correct input and output types rather than the generic XmlObject types that the JCA control uses. Since the control is just a proxy that uses a reflection to call the relevant method, it handles the casting for you. You are no longer required to write custom code that does the cast or transformations that are cast between an XmlObject.

For example, instead of the generic XmlObject

```
XmlObject service(XmlObject input) throws java.lang.Exception;
```

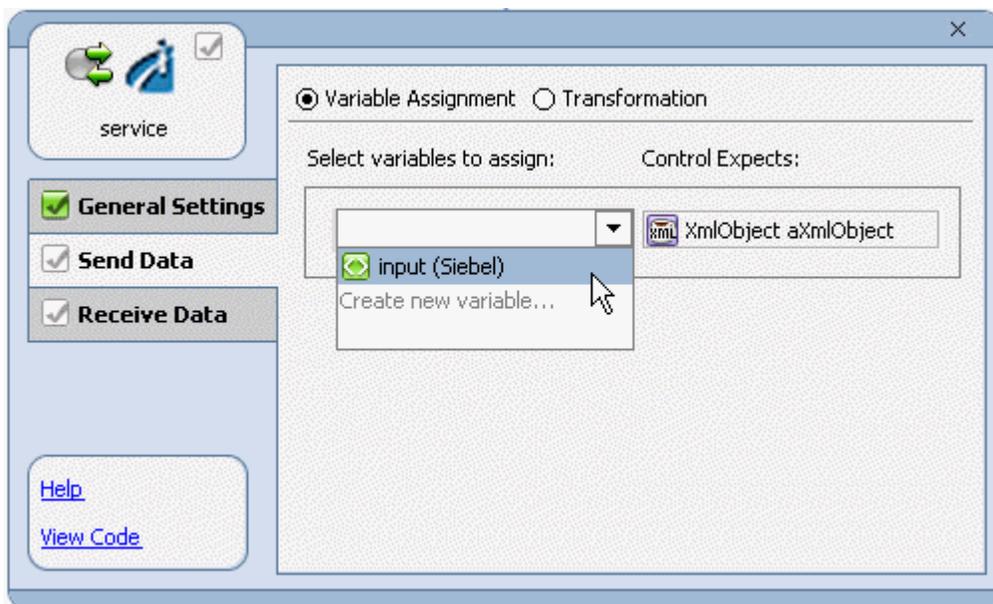
you call:

```
public SiebelResponseDocument SiebJCA(SiebelDocument aRequest) throws  
Exception;
```

### Example: Using Dynamic Class Casting

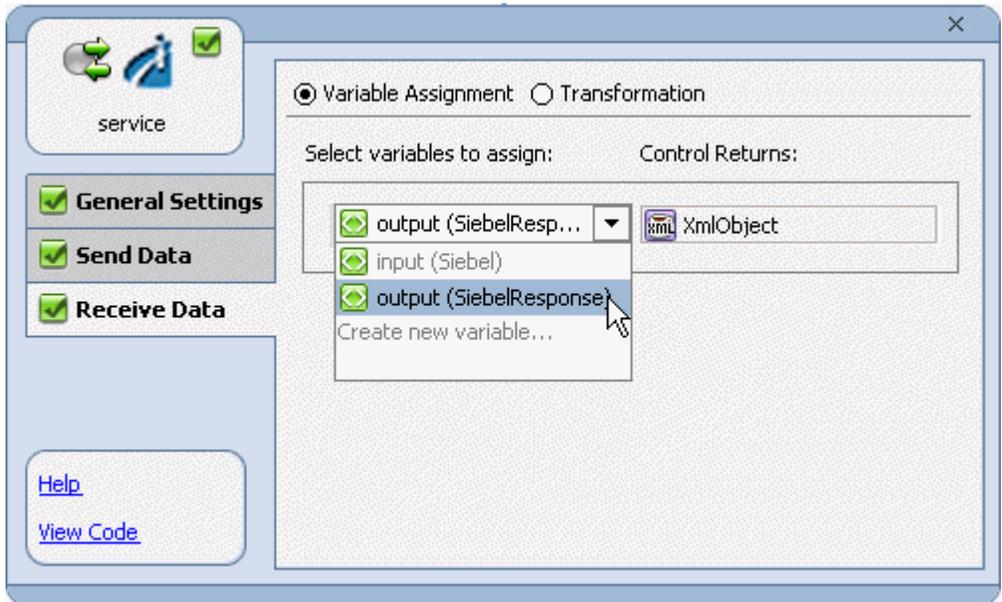
The following example uses dynamic class casting to specify a schema-based input XmlObject to be casted into a pure XmlObject as a service method, which is expected by the CCI control.

The following image shows



The following example uses dynamic class casting where the CCI control returns a pure XmlObject, which is casted dynamically into a schema-based output XmlObject.

The following image shows





---

---

## APPENDIX B

# Using Application Explorer in BEA WebLogic Workshop for Event Handling

### Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Understanding Event Functionality
- Listening for a Siebel Integration Object Event
- Creating an Event Port
- Modifying an Event Port
- Creating a Channel
- Modifying a Channel
- Deploying iWay Components in a Clustered BEA WebLogic Environment

This section describes how to use the Java Swing implementation of Application Explorer as deployed in BEA WebLogic Workshop. Application Explorer deployed in WebLogic Workshop is functionally similar to the Servlet Application Explorer.

For more information on managing failover and load balancing in a clustered environment, see the BEA clustering documentation.

## Starting Application Explorer in BEA WebLogic Workshop

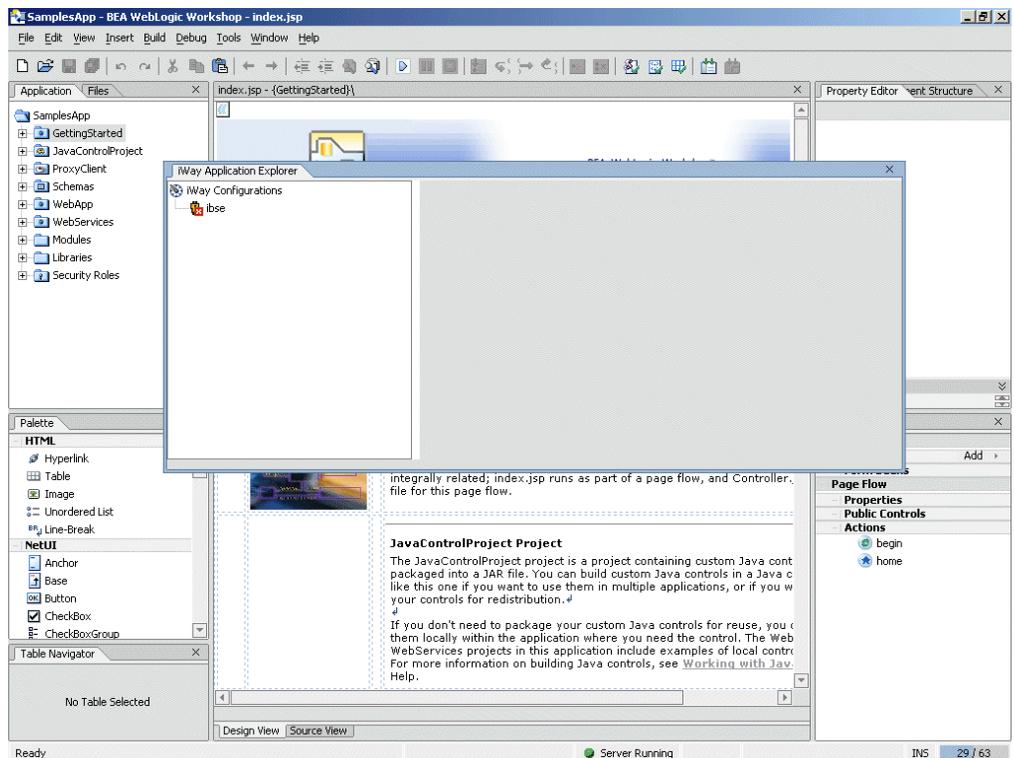
The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

### Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer:

1. Ensure the server on which Application Explorer is deployed is started.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

The following image shows the Application Explorer window that opens, with the BEA WebLogic Workshop window in the background. On the left of the Application Explorer window is a navigation pane showing configurations. The right pane is empty.



You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

## 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 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 Application Explorer. To create an event, you must create a port and a channel.

- 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 B-4.

- 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 B-15.

## 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.

You use the XDR schema 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 A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Integration 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 B-3.

### Creating an Event Port For a Siebel Integration Object

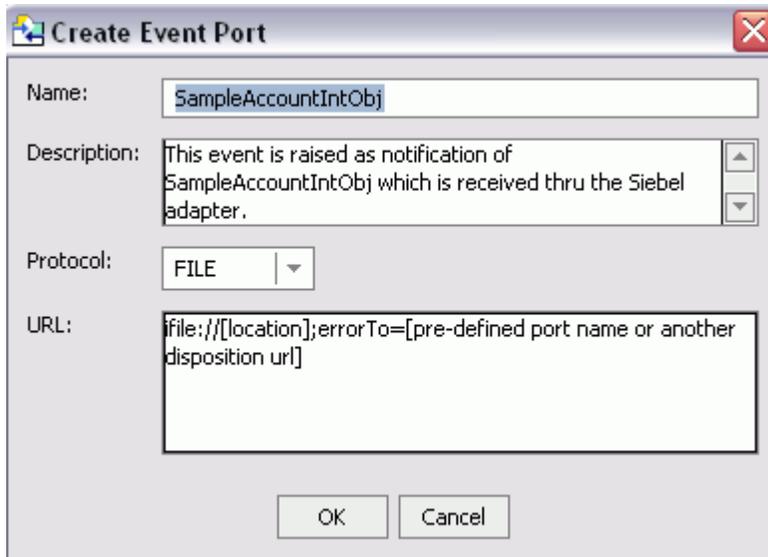
For Siebel Integration Objects, you can create an event port directly by selecting the object from the list of Integration Objects.

#### Procedure: How to Create an Event Port for an Integration Object

To create an event port from the Service Adapters tab:

1. Select the Integration Object event you created.
2. Right-click the event and select *Create Event Port*.

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



The screenshot shows a dialog box titled "Create Event Port" with a close button in the top right corner. The dialog contains the following fields and controls:

- Name:** A text input field containing "SampleAccountIntObj".
- Description:** A text area containing "This event is raised as notification of SampleAccountIntObj which is received thru the Siebel adapter." with scroll arrows on the right.
- Protocol:** A dropdown menu currently showing "FILE".
- URL:** A text area containing "file://[location];errorTo=[pre-defined port name or another disposition url]".
- Buttons:** "OK" and "Cancel" buttons at the bottom.

- a. In the Name field, type a name for the event port.
- b. In the Description field, provide a brief description.
- c. From the drop-down list, select the required disposition, for example, File.
- d. In the URL field, type the disposition URL.

3. Click *OK*.

For more information on specifying dispositions, see *Creating an Event Port From the Event Adapters Node*.

## Creating an Event Port From the Event Adapters Node

The following procedures describe how to create an event port from the Event Adapters window for various dispositions using Java Swing Application Explorer.

The following dispositions are available when using Application Explorer in conjunction with an iBSE deployment:

- File
- iBSE
- MSMQ
- JMSQ
- SOAP
- HTTP
- MQ Series

**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 deployment.

- File
- JMSQ
- HTTP
- 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 B-3.

For Integration Objects, you also can create an event port directly from the list of Integration Objects. For more information, see *Creating an Event Port For a Siebel Integration Object* on page B-4.

## Procedure: How to Create an Event Port for File

To create a specific event port for File:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- a. Type a name for the event port and provide a brief description.
- b. From the drop-down list, select the *File* disposition.
- c. From the Disposition Protocol drop-down list, select *File*.
- d. In the URL field, provide a destination where the event data will be 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.

To create a File disposition, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
location	The full directory path and file name to which the data is written.
errorDest	The location where error logs are sent. Optional. Can be a predefined port name or another disposition URL. The URL must be complete, including the protocol.

4. Click *OK*.

In the left pane, the event port appears below the ports node. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-15.

### Procedure: How to Create an Event Port for iBSE

You can call an Integration Business Service created through Integration Business Services Engine (iBSE).

To create an event port for iBSE:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- 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]
```

To create an iBSE disposition, provide values from the following table, which includes a 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.

4. Click *OK*.

In the left pane, the event port appears below the ports node. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-15.

### Procedure: How to Create an Event Port for JMSQ

To create an event port for a JMS queue:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- 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 URL 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=;
```

To create a JMS disposition, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
myQueueName or jmsqueue	JNDI name of a queue to which events are emitted.
myQueueFac or jmsfactory	Resource that contains information about the JMS Server. The WebLogic connection factory is: <code>javax.jms.QueueConnectionFactory</code>

Parameter	Description
jndiurl	<p>URL to use to contact the JNDI provider. The syntax of this URL depends on the JNDI provider being used. This value corresponds to the standard JNDI property,</p> <p><code>java.naming.provider.url.</code></p> <p>The URL of the WebLogic Server is</p> <p><code>t3://host:port</code></p> <p>where:</p> <p><code>host</code></p> <p>Is the machine name where WebLogic Server is installed.</p> <p><code>port</code></p> <p>Is the port on which WebLogic Server is listening. The default port, if not changed at installation, is 7001.</p>
jndifactory	<p>Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider.</p> <p>For WebLogic Server, the WebLogic factory is:</p> <p><code>weblogic.jndi.WLInitialContextFactory.</code></p>
user	Valid user name required to access a JMS server.
password	Valid password required to access a JMS server.
errorTo	Predefined port name or another disposition URL to which error logs are sent. Optional.

4. Click OK.

In the left pane, the event port appears below the ports node. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

You are now ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-15.

### Procedure: How to Create an Event Port for MSMQ

To create an event port for MSMQ:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.

## Creating an Event Port

3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- 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]
```

To create an MSMQ disposition, provide values from the following table, which includes a 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 <i>Private\$</i> .
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.

4. Click *OK*.

In the left pane, the event port appears below the ports node. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-15.

## Procedure: How to Create a Port for the SOAP Disposition

To create a port for a SOAP disposition:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- a. Type a name for the event port and provide a brief description.
- b. From the Disposition Protocol drop-down list, select *SOAP*.
- c. In the URL field, enter an 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]
```

To create a SOAP disposition, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
wSDL-url	<p>The URL to the WSDL file that is required to create the SOAP message. For example:</p> <pre>http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl</pre> <p>where:</p> <pre>webservice</pre> <p>Is the name of the Web service you created using Application Explorer.</p> <p>This value can be found by navigating to the Integration Business Services tab and opening the <i>Service Description</i> link in a new window. The WSDL URL appears in the Address field.</p> <p>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.</p>
soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.

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.

4. 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 HTTP Disposition

To create an event port for HTTP disposition:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- a. Type an event port name and a brief description.
- b. From the disposition protocol drop-down list, select *HTTP*.
- c. In the URL 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];
```

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

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

To create an HTTP disposition when using an **IBSE** deployment, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
myurl	URL target for the post operation, for example, <a href="http://myhost:1234/docroot">http://myhost:1234/docroot</a>
responseTo	Predefined port name or another disposition URL to which response documents are sent. Optional.

To create an HTTP disposition when using a **JCA** deployment, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
host:port	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	Universal resource identifier that completes the URL specification.

**4.** Click *OK*.

In the left pane, the event port appears below the ports node. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

### Procedure: How to Create an Event Port for an MQ Series Disposition

To create an event port for an MQ Series disposition:

- 1.** Expand the *Event Adapters* node.
- 2.** Expand the *Siebel* node.
- 3.** Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box appears.

- a.** Type an event port name and a brief description.
- b.** From the disposition protocol drop-down list, select *MQ Series*.

## Creating an Event Port

- c. In the URL field, enter an MQ Series destination.

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

```
mqseries:/[qManager]/[qName];host=[hostname];port=[port];  
channel=[channelname];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=
```

To create an MQ Series disposition, provide values from the following table, which includes a name and description of each parameter.

:

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	Host on which the MQ Server is located (for the MQ Client only).
port	Port number to connect to an MQ Server queue manager (for the MQ client only).
channel	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (for the MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
errorTo	Predefined port name or another disposition URL to which error logs are sent. Optional.

4. Click *OK*.

In the left pane, the newly created event port appears under the port section of the event adapter. To review the port settings, select the port name. A table summarizing the port settings appears in the right pane.

## Modifying an Event Port

---

To review the port settings, select the port name. A table summarizing the port settings appears in the right pane. The following procedures describe how to modify and delete an event port.

### Procedure: How to Edit an Event Port

To edit an existing event port:

1. Right-click the event port you want to edit and select *Edit*.  
The Edit Port pane opens.
2. Make the required changes and click *OK*.

### Procedure: How to Delete an Event Port

To delete an event port:

1. To view the available ports, click the *ports* node.
2. Right-click the event port you want to delete and select *Delete*.  
The event port disappears from the list in the left pane.

## Creating a Channel

---

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

- HTTP
- File
- MQ Series

### Procedure: How to Create an HTTP Channel

To create an HTTP channel:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Channels* node and select *Add Channel*.  
The Add Channel dialog box opens.
  - a. Type a name for the channel, for example, NewChannel.
  - b. Type a brief description.

## Creating a Channel

- c. From the Protocol drop-down list, select *HTTP Listener*.
  - d. Select an event port from the list of current ports.
  - e. To transfer the port to the list of available ports, click the double right (>>) arrow. To associate all the event ports, control-click to select each port or click one port and press *Control+A*. Then, click the double right (>>) arrow button.
4. Click *Next*.

The Basic dialog box appears.

To create an HTTP channel, provide values from the following table, which includes a name and description of each parameter.

Parameter	Description
www root	Root directory where Siebel event data is posted.
Listener port	Port on which to listen for Siebel event data.
Synchronization Type	Choose from the following three options: REQUEST, REQUEST_RESPONSE, or REQUEST_ACK.

5. Click *OK*.

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.



- a. To review the settings for the channel, select the channel.  
In the right pane, Detail and HTTP Listener tabs summarize the channel information. All the information is associated with the channel you created.
- b. To start the channel, right-click the channel node you just created and select *Start*.  
The channel you created becomes active.  
The X that was over the icon disappears.
- c. To stop the channel, right-click the channel node and select *Stop*.

## Procedure: How to Create a File Channel

To create a File channel:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Channels* node and select *Add Channel*.

The Add Channel dialog box opens.

- a. Type a name for the channel, for example, *NewChannel*.
  - b. Type a brief description.
  - c. From the Protocol drop-down list, select *File Listener*.
  - d. Select an event port from the list of current ports.
  - e. To transfer the port to the list of available ports, click the double right (>>) arrow. To associate all the event ports, control-click to select each port or click one port and press *Control+A*. Then, click the double right (>>) arrow button.
4. Click *Next*.

The File Listener dialog box opens with three tabs in the right pane.

- a. In the Request tab, provide values for the parameters from the following table, which includes a name and description of each parameter.

Parameter	Description
Polling Location	Target file system location for the Siebel XML file.
File Mask	File name to use 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, which includes a name and description of each parameter.

Parameter	Description
Synchronization Type	Target file system location for the Siebel XML file.
Response/Ack Directory	Choose from the following three options: REQUEST, REQUEST_RESPONSE, or REQUEST_ACK.

## Creating a Channel

- c. In the Advanced tab, provide values for the parameters from the following table, which includes a 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.
Processing Mode	Choose Sequential or Threaded. <ul style="list-style-type: none"><li>• Sequential indicates single processing of requests.</li><li>• Threaded indicates processing of multiple requests simultaneously.</li></ul>
Thread limit	If you selected threaded processing, indicate the maximum number of requests that can be processed simultaneously.

5. Click *OK*.

In the left pane, the new channel appears in the list of channels. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

- a. To review the settings for the channel, select the channel.

In the right pane, the Detail, Request, Response, and Advanced tabs summarize the channel information. All the information is associated with the channel you created.

- b. To start the channel, right-click the channel node you just created and select *Start*.

The channel you created becomes active.

The X that was over the icon disappears.

- c. To stop the channel, right-click the channel node and select *Stop*.

### Procedure: How to Create a MQ Series Channel

To create a MQ Series channel:

1. Expand the *Event Adapters* node.
2. Expand the *Siebel* node.
3. Right-click the *Channels* node and select *Add Channel*.

The *Add Channel* dialog box 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*.
  - d. Select an event port from the list of current ports.
  - e. To transfer the port to the list of available ports, click the double right (>>) arrow. To associate all the event ports, control-click to select each port or click one port and press *Control+A*. Then, click the double right (>>) arrow button.
4. Click *Next*.

The MQ Listener dialog box appears with three tabs in the right pane.

- a. In the Request tab, provide values for the parameters from the following table, which includes a 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).
MQ server port for MQClient operation	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.

Parameter	Description
Request queue name	<p>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:</p> <p><i>Host\queue type\$qName</i></p> <p>where:</p> <p><i>Host</i></p> <p>Is the machine name where the MQ Series queuing system is running.</p> <p><i>queue type</i></p> <p>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.</p> <p><i>qName</i></p> <p>Is the name of the queue where messages are placed, for example,</p> <p><i>iwaykxcl\Private\$\siebel</i></p>

- b. In the Response tab, provide values for the parameters from the following table, which includes a name and description of each parameter.

Parameter	Description
Synchronization Type	Choose from the following three options: REQUEST, REQUEST_RESPONSE, or REQUEST_ACK.

- c. In the Advanced tab, provide values for the parameters from the following table, which includes a 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.

Parameter	Description
Mode of operation	Choose Sequential or Threaded. <ul style="list-style-type: none"> <li>Sequential indicates single processing of requests.</li> <li>Threaded indicates processing of multiple requests simultaneously.</li> </ul>
Thread limit	If you selected threaded processing, indicate the maximum number of requests that can be processed simultaneously.

5. Click *OK*.

In the left pane, the new channel appears in the list of channels. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

a. To review the settings for the channel, select the channel.

In the right pane, the Detail, Request, Response, and Advanced tabs summarize the channel information. All the information is associated with the channel you created.

b. To start the channel, right-click the channel node you just created and select *Start*.

The channel you created becomes active.

The X that was over the icon disappears.

c. To stop the channel, right-click the channel node and select *Stop*.

## Modifying a Channel

---

The following procedures describe how to edit and delete a channel using Application Explorer. To review the channel settings, you select the channel name. In the right pane, a table appears that summarizes the information associated with the channel you created.

### Procedure: How to Edit a Channel

To edit a channel:

1. In the left pane, right-click the channel you want to edit and select *Edit*.

The Edit Channel dialog box appears.

2. Make the required changes to the channel configuration and click *OK*.

## Procedure: How to Delete a Channel

To delete a channel:

1. Right-click the channel you want to delete.
2. Select *Delete*.

The channel disappears from the list in the left pane.

## Deploying iWay Components in a Clustered BEA WebLogic Environment

---

Events can be configured in a clustered BEA WebLogic environment. You can deploy iBSE or JCA to this environment. This topic uses iBSE as an example, but you can follow the same procedures when deploying JCA. The only difference is that you need to deploy the JCA connector .RAR file to the clustered environment.

A cluster consists of multiple server instances running simultaneously, yet appears to clients to be a single server instance. The server instances that contain a cluster can be run on one machine, but are usually run on multiple machines.

Clustering provides the following benefits:

- Load balancing. For more information, see the BEA clustering documentation.
- High availability

Service requests are processed through the HTTP router and routed to an available managed server.

Events are server-specific and are not processed through the HTTP router. You must configure each server separately.

For more information, see *How to Configure Application Explorer Communications* in the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

## Procedure: How to Deploy iWay Components in a Clustered Environment

To deploy iWay components in a clustered environment:

1. Using the BEA Configuration Wizard:
  - a. Configure an administrative server to manage the managed servers.
  - b. Add and configure as many managed servers as required.
  - c. Add and configure an HTTP router. This does not have to be a part of WebLogic and can be an outside component.
  - d. If you configure the HTTP router within WebLogic, start it by entering the following command:

`StartManagedWebLogic HTTPROUTER http://localhost:7001`

where:

`HTTPROUTER`

Is the name of the server on which the HTTP router is running.

`http://localhost:7001`

Is the location of the admin console.

- e. Add the managed servers to your cluster/clusters.

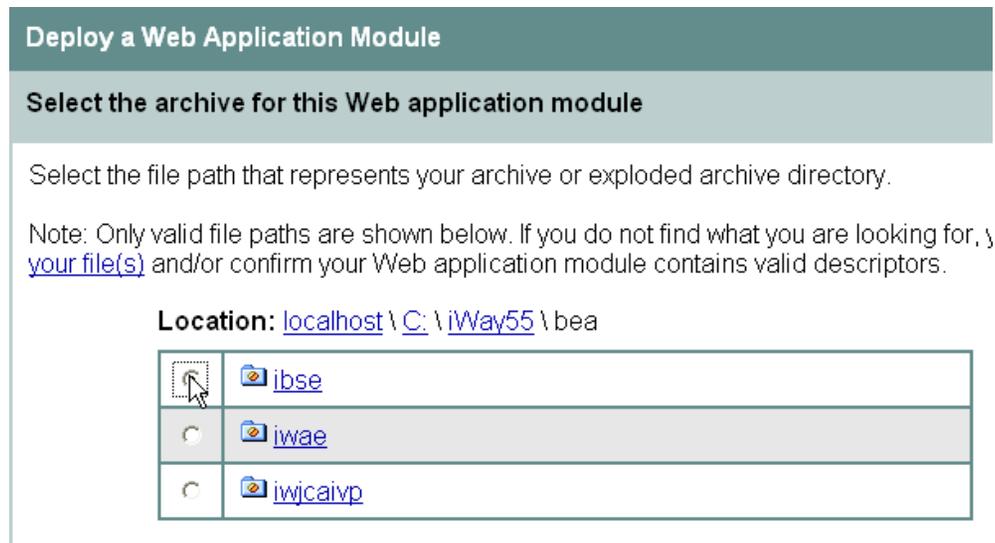
For more information on configuring WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

2. Start the WebLogic Server and open WebLogic Server Console.
3. Deploy iBSE to the cluster by selecting *Web Application Modules* from the Domain Configurations section, and clicking *Deploy a new Web Application Module*.

A page appears for you to specify where the Web application is located.

4. To deploy iBSE, select the option button next to the *ibse* directory and then click *Target Module*.

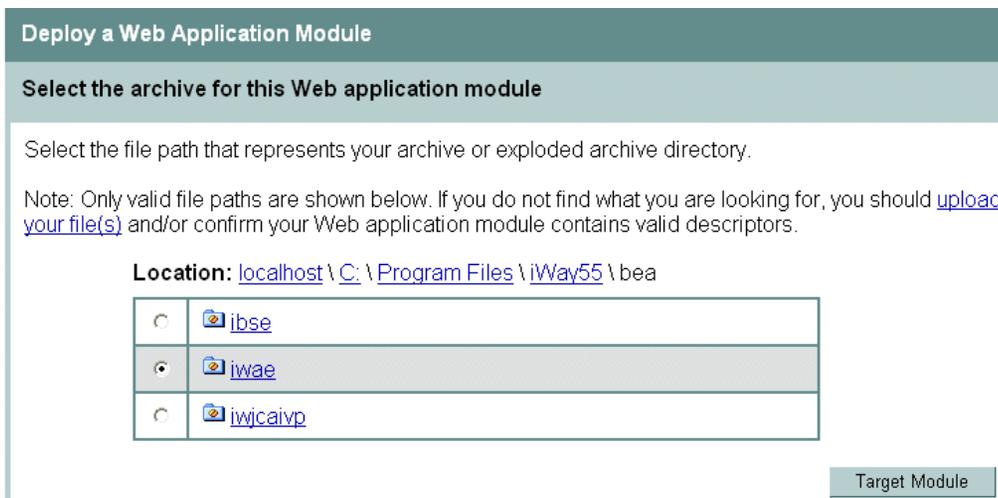
The following image shows the Deploy a Web Application Module window where you select an archive for the Web Application module. It provides the file path of your archive and a table where you select an option button for the file path you want to use.



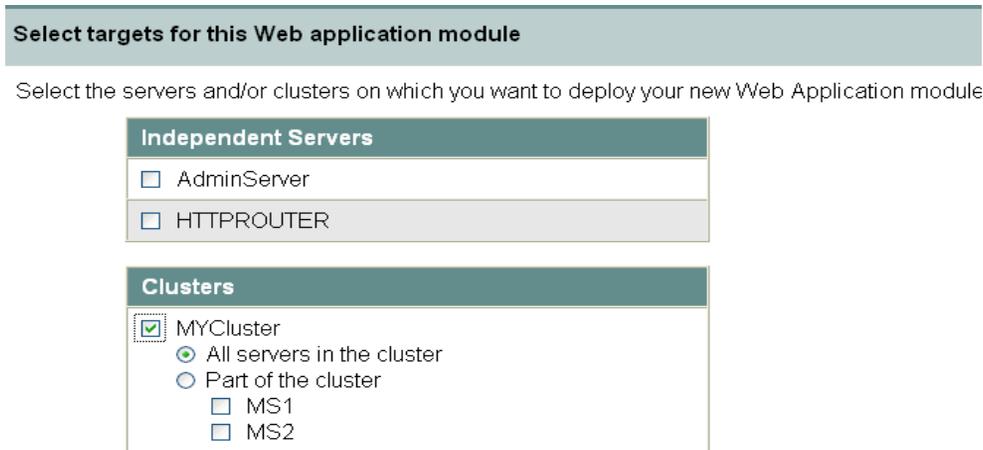
5. To deploy servlet Application Explorer, select the option button next to the iwaec directory and then click *Target Module*.

If you are using servlet Application Explorer, deploy it only on the admin server or one of the managed servers.

The following image shows the Deploy a Web Application Module window where you select an archive for the Web Application module. It provides the file path of your archive and a table where you select an option button for the file path you want to use. Beneath the table is the Target Module button.

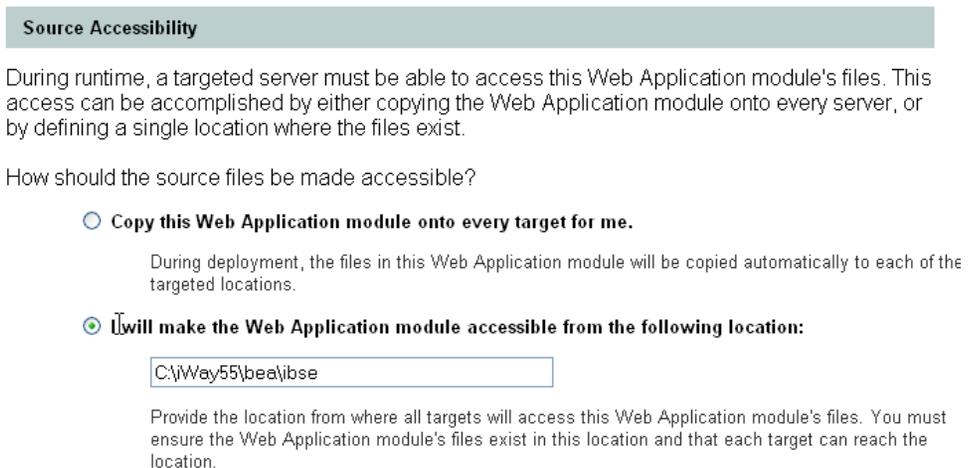


The following image shows the Select targets for this Web application module window that opens with a description of how to select your targets. The Independent Servers pane contains two check boxes and the Clusters pane contains check boxes and option buttons.



6. Select the servers and/or clusters on which you want to deploy the application and click *Continue*.

The following image shows the Source Accessibility window that opens with a description of how a server will access the Web Application module source files and two option buttons where you select a method for accessing the files.



7. Select the *I will make the Web Application module accessible from the following location* option button and provide the location from which all targets will access iBSE.

iWay Software recommends that you use a single instance of iBSE, rather than copying iBSE onto every target.

**Note:** iBSE must use a database repository (SQL or Oracle). Do not use a file repository. You can select this in the Repository Type drop-down list in the iBSE monitoring page. After configuring a database repository, you must restart all of the managed servers.

<http://hostname:port/ibse/IBSEConfig/>

where:

*hostname*

Is where your application server is running. Use the IP address or machine name in the URL; do not use localhost.

*port*

Is the port specific to each server, since you deploy iBSE to an entire cluster. For example, 8001, 8002, or any other port that is specified for each managed node.

8. Click *Deploy*.

## Procedure: How to Configuring Ports and Channels in a Clustered Environment

To configure ports and channels in a clustered environment:

1. Open Swing Application Explorer in BEA WebLogic Workshop.
2. Create a new connection to the iBSE instance. For information on creating a new configuration, see *Create a New Configuration for iBSE or JCA* on page A-3.

The following image shows the New Configuration pane, with a drop-down list where you select a service provider and an iBSE URL field where you provide a URL for the configuration. Beneath the fields are two action buttons.



**Note:** Use the IP address or machine name in the URL; do not use localhost.

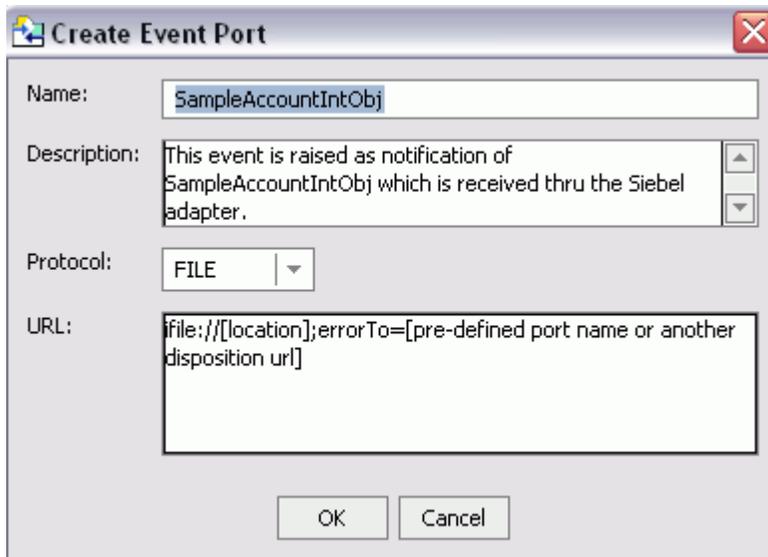
3. Connect to the new configuration and select the *Events* node in the left pane of Application Explorer.

The following image shows the new configuration selected in the left navigation pane of Application Explorer.



4. Add a new port. For more information, see *Creating an Event Port* on page B-4.

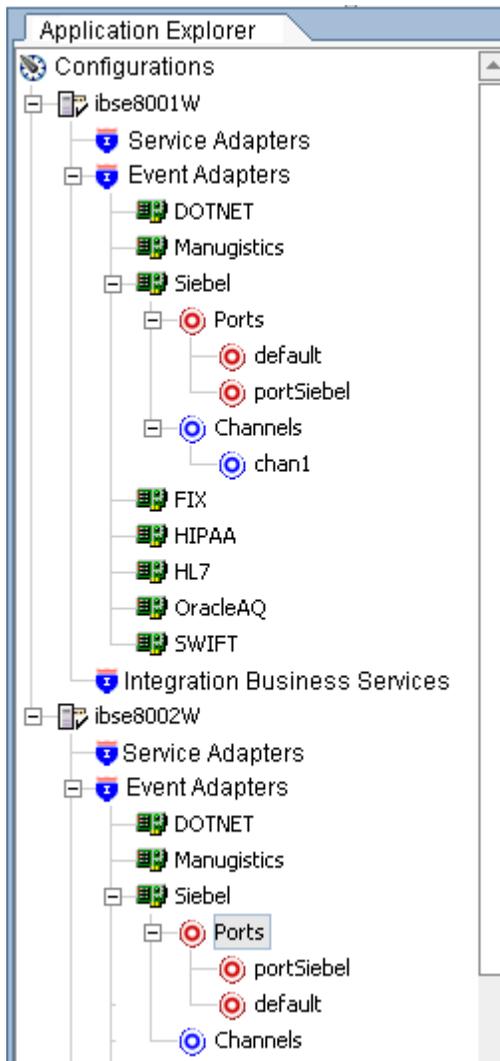
The following image shows the Create Event Port pane with four fields where you provide information for creating a new port. Beneath the fields are two action buttons.



5. Create a channel and add the port you created. For more information, see *Creating a Channel* on page B-15.
6. Click *Next* and enter the application server parameters.
7. Start the channel.
8. Create a new configuration and connect to the second iBSE instance.

The connection to iBSE must be configured to each instance of the managed server.

The following image shows two configurations listed in the Application Explorer navigation pane on the left.



The following operations performed on one managed server will be replicated on all other managed servers:

- Create port and channel: Creates the channel and port under all available servers.
- Delete port and channel. Deletes the port and channel under all available servers.

The following operations must be performed on each server:

- Start channel. Starts the channel for the specific server.
- Stop channel. Stops the channel for the specific server.



---

---

## APPENDIX C

### 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 BEA WebLogic 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" />
            
```

## Sample Files

```
<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" />
```

## Sample Files

```
<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 UIId" />
<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" />
```

## Sample Files

```
<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" />

```

## Sample Files

```
<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" />

```

## Sample Files

```
<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_spcUIId" 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"

```

## Sample Files

```
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" />

```

## Sample Files

```
<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"

```

## Sample Files

```
</>
  <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" />

```

## Sample Files

```
<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_spcUIId" 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"

```

## Sample Files

```
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" />

```

## Sample Files

```
<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>
```

*Sample Files*

---

---

## APPENDIX D

# 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 BEA WebLogic 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*.

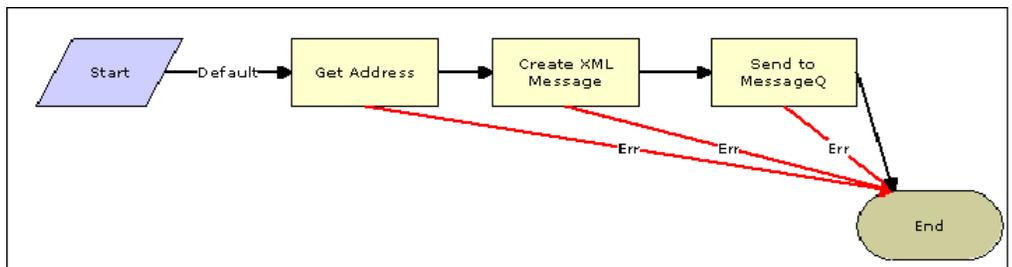
## Siebel Workflow - Outbound

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 the BEA WebLogic Adapter for Siebel. 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.



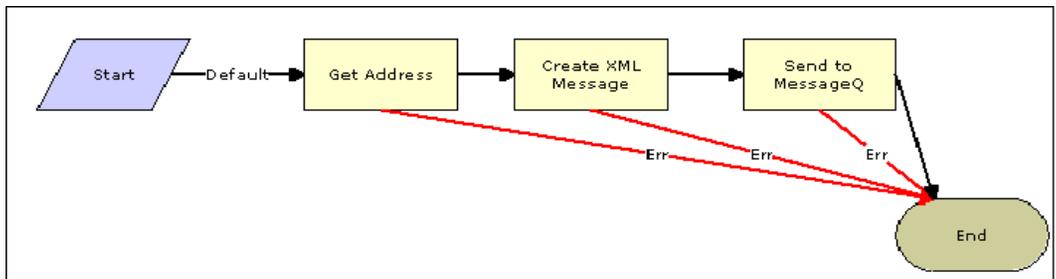
## Siebel Workflow - Inbound

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.



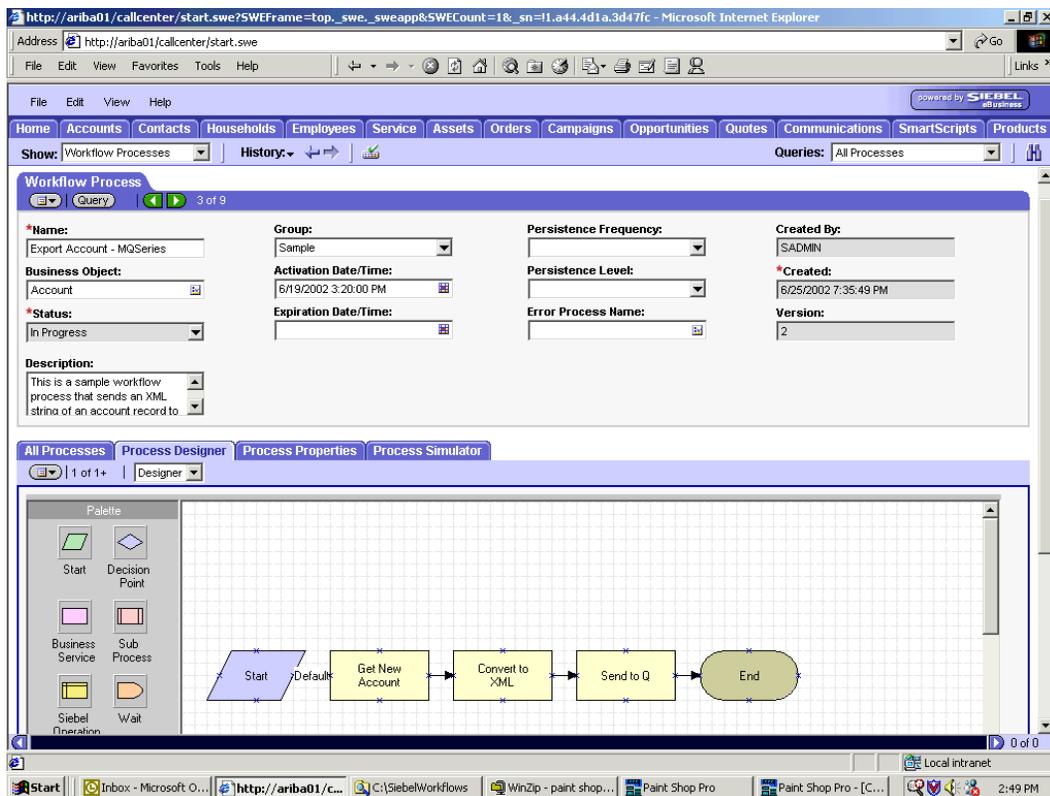
## Creating a Siebel Workflow

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.



## 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.

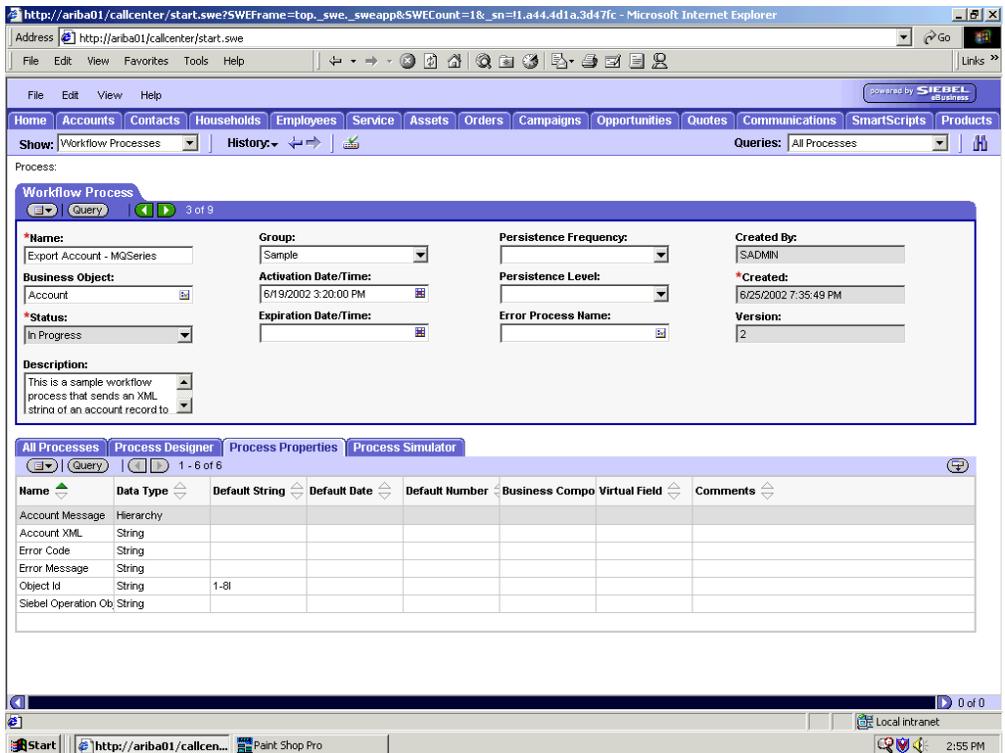
To create a Siebel Workflow:

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.

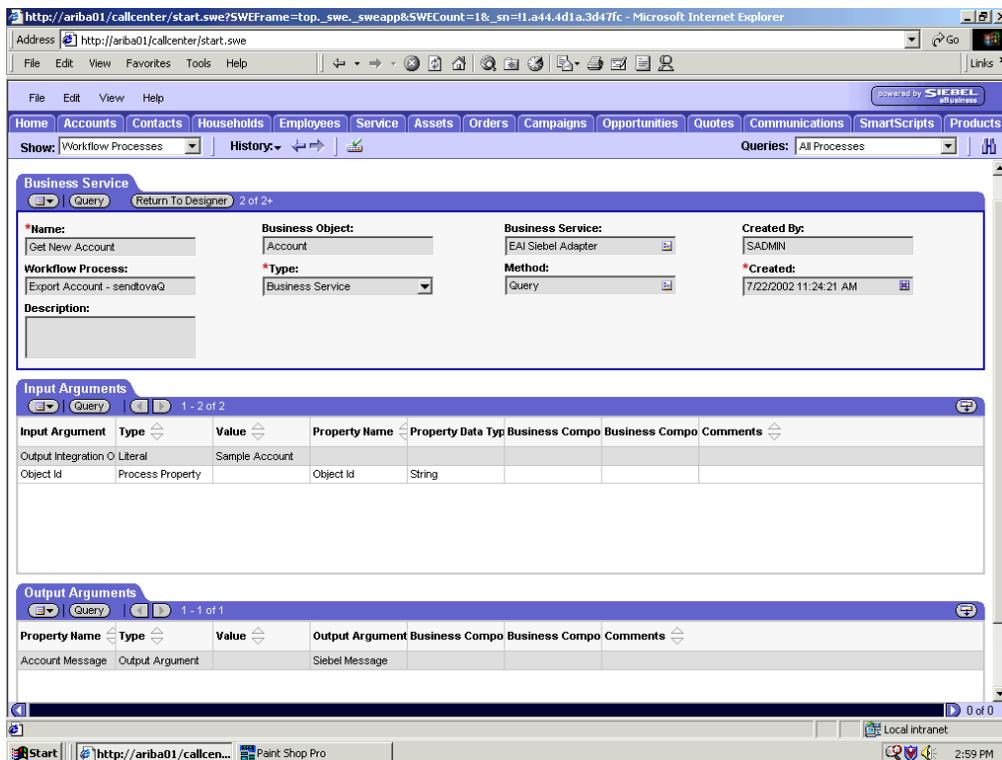


2. Use the Siebel Workflow Administration windows to create a Workflow.
3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

Using the Query method, the Business Service obtains the Account information from Siebel.

Output from this Business Service is generated in hierarchical format.

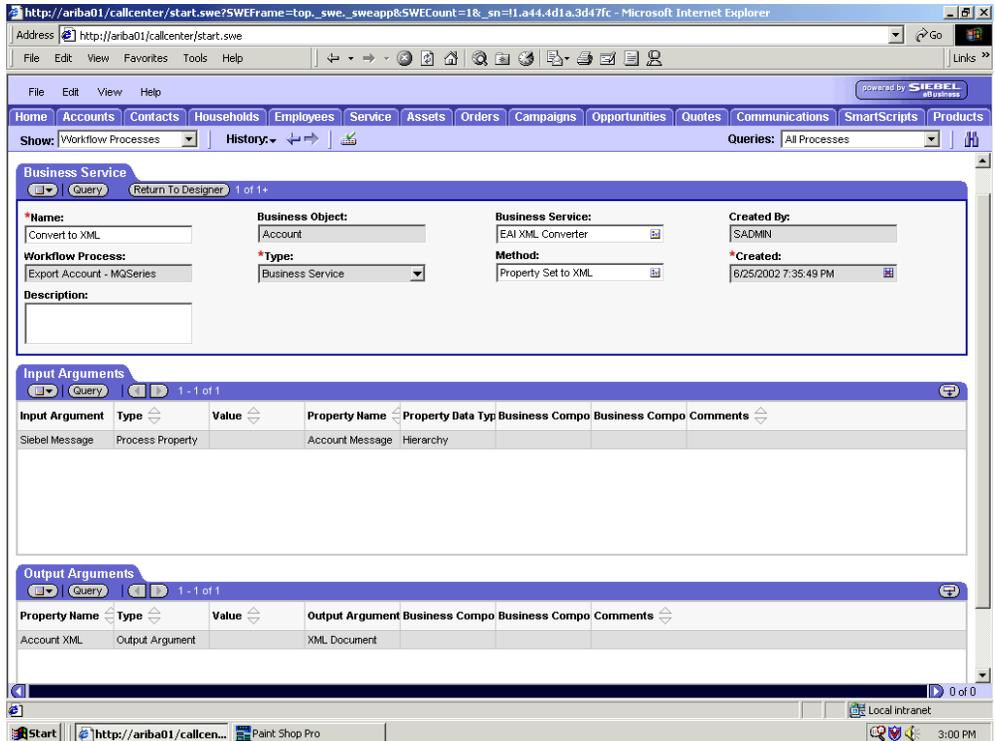
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



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.

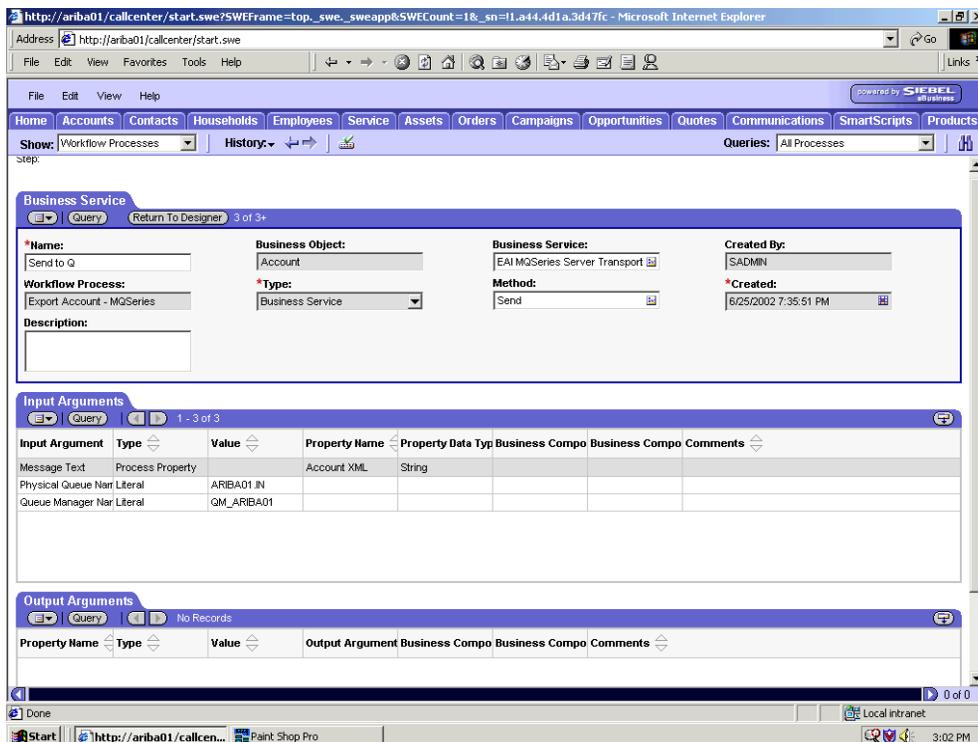
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



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.

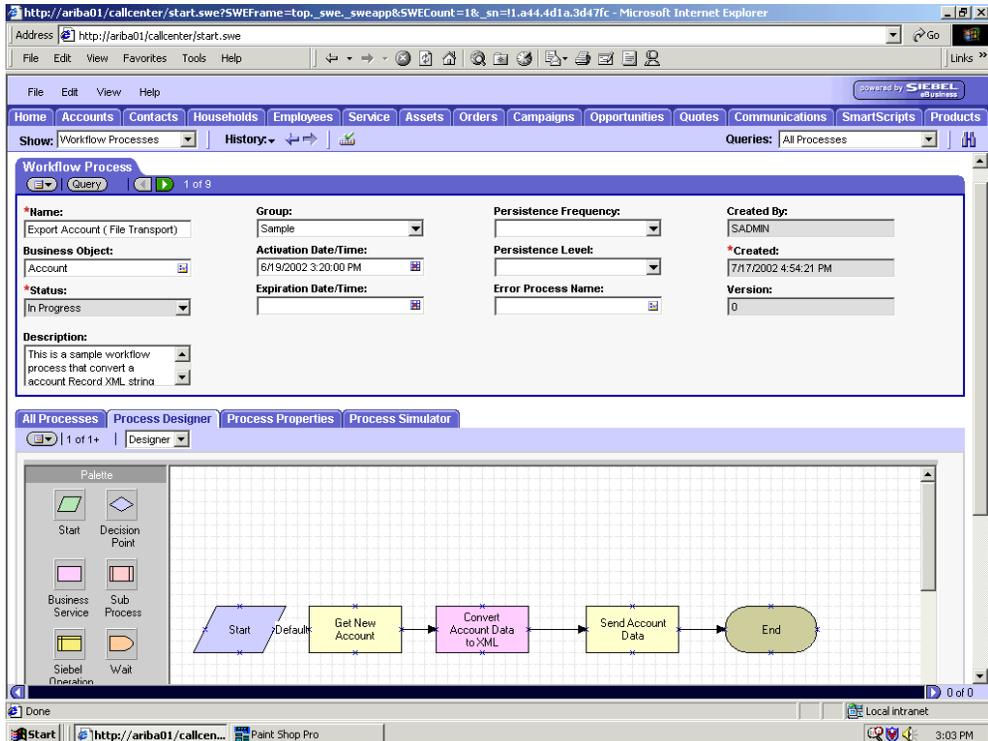


## Creating a Siebel Workflow for an Event Using File Transport

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.

## Creating a Siebel Workflow

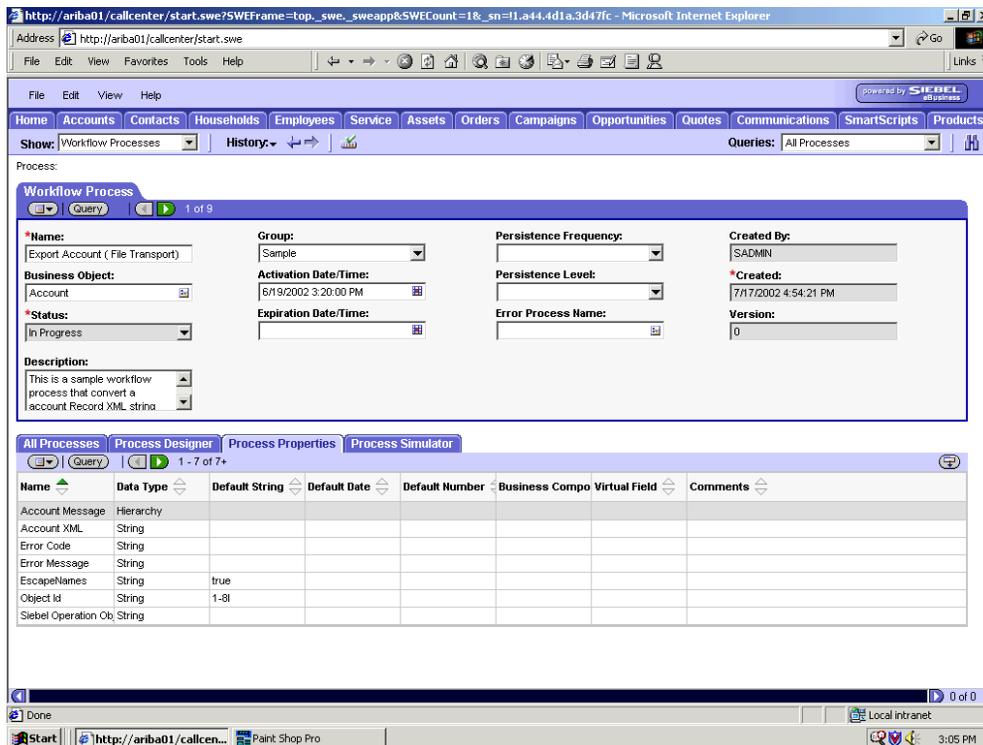
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists 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.



### 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.

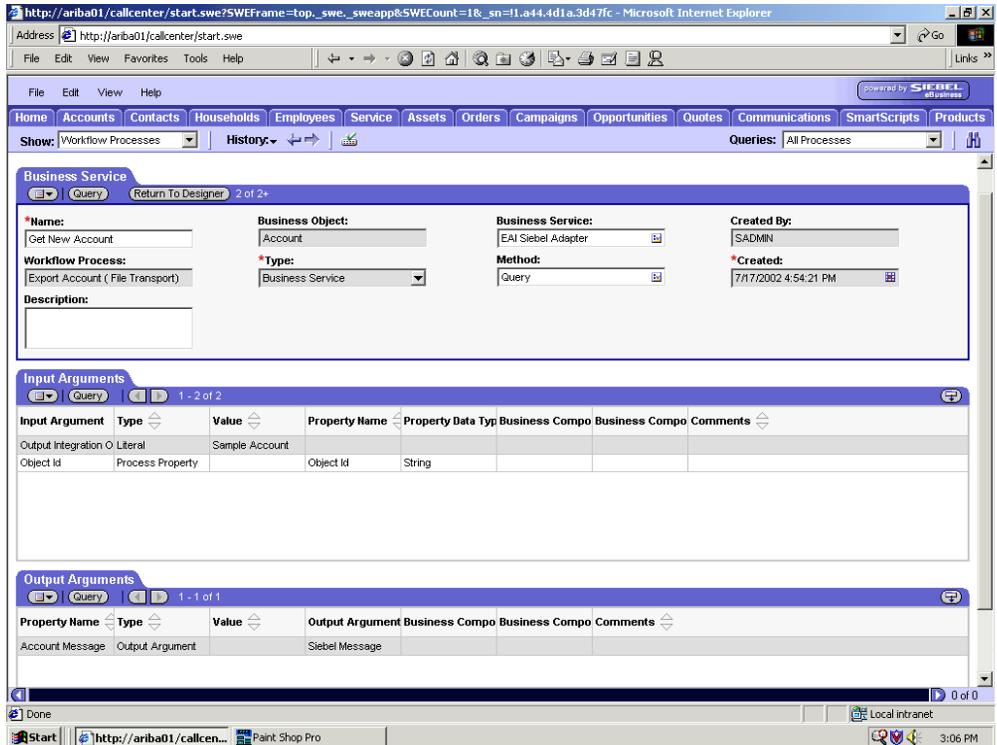
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists 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.



To create a Siebel Workflow:

- 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.
- 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.

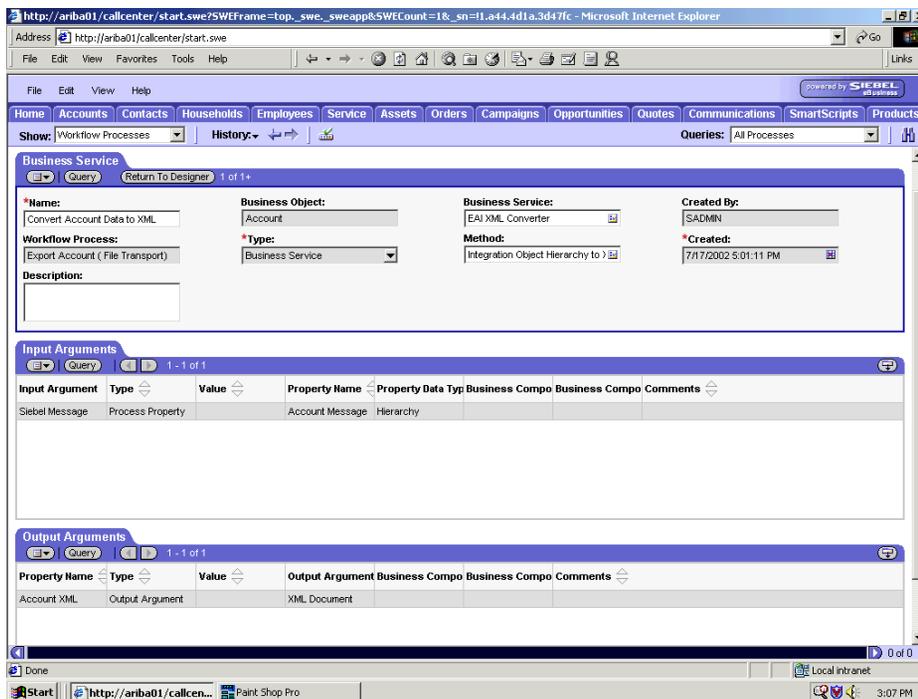


3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

Using the Query method, the Business Service obtains the Account information from Siebel.

Output from this Business Service is generated in hierarchical format.

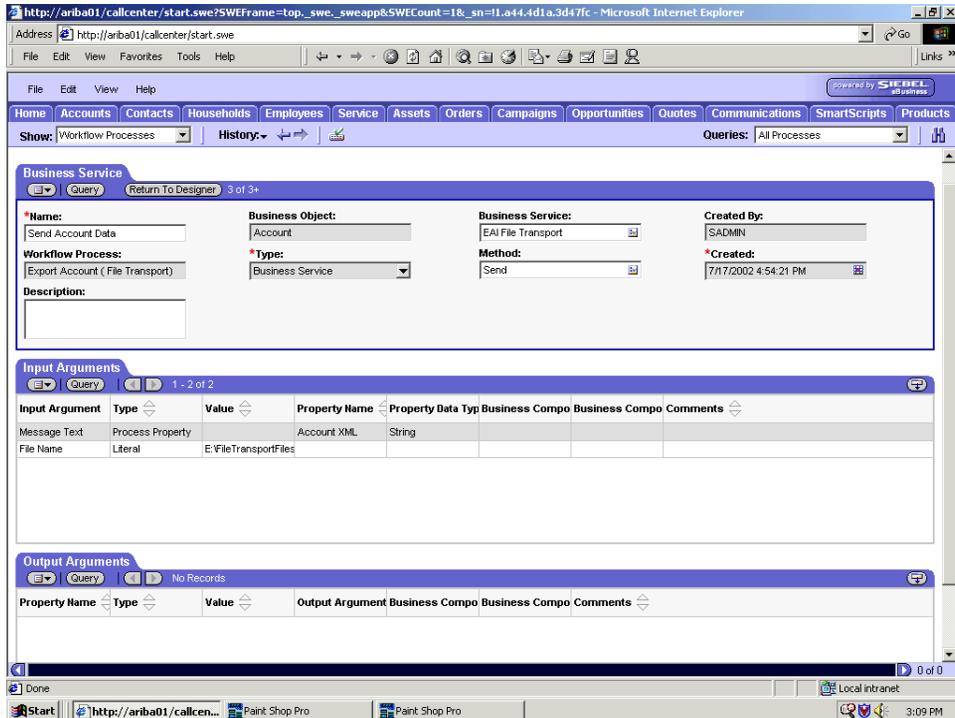
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



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.



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.

The screenshot displays the Siebel Workflow Administration interface within a Microsoft Internet Explorer browser window. The address bar shows the URL `http://arba01/callcenter/start.swe`. The application has a menu bar (File, Edit, View, Help) and a navigation bar with tabs for various entities: Home, Accounts, Contacts, Households, Employees, Service, Assets, Orders, Campaigns, Opportunities, Quotes, Communications, SmartScripts, and Products. Below the navigation bar, there are controls for 'Show' (set to 'Workflow Processes'), 'History', and 'Queries' (set to 'All Processes').

The main content area is divided into two panes. The upper pane, titled 'Workflow Process', contains a form for defining workflow parameters:

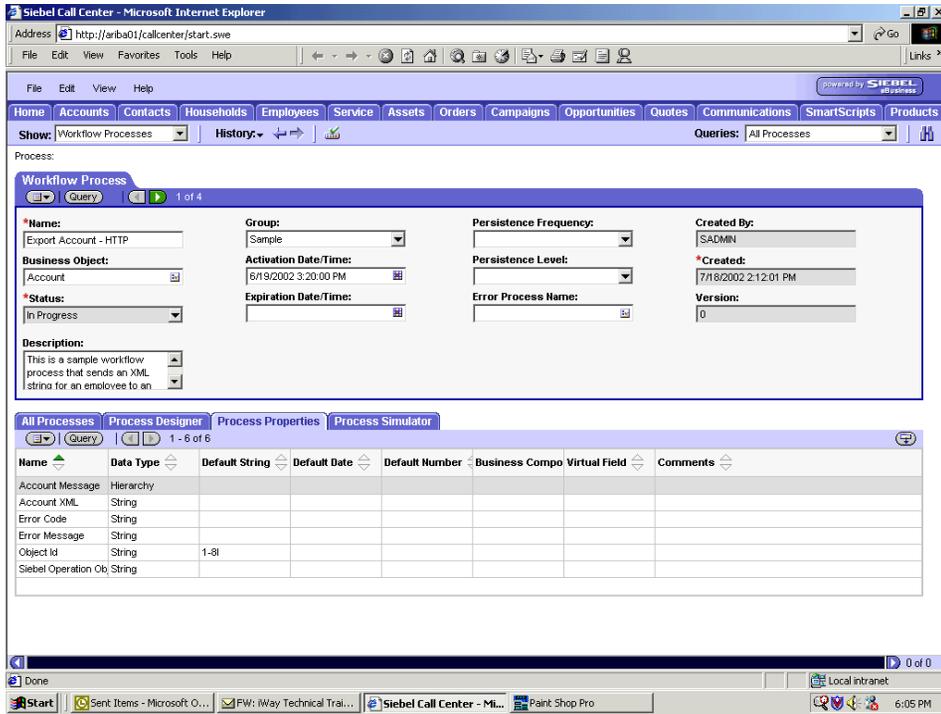
- Name:** Export Account - HTTP
- Group:** Sample
- Persistence Frequency:** (dropdown)
- Created By:** SADMIN
- Business Object:** Account
- Activation Date/Time:** 6/19/2002 3:20:00 PM
- Persistence Level:** (dropdown)
- \*Created:** 7/18/2002 2:12:01 PM
- \*Status:** In Progress
- Expiration Date/Time:** (dropdown)
- Error Process Name:** (dropdown)
- Version:** 0
- Description:** This is a sample workflow process that sends an XML string for an employee to an...

The lower pane, titled 'Process Designer', is active and shows a workflow diagram on a grid. On the left is a 'Palette' with icons for Start, Decision Point, Business Service, Sub Process, Siebel Operation, and Wait. The workflow diagram consists of the following steps:

- Start** (blue diamond)
- Default** (text label)
- Get New Account** (yellow rectangle)
- Convert to XML** (yellow rectangle)
- Send - HTTP** (yellow rectangle)
- End** (green oval)

The taskbar at the bottom shows several open applications, including 'Sent Items - Microsoft O...', 'FW: iWay Technical Trai...', 'Siebel Call Center - M...', and 'Paint Shop Pro'. The system clock indicates the time is 6:04 PM on a local intranet.

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.



To create a Siebel Workflow:

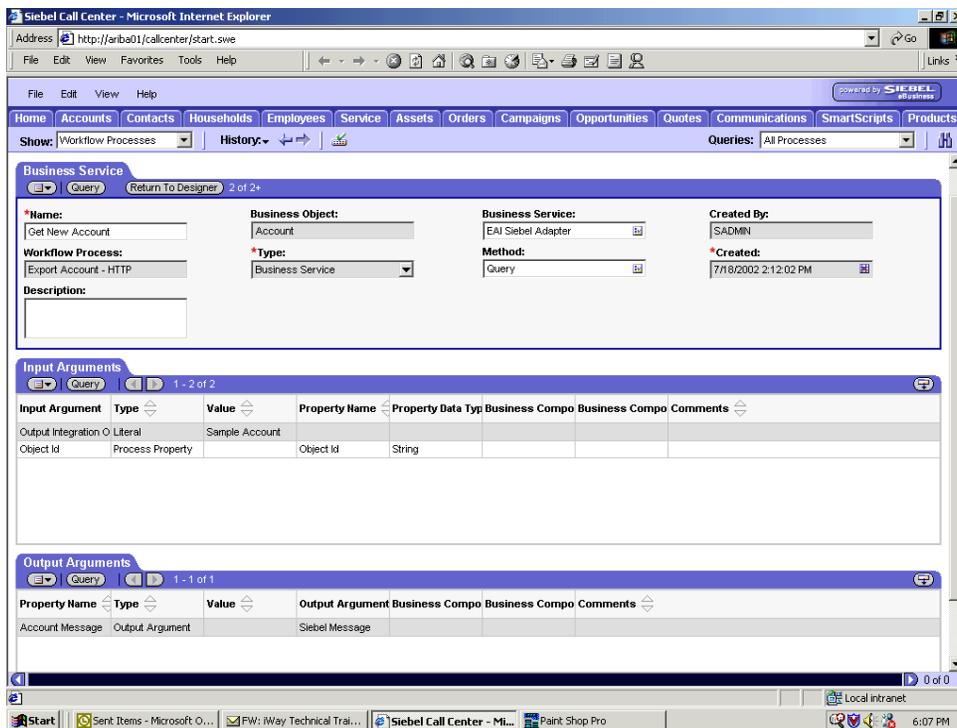
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

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.

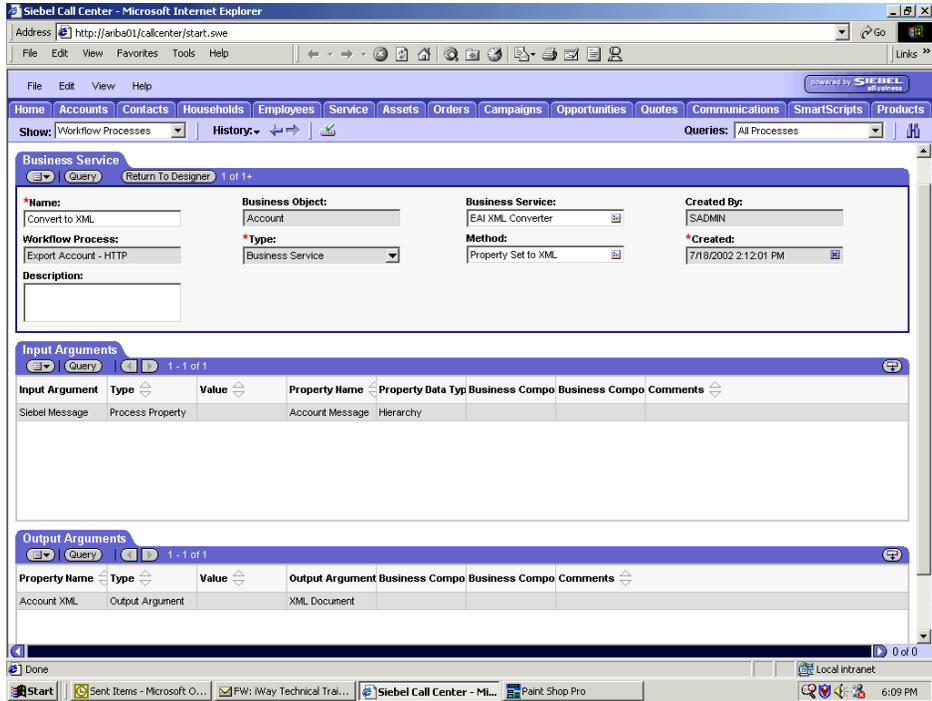


- Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it *Get New Account*.

Using the Query method, the Business Service obtains the Account information from Siebel.

Output from this Business Service is generated in hierarchical format.

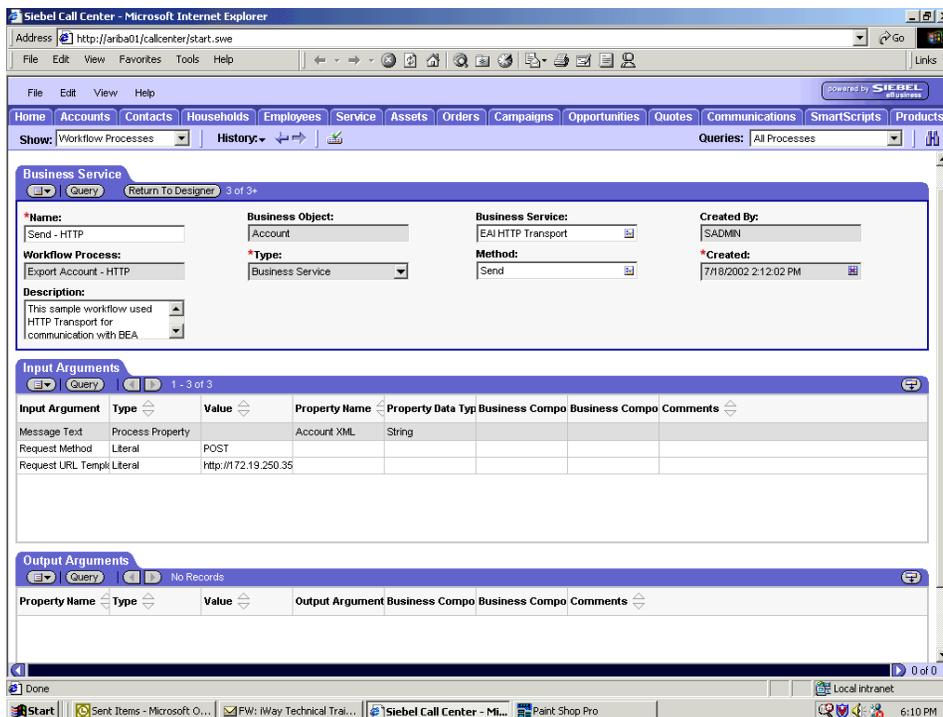
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



**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.



##### 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.

## Creating a Siebel 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 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.

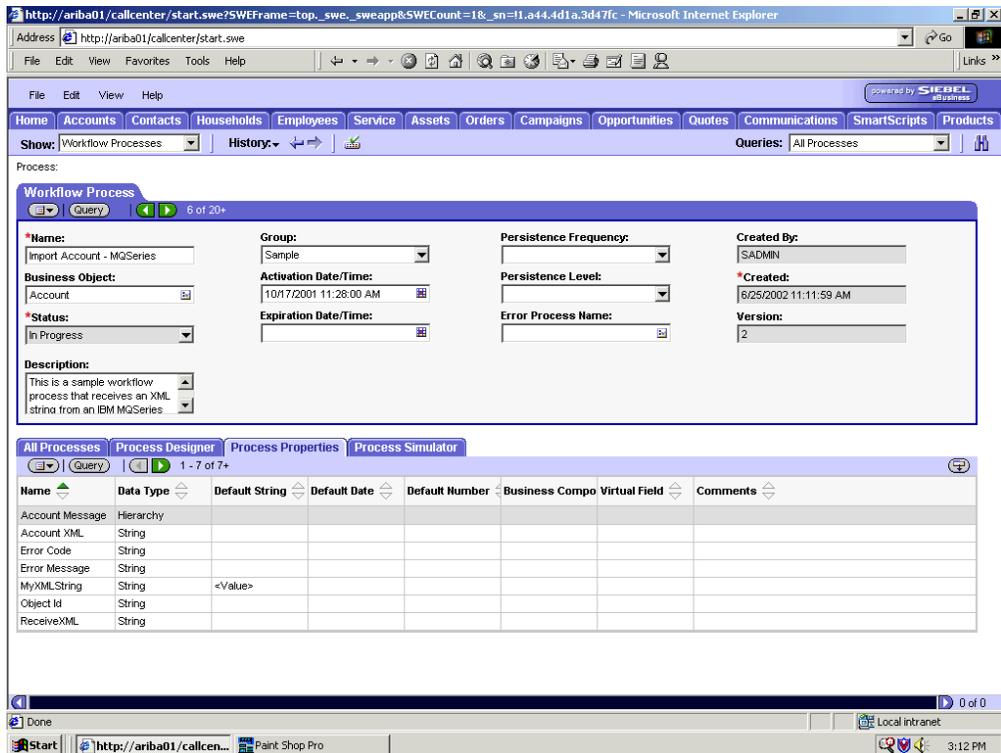
The screenshot displays the Siebel Workflow Administration application in a Microsoft Internet Explorer browser window. The address bar shows the URL: `http://ariba01/callcenter/start.swe?SWEFrame=top_swe_sweapp&SWECount=1&_sn=11.a44.4d1a.3d47fc`. The application interface features a top navigation bar with tabs for Home, Accounts, Contacts, Households, Employees, Service, Assets, Orders, Campaigns, Opportunities, Quotes, Communications, SmartScripts, and Products. Below this is a 'Show' dropdown menu set to 'Workflow Processes' and a 'Queries' dropdown menu set to 'All Processes'. The main content area is divided into two sections. The upper section, titled 'Workflow Process', contains a form for defining process details. The lower section, titled 'Process Designer', shows a workflow diagram on a grid. The diagram consists of five nodes connected by arrows: a blue parallelogram 'Start' node, a pink rectangle 'Receive' node, a yellow rectangle 'Get XML from MQ & Convert to Internal' node, a yellow rectangle 'Update Account' node, and a green oval 'End' node. A 'Default' label is positioned between the Start and Receive nodes. To the left of the diagram is a 'Palette' containing various workflow elements: Start, Decision Point, Business Service, Sub Process, Siebel Iteration, and Wait. The bottom of the browser window shows the taskbar with the Start button, a taskbar button for 'http://ariba01/callcen...', and a system tray showing 'Local intranet' and the time '3:11 PM'.

Field	Value
Name	Import Account - MQSeries
Group	Sample
Persistence Frequency	
Created By	SADMIN
Business Object	Account
Activation Date/Time	10/17/2001 11:28:00 AM
Persistence Level	
*Created	6/25/2002 11:11:59 AM
*Status	In Progress
Expiration Date/Time	
Error Process Name	
Version	2
Description	This is a sample workflow process that receives an XML string from an IBM MQSeries

## 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.



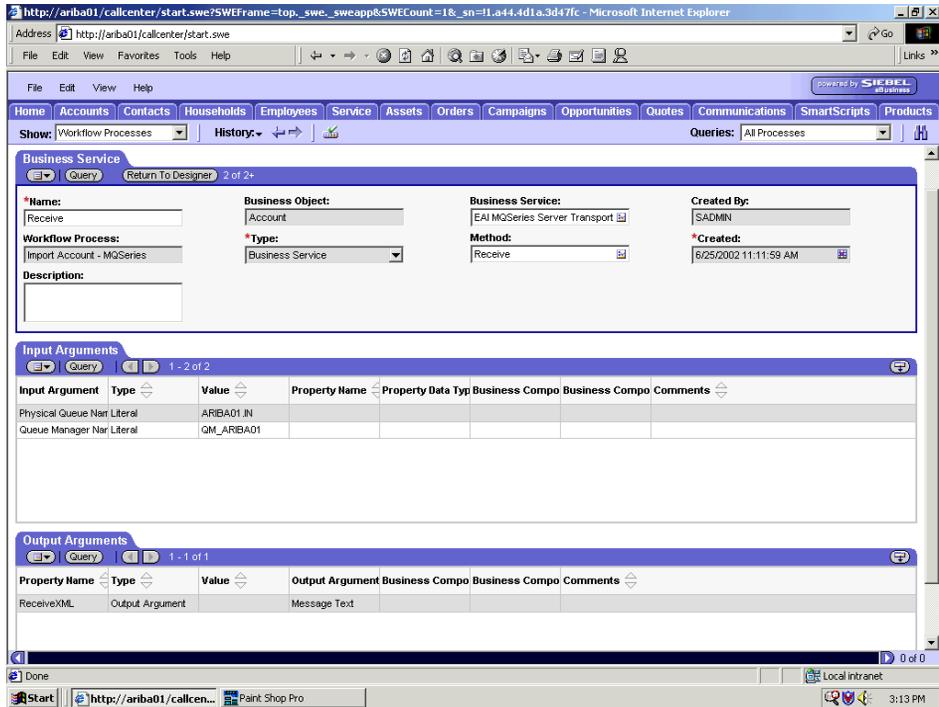
To create a Siebel Workflow:

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.

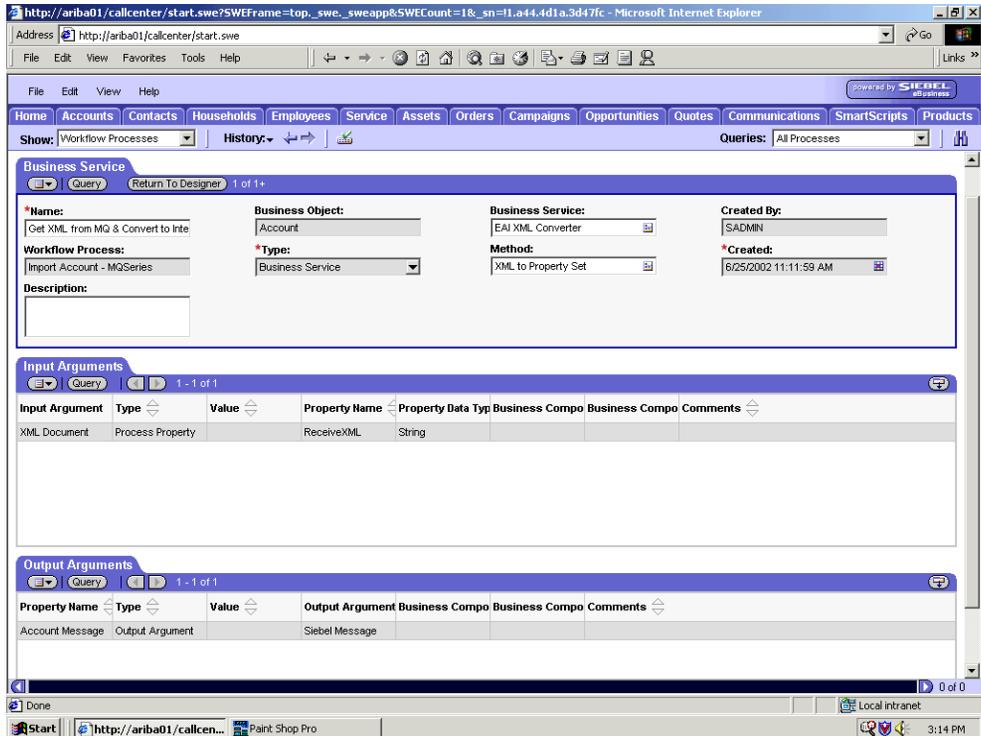


**2.** Define an EAI MQSeries Server Transport Business Service step and call it *Receive*.

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.

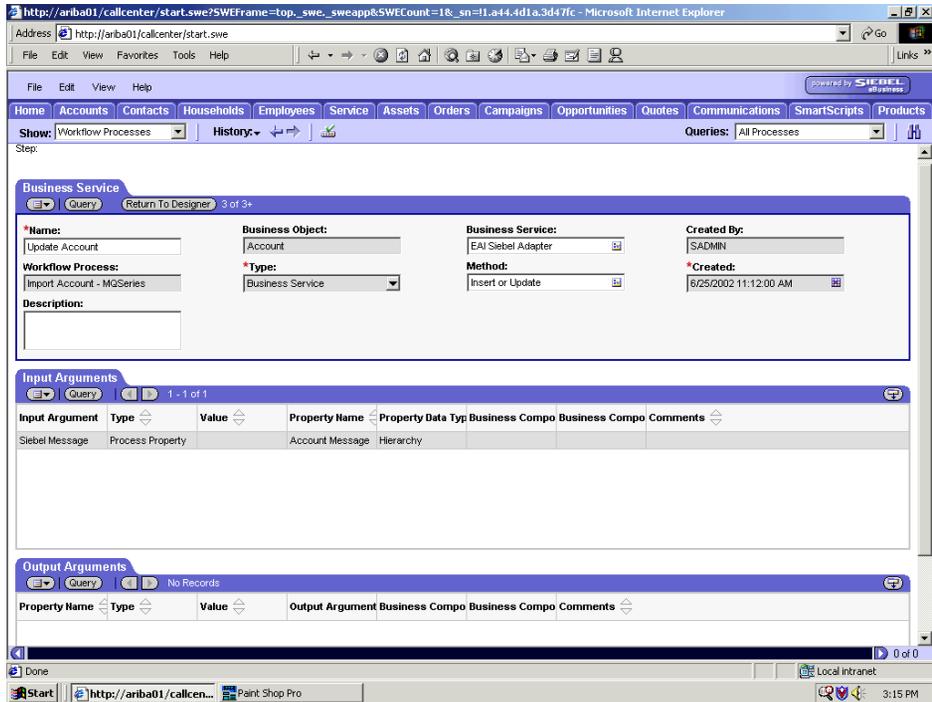
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



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.



**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.

The screenshot displays the Siebel Workflow Administration interface within a Microsoft Internet Explorer browser window. The address bar shows the URL: `http://ariba01/callcenter/start.swe?SWEframe=top_swe_sweapp&SWEcount=1&_sn=11.a44.4d1a.3d47fc`. The browser window title is "http://ariba01/callcenter/start.swe - Microsoft Internet Explorer".

The application interface features a top navigation bar with tabs for: Home, Accounts, Contacts, Households, Employees, Service, Assets, Orders, Campaigns, Opportunities, Quotes, Communications, SmartScripts, and Products. Below this is a "Show:" dropdown menu set to "Workflow Processes" and a "Queries:" dropdown menu set to "All Processes".

The main content area is divided into two panes. The upper pane, titled "Workflow Process", contains the following configuration fields:

- Name:** Import Account (File Transport)
- Group:** Sample
- Persistence Frequency:** (empty dropdown)
- Created By:** SADMIN
- Business Object:** Account
- Activation Date/Time:** 6/19/2002 3:20:00 PM
- Persistence Level:** (empty dropdown)
- \*Created:** 7/18/2002 8:41:15 AM
- \*Status:** In Progress
- Expiration Date/Time:** (empty dropdown)
- Error Process Name:** (empty dropdown)
- Version:** 0
- Description:** This is a sample workflow process that receives an Account Record XML and...

The lower pane, titled "Process Designer", shows a workflow diagram on a grid. The diagram consists of the following steps:

- Start:** A blue parallelogram shape.
- Receive Account Data:** A yellow rounded rectangle.
- Convert from XML:** A pink rounded rectangle.
- Update or Insert New Account:** A yellow rounded rectangle.
- End:** A green oval shape.

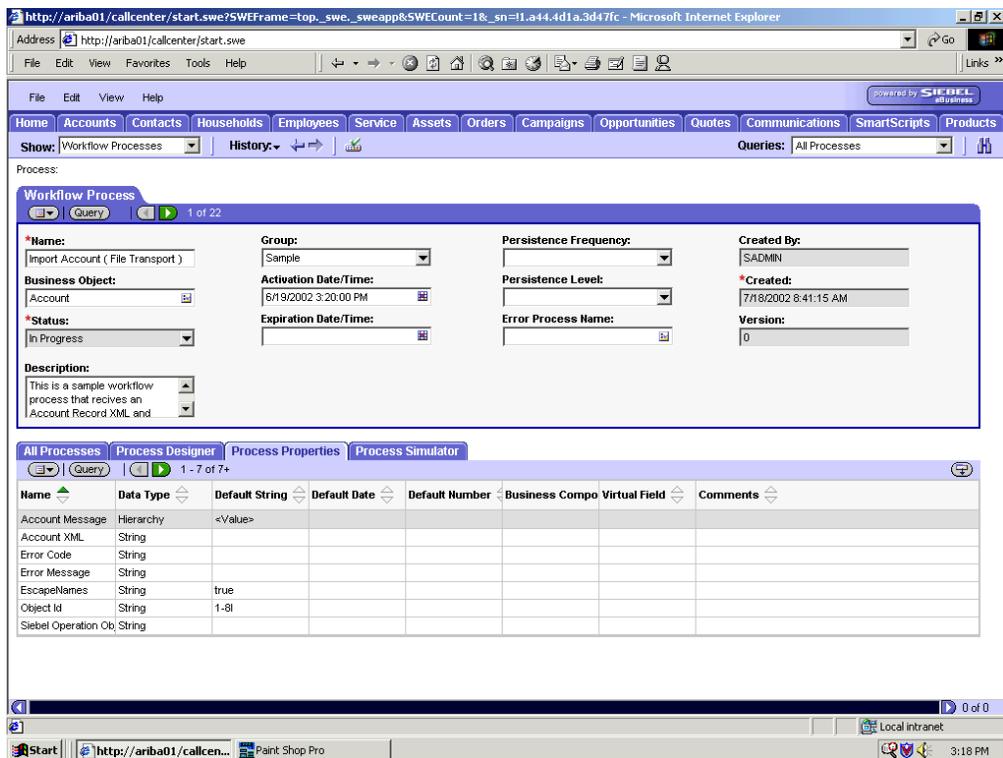
Arrows indicate the flow from Start to Receive Account Data, then to Convert from XML, then to Update or Insert New Account, and finally to End. A "Default" label is positioned between the Start and Receive Account Data steps. To the left of the diagram is a "Palette" containing icons for Start, Decision Point, Business Service, and Sub Process.

The bottom of the browser window shows the taskbar with the Start button, a taskbar button for "http://ariba01/callcen...", and a taskbar button for "Paint Shop Pro". The system tray on the right shows "Local intranet" and the time "3:17 PM".

## 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.



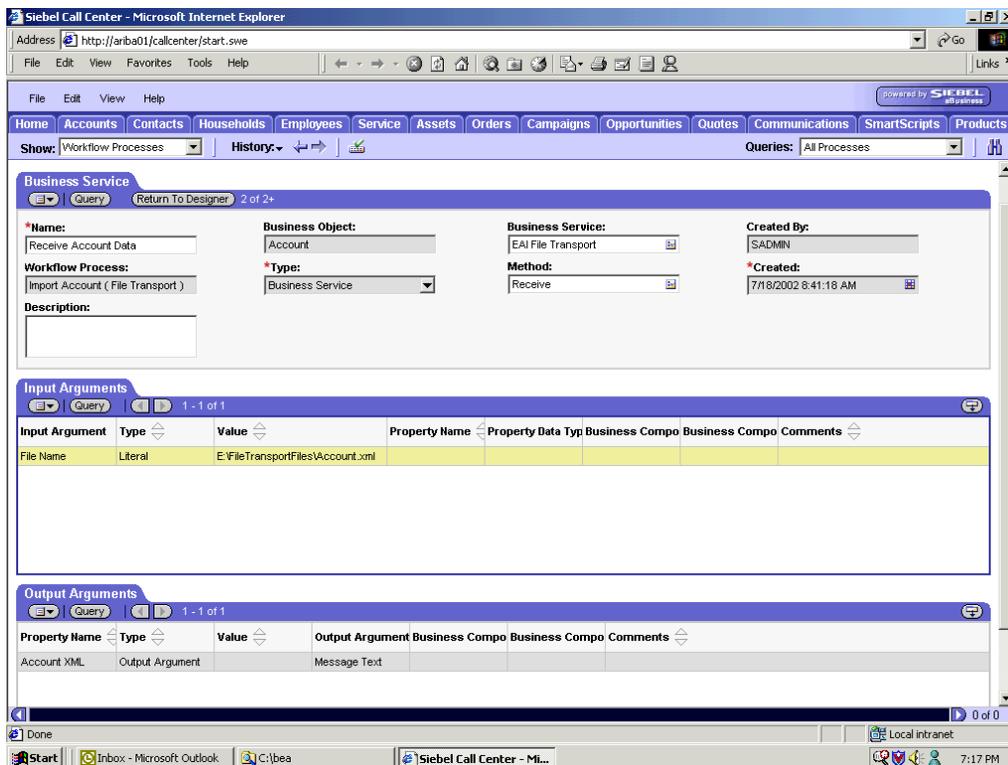
To create a Siebel Workflow:

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.

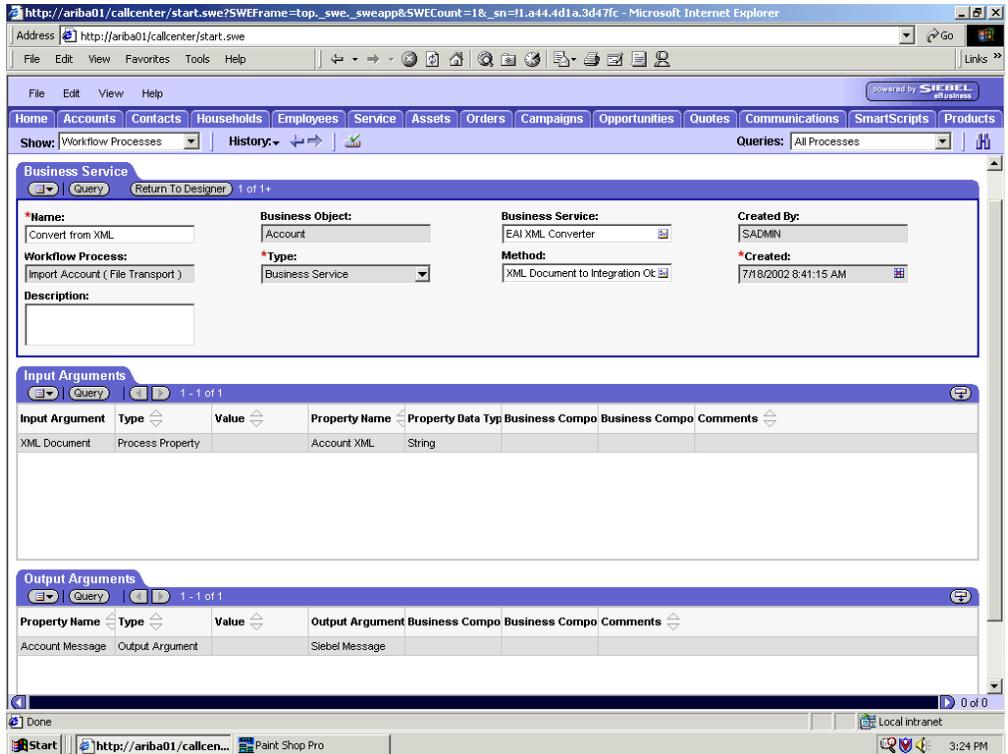


2. Define an EAI FileTransport Business Service step and call it *Receive Account Data*.

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.

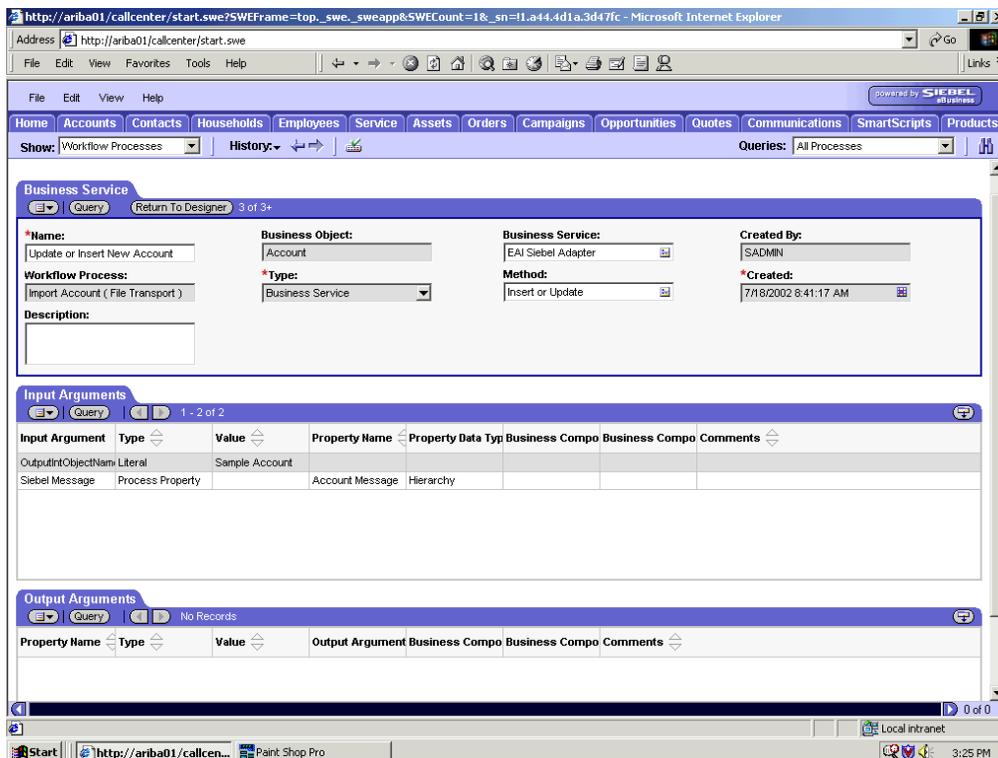
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists 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.



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.



4. Define an EAI Siebel Adapter Business Service step and call it *Update or Insert New Account*.

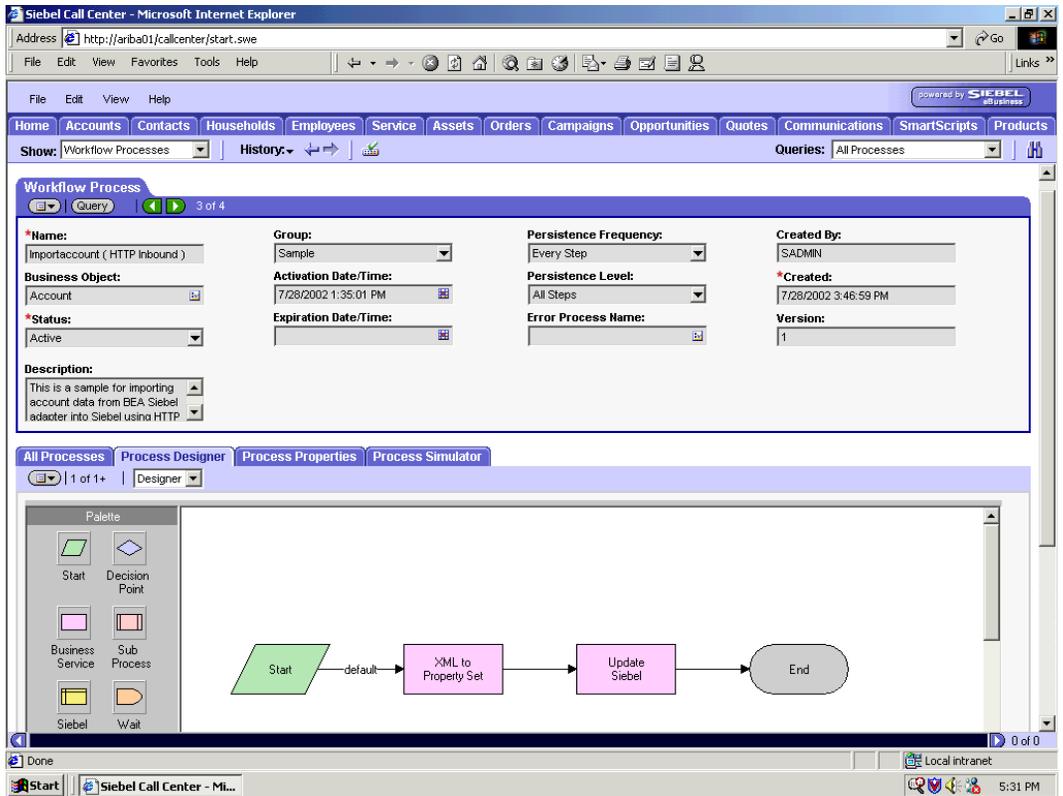
This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

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.

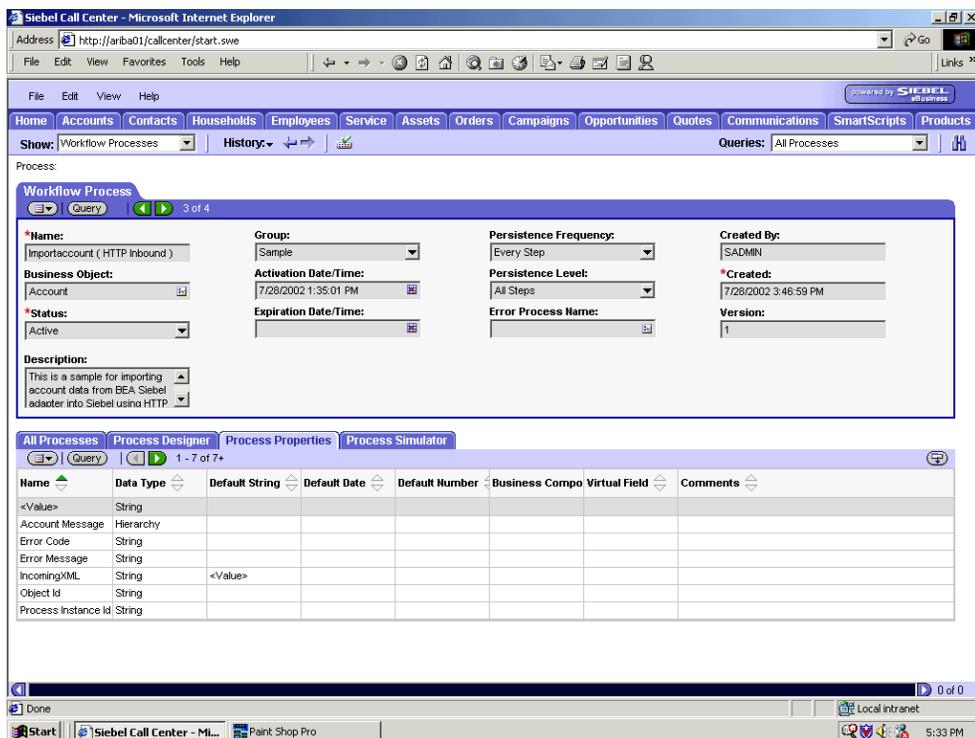
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Workflow Process tab, which includes several fields and drop-down lists 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.



## 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.



To create a Siebel Workflow:

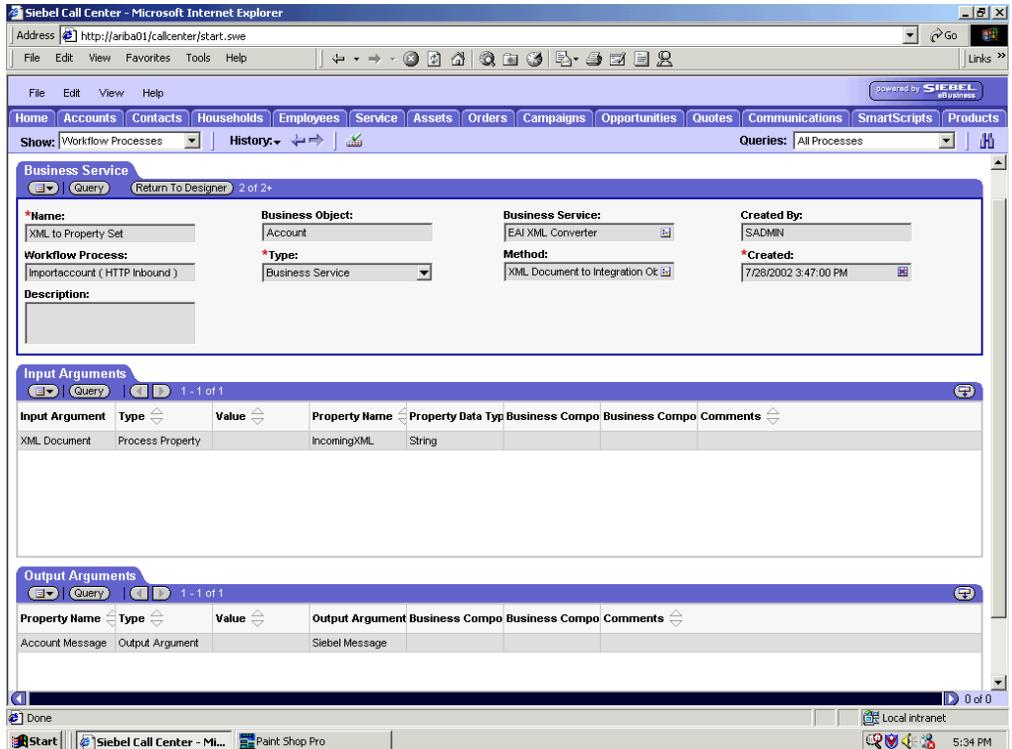
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

Account message contains the Siebel Account data in hierarchical format.

Account XML specifies the Siebel Account data that the workflow converted to XML.

## Creating a Siebel Workflow

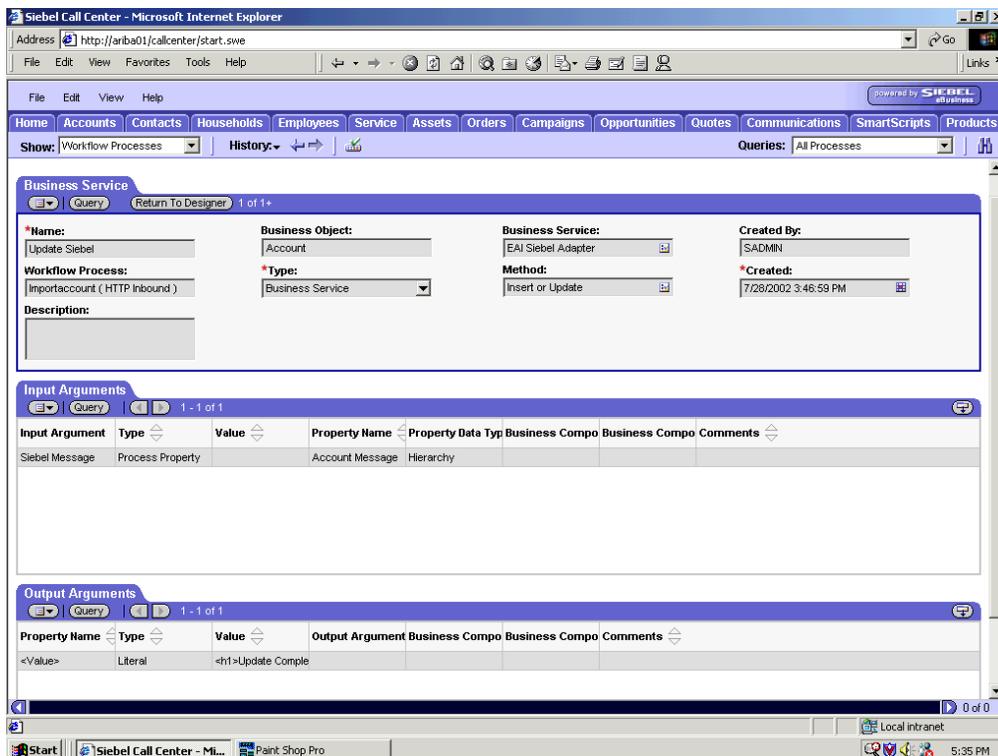
The following image shows the Siebel Workflow Administration window. The window includes fourteen tabs across the top, a Show drop-down list in the upper left, and a Queries drop-down list in the upper right. The upper pane shows the Business Service tab, which includes several fields and drop-down lists for defining the 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.



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.



### 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.

