iWay

iWay Adapter for Baan for BEA WebLogic User's Guide
Version 5 Release 5

February 11, 2005
DN3501334.0205

Preface

This documentation describes how to use the iWay Adapter for Baan for BEA WebLogic. It is intended for system integrators who develop client interfaces between Baan and other applications.

How This Manual Is Organized

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

Chapter/Appendix		Contents
1	Introducing the iWay Adapter for Baan for BEA WebLogic	Describes the features of the iWay Adapter for Baan for BEA WebLogic and provides information for accomplishing your integration projects.
2	Creating XML Schemas and Business Services	Describes how to use Servlet Application Explorer to generate XML schemas, view metadata, and create business services
3	Listening for Database Events	Describes how to use the iWay Adapter for Baan for BEA WebLogic to listen for events in a relational table.
4	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
5	Management and Monitoring	Describes how to use managing and monitoring tools provided by iBSE and JCA to gauge the performance of your run-time environment.
A	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services	Describes how to use Java Swing Application Explorer in BEA WebLogic Workshop to create XML schemas for Baan.
B	Using Application Explorer in BEA WebLogic Workshop for Event Handling	Describes how to use Application Explorer running in BEA WebLogic Workshop.

Documentation Conventions

The following table lists the conventions that apply in this manual and a description of each.

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

Related Publications

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

Customer Support

Do you have questions about the iWay Adapter for Baan for BEA WebLogic?

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

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

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

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

Help Us to Serve You Better

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

The following tables list the specifications our consultants require.

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

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

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

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

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

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

Version	**Vendor**

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

Request/Question	**Error/Problem Details or Information**
Provide usage scenarios or summarize the application that produces the problem.	
Did this happen previously?	
Can you reproduce this problem consistently?	
Any **change in the application environment**: software configuration, EIS/ database configuration, application, and so forth?	

Request/Question	Error/Problem Details or Information
Under what circumstance does the problem *not* occur?	
Describe the **steps** to reproduce the problem.	
Describe the **problem**.	
Specify the **error** message(s).	

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

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

User Feedback

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

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

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

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

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

Contents

CHAPTER 1

Introducing the iWay Adapter for Baan for BEA WebLogic

Topics:

- iWay Adapter for Baan for BEA WebLogic Overview
- Sending Requests to Baan
- Listening for Baan Events
- Deployment Information for the iWay Adapter for Baan for BEA WebLogic

The following section provides an overview of the iWay Adapter for Baan for BEA WebLogic and a description of how it works, including key features and functionality.

iWay Adapter for Baan for BEA WebLogic Overview

The iWay Adapter for Baan for BEA WebLogic provides a means to exchange real-time business data between Baan systems and other applications, databases, and external business partner systems. The adapter enables these systems to:

- **Send requests to Baan** and receive responses.

  This is referred to as service mode. For more information, see *Sending Requests to Baan*.

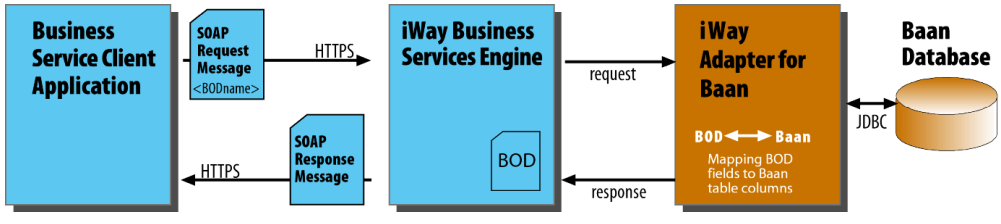
- **Listen for Baan events**, such as changes to the Baan database made by users and applications.

  The event information is then sent to the appropriate system. This is referred to as event mode. For more information, see *Listening for Baan Events* on page 1-4.

Sending Requests to Baan

The iWay Adapter for Baan for BEA WebLogic enables an application to send a request to Baan to perform a particular Baan service. The adapter's agent automatically performs all required mapping between your request document and the Baan database. It then uses the data from the request document to execute the requested operation against the Baan database using JDBC™ calls. The agent returns the result (or status) to the application or redirects it to a different destination, if desired.

When you deploy the adapter using Integration Business Services Engine (iBSE), the following illustrates the equivalent process.



Open Applications Group (OAG) Business Object Documents (BODs) provide an industry standard interface between your application and Baan. A request document references a particular OAG BOD that is associated with a specific Baan table and type of operation. The adapter's agent uses the request document to map the BOD fields to the Baan database. It then uses this information to execute the requested operation against the Baan database through JDBC calls. The agent returns the results (or status) to the application or redirects it to a different destination.

The adapter provides sample request documents and corresponding sample response documents that show both successful and unsuccessful responses. You can use these samples to understand request document formats, create your own customized request document, or develop logic that processes response documents. The following table lists the sample request documents available through the adapter.

For information about OAG BOD schemas, instances, explanations, and testing tools, download the latest Open Applications Group Interface Specification from

www.openapplications.org

Baan Table Alias	Sample Request Document	Database Operation
BillOfMaterials	bom_create.xml	Insert
BillOfMaterials	bom_delete.xml	Delete
BillOfMaterials	bom_get.xml	Query
BillOfMaterials	bom_sync.xml	Update
Customer	customer_create.xml	Insert
Customer	customer_delete.xml	Delete
Customer	customer_get.xml	Query
Customer	customer_get_multi.xml	Query (more than one)
Customer	customer_sync.xml	Update
Inventory	inventory_create.xml	Insert
Inventory	inventory_delete.xml	Delete
Inventory	inventory_get.xml	Query
Inventory	inventory_sync.xml	Update
Item	item_create.xml	Insert
Item	item_delete.xml	Delete
Item	item_get.xml	Query
Item	item_sync.xml	Update
PurchaseOrder	purchase_create.xml	Insert
PurchaseOrder	purchase_delete.xml	Delete

Baan Table Alias	Sample Request Document	Database Operation
PurchaseOrder	purchase_get.xml	Query
PurchaseOrder	purchase_sync.xml	Update
SalesOrder	salesorder_create.xml	Insert
SalesOrder	salesorder_delete.xml	Delete
SalesOrder	salesorder_get.xml	Query
SalesOrder	salesorder_sync.xml	Update

Listening for Baan Events

The iWay Adapter for Baan for BEA WebLogic enables an application to listen to a Baan database for changes made by users and applications. It captures the required information about each change and sends it to the destination you specify. For more information, see Chapter 3, *Listening for Database Events*.

Listening for events is useful for applications that are initiated upon a specific condition or function occurring inside of Baan. The adapter can be configured to notify the event's subscriber that the condition or function occurred. The notification can be transmitted through a variety of communication protocols.

Deployment Information for the iWay Adapter for Baan for BEA WebLogic

The iWay Adapter for Baan for BEA WebLogic works in conjunction with the following components:

- Application Explorer

and either

- Integration Business Services Engine (iBSE)

or

- iWay Enterprise Connector for J2EE™ Connector Architecture (JCA)

Application Explorer, used to configure Baan 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 iWay 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 iWay Adapters instead of using Web services.

Both iBSE and the iWay connector for JCA are deployed to an application server, such as BEA WebLogic Server, with Application Explorer and the adapters.

Deployment Information Roadmap

The following table lists the location of deployment information for the iWay Adapter for Baan for BEA WebLogic. A description of the Integration Business Services Engine (iBSE) and the iWay Enterprise Connector for J2EE Connector Architecture (JCA) follow the table.

Deployed Component	For more information, see
Application Explorer	• Chapters 2, 3, and 4, and Appendix A of this guide • *iWay Installation and Configuration for BEA WebLogic*
Integration Business Services Engine (iBSE)	• *iWay Installation and Configuration for BEA WebLogic*
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	• *iWay Installation and Configuration for BEA WebLogic*

The Integration Business Services Engine (iBSE)

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, but 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 iWay Enterprise Connector for J2EE Connector Architecture (JCA)

The iWay Enterprise Connector for J2EE Connector Architecture (JCA) enables developers of JCA-compliant applications to deploy iWay adapters as JCA resources. The connector is supported on J2EE-compliant application servers.

The iWay Connector for JCA is distributed as a standard Resource Adapter Archive (RAA) 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 and Business Services

Topics:

- Generating a Schema or a Business Service
- Starting Application Explorer
- Opening a Connection to a Database
- Closing or Deleting a Connection to a Database
- Editing Connection Parameters
- Viewing Metadata
- Example Request, Response, and Event Documents
- Generating a Business Service for an XML Request Document

This section describes how to use Servlet Application Explorer to:

- **Generate XML schemas** that define request and response documents. You can use these schemas when you create request documents and when you develop logic for processing response documents.
- **View metadata** that describes your SQL statements. You can use the metadata when you create request documents and when you develop logic for processing response documents.
- **Create business services** (also known as Web services) for your SQL statements.

Generating a Schema or a Business Service

You can use Servlet Application Explorer to connect to a Baan database. You can generate request and response document schemas. You also can generate and test a business service.

To generate request and response document schemas or a business service:

1. **Start Application Explorer and open a new or existing connection** to a relational or non-relational database management system, as described in *Opening a Connection to a Database* on page 2-4.

   After you launch Application Explorer, you can expand the Service Adapters node to view a list of adapters installed on your system.

   After you finish using a connection, you can close it. If you won't require the connection in the future, you can delete it. For more information, see *Closing or Deleting a Connection to a Database* on page 2-9.

   Note: When you close Application Explorer, it automatically closes all open connections.

2. **Generate XML schemas** that define request and response documents for your SQL statements, as described in *Example Request, Response, and Event Documents* on page 2-11.

   You can use the schemas when you create request documents and when you develop logic to process responses.

3. **Create a request document** for each operation against each table.

   You can use a third-party XML tool to generate a request document from the XML schema.

   You also may find it helpful to examine the metadata describing your SQL statements and procedures, as described in *Viewing Metadata* on page 2-10. For information about request and response document formats, see *Example Request, Response, and Event Documents* on page 2-11.

4. **Generate a business service** (also known as a Web service) for an SQL statement.

   To generate a business service, you must first deploy the adapter to a server host with the servlet Integration Business Services Engine (iBSE).

   For configuration and deployment information, see the *iWay 5.5 Installation and Configuration* manual. For more information on Web services, see *Generating a Business Service for an XML Request Document* on page 2-19.

Starting Application Explorer

Servlet Application Explorer is the adapter configuration tool and user interface between the iWay Adapter for Baan for BEA WebLogic and the Baan database. Before you can work with your adapter, you must start Application Explorer.

Procedure: How to Start Application Explorer

To start Application Explorer,

1. Ensure that the server running Application Explorer is started.
2. Enter the Application Explorer address in a browser, for example:

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

   where:

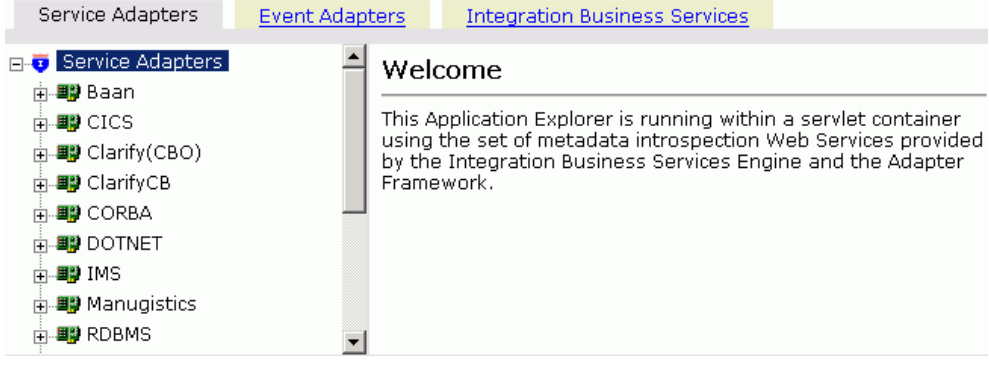
   hostname

   Is the machine where your application server is installed.

   port

   Is the port number on which the application server is listening.

   Application Explorer opens.



The Available Hosts drop-down list in the top frame determines which iWay JCA Connector and Servlet iBSE instances you can access. For information about adding instances, see the *iWay Installation and Configuration Guide*.

Three tabs appear near the top of the Application Explorer window. From left to right they are:

- Service Adapters, where you create and manage connections to the Baan database.
- Event Adapters, where you configure Baan database event listening.

- Integration Business Services, where you generate XML schemas, and create and view business services.

  The left pane of the window contains an expandable list of adapter nodes, events, or business services, depending on the tab that is selected. The right pane provides the details of the selected adapter, event, or service, and is the work area where you will define and modify adapter functions and services.

3. To display the list of available adapters, click the *Service Adapters* node located in the left pane.

You are ready to add new targets to Baan.

Opening a Connection to a Database

Although you create and open connections the same way, the specific connection you open depends on the type of data you wish to access. If you are generating schemas for SQL statements for a database, open a connection under the Baan node.

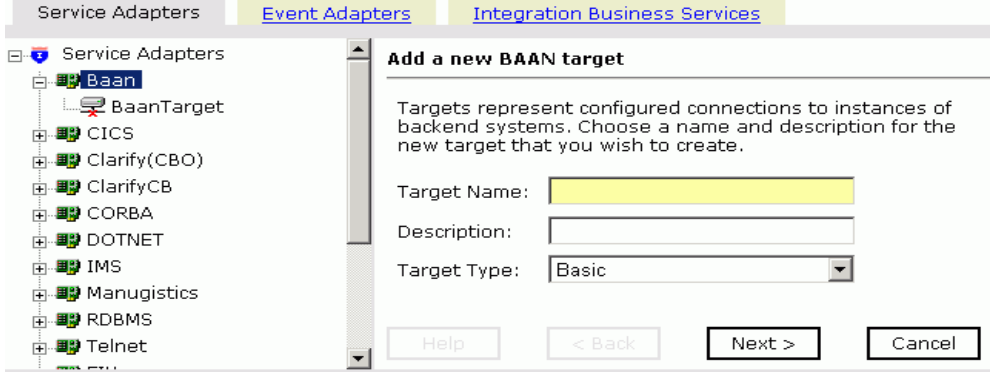
You can connect to a database by:

- Defining a new target, as described in *How to Define a New Connection on page 2-4*.
- Connecting to an existing Baan target, as described in *How to Open an Existing Connection* on page 2-8.

Procedure: How to Define a New Connection

To define a new connection:

1. In the left pane of Application Explorer, click the *Baan* node.
2. In the right pane, move the pointer over *Operations* and select *Define a new target*.



The Add a new target pane opens on the right. It provides three fields, followed by a help button and three action buttons.

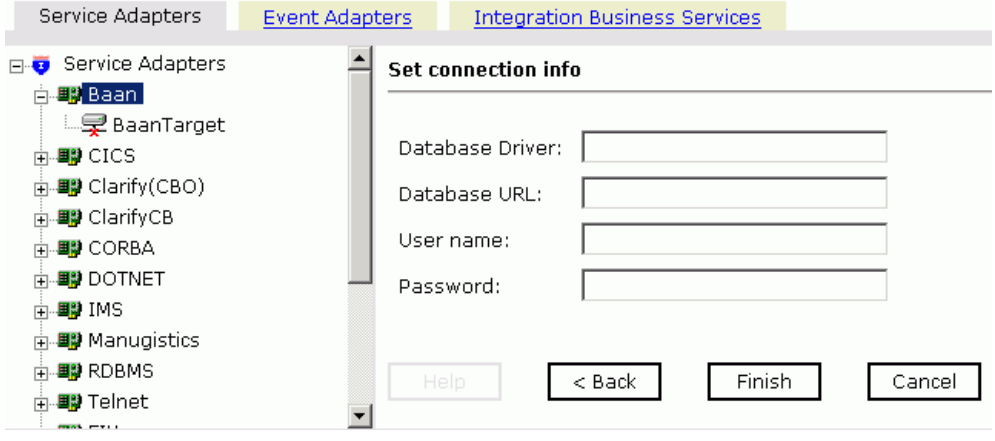
a. In the Target Name field, type a descriptive name for the target, for example, BaanTarget.

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

c. From the Target Type drop-down list, select the type of target to connect to.

3. Click *Next*.

The Set connection info pane appears on the right. The fields that appear in this pane are specific to the type of database to which you chose to connect. There are four fields followed by a help button and three action buttons.



Note: The Baan connection parameters are consistent with those found in your Baan system. For more information on parameter values that are specific to your Baan configuration, consult your Baan system administrator.

4. Enter the connection information based on the parameters in the following table:

Parameter Name	Parameter Value

Database Driver (required)	The fully-qualified name of the JDBC driver used to access the database. The name of the driver for:
	• **Oracle** is
	`oracle.jdbc.driver.OracleDriver`
	• **Informix** is
	`com.informix.jdbc.IfxDriver`
	• **Microsoft SQL Server** is
	`com.microsoft.jdbc.sqlserver.SQLServerDriver`
	• **ODBC** is
	`sun.jdbc.odbc.JdbcOdbcDriver`

Parameter Name	Parameter Value

Database URL (required)	The URL of the JDBC driver used to communicate to the database. The URL format for:
	• **Oracle** is
	`jdbc:oracle:thin:@host:port:servicename`
	for example,
	`jdbc:oracle:thin:@oracle11x:1523:VIS`
	• **Informix** is
	`jdbc:informix-sqli://[hostname]:[port/database]`
	`:informixserver=[your-sever-name]`
	for example,
	`jdbc:informix-sqli://unxsol26:5053/qaeda:informixserver=online920`
	• **Microsoft SQL Server** is
	`jdbc:microsoft:sqlserver://hostname:port[;property=value...]`
	for example,
	`jdbc:microsoft:sqlserver://server1:1433;user=test;password=secret`
	• **ODBC** is
	`jdbc:odbc:DataSrcName[;<att-name>=<att-val>]...`
	where:
	DataSrcName
	Is the name of the data source you configured for ODBC.
	att-name
	Is a data source attribute name. You can specify multiple attribute=value pairs.
	att-val
	Is the value of the attribute.
	for example,
	`jdbc:odbc:Baan`

Parameter Name	Parameter Value
User name (required)	The user ID to access the database.
Password (required)	The password that corresponds to the user ID.

Note: When connecting to the iWay server component for access to non-relational databases, the connection information must be the same for all databases in the server component system.

5. Click *Finish*.

   The new connection name appears in the left pane under the node where you created the connection. For information on connecting to the node, see *How to Open an Existing Connection* on page 2-8.

You have finished creating the new connection.

Procedure: How to Open an Existing Connection

To open an existing connection:

1. In the left pane of Application Explorer, expand the *Baan* node and select the connection you want to open.

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

   A confirmation window opens and displays the connection information.

   a. Supply a password, if required.

   b. Edit the connection parameters, if required.

3. Click *OK*.

   If the parameters are correct and the Baan or server component is available, the node displays a plus sign to indicate you are connected to that node. If not, an error message appears in the right pane.

Closing or Deleting a Connection to a Database

To manage connections, you can:

- **Close a connection** that is not currently in use.

  Although you can maintain multiple open connections, it is a good practice to close connections that are not in use.

- **Delete a connection** that is no longer required.

  You can delete a connection whether or not it is closed; if open, it automatically closes before being deleted.

Procedure: How to Close a Connection

To close a connection:

1. In the left pane of Application Explorer, select the connection you want to close.
2. Move the pointer over *Operations* and select *Disconnect* from the pop-up menu.
3. If you want to re-establish the connection, move the pointer over *Operations* and select *Connect* from the pop-up menu.

   Disconnecting from the target closes the connection, but the connection still appears in the left pane so that you can re-open it.

   The node icon no longer has the plus sign next to it, but now has a red mark below the icon to reflect that the connection is closed.

Procedure: How to Delete a Connection

To delete a connection from the list of existing connections:

1. If you have not done so, connect to a database, as described in *Opening a Connection to a Database* on page 2-4.
2. Expand the *Baan* node to view the list of connections.
3. Click the connection you want to delete.
4. In the right pane, move the pointer over *Operations* and select *Delete*.

   Alternatively, you can select *Disconnect* to close the connection while retaining it for future use.

Editing Connection Parameters

Once a target exists, you can make changes to the parameters at any time. The Edit function is only available when you select a disconnected target.

Procedure: How to Edit a Connection

To edit a connection:

1. In the left pane of the Service Adapters tab, click a disconnected node.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

   The Edit window appears in the right pane.
3. Make the desired changes to the target name or description, and Click *Next*>.

   Set connection info window appears in the right pane.
4. Make any necessary changes to the parameters found in the Set connection info window.
5. Click *Finish* to activate the changes.

Viewing Metadata

Viewing metadata is useful in determining the tables and fields to use when creating request documents. Application Explorer displays the table parameters, properties, data types, and other attributes.

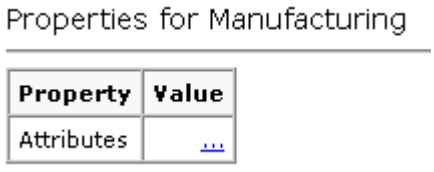
Procedure: How to View Table Metadata

To view table metadata:

1. If you have not done so, connect to a database, as described in *Opening a Connection to a Database* on page 2-4.
2. Expand the *Tables* node under the desired connection.
3. Scroll down and select the specific table to review.

   Note: Although the list of tables includes all tables in the database, the user ID specified for the connection may not have access to the specified table. If this is the case, the creation of schemas fails.

When you select a specific table, a metadata summary table appears in the right pane, as shown in the following figure. The table has two columns labeled property and value. An ellipsis appears in the value column to indicate there is additional information.



Click the ellipsis symbol in the row you want to review to display the information it holds.

The properties appear in the right pane.

Example Request, Response, and Event Documents

You can generate request document schemas using Application Explorer. You can generate request document *instances* using a third party XML tool and submit those documents to the Baan or iWay agent.

This topic provides examples of request and response XML documents, as well as an example of an event document.

Request Document Examples

This topic provides examples of request document that are based on schemas created for a request XML file.

Example: GET Item Request Document

The following is an example of a request document to retrieve an item from Baan.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SYNC_ITEM_005>
<CNTROLAREA>
 <BSR>
  <VERB value="GET">GET</VERB>
  <NOUN value="ITEM">ITEM</NOUN>
  <REVISION value="005" />
 </BSR>
 <SENDER>
  <LOGICALID />
  <COMPONENT />
  <TASK />
  <REFERENCEID />
  <CONFIRMATION />
  <LANGUAGE />
  <CODEPAGE />
  <AUTHID />
 </SENDER>
 <DATETIME qualifier="ACCOUNTING" type="T" index="">
  <YEAR />
  <MONTH />
  <DAY />
  <HOUR />
  <MINUTE />
  <SECOND />
  <SUBSECOND />
  <TIMEZONE />
 </DATETIME>
</CNTROLAREA>
<DATAAREA>
 <SYNC_ITEM>
  <ITEMHEADER>
    <ITEM>9999</ITEM>
  </ITEMHEADER>
 </SYNC_ITEM>
</DATAAREA>
</SYNC_ITEM_005>
```

Example: SYNC Item Request Document

The following is an example of a request document to update an item in Baan.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SYNC_ITEM_005>
 <CNTROLAREA>
  <BSR>
   <VERB value="SYNC">SYNC</VERB>
   <NOUN value="ITEM">ITEM</NOUN>
   <REVISION value="005" />
  </BSR>
  <SENDER>
   <LOGICALID />
   <COMPONENT />
   <TASK />
   <REFERENCEID />
   <CONFIRMATION />
   <LANGUAGE />
   <CODEPAGE />
   <AUTHID />
  </SENDER>
  <DATETIME qualifier="ACCOUNTING" type="T" index="">
   <YEAR />
   <MONTH />
   <DAY />
   <HOUR />
   <MINUTE />
   <SECOND />
   <SUBSECOND />
   <TIMEZONE />
  </DATETIME>
 </CNTROLAREA>
 <DATAAREA>
  <SYNC_ITEM>
   <ITEMHEADER>
    <ITEM>9999</ITEM>
    <ITEMDESC>Change Desc</ITEMDESC>
   </ITEMHEADER>
  </SYNC_ITEM>
 </DATAAREA>
</SYNC_ITEM_005>
```

Successful Response Document Examples

This topic provides examples of successful response documents that are returned from Baan.

Example: GET Item Response Schema

The following is an example of a successful response document that is returned from Baan when you execute a GET method.

item_get_response.xml

```
<?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>
   <selectResponse xmlns="urn:iwaysoftware:ibse:jul2003:select:response"
     cid="EE87E6EB69CA5B5DD3065787C291D4B3">
     <SYNC_ITEM_005 xmlns:SOAP-
       ENV="http://schemas.xmlsoap.org/soap/envelope/">
       <CNTROLAREA>
        <BSR>
         <VERB value="GET">GET</VERB>
         <NOUN value="ITEM">ITEM</NOUN>
         <REVISION value="005" />
        </BSR>
        <SENDER>
         <LOGICALID />
         <COMPONENT />
         <TASK />
         <REFERENCEID />
         <CONFIRMATION />
         <LANGUAGE />
         <CODEPAGE />
         <AUTHID />
        </SENDER>
        <DATETIME index="" type="T" qualifier="ACCOUNTING">
         <YEAR />
         <MONTH />
         <DAY />
         <HOUR />
         <MINUTE />
         <SECOND />
         <SUBSECOND />
         <TIMEZONE />
        </DATETIME>
      </CNTROLAREA>
```

```
    <DATAAREA>
      <SYNC_ITEM>
       <ITEMHEADER>
        <ITEM>9999</ITEM>
        <ITEMDESC>Change Desc</ITEMDESC>
        <QUANTITY qualifier="WEIGHT">0.0</QUANTITY>
        <COMMODITY index="2">901</COMMODITY>
        <UOM>pcs</UOM>
        <UNITTYPE>pcs</UNITTYPE>
        <LOTSNFLAG>3</LOTSNFLAG>
        <QUANTITY qualifier="SHELFLIFE">
        0
         <UOM>4</UOM>
        </QUANTITY>
        <PROPERTY index="3">2</PROPERTY>
        <PROPERTY index="5">2</PROPERTY>
        <ITEMVALUE>
         <OPERAMT type="F" qualifier="UNIT">
         0.0
         <CURRENCY>USD</CURRENCY>
        </OPERAMT>
        <OPERAMT type="T" qualifier="UNIT">
         <UOM>pcs</UOM>
        </OPERAMT>
        <DATETIME qualifier="EFFECTIVE">1.0</DATETIME>
       </ITEMVALUE>
       <PROPERTY index="6">2</PROPERTY>
       <PROPERTY index="8">2</PROPERTY>
      </ITEMHEADER>
     </SYNC_ITEM>
    </DATAAREA>
   </SYNC_ITEM_005>
  </selectResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Unsuccessful Response Document Examples

This topic provides examples of unsuccessful response documents that are returned from Baan when a request was unable to execute.

Example: DELETE Customer Response Failure

The following is an example of an error document that is returned from Baan when a DELETE method is executed.

customer_delete_error.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<eda>
  <error cnctag="" code="-100" source="process" timestamp="TIME-QA-QA
  QA:QA:QA">Problem processing agent request, type FAIL, Source AGENT:
  Error: No rows affected</error>
<data>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
.
.
.
.
</data>
</eda>
```

Example: GET Customer Response Failure

The following is an example of an error document that is returned from Baan when a GET method is executed.

customer_get_failure.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<eda>
  <error cnctag="" code="-100" source="process" timestamp="TIME-QA-QA
  QA:QA:QA">Problem processing agent request, type FAIL, Source AGENT:
  Error : Select returned 0 rows</error>
<data>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
.
.
.
.
</data>
</eda>
```

Example: INSERT Customer Response Failure

The following is an example of an error document that is returned from Baan when an INSERT method is executed.

customer_insert_failure.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<eda>
<error cnctag="" code="-100" source="process" timestamp="TIME-QA-QA
  QA:QA:QA">Problem processing agent request, type FAIL, Source AGENT:
  Error: Failed to execute query, java.sql.SQLException: [Microsoft][ODBC
  SQL Server Driver][SQL Server]Attempt to insert duplicate key row in
  object 'ttccom010550' with unique index 'Ittccom010550_1a'</error>
<data>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
.
.
.
.
</data>
</eda>
```

Example: SYNC Customer Response Failure

The following is an example of an error document that is returned from Baan when a SYNC method is executed.

customer_sync_failure.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<eda>
<error cnctag="" code="-100" source="process" timestamp="TIME-QA-QA
  QA:QA:QA">Problem processing agent request, type FAIL, Source AGENT:
  Error: No rows affected</error>
<data>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
.
.
.
.
</data>
</eda>
```

Event Document Example

This topic provides a sample event document.

Example: Event Output Document

The following is an example of an event triggered in Baan.

```
<JDBCODBC>
  <row>
    <event_id>221</event_id>
    <object>Customer</object>
    <object_id>999</object_id>
    <table_name>TTCCOM010550</table_name>
    <key_field>t_cuno</key_field>
    <verb>Insert</verb>
    <user_name>sa</user_name>
  </row>
  <row>
    <event_id>222</event_id>
    <object>Customer</object>
    <object_id>1234</object_id>
    <table_name>TTCCOM010550</table_name>
    <key_field>t_cuno</key_field>
    <verb>Insert</verb>
    <user_name>sa</user_name>
  </row>
  <row>
    <event_id>223</event_id>
    <object>Item</object>
    <object_id>99991</object_id>
    <table_name>TTIITM001550</table_name>
    <key_field>t_item</key_field>
    <verb>Update</verb>
    <user_name>sa</user_name>
   </row>
   <row>
    <event_id>235</event_id>
    <object>Inventory</object>
    <object_id>900</object_id>
    <table_name>TTDINV001550</table_name>
    <key_field>t_cwar</key_field>
    <verb>Delete</verb>
    <user_name>sa</user_name>
   </row>
</JDBCODBC>
```

Generating a Business Service for an XML Request Document

You can generate a business service (also known as a Web service) for an XML request document. To generate a Web service for an SQL statement for a relational or non-relational database, you must deploy the iWay Adapter for Baan for BEA WebLogic in a business services environment using the servlet version of the Integration Business Services Engine (iBSE).

Ensure you properly configure the servlet iBSE. For more information on installing and deploying iWay components, see the *iWay 5.5 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 *iWay 5.5 Installation and Configuration* manual and the i*Way Connector for JCA for BEA WebLogic Server User's Guide, Version 5 Release 5*.

Procedure: How to Generate a Business Service for an XML Request Document

To generate a business service for an XML request document for a relational or non-relational database:

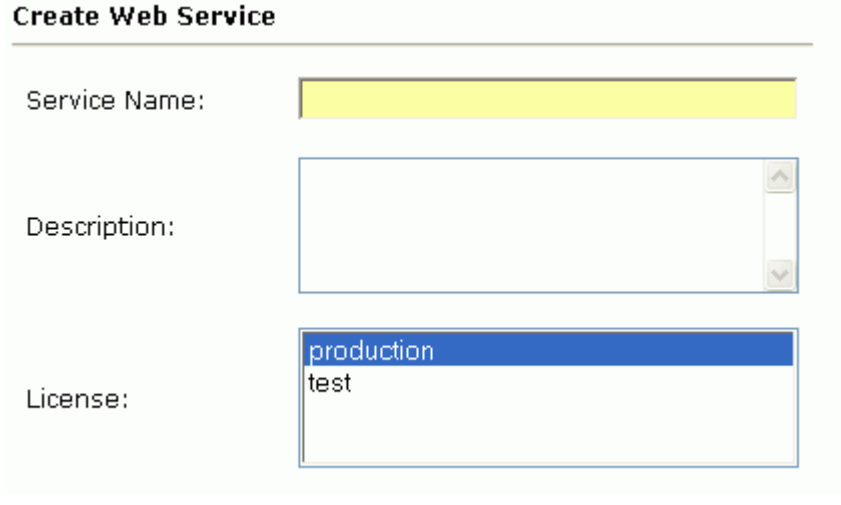
1. If you have not done so, connect to a database, as described in *Opening a Connection to a Database* on page 2-4.
2. Expand the node to view the available methods.
3. Click the method for which you want to create a business service.
4. In the right pane, move the pointer over *Operations* and select *Create Integration Business Service*.

The Create Web Service information appears in the right pane. This pane provides two radio buttons that allow you to select either a new or existing service, a help button, and three action buttons.

5. Choose to either *Create a new service* or *Use an existing service* by selecting the appropriate radio button.
6. Click *Next*>.

   If you select *Use an existing service*, a drop-down list appears. Select the business service to which you want to add the new service.

   If you select *Create a new service*, the Create Web Service pane appears providing three fields to enter information about the new service.



   a. In the Service Name field, type a name to identify the Web service.

      This name will appear 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 the business service. To select more than one, hold down the *Ctrl* key and click the licenses.

7. Click *Next*.

   The right pane displays the next Create Web Service pane that prompts you for information about the method of the service.

a. In the Method Name field, type a name to identify the method to add to the business service.

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

8. Click *Finish*.

Application Explorer switches the view to the Integration Business Services tab, and the new business service appears in the left pane. The test pane for the method opens in the right pane. This pane is described in the *Testing a Business Service* section.

Testing a Business Service

After a business service is created, test it to ensure it functions properly. iWay provides a test tool for testing the business service.

Procedure: How to Test a Business Service

To test a business service:

1. Click the *Integration Business Services* tab of Application Explorer to access business services.

2. Expand the list of business services under Integration Business Services.

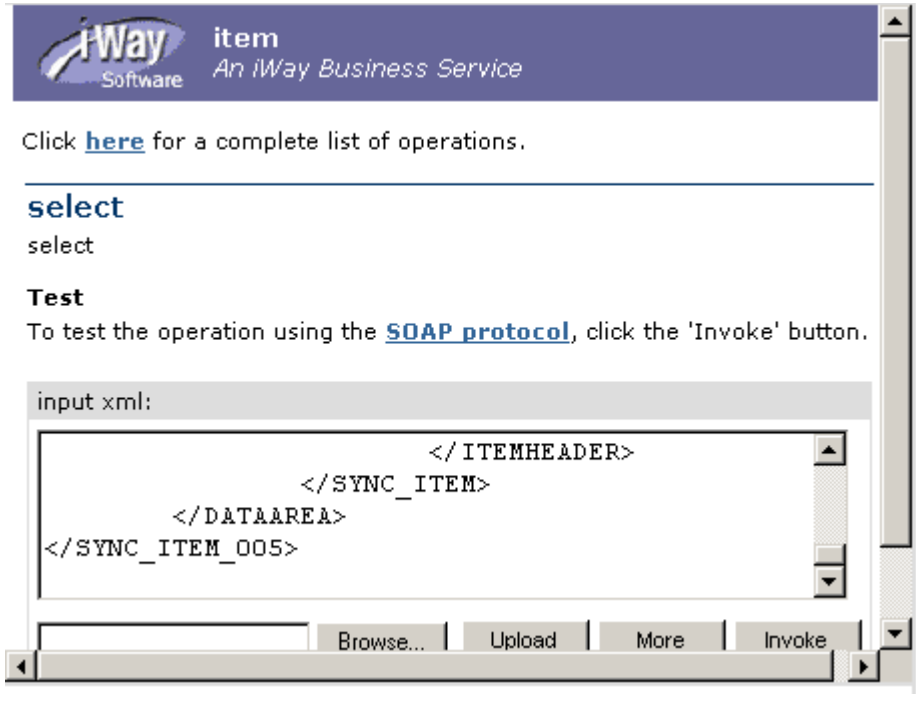
3. Expand the *Services* node.

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

4. Select the name of the business service you want to test.

The test option appears in the right pane.

If you are testing a Web service that requires XML input, an input xml field appears in the test pane. You can paste the XML input or browse to an XML file. Below the XML text field is the browse field and three action buttons.

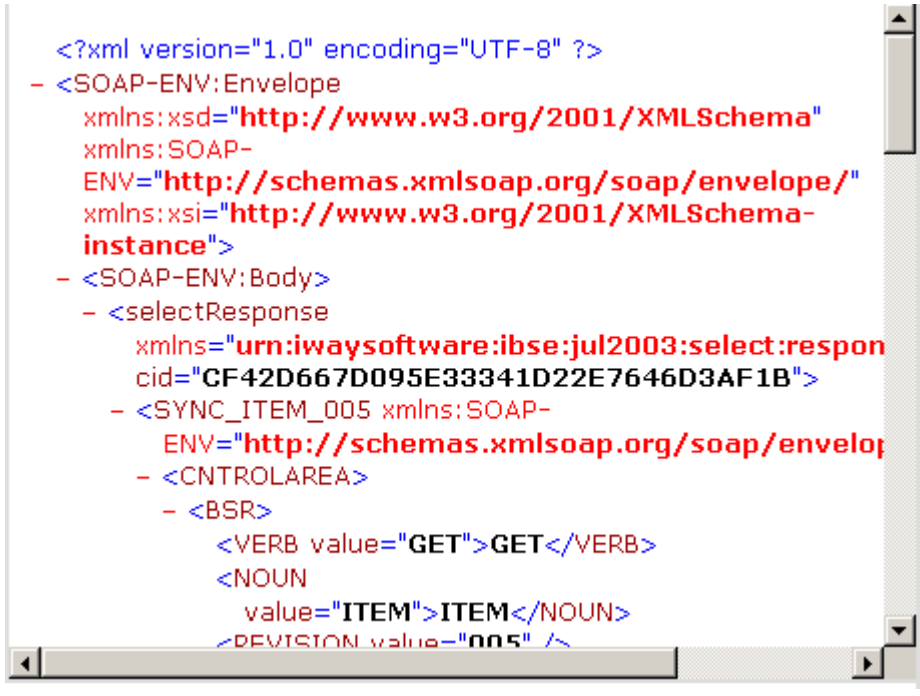


5. Provide the appropriate input to the input xml text area using copy and paste from an XML file, or browse to the XML file and click *Upload*.

   The XML file content is displayed in the XML input area of the test pane.

6. Click *Invoke*.

Application Explorer displays the test results in the window in the right pane. The following is an example of test results.



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 Baan. The user name and password values that you provided for Baan 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>
```

Generating WSDL From a Web Service

Generating Web Services Description Language (WSDL) from a Web service enables you to make the Web service available to other services within a host server such as the BEA WebLogic Server.

Procedure: How to Generate WSDL From a Web Service

To generate WSDL from a Web service:

1. If you are not already in the Integration Business Services tab, click the tab to access it.
2. In the left pane, expand the list of services to view the service for which you want to generate WSDL.
3. Click the service.

   The link for the service appears in the right pane.
4. Right-click the link and choose *Save Target As* from the pop-up menu.
5. Choose a location for the file and add a .wsdl file extension.
6. Click *Save*.

   For example, saving a Web service called PMS for a VSAM database creates a file named PMS.wsdl.

   Note: The file extension must be .wsdl.

Example: Viewing WSDL Generated from a Web Service

The following is an example of a WSDL file for a Web service called PMS created from a parameterized SQL statement against a VSAM database.

```
<definitions xmlns:tns="urn:schemas-iwaysoftware-com:iwse"
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:VSAM:response"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:VSAM"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"><types><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="ibsinfo"><xs:complexType><xs:sequence><xs:element type="xs:string"
name="service"/><xs:element type="xs:string" name="method"/><xs:element
type="xs:string" name="license"/><xs:element type="xs:string"
minOccurs="0" name="disposition"/><xs:element type="xs:string"
minOccurs="0" name="Username"/><xs:element type="xs:string" minOccurs="0"
name="Password"/><xs:element type="xs:string" minOccurs="0"
name="language"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="adapterexception"><xs:complexType><xs:sequence><xs:element
type="xs:string"
name="error"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:iwaysoftware:ibse:jul2003:VSAM"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:VSAM"
elementFormDefault="qualified"><xs:element
name="VSAM"><xs:complexType><xs:sequence><xs:element type="xs:string"
name="emp_id"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema

targetNamespace="urn:iwaysoftware:ibse:jul2003:VSAM:response"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:VSAM:response"
elementFormDefault="qualified"><xs:element
name="VSAMResponse"><xs:complexType><xs:sequence><xs:element
name="RESULT"><xs:complexType><xs:sequence><xs:element
name="PMSVSAM"><xs:complexType><xs:sequence><xs:element minOccurs="0"
name="RESULTSET_1"><xs:complexType><xs:sequence><xs:element minOccurs="0"
name="ROW" maxOccurs="unbounded"><xs:complexType><xs:sequence><xs:element
type="xs:string" name="EMP_ID"/><xs:element type="xs:string"
```

```
name="FIRST_NAME"/><xs:element type="xs:string"
name="LAST_NAME"/><xs:element type="xs:string" name="DEPT"/><xs:element
type="xs:string"
name="COMP_NAME"/></xs:sequence></xs:complexType></xs:element></xs:sequen
ce></xs:complexType></xs:element></xs:sequence></xs:complexType></xs:elem
ent></xs:sequence></xs:complexType></xs:element>
</xs:sequence><xs:attribute type="xs:string" use="required"
name="cid"/></xs:complexType></xs:element></xs:schema>   </types><message
name="VSAMIn"><part element="m1:VSAM" name="parameters"/>
</message><message name="VSAMOut"><part element="m11:VSAMResponse"
name="parameters"/>   </message><message name="PMSHeader"><part
element="tns:ibsinfo" name="header"/>   </message><message
name="AdapterException"><part element="tns:adapterexception"
name="fault"/>
   </message><portType name="PMSSoap"><operation
name="VSAM"><documentation/><input message="tns:VSAMIn"/><output
message="tns:VSAMOut"/><fault message="tns:AdapterException"
name="AdapterExceptionFault"/></operation>
   </portType><binding type="tns:PMSSoap" name="PMSSoap"><soap:binding
style="document"
transport="http://schemas.xmlsoap.org/soap/http"/><operation
name="VSAM"><soap:operation style="document"
soapAction="PMS.VSAMRequest@production@@"/><input><soap:body
use="literal"/><soap:header part="header" message="tns:PMSHeader"
use="literal"/>
         </input><output><soap:body use="literal"/>
         </output><fault name="AdapterExceptionFault"><soap:fault
use="literal" name="AdapterExceptionFault"/></fault></operation>
   </binding><service name="PMS"><documentation>PMS</documentation><port
binding="tns:PMSSoap" name="PMSSoap1"><soap:address
location="http://iwayntk1:7001/ibse/IBSEServlet/XDSOAPRouter"/></port></s
ervice></definitions>
```

Deleting a Business Service

If a business service is no longer required you can delete it from the Services list.

Procedure: How to Delete a Business Service

To delete a business service:

1. Click the *Integration Business Services* tab of Application Explorer, and expand the *Integration Business Services* node.
2. Select the *Services* node.

   The right pane displays a list of available services.
3. Select the check box next to the service(s) you want to delete.

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

   A confirmation message appears in the right pane.

5. Click *OK* to delete the service(s).

CHAPTER 3

Listening for Database Events

Topics:

- Understanding Event Functionality
- Creating an Event Port
- Creating a Channel

This section describes how to use the iWay Adapter for Baan for BEA WebLogic, deployed to an application server such as BEA WebLogic Server, to listen for events in a relational table. Several listening techniques are available, enabling you to choose the technique that best suits your requirements.

Understanding Event Functionality

Events are generated as a result of activity in a database or application 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 Servlet Application Explorer. To create an event, you must create a:

- **Port**

  A port associates a particular business object exposed by the iWay Adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.

- **Channel**

  A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the iWay Adapter. For more information, see *Creating a Channel* on page 3-14.

Creating an Event Port

The following procedures describe how to create an event port using Servlet Application Explorer. You can create a port from the Service Adapters tab or from the Event Adapters tab.

When you use Application Explorer with an Integration Business Services Engine (iBSE) implementation, the following port dispositions are available:

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

When you use Application Explorer with a JCA implementation, the following port dispositions are available:

- File
- HTTP
- JMS
- MQ Series

Procedure: How to Create an Event Port for File

To create an event port for File using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

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

   d. In the Disposition 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=errorDest]
      ```

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

      ```
      location
      ```

The following table describes the disposition parameters.

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.

For example:

```
ifile://c:\temp\BaanEvent.txt;errorTo=ifile://c:\temp\error
```

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 3-14.

Procedure: How to Create an Event Port for iBSE

You can call a Web Service created through iBSE. To create an event port for iBSE using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

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

   d. In the Disposition field, enter an iBSE destination using the following format:

      `ibse:/svcName.methName[;responseTo=respDest][;errorTo=errorDest]`

      The following table describes the disposition parameters.

Parameter	Description
svcName	Name of the service created with iBSE.
methName	Name of the method created for the Web service.
respDest	Location where responses to the Web service are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

Procedure: How to Create an Event Port for MSMQ

To create an event port for a MSMQ using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

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

3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port window opens in the right pane.

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

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

   b. In the Description field, type a brief description of the port you are creating, for example, "Queue1 created for VSAM output".

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

   d. In the Disposition field, enter an MSMQ destination using the format:

      ```
      msmq:/host/queueType/queueName[;errorTo=errorDest]
      ```

      The following table defines the disposition parameters.

Parameter	Description
host	Name of the host on which the Microsoft Queuing system runs.
queueType	Type of queue. For private queues, enter *Private$*. 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.
queueName	Name of the queue where messages are placed.
errorDest	Location to which error logs are sent. Optional. Can be a predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

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

   You are now ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

Procedure: How to Create an Event Port for JMS Queue

To create an event port for a JMS queue using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

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

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

      When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

      ```
      jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
         user=userID;password=pass[;errorTo=errorDest]
      ```

      When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

      ```
      jms:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory
      ```

      The following table describes the disposition parameters.

Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource which contains information about the JMS Server. The WebLogic connection factory is: `javax.jms.QueueConnectionFactory`

Parameter	Description
jndi_url	The URL of the application server. For BEA WebLogic Server, jndi_url is `t3://host:port` where: *host* Is the machine name where WebLogic Server is installed. *port* Is the port on which WebLogic server is listening. The default port, if not changed at installation, is 7001.
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY, provided by the JNDI service provider. For WebLogic Server, the WebLogic factory is `weblogic.jndi.WLInitialContextFactory.`
userID	A user ID associated with this queue.
pass	The password for this user ID.
errorDest	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.

5. Click *OK*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

Procedure: How to Create a Port for the SOAP Disposition

To create a port for a SOAP disposition using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

   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=action[;responseTo=respDest]
         [;errorTo=errorDest]
      ```

      The following table describes the disposition parameters.

Parameter	Description

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

Parameter	Description

soapaction	The method that will be called by the SOAP disposition. For example:
	`webservice.method@test@@`
	where:
	`webservice`
	Is the name of the Web service you created using Application Explorer.
	`method`
	Is the method being used.
	`test`
	Is the license that is being used by the Web service.
	This value can be found by navigating to the iWay Business Services tab and opening the *Service Description* link in a new window. Perform a search for *soapAction*.
	You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.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*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

Procedure: How to Create an Event Port for HTTP Disposition

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

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

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

   d. In the Disposition field, enter an HTTP destination.

      When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

      ```
      ihttp://url;responseTo=respDest
      ```

      When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

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

      The following table describes the disposition parameters.

Parameter	Description
url	URL target for the post operation.

Parameter	Description
respDest	Location where responses are posted. Can be a predefined port name or another full URL. Optional. Can be a predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the host on which the Web server resides.
port	Port number on which the Web server is listening.
uri	Universal resource identifier that completes the url specification.

5. Click *OK*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

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

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

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *baan* 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 connection.

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

   b. In the Description field, type a brief description of the port you are creating.

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

   d. In the Disposition field, enter an MQ Series destination.

      When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;
   channel=chanName[;errorTo=errorDest]
```

When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

```
mq:/qManager/qName;host=hostName;port=portNum;
   channel=chanName
```

The following table describes the disposition parameters.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName	Name of the queue where messages are to be placed.
hostName	Name of host on which MQ Series resides (for the MQ client only).
portNum	Port number for connecting to an MQ Server queue manager (for the MQ client only).
chanName	Case-sensitive name of the channel that connects to the remote MQ Server queue manager (for the MQ client only). The default MQ Series channel name is SYSTEM.DEF.SVRCONN.
errorDest	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.

5. Click *OK*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-14.

Editing or Deleting an Event Port

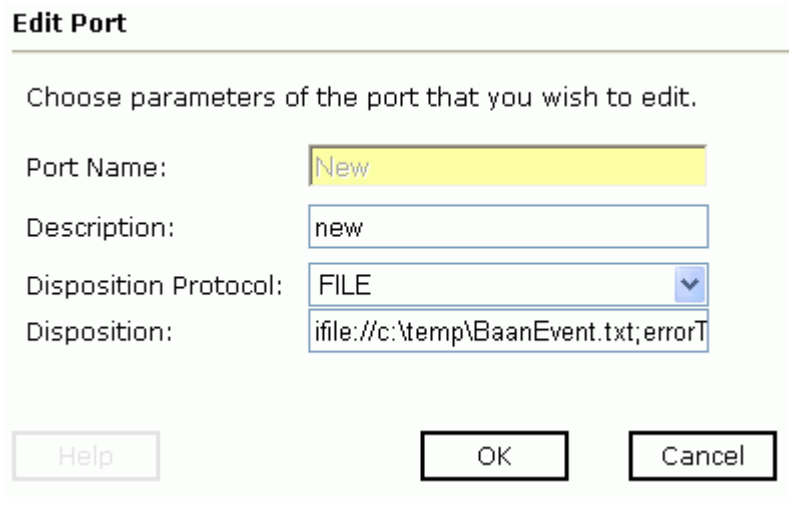
The following procedures explain how to modify or delete an event port.

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 Edit Port pane opens. This pane provides four fields, a help button, and two action buttons.



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 procedure describes how to create a channel for your event using Servlet Application Explorer. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

To create a channel using Application Explorer:

1. Click the *Event Adapters* tab.

The iWay Adapters that support events appear in the left pane of this tab.

2. Expand the *iWay Adapter* node.

   The ports and channels nodes appear in the left pane.

3. Expand the *baan* node, then click the *channels* node.

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

   The Add a new Baan channel pane opens. This pane provides three fields, a help button, and three action buttons.

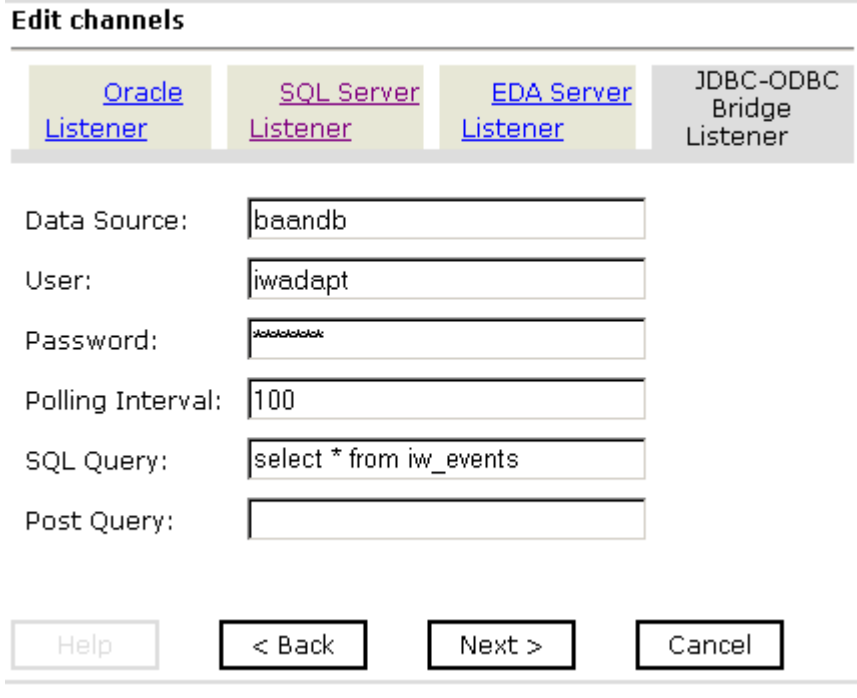
   a. In the Channel Name field, type a name for the channel, for example, NewChannel.

   b. In the Description field, type a brief description of the channel you are creating.

   c. From the Channel Type drop-down list, select a channel type (for example, Table Listener).

5. Click *Next*.

   The Edit channels window opens in the right pane showing four tabs that represent the types of available listeners.



The Edit channels window opens in the right pane. This pane provides four tabs that represent the available listeners. They are Oracle, SQL Server, EDA Server, or JDBC-ODBC listener. Each tab displays parameter fields related to the specific listener. These fields are described in the following steps.

Note: If you are configuring listening capabilities for a non-relational database, select an EDA Server listener.

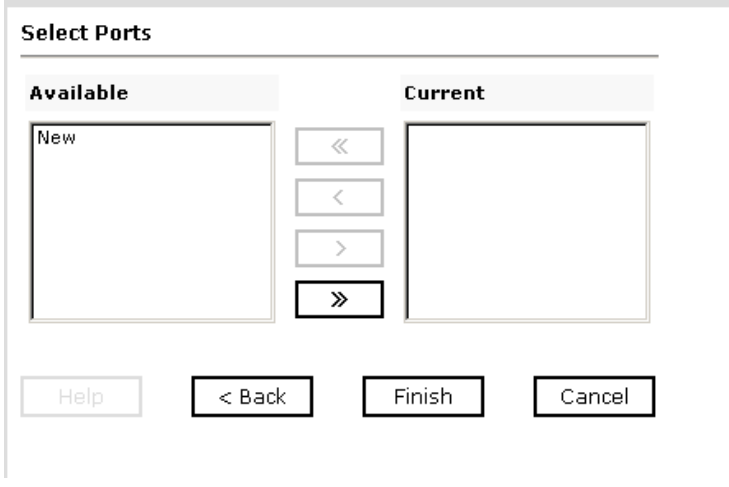
a. Select a listener by clicking the corresponding tab.

b. Enter the system information that is specific to the database on which you are listening based on the descriptions in the following table.

Parameter Name	Parameter Value
Host	Name of the system where the database resides.
Port	Port number where the system listens.
SID	The database service, which acts as a pointer to the database. **Note:** This parameter is available only when configuring an Oracle listener.
Data Source	The database name.
Database Name	The database name.
User	A user ID to access the database.
Password	A valid password that corresponds to the user ID.
Polling Interval	A value, in seconds, at which to check for new input.
SQL Query	An SQL Query, for example: `select * from iw_events`
Post Query	Leave blank. Entering a value may corrupt data.

6. Click *Next*.

The Select Ports window opens in the right pane. A list of available ports appear in Available field on the left, and the ports that are currently associated appear in the Current field on the right. This pane contains a series of buttons to transfer ports between the Available and Current panes, a help button, and three action buttons.



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

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

7. Click *Finish*.

The summary window opens.

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

The channel also 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.

Procedure: How to Start a Channel

To start a channel:

1. In the Event Adapters tab of Application Explorer, expand the *baan* node located in the left pane and select the channel you want to start.
2. 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 in the left pane disappears.
3. 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 of the Event Adapters tab, 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 of the Event Adapters tab, select the channel you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

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

   The channel disappears from the list in the left pane.

CHAPTER 4

Using Web Services Policy-Based Security

Topics:

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

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

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

iWay Business Services Policy-Based Security

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

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

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

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

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

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

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

Configuring iWay Business Services Policy-Based Security

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

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

You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 4-10.

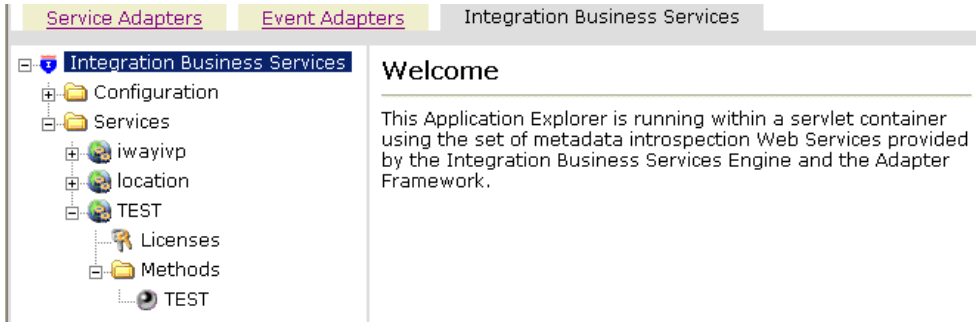
Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

Procedure: How to 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 iWay Business Services. The iWay Business Services tab is active and displays a Welcome screen on the right. The image shows the iWay Business Services node expanded in the left pane.



   a. Click the *iWay Business Services* tab.

   b. Expand the *Configuration* node.

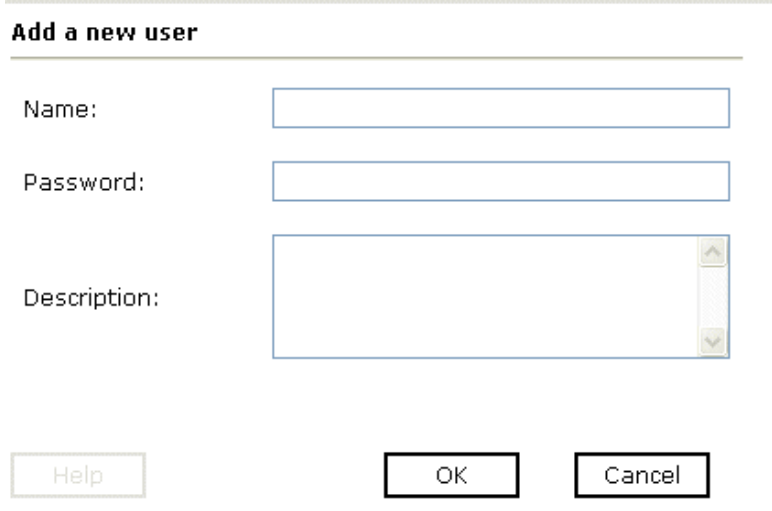
   c. Expand the *Security* node.

   d. Expand the *Users and Groups* node.

e. Select *Users*.

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

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



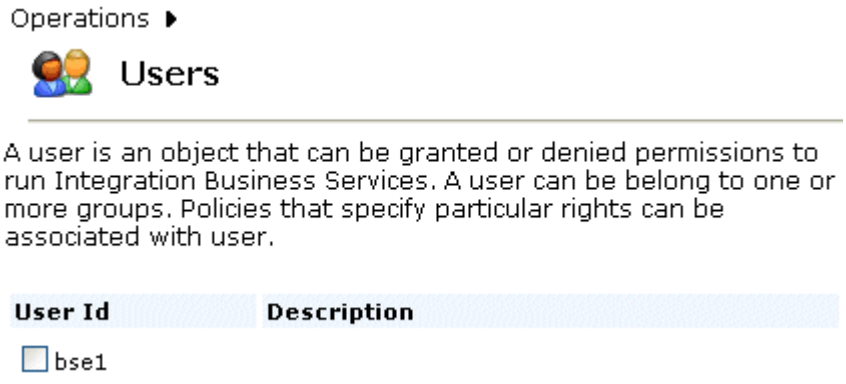
a. In the Name field, type a user ID.

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

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

3. Click *OK*.

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

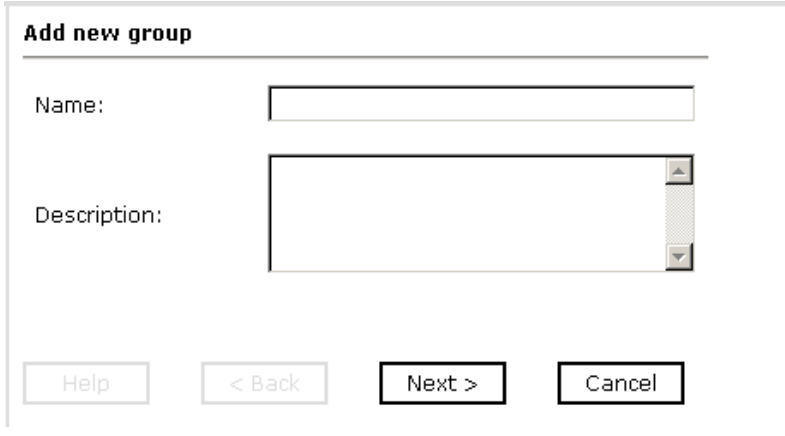


Procedure: How to How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

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

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

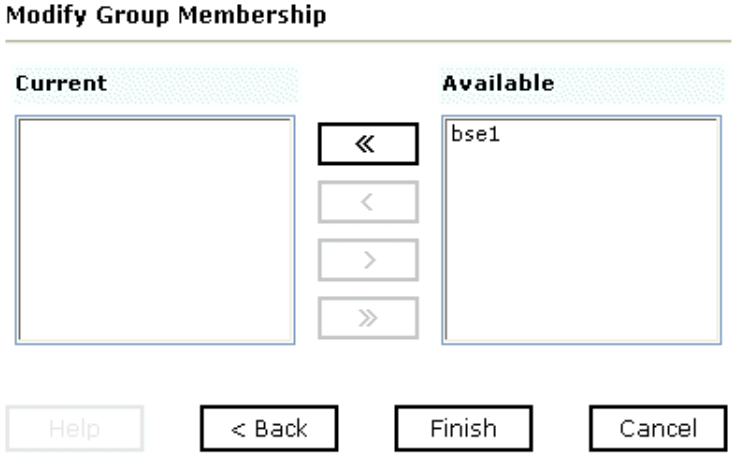


a. In the Name field, type a a name for the group.

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

3. Click *Next*.

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

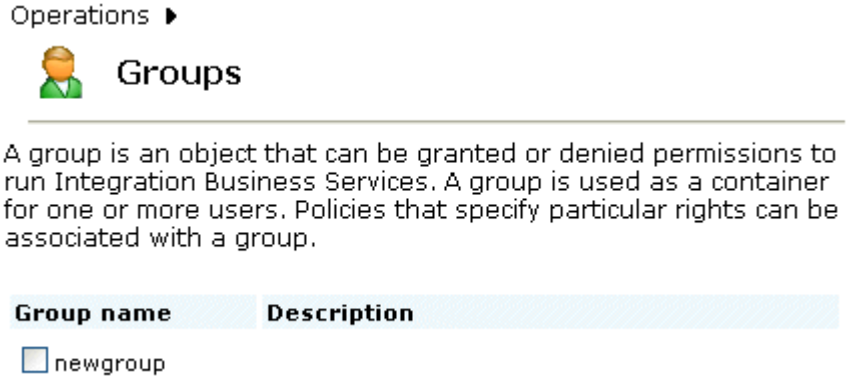


You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

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

The new group is added.

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



Procedure: How to How to Create an Execution Policy

To create an execution policy:

1. Open *Servlet Application Explorer*.

a. Click the *iWay Business Services* tab.

b. Expand the *Configuration* node.

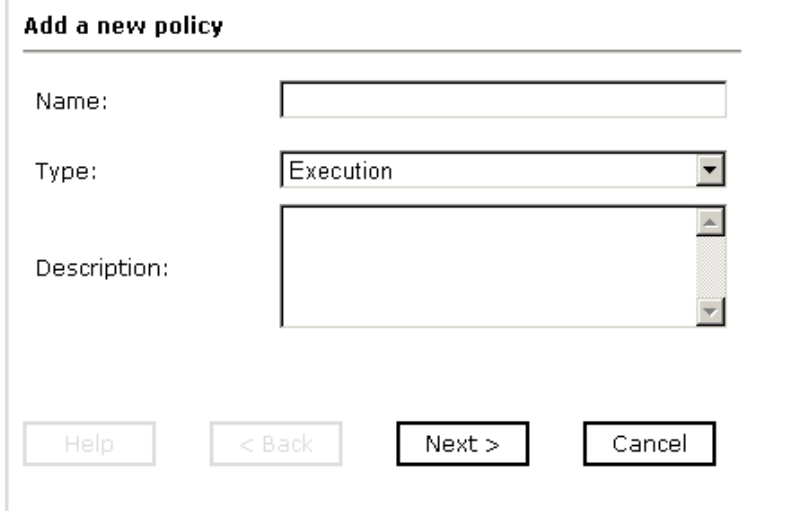
c. Select *Policies*.

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

o

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.



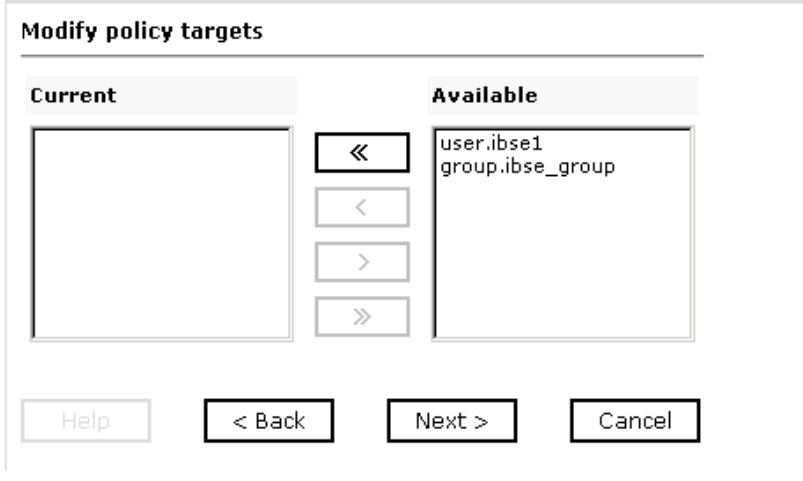
a. In the Name field, type a a name for the policy.

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.

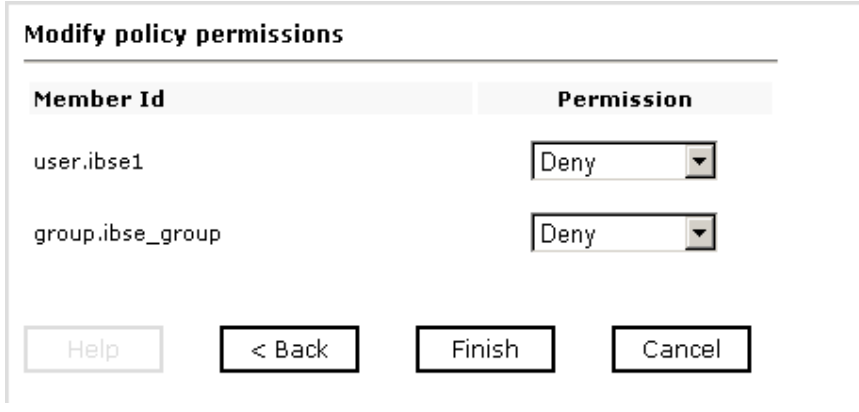


4. Select a minimum of one user or group from the Available pane.

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.

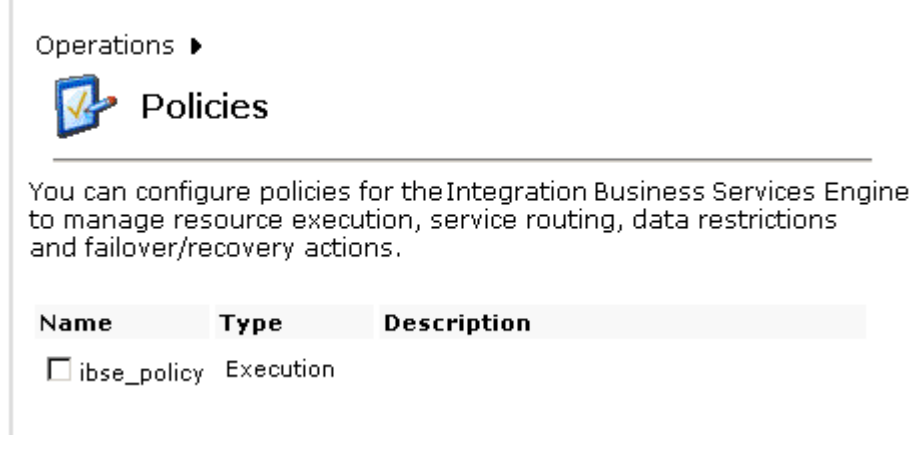
5. Click *Next*.

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.



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.



Procedure: How to How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

1. Open *Servlet Application Explorer.*

a. Select the *iWay Business Services* tab.

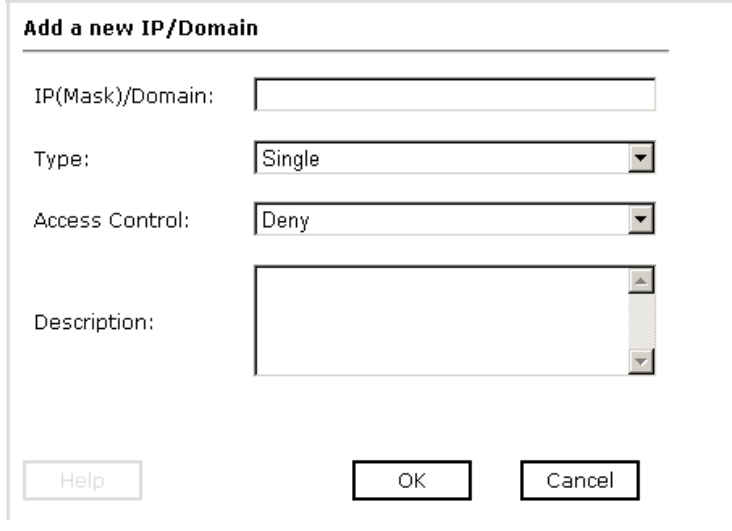
b. Expand the *Configuration* node.

c. Expand the *Security* node.

d. Select *IP and Domain*.

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

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



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.

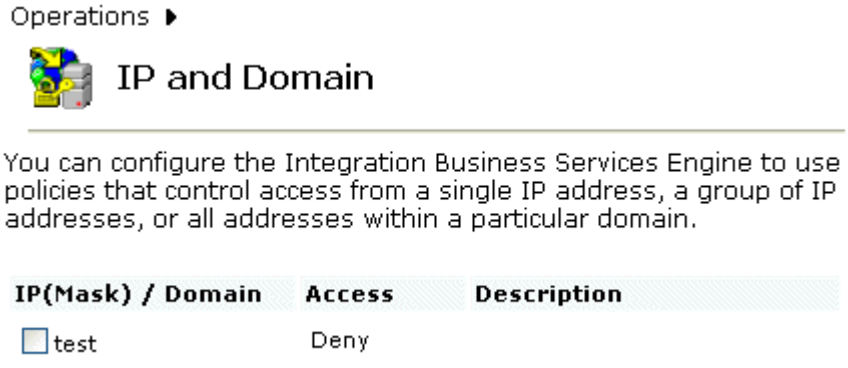
If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

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).



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
- Configuring Connection Pool Sizes
- Migrating Repositories
- Exporting or Importing Targets
- Retrieving or Updating Web Service Method Connection Information
- Starting or Stopping a Channel Programmatically

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

Managing and Monitoring Services and Events Using iBSE

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

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that your 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.

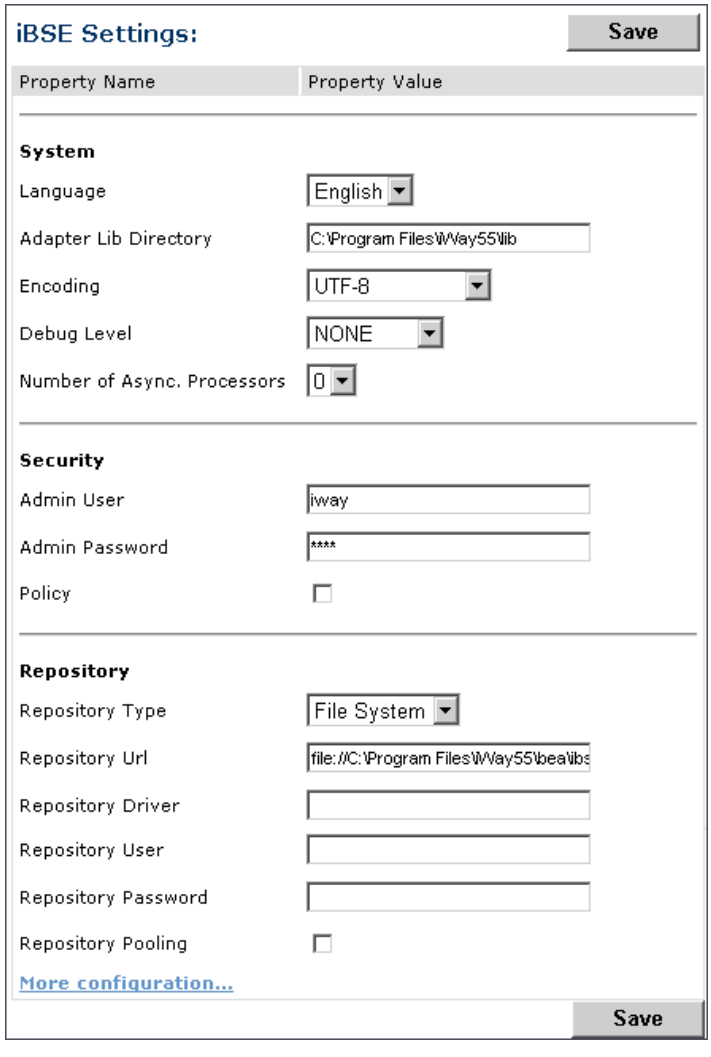
   port

   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.



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:

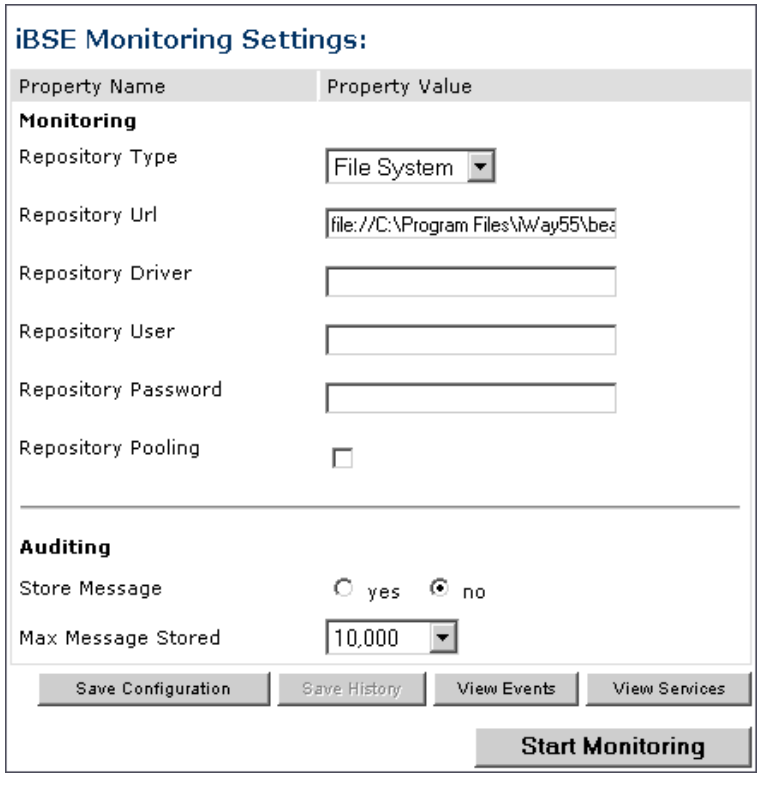
localhost

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.



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

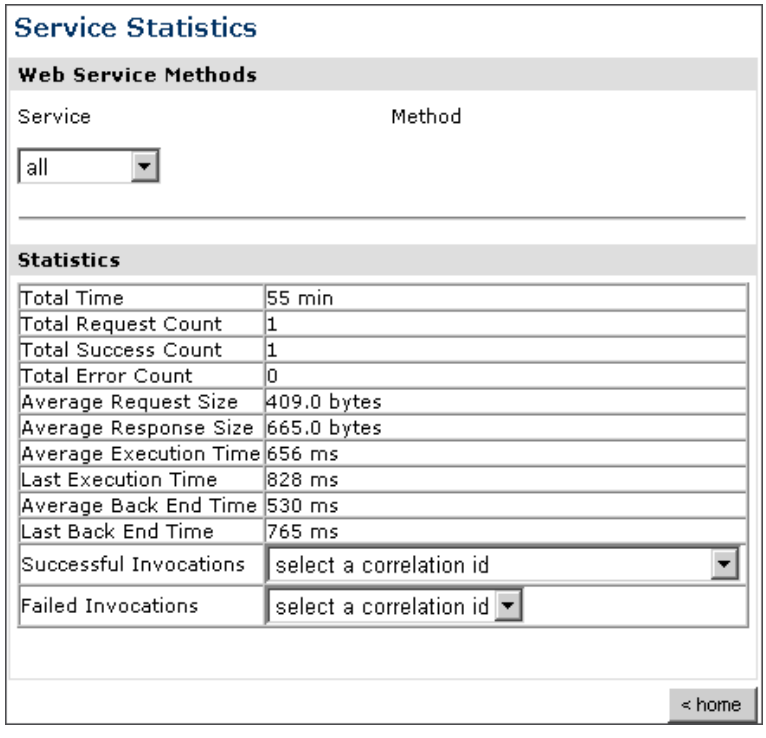
To monitor services:

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.



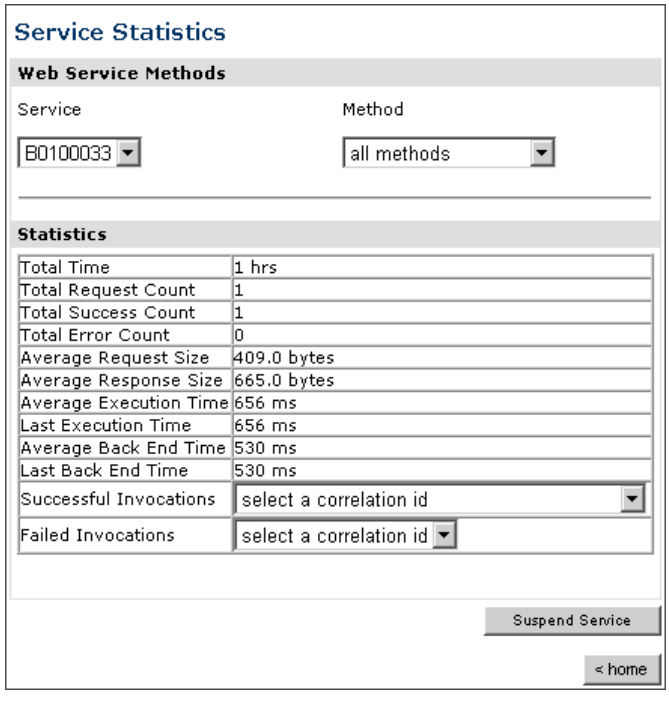
The system level summary provides services statistics at a system level.

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

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

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

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

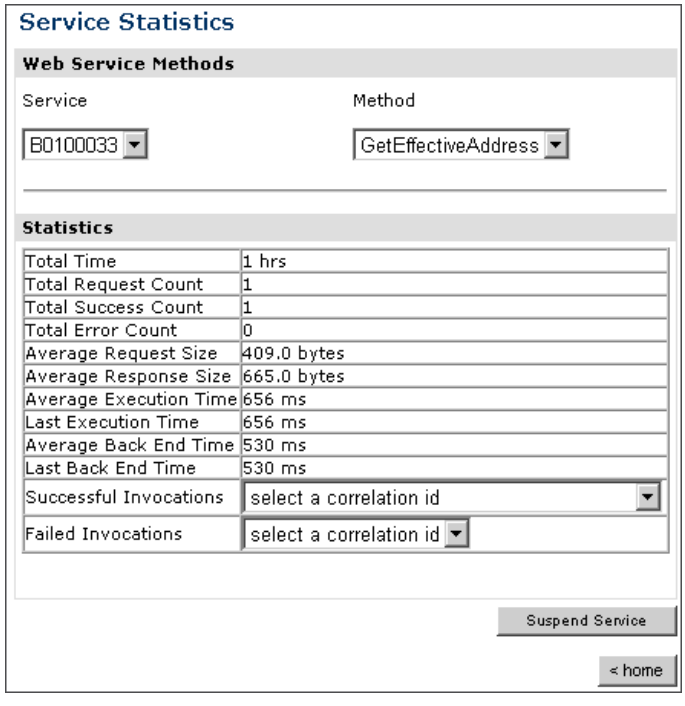


a. To stop a service at any time, click *Suspend Service*.

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

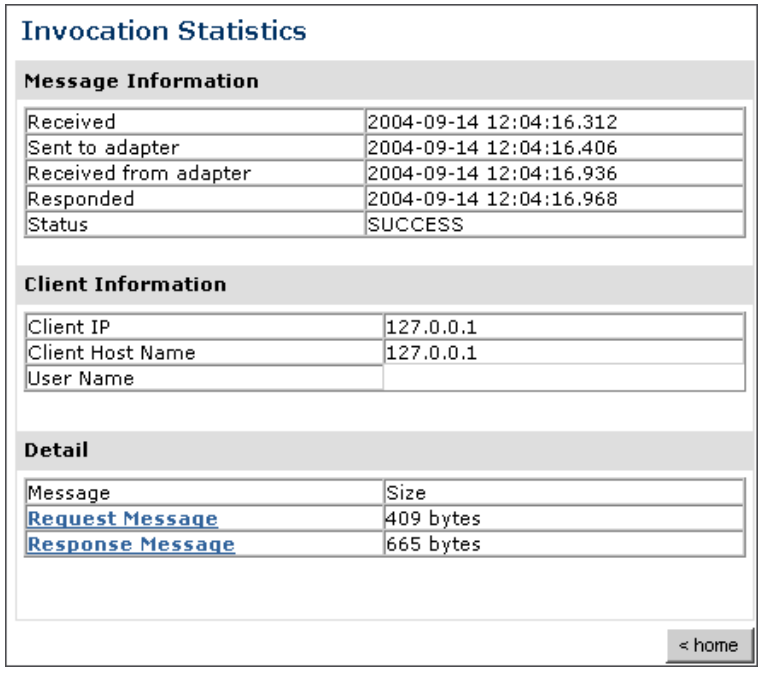
5. Select a method for the service from the Method drop-down list.

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).



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.



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.

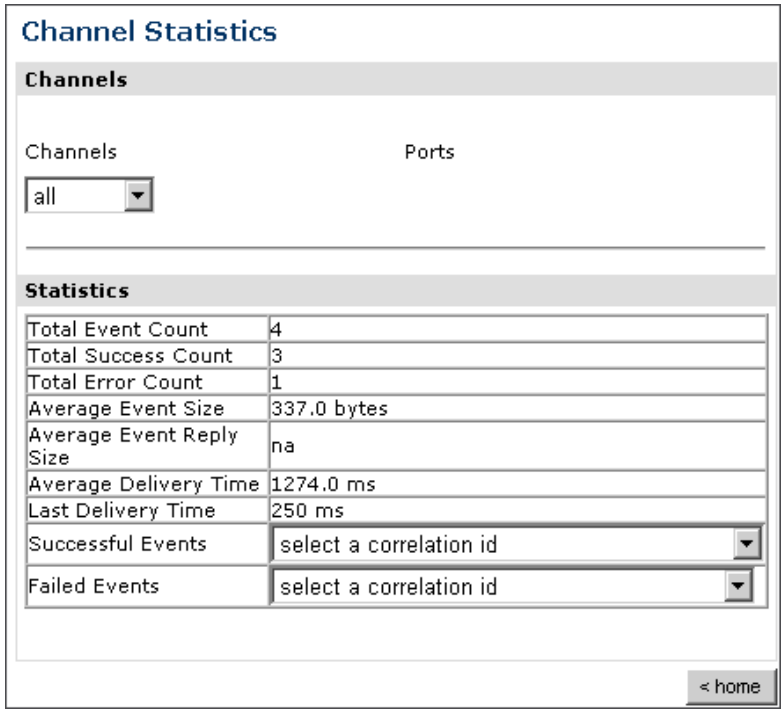
8. To return to the iBSE Monitoring Settings window, click *home*.

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.



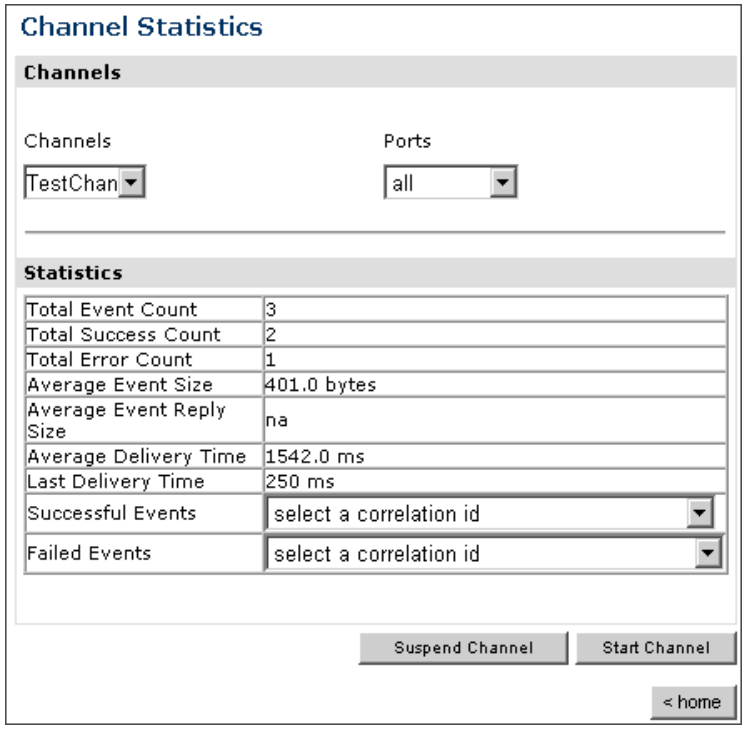
The system level summary provides event statistics at a system level.

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

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

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

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

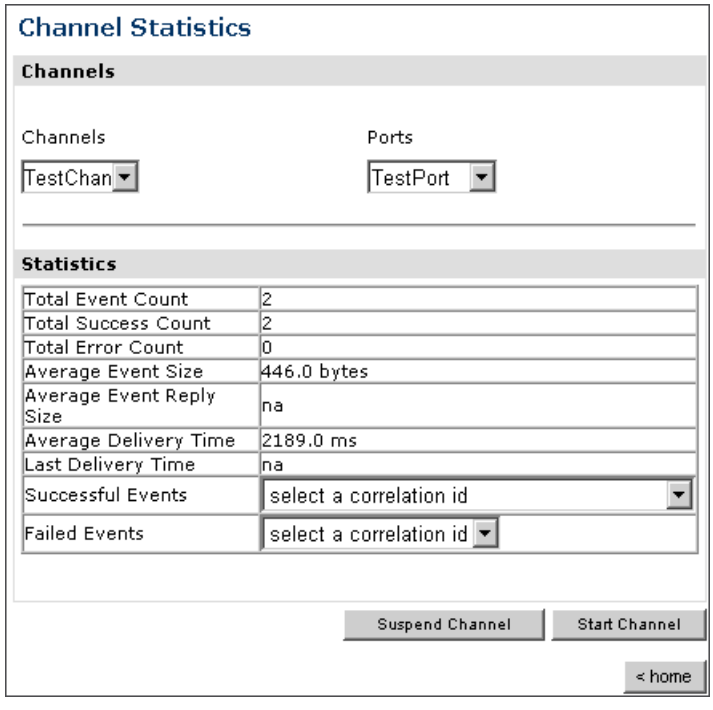


a. To stop a channel at any time, click *Suspend Channel*.

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

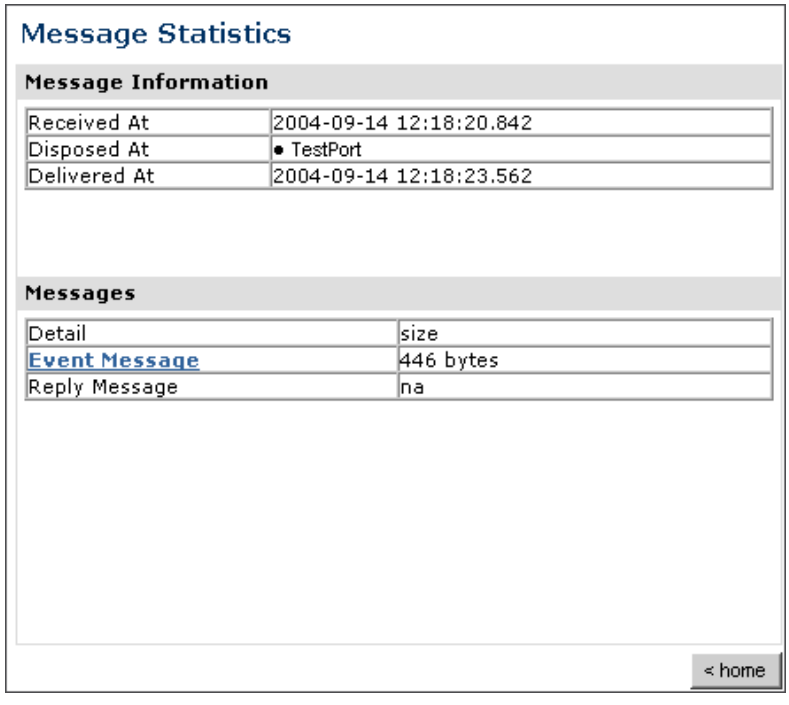
5. From the Ports drop-down list, select a port for the channel.

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.



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.



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

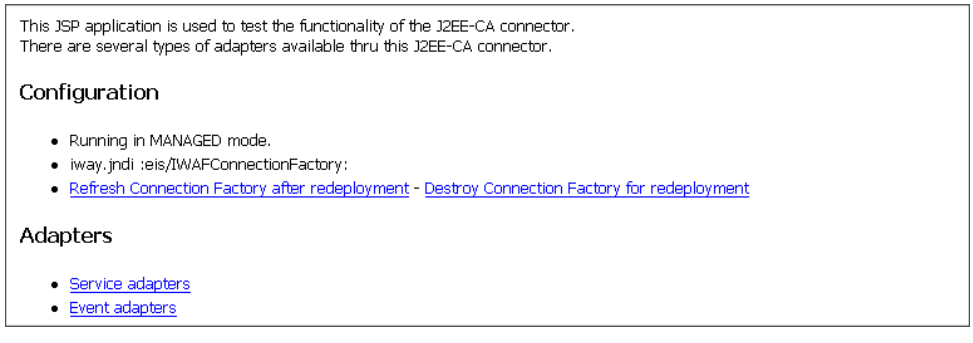
   Is the name of the machine where your application server is running.

   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.

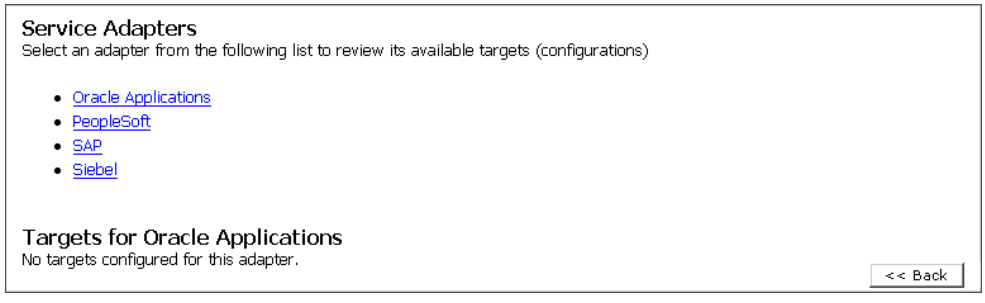
a. Click *Destroy Connection Factory for redeployment*.

b. Redeploy the JCA connector module using the BEA WebLogic Server console.

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

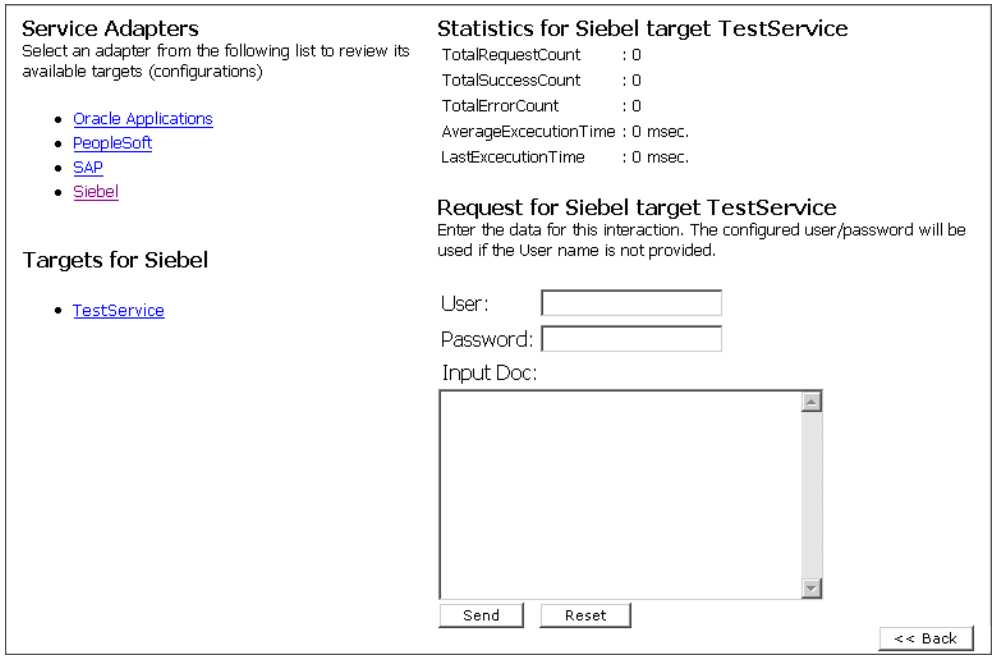
3. Click *Service adapters*.

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.



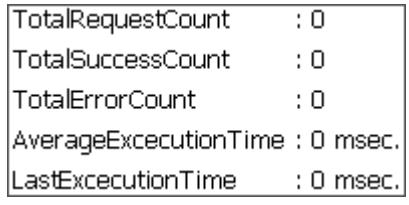
a. Click the desired target for your service adapter.

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.



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

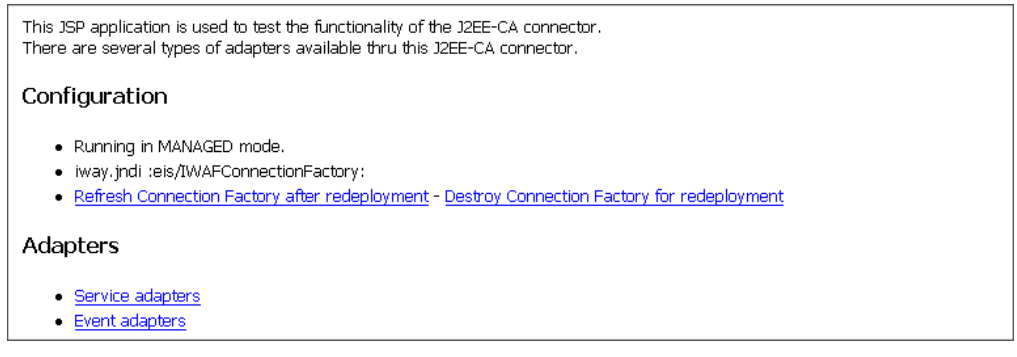
   Is the name of the machine where your application server is running.

   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.



   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*.

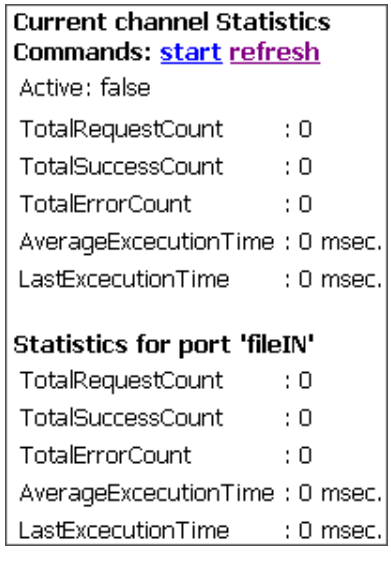
   The Event Adapters page opens.

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.



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.

Configuring Connection Pool Sizes

The following topic describes how to configure connection pool sizes for the JCA connector.

Procedure: How to Configure Connection Pool Sizes

To configure connection pool sizes:

1. Open the extracted ra.xml file in a text editor.

2. Locate and change the following setting:

pool-params. The JCA Resource Connector has an initial capacity value of 0 by default and cannot be changed. The maximum capacity value is 10 by default and can be changed to a higher value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE weblogic-connection-factory-dd (View Source for full
doctype...)>
- <weblogic-connection-factory-dd>
  <connection-factory-name>IWAFJCA</connection-factory-name>
  <jndi-name>eis/IWAFConnectionFactory</jndi-name>
  - <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>10</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>
  <security-principal-map />
</weblogic-connection-factory-dd>
```

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 5-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

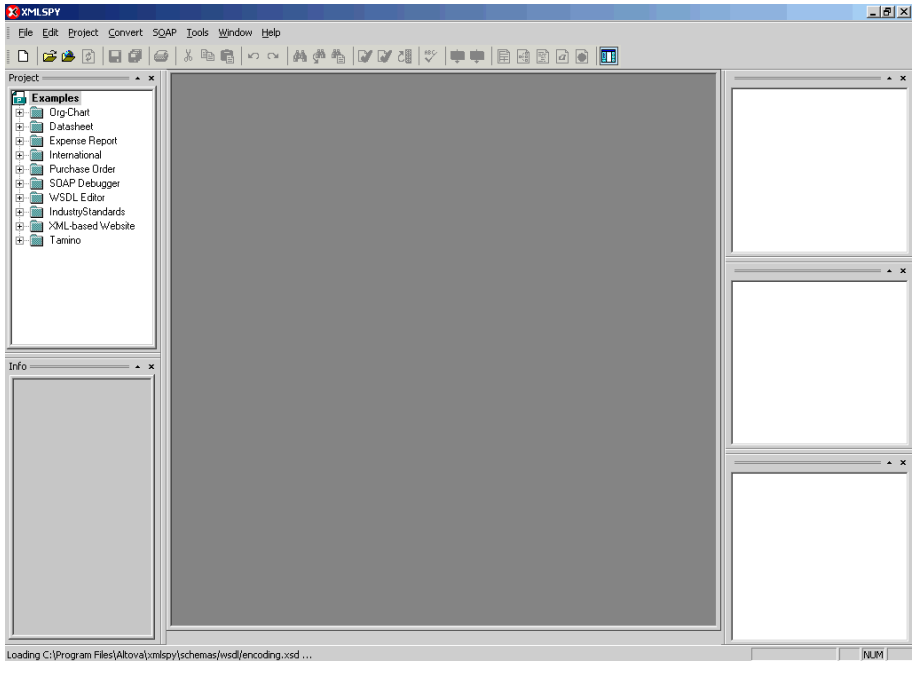
To migrate an iBSE repository that is configured for Oracle:

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

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

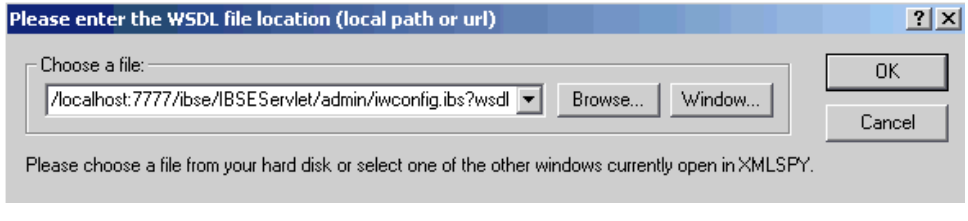
2. Open a third party XML editor, for example, XMLSPY.

   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.



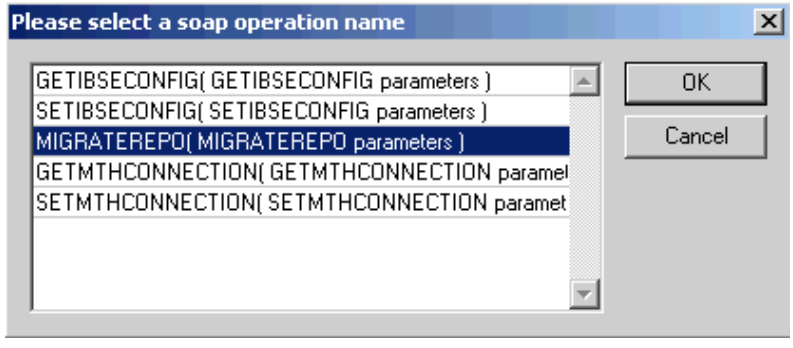
3. From the SOAP menu, select *Create new SOAP request*.

   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.



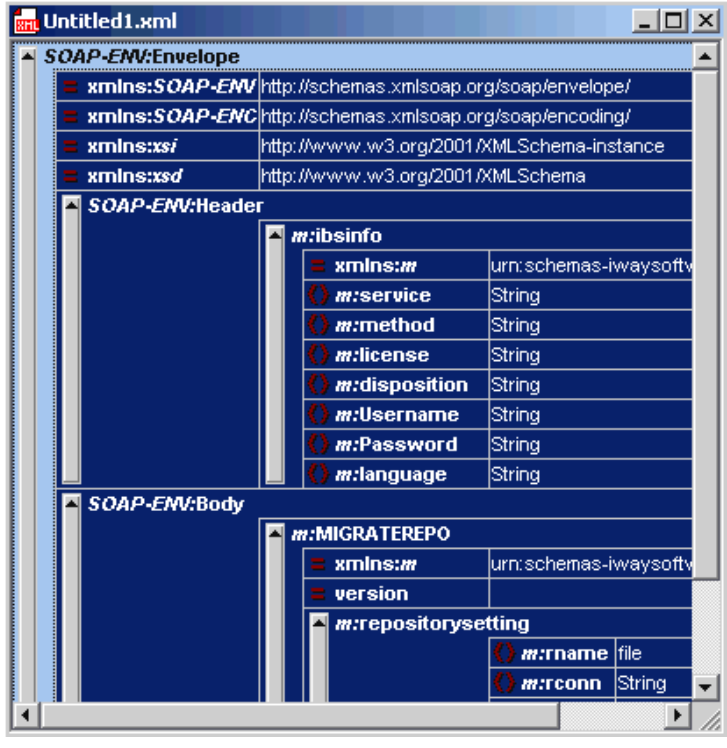
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

   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:

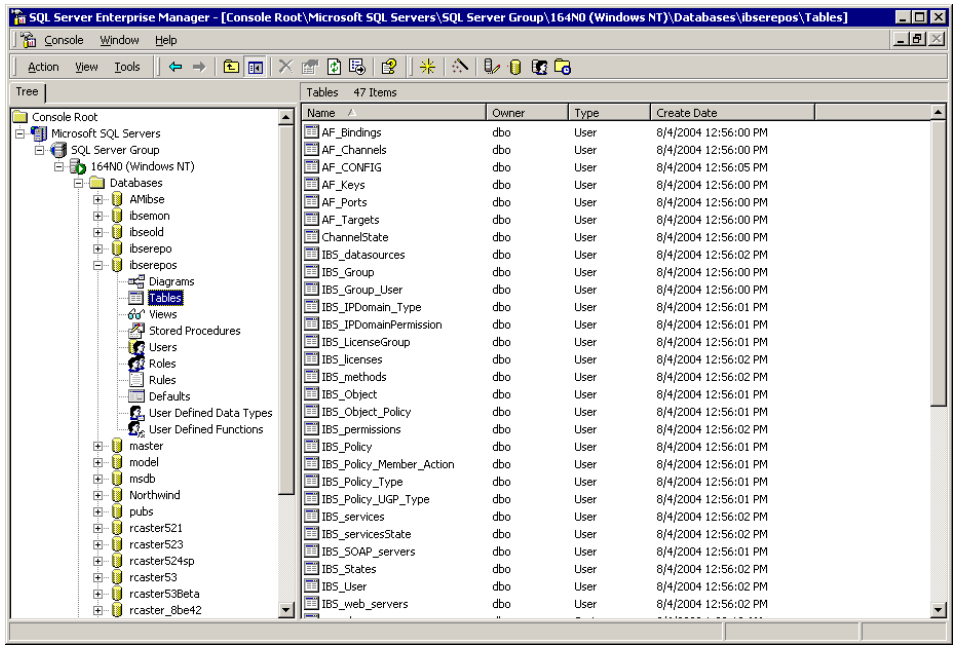
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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:

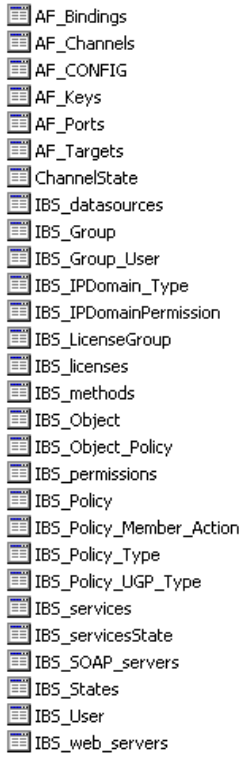
   For Oracle 8:

   ```
   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.



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:

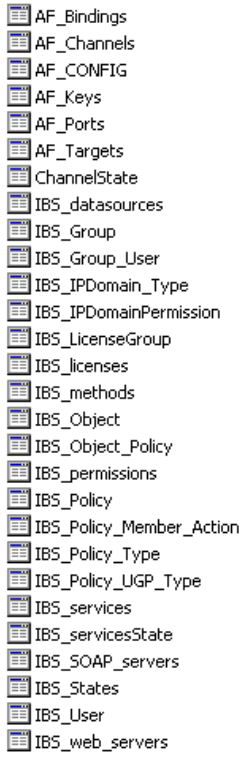
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
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.



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:

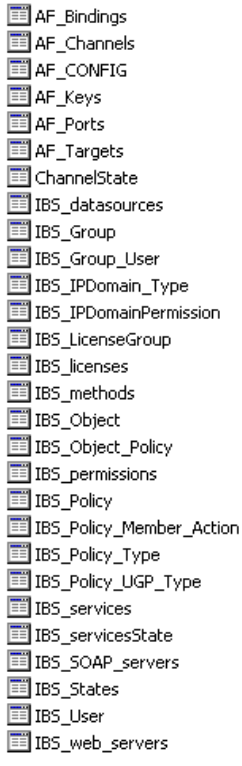
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
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.



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

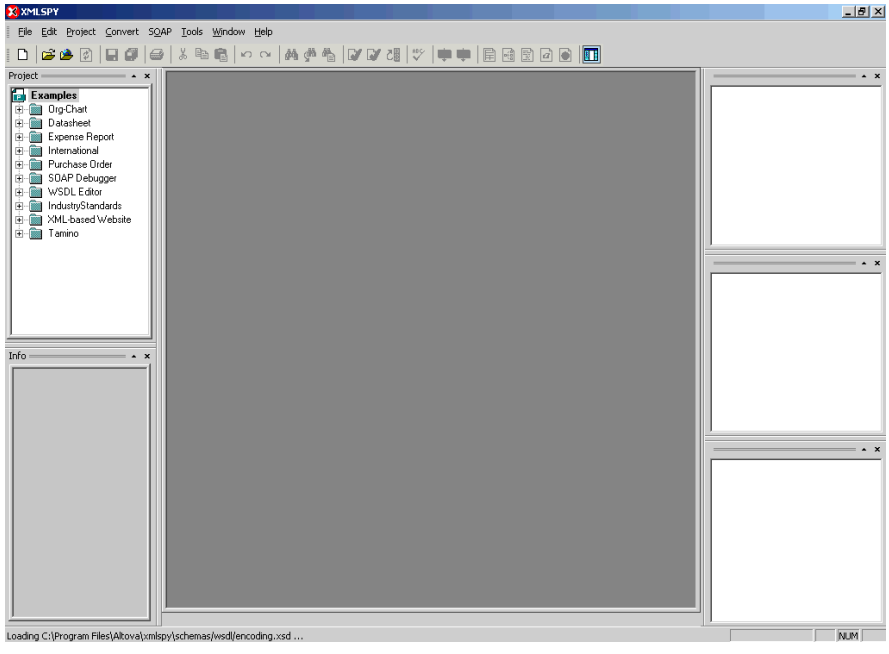
To export a target:

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.

   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

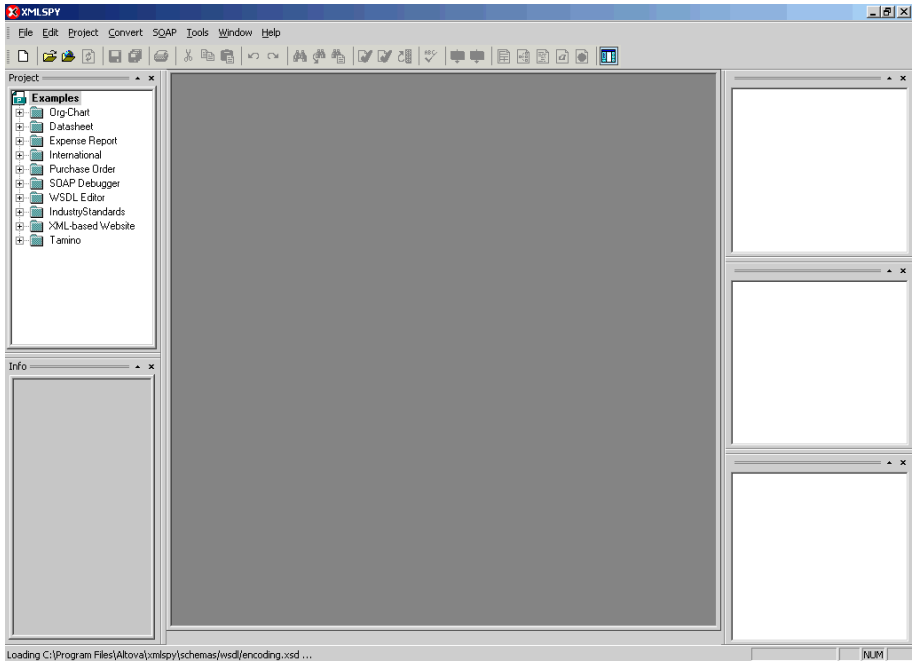
To import a target:

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.

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>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</m:contents>
</m:targetinstance>
</m:IMPORTTARGET>
```

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

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

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

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

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

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

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

Retrieving or Updating Web Service Method Connection Information

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

Procedure: How to Retrieve Web Service Method Connection Information

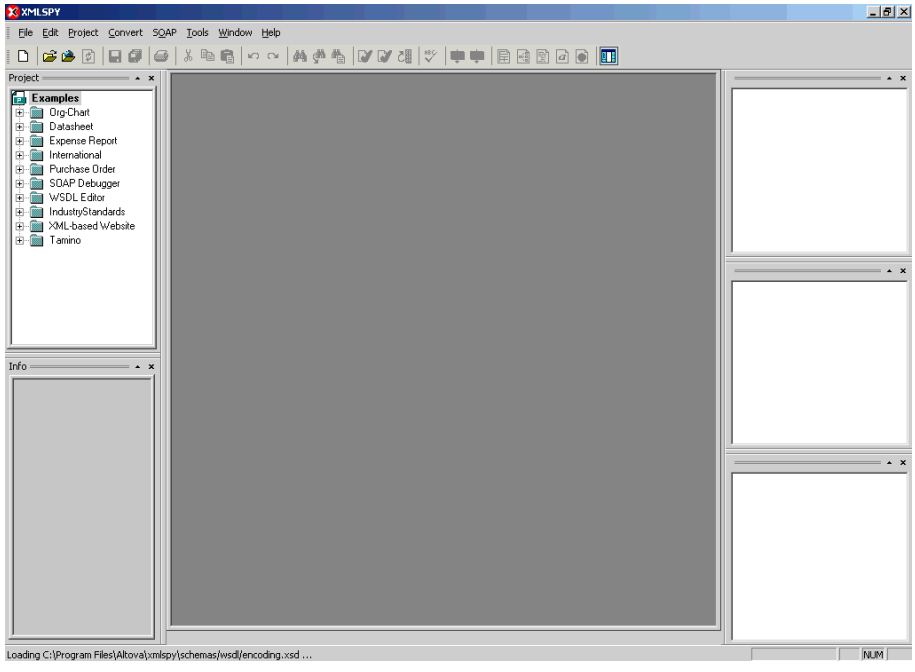
To retrieve Web service method connection information:

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

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

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

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



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

The WSDL file location dialog box opens.

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

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

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

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

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

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

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

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

8. Locate the following section:

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

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

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

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

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

Procedure: How to Update Web Service Method Connection Information

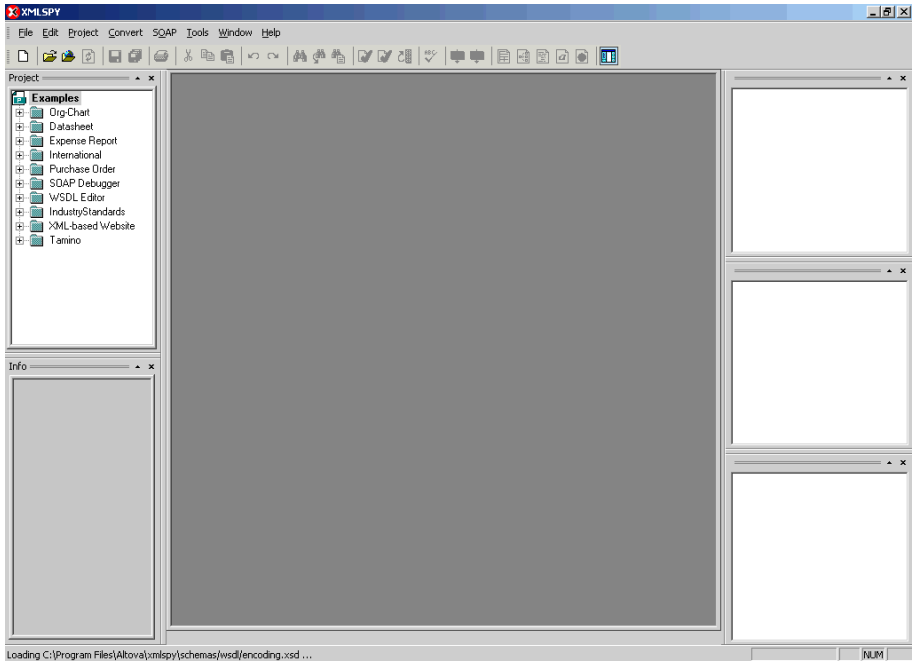
To update Web service method connection information:

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

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

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

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



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

The WSDL file location dialog box opens.

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

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

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

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

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

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

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

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

8. Locate the following section:

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

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

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

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

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

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

Starting or Stopping a Channel Programmatically

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

Procedure: How to Start a Channel Programmatically

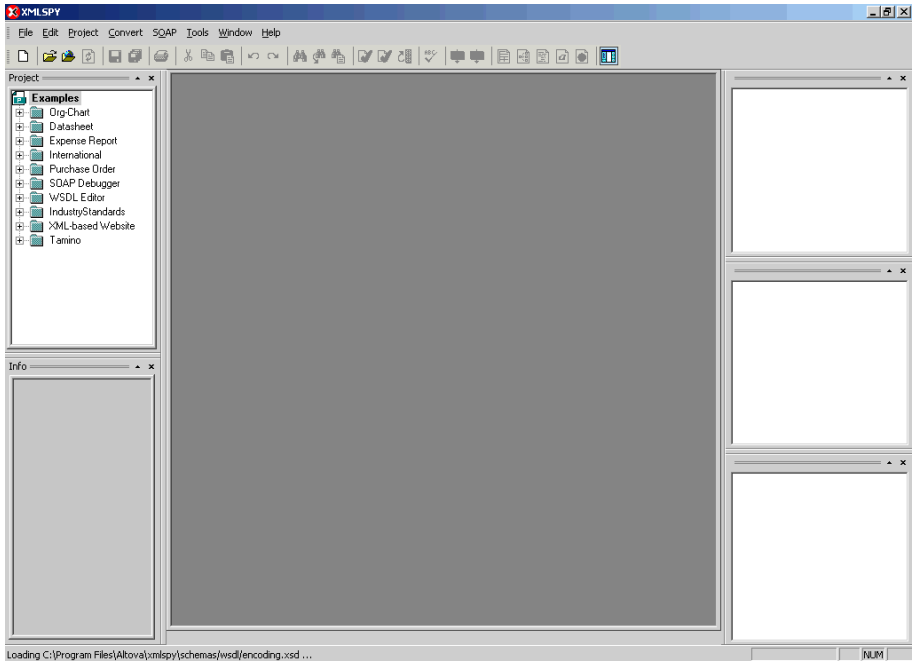
To start a channel programmatically:

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

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

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

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

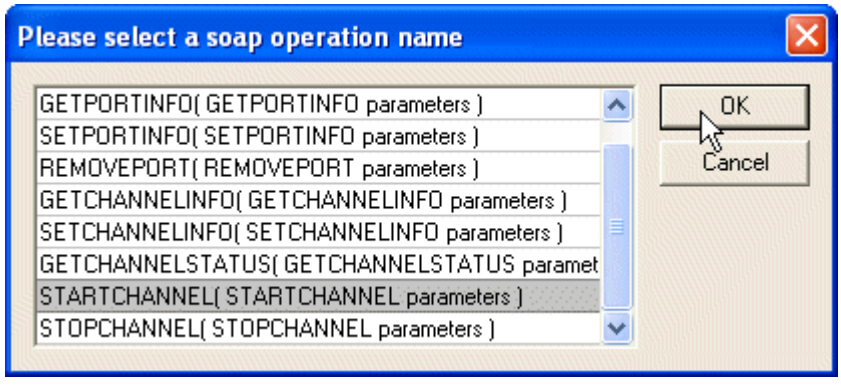


3. From the SOAP menu, select *Create new SOAP request*.

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

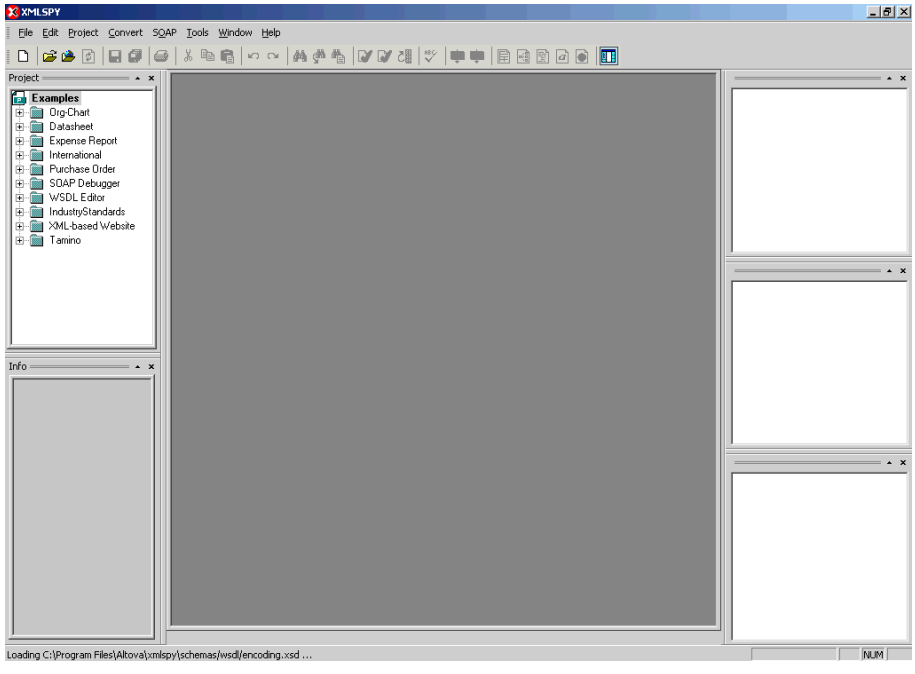
To stop a channel programmatically:

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.

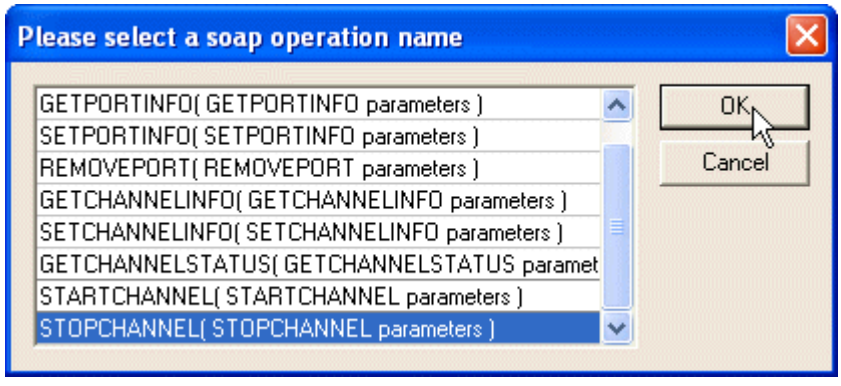


3. From the SOAP menu, select *Create new SOAP request*.

   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.*

APPENDIX A

Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Creating a New Configuration
- Connecting to Baan
- Creating an XML Schema
- Creating an Integration Business Service
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an Extensible CCI Control

This section describes how to use Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas for Baan. In addition, this section provides information on creating Web services that are published by the Integration Business Services Engine (iBSE).

Starting Application Explorer in BEA WebLogic Workshop

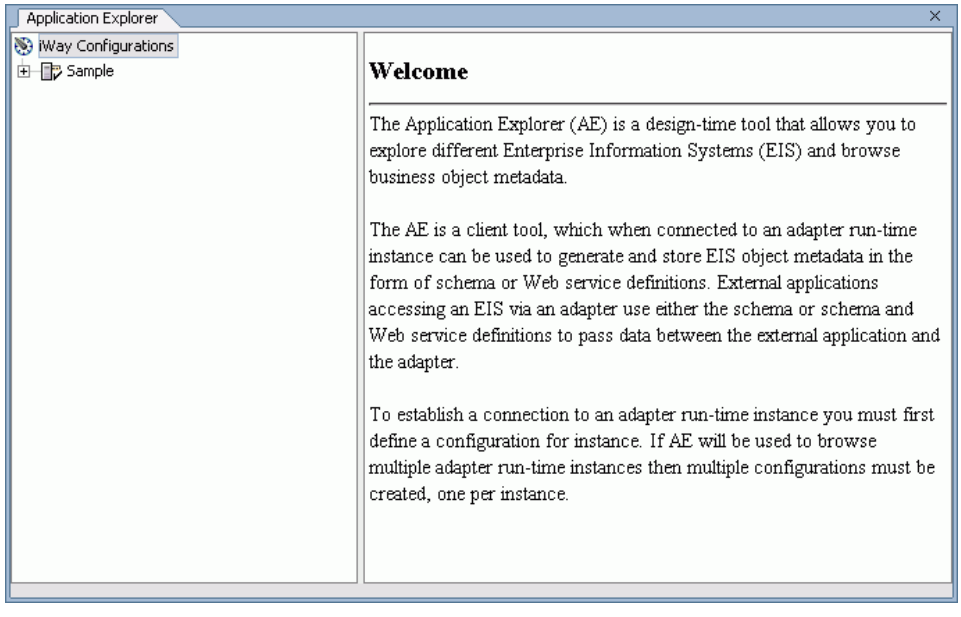
The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer running in BEA WebLogic Workshop:

1. Before starting Application Explorer, ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

   Application Explorer opens in BEA WebLogic Workshop.



   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

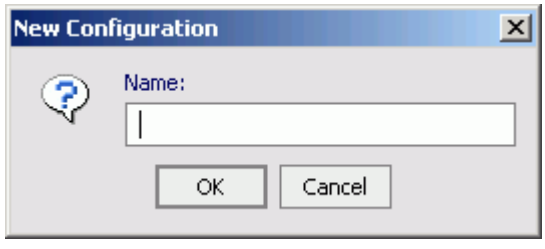
Before you can start using Application Explorer, you must define a new configuration for iBSE or JCA.

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

To create a new configuration:

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

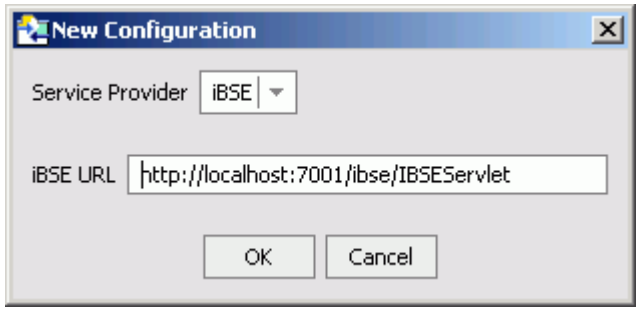
   The New Configuration dialog box opens.



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.

   The following dialog box opens.



3. 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
```

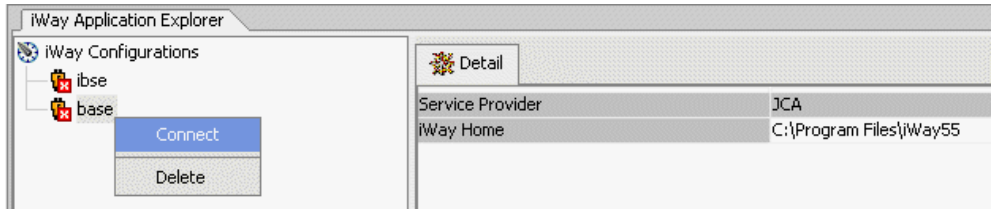
where:

iWay55

Is the full path to your iWay installation.

A node representing the new configuration appears under the iWay Configurations node. The right pane provides details of the configuration you created.

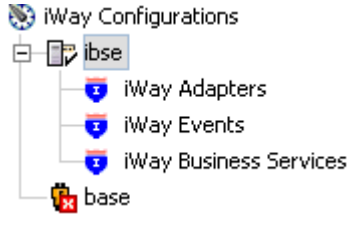
After you add your configuration, you must connect to it.



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

The iWay Adapters and iWay Events nodes appear.

When you connect to iBSE, the iWay Adapters, iWay Events, and Integration Business Services nodes appear.



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

   The iWay Adapters list includes a Baan node that enables you to connect to Baan 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-13.

   The iWay Events list includes a Baan node that enables you to create ports and channels for Baan event handling. For more information, see *Understanding Event Functionality* in Appendix B, *Using Application Explorer in BEA WebLogic Workshop for Event Handling*.

Connecting to Baan

To browse Baan, you must create a Baan target and connect to it. The target serves as your connection point. You must establish a connection to Baan every time you start Application Explorer or after you disconnect from Baan.

The left pane displays the application systems supported by Application Explorer. These are based on the iWay adapters you installed and are licensed to use.

Creating and Connecting to a Target

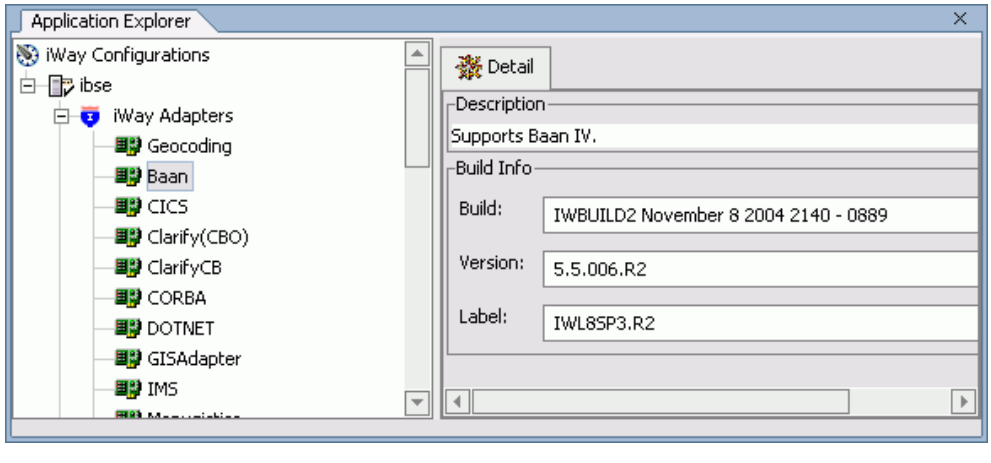
To connect to Baan for the first time, you must create a new target. The target is automatically saved after it is created.

Procedure: How to Create a New Target

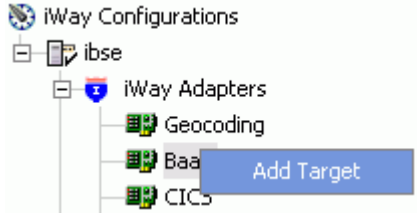
To create a target:

1. In the left pane, expand *iWay Adapters* and click the *Baan* node.

   Descriptive information (for example, title and product version) for the iWay Adapter for Baan appears in the right pane.

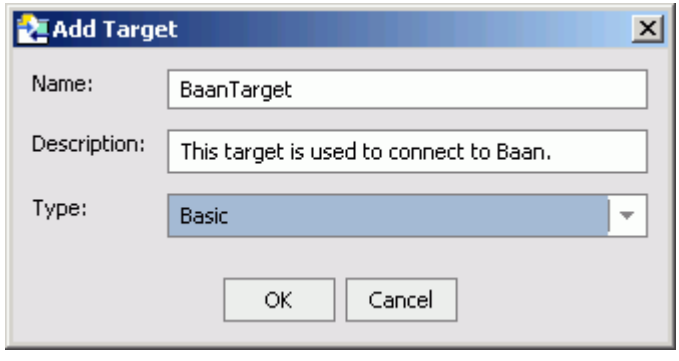


2. To view the options, right-click the *Baan* node.



3. Select *Add Target*.

   The Add target dialog box opens.



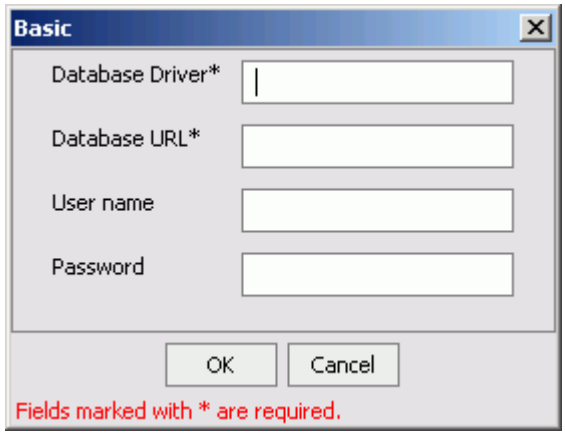
   a. In the Name field, type a descriptive name for the target, for example, BaanTarget.

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

   c. From the Target Type drop-down list, select *Basic*.

4. Click *OK*.

The Basic dialog box opens.



Note: The Baan connection parameters are consistent with those found in your Baan system. For more information on parameter values that are specific to your Baan configuration, consult your Baan system administrator.

5. Enter the connection information based on the parameters in the following table:

Parameter Name	Parameter Value

Database Driver (required)	The fully-qualified name of the JDBC driver used to access the database. The name of the driver for:
	Oracle is
	`oracle.jdbc.driver.OracleDriver`
	Informix is
	`com.informix.jdbc.IfxDriver`
	Microsoft SQL Server is
	`com.microsoft.jdbc.sqlserver.SQLServerDriver`
	ODBC is
	`sun.jdbc.odbc.JdbcOdbcDriver`

Parameter Name	Parameter Value

Database URL (required)	The URL of the JDBC driver used to communicate to the database. The URL format for:
	Oracle is
	`jdbc:oracle:thin:@host:port:servicename`
	for example,
	`jdbc:oracle:thin:@oracle11x:1523:VIS`
	Informix is
	`jdbc:informix-sqli://[hostname]:[port/database]:informixserver=[your-sever-name]`
	for example,
	`jdbc:informix-sqli://unxsol26:5053/qaeda:informixserver=online920`
	Microsoft SQL Server is
	`jdbc:microsoft:sqlserver://hostname:port[;property=value...]`
	for example,
	`jdbc:microsoft:sqlserver://server1:1433;user=test;password=secret`
	ODBC is
	`jdbc:odbc:DataSrcName[;<att-name>=<att-val>]...`
	where:
	DataSrcName
	Is the name of the data source you configured for ODBC.
	att-name
	Is a data source attribute name. You can specify multiple attribute=value pairs.
	att-val
	Is the value of the attribute.
	for example,
	`jdbc:odbc:Baan`

Parameter Name	Parameter Value
User name (required)	The user ID to access the database.
Password (required)	The password that corresponds to the user ID.

6. Click *OK*.

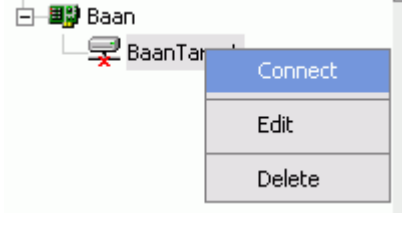
   In the left pane, the new target (BaanTarget) appears below the Baan node.

   You can now connect to the target you defined.

Procedure: How to Connect to a Target

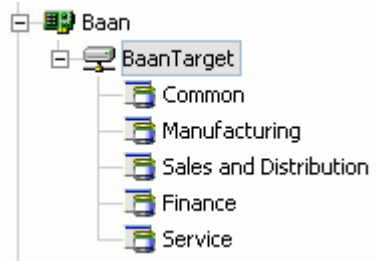
To connect to a Baan target:

1. In the left pane, expand the *Baan* node and select the target to which you want to connect, for example, BaanTarget.
2. In the right-pane, enter your password in the Password field.
3. In the left pane, right-click the target and select *Connect*.



   The BaanTarget node in the left pane changes to reflect that a connection was made.

4. Expand the target node to reveal the list of Baan interfaces.



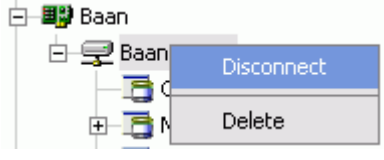
Managing a Target

Although you can maintain multiple open connections to different application systems, iWay Software recommends that you close connections when they 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 also can delete a target. 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:



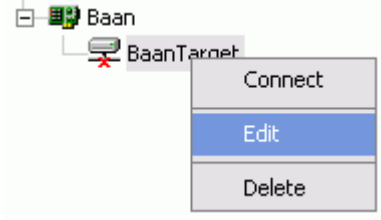
1. Right-click the *Baan* target from which you want to disconnect.
2. Select *Disconnect*.

   Disconnecting from the application system drops the connection, but the node remains. The BaanTarget node in the left pane changes to reflect that you disconnected from the target.

Procedure: How to Edit a Target

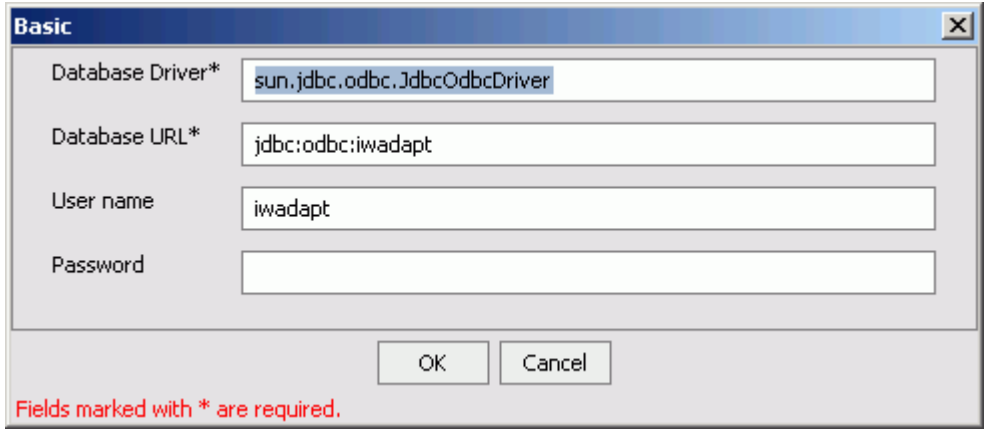
To edit a target:

1. Ensure that the target you want to edit is disconnected.



2. In the left pane, right-click the target and select *Edit*.

   The following dialog box opens.



3. Change the properties in the dialog box as required and click *OK*.

Procedure: How to Delete a Target

To delete a target:

1. In the left pane, right-click the target.

2. Select *Delete*.

   The BaanTarget node disappears from the left pane.

Creating an XML Schema

After you create a new configuration and connect to Baan, Application Explorer enables you to create a request or response schema.

Procedure: How to Create a Request and Response Schema

To create a request and response schema:

1. Expand the *Baan* node and select the node for which you want to create the schema.

   The following XML schemas appear for the interface:

   - Request (in this example MT540 to MT520)
   - Response (in this example MT520 to MT540)

2. To view the appropriate schema in the right pane, click the *Request Schema* or the *Response Schema* tab.

Reference: Schema Location

After you browse the Component Interfaces and make a selection, the request and response XML schemas are automatically created for that Component Interface and stored in the repository you created, for example:

```
drive:\Program Files\iWay55\bea\ibse\wsdl\schemas\service\Baan
\BaanTarget\SA45280C
```

where:

BaanTarget

Is the name of the Baan target.

SA45280C

Is a randomly generated folder name indicating where the schemas are stored.

Creating an Integration Business Service

You can create an Integration Business Service (also known as a Web service) for objects you want to use with your adapter. To generate a business service, you must deploy the iWay Adapter for Baan using the Integration Business Services Engine (iBSE). iBSE exposes functionality as Web services and serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a "black box" that may require input and delivers a result. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

You can make a Web service available to other services within a host server by generating WSDL (Web Services Description Language) from the Web service.

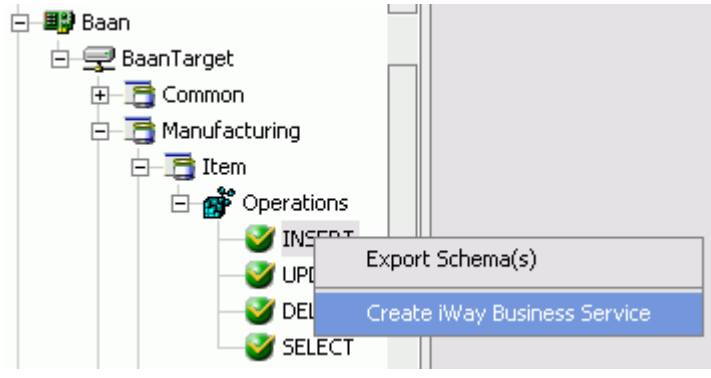
Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate iWay Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use iWay Web services, you export the WSDL to a directory accessible to BEA WebLogic Workshop.

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 *iWay Installation and Configuration for BEA WebLogic* manual and the i*Way Connector for JCA for BEA WebLogic Server User's Guide*.

Procedure: How to Create an Integration Business Service

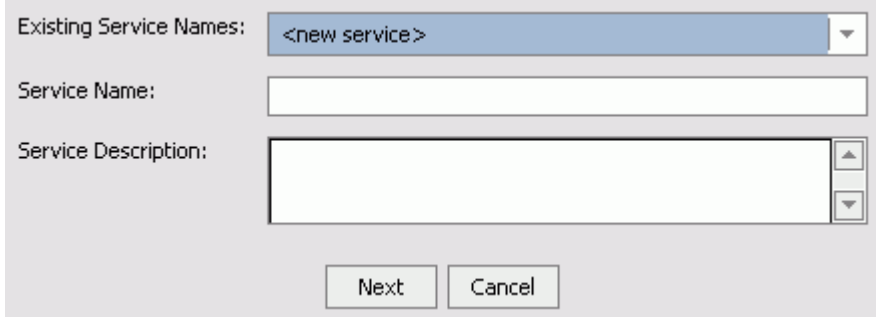
To create an Integration Business Service:

1. Expand the *Baan* node and select the interface for which you want to create a business service.



2. Right-click and select *Create Integration Business Service*.

   The Create Integration Business Service dialog box opens.



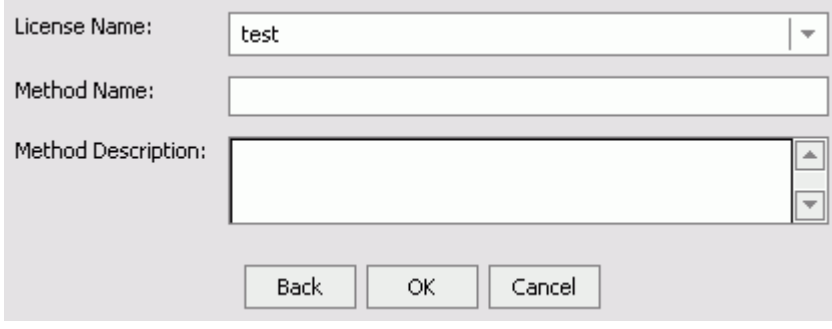
   a. From the Existing Service Names drop-down list, select whether you want to create a new service name or use an existing service name.

   b. In the Service Name field, type a name for the business service, for example, INSERTrequest.

   c. In the Service Description field, type a brief description of the business service.

3. Click *Next*.

   The Create Integration Business Service dialog box displays additional fields.



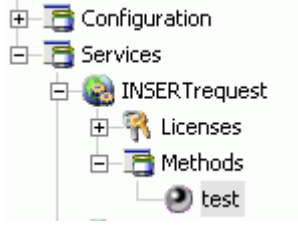
   a. From the License Name drop-down list, select a license.

   b. In the Method Name field, type a name for the method.

   c. In the Method Description field, type a brief description for the method.

4. Click *OK*.

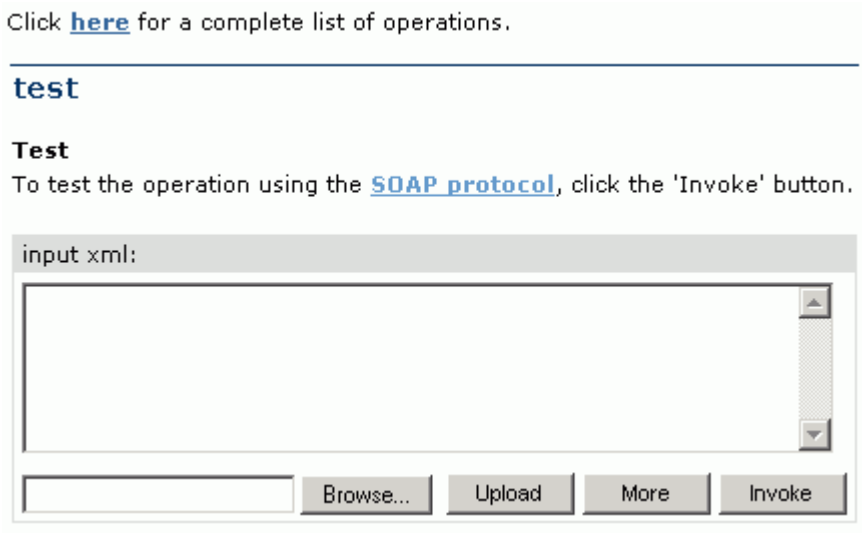
   The business service and method appear below the Integration Business Services node.



   In the left pane, all the available business services that were created appear.

5. Click the node for which you created the business service in the right pane.

The test pane opens in a new browser window.



6. To invoke the service, enter a sample XML document in the input xml field.
7. Click *Invoke*.

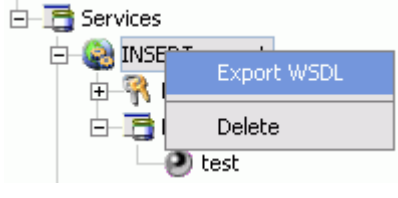
   The result appears in the right pane.

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 for Use in BEA WebLogic Workshop Workflows

To export WSDL to a directory accessible to BEA WebLogic Workshop:

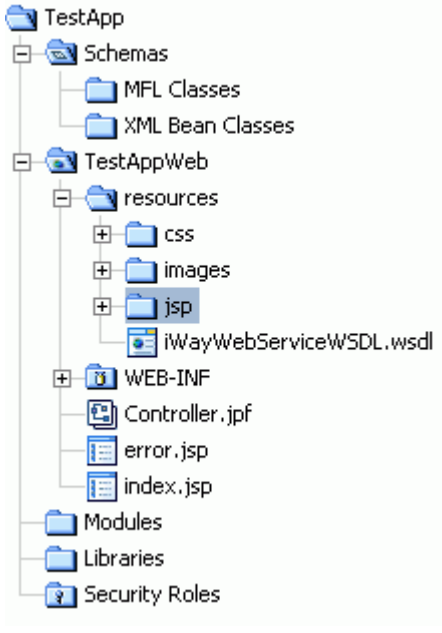


1. After you create a Web service, right-click the Web service name and select *Export WSDL*.

   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 WSDL file appears under 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 Baan. The user name and password values that you provided for Baan 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 the Java Swing version of 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.

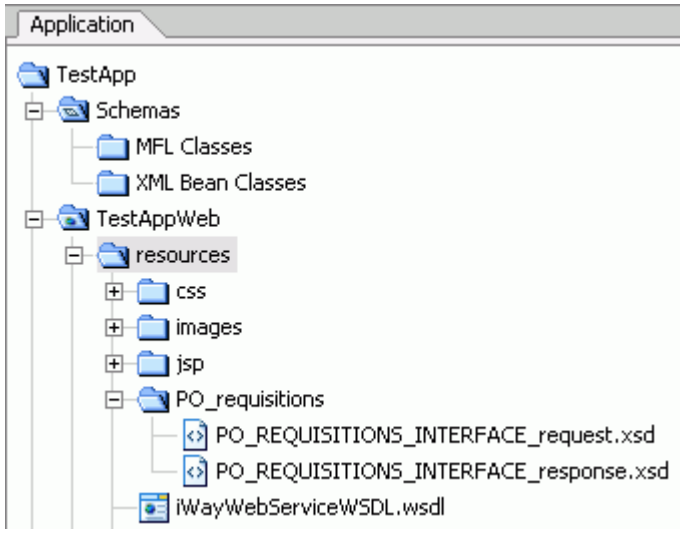
For more information on exporting a WSDL file, see *Export iWay WSDL for Use in BEA WebLogic Workshop Workflows* on page A-16.

Procedure: How to Add a Web Service Control

To add a Web service control:

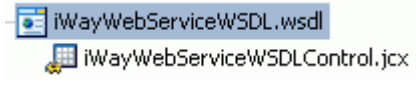
1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

   For example, a WSDL file saved to the \resources directory in your BEA WebLogic Workshop Web application directory structure appears as follows.



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

   The control for the WSDL appears below the WSDL file in the resources tree.



Adding an Extensible CCI Control

An iWay control enables access to resources provided by Application Explorer when it is used in conjunction with a JCA deployment. You must add an iWay control before using it in a BEA WebLogic Workshop application workflow.

The following topic 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:

```
public ResponseDataType MethodName(RequestDataType VariableName) throws
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

Is the XML Bean Class value that is generated from the request schema.

VariableName

Is the request variable that stores the request document, which is used as input by the extensible CCI control.

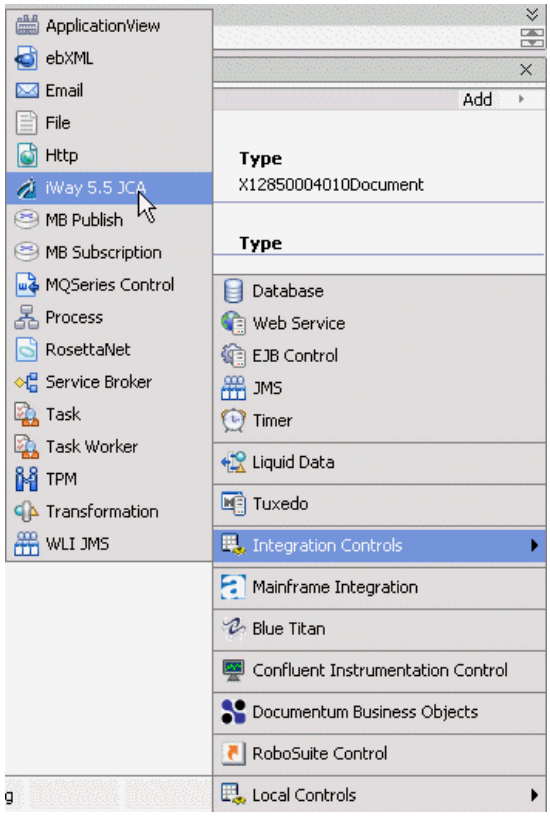
In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control* on page A-21.

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-27.

Example Defining a Control Using the Extensible CCI Control

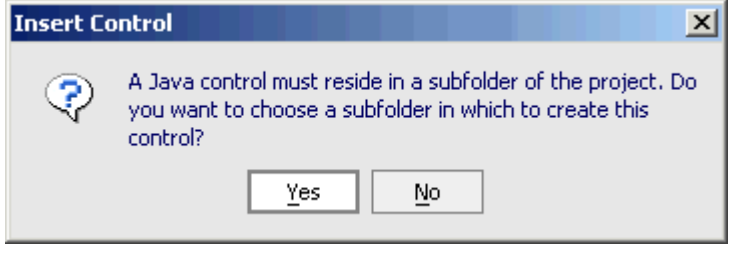
The following sample JCX demonstrates how to define a control for Baan using the extensible CCI control in BEA WebLogic Workshop.

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



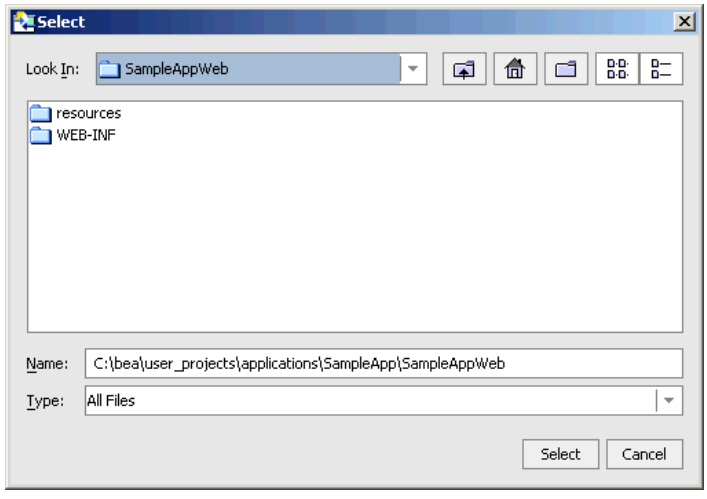
2. Click *Add* from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

   The Insert Control message box opens.



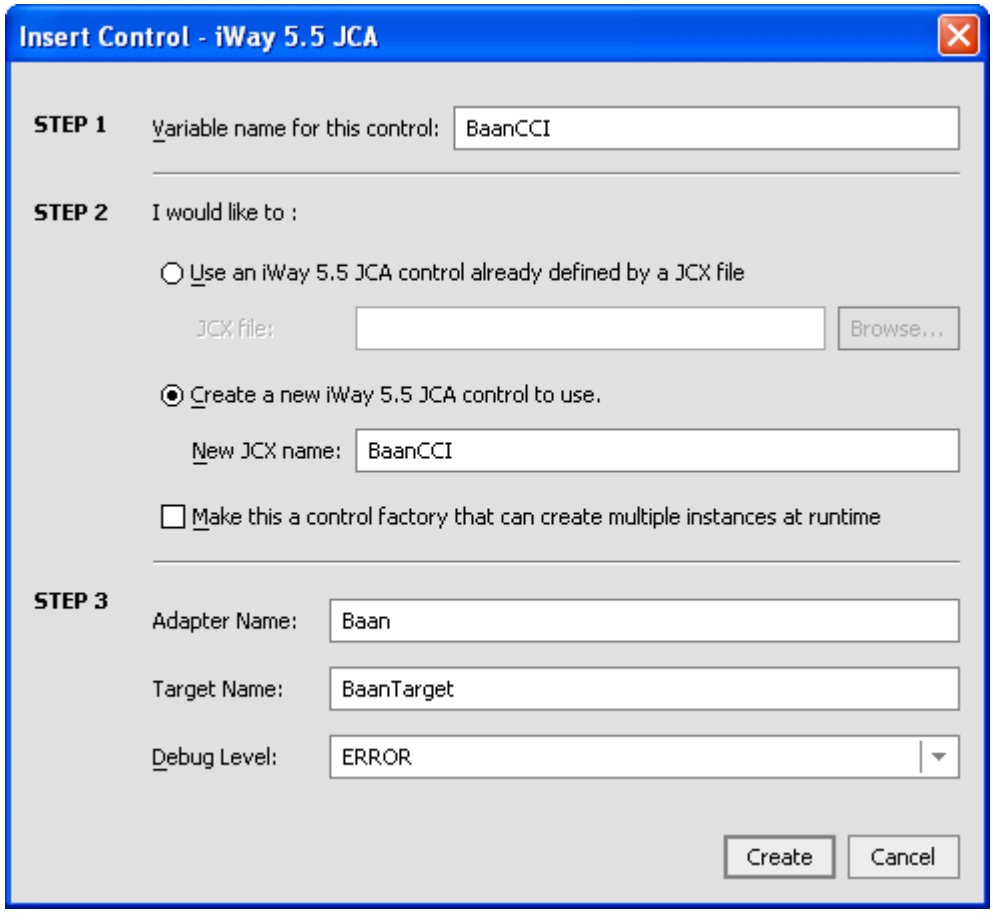
3. Click *Yes*.

The Select dialog box opens.



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

The Insert Control - iWay 5.5 JCA dialog box opens.



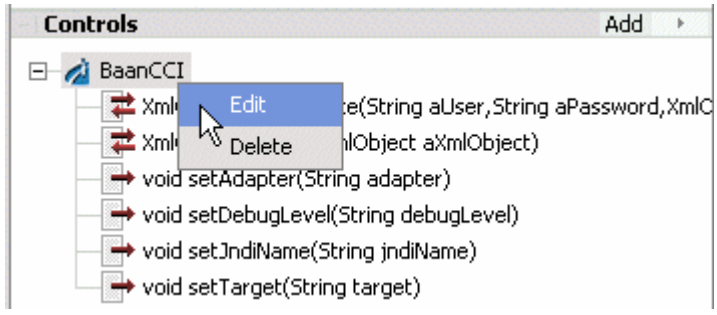
a. Provide a variable name for the control.

b. Click *Create a new iWay 5.5 JCA control to use* and provide a new JCX name.

c. Enter the adapter name, target name, and select a debug level from the drop-down list.

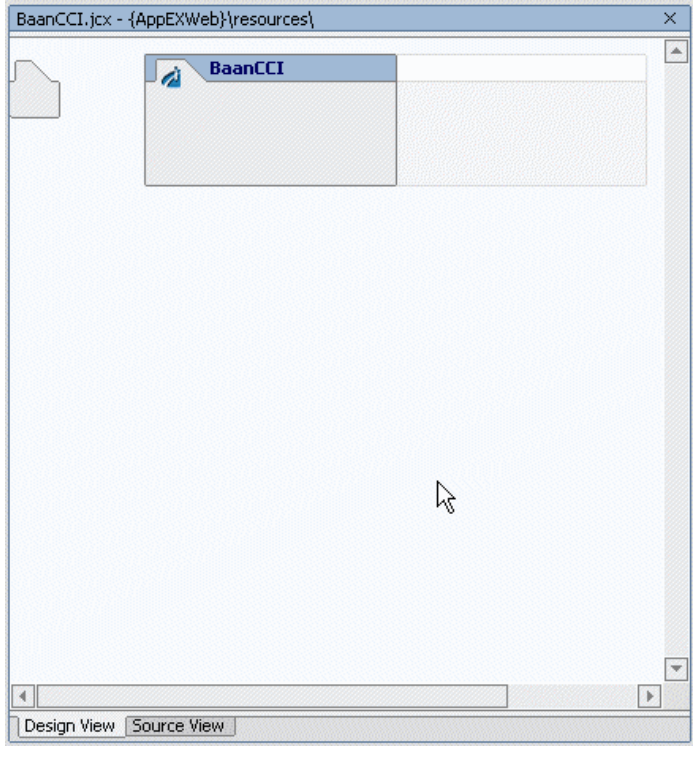
5. Click *Create*.

A new JCX file is created.



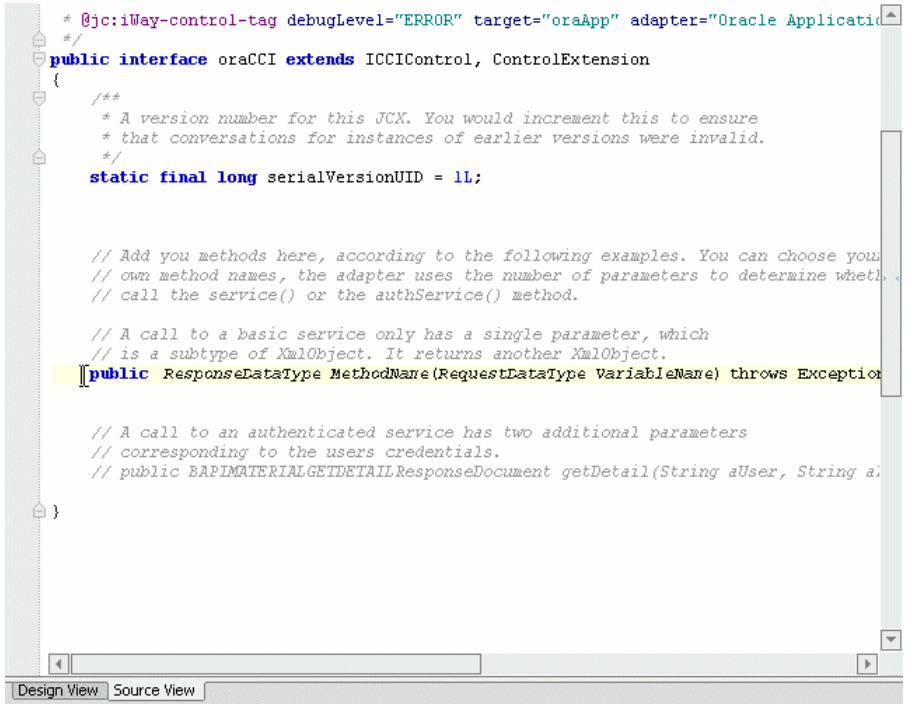
6. Right-click the control, for example, BaanCCI, and select *Edit*.

The Design View for the control opens.



7. Click the *Source View* tab.

The Source View for the control opens.



Perform the following steps:

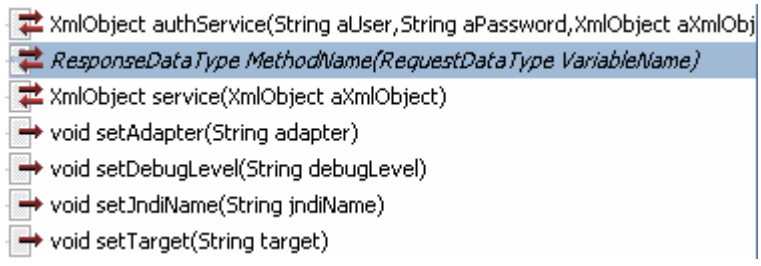
a. Uncomment the public class definition.

b. Change the existing response data type to match your response data type that is generated from your Baan response schema.

c. Change the existing method name to match your method.

d. Change the existing request data type to match your request data type that is generated from your Baan request schema.

The following control is now available in BEA WebLogic Workshop and can be added to a workflow:



Note: You can view available data types under 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.

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 ResponseDataType MethodName(RequestDataType VariableName) throws
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

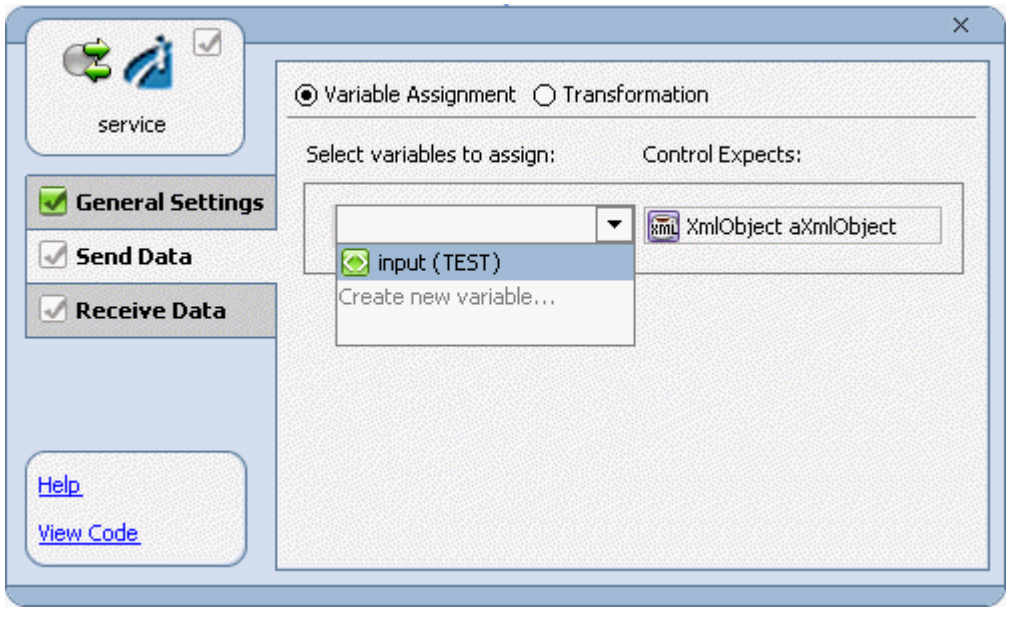
Is the XML Bean Class value that is generated from the request schema.

VariableName

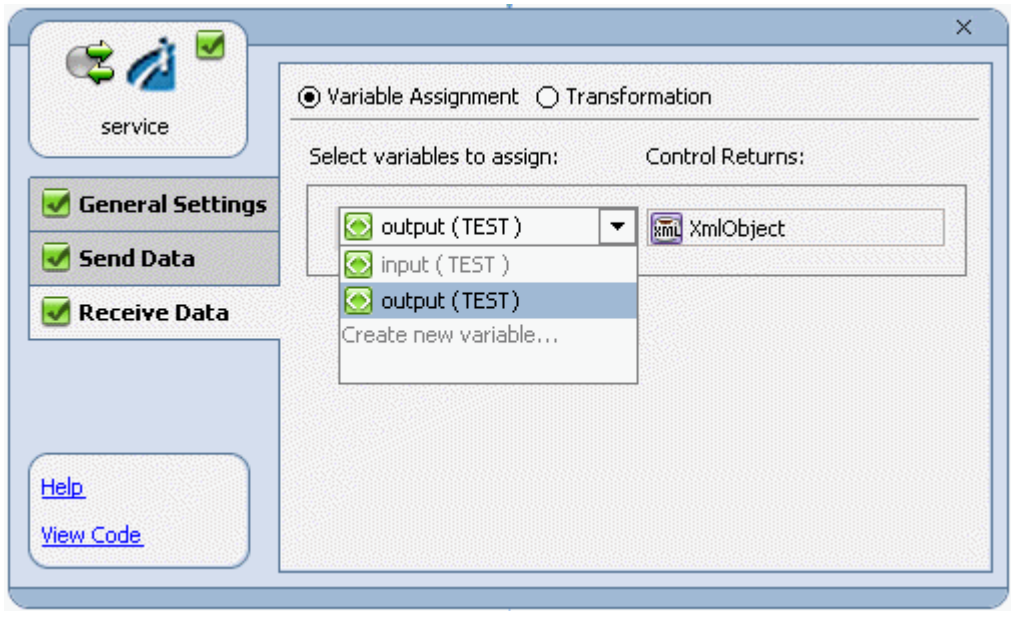
Is the request variable that stores the request document, which is used as input by the extensible CCI control.

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



APPENDIX B

Using Application Explorer in BEA WebLogic Workshop for Event Handling

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Understanding Event Functionality
- 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 Java Swing Application Explorer running in BEA WebLogic Workshop to create events for Baan. In addition, this section provides information on using events in a clustered BEA WebLogic environment.

Starting Application Explorer in BEA WebLogic Workshop

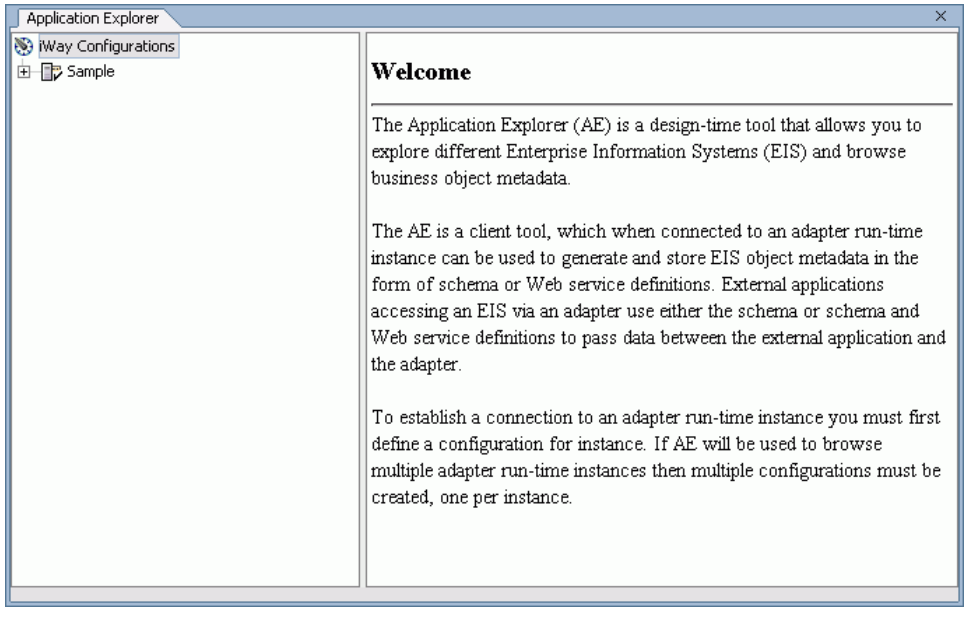
The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer running in BEA WebLogic Workshop:

1. Before starting Application Explorer, ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

   Application Explorer opens in BEA WebLogic Workshop.



   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 activity in an application system. You can use events to trigger an action in your application. For example, Baan may generate an event when customer information is updated. If your application must perform in response to activity, 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 define an event, you must create a port and a channel.

- Port

  A port associates a particular business object exposed by the 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-3.

- Channel

  A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the adapter. For more information, see *Creating a Channel* on page B-20.

Creating an Event Port

The following procedures describe how to create an event port using Application Explorer. The following port dispositions are available when using iBSE:

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

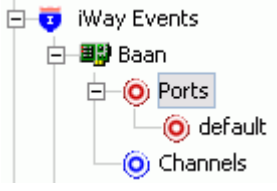
Note: The MAIL disposition option will be supported in a future release.

With a JCA implementation, the following port dispositions are available:

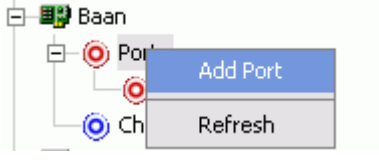
- File
- JMSQ
- MQ Series
- HTTP

Procedure: How to Create an Event Port for File

To create an event port for File:

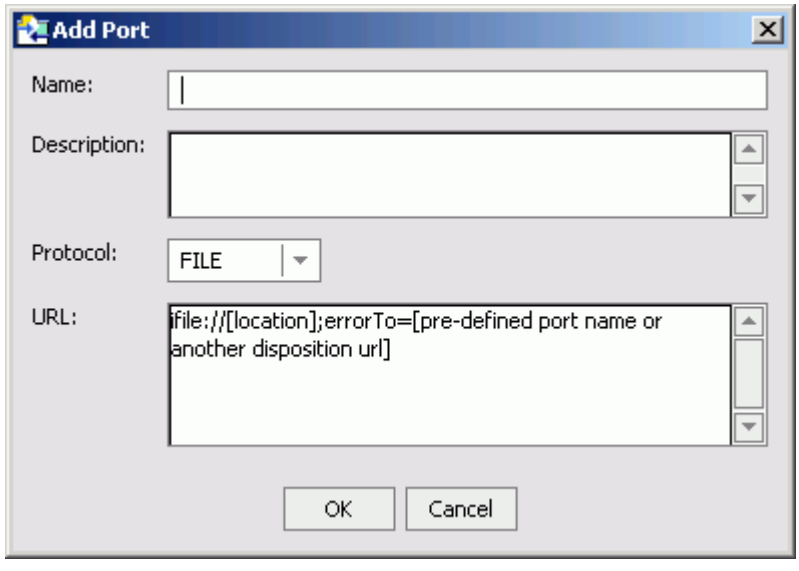


1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.



2. Right-click and select *Add Port*.

The Add Port dialog box opens.



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

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

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

d. In the URL field, type a destination file to which the event data is written, using the following format:

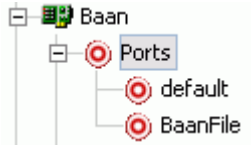
```
file://location[;errorTo=errorDest]
```

The following table describes the URL parameters.

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

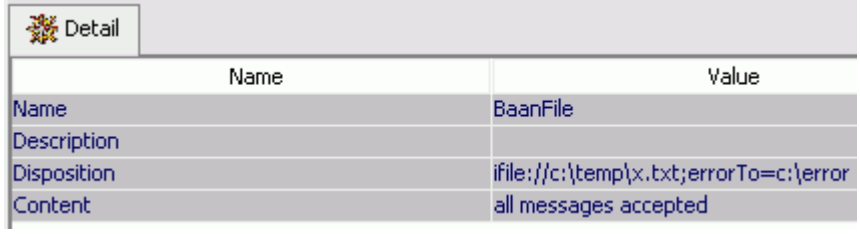
3. Click *OK*.

   In the left pane, the event port appears below the Ports node.



4. To review the port settings, select the port name.

   In the right pane, a table appears that summarizes the information associated with the event port you created.



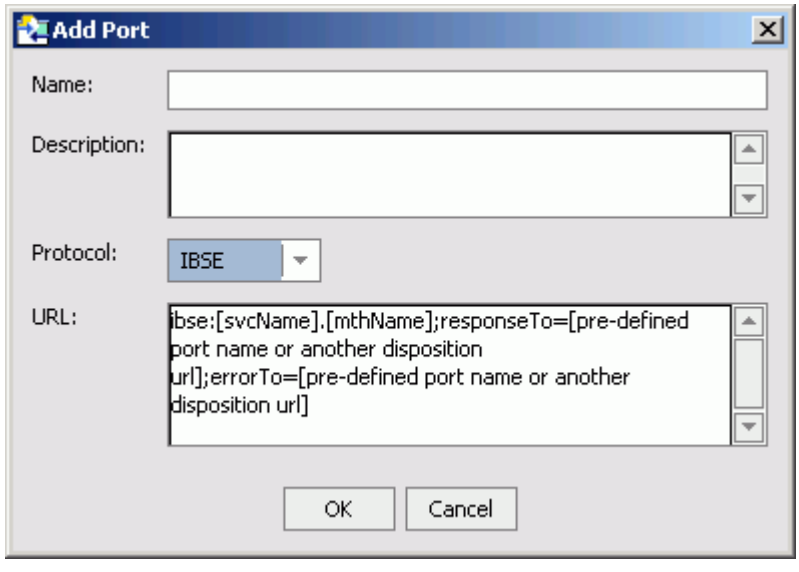
   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-20.

Procedure: How to Create an Event Port for iBSE

To create an event port for iBSE:

1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



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

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

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

d. In the URL field, enter an iBSE destination using the following format:

```
ibse:/svcName.methName[;responseTo=respDest][;errorTo=errorDest]
```

The following table describes the disposition parameters.

Parameter	Description
svcName	Name of the service created with iBSE.
methName	Name of the method created for the Web service.
respDest	Location where responses to the Web service are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click *OK*.

In the left pane, the event port appears below the Ports node.

4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

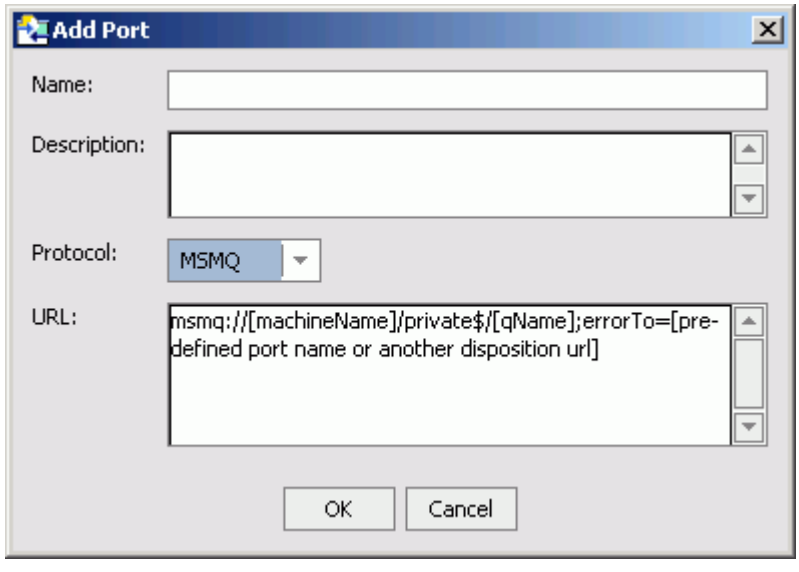
You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-20.

Procedure: How to Create an Event Port for MSMQ

To create an event port for a Microsoft Message Queuing (MSMQ) queue:

1. In the left pane of Application Explorer, expand the Baan node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



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

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

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

d. In the URL field, enter an MSMQ destination in the following format:

```
msmq:/host/queueType/queueName[;errorTo=errorDest]
```

The following table defines the disposition parameters.

Parameter	Description
host	Name of the host on which the Microsoft Queuing system runs.
queueType	The type of queue. For private queues, enter *Private$*. 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.
queueName	Name of the queue where messages are placed.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click *OK*.

In the left pane, the event port appears below the Ports node.

4. To review the port settings, select the port name.

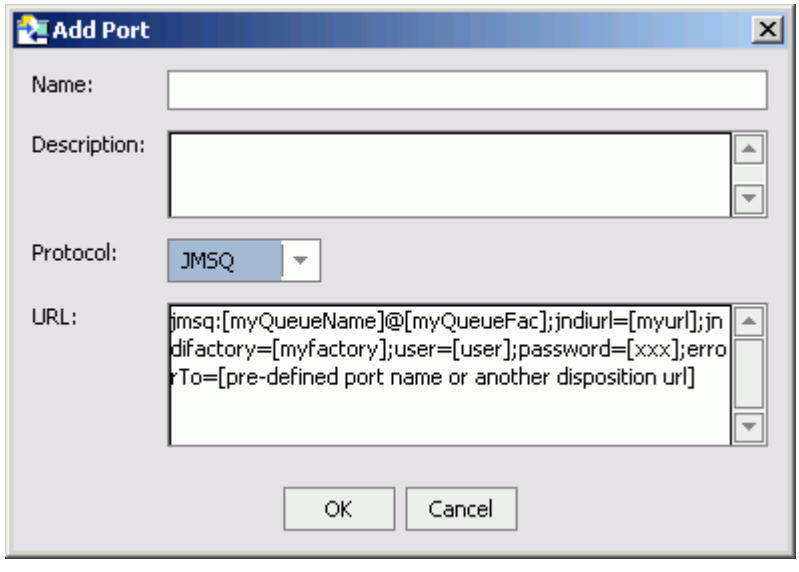
In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-20.

Procedure: How to Create a Port for JMS

To create a port for a JMS queue:

1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



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

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

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

d. In the URL field, enter a JMSQ destination using the following format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

The following table describes the URL parameters.

Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource that contains information about the JMS Server. The WebLogic connection factory is: `javax.jms.QueueConnectionFactory`
jndi_url	The URL of the application server. For BEA WebLogic Server, the URL is `t3://host:port` where: *host* Is the machine name where BEA WebLogic Server resides. *port* Is the port on which BEA WebLogic Server is listening. The default port if not changed at installation, is 7001.
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the WebLogic factory is weblogic.jndi.WLInitialContextFactory.
userID	User ID associated with this queue.
pass	Password associated with this user ID.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click *OK*.

The event port appears below the Ports node in the left pane.

4. To review the port settings, select the port name.

   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 B-20.

Procedure: How to Create a Port for the SOAP Disposition

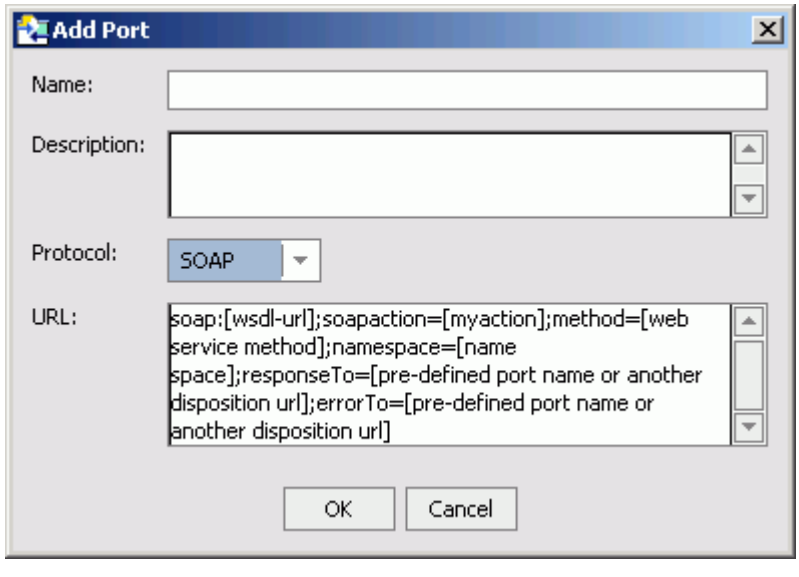
This topic describes how to configure the SOAP disposition for synchronous event processing.

The SOAP disposition allows an event response to launch a Web service specified by a WSDL file. A soapaction is optional, the default is "".

To create a port for a SOAP disposition using Application Explorer:

1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



   a. In the Name field, type a name for the event port, for example, BaanSOAP.
   b. In the Description field, type a brief description.
   c. From the Protocol drop-down list, select *SOAP*.

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

```
soap:wsdl-url;soapaction=action[;responseTo=respDest]
   [;errorTo=errorDest]
```

The following table lists and describes the disposition parameters for SOAP.

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

3. Click *OK*.

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

   You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page B-20.

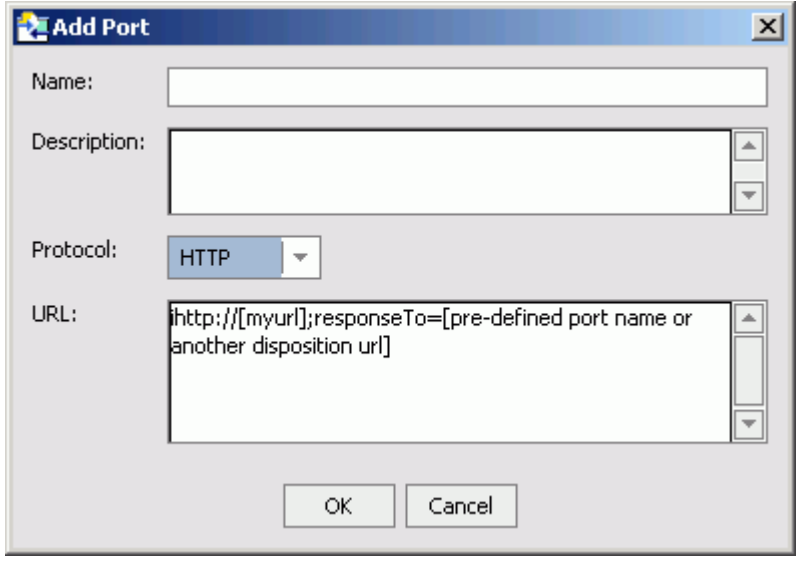
Procedure: How to Create an Event Port for HTTP

The HTTP disposition uses an HTTP URL to specify an HTTP end point to which the event document is posted.

To create an event port for HTTP disposition:

1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



   a. In the Name field, type a name for the event port, for example, BaanHTTP.
   b. In the Description field, type a brief description.
   c. From the Protocol drop-down list, select *HTTP*.
   d. In the URL field, enter an HTTP destination using the following format:

      ihttp://*url*;responseTo=*respDest*

The following table describes the URL parameters.

Parameter	Description
url	The URL target for the post operation.
respDest	Location where responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the host on which the Web server resides.
port	Port number on which the Web server is listening.

3. Click *OK*.

The event port appears below the Ports node in the left pane.

4. To review the port settings, select the port name.

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 B-20.

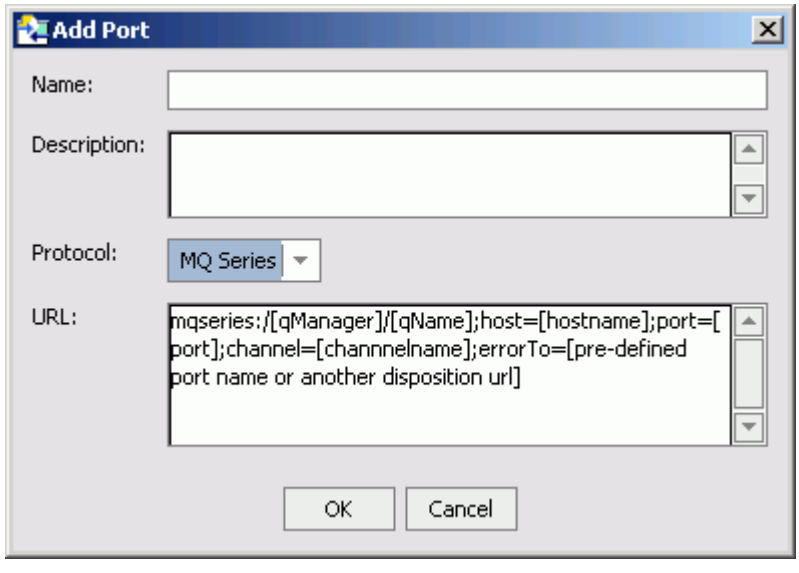
Procedure: How to Create an Event Port for MQ Series

The MQ Series disposition allows an event to be enqueued to an MQ Series queue. Both queue manager and queue name may be specified.

To create a port for an MQ Series queue:

1. In the left pane of Application Explorer, expand the *Baan* node under iWay Events, and then select *Ports*.
2. Right-click and select *Add Port*.

   The Add Port dialog box opens.



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

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

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

d. In the URL field, enter an MQ Series destination using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;
channel=chanName[;errorTo=errorDest]
```

The following table describes the URL parameters.

Parameter	Description
qManager	Name of queue manager to which the server must connect.
qName	Name of the queue where messages are placed.
hostName	Name of the host on which MQ Series resides (MQ client only).
portNum	Port number for connecting to an MQ Server queue manager (MQ client only).
chanName	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default MQ Series channel name is SYSTEM.DEF.SVRCONN.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click *OK*.

The event port appears below the Ports node in the left pane.

4. To review the port settings, select the port name.

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 B-20.

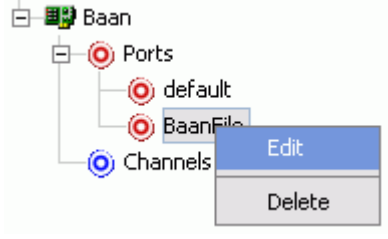
Modifying an Event Port

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

Procedure: How to Edit an Event Port

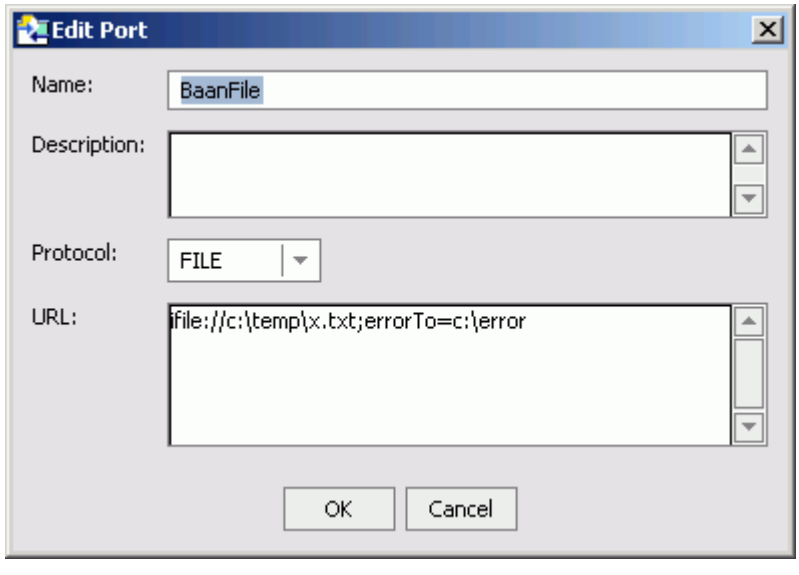
To edit an event port:

1. To view the available ports, click the *Ports* node in the left pane.



2. Right-click the port you want to edit, and select *Edit*.

   The Edit Port dialog box opens.

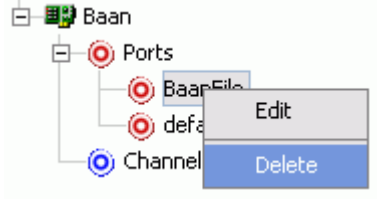


3. Make the required changes and click *OK*.

Procedure: How to Delete an Event Port

To delete an existing event port:

1. To view the available ports, click the *Ports* node in the left pane.



2. Right-click the port you want to remove, and select *Delete*.

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

Creating a Channel

The following procedure describes how to create a channel for a Baan event. All defined event ports must be associated with a channel.

You can create a Table Listener using Application Explorer:

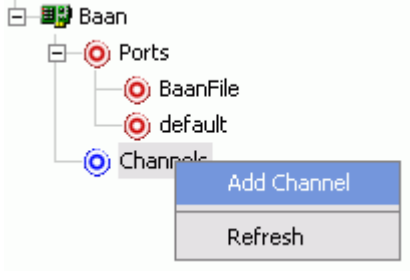
Procedure: How to Create a Channel

1. In the left pane, below the configuration you created, expand the *iWay Events* node.

   The list of adapters appears.

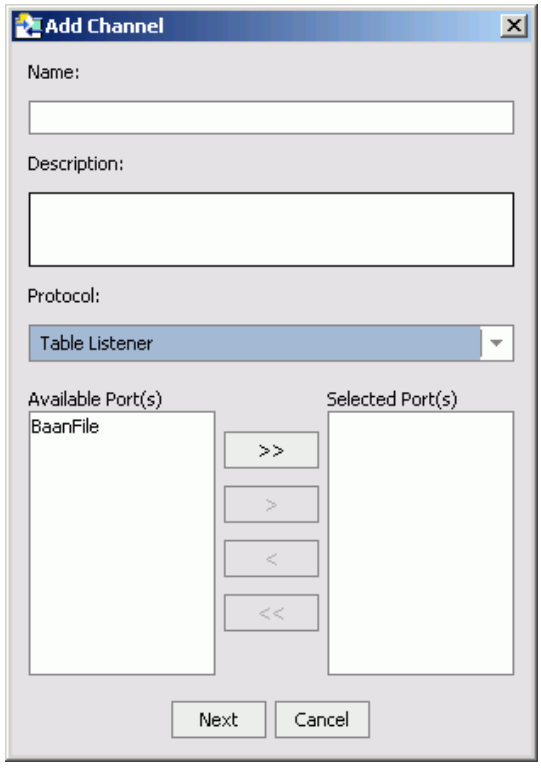
2. Click the adapter node, for example, Baan.

   The node expands and displays the Ports and Channels nodes.



3. Right-click the *Channels* node and select *Add Channel*.

The Add Channel dialog box opens.



a. In the Name field, type a name for the channel, for example, BaanChannel.

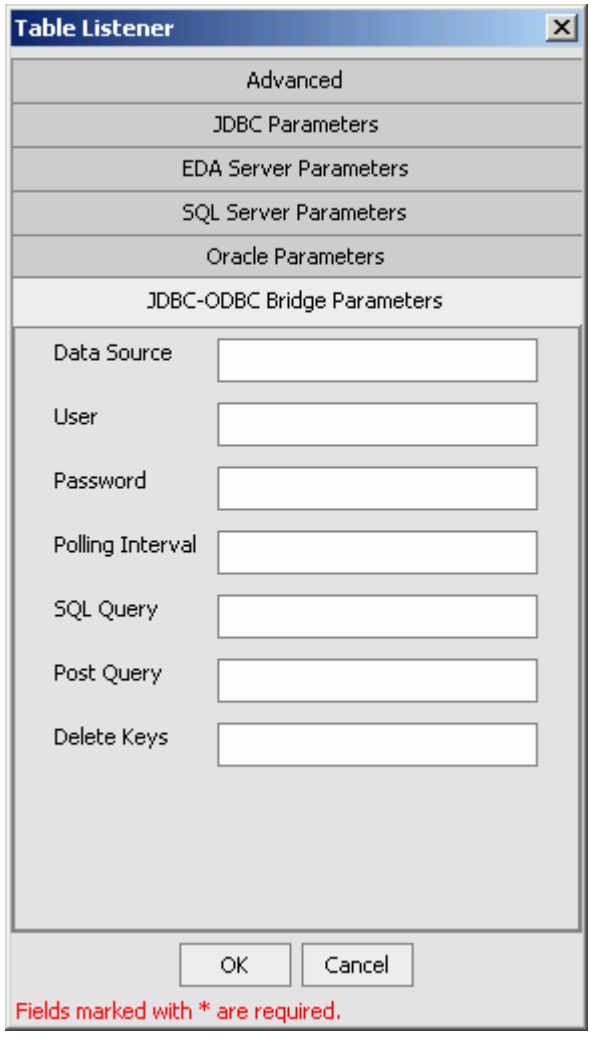
b. In the Description field, type a brief description.

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

d. To associate one or more available ports with this channel, select the port in the Available box and click the double right arrow (>>) button to move it to the Selected box.

4. Click *Next*.

The Table Listener dialog box opens.



This dialog box has tabs that represent the available listeners. They are Oracle, SQL Server, EDA Server, JDBC, or JDBC-ODBC listener. Each tab displays parameter fields related to the specific listener. There is also an Advanced tab.

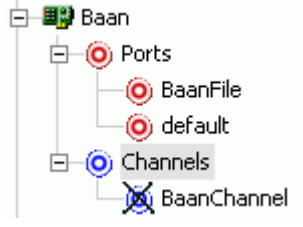
Note: If you are configuring listening capabilities for a non-relational database, select an EDA Server listener.

5. Select a listener by clicking the corresponding tab.

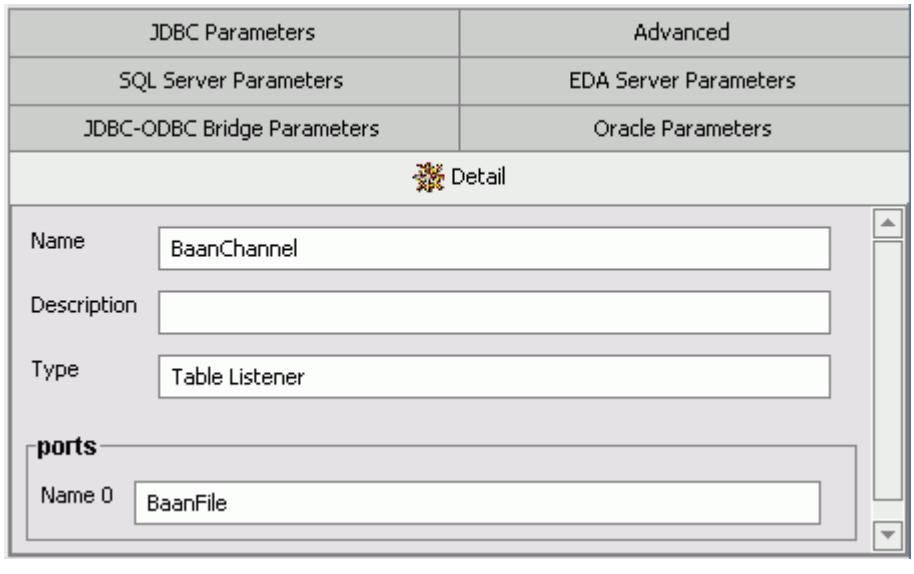
   Enter the system information that is specific to the database on which you are listening.

6. Click *OK*.

   The channel appears below the Channels node in the left pane.

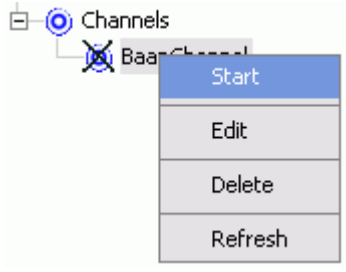


When you select the event port, the channel information appears in the right pane.



A Ports area appears on the Details tab that displays the name of the event port you assigned to this channel.

You are ready to start your channel to listen for events.



7. To activate your event configuration, right-click the channel node, for example, BaanChannel.

   a. Select *Start*.

   b. To stop the channel at any time, right-click the channel and select *Stop*.

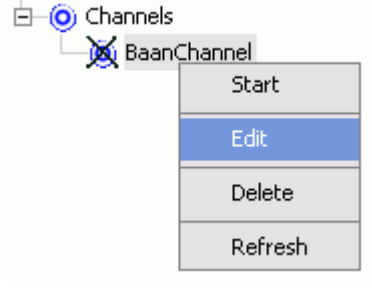
Modifying a Channel

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

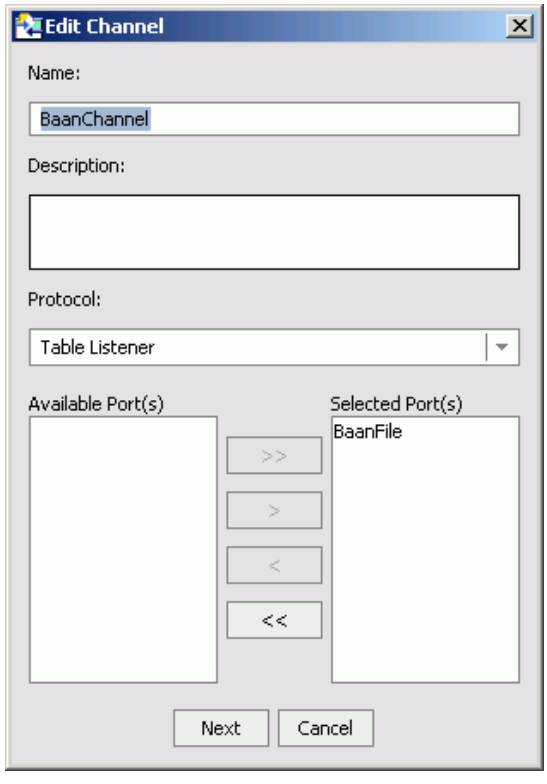
To edit a channel:

1. To view the available channels, click the *Channels* node in the left pane.



2. Right-click the channel you want to edit, for example, BaanChannel, and select *Edit*.

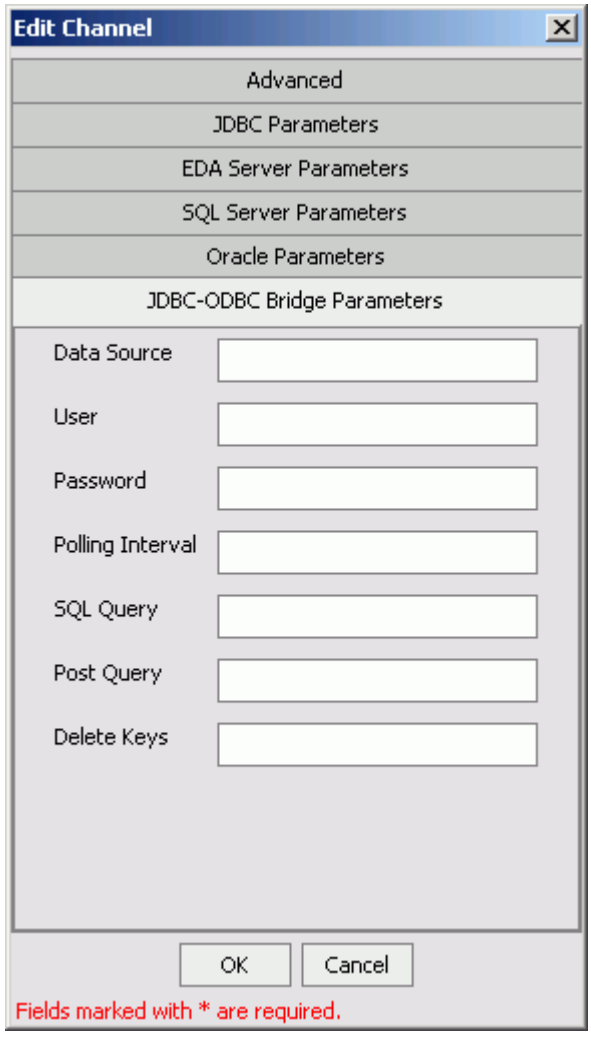
   The Edit Channel dialog box opens.



3. Make the required changes to the channel configuration.

4. Click *Next*.

The following dialog box opens.

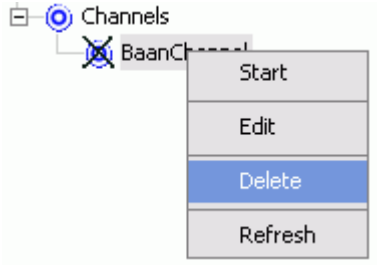


5. Make the required changes and click *OK*.

Procedure: How to Delete a Channel

To delete an existing channel:

1. In the left pane, right-click the channel, for example, BaanChannel.



2. Select *Delete*.

   The channel disappears from the Channels list.

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
- 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.

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.

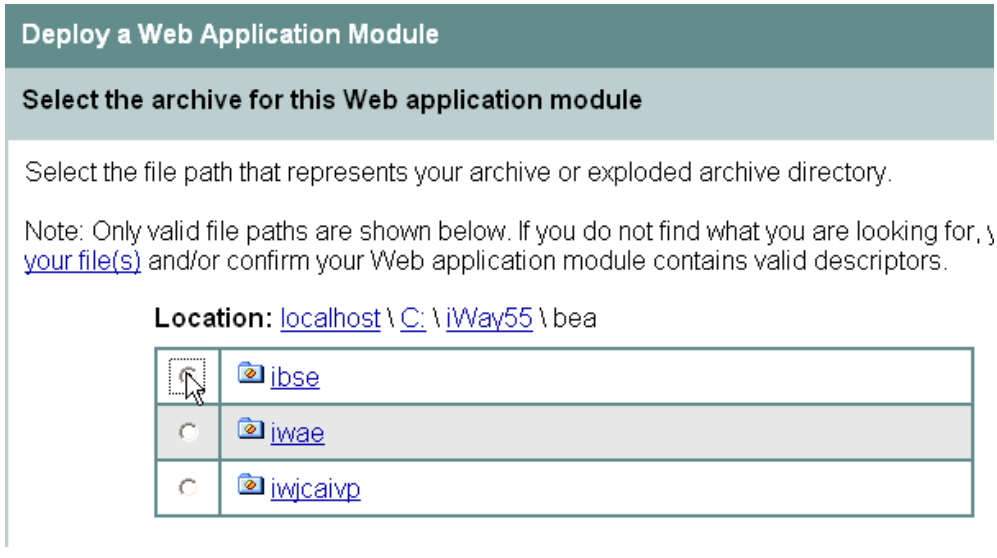
For more information on configuring WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

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*.

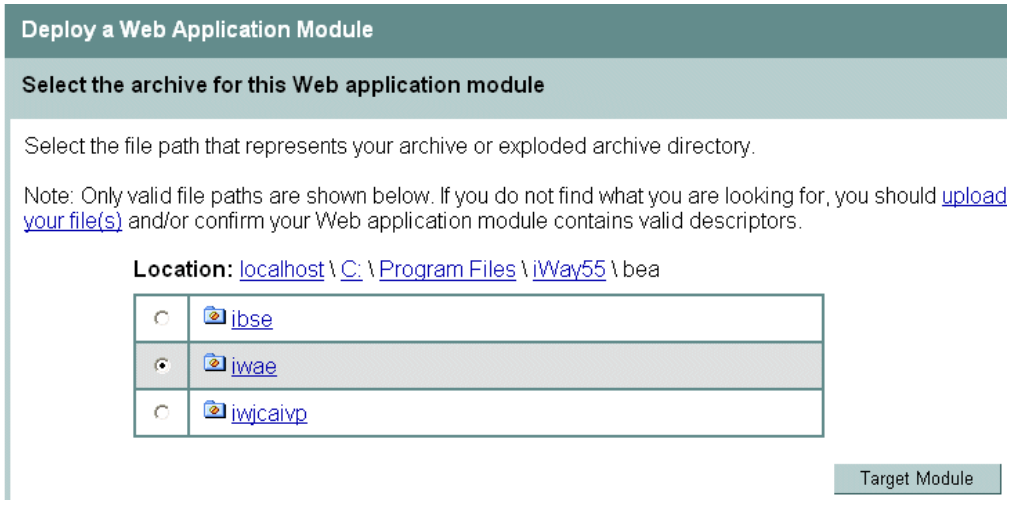
A page appears for you to specify where the Web application is located.

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

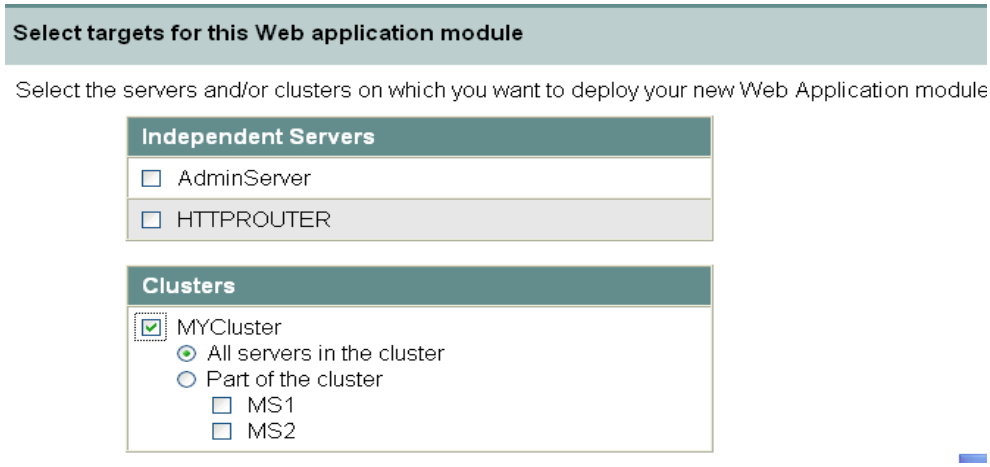


5. To deploy servlet Application Explorer, select the option button next to the iwae directory and then click *Target Module*.

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

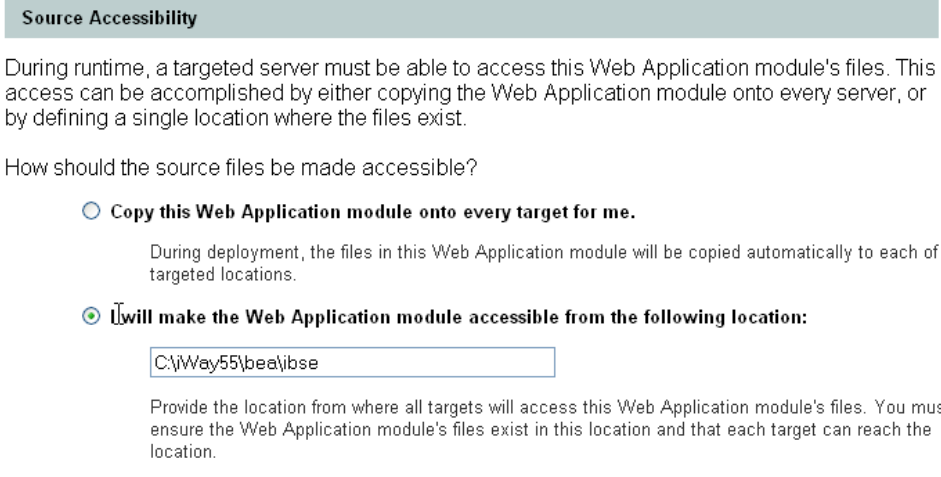


The following window opens.



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

The following window opens.



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 Configure Ports and Channels in a Clustered Environment

You can use Swing Application Explorer deployed in BEA WebLogic WorkShop or Servlet Application Explorer to configure ports and channels in a clustered environment.

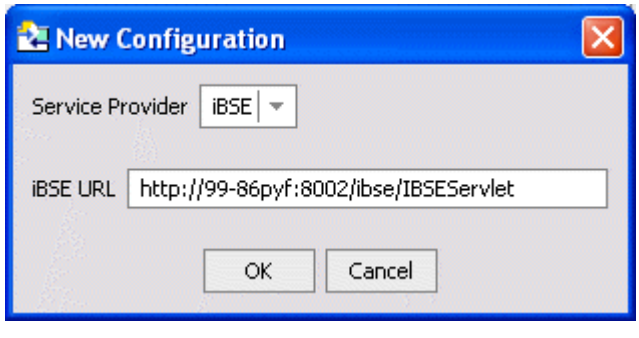
Note: Before using Servlet Application Explorer in a clustered environment, you must edit the web.xml file and specify the correct URL to your iBSE deployment. The default location on Windows is:

```
C:\Program Files\iWay55\bea\iwae\WEB-INF\web.xml
```

For more information on configuring the web.xml file for Servlet Application Explorer, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

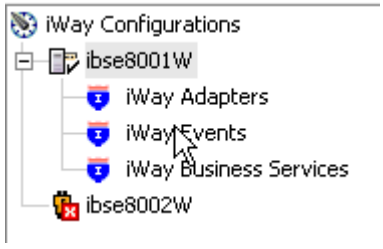
To configure ports and channels in a clustered environment:

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 *Creating a New Configuration* in Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services*.



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

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

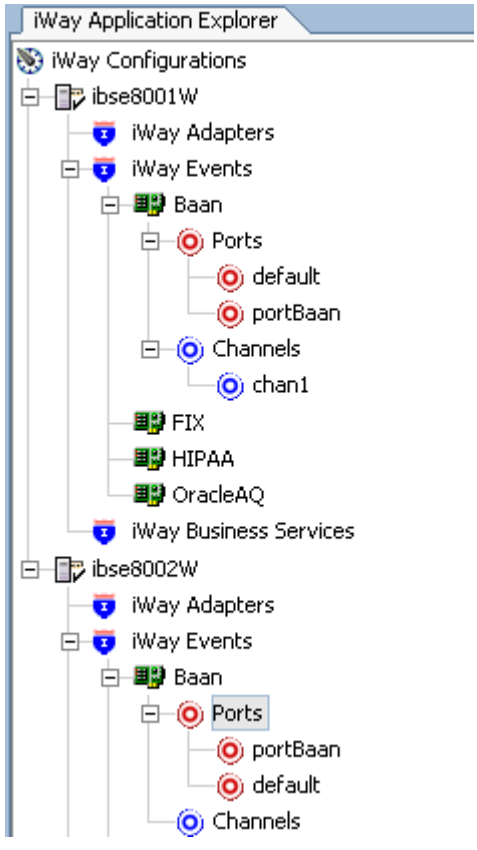


4. Add a new port for the Baan adapter. For more information, see *Creating an Event Port* on page B-3
5. Create a channel and add the port you created. For more information, see *Creating a Channel* on page B-20.
6. Click *Next* and enter the application server parameters.
7. Start the channel.

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 graphic shows two configurations.



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.

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail:	Documentation Services - Customer Support Information Builders, Inc. Two Penn Plaza New York, NY 10121-2898
Fax:	(212) 967-0460
E-mail:	books_info@ibi.com
Web form:	http://www.informationbuilders.com/bookstore/derf.html

Name:_____

Company:_____

Address:_____

Telephone:_____Date:_____

E-mail:_____

Comments:

Reader Comments

Information Builders, Two Penn Plaza, New York, NY 10121-2898 **(212) 736-4433**