iWay Software
An Information Builders Company

iWay

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

bea 8.1 VALIDATED
BEA WEBLOGIC PLATFORM

February 11, 2005
DN3501476.0205

Preface

This document is written for system integrators who develop client interfaces between CORBA and other applications. It describes how to use the iWay Adapter for CORBA for BEA WebLogic and how to develop application environments with a specific focus on message integration.

How This Manual Is Organized

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

Chapter/Appendix		Contents
1	Introducing the iWay Adapter for CORBA for BEA WebLogic	Introduces the iWay Adapter for CORBA for BEA WebLogic.
2	Creating XML Schemas and Business Services	Describes how to open a connection to CORBA, create request and response schemas for CORBA integration objects, and create business services.
3	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
4	Management and Monitoring	Describes how to configure and use monitoring tools provided by iBSE and JCA to gauge the performance of your run-time environment.
A	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and Web services.
B	Using CORBA Implementations With the Adapter	Provides details about using the adapter with JacORB, VisiBroker for Java, and Orbacus.
C	Supported IDL Types	Lists the Interface Definition Language (IDL) types that are supported by the iWay Adapter for CORBA for BEA WebLogic.

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

Related Publications

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

Customer Support

Do you have questions about iWay Adapter for CORBA 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 CORBA 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 CORBA for BEA WebLogic

Topics:

- About the iWay Adapter for CORBA for BEA WebLogic
- Features of the iWay Adapter for CORBA for BEA WebLogic
- Getting Started With the iWay Adapter for CORBA for BEA WebLogic
- Deployment Information for the iWay Adapter for CORBA for BEA WebLogic

This section introduces the iWay Adapter for CORBA for BEA WebLogic, describes its features, and provides an overview of how it works.

About the iWay Adapter for CORBA for BEA WebLogic

A number of companies and application providers used Common Object Request Broker Architecture (CORBA) for internet and legacy C++ application development, particularly before the popularity of Java or J2EE and XML or Web service architectures. The iWay Adapter for CORBA for BEA WebLogic integrates existing CORBA services with a BEA WebLogic Server so that existing IT investments can be integrated into J2EE applications and deployed as Web services.

The iWay Adapter for CORBA for BEA WebLogic enables CORBA-based applications on a BEA WebLogic Server to communicate with other applications integrated by the iWay adapter suite. Access to CORBA environments is provided through the adapter, which uses CORBA Interface Definition Language (IDL) entries to generate local, remote-based services. Applications make calls to a remote service that, in turn, invokes a CORBA method that returns information to the service.

A CORBA object provides distributed object capability between applications in a network. Although a CORBA object is implemented using a standard programming language, each CORBA object has a clearly defined interface, defined using the CORBA IDL. The definition of a CORBA object is consistent with the definition presented by the Object Management Group (OMG). OMG has specifications and documents that provide complete details on objects. For more information, visit the OMG Web site, located at http://www.omg.org.

The iWay Adapter for CORBA for BEA WebLogic provides a means to exchange real-time business data between CORBA servers and other application, database, or external business partner systems.

The adapter uses BEA WebLogicintegration and XML messages to allow non-CORBA applications to communicate and exchange transactions with CORBA. Applications that need to cause a CORBA business event use the adapter to send request messages to CORBA. If the request is for retrieving data from CORBA, then the adapter sends the application a response message with the data.

Features of the iWay Adapter for CORBA for BEA WebLogic

The iWay Adapter for CORBA for BEA WebLogic supplies everything you require to integrate your business processes and enterprise applications with your CORBA system. The iWay Adapter for CORBA for BEA WebLogic provides the following features:

- Support for several ORBs, including JacORB, VisiBroker for Java, and Orbix.

  Note: The adapter supports Orbix using orbacus libraries and IR.

- Guaranted synchronous and asynchronous bi-directional message interactions between a BEA WebLogic Server and an ORB.
- Data transfer between a process running within a BEA WebLogic Server and an ORB.

- Service adapter integration operations providing end-to-end business process management using XML schemas.
- Application Explorer, which can be used to create XML schemas and Web services.
- Integration without custom coding.
- Capability of business processes to request and receive data from your CORBA system using services.
- Standards-based adapter services.

  The adapter services provide extensions to the J2EE Connector Architecture (JCA) version 1.0 from Sun Microsystems, Inc.

  For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`
- Scalable, reliable, and secure access to your CORBA system.

  BEA WebLogic provides clustering, load balancing, and resource pooling for a scalable solution.

  For more information about scalability, see the following URL:

  `http://edocs.bea.com/wls/docs81/cluster/index.html`
- The benefits of the fault-tolerant features of BEA WebLogic.

  For more information about high availability, see the following URL:

  `http://edocs.bea.com/wli/docs81/deploy/index.html`
- The security features of BEA WebLogic and the security of your CORBA system.

  For more information about security, see the following URL:

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

Getting Started With the iWay Adapter for CORBA for BEA WebLogic

This topic gives an overview of how to get started using the iWay Adapter for CORBA for BEA WebLogic within the context of an application integration solution. Integration with CORBA involves the following steps:

- Designing the Application Integration Solution.
- Determining the Required CORBA Business Workflows.
- Generating Schemas for CORBA Integration Objects.

Designing the Application Integration Solution

Designing an application integration solution includes (but is not limited to) tasks such as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining the components that will be involved in the integration, such as Web services or business processes designed in WebLogic Workshop.
- Determining the external systems and technologies that will be involved in the integration, such as CORBA systems and other EIS.
- Determining the iWay adapters that will be required, such as the iWay Adapter for CORBA for BEA WebLogic. An application integration solution can involve multiple adapters.

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

Determining the Required CORBA Business Workflows

Within the larger context of an application integration project, you must determine the specific CORBA integration objects and workflows that are required to support the business processes in the application integration solution.

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

- Type of CORBA integration objects, workflows, and transport used to access the CORBA system.
- CORBA transactions involved in business processes.
- Logons required to access CORBA transports and perform the required operations.
- Determining whether services should be processed synchronously or asynchronously.

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

Generating Schemas for CORBA Integration Objects

After identifying the CORBA integration objects and workflows required for the application integration solution, you must generate the XML schemas to use to exchange data with one or more CORBA systems. Services require two XML schemas: one for the CORBA request and another for the CORBA response.

Use Application Explorer to generate XML schemas for CORBA operations. For more information, see Chapter 2, *Creating XML Schemas and Business Services*.

Deployment Information for the iWay Adapter for CORBA for BEA WebLogic

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

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

iWay Application Explorer is used to configure database connections and create Web services and events. It 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 iWay Application Explorer and the adapters.

Deployment Information Roadmap

The following table lists the location of deployment information for the iWay Adapter for CORBA and iWay Application Explorer. 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
iWay Application Explorer	Chapter 2 and Appendix A of this guide *iWay Installation and Configuration for BEA WebLogic* *iWay Servlet Application Explorer for BEA WebLogic User's Guide*
Integration Business Services Engine (iBSE)	*iWay Installation and Configuration for BEA WebLogic*
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	*iWay Connector for JCA for BEA WebLogic User's Guide* *iWay Installation and Configuration for BEA WebLogic*

The Integration Business Services Engine

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

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

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

Web services is a distributed programming architecture that overcomes hurdles with Enterprise Application Integration (EAI) 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

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 the BEA WebLogic Server.

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

CHAPTER 2

Creating XML Schemas and Business Services

Topics:

- Integrating With CORBA
- Starting Servlet iWay Application Explorer
- Creating and Managing a Connection
- Creating an XML Schema
- Understanding Integration Business Services

The iWay Adapter for CORBA for BEA WebLogic uses XML documents to communicate with your CORBA system integration objects for services. The format of these XML documents is determined by schemas you generate using Application Explorer.

This section describes how to open a connection to CORBA, create XML schemas for CORBA integration objects, and create business services (or Web services).

Integrating With CORBA

iWay Application Explorer supports the creation of schemas based on specific tables and resulting answer sets. To obtain metadata about the Object Request Broker (ORB), Application Explorer connects to the Interface Repository. The iWay Adapter for CORBA for BEA WebLogic extracts the definition of CORBA servers and converts them to XML schemas and service XML request and response definitions. You can see the original definitions of the CORBA servers using Application Explorer.

Application Explorer displays all objects defined by its Interface Definition Language (IDL) that are loaded into the Interface Repository (IFR). After creating a connection in Application Explorer, you can use the Explorer to verify that the system definition was entered correctly.

Using the iWay Adapter for CORBA for BEA WebLogic, Application Explorer populates each object with data retrieved from the Interface Repository. The contents of the Interface Repository appear in the Application Explorer tree. Expanding this interface displays its methods, return arguments, and parameters. Application Explorer displays a list of modules, appearing as folders. These IFR modules represent the different work areas of CORBA.

The XML schema defines the format of XML requests and corresponding responses to the service adapter. The schema is a language-neutral interface description in XML format that declares the types, objects, and methods for the CORBA system. Conceptually, the XML schema is the same as the CORBA IDL.

Note: Before creating schemas, you can save time by verifying that your ORB infrastructure is properly configured, your server is registered in the Naming Service or its object reference is available, and your interface repository (IFR) is running and populated.

Starting Servlet iWay Application Explorer

Before you can use Application Explorer, you must start the server where Application Explorer is running.

Procedure How to Start Application Explorer

1. Ensure the server is started where Application Explorer is running.
2. Enter the following URL in your browser window:

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

   where:

   hostname

   Is the machine where Application Explorer is installed.

   port

   Is the port number for iBSE. The default port is 7001.

   Application Explorer opens.

   The Available Hosts drop-down list appears in the upper-right. 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 your CORBA application, view metadata, and create schemas and business services.

   Event Adapters, where you configure event listening for available applications.

   Integration Business Services, where you run business services.

   Depending on the active tab, the left pane of the window contains an expandable list of adapter nodes (based on the iWay adapters installed), events, or business services. The right pane provides details of a selected adapter, event, or service and is the work area where you define and modify adapter functions and services.

   The Available Hosts drop-down list specifies to which Servlet iBSE instance or JCA instance you connect.

   For more information on accessing different instances of a JCA installation or a Servlet iBSE, see the *iWay 5.5 Installation and Configuration* documentation.

   You are now ready to define a new target to your CORBA application.

Creating and Managing a Connection

To access an adapter, you must define a target that connects to the adapter. After the defined target is created, it automatically is saved. You must establish a connection to the defined target every time you start Application Explorer or after disconnecting.

Although you can maintain multiple open connections, iWay Software recommends disconnecting from targets that are not in use. For more information, see *How to Disconnect From a Defined Target* on page 2-9.

After you create a defined target using Application Explorer, you can edit any information that you provided during the creation process. For more information, see *How to Edit a Defined Target* on page 2-9.

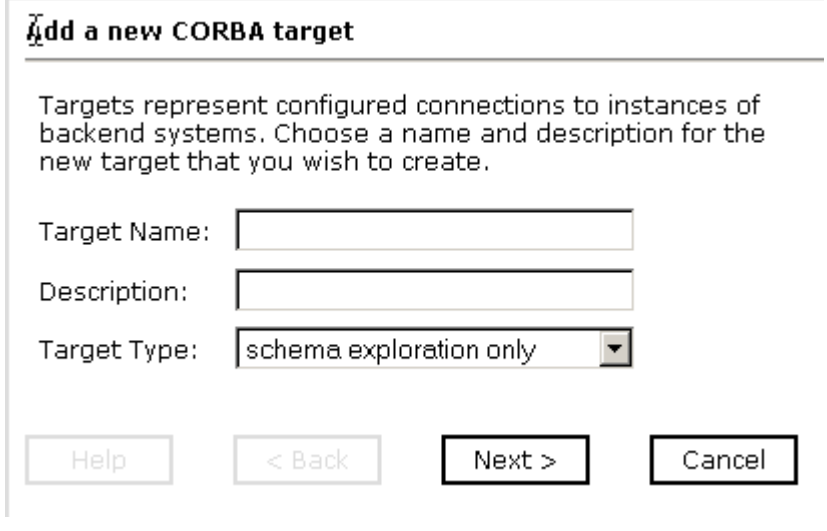
You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of CORBA targets in the left pane of the explorer. For more information, see *How to Delete a Defined Target* on page 2-10.

Procedure How to Define a New Target

To define a new target:

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

   The Add a new CORBA target dialog box opens in the right pane.



a. In the Target Name field, type a descriptive name for the target (for example, CORBA_Connection).

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

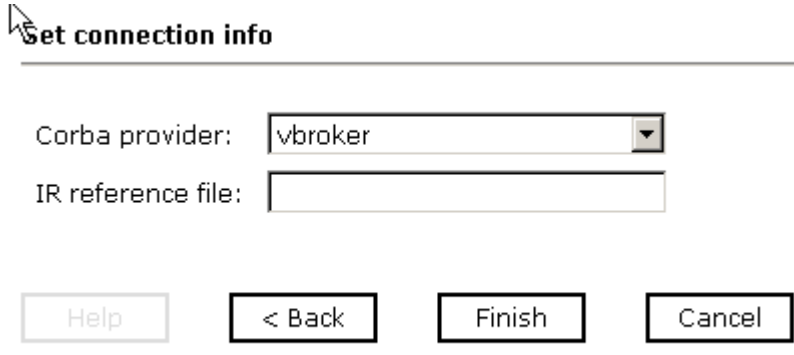
c. From the Target Type drop-down list, select a target type (for example, schema exploration only).

For information about the target type options, see the following steps, including the table that lists the target parameters and a description of each.

4. Click *Next*.

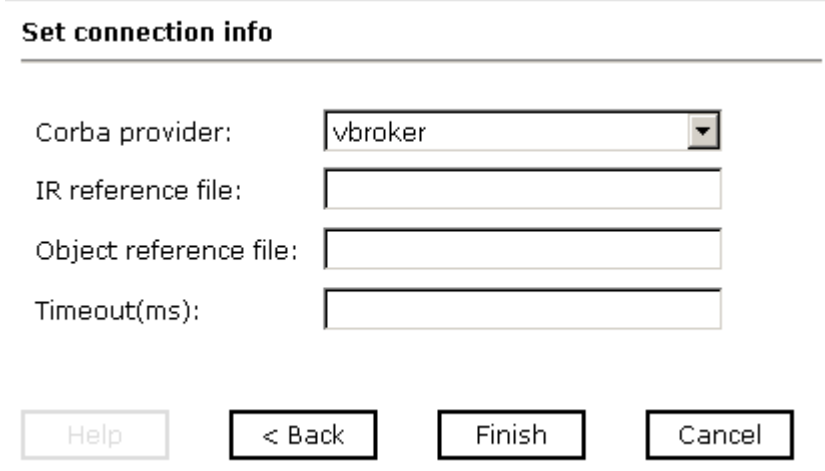
The dialog box that opens depends on the target type you selected.

If you selected a target type of **schema exploration only**, the Set Connection info dialog box opens containing two fields (Corba provider and IR reference file) and three active buttons (Back, Finish, and Cancel) as shown in the following image.

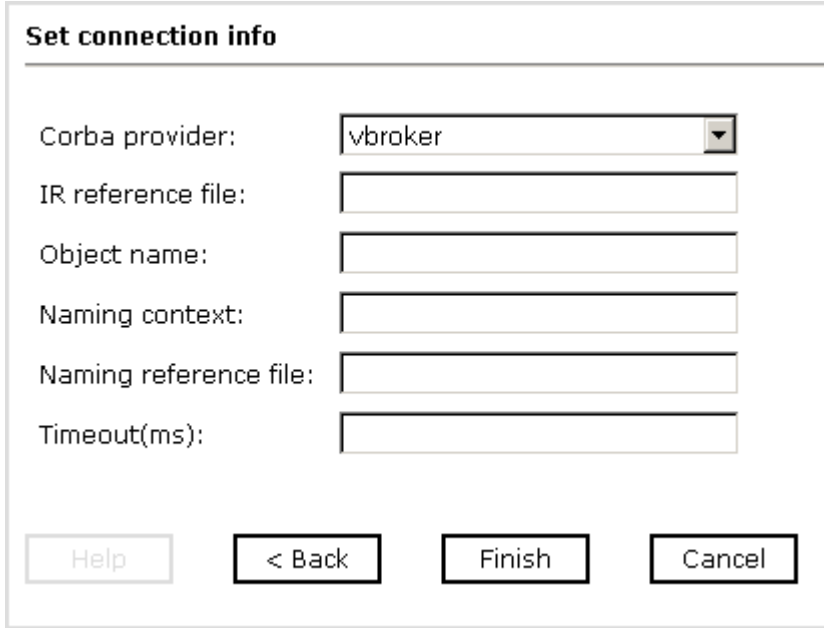


Note: The CORBA connection parameters are consistent with those found in your CORBA system. For more information on parameter values that are specific to your CORBA configuration, consult your CORBA system administrator.

If you selected a target type of **configuration without naming**, the Set Connection info dialog box opens containing four fields, Corba provider, IR reference file, Object reference file, and Timeout(ms) and three active buttons (Back, Finish, and Cancel) as shown in the following image.



If you selected a target type of **configuration with naming**, the Set Connection info dialog box opens containing six fields, Corba provider, IR reference file, Object name, Naming context, Naming reference file, and Timeout(ms) and three active buttons (Back, Finish, and Cancel) as shown in the following image.



5. Type the appropriate information for your target type based on the information in the following table which lists the target parameters and a description of each.

Target Parameter	Description
Corba provider	Type of client ORB libraries through which the service is sent. Possible values are vbroker, orbacus, and jacorb. For more information, see Appendix B, *Using CORBA Implementations With the Adapter*.
IR reference file	Name and path of the Interoperable Object Reference file that specifies the location of the Interface Repository service.
Object reference file	Name and path of the Interoperable Object Reference file that specifies the location of the CORBA object. Enables you to specify the location of an object using a direct IOR reference.

Target Parameter	Description
Timeout(ms)	Maximum time, in milliseconds, that a service waits for a CORBA object to respond before the service terminates. The default value, 0, specifies that the service waits indefinitely for a response.
Object name	Name of the object registered in the Naming Service (for example, bea.clubmed). This parameter, together with Naming context, enables you to specify the location of an object using an indirect Naming Service reference.
Naming context	ORB name context. This parameter, together with Object name, enable you to specify the location of an object using an indirect Naming Service reference.
Naming reference file	Name and path of the Interoperable Object Reference file that specifies the location of the Naming Service. This parameter, together with Naming context and Object name, enables you to specify the location of an object using an indirect Naming Service reference.

6. Click *Finish*.

   In the left pane, the target name appears under the node where you created the new target. You have finished creating the new target.

Procedure How to Connect to a Defined Target

To connect to a defined target:

1. In the left pane, expand the *Service Adapters* node.
2. Expand the *CORBA* node.
3. Click the target name (for example, CORBA_Connection) under the CORBA node.
4. In the right pane, move the pointer over *Operations* and select *Connect*.

   The Connect to CORBA_Connection dialog box opens, populated with values you entered for the connection parameters.
5. Verify your connection parameters and then click *OK*.

If the parameters are correct and the CORBA component is available, the node under the CORBA node displays a plus sign indicating that you are connected to the defined target. Otherwise, an error message appears in the right pane.

Procedure How to Disconnect From a Defined Target

To disconnect from a defined target:

1. Expand the *Service Adapters* node.
2. Expand the *CORBA* node.
3. Click the target name (for example, CORBA_Connection) under the CORBA node.
4. In the right pane, move the pointer over *Operations* and select *Disconnect*.

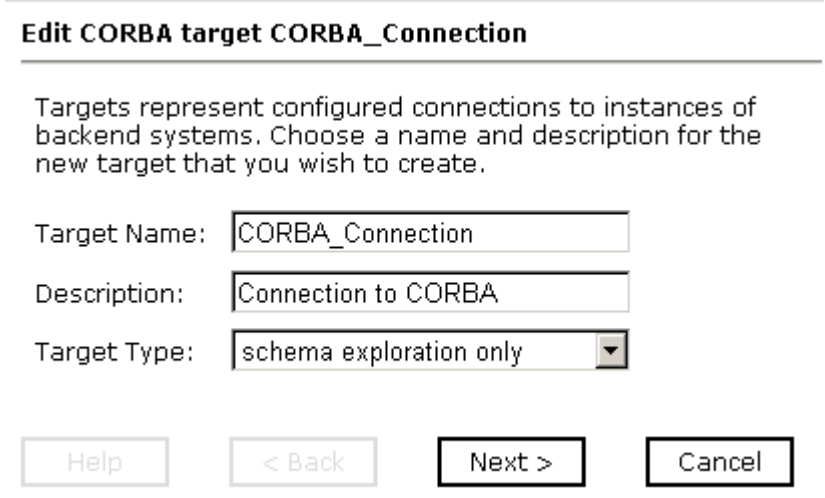
   Disconnecting from the application closes the connection, but the connection still appears in the left pane so that you can reopen it. The connection node now has an x icon, indicating that it is closed, as shown in the following image.

   When you want to re-establish a connection, Connect is available from the pop-up menu.

Procedure How to Edit a Defined Target

1. In the left pane of Application Explorer, expand the *Service Adapters* node.
2. Expand the *CORBA* node and select the defined target (for example, CORBA_Connection) you want to edit.
3. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit dialog box opens in the right pane containing three fields (Target Name, Description, and Target Type) and two active buttons (Next and Cancel) as shown in the following image.



4. Modify the target information as required and then click *Next*.

   The Set connection info dialog box opens in the right pane containing the connection parameters and three active buttons (Back, Finish, and Cancel).

5. Modify the connection information as required and then click *Finish*.

Procedure **How to Delete a Defined Target**

1. Expand the *Service Adapters* node.
2. Expand the *CORBA* node.
3. Click the target name (for example, CORBA_Connection) under the CORBA node.
4. In the right pane, move the pointer over *Operations* and select *Delete*.

   A message appears, prompting you to confirm the deletion of the node.

5. Click *OK*.

   The node disappears from the list of available connections.

Creating an XML Schema

Each service the iWay Adapter for CORBA for BEA WebLogic uses must be defined by a schema. Application Explorer generates XML schemas for service requests and service responses.

- **Service requests** are requests for action that your application makes to your CORBA system. As part of the definition, the request schema defines the input parameters required by the CORBA system.
- **Service responses** are the way the CORBA system responds to the service request. A service response schema defines this service response. Service requests always have a corresponding service response.

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

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

  ```
  C:\Program Files\iway55\bea\ibse\wsdl\schemas\service\CORBA\

  CORBA_Connection
  ```

  where:

  CORBA_Connection

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

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

  ```
  C:\Program Files\iWay55\config\base\schemas\CORBA\CORBA_Connection
  ```

  where:

  CORBA_Connection

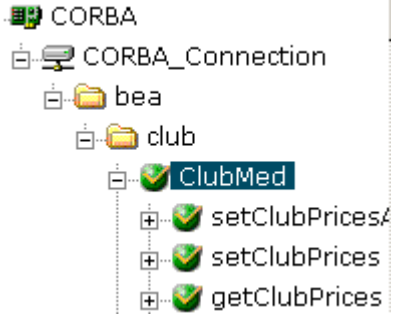
  Is the name of the connection to the CORBA system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

Procedure How to Create a Request Schema and a Response Schema

To create request and response schemas:

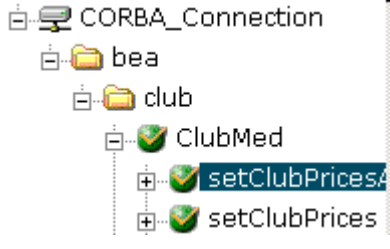
1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Expand the tree under the integration objects to see the items for which you can create a schema.

   The following image shows the tree with the CORBA connection expanded.



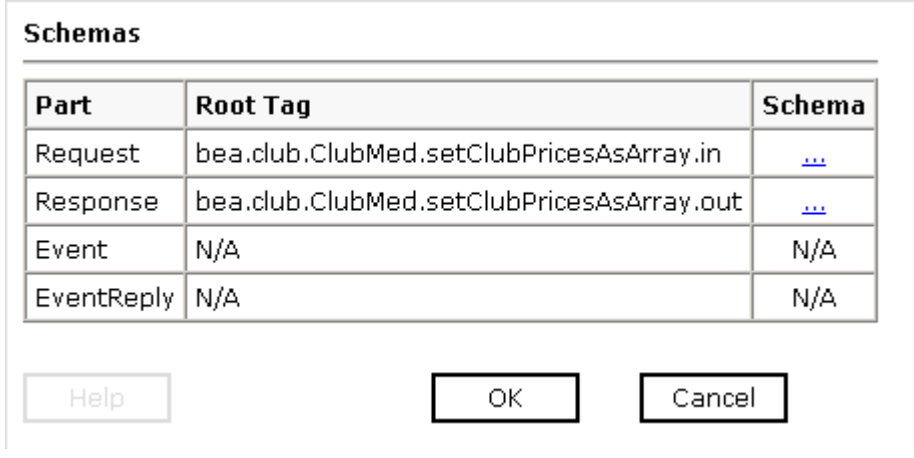
3. Expand and then select the node beneath the integration object for which you want to create the schema.

   The following image shows the tree with the integration object selected.



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

A table that lists the created schemas appears in the right pane as shown in the following image. The table has columns with labels for Part, Root Tag, and Schema. You can click in the schema column to view the corresponding request or response. The OK and Cancel buttons are active.



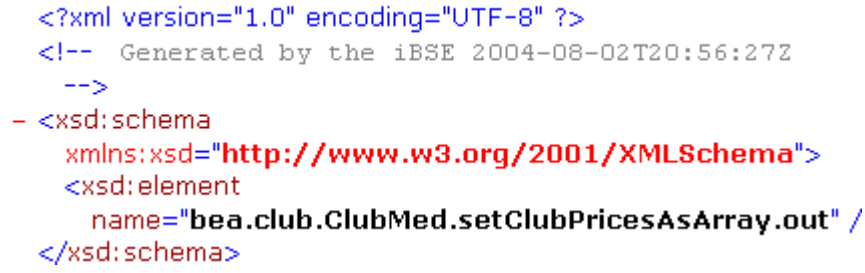
5. To view the request schema, click the ellipsis symbol that is located in the third column of the Request row.

The following image is an example of a request schema.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  Generated by the iBSE 2004-06-17T15:49:10Z
  -->
- <xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <xsd:element
      name="bea.club.ClubMed.setClubPricesAsArray.in">
    - <xsd:complexType>
      - <xsd:sequence>
          <xsd:element name="club"
            type="xsd:string" />
          <xsd:element name="prices"
            type="bea.club.pricesArray_type" />
        </xsd:sequence>
      - <xsd:attribute name="ref" use="optional">
        - <xsd:simpleType>
          - <xsd:restriction base="xsd:string">
```

6. To view the response schema, click the ellipsis symbol that is located in the third column of the Response row.

   The following image is an example of a response schema.



Understanding Integration Business Services

Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the adapter. The Integration Business Services Engine exposes functionality as Web services. It 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. A Web service integrates within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

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 *iWay Connector for JCA for BEA WebLogic User's Guide*.

After a business service is created, test it to ensure that it functions properly. iWay provides a test tool for testing the business service. For more information, see *How to Test a Business Service* on page 2-16.

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 BEA WebLogic Server. For more information, see *How to Generate WSDL From a Web Service* on page 2-18.

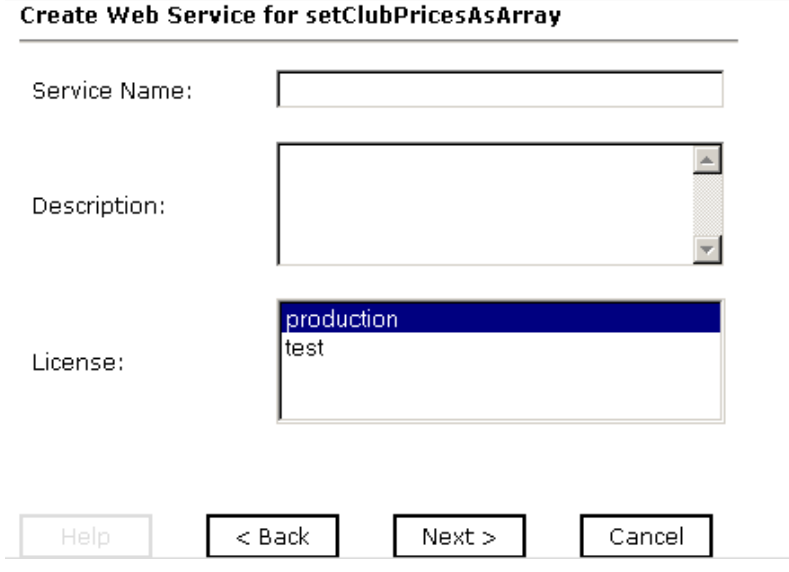
Procedure How to Create a Business Service

To create a business service:

1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Expand the tree under the integration objects to see the items for which you may create a schema.
3. Expand and then select the node beneath the integration object for which you want to create the business service.
4. In the right pane, move the pointer over *Operations* and select *Create Integration Business Services*.

   The Create Web Service information appears in the right pane.
5. Choose whether to create a new service or use an existing service.

   If you select **Create a new service**, the Create Web Service dialog box opens in the right pane.



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

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

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

If you select **Use an existing service**, a drop-down list appears from which you must select an existing service.

6. Click *Next*.

   A Create Web Service dialog box with the Method Name and Description fields opens.

   a. In the Method Name field, type a name to specify the name of the method.

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

7. Click *Finish*.

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

Procedure How to Test a Business Service

To test a business service:

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

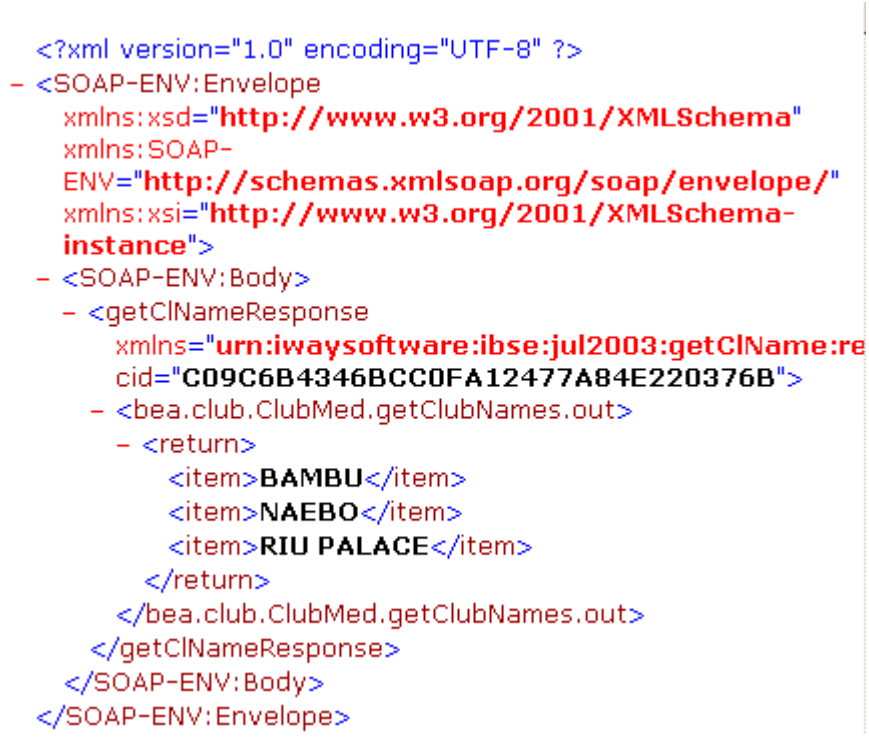
   The business service name appears as a link in the right pane.

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

The test option appears in the right pane as shown in the following image. This pane provides a text field in which to paste the XML input or a field to browse to a file that can be uploaded. In addition to the Browse button to the right of the browse filed are three active buttons (Upload, More, and Invoke).

6. Provide the appropriate XML input.
7. Click *Invoke*.

Application Explorer displays the results in the right pane.



Procedure How to Generate WSDL From a Web Service

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

   The link for the service appears in the right pane.
4. Right-click the *Service Description* link and choose *Save Target As*.
5. Choose a location for the file and specify .wsdl for the extension.

   Note: The file extension must be .wsdl.
6. Click *Save*.

CHAPTER 3

Using Web Services Policy-Based Security

Topics:

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

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

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

Integration Business Services Policy-Based Security

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

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

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

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

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

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

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

Configuring Integration Business Services Policy-Based Security

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

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 3-7.

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 3-11.

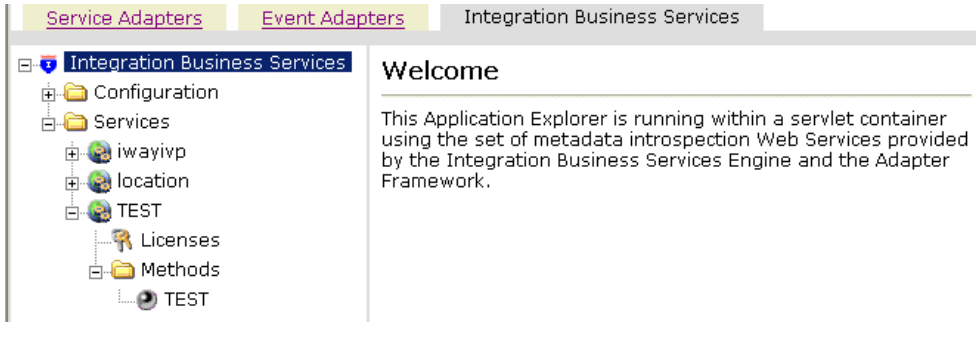
Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

Procedure How to Create a User to Associate With a Policy

To create a user to associate with a policy:

1. Open *Servlet Application Explorer*.

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



   a. Click the *Integration Business Services* tab.

   b. Expand the *Configuration* node.

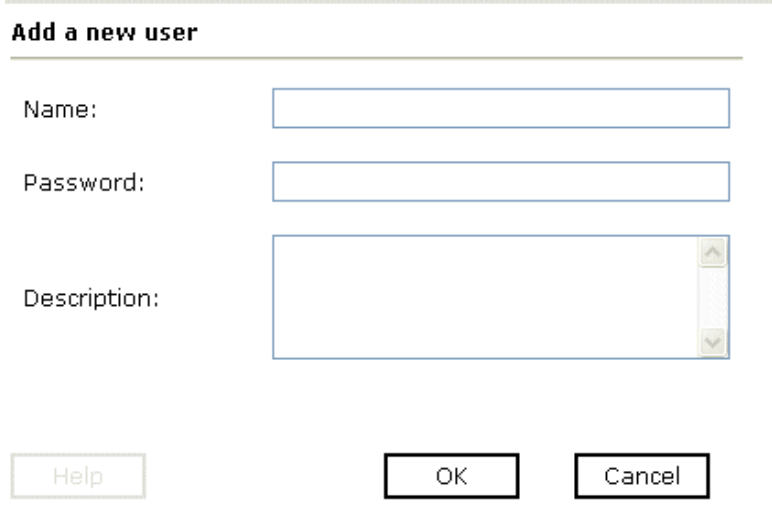
   c. Expand the *Security* node.

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.



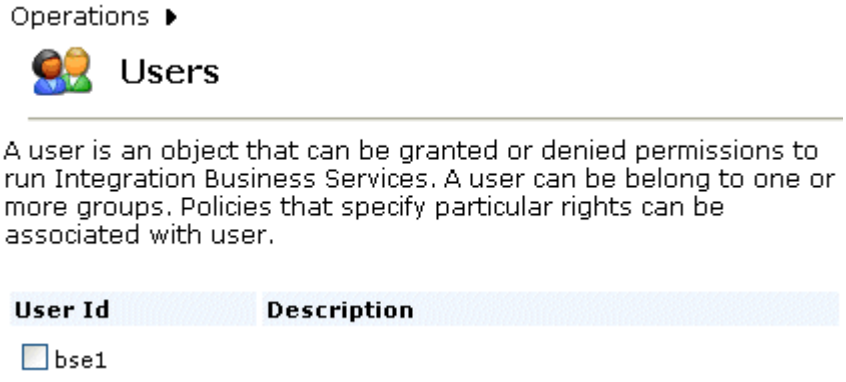
a. In the Name field, type a user ID.

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.

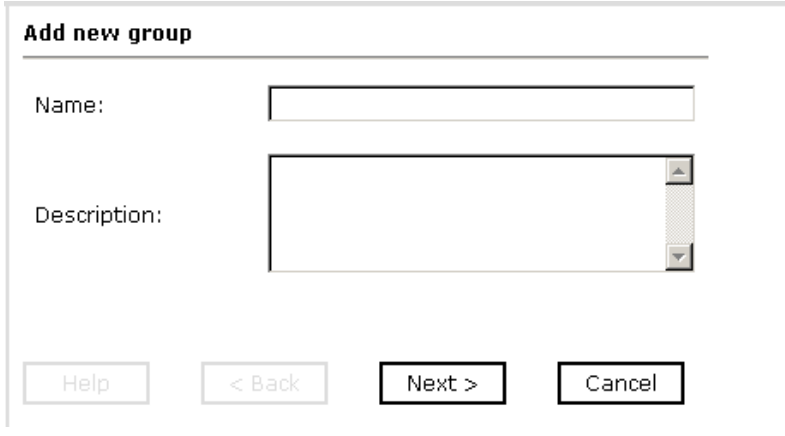


Procedure How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

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

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

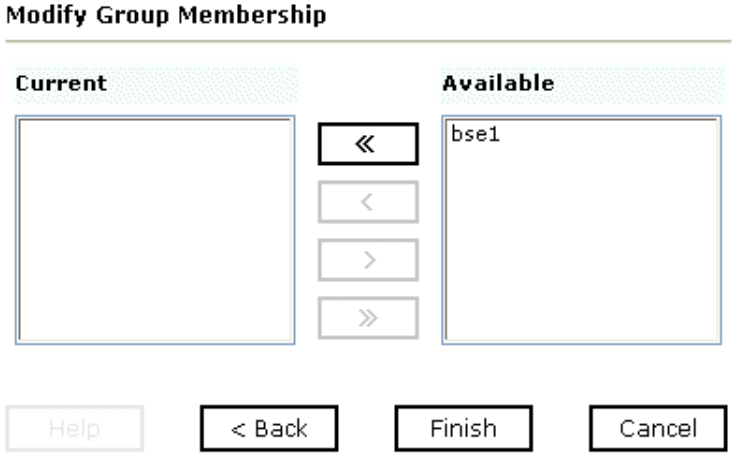


a. In the Name field, type a a name for the group.

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.

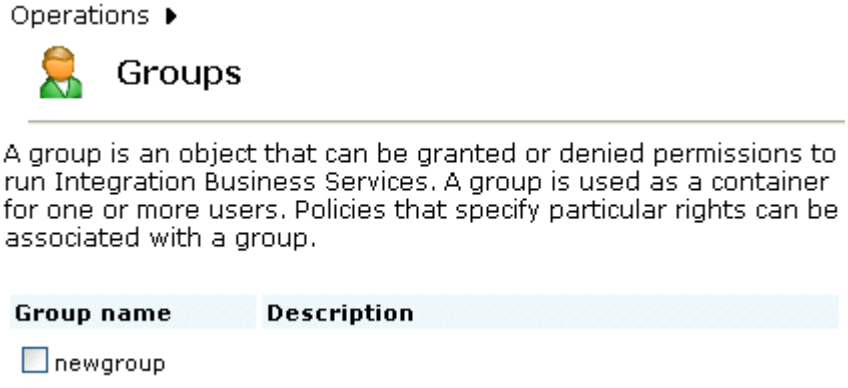


You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

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 Create an Execution Policy

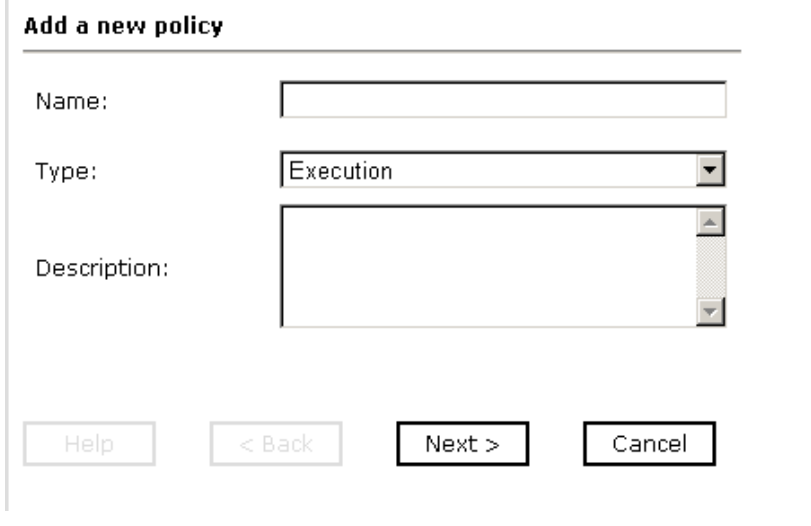
To create an execution policy:

1. Open *Servlet Application Explorer*.
   a. Click the *Integration Business Services* tab.
   b. Expand the *Configuration* node.
   c. Select *Policies*.

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

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

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



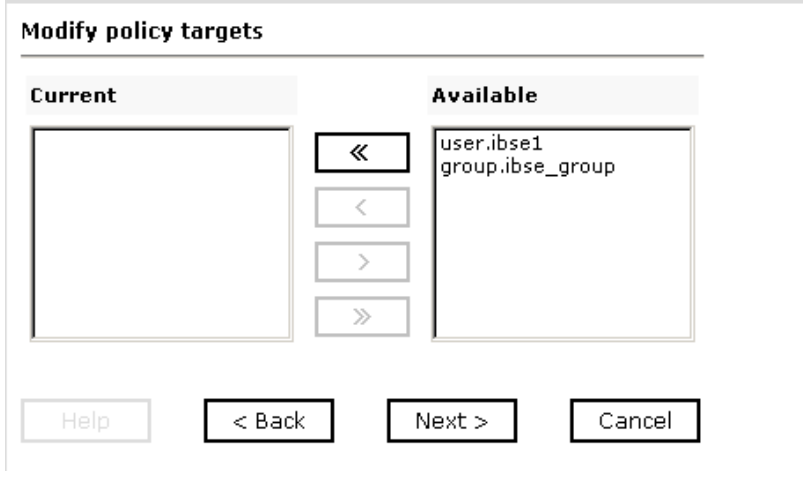
a. In the Name field, type a a name for the policy.

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.

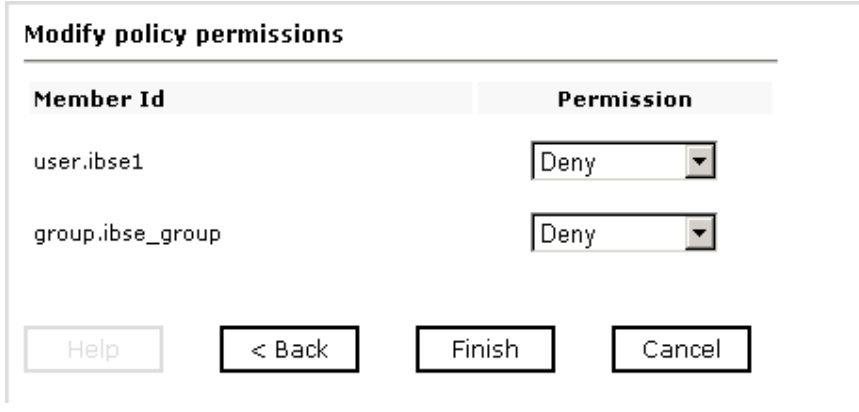


4. Select a minimum of one user or group from the Available pane.

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.

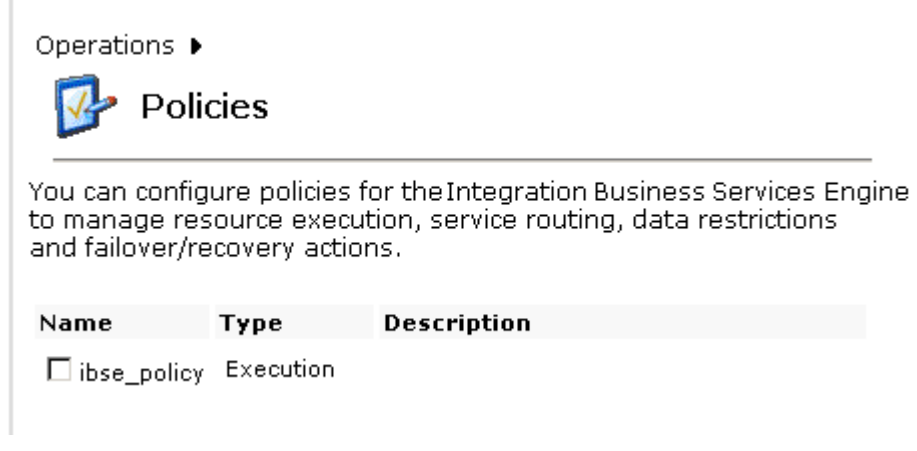
5. Click *Next*.

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.

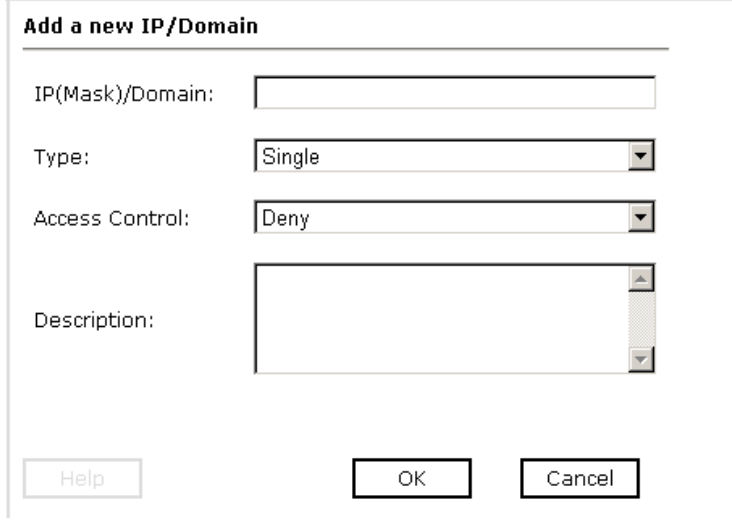


Procedure How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

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

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



   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.

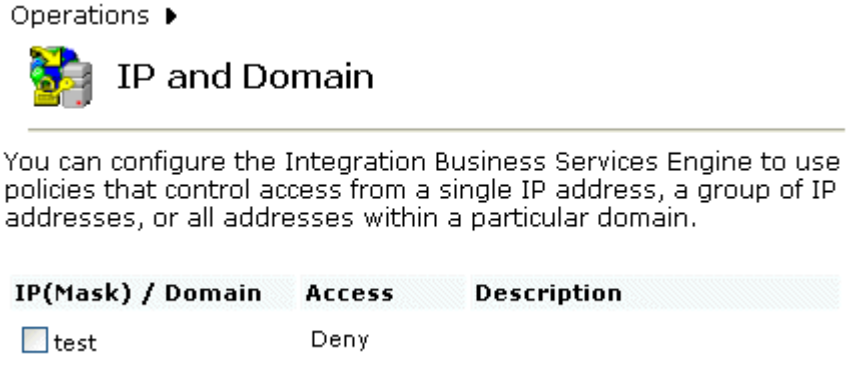
      If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

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 4

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 Integration Business Services Engine (iBSE) and the iWay Connector for JCA to measure the performance of your run-time environment. This section describes how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

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

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

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

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

   where:

   localhost

   Is the machine where the application server is running.

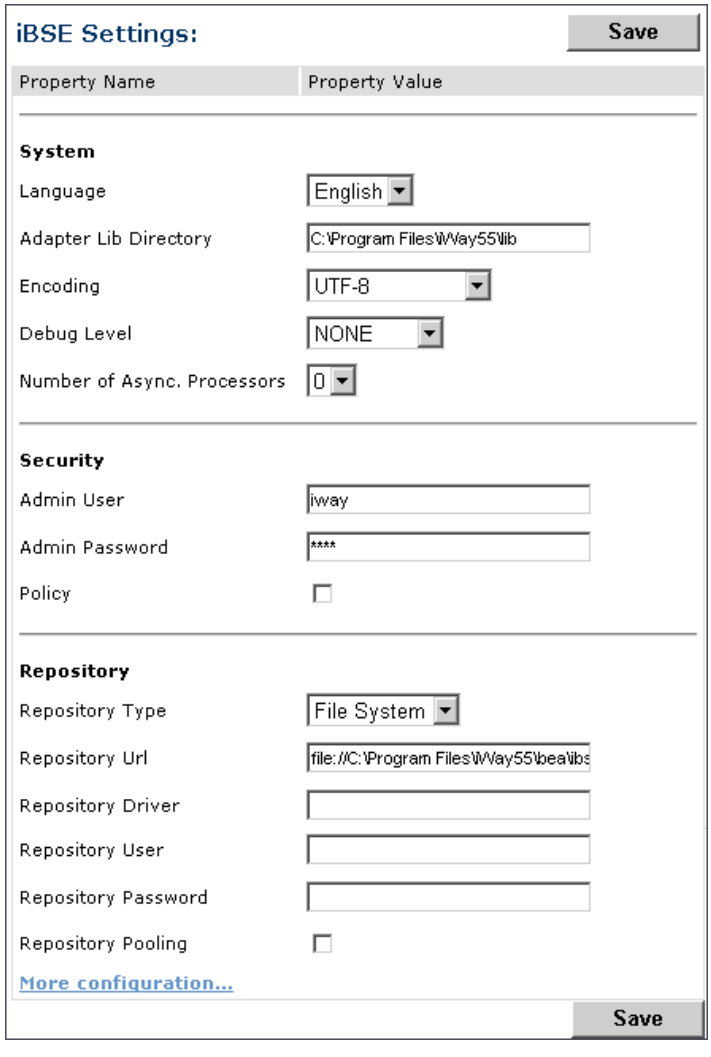
   port

   Is the HTTP port for the application server.

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

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

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



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:

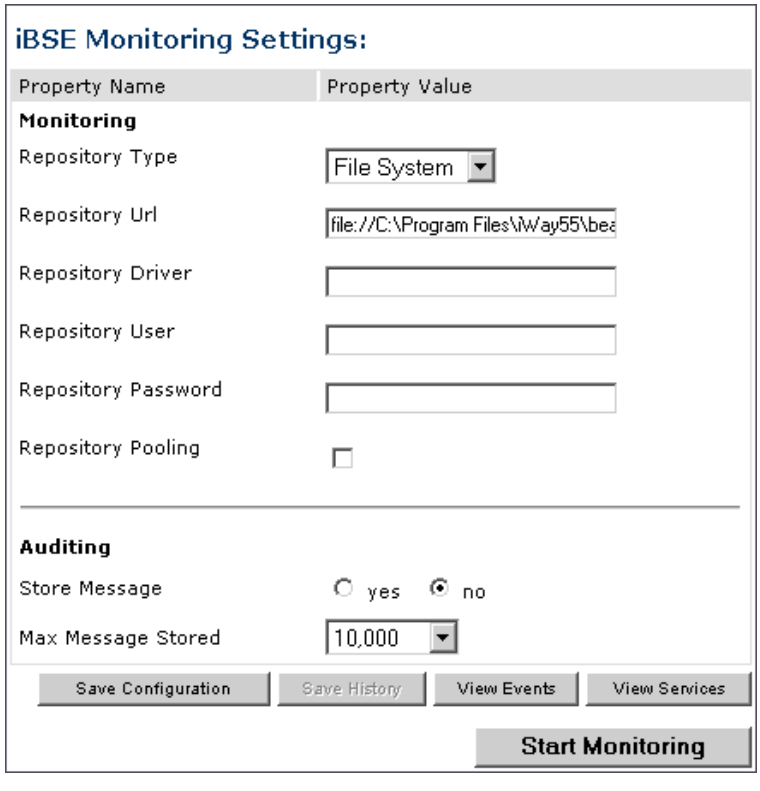
localhost

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

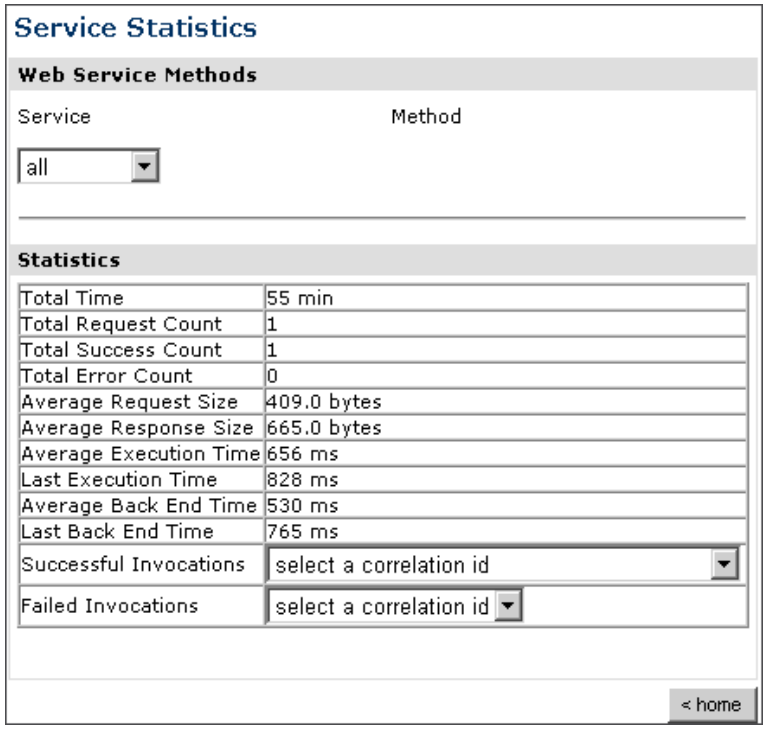
To monitor services:

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.



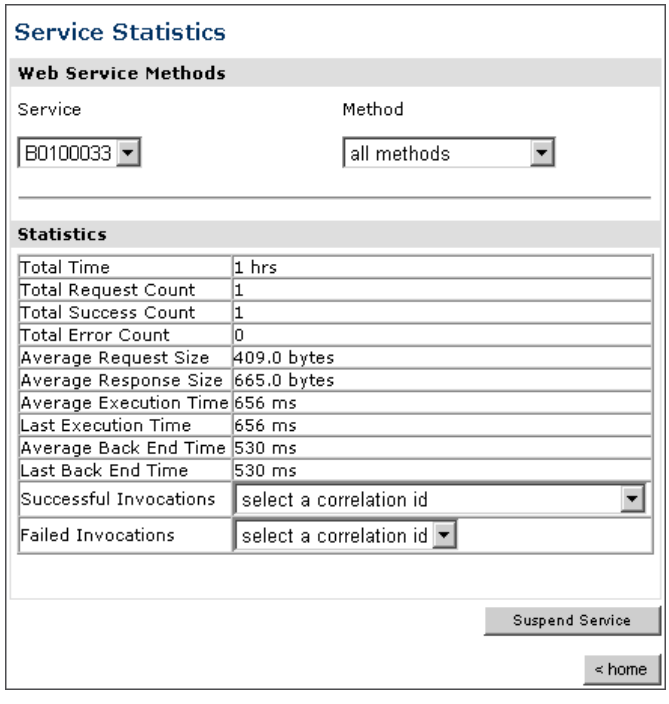
The system level summary provides services statistics at a system level.

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).

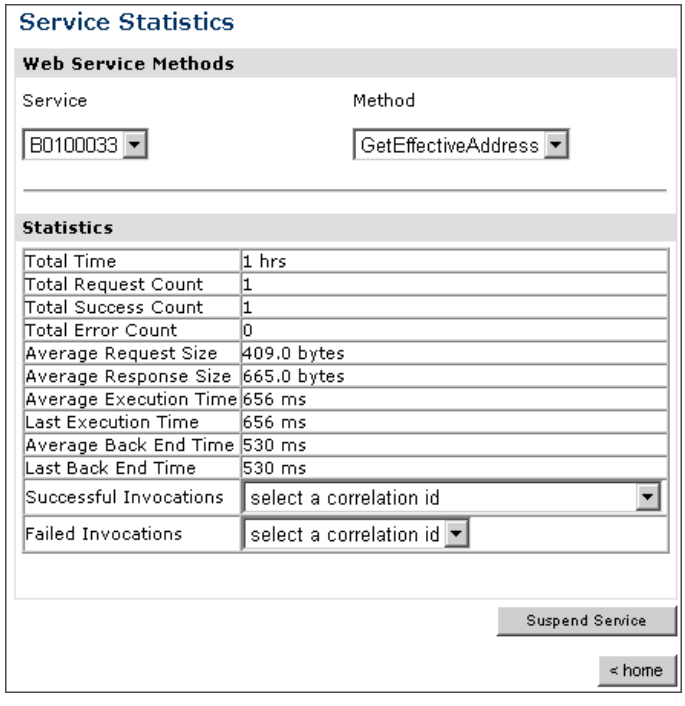


a. To stop a service at any time, click *Suspend Service*.

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

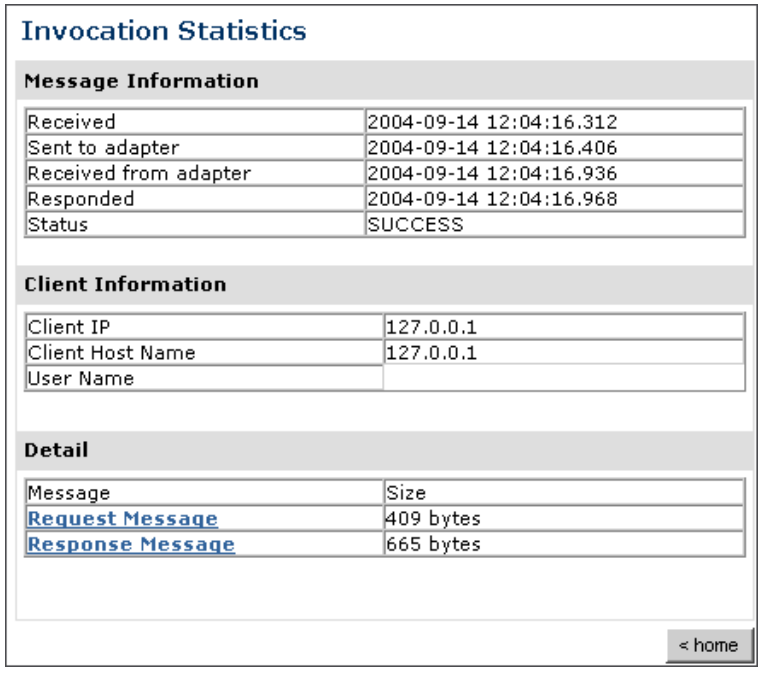
5. Select a method for the service from the Method drop-down list.

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.

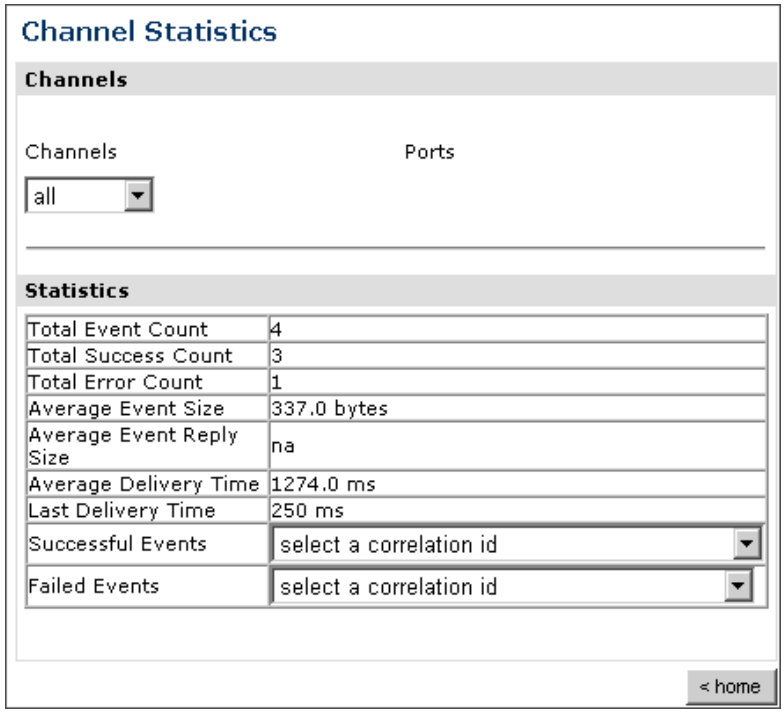
8. To return to the iBSE Monitoring Settings window, click *home*.

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.



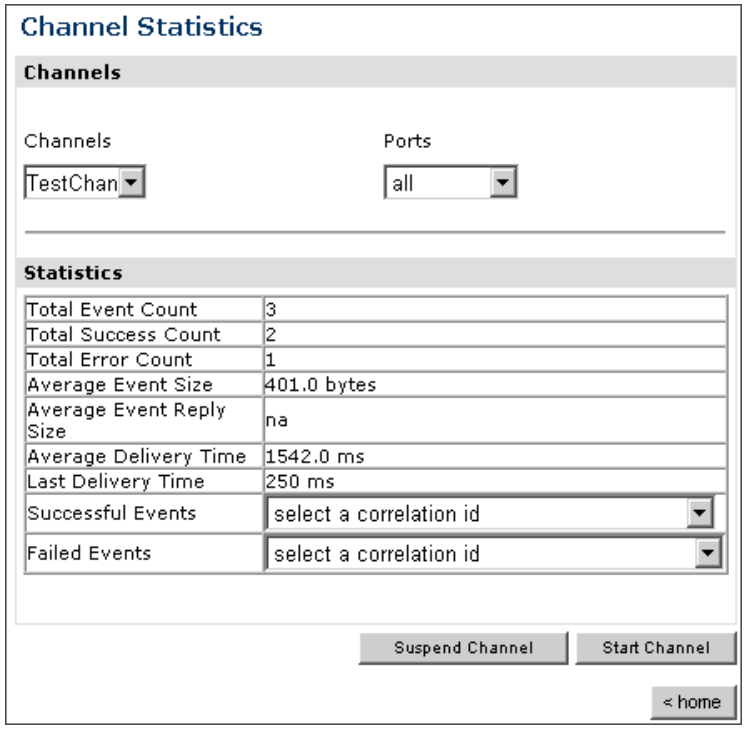
The system level summary provides event statistics at a system level.

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.

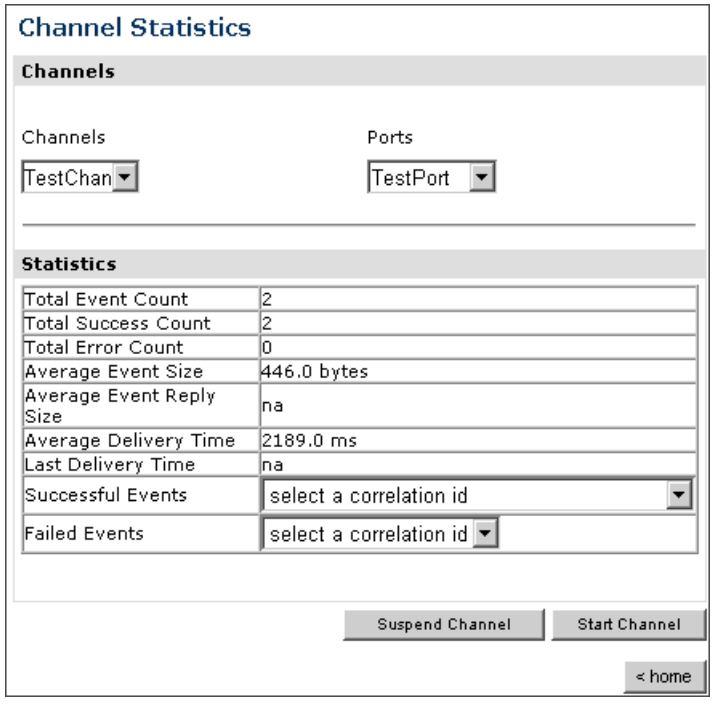


a. To stop a channel at any time, click *Suspend Channel*.

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

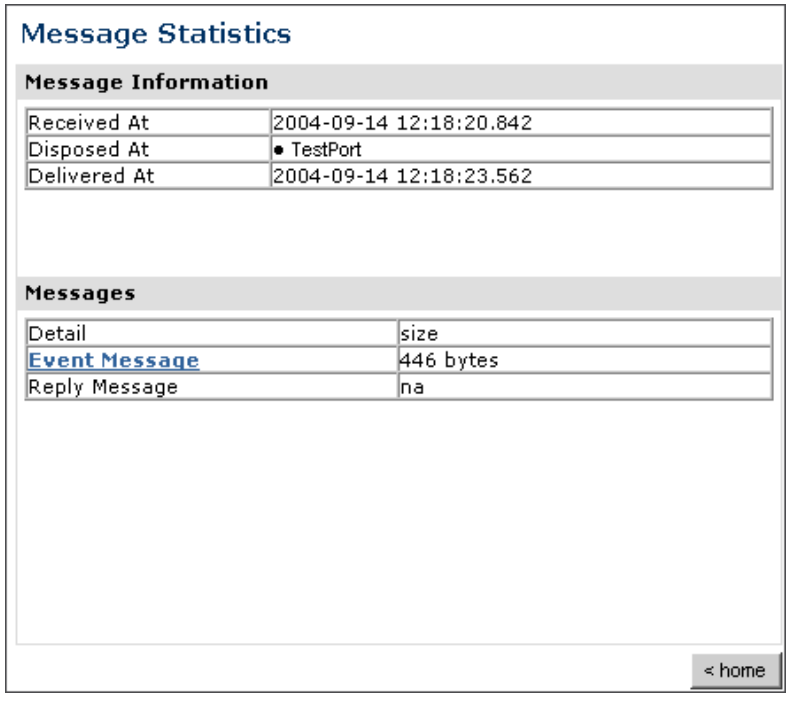
5. From the Ports drop-down list, select a port for the channel.

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

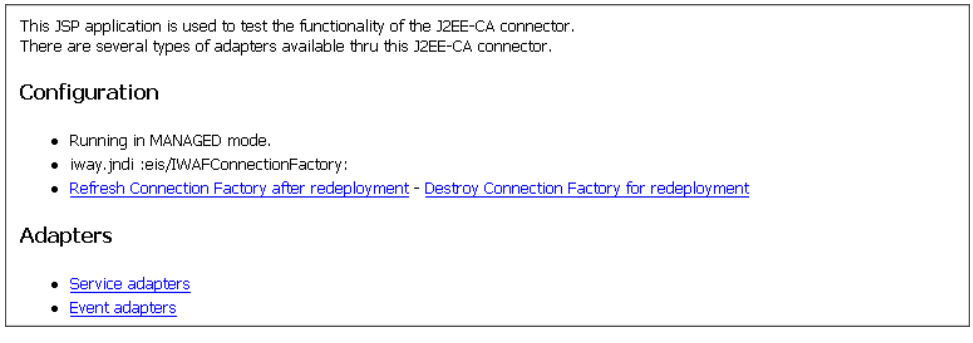
   Is the name of the machine where your application server is running.

   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.

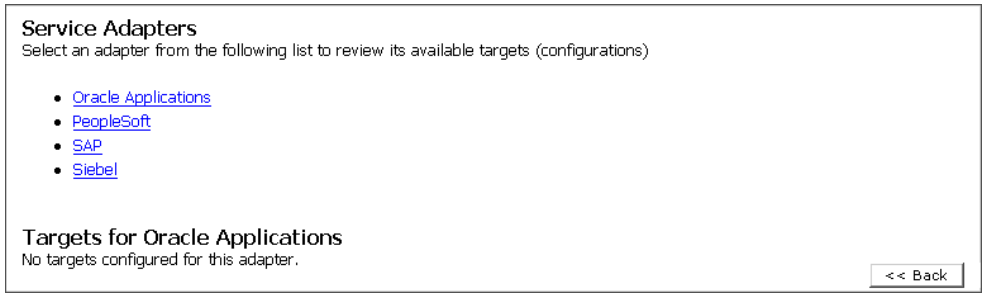
a. Click *Destroy Connection Factory for redeployment*.

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

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

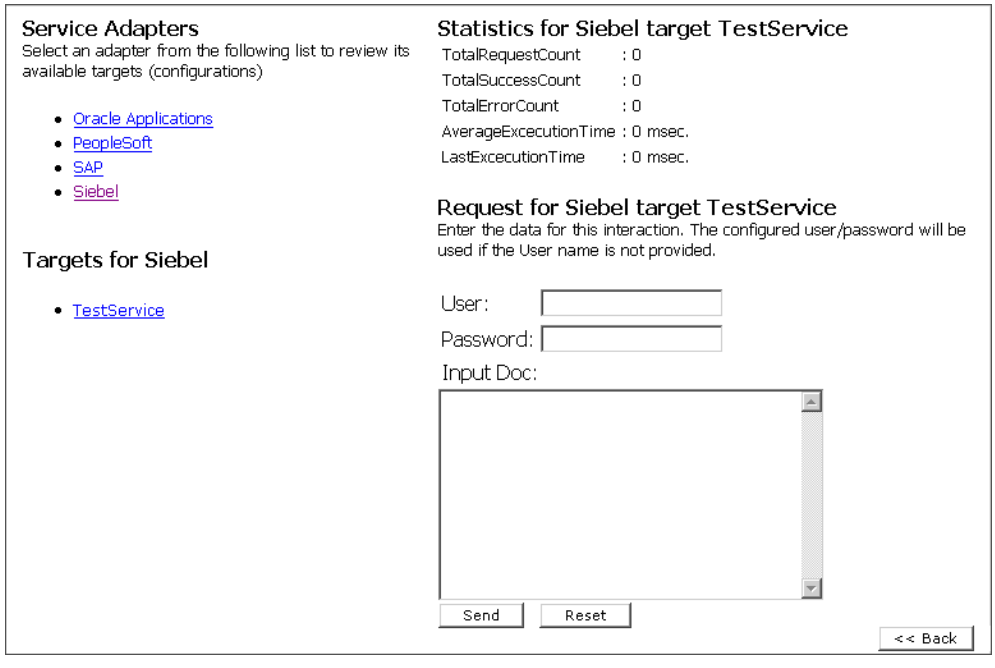
3. Click *Service adapters*.

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.



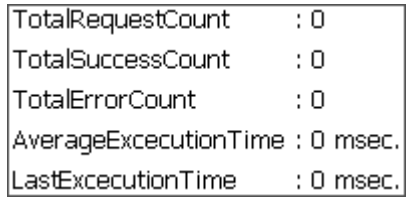
a. Click the desired target for your service adapter.

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

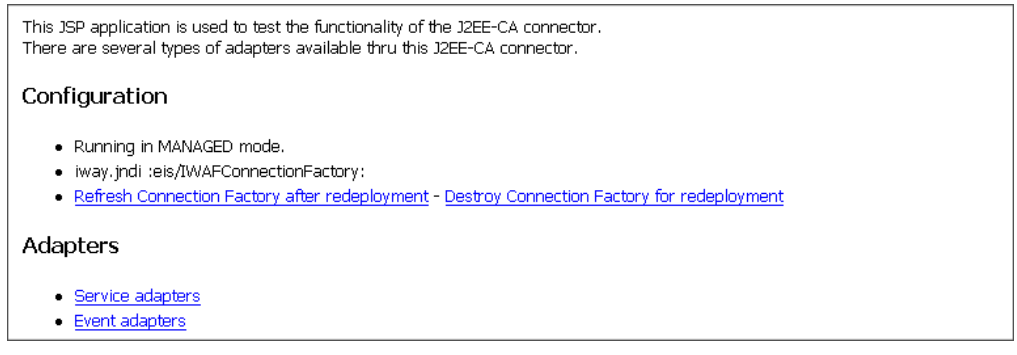
   Is the name of the machine where your application server is running.

   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*.

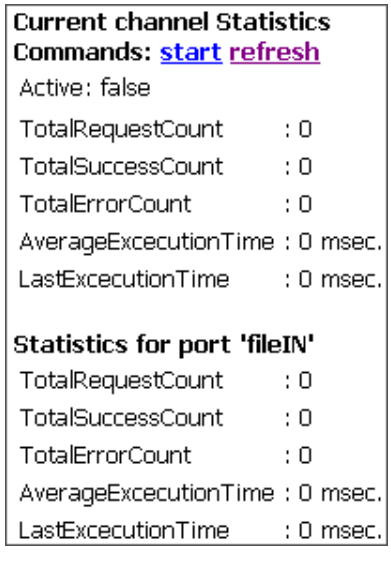
   The Event Adapters page opens.

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 4-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

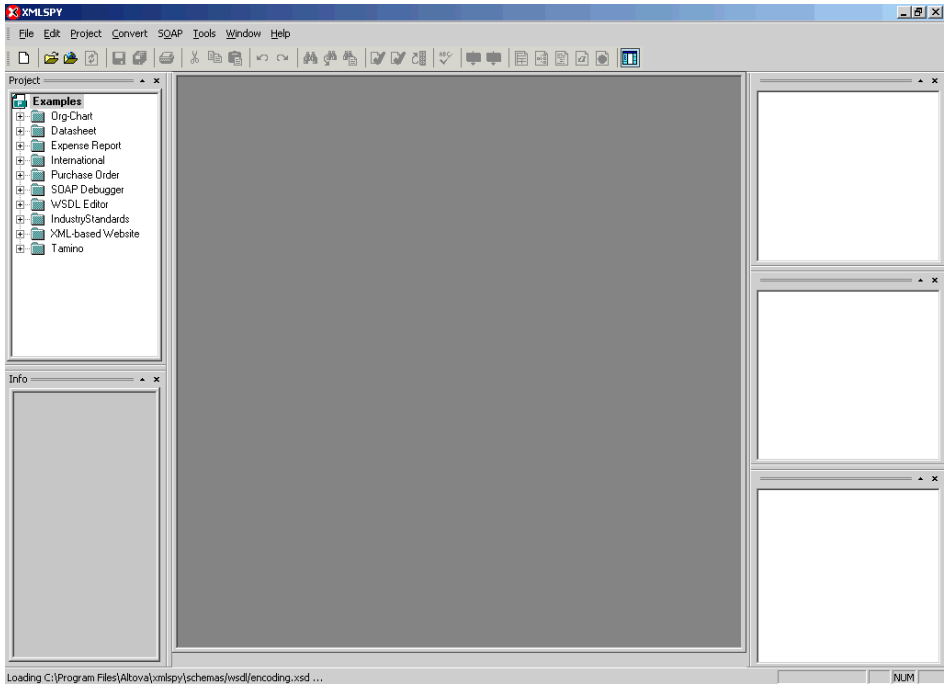
To migrate an iBSE repository that is configured for Oracle:

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

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

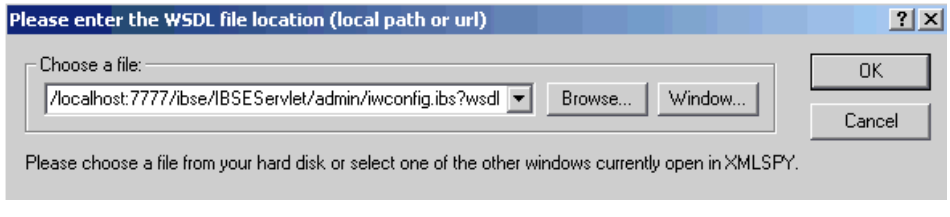
2. Open a third party XML editor, for example, XMLSPY.

   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.



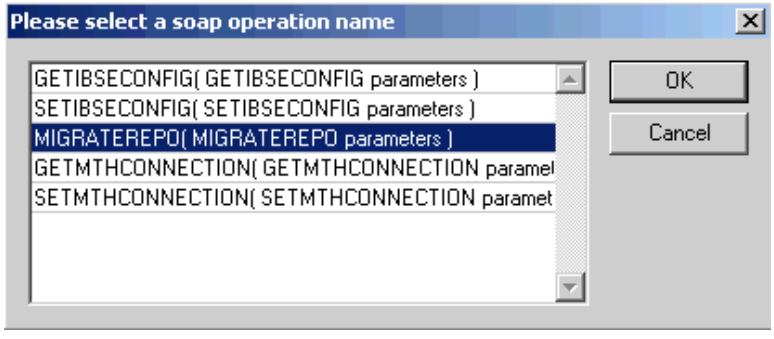
3. From the SOAP menu, select *Create new SOAP request*.

   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.



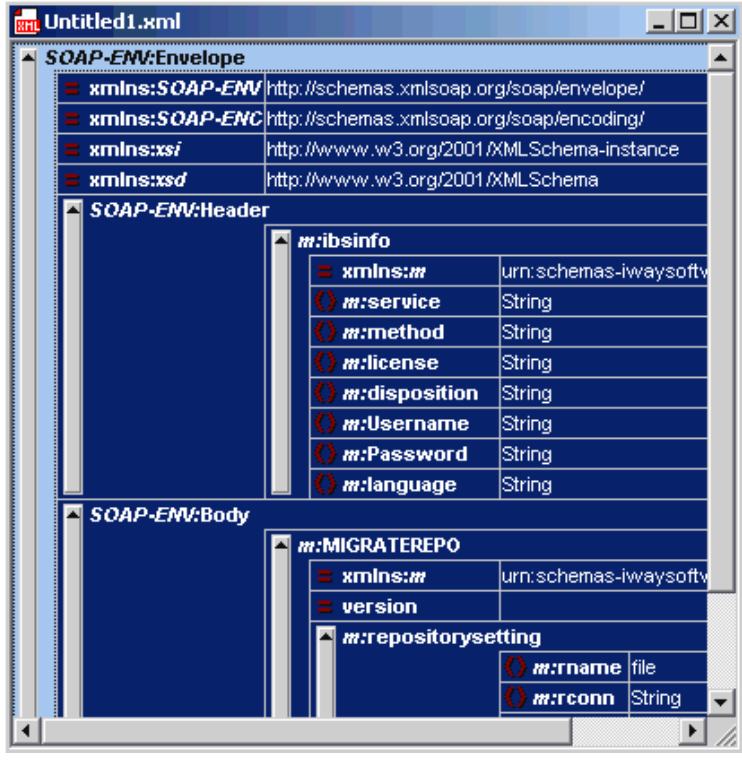
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

   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:

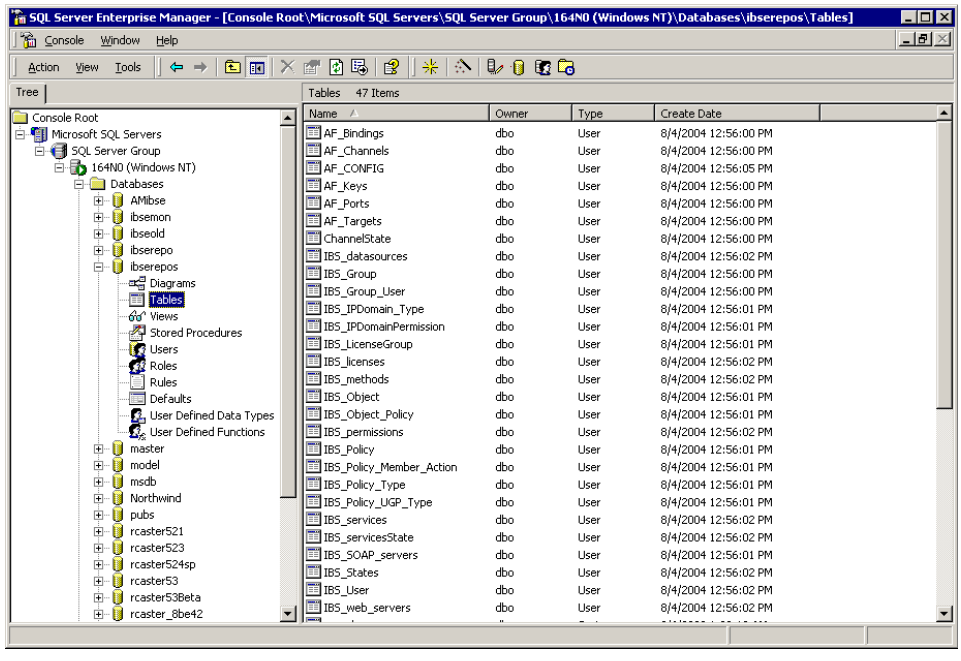
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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:

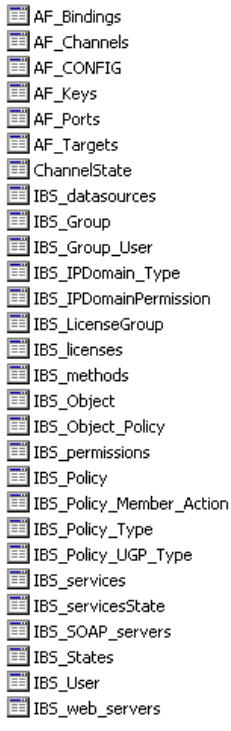
   For Oracle 8:

   ```
   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:

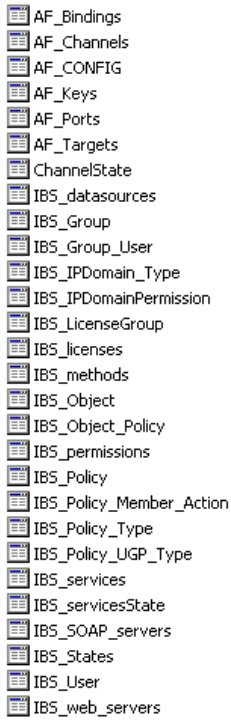
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
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:

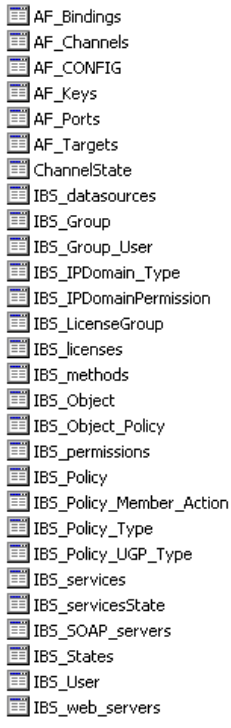
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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

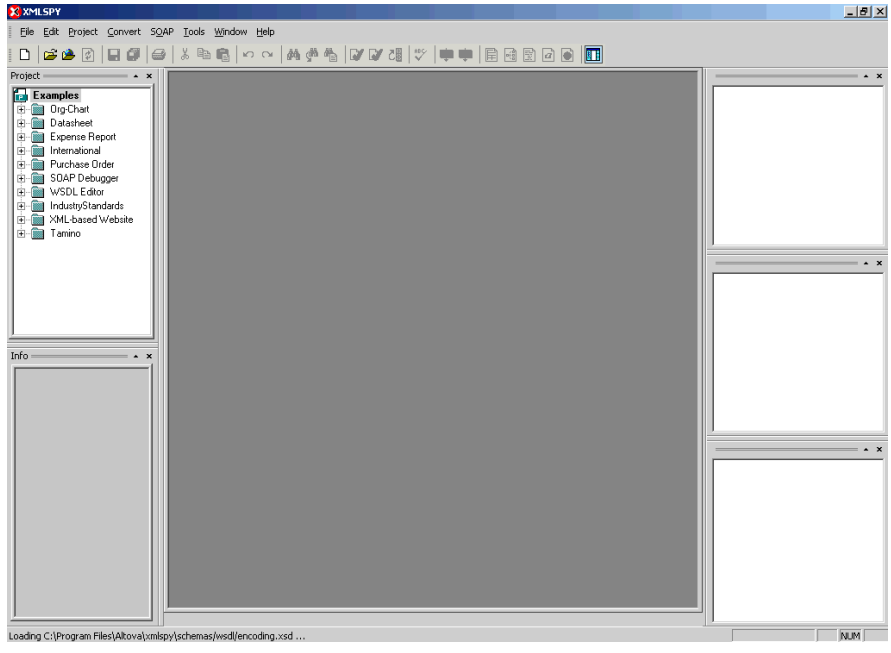
To export a target:

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

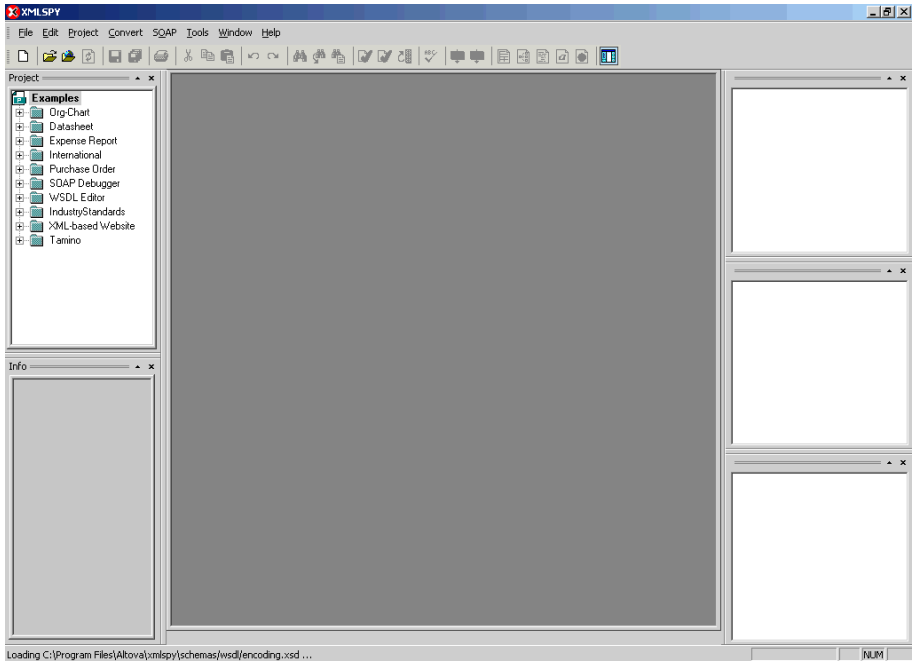
To import a target:

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

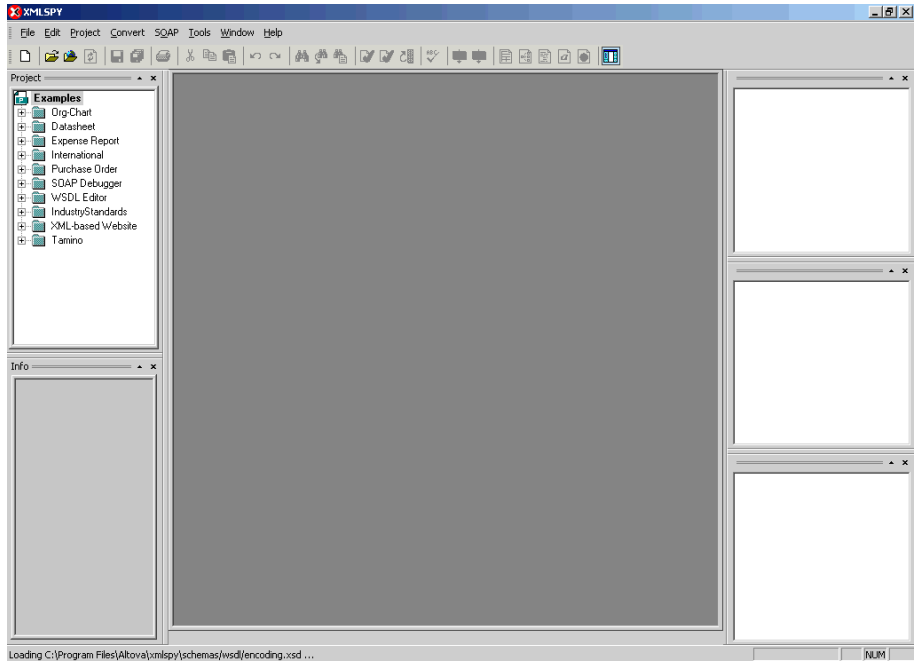
To retrieve Web service method connection information:

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

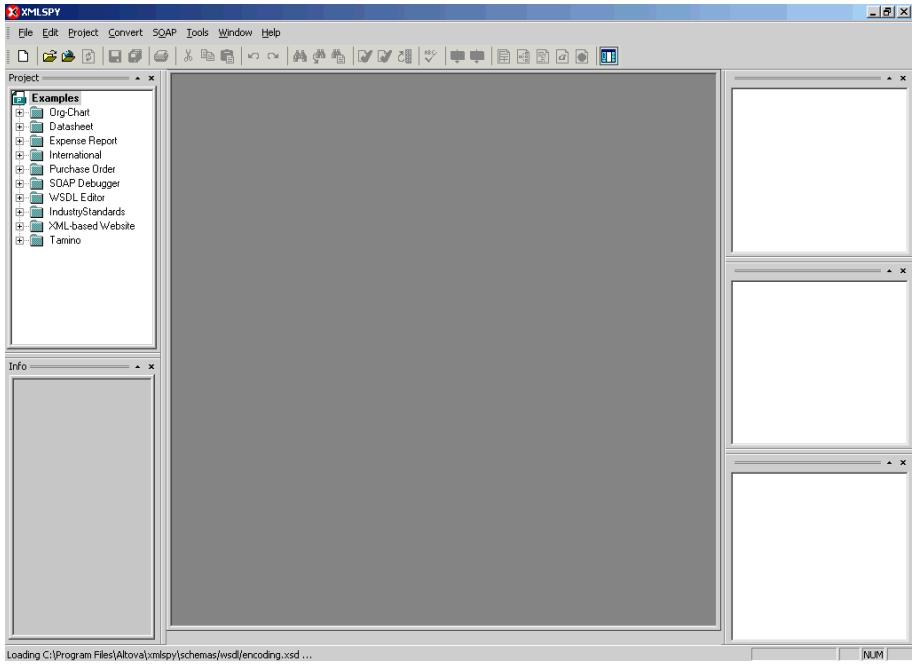
To update Web service method connection information:

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

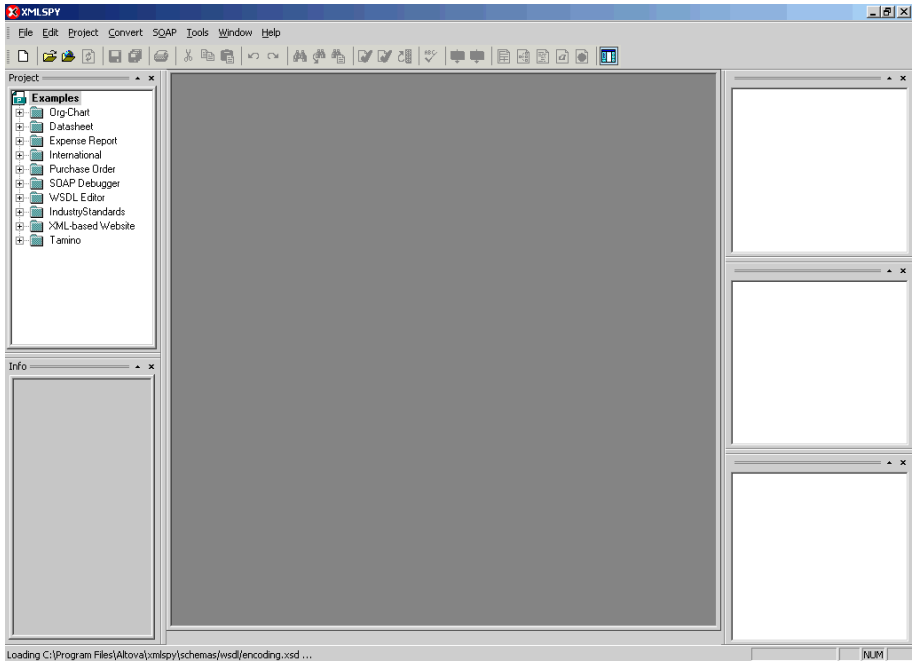
To start a channel programmatically:

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.

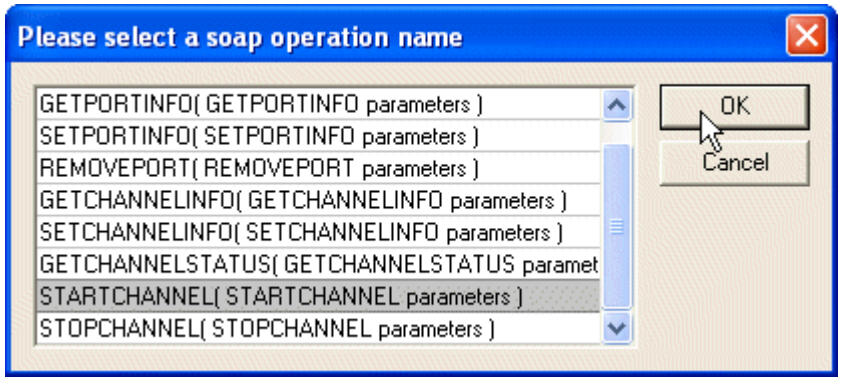


3. From the SOAP menu, select *Create new SOAP request*.

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

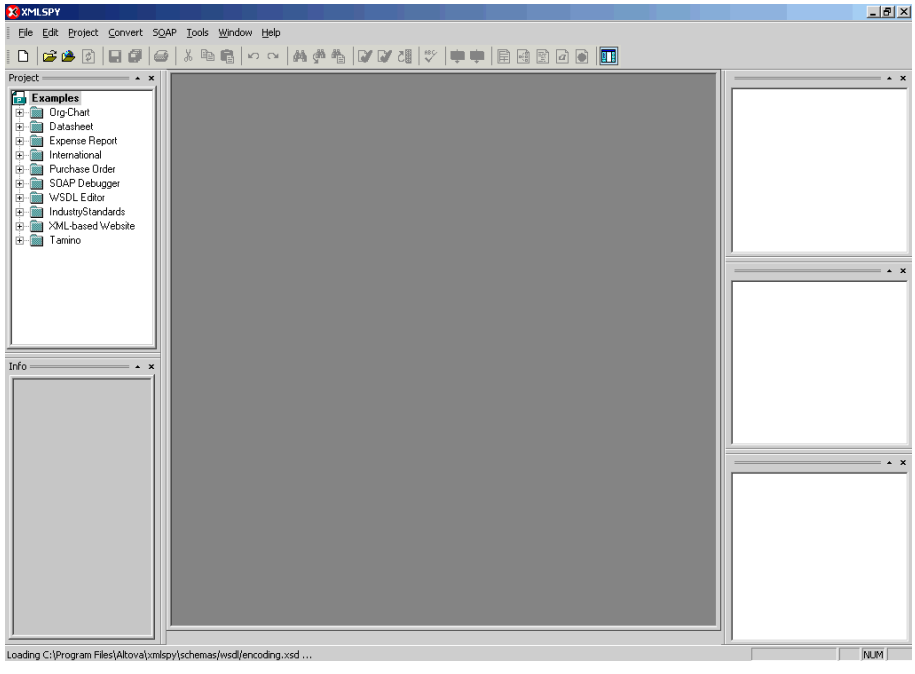
To stop a channel programmatically:

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.

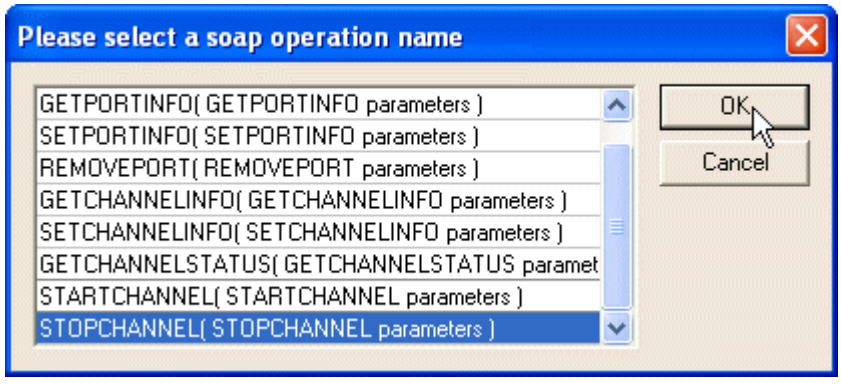


3. From the SOAP menu, select *Create new SOAP request*.

   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:

- Integrating With CORBA
- Starting Application Explorer in WebLogic Workshop
- Creating and Managing a Connection
- Creating an XML Schema
- Understanding iWay Business Services
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an iWay Extensible CCI Control to a BEA WebLogic Workshop Application

This section describes how to use Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas for Corba. Application Explorer deployed in WebLogic Workshop is functionally similar to the Servlet Application Explorer. In addition, this section provides information on creating XML schemas for CORBA integration objects, and creating Web services.

Integrating With CORBA

The iWay Application Explorer supports the creation of schemas based on specific tables and resulting answer sets. To obtain metadata about the Object Request Broker (ORB), Application Explorer connects to the Interface Repository. The iWay Adapter for CORBA for BEA WebLogic extracts the definition of CORBA servers and converts them to XML schemas and service XML request and response definitions. You can see the original definitions of the CORBA servers using Application Explorer.

Application Explorer displays all objects defined by its Interface Definition Language (IDL) that are loaded into the Interface Repository (IFR). After creating a connection in Application Explorer, you can use the Explorer to verify that the system definition was entered correctly.

Using the iWay Adapter for CORBA for BEA WebLogic, Application Explorer populates each object with data retrieved from the Interface Repository. The contents of the Interface Repository appear in the Application Explorer tree. Expanding this interface displays its methods, return arguments, and parameters. Application Explorer displays a list of modules, appearing as folders. These IFR modules represent the different work areas of CORBA.

The XML schema defines the format of XML requests and corresponding responses to the service adapter. The schema is a language-neutral interface description in XML format that declares the types, objects, and methods for the CORBA system. Conceptually, the XML schema is the same as the CORBA IDL.

Note: Before creating schemas, you can save time by verifying that your ORB infrastructure is properly configured, your server is registered in the Naming Service or its object reference is available, and your interface repository (IFR) is running and populated.

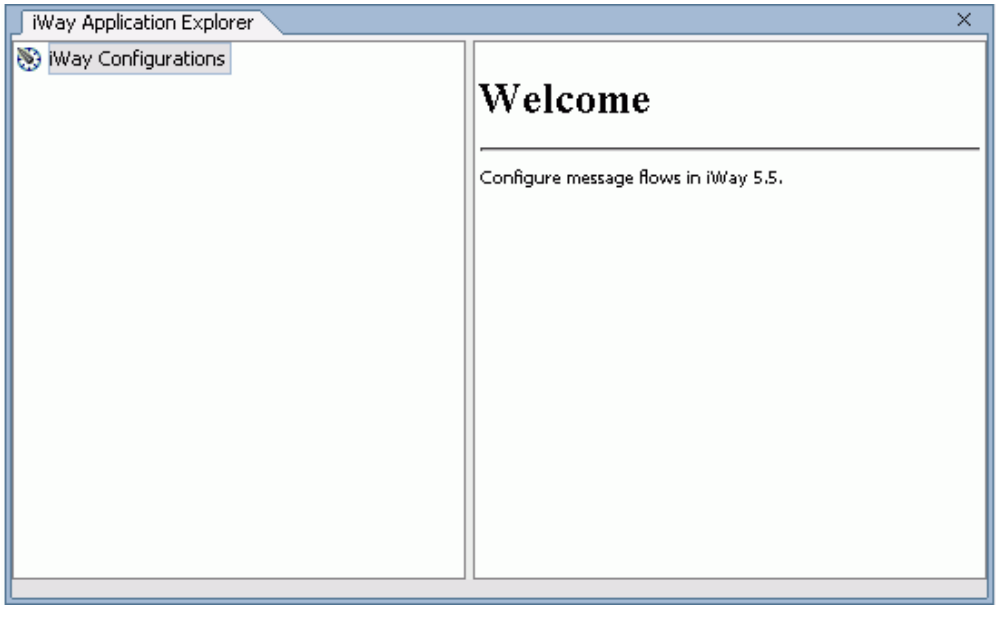
Starting Application Explorer in WebLogic Workshop

You can use iWay Application Explorer with an iBSE or a JCA configuration. Before you can use Application Explorer, you must start BEA WebLogic Server.

Procedure **How to Start Application Explorer in WebLogic Workshop**

1. Start WebLogic Workshop.
2. Ensure that the server on which Application Explorer is deployed is started. If it is not started, select *WebLogic Server* from the Tools menu and then click *Start WebLogic Server*.
3. From the View menu, select *Windows* and then click *iWay Application Explorer*.

   Application Explorer opens as a frame within the Workshop.

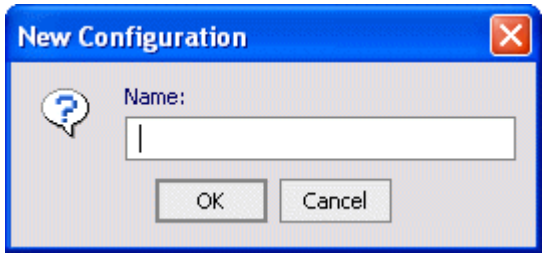


Procedure How to Define a New Configuration

Before you can start using Application Explorer, you must define a new configuration by performing the following steps:

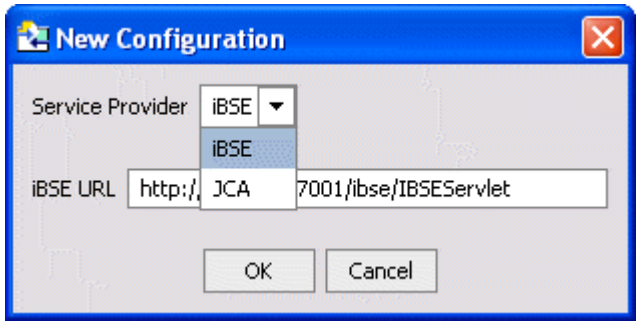
1. Right-click *iWay Configurations* and select *New*.

   The New Configuration dialog box opens:



2. Enter a name for the new configuration (for example, CORBA) and click *OK*.

   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,

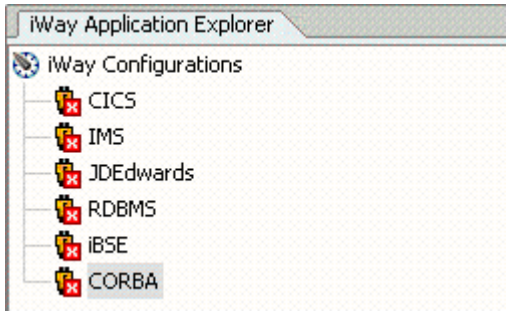
```
C:\Program Files\iWay55
```

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.



Procedure How to Connect to a New Configuration

Right-click the configuration to which you want to connect (for example, CORBA), and select *Connect*.

Nodes are displayed for iWay Service Adapters, iWay Event Adapters, and iWay Business Services (also known as Web services):

You are now ready to define new targets to CORBA.

Creating and Managing a Connection

To access an adapter, you must define a target that connects to the adapter. After the defined target is created, it automatically is saved. You must establish a connection to the defined target every time you start Application Explorer or after disconnecting.

Procedure How to Define a New Target

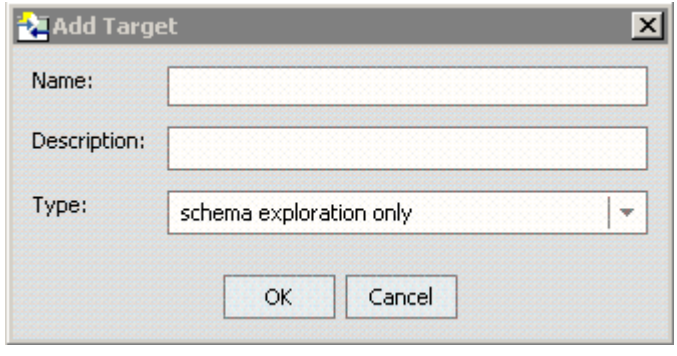
1. Expand the *iWay Service Adapters* node.

   The application systems supported by Application Explorer display as nodes based on the iWay adapters installed.

2. Expand the *CORBA* node.

3. Right-click the *CORBA* node and select *Add Target*.

   The Add Target dialog box opens:

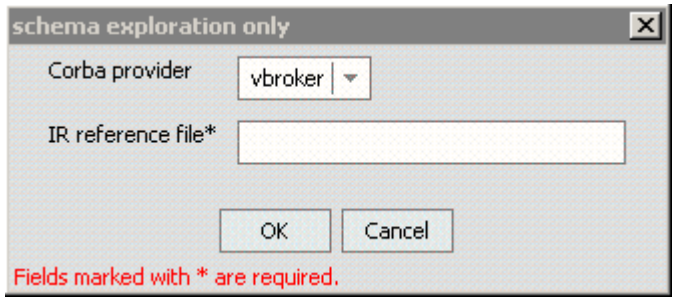


   a. Type a name (for example, CORBA_Connection) and a brief description for the new target.

   b. From the Type drop-down list, select the type of target (for example, schema exploration only). See the screens and table in step 4 to see which option to select.

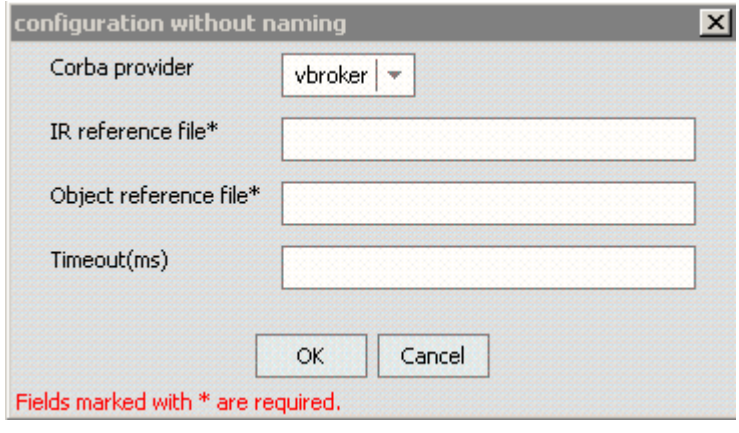
4. Click *OK*. The dialog box that opens depends on the target type you selected.

   - If you selected a target type of schema exploration only, a dialog box opens containing two parameters and two action buttons (OK and Cancel).

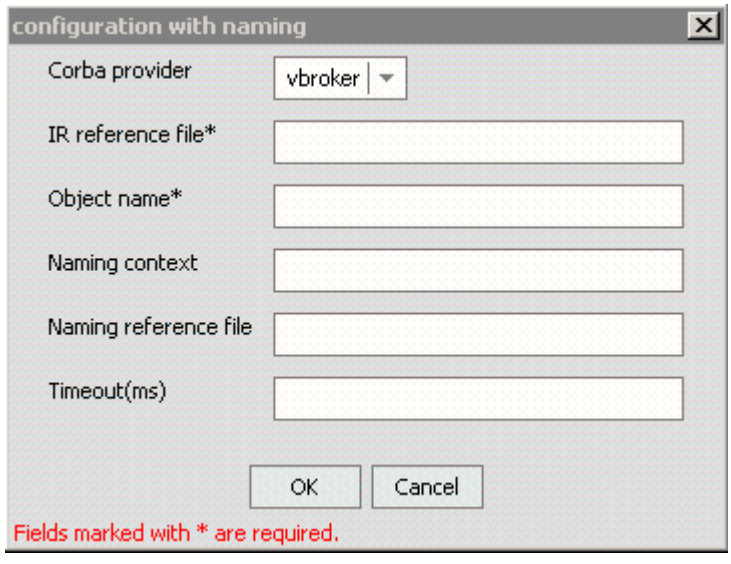


   Note: The CORBA connection parameters are consistent with those found in your CORBA system. For more information on parameter values that are specific to your CORBA configuration, consult your CORBA system administrator.

   - If you selected a target type of configuration without naming, a dialog box opens containing four parameters and two action buttons (OK and Cancel).



- If you selected a target type of configuration with naming, a dialog box opens containing six parameters and two action buttons (OK and Cancel).



a. Type the appropriate information for your target type based on the information in the following table.

Target Parameter	Description
Corba provider	Type of client ORB libraries through which the service will be sent. Possible values are vbroker, orbacus, and jacorb.
IR reference file	Name and path of the Interoperable Object Reference file that specifies the location of the Interface Repository service.
Object reference file	Name and path of the Interoperable Object Reference file that specifies the location of the CORBA object. This parameter enables you to specify the location of an object using a direct IOR reference.
Timeout(ms)	Maximum time, in milliseconds, that a service will wait for a CORBA object to respond before the service terminates. The default value, 0, specifies that the service will wait indefinitely for a response.

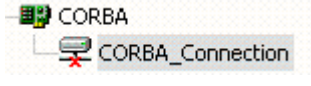
Target Parameter	Description
Object name	Name of the object registered in the Naming Service (for example, bea.clubmed). This parameter, together with Naming context, enables you to specify the location of an object using an indirect Naming Service reference.
Naming context	ORB's name context. This parameter, together with Object name, enable you to specify the location of an object using an indirect Naming Service reference.
Naming reference file	Name and path of the Interoperable Object Reference file that specifies the location of the Naming Service. This parameter, together with Naming context and Object name, enables you to specify the location of an object using an indirect Naming Service reference.

5. Click *OK*.

   After the extraction finishes, the new target, CORBA_Connection, appears under the CORBA node.

Procedure How to Connect to a Defined Target

1. Expand the *iWay Service Adapters* node.
2. Expand the *CORBA* node.
3. Click the target name (for example, CORBA_Connection) under the CORBA node.

   The Connection dialog box opens, populated with values you entered for the connection parameters.
4. Verify your connection parameters.
5. Right-click the target name and select *Connect*.

The x icon disappears, indicating that the node is connected.

Disconnecting From a Defined Target

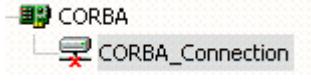
Although you can maintain multiple open connections, iWay Software recommends disconnecting from targets that are not in use.

Procedure How to Disconnect From a Defined Target

1. Expand the *iWay Service Adapters* node.
2. Expand the *CORBA* node.
3. Right-click the target to which you are connected (for example, CORBA_Connection), and select *Disconnect*.

   Disconnecting from CORBA_Connection drops the connection with CORBA, but the node remains.

   The x icon appears, indicating that the node is disconnected.



Editing a Defined Target

After you create a defined target using Application Explorer, you can edit any information that you provided during the creation process.

Procedure How to Edit a Defined Target

1. Expand the *iWay Service Adapters* node.
2. Expand the *CORBA* node.
3. Right-click the target to which you are connected (for example, CORBA_Connection), and select *Edit*.

   The Edit dialog box opens containing the connection parameters.
4. Edit the information as needed and then click *OK*.

Deleting a Defined Target

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

Procedure **How to Delete a Defined Target**

1. Expand the *iWay Service Adapters* node.
2. Expand the *CORBA* node.
3. Right-click the target to which you are connected (for example, CORBA_Connection), and select *Delete*.

   The node disappears from the list of available connections.

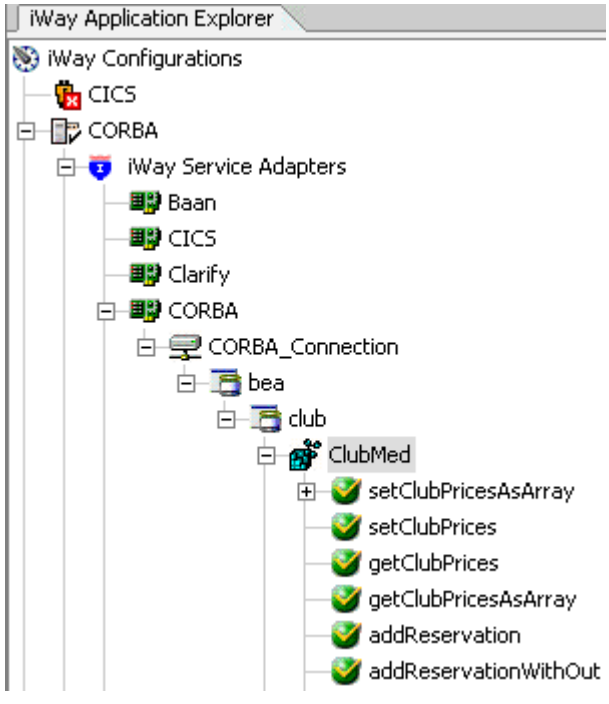
Creating an XML Schema

Each service the iWay Adapter for CORBA for BEA WebLogic uses must be defined by schemas. Application Explorer generates XML schemas for service requests and service responses.

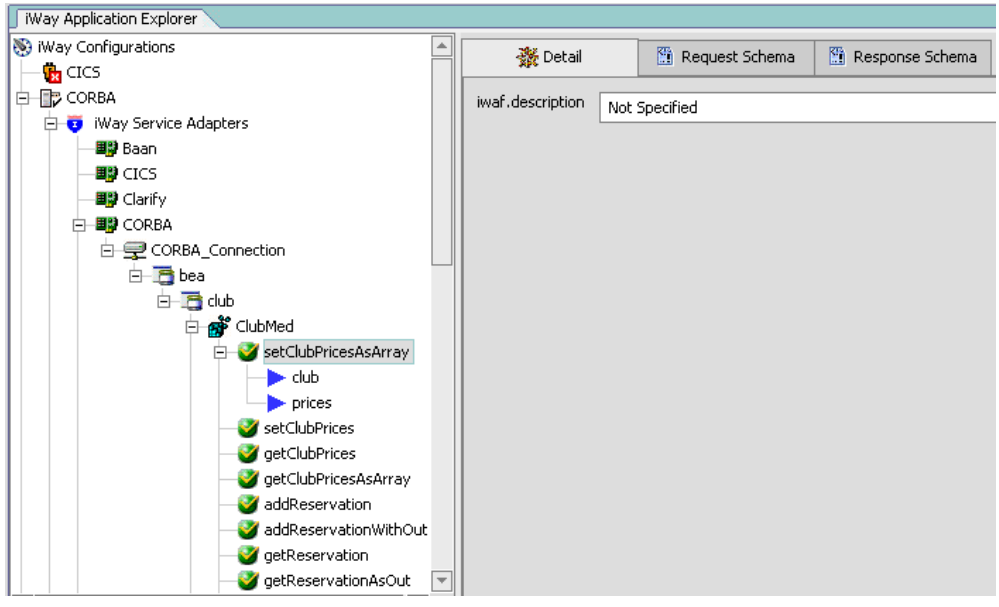
- **Service requests** are requests for action that your application makes to your CORBA system. As part of the definition, the request schema defines the input parameters required by the CORBA system.
- **Service responses** are the way the CORBA system responds to the service request. A service response schema defines this service response. Service requests always have a corresponding service response.

Procedure How to Create a Request Schema and a Response Schema

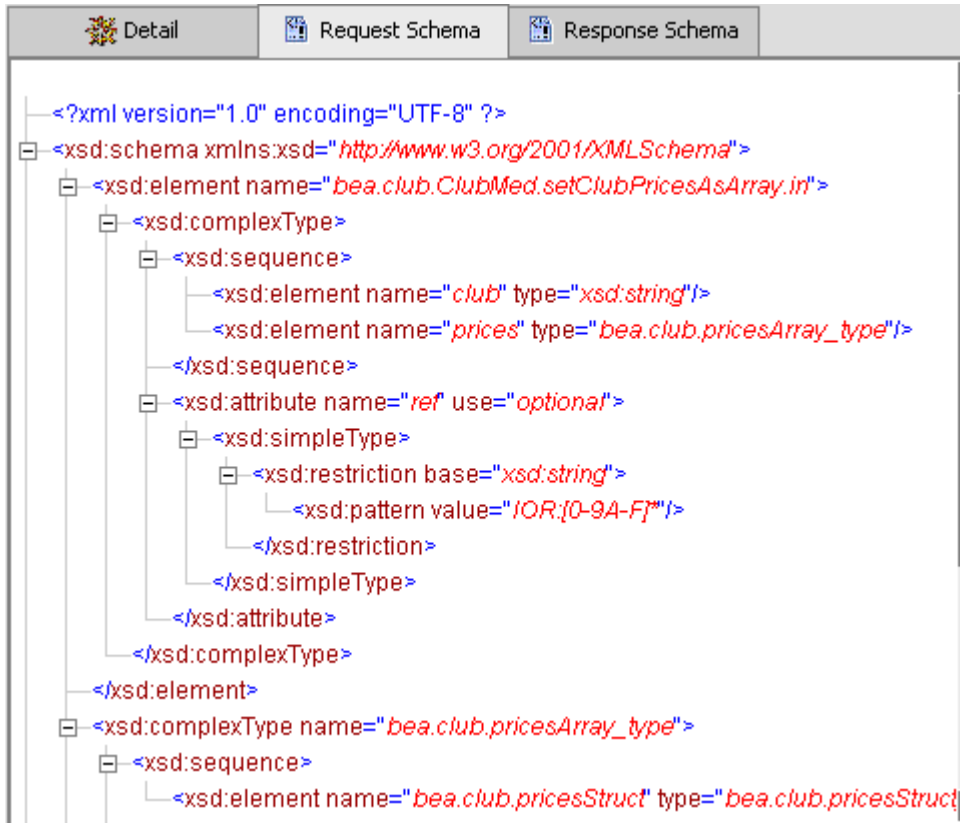
1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page A-9.
2. Expand the tree under the integration objects to see the items for which you may create schemas.



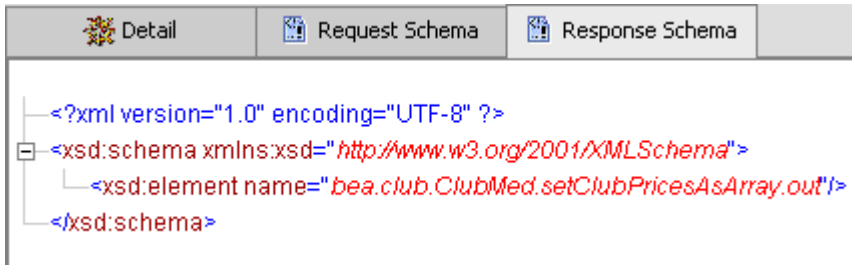
3. Expand and then select the node for which you want to create the schemas. The schemas are automatically created when you select the node. A screen appears in the right pane containing Detail, Request Schema, and Response Schema tabs.



Click the *Request Schema* tab to view the request schema information.



Click the *Response Schema* tab to view the response schema information.



Reference Schema Location

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

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

  ```
  C:\Program Files\iway55\bea\ibse\wsdl\schemas\service\CORBA\
  CORBA_Connection
  ```

  where:

  CORBA_Connection

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

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

  ```
  C:\Program Files\iWay55\config\base\schemas\CORBA\CORBA_Connection
  ```

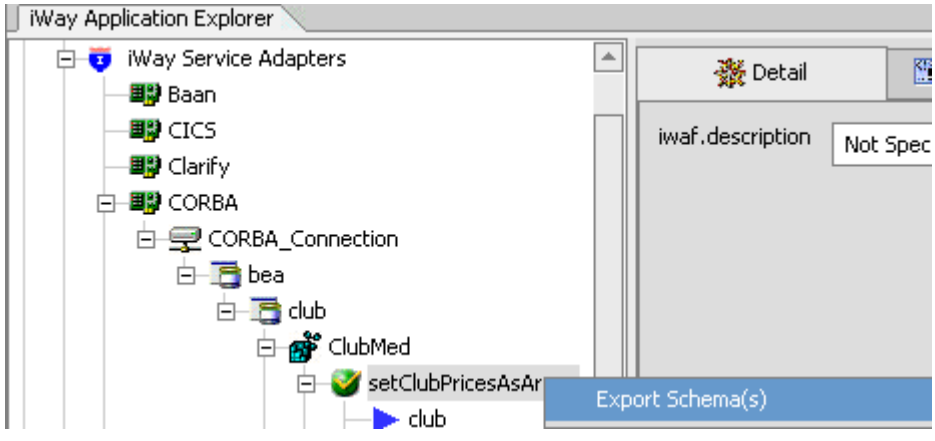
  where:

  CORBA_Connection

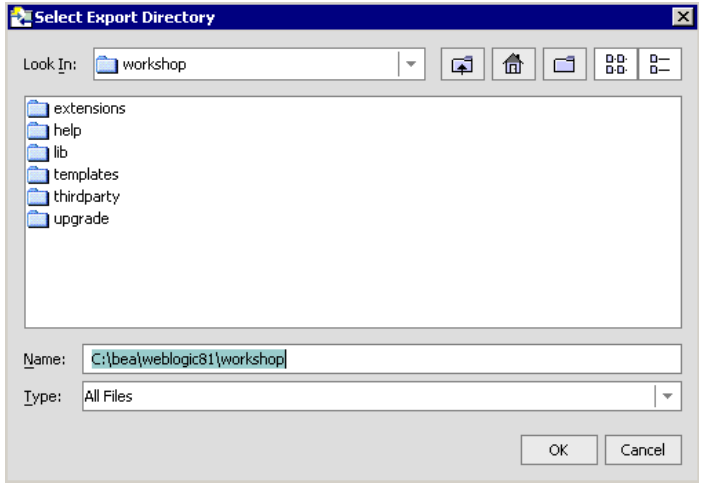
  Is the name of the connection to the CORBA system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

Procedure How to Export a Schema

1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page A-9.
2. Expand the tree under the integration objects to see the items for which you may create a schema:
3. Expand and then right-click the node beneath the integration object for which you want to export a schema, and then select *Export Schema*.



4. The Select Export Directory dialog box opens:



5. Select the directory to which you want to save the schema and click *OK*.

Understanding iWay Business Services

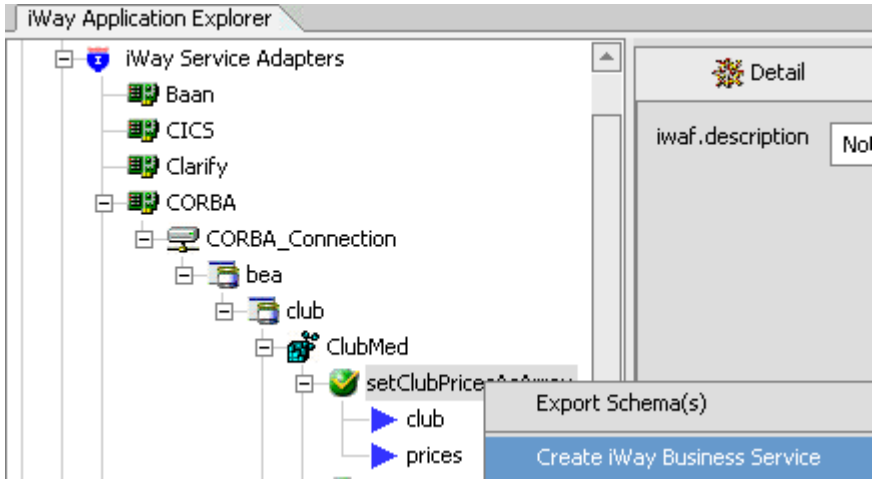
Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the adapter. The Integration Business Services Engine exposes functionality as Web services. It serves as a gateway to heterogeneous back-end applications and databases.

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. A Web service integrates within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

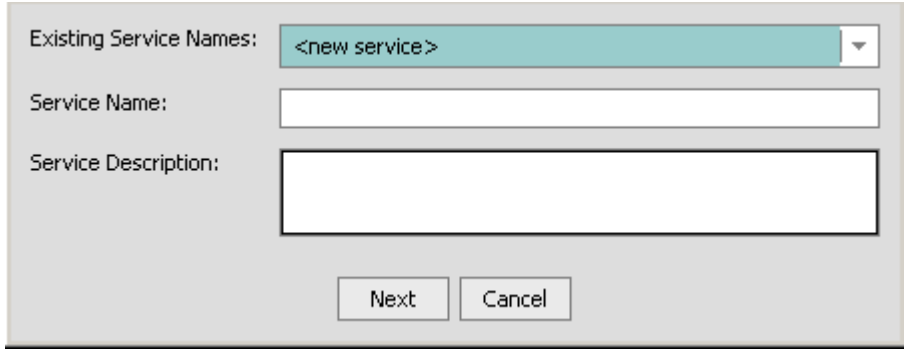
Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation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

Procedure How to Create a Business Service

1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page A-9.
2. Expand the tree under the integration objects to see the items for which you may create a schema:
3. Expand and then right-click the node beneath the integration object for which you want to export a schema, and then select *Create iWay Business Service*.



4. The service information dialog box opens:



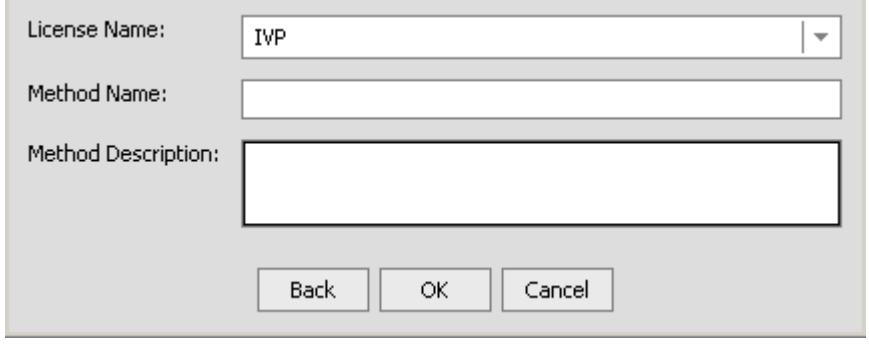
a. Select either a new service or an existing service from the Existing Service Names drop-down list.

b. Type a service name if you are creating a new service. This name identifies the Web service in the list of services under the iWay Business Services node.

c. Type a description for the service.

5. Click *Next*.

The license and method dialog box opens:



a. In the License field, select one or more license codes to assign to the Web Service. To select more than one, hold down the *Ctrl* key and click the licenses.

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

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

6. Click *OK*.

Application Explorer expands the iWay Business Services node in the left pane to show the newly created business service and presents a test input area in the right pane.

Testing a Business Service

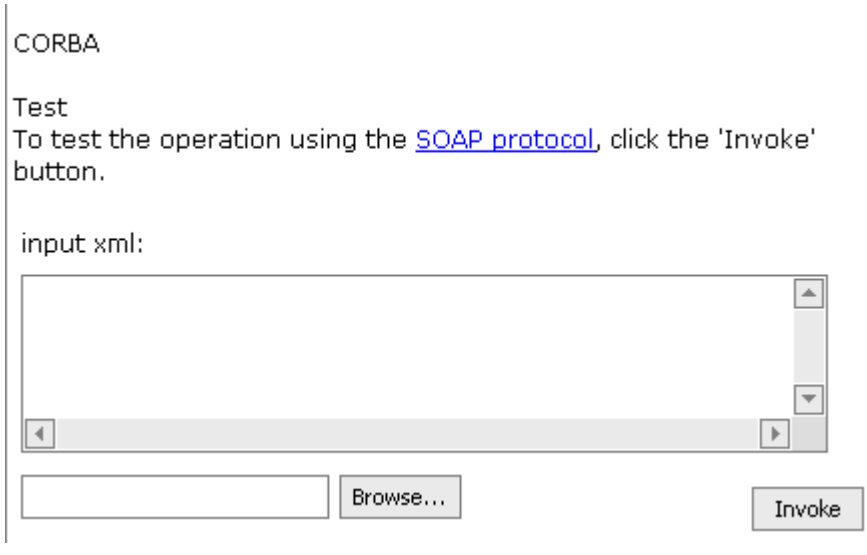
After a business service is created, use the test tool to ensure that it functions properly.

Procedure How to Test the Business Service

1. If you are not in the iWay Business Services node of Application Explorer, click the node to access business services.
2. If it is not expanded, expand the list of business services under iWay Business Services.
3. Expand the *Services* node.
4. Select the name of the business service you want to test (for example, CORBA).

   The business service name appears as a link in the right pane.
5. In the right pane, click the named business services link.

   The test option appears in the right pane.



6. In the input xml field, either type a sample XML document that queries the service, or browse to the location of an XML instance and click *Open*.
7. Click *Invoke*.

   Application Explorer displays the results in the right pane.

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 BEA WebLogic Server.

Procedure **How to Generate WSDL From a Web Service**

1. Expand the *iWay Business Services* node.
2. Expand the *Services* node to display the Web service for which you want to generate WSDL.
3. Right-click the Web service and select *Export WSDL*.

   The Save dialog box opens.
4. Choose a location for the file and specify .wsdl for the file extension.

   Note: The file extension must be .wsdl.
5. Click *Save*.

Adding a Control for an iWay Resource in BEA WebLogic Workshop

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

Adding a Web Service Control to a BEA WebLogic Workshop Application

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

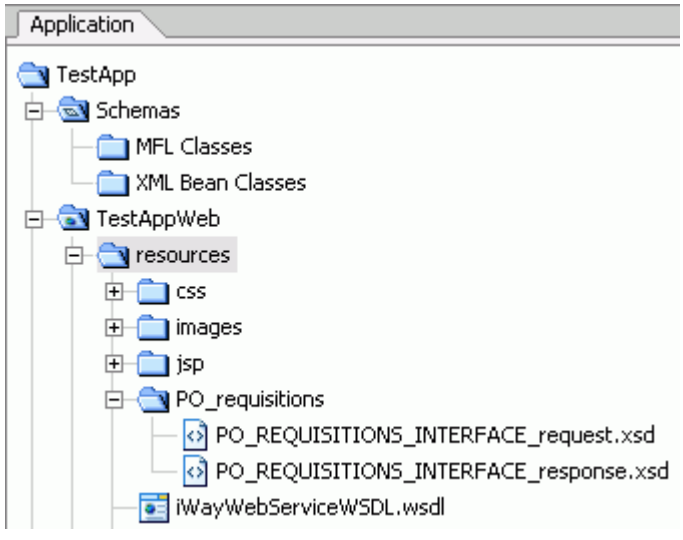
For more information on exporting a WSDL file, see *How to Generate WSDL From a Web Service* on page A-20.

Procedure How to Add a Web Service Control

To add a Web service control:

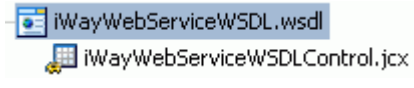
1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

   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 iWay Extensible CCI Control to a BEA WebLogic Workshop Application

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.

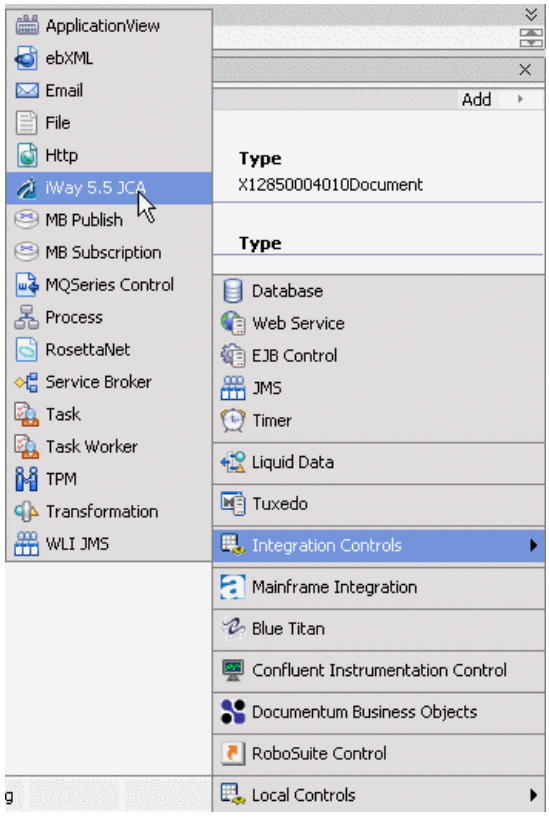
In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control*.

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-29.

Example Defining a Control Using the Extensible CCI Control

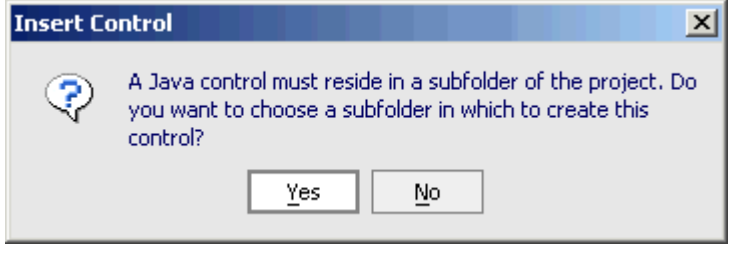
The following sample JCX demonstrates how to define a control for Corba using the extensible CCI control in BEA WebLogic Workshop.

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



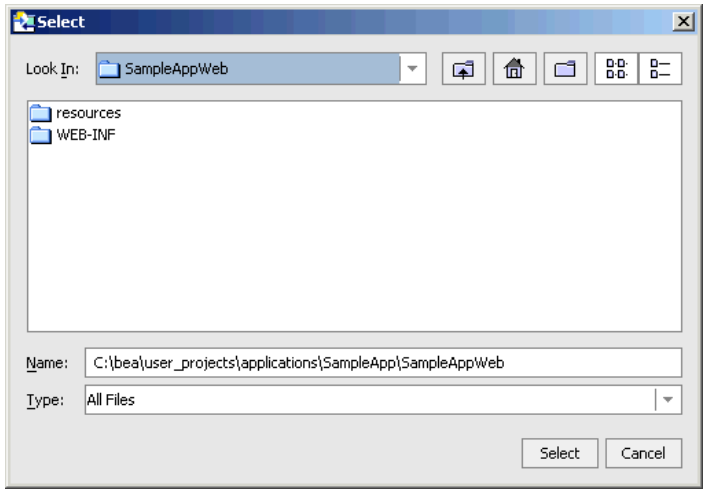
2. Click *Add* from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

   The Insert Control message box opens.



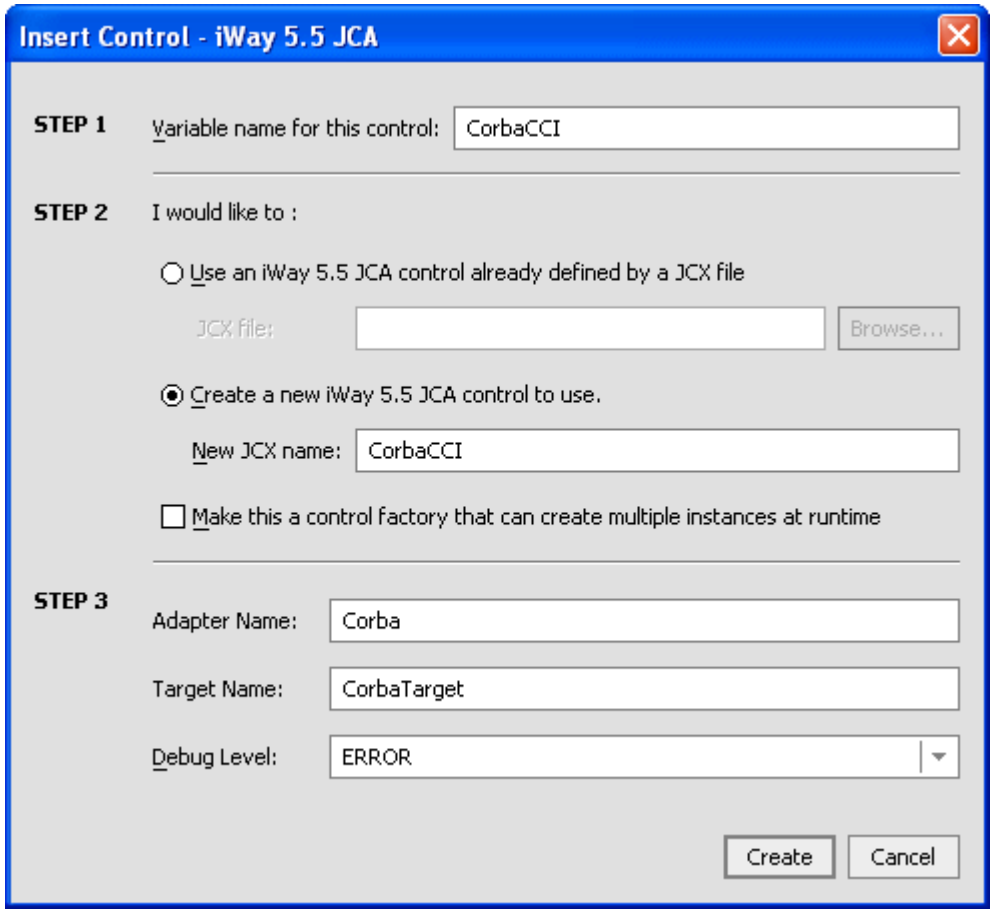
3. Click *Yes*.

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.



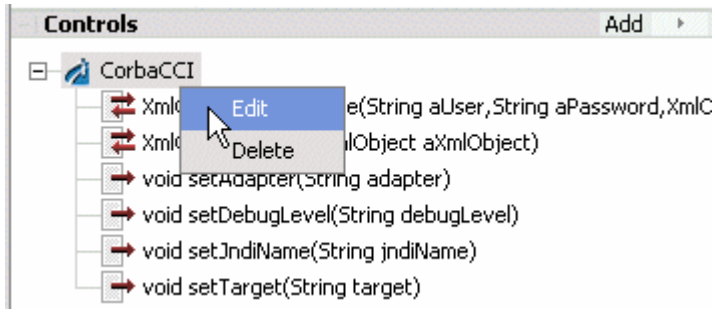
a. Provide a variable name for the control.

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.

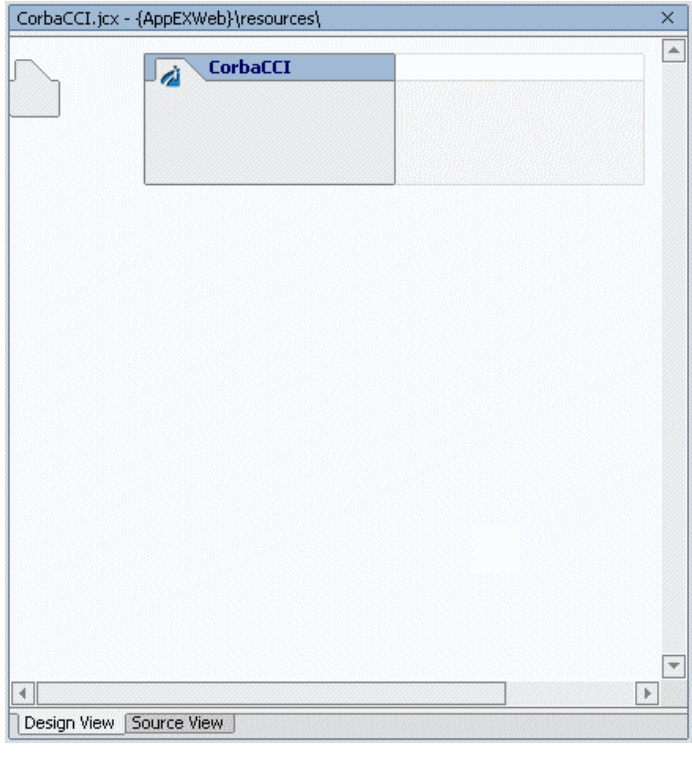
5. Click *Create*.

A new JCX file is created.



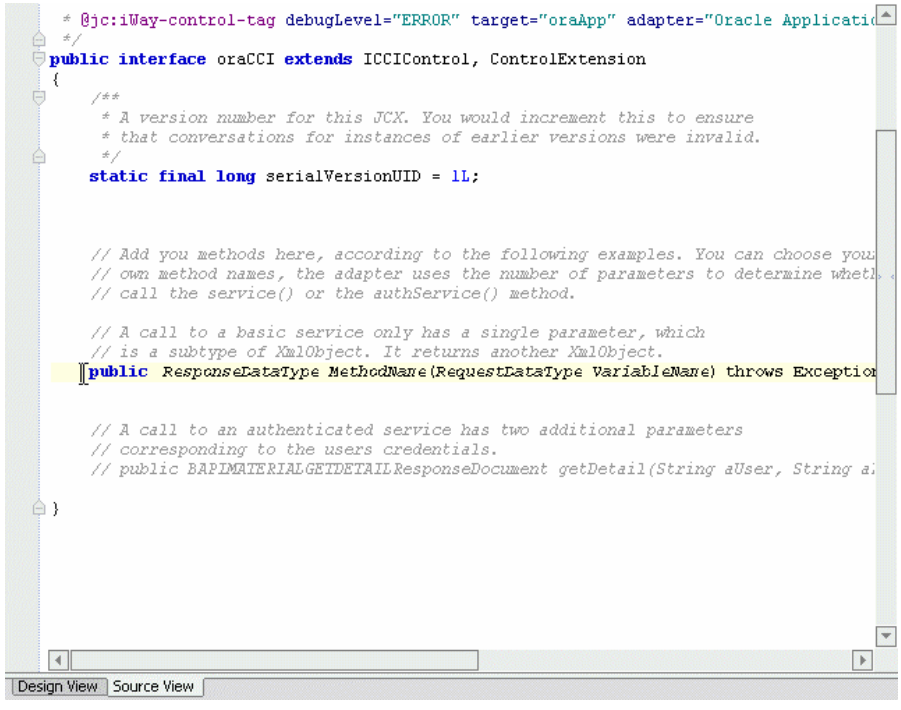
6. Right-click the control, for example, CorbaCCI, and select *Edit*.

The Design View for the control opens.



7. Click the *Source View* tab.

The Source View for the control opens.



Perform the following steps:

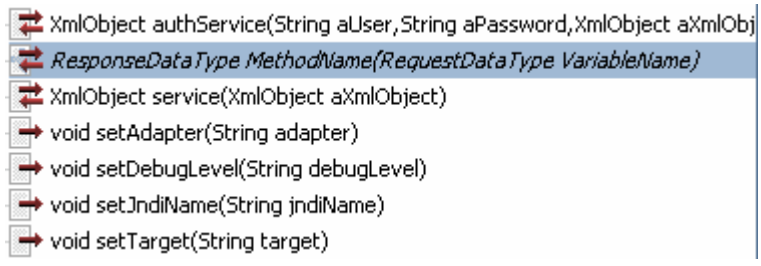
a. Uncomment the public class definition.

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

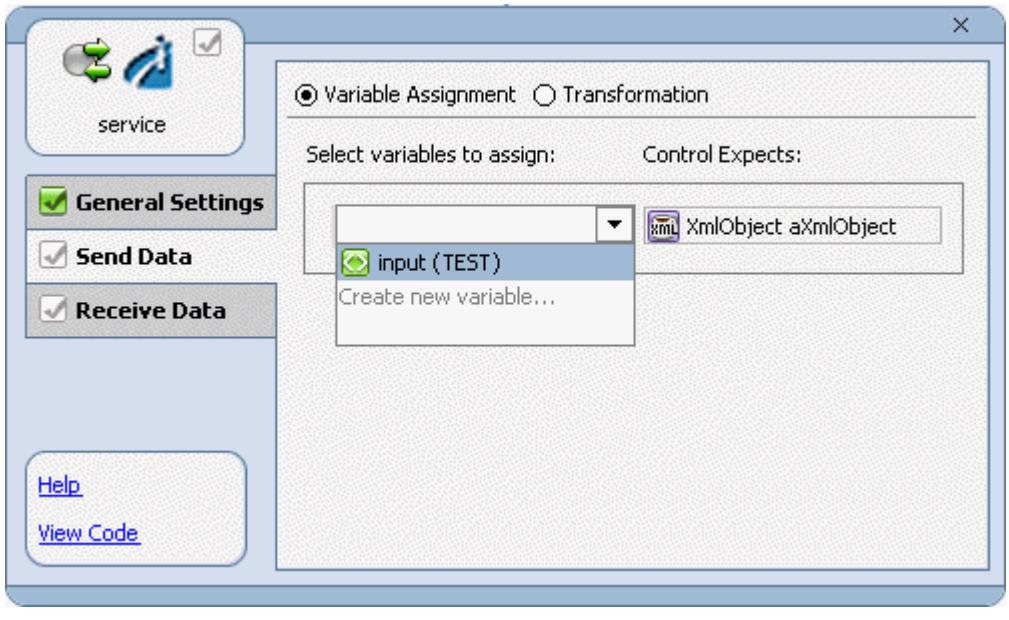
Is the XML Bean Class value that is generated from the request schema.

VariableName

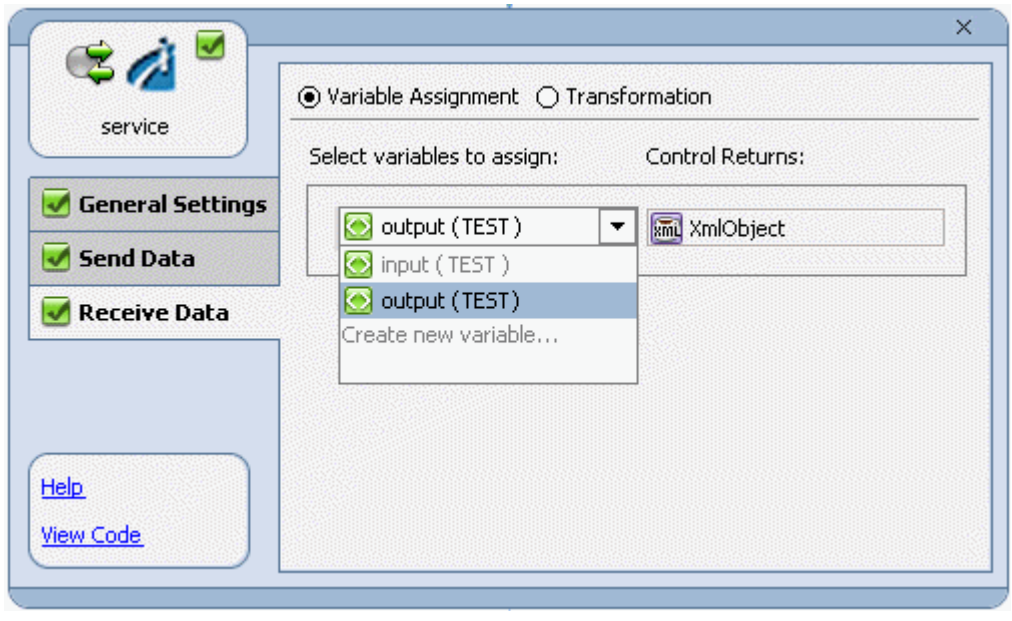
Is the request variable that stores the request document, which is used as input by the extensible CCI control.

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 CORBA Implementations With the Adapter

Topics:

- Using JacORB With the Adapter
- Using VisiBroker for Java With the Adapter
- Using Orbacus With the Adapter

This section provides details about using the adapter with JacORB, VisiBroker for Java, and Orbacus.

Note: The adapter supports Orbix using Orbacus libraries and IR. Orbacus 4.1.3 is at the same CORBA level as the Orbix product. Both products come from the same vendor.

Using JacORB With the Adapter

The iWay Adapter for CORBA for BEA WebLogic includes a sample ORB called JacORB. JacORB is an open source Java implementation of the Object Management Group CORBA specification. It is supplied with your software to enable you to test the adapter.

JacORB is designed to comply with CORBA 2.3 Java language mapping, and supports commonly used CORBA services. It runs on all platforms that implement the Java Virtual Machine (JVM). JacORB is made available under the terms of the GNU Library General Public License (LGPL). Commercial Support support is provided by Object Computing Inc., a Sun Authorized Java Center and member of the OMG; for more information, go to

http://www.ociweb.com

JacORB operates with any CORBA-compliant ORB over IIOP. In practice, JacORB has been used successfully with at least the following ORBs: MICO, TAO, Orbacus, Iona Orbix, Borland VisiBroker, ORBit, omniORB, Vitria C++, and Java. ORB interoperability is made simple by using a foreign name service IOR in the file where the iWay Adapter for CORBA for BEA WebLogic looks up the name server OAR. This can be configured in the jacorb.properties file.

The JacORB Name Service

Name servers are used to locate objects using a human–readable reference (a name) rather than a machine or network address. If objects providing a certain service are looked up using the service name, clients are separated from the actual locations of the objects that provide the service. The binding from name to service can be changed without the client's knowledge.

JacORB provides an implementation of the OMG Interoperable Naming Service (INS), which supports the binding of names to object references (and looking up object references using these names). It also enables clients to easily convert names to strings and vice versa. The JacORB name service comprises two components: the name server program and a set of interfaces and classes used to access the service.

The JacORB Interface Repository

Run-time type information in CORBA is managed by the ORB Interface Repository (IFR) component. It allows applications to request, inspect, and modify IDL type information dynamically. For example, the IR enables applications to determine the operations an object supports. Some ORBs also may require the IR to determine whether a given object type is a subtype of another. However, most ORBs can operate without the IR by encoding this kind of type information in the helper classes generated by the IDL compiler.

In essence, the IR is just another remotely accessible CORBA object that offers operations to retrieve (and theoretically, modify) type information. The IR manages type information in a hierarchical containment structure that corresponds to constructs within IDL specifications: modules contain definitions of interfaces, structures, constants, and so on. Interfaces in turn contain definitions of exceptions, operations, attributes, and constants.

Building and Running the JacORB Request Broker

To build and run the JacORB Request Broker, you need ANT 1.4.1, a Java-based build tool.

Procedure: How to How to Build and Run the JacORB Request Broker

To build and run the JacORB Request Broker:

1. Unzip *JacORB1_4_beta4-full.zip*.

   You can locate this file in BEA_CORBA_SAMPLES.zip in the adapter installation directory. It creates the directory JacORB1_4_beta4 on the selected drive. For example, if you unzip to drive D, the result is D:\JacORB1_4_beta4.

2. Unzip *beacorba.zip* in the directory JacORB1_4_beta4 that you just created.

   This file is found in BEA_CORBA_SAMPLES.zip in the adapter installation directory.

3. Copy *JacORB1_4_beta4\jacorb_properties.template* and rename the copy *JacORB1_4_beta4\jacorb.properties*.

4. Edit the jacorb.properties file.

   a. In the Initial References Configuration section of the file, uncomment the following line:

      ```
      #ORBInitRef.NameService=file:/d:/JacORB1_4_beta4/bea/ns_ref.txt
      ```

   b. In the same line, replace *d:* with the drive and path into which you unzipped your JacORB files.

   c. If the next line—*#ORBInitRef.NameService=file...*—is uncommented, comment it out.

5. Copy *JacORB1_4_beta4\bea\jaco.bat* into the JacORB\bin directory.

6. Edit *JacORB1_4_beta4\bea\setenv-sample.bat* to specify directories in the following statements:

```
set JAVA_HOME=jdk_directory
set JACORB_HOME=JacORB1_4_beta4_directory
set ANT_HOME=ant_tool_directory
```

where:

jdk_directory

Is the directory where your JDK resides.

JacORB1_4_beta4_directory

Is the directory where JacORB resides.

ant_tool_directory

Is the directory where your ANT tool resides.

For example:

```
set JAVA_HOME=c:\jdk1.3
set JACORB_HOME=d:\JacORB1_4_beta4
set ANT_HOME=c:\jakarta-ant-1.4
```

7. To build the JacORB application, in a new DOS command window, execute the following:

```
drive&path:JacORB1_4_beta4\bea\setenv-sample.bat
drive&path:JacORB1_4_beta4\bea\club\ant
```

where:

drive&path

Is the drive and path into which you unzipped your JacORB files.

8. To start the Interface Repository service, in a new DOS command window, execute the following:

```
JacORB1_4_beta4\bea\setenv-sample.bat
ir repository_class_path IOR_filename
```

where:

repository_class_path

Is the path to your repository class files.

IOR_filename

Is the name of the Interface Object Repository file, for example,

```
JacORB1_4_beta4\bea\ir ..\classes ir_ref.txt
```

9. To start the name service, in a new DOS command window, execute the following:

```
drive&path:\JacORB1_4_beta4\bea\setenv-sample.bat
```

where:

drive&path

Is the drive and path into which you unzipped your JacORB files.

```
ns [ins_filename][-p port] [-t timeout]
```

where:

ins_filename

Is the name of the Naming Service file specified in the jacorb.properties file.

port

Is the number of the port on which the service is listening.

timeout

Is the server timeout, for example,

```
JacORB1_4_beta4\bea\ns ns_ref.txt
```

10. To start the Java interpreter explicitly, in a new DOS command window, execute the following:

```
drive&path:\JacORB1_4_beta4\bea\ setenv-sample.bat
```

where:

drive&path

Is the drive and path into which you unzipped your JacORB files.

```
jaco jacorb.naming.NameServer [filename][-p port][-t timeout]
```

where:

jacorb.naming.NameServer

Is the name of the Name Server.

filename

Is the name of the Naming Service file specified in the jacorb.properties file.

port

Is the number of the port on which the service is listening.

timeout

Is the server timeout, for example,

```
JacORB1_4_beta4\bea\jaco bea.club.ClubServer
```

Using VisiBroker for Java With the Adapter

VisiBroker is a complete CORBA 2.3 Object Request Broker (ORB) that supports the development, deployment, and management of distributed object applications across a variety of hardware platforms and operating systems. In addition to VisiBroker (the ORB), three other components are available with VisiBroker:

- Naming Service, which allows you to associate one or more logical names with an object implementation and to store those names in a namespace. It also lets client applications use this service to obtain an object reference using the logical name assigned to that object.
- Event Service, which provides a facility that separates the communication between objects. It provides a *supplier-consumer* communications model that allows multiple *supplier objects* to send data asynchronously to multiple *consumer objects* through an event channel.
- Gatekeeper, which runs on a Web server and enables client programs to locate and use objects that do not reside on the Web server and to receive callbacks, even when firewalls are used. The Gatekeeper can also be used as an HTTP daemon, thereby eliminating the requirement for a separate HTTP server during the application development phase.

The iWay Adapter for CORBA for BEA WebLogic supports VisiBroker for Java Version 4.5 and communicates using IIOP version 1.1. Applications created with VisiBroker for Java can communicate with object implementations developed with VisiBroker for C++.

VisiBroker Requirements

VisiBroker requires the Java Development Kit (JDK) or the Java Runtime Environment (JRE). You can obtain these tools from the Sun Microsystems Web site at http://java.sun.com/). JRE version 1.2.2 or higher is required to run the VisiBroker Console. You must install the JRE before you install VisiBroker. However, VisiBroker supports any current version of Java for your applications.

The iWay Adapter for CORBA for BEA WebLogic requires that an IOR file be available to locate the reference to the Interface Repository. However, the VisiBroker IOR file is not automatically output to a file. Modify your IR startup procedure to automatically output the start-up IOR reference to a file. For example, use the following command to automatically output the IOR reference to ir.ior:

```
irep myIr >ir.ior
```

VisiBroker configuration and run-time requirements for using the iWay Adapter for CORBA for BEA WebLogic include:

- The PATH environment variable must point to the VisiBroker libraries used by the iWay Adapter for CORBA for BEA WebLogic.
- The VisiBroker JAR file must be in the class path.
- The IFR must be populated with the IDL of the objects for which you want to create Web services.
- The VisiBroker Naming Service and IFR must be running.
- The CORBA servers you are using must be running or be set up to start on demand.

Syntax: How to Start the Interface Repository

With VisiBroker you start the Interface Repository using the following command:

```
irep myIr >ir.ior
```

where:

myIR

Is the startup IOR reference.

Syntax: How to Start the Naming Service

With VisiBroker you start the naming service using the following command:

```
$start nameserv NS_name
```

where:

NS_name

Is the name of the naming service.

Procedure: How to Verify the Server

To verify that your server is registered in the Naming Service and your IFR is loaded:

1. Select *VisiBroker* and then *VisiBroker Console* from the Start Programs menu.
2. Expand VisiBroker ORB Services.
3. Expand the *Naming Services* folder or the *IFR* folder.

   A list of the Naming Service or IFR objects appears in the right pane.

Alternatively, you can enter the following command at a command prompt:

```
osfind
```

This command finds the name of the server running the VisiBroker Naming Service. It is usually the machine where VisiBroker is installed.

For more information, refer to the *VisiBroker for Java Installation Guide*, available at the following URL:

http://info.borland.com/techpubs/books/vbj/vbj45/framesetindex.html

Using Orbacus With the Adapter

Orbacus is CORBA compliant product that is designed for rapid development, deployment, and support of your C++ or Java programs. Orbacus is provided in source code form and may be easily embedded into memory constrained applications.

Orbacus is recommended to organizations seeking to:

- Build integrated or distributed systems using a CORBA-compliant ORB.
- Cost effectively build a CORBA system.
- Access source code for better control of their applications.

Benefits of Orbacus

The benefits of using Orbacus with the adapter are that it provides:

- An infrastructure for applications to perform well and be scalable.
- A CORBA ORB for a large range of applications.
- Code that is easy to embed.

APPENDIX C

Supported IDL Types

Topics:

- IDL Base Types Supported
- IDL Constructed Types Supported

This section lists the Interface Definition Language (IDL) types that are supported by the adapter.

IDL Base Types Supported

The iWay Adapter for CORBA for BEA WebLogic supports the following IDL base types:

- boolean
- char
- double
- float
- long
- long long
- octet
- wcar
- wstring
- wstring<10>
- short
- string
- unsigned long
- unsigned long long
- unsigned short
- wchar

IDL Constructed Types Supported

The iWay Adapter for CORBA for BEA WebLogic supports the following IDL constructed types:

- array (where dimension is less than or equal to 2)
- enum
- interface (as a return value, but not within a return value of another constructed type)
- sequence
- struct
- typedef
- union

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**

iWay Adapter for CORBA for BEA WebLogic User's Guide DN3501476.0205
Version 5 Release 5