iWay

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

October 15, 2004
DN3501476.1004

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 iWay Application Explorer in BEA WebLogic Workshop	Describes the use of iWay Application Explorer as implemented in BEA WebLogic Workshop.
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 when you call, please provide the following information:

- Your six-digit site code number (*xxxx.xx*).
- Your software configuration.

The following table lists the information to provide about your software configuration.

	Version-Build Date	HF/Service Pack	Patches	OS	Java Version
iWay Product					
Third-party Application Server					
EIS (adapter target)					

Note: For the EIS, ensure you record the application or database name and release level, including minor versions, for example, 4.6.1.

- The exact nature of the error or problem, specified as follows:
  - Steps to reproduce the problem.
  - Problem description (be as specific as possible).
  - Error message(s).
- To best define the problem, provide the following:
  - Screen captures of the error
  - Error output files
  - Trace files and log files
  - Log transaction
  - XML schemas and/or document instances
  - Other input documents (for example, transformations)
  - Configuration files (all are applicable):

    .xch files

    config.xml file

    base.xml file

    repository.xml file

    ibserepo.xml file

.dic files

.rules files

- Environment variable settings:

  IWAY55

  IWAY55OEM

  CLASSPATH

  JAVA_HOME

  ACBDIR

  CBDIR (UNIX)

- Has the process, procedure, or query ever worked in its current form? Has it changed recently? If so, how (provide specific details)? How often does the problem occur?
- Can this problem be reproduced? If so, how? Can it be consistently reproduced?
- Have you tried to reproduce your problem in the simplest form possible?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production?
- Do you just have questions about functionality or documentation?

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
- Benefits 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 have 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 allows 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 a number of 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 WebLogic integration 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.

Benefits of the iWay Adapter for CORBA for BEA WebLogic

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

- 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 can be achieved without custom coding.
- Business processes can request and receive data from your CORBA system using services.
- Adapter services are standards-based. 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/`

- The iWay Adapter for CORBA for BEA WebLogic provides 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 iWay Adapter for CORBA for BEA WebLogic benefits from 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 iWay Adapter for CORBA for BEA WebLogic is secure, using 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 section 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 which components will be involved in the integration,such as Web services or business processes designed in WebLogic Workshop.
- Determining which external systems and technologies will be involved in the integration,such as CORBA systems and other EISs.
- Determining which iWay Adapters 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). Note that 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 which specific CORBA integration objects and workflows 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.
- Logins required to access CORBA transports and perform the required operations.
- 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 that will be used 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 in conjunction with the following components:

- iWay Application Explorer

and

- iWay Business Services Engine (iBSE)

or

- 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 iWay 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 iWay 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, 3, and Appendix A of this guide • *iWay Installation and Configuration for BEA WebLogic* • *iWay Servlet Application Explorer for BEA WebLogic*
iWay 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 iWay Business Services Engine

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

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

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

Web services is a distributed programming architecture that 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
- Generating a Request and a Response Schema
- Understanding iWay Business Services

The iWay Adapter for CORBA for BEA WebLogic uses XML documents to communicate with your CORBA system's 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

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 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 Application Explorer. The default port is 7001.

   Application Explorer opens.

   The Available Hosts drop-down list appears in the upper-right corner. Three tabs appear near the top of the Application Explorer screen. From left to right they are:

   - iWay Adapters, where you create and manage connections to your CORBA application, view metadata, and create schemas and business services.
   - iWay Events, where you configure event listening for available applications.
   - iWay Business Services, where you run business services.

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

   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.

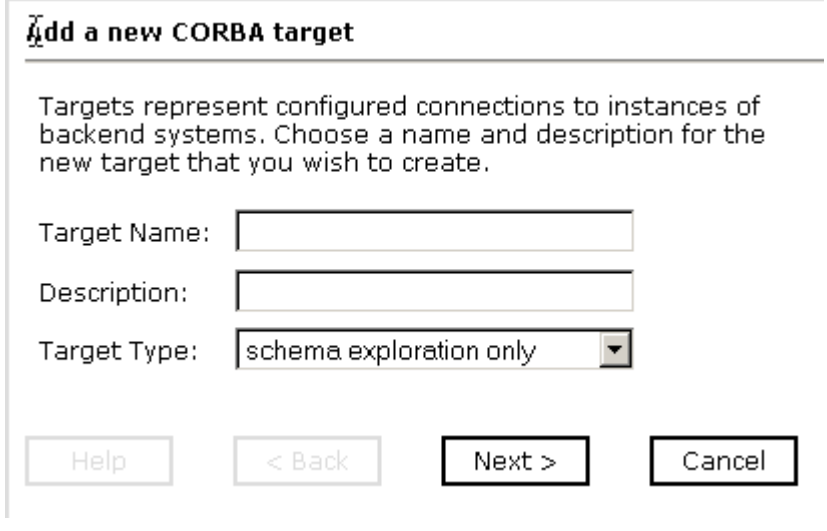
Creating and Managing a Connection

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.

Procedure How to Define a New Target

1. In the left pane of Application Explorer, expand the *iWay 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:



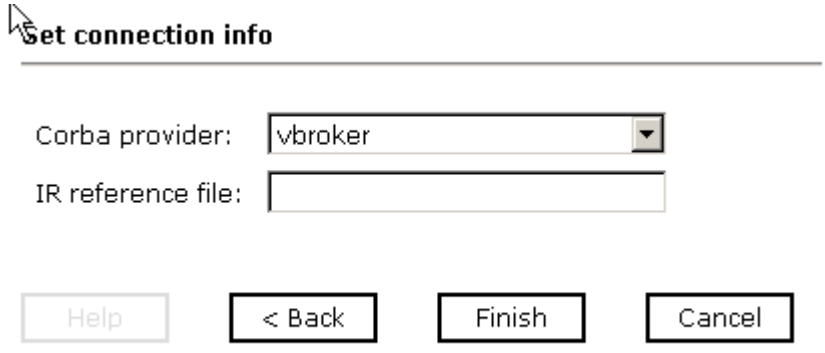
   a. In the Target Name field, type a descriptive name for the target (for example, CORBA_Connection).

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

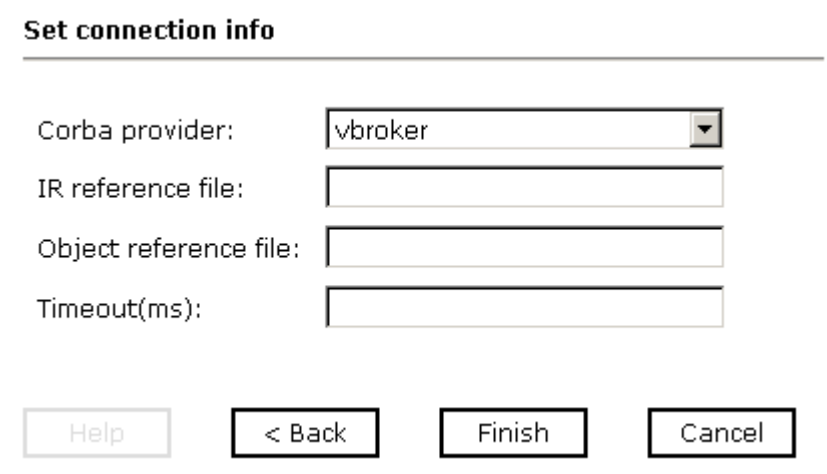
   c. From the Target Type drop-down list, select a target type (for example, schema exploration only). See the screens and table in step 4 to see which option to select.

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

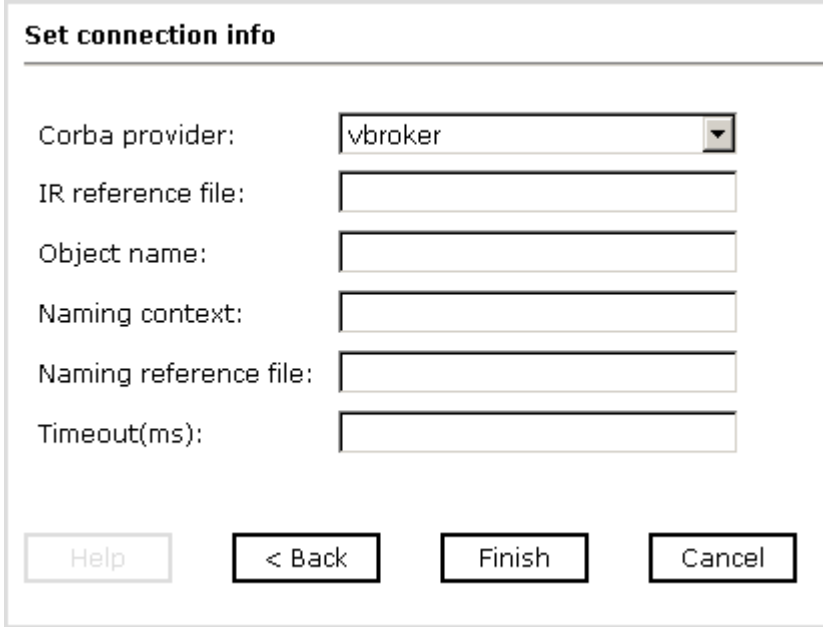
- If you selected a target type of schema exploration only, the Set Connection info dialog box opens containing two parameters and three action buttons (Back, Finish, and Cancel).



- If you selected a target type of configuration without naming, the the Set Connection info dialog box opens containing four parameters and three action buttons (Back, Finish, and Cancel).



- If you selected a target type of configuration with naming, the Set Connection info dialog box opens containing six parameters and three action buttons (Back, Finish, 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. For more information about these providers, 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. This parameter 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 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.
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.

b. 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

1. In the left pane, expand the *iWay 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 JMS component is available, the node under the JMS node displays a plus sign indicating that you are connected to the defined target. Otherwise, an error message appears in the right pane.

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 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 re-open it. The connection node now has an x icon, indicating that it is closed, as shown in the following figure:

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

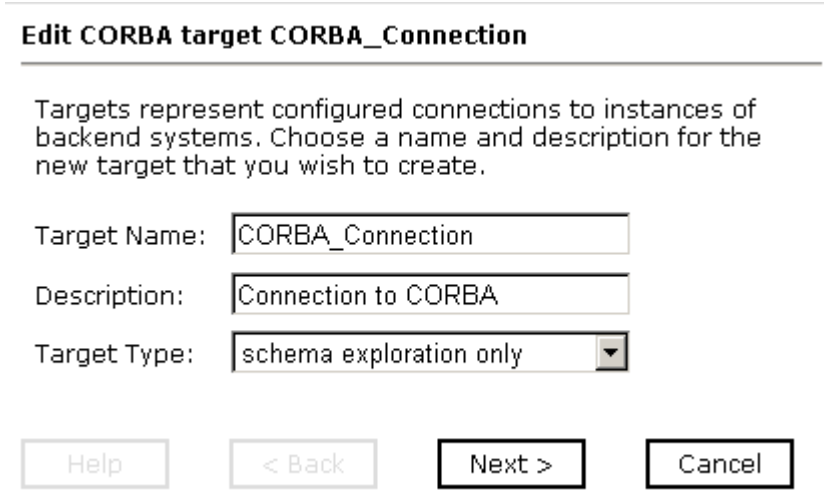
Editing a Defined Target

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. In the left pane of Application Explorer, expand the *iWay 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 action buttons (Next and Cancel).



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 action buttons (Back, Finish, and Cancel).

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

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 JMS targets in the left pane of the explorer.

Procedure How to Delete a Defined Target

1. Expand the *iWay 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.

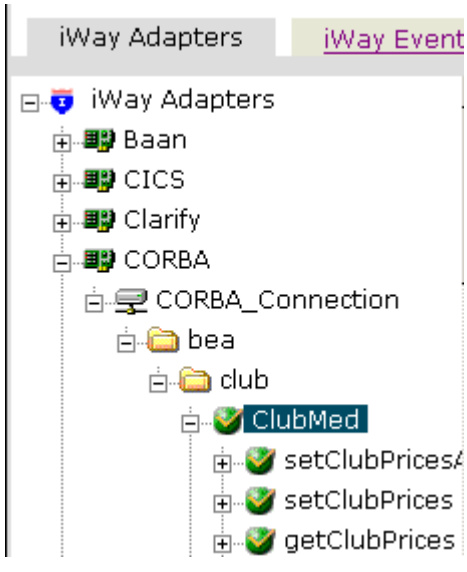
Generating a Request and a Response Schema

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.

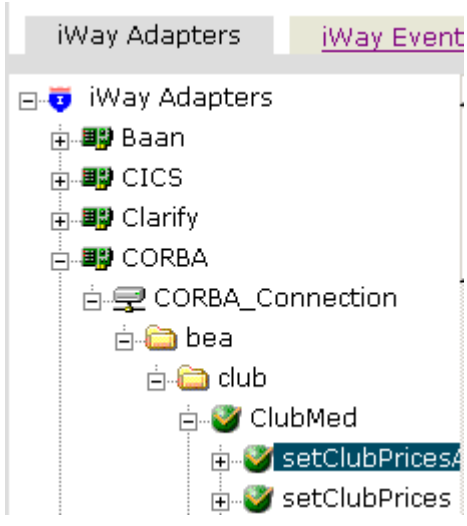
- **Service requests** are requests for action that your application makes to your CORBA system. As part of the definition, the request schema defines the input parameters required by the CORBA system.
- **Service responses** are the way the CORBA system responds to the service request. A service response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

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 2-7.
2. Expand the tree under the integration objects to see the items for which you may create a schema:

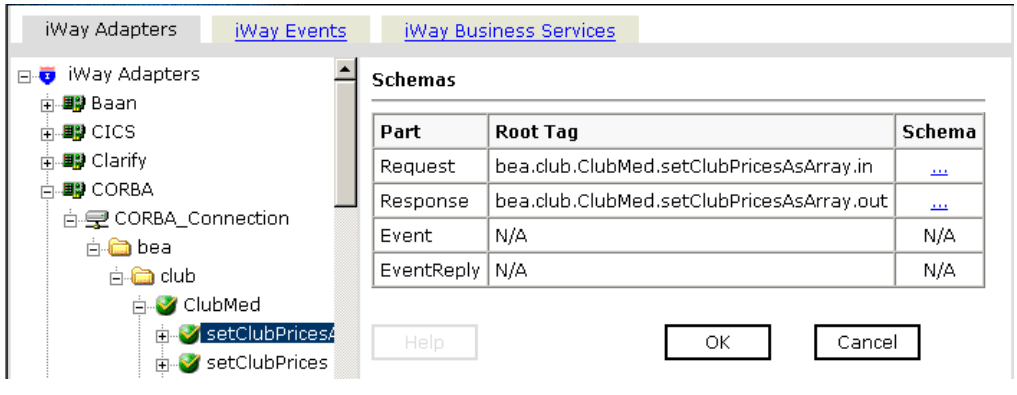


3. Expand and then select the node beneath the integration object for which you want to create the schema:



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.



5. To view the request schema, click the ellipsis symbol that is located in the third column of the Request row. The following is an illustration 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 is an illustration of a response schema.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  Generated by the iBSE 2004-08-02T20:56:27Z
  -->
- <xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element
      name="bea.club.ClubMed.setClubPricesAsArray.out" /
  </xsd:schema>
```

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 JMS 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.

Understanding iWay Business Services

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

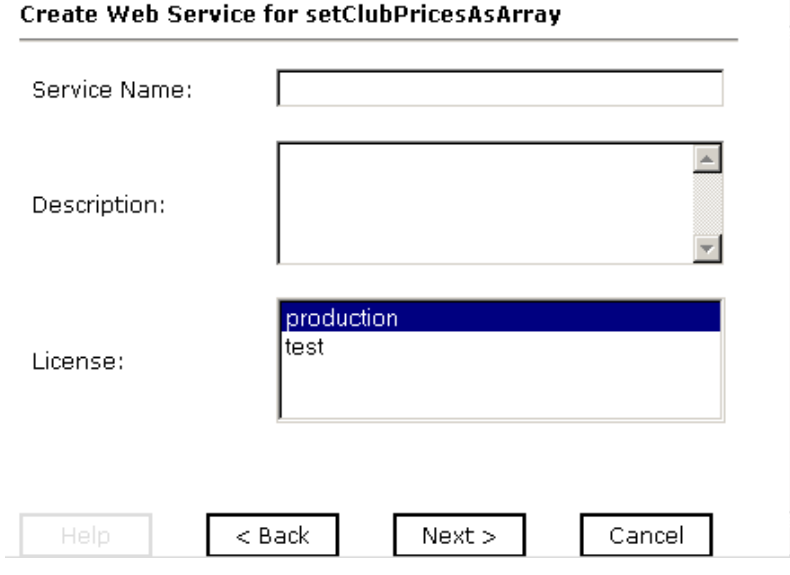
A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a "black box" that may require 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 2-7.
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 iWay Business Service*.

   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 *Use an existing service*, a drop-down list appears from which you must select the service. Select an existing service and proceed to step 6.
   - If you select *Create a new service*, the Create Web Service dialog box opens in the right pane:



If you are creating a new service, type the following parameters:

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

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

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

6. Click *Next*.

Another 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 iWay Business Services tab, and the new business service appears in the left pane.

Testing a Business Service

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

Procedure How to Test a Business Service

1. If you are not on the iWay 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 iWay Business Services.

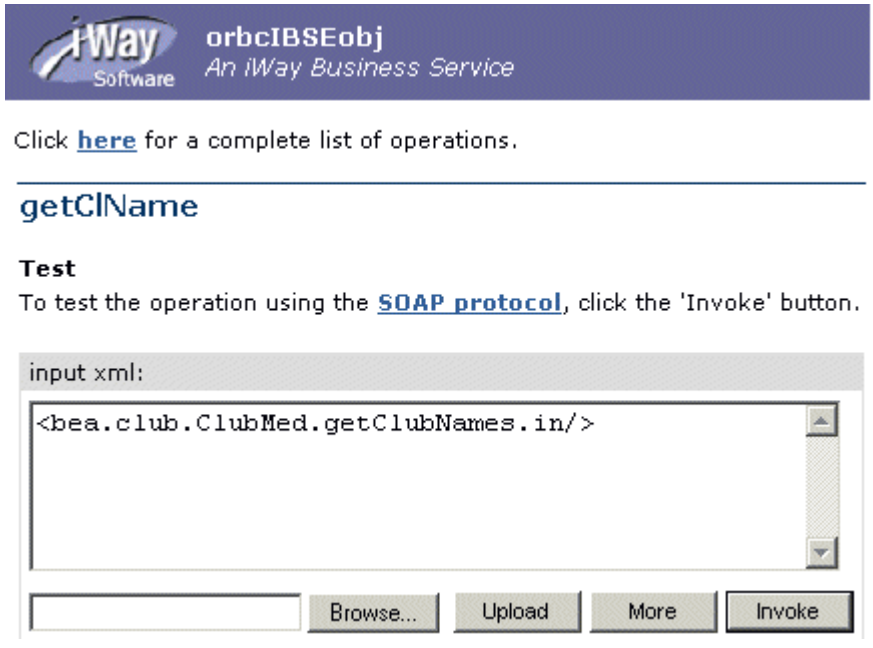
3. Expand the *Services* node.

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.

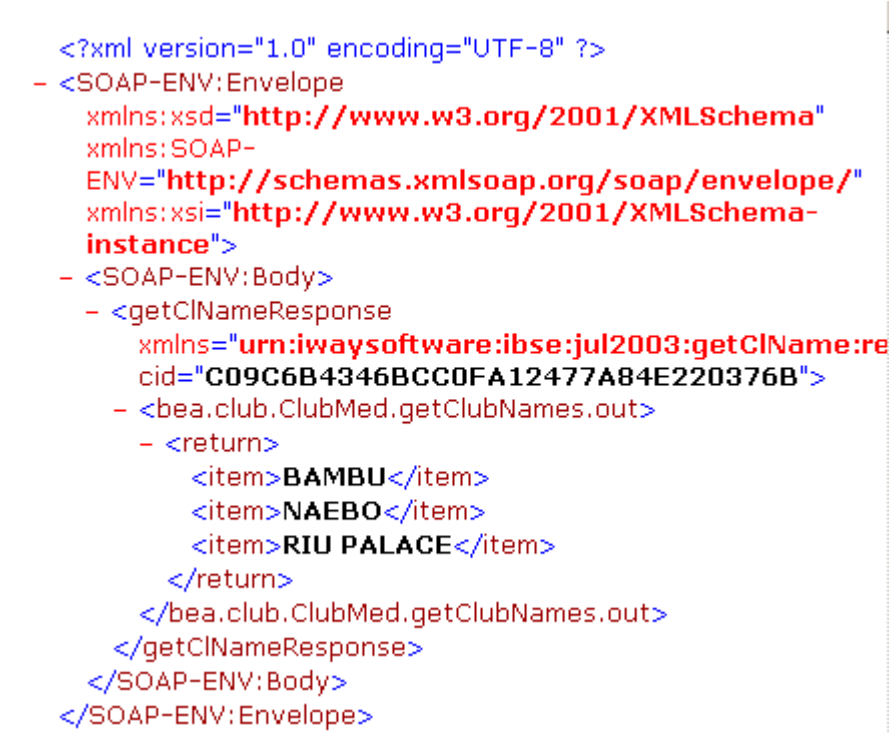


This pane provides a text field in which to paste the XML input or browse to a file that can be uploaded. Below the text field is the browse field and three action buttons.

6. Provide the appropriate XML input.

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. If you are not already in the iWay 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*.

Credential Mapping

For each SOAP request that is received, iBSE checks to see if a user name and password is included in the SOAP header. If a user name and password is available, iBSE acquires this information and replaces the values retrieved from the repository when pushing the request to the iWay Adapter.

CHAPTER 3

Using Web Services Policy-Based Security

Topics:

- Web Services Policy-Based Security
- Configuring Web Services Policy-Based Security

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

Web Services Policy-Based Security

Web services provide a layer of abstraction between the back-end business logic they invoke and the user or application running the Web 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 Web service.

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

A policy is a set of privileges dealing with the execution of an iWay Business Service (iBS) that can be applied to an existing or new iBS. When you set specific rights or privileges inside a policy, you do not have to recreate privileges for every iBS that has security concerns in common with other iWay Business Services. Instead, you can use one policy for many iWay Business Services.

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

The iBS administrator creates an "instance" of a policy type, names it, associates individual users and/or groups (a collection of users), and then applies that policy to one or more iWay 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 that specific iBS. The policy type that is supported is Resource Execution, which 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, anybody can execute the iBS, until the Resource Execution policy is associated to the iBS. At that time, only those granted execution permission, or users who are not part of a group that was denied execution permissions, have access to the iBS.

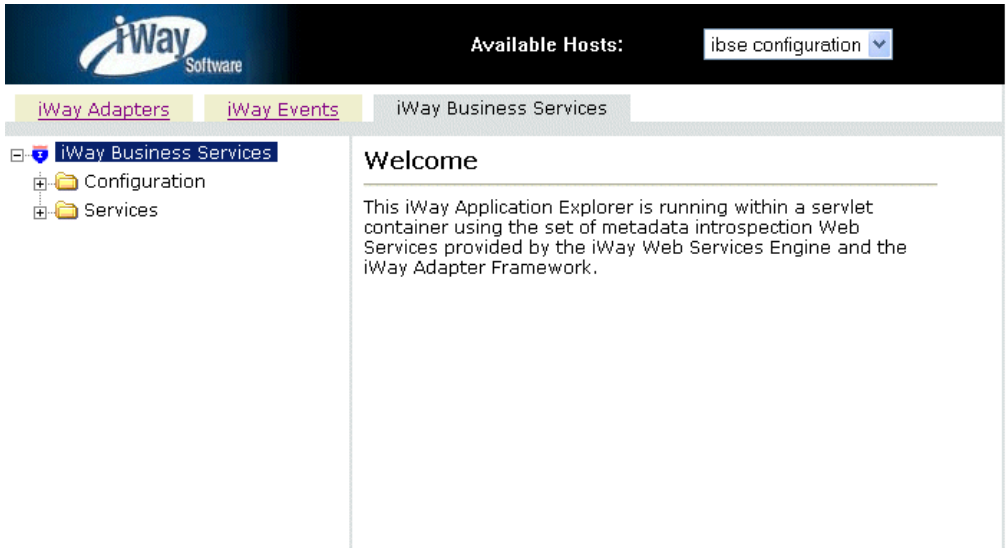
Configuring Web Services Policy-Based Security

The following procedure describes how to configure iBSE policy-based security.

Procedure How to Create and Associate a User With a Policy

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 iWay Servlet Application Explorer.

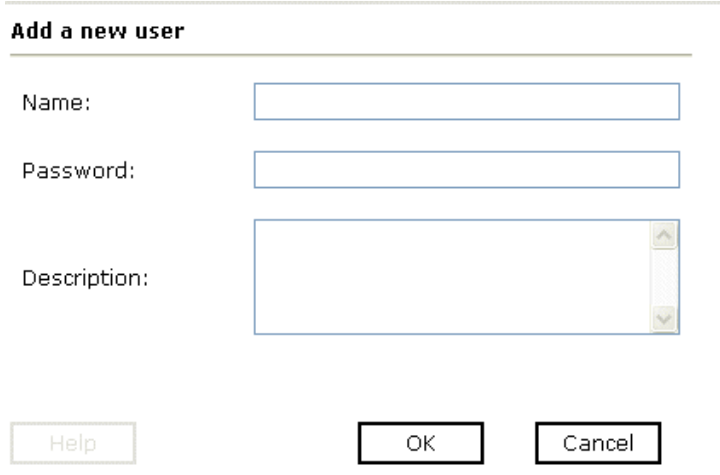
1. Open *iWay Servlet Application Explorer*.



   a. Select the *iWay Business Services* tab.
   b. Expand the *Configuration* node.
   c. Expand the *Security* node.
   d. Expand the *Users and Groups* node.
   e. Select *Users*.

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

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



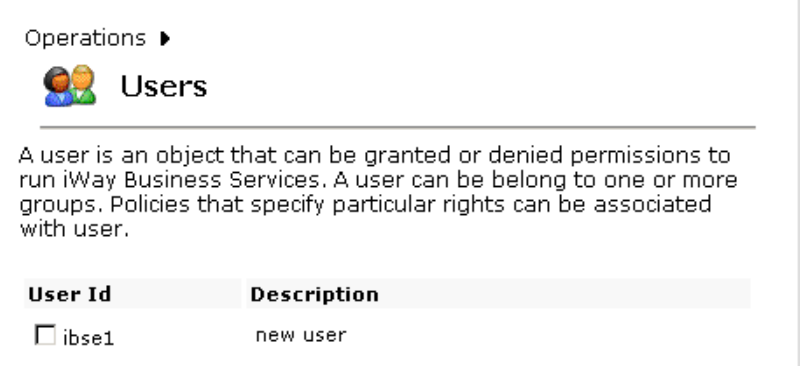
   a. In the Name field, type a user ID.

   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 new user is added to the configuration.

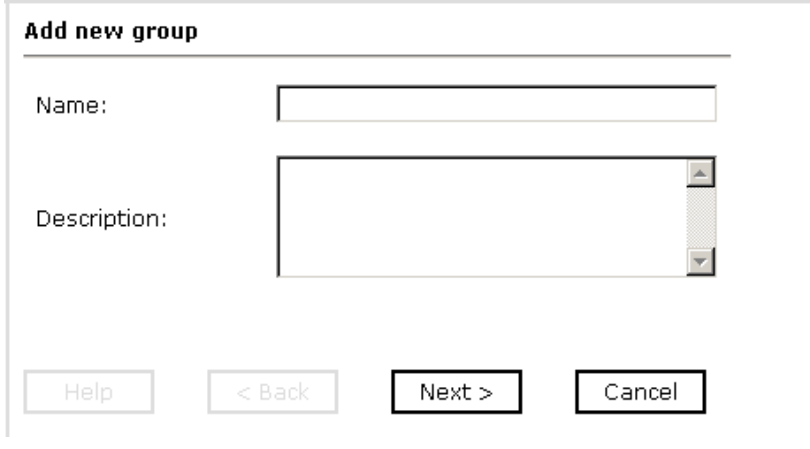


Procedure How to Create a Group to Use With a Policy

To create a group to use with a policy:

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

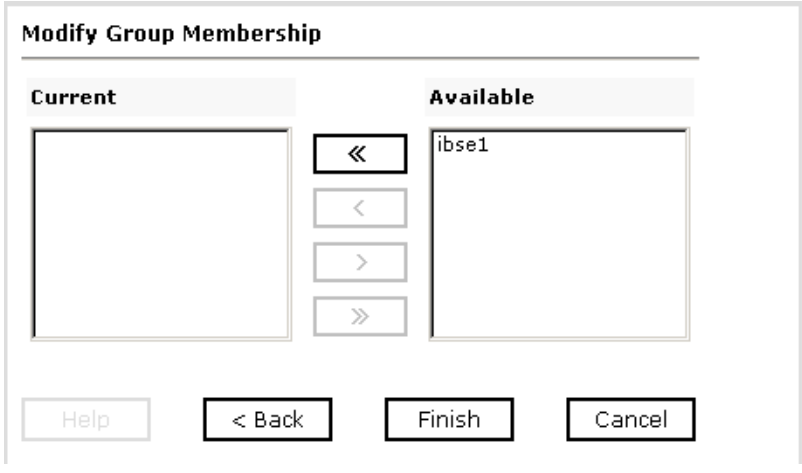
   The Add new group dialog box opens.



   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*.

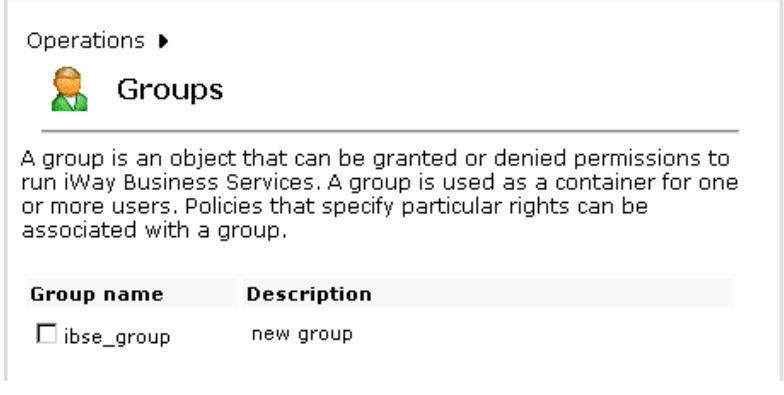
The Modify Group Membership dialog box opens.



You can either highlight a single user in the list of available users and add it 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 to the configuration.

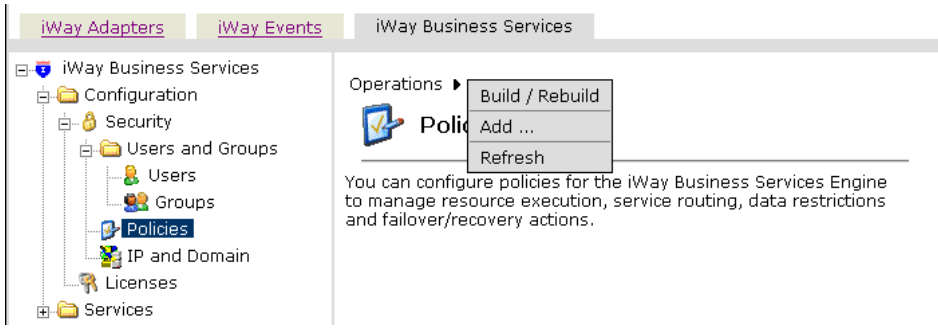


Procedure How to Create an Execution Policy

An execution policy governs who can execute the iBS to which the policy is applied.

To create a group to use with a policy:

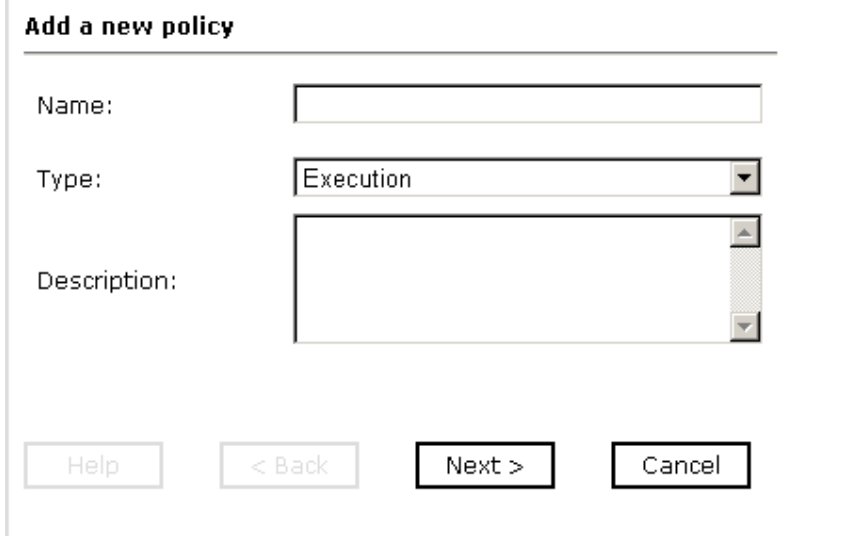
1. Open *iWay Servlet Application Explorer.*
   a. Select the *iWay Business Services* tab.
   b. Expand the *Configuration* node.



   c. Select *Policies*.

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

   The Add a new policy dialog box opens.



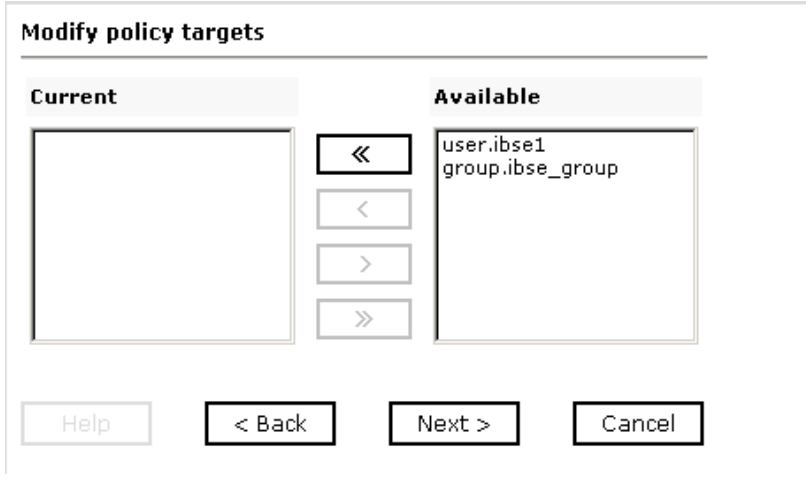
a. In the Name field, type a a name for the policy.

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 Modify policy targets dialog box opens.

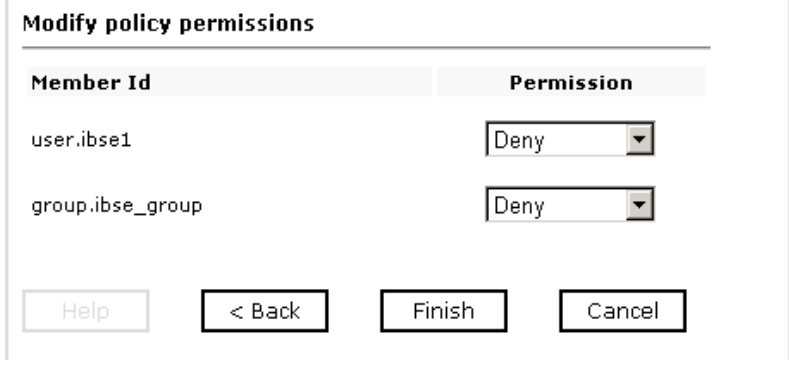


4. Select a minimum of one user or group from the Available pane.

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*.

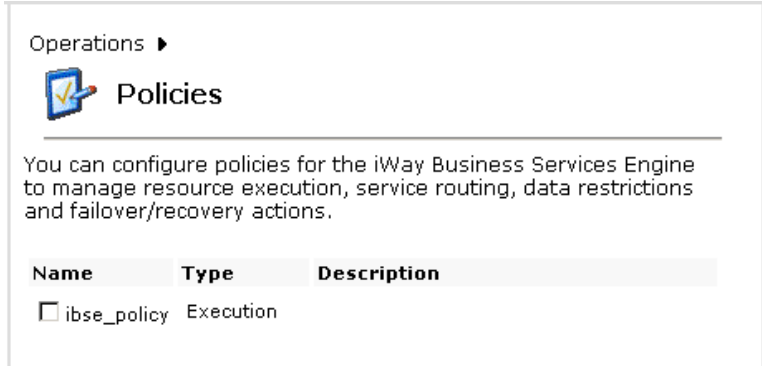
The Modify policy permissions dialog box opens.



You select whether users or groups may execute the iBS.

6. From the Permission drop-down lists, select *Grant* to permit execution or *Deny* to restrict execution.
7. Click *Finish*.

   The following pane summarizes your configuration.

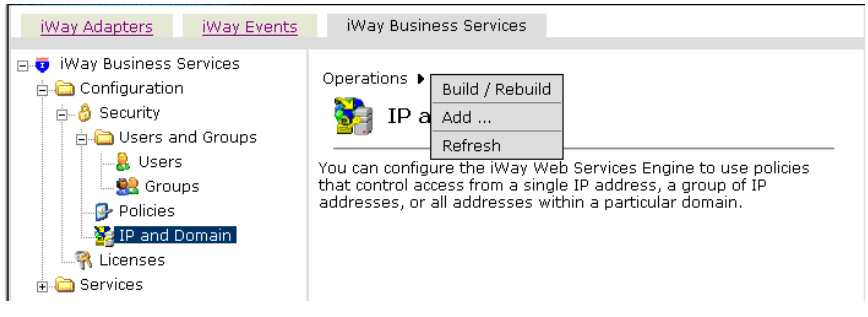


Configuring the IP and Domain Restrictions Policy Type

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 individual Web services. You need not create a policy, however, you must enable the Security Policy option in iWay Servlet Application Explorer.

Procedure How to Configure IP and Domain Restrictions

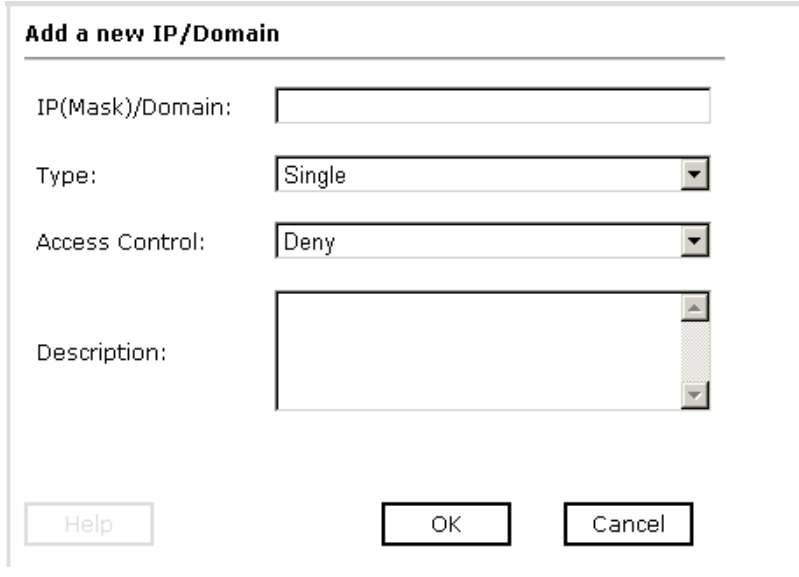
1. Open *iWay Servlet Application Explorer.*
   a. Select the *iWay Business Services* tab.
   b. Expand the *Configuration* node.
   c. Expand the *Security* node.



   d. Select *IP and Domain*.

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

The Add a new IP/Domain dialog box opens.



a. 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.

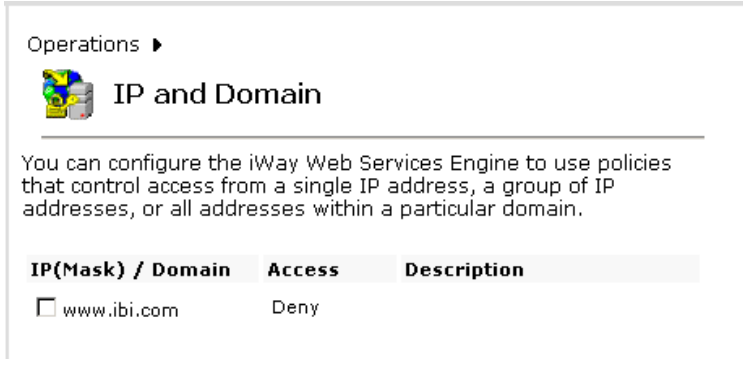
If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.

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

b. From the Type drop-down list, select the type of restriction.

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 pane summarizes your configuration.



CHAPTER 4

Management and Monitoring

Topics:

- Managing and Monitoring Services and Events Using iBSE
- Managing and Monitoring Services and Events Using iWay JCA
- Setting Engine Log Levels
- Configuring Connection Pool Sizes

Once you have created services and events using iWay Application Explorer, you can use managing and monitoring tools provided by iBSE and JCA to gauge the performance of your run-time environment. The following section describe how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

iBSE provides a console to manage and monitor services and events currently in use and 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 Access the Monitoring Console

To access the monitoring console:

1. Ensure that BEA WebLogic Server is started.
2. Enter the following URL in your Web browser:

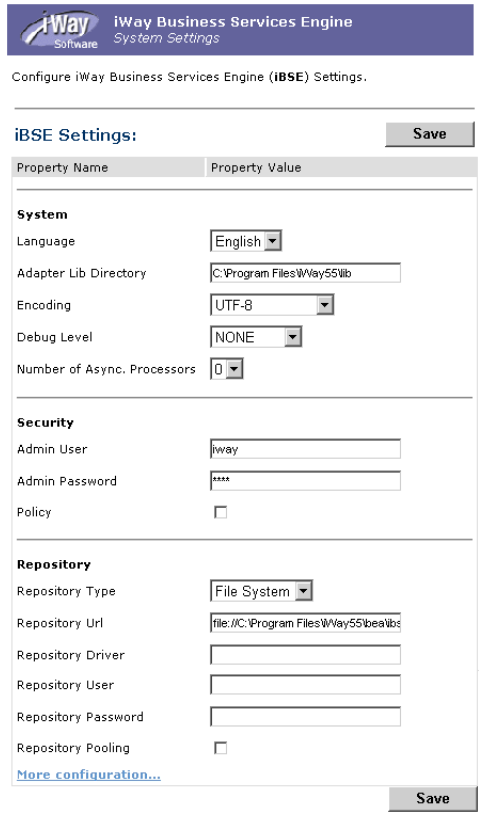
   `http://localhost:7001/ibse/IBSEConfig`

   where:

   localhost

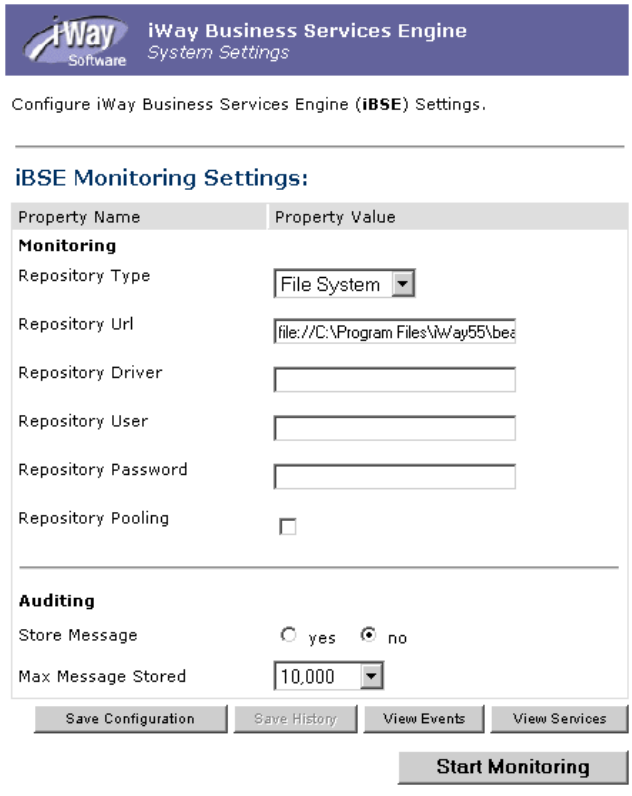
   Is where your application server is running.

The iBSE Settings page opens:



3. Scroll to the bottom of the page and click *More configuration*.

The iBSE Monitoring Settings page opens:



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

`http://localhost:7001/ibse/IBSEStatus`

where:

localhost

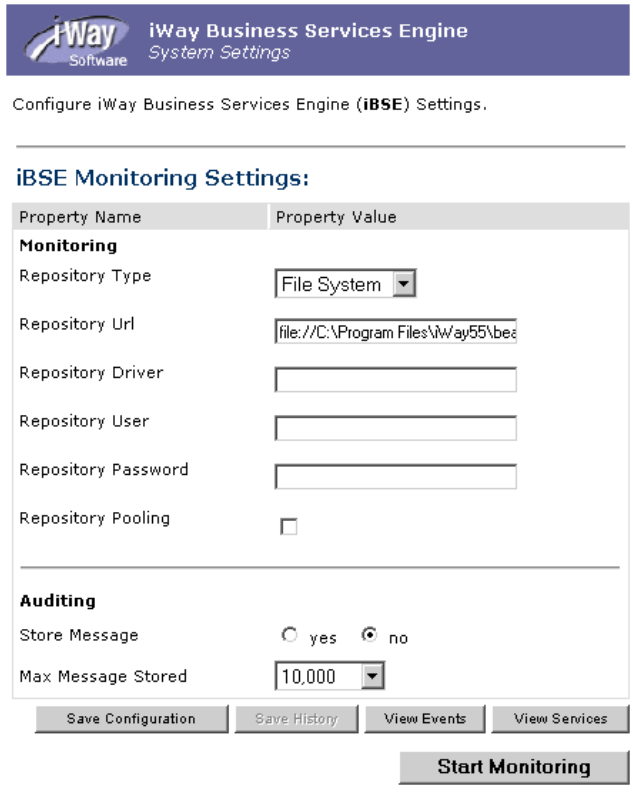
Is where your application server is running.

Procedure How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that BEA WebLogic Server is started.
2. Access the monitoring console.

The iBSE Monitoring Settings page opens:



3. Perform the following steps in the Monitoring section:
   a. Select the type of repository you are using from the Repository Type drop-down list.
   b. Enter a JDBC URL to connect to the database in the Repository URL field.
   c. Enter a JDBC Class to connect to the database in the Repository Driver field.
   d. Enter a user ID and password to access the monitoring repository database.
   e. Click the *Repository pooling* check box if you want to enable pooling.
4. Perform the following steps in the Auditing section:
   a. 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.
   b. 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's resources, consult your system administrator.

5. Click *Save Configuration*.
6. Click *Start Monitoring*.

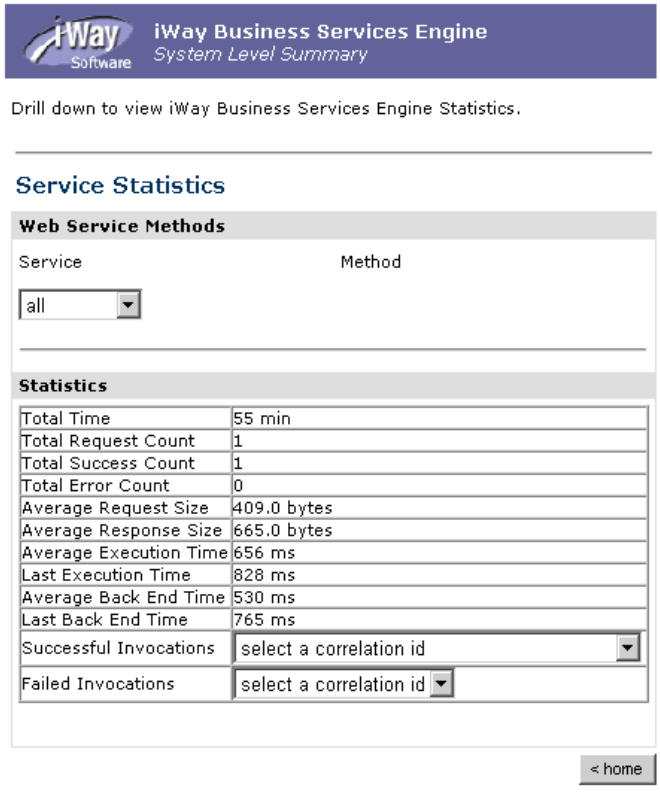
iBSE begins to monitor all services and events currently in use and store messages, if you selected this option. If you want to stop monitoring, click *Stop Monitoring*.

Procedure **How to Monitor Services**

To monitor services:

1. Ensure that BEA WebLogic Server is started.
2. Click *Start Monitoring* from the iBSE Monitoring Settings page.
3. Click *View Services*.

The System Level Summary page opens.



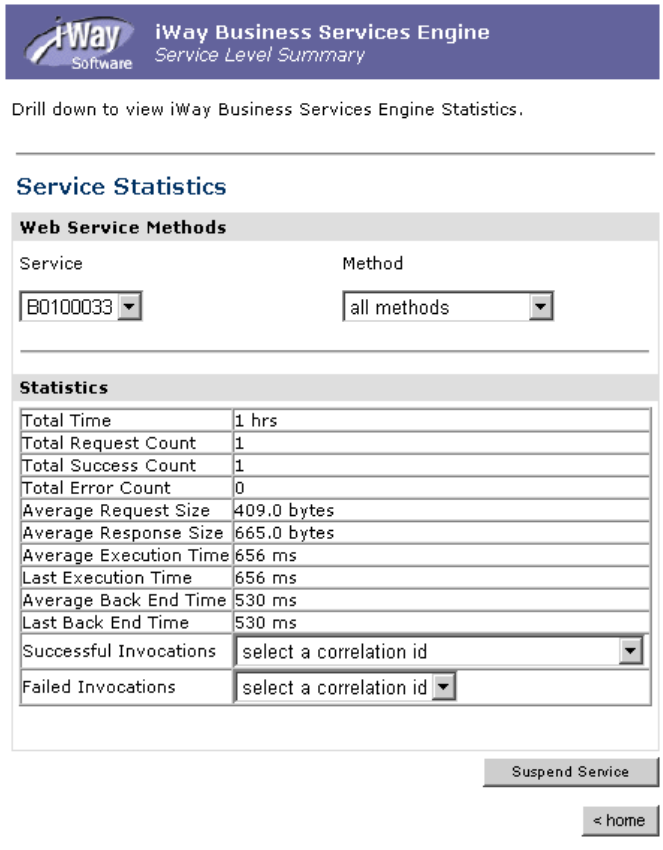
The system level summary provides services statistics at a system level. The following table provides a description of each statistic.

Statistic	Description
Total Time	The total amount of time iBSE is monitoring services. This time starts when you click *Start Monitoring* from the iBSE Monitoring Settings page.
Total Request Count	The total number of services requests that were made during this monitoring session.
Total Success Count	The total number of successful service executions.

Statistic	Description
Total Error Count	The total number of errors that were encountered.
Average Request Size	The average size of a service request that is available.
Average Response Size	The average size of a service response size that is available.
Average Execution Time	The average execution time for a service.
Last Execution Time	The last execution time for a service.
Average Back End Time	The average back end time.
Last Back End Time	The last back end time.
Successful Invocations	A list of successful services listed by correlation ID. Select a service from the drop-down list to retrieve more information for that service.
Failed Invocations	A list of failed services listed by correlation ID. Select a service from the drop-down list to retrieve more information for that service.

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

The Service Level Summary page opens.



A list of available methods for that service appears in the Method drop-down list.

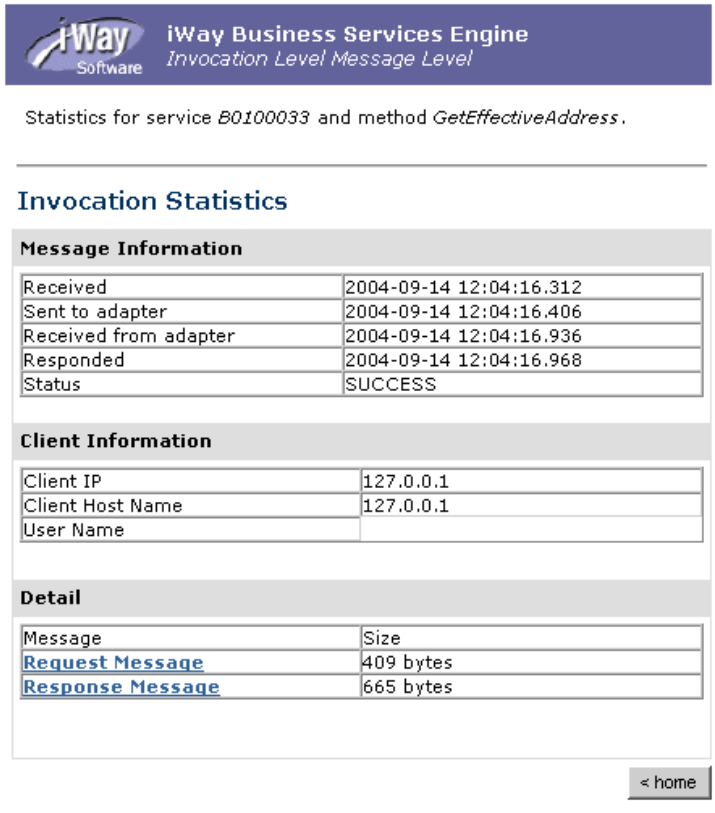
To stop a service at any time, click *Suspend Service*. To start the service, click *Resume Service*.

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

The Method Level Summary page opens.

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

The Invocation Level Statistics page opens.



Information pertaining to the message and client is provided.

7. Click the *Request Message* link to view the XML request document in your Web browser.

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

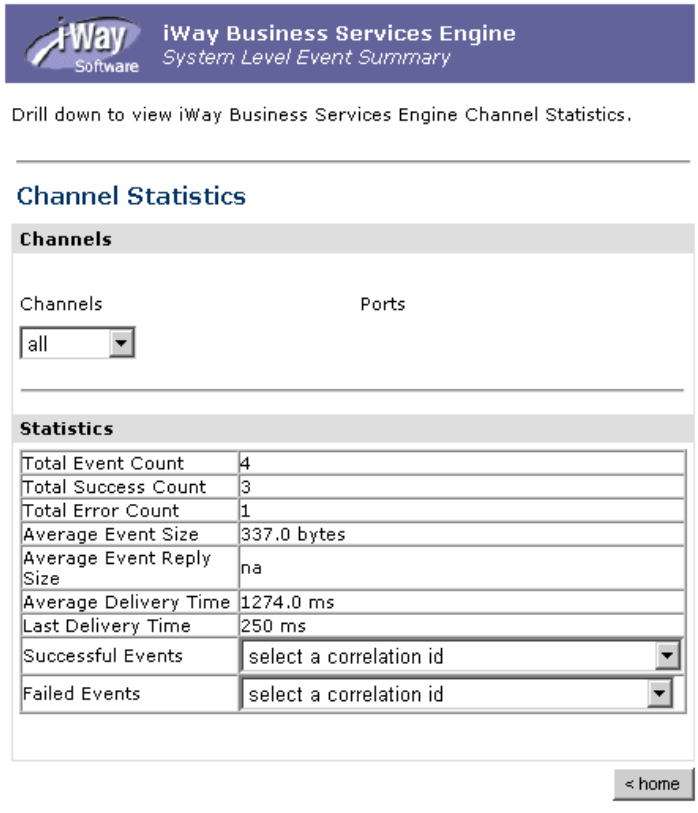
8. Click *home* to return to the iBSE Monitoring Settings page.

Procedure How to Monitor Events

To monitor events:

1. Ensure that BEA WebLogic Server is started.
2. Click *Start Monitoring* from the iBSE Monitoring Settings page.
3. Click *View Events.*

The System Level Summary page opens.



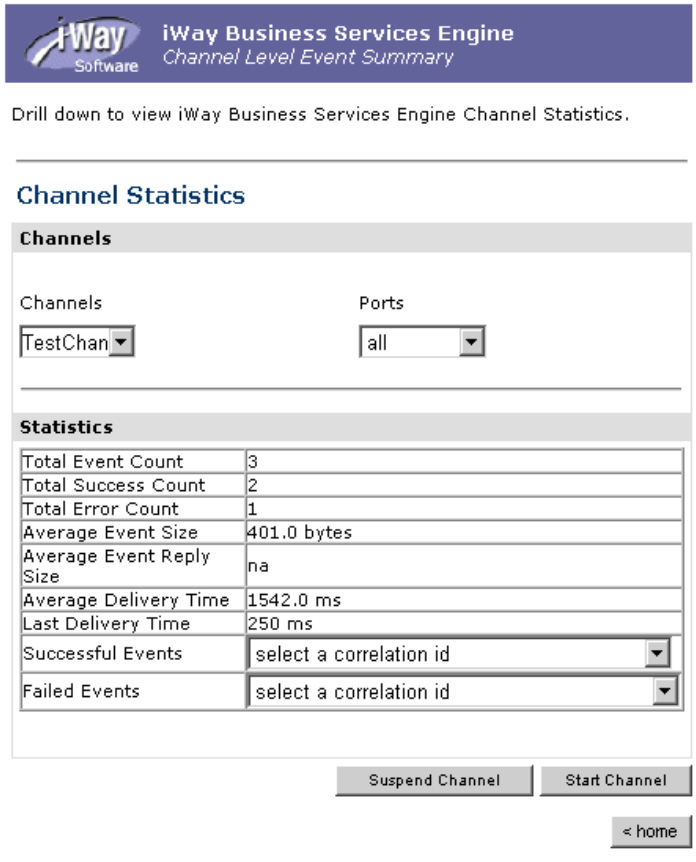
The system level summary provides event statistics at a system level. The following table provides a description of each statistic.

Statistic	Description
Total Event Count	The total number of events.
Total Success Count	The total number of successful event executions.
Total Error Count	The total number of errors that were encountered.
Average Event Size	The average size of an event request that is available.

Statistic	Description
Average Event Reply Size	The average size of an event response that is available.
Average Delivery Time	The average delivery time for an event.
Last Execution Time	The last execution time for an event.
Last Delivery Time	The last delivery time.
Successful Events	A list of successful events listed by correlation ID. Select an event from the drop-down list to retrieve more information for that event.
Failed Events	A list of failed events listed by correlation ID. Select an event from the drop-down list to retrieve more information for that event.

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

The Channel Level Event Summary page opens.

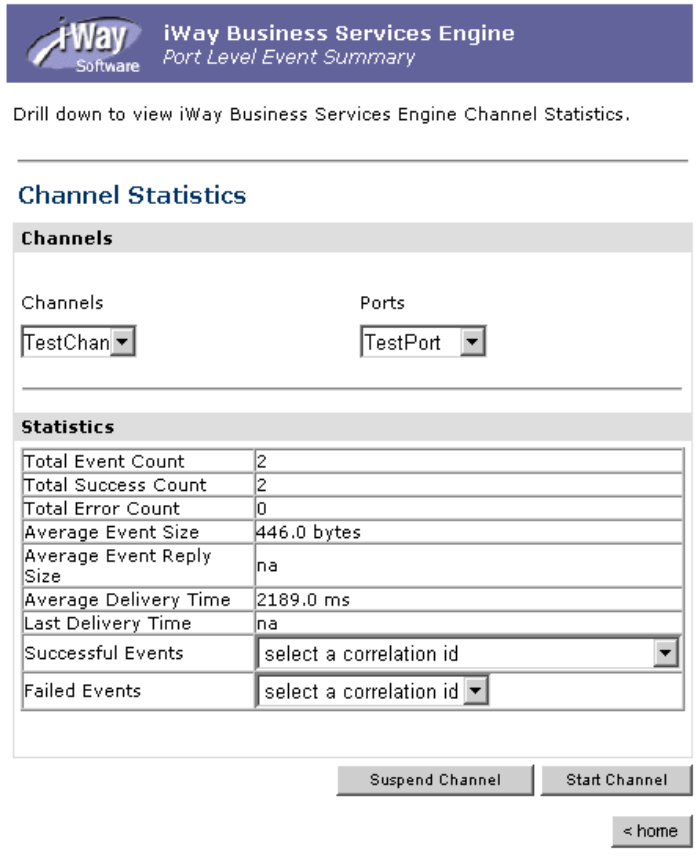


A list of available ports for that channel appears in the Ports drop-down list.

To stop a channel at any time, click *Suspend Channel*. To start the service, click *Start Channel*.

5. Select a port for the channel from the Ports drop-down list.

The Port Level Event Summary page opens.



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

The Event Level Statistics page for the channel and port you selected opens.

Information pertaining to the event message is provided.

7. Click the *Event Message* link to view the XML event document in your Web browser.
8. Click *home* to return to the iBSE Monitoring Settings page.

Managing and Monitoring Services and Events Using iWay JCA

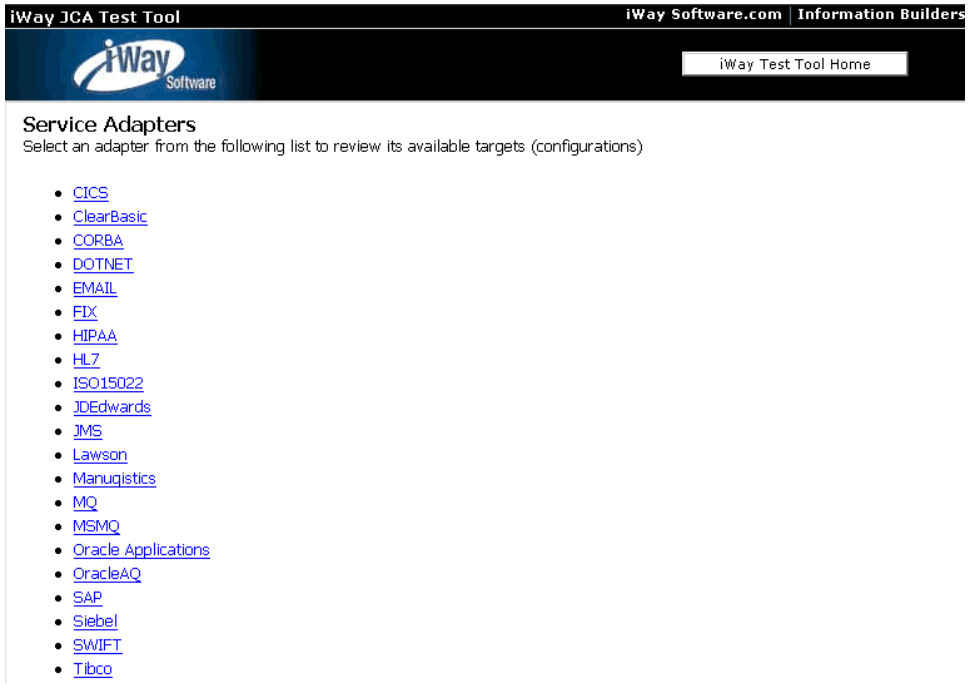
The following topics describe how to test service and event adapters using the iWay JCA Installation Verification Program (IVP).

Procedure How to Test the iWay Service Adapters

To test the iWay service adapters using the IVP:

1. To ensure that the targets you configured in iWay Application Explorer appear in the IVP, click *Refresh Manage Connection Factory.*
2. To display the available adapters, click the *Service adapters* link.

The following window, showing the list of deployed service adapters, opens.



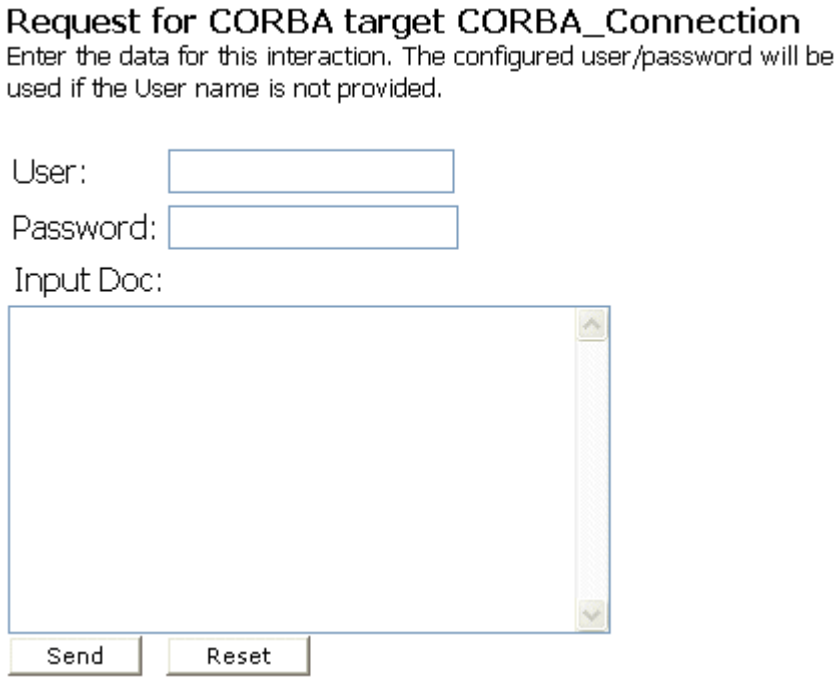
3. Select the adapter that you want to test.

The adapter displays all of the targets currently configured in the iWay repository for that adapter.

The following window shows that there is one target, CORBA_Connection, configured for the iWay Adapter for CORBA.

4. Click the desired target, for example, *CORBA_Connection*.

   The following pane, showing an input area in which you can provide XML code with which to test the adapter, opens.



5. Enter a username and password to connect to CORBA.
6. In the input area, enter a request document built from the iWay request schema.
7. Click *Send*.

   A response is returned from CORBA.

Testing the iWay Event Adapters Using the IVP

The iWay JCA Installation Verification Program (IVP) enables you to start and stop iWay event channels.

The tool also enables you to monitor events and provides statistics on channels.

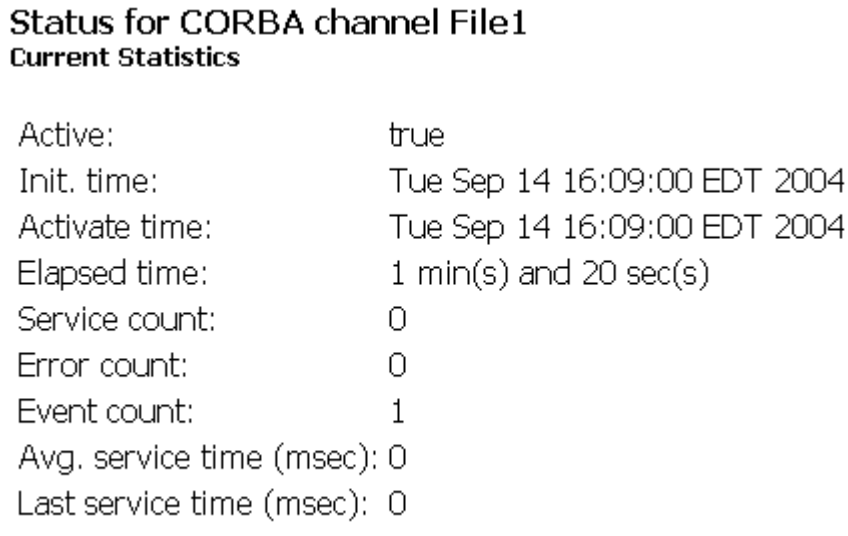
Procedure How to Test the iWay Event Adapters

To test the iWay event adapters using the IVP:

1. Click *Refresh Manage Connection Factory.*
2. To display the available adapters, click the *Events adapter* link.
3. Select the adapter that you want to monitor, for example, CORBA.

   The tool displays the channels that you already configured.

4. Click the *start* hyperlink to start the channel.



   Statistics for the event channel are returned, including:

   - The status of the channel.
   - The time the channel was initialized.
   - The number of events.
   - The event response time.

5. To stop the channel, click the *stop* hyperlink.

Monitoring Services

The following section describes how to use the iWay JCA Installation Verification Program (IVP) in Managed mode and monitor services in BEA WebLogic.

Procedure How to Use iWay JCA IVP in Managed Mode.

To use iWay JCA IVP in managed mode:

1. Open the *web.xml* file in a text editor.

   It is located in the following directory:

   ```
   <installDir>\bea\iwjcaivp\WEB-INF\web.xml
   ```

   where:

   ```
   <installDir>
   ```

   Is the location of your iWay 5.5 installation.

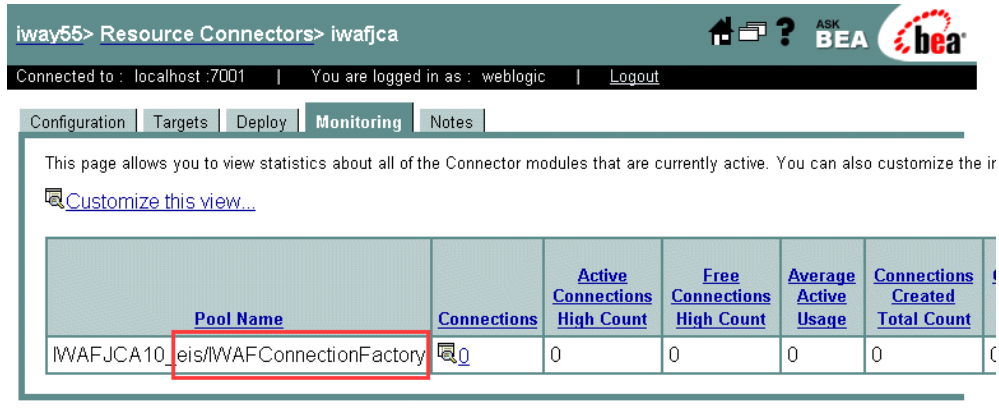
2. Locate the following lines:

   ```
   <context-param><param-name>iway.jndi</param-name><param-value></param-value><description>JNDI name for the IWAF JCA Resource Adapter. If not provided, the application will create a new one based on iway.home, iway.config and iway.loglevel.</description></context-param>
   ```

3. Enter the path to the JCA module for the iway.jndi parameter, for example:

   ```
   <param-value>eis/IWAFConnectionFactory</param-value>
   ```

   You can find this value by browsing to the Resource Connectors section in BEA WebLogic and checking the Pool Name for the JCA connector module. For example:



4. Restart WebLogic Server or redeploy the JCA connector module.

5. Open a browser to:

`http://hostname:port/iwjcaivp`

where:

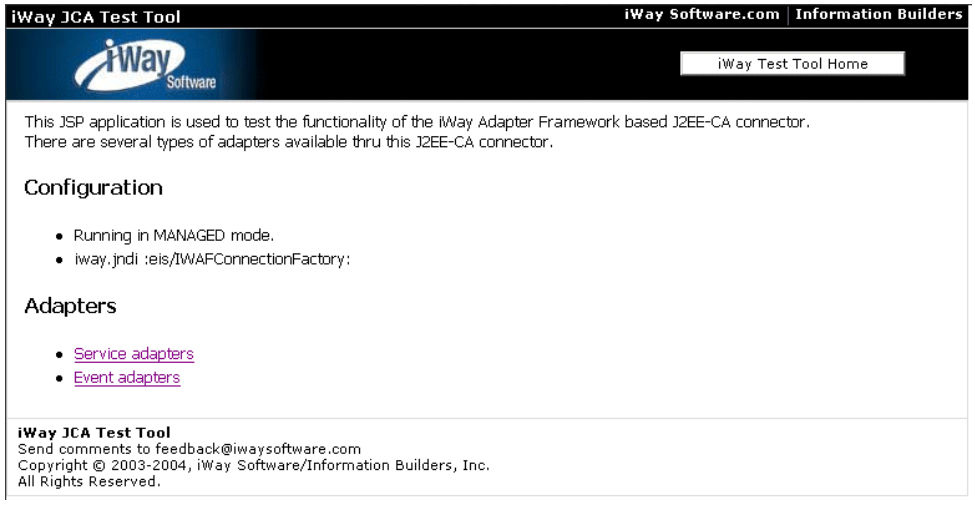
hostname

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

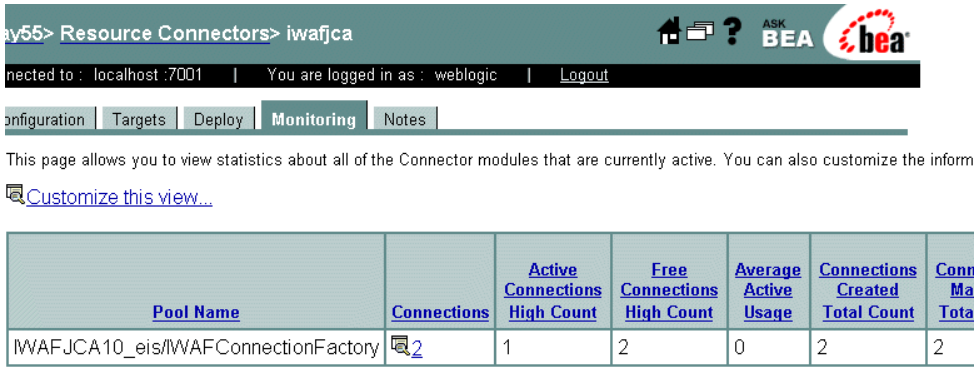
port

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

The iWay JCA Test Tool window opens and provides links for viewing iWay Service or Event adapters. Notice that it is now running in managed mode.



6. Test a service you have created for an iWay Adapter using Application Explorer.
7. Return to the Resource Connectors section in BEA WebLogic.



Monitoring statistics pertaining to the services you have executed are now available.

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:

   `http://hostname:port/ibse/IBSEConfig`

   where:

   hostname

   Is the hostname of the application server machine.

   port

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

   For example:

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

2. In the top *System* area, specify the level of tracing from the *Debug* drop-down list.
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 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. Also, be sure to review logs generated by your application server.

Leave the remainder of this file unchanged.

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

Configuring Connection Pool Sizes

The following section describes how to configure connection pool sizes using JCA.

Procedure How to Configure Connection Pool Sizes

To configure connection pool sizes:

1. Open the extracted weblogic-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.

APPENDIX A

Using iWay Application Explorer in BEA WebLogic Workshop

Topics:

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

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

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.

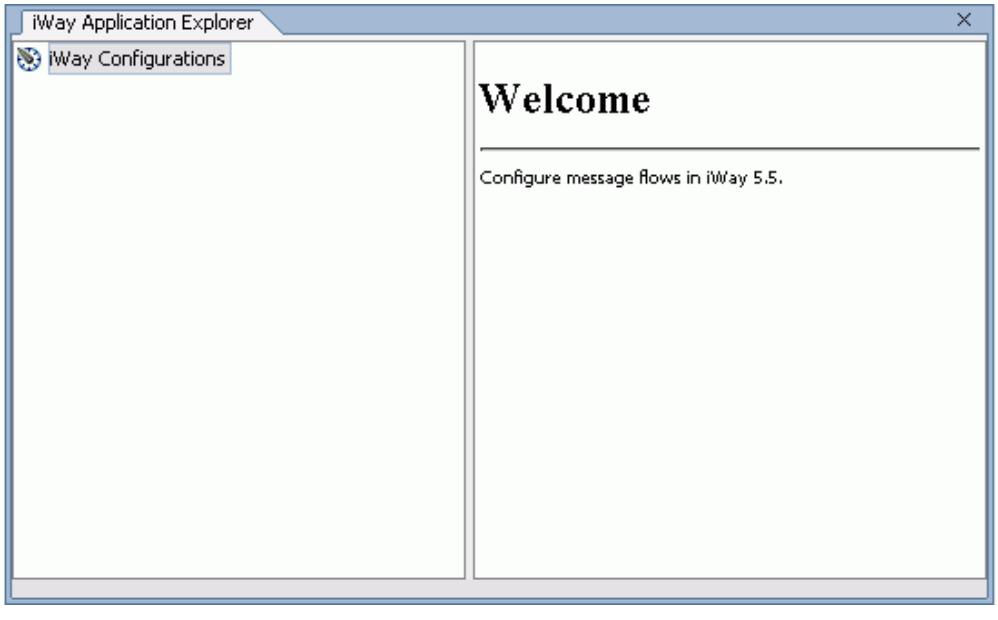
Starting iWay Application Explorer in WebLogic Workshop

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.

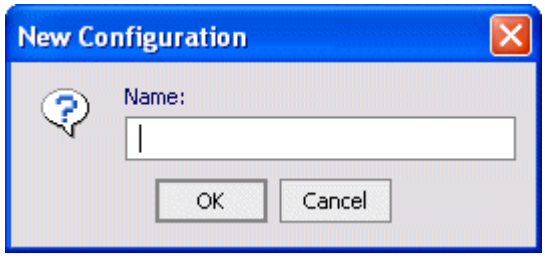


Procedure How to Define a New Configuration

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

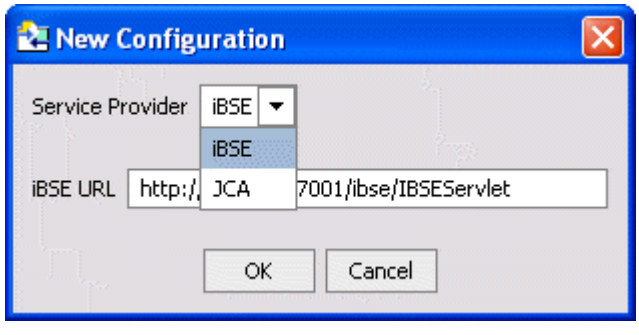
1. Right-click *iWay Configurations* and select *New*.

   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,

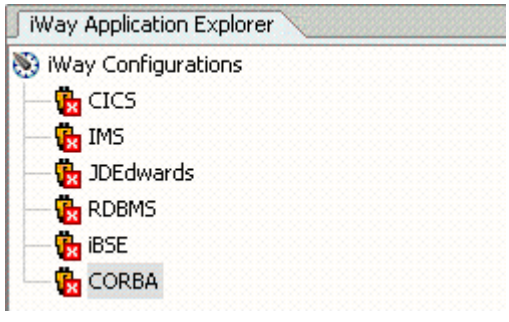
```
C:\Program Files\iWay55
```

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

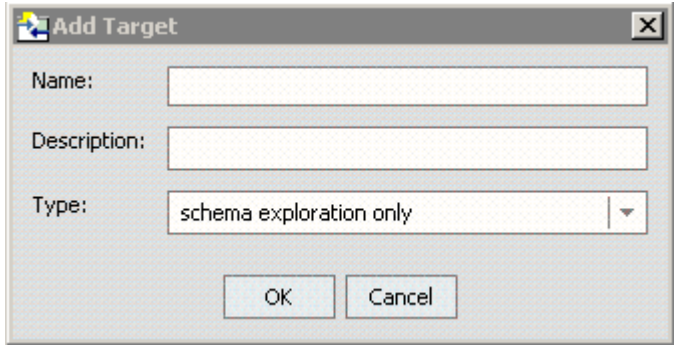
1. Expand the *iWay Service Adapters* node.

   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:

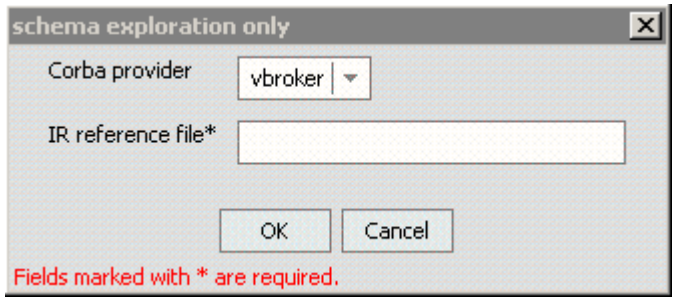


   a. Type a name (for example, CORBA_Connection) and a brief description for the new target.

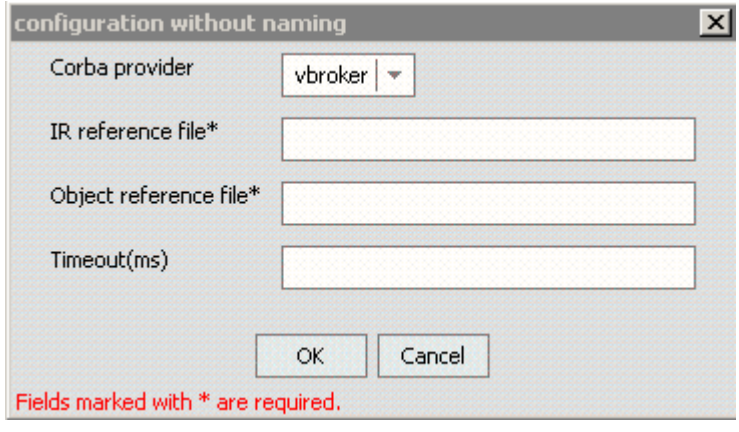
   b. From the Type drop-down list, select the type of target (for example, schema exploration only). See the screens and table in step 4 to see which option to select.

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

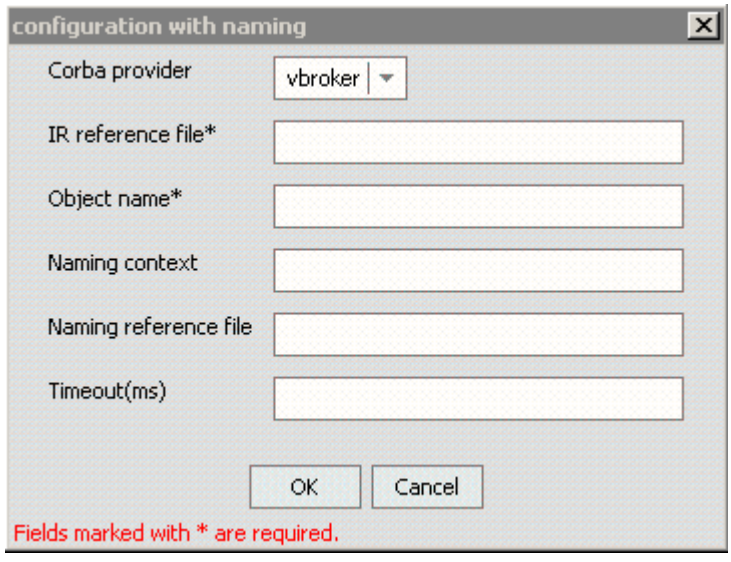
- If you selected a target type of schema exploration only, a dialog box opens containing two parameters and two action buttons (OK and Cancel).



- 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 tsix 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.

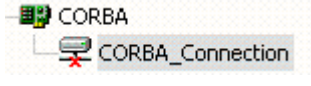
Target Parameter	Description
Object name	Name of the object registered in the Naming Service (for example, bea.clubmed). This parameter, together with Naming context, enables you to specify the location of an object using an indirect Naming Service reference.
Naming context	ORB's name context. This parameter,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

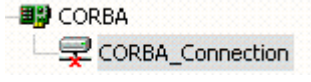
Although you can maintain multiple open connections, iWay Software recommends disconnecting from targets that are not in use.

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.

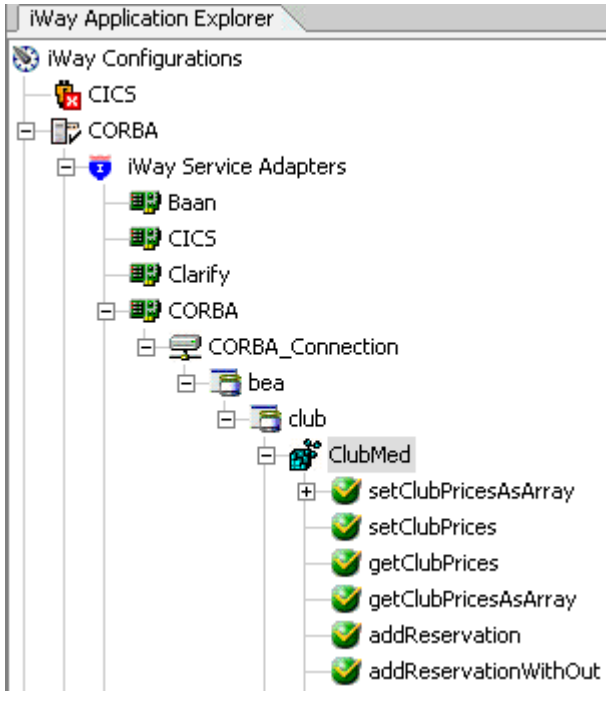
Generating a Request and a Response Schema

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.

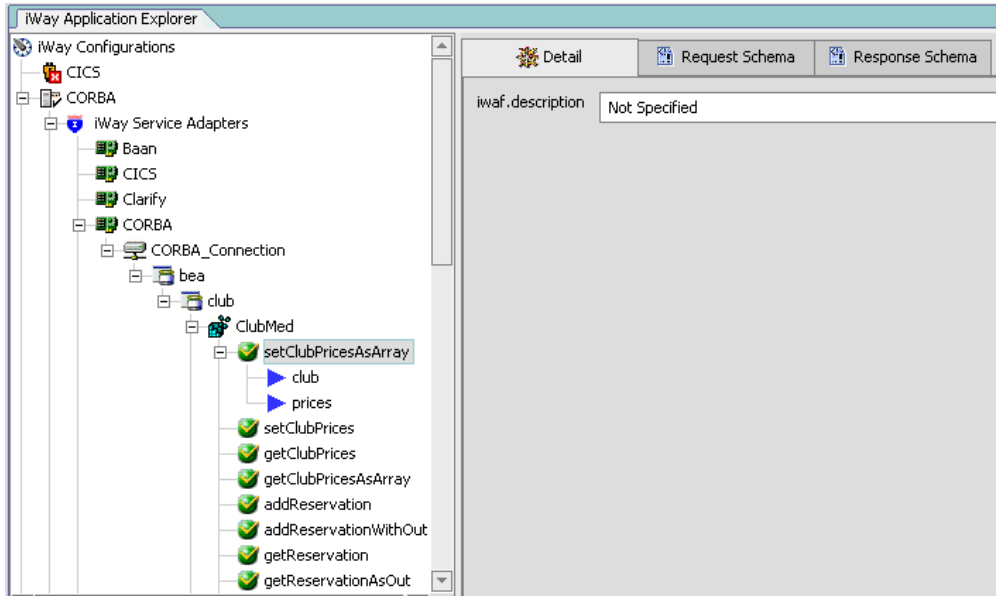
- **Service requests** are requests for action that your application makes to your CORBA system. As part of the definition, the request schema defines the input parameters required by the CORBA system.
- **Service responses** are the way the CORBA system responds to the service request. A service response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

Procedure How to Create a Request Schema and a Response Schema

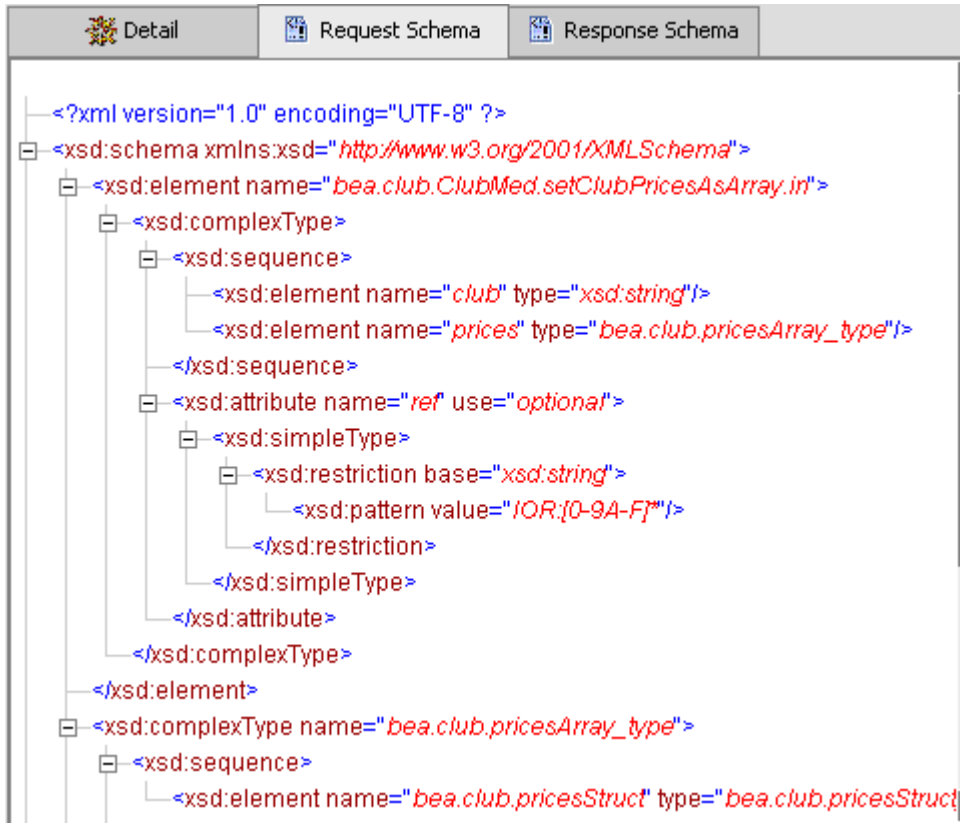
1. If you are not connected to a CORBA target, connect to one, as described in *How to Connect to a Defined Target* on page A-9.
2. Expand the tree under the integration objects to see the items for which you may create schemas.



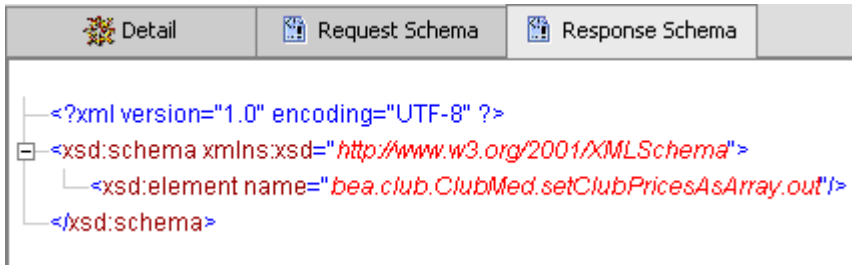
3. Expand and then select the node for which you want to create the schemas. The schemas are automatically created when you select the node. A screen appears in the right pane containing Detail, Request Schema, and Response Schema tabs.



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 JMS 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
  ```

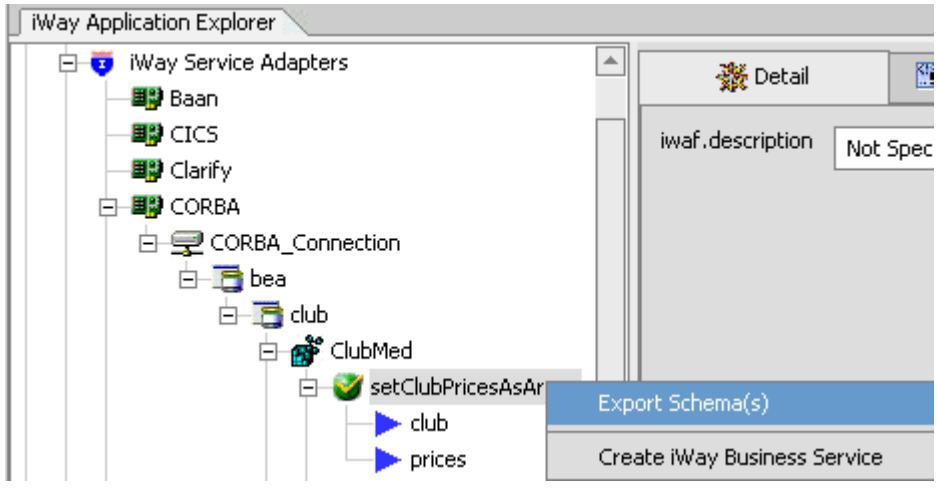
  where:

  CORBA_Connection

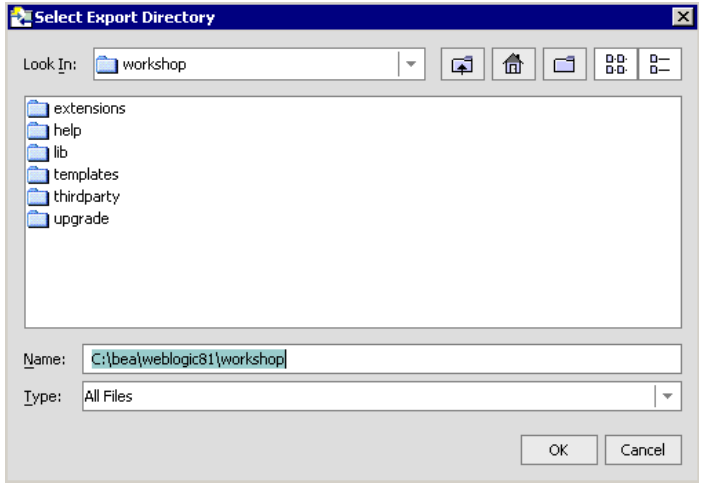
  Is the name of the connection to the CORBA system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

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

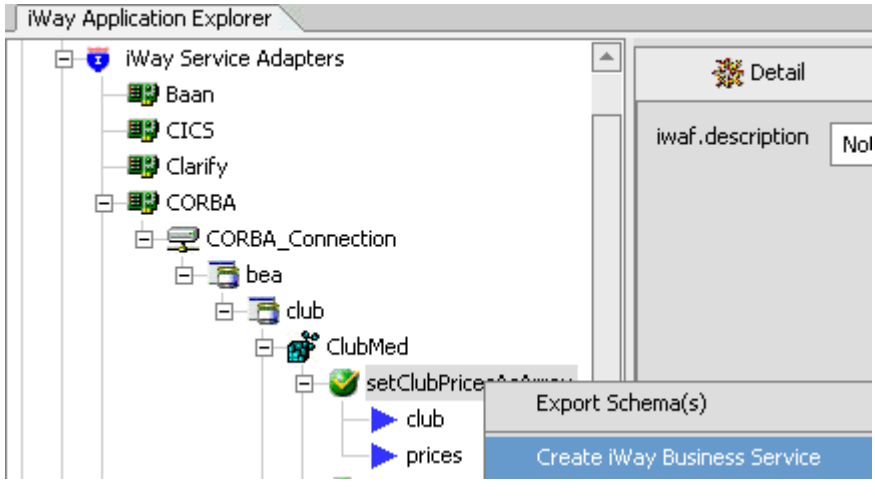
Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the adapter. The iWay Business Services Engine exposes functionality as Web services. It serves as a gateway to heterogeneous back-end applications and databases.

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.

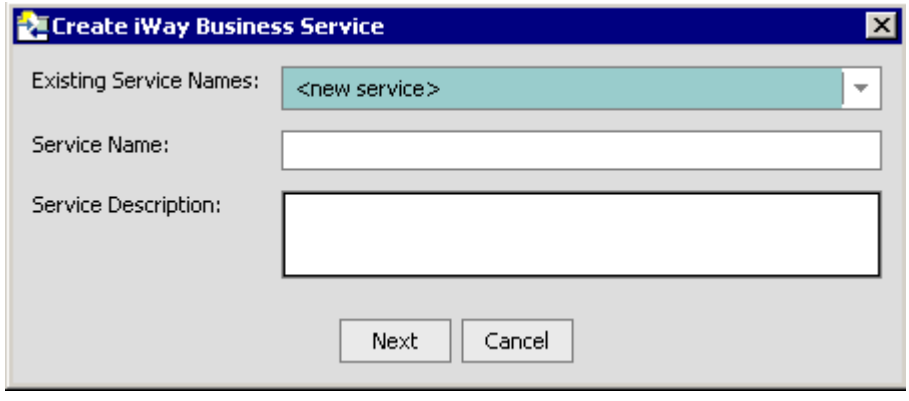
Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation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



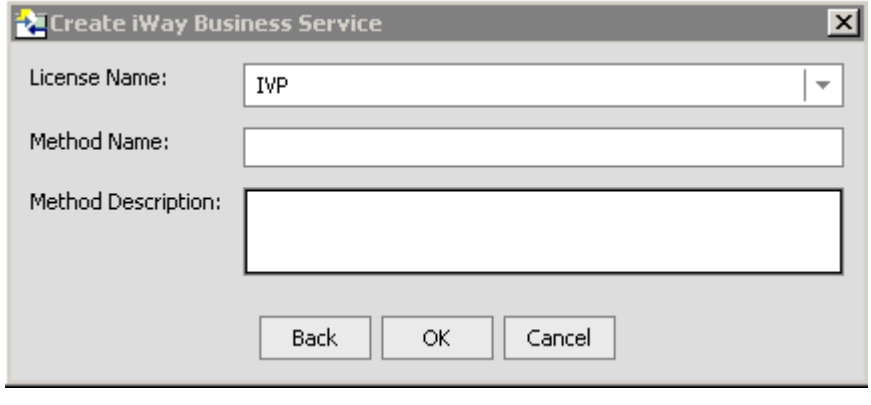
a. Select either a new service or an existing service from the Existing Service Names drop-down list.

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

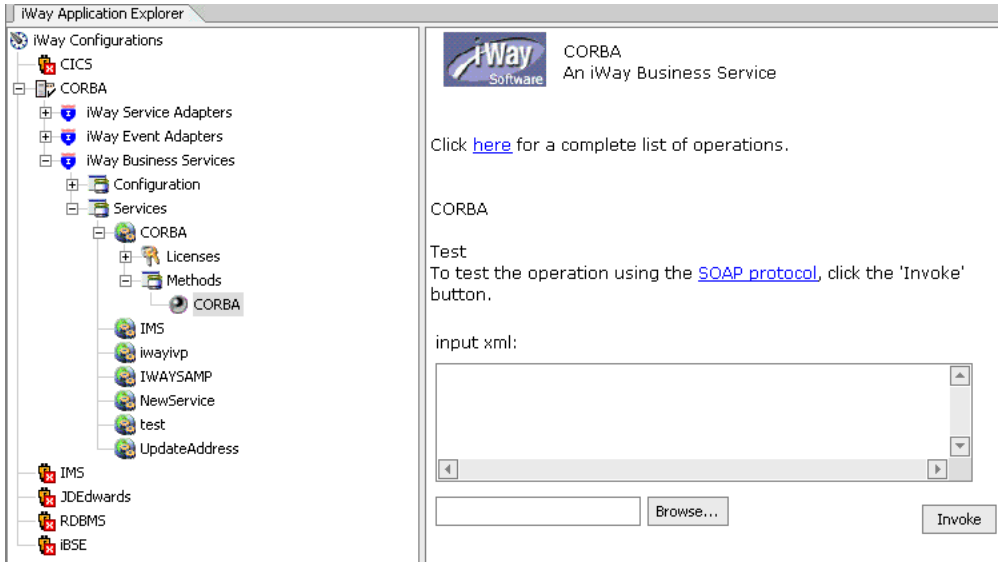
After a business service is created, use the test tool to ensure that it functions properly.

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*.

Credential Mapping

For each SOAP request that is received, iBSE checks to see if a user name and password is included in the SOAP header. If a user name and password is available, iBSE acquires this information and replaces the values retrieved from the repository when pushing the request to the iWay Adapter.

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.

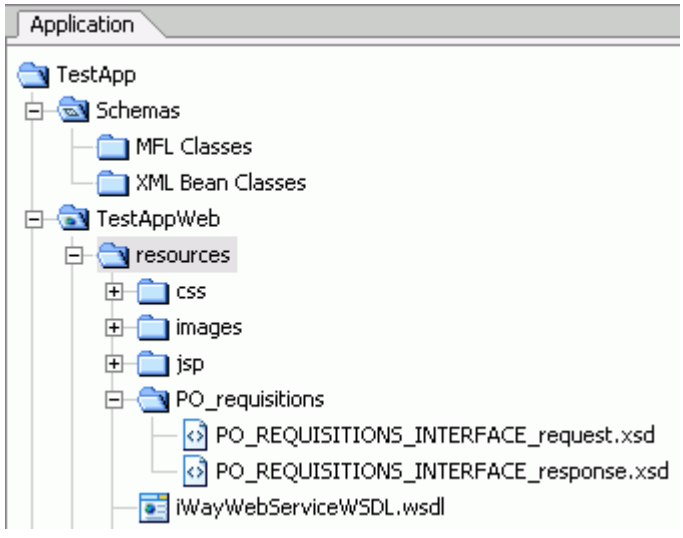
For more information on exporting a WSDL file, see *How to Generate WSDL From a Web Service* on page A-20.

Procedure How to Add a Web Service Control

To add a Web service control:

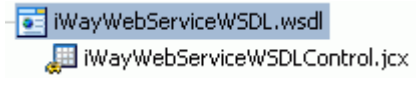
1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

   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.



Extensible CCI Control

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

Overview

The extensible iWay CCI control now offers:

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

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

```
sapComDocumentSapRfcFunctions.BAPIMATERIALGETDETAILResponseDocument
getDetail(sapComDocumentSapRfcFunctions.BAPIMATERIALGETDETAILDocument
aRequest) throws java.lang.Exception
```

In addition, the extensible CCI control now generates a JCX file to which you can add your own methods.

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 will take care of the casting for you. There is no longer a need 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;
```

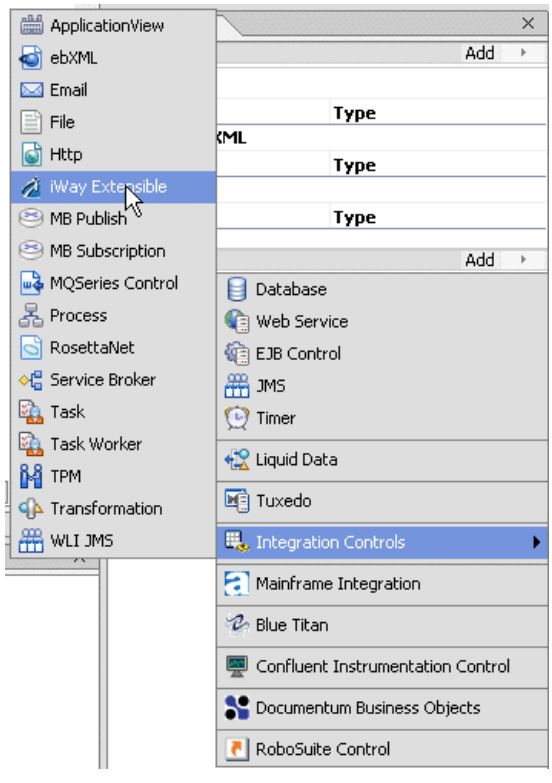
you will be calling:

```
BAPIMATERIALGETDETAILResponseDocument
getDetail(BAPIMATERIALGETDETAILDocument aRequest) throws
java.lang.Exception;
```

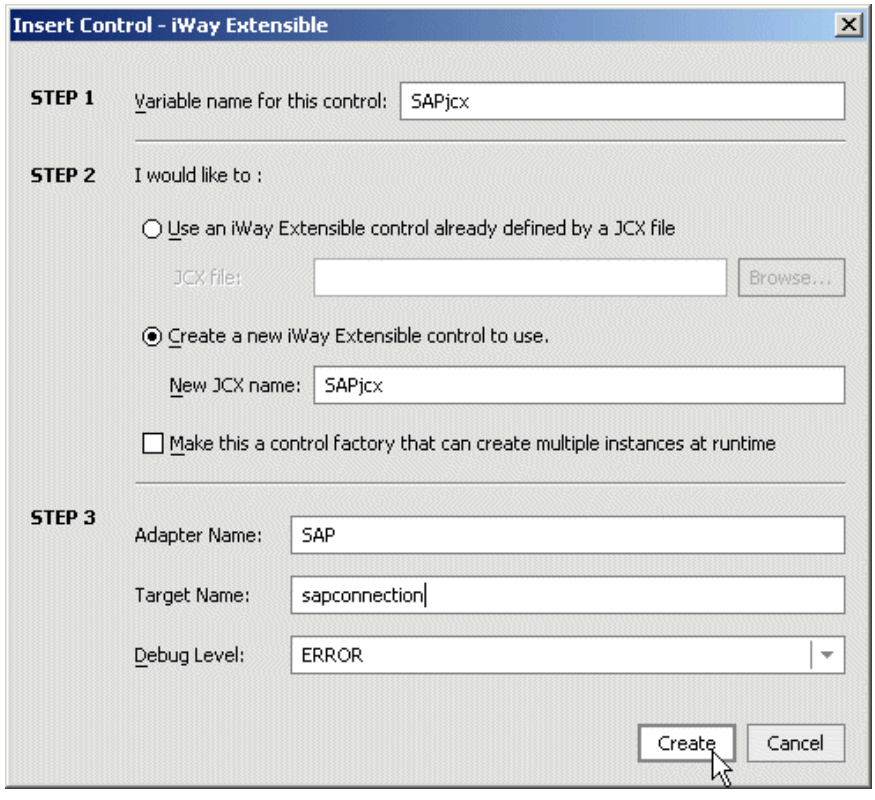
Example Defining a Control Using the Extensible CCI Control

The following sample JCX demonstrates how to define a control that uses the SAP BAPI_MATERIAL_GET_DATA using the extensible CCI control in BEA WebLogic Workshop.

1. Start BEA WebLogic Workshop and create a new project.
2. Click *Integration Controls* and select *iWay Extensible*.



The Insert Control - iWay Extensible dialog box opens.



3. Perform the following steps:

a. Provide a variable name for this control.

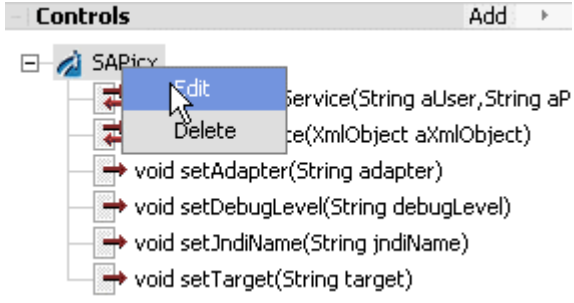
b. Click *Create a new iWay Extensible control* to use and provide a new JCX name.

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

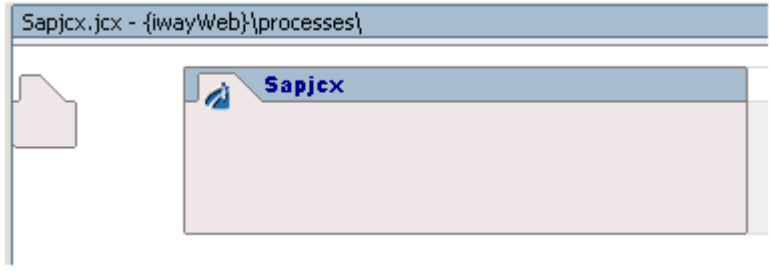
4. Click *Create*.

A new JCX file is created.

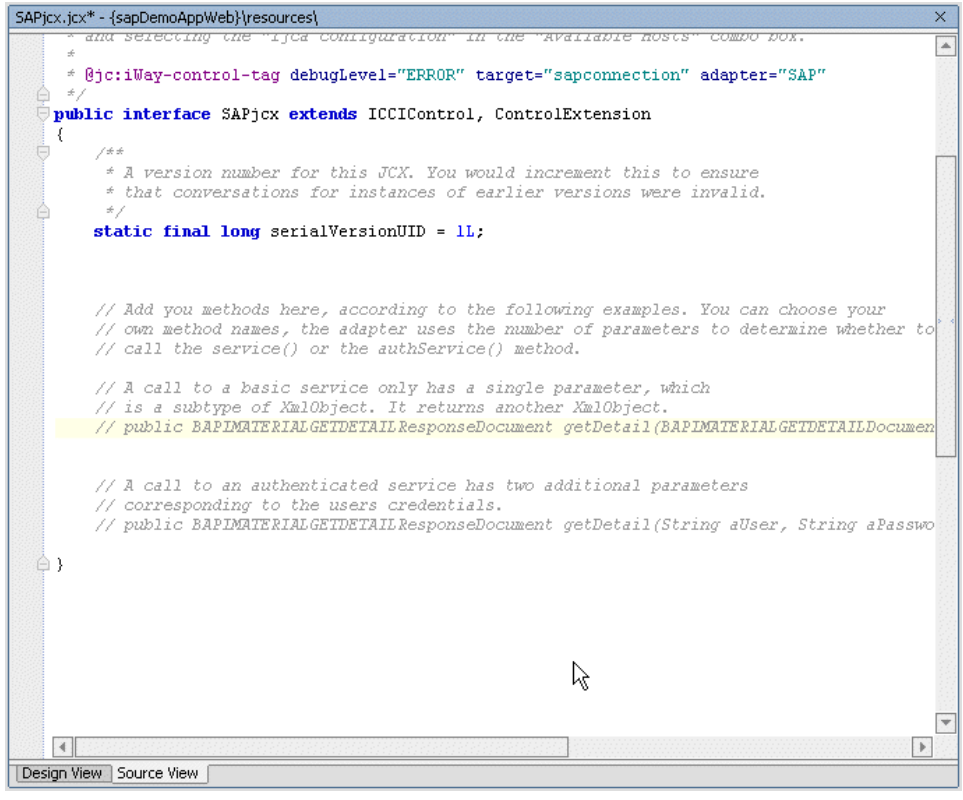
To edit an existing control, right click the control and selec t *Edit*.



The Design view is displayed.



5. Click *Source View*.



You can add your own methods that call the adapter's services.

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's 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's Interoperable Naming Service (INS), which supports the binding of names to object references (and looking up object references using these names). It also allows 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's Interface Repository (IFR) component. It allows applications to request, inspect, and modify IDL type information dynamically. For example, the IR enables applications to find out which operations an object supports. Some ORBs may also need the IR to find out whether a given object's type is a subtype of another, but 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 Build and Run the JacORB Request Broker

1. Unzip JacORB1_4_beta4-full.zip.

   This file is found in BEA_CORBA_SAMPLES.zip in the adapter's 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 created in step 1.

   This file is found in BEA_CORBA_SAMPLES.zip in the adapter's 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 file's Initial References Configuration section, uncomment this 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. Build the JacORB application. In a new DOS command window, execute the following:

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

   b. *drive&path*:JacORB1_4_beta4\bea\club\ant

   where:

   drive&path

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

8. Start the Interface Repository service. In a new DOS command window, execute the following:

   a. JacORB1_4_beta4\bea\setenv-sample.bat

   b. 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. Start the name service. In a new DOS command window, execute the following:

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

where:

drive&path

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

b. 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. Start the Java interpreter explicitly. In a new DOS command window, execute the following:

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

where:

drive&path

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

b. 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 being 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 (`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, VisiBroker's IOR file is not automatically output to a file. Modify your IR startup procedure to automatically output the startup 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, perform the following steps:

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 should appear 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 *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.
- Easily embeddable code.

APPENDIX C

Supported IDL Types

Topics:

- IDL Base Types Supported
- IDL Constructed Types Supported

This section lists the Interface Definition Langauge (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:

- boolen
- 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**