iWay

iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic User's Guide Version 5 Release 5

February 11, 2005
DN3501332.0205

Preface

This documentation describes how to configure and use the iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic.

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 Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic	Introduces the adapter environment.
2	Configuring the Adapter	Describes how to configure a connection to the adapter.
3	Designing the Adapter	Describes how to create transactions for the adapter. It also provides information on how to create iWay Business Services, which expose functionality as Web services.
4	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
5	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.
6	Using WebLogic Workshop	Describes how to create and access a Web service using WebLogic Workshop
A	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and Web services for IMS/TM.

Chapter/Appendix		Contents
B	Running the Adapter Using LU6.2 Communication	Contains technical information that you can use as a guide to ensure LU6.2 communication to the IMS/TM region.
C	Sample Requests, Schemas, and Cobol File Descriptions	Provides documents and schemas for the sample transaction, PART and the Cobol descriptions used as input for the sample IMS/TM transactions.
D	Debugging and Troubleshooting	Includes tips and techniques for debugging the adapter.

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

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

Customer Support

Do you have questions about the iWay Transaction Adapter for IMS/TM for Java Technologies 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 Transaction Adapter for IMS/TM for Java Technologies 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 number (*xxxx.xx*) when you call.

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

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

Help Us to Serve You Better

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

The following tables list the specifications our consultants require.

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

Container Version	

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

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

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

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

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

Version	**Vendor**

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

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

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

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

File	Error/Problem Details or Information
XML schema	
XML instances	
Other input documents (transformation)	
Error screen shots	
Error output files	
Trace and log files	
Log transaction	

User Feedback

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

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

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

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

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

Contents

CHAPTER 1

Introducing the iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic

Topics:

- Overview of the Adapter
- The iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic
- Deployment Information for the Adapter

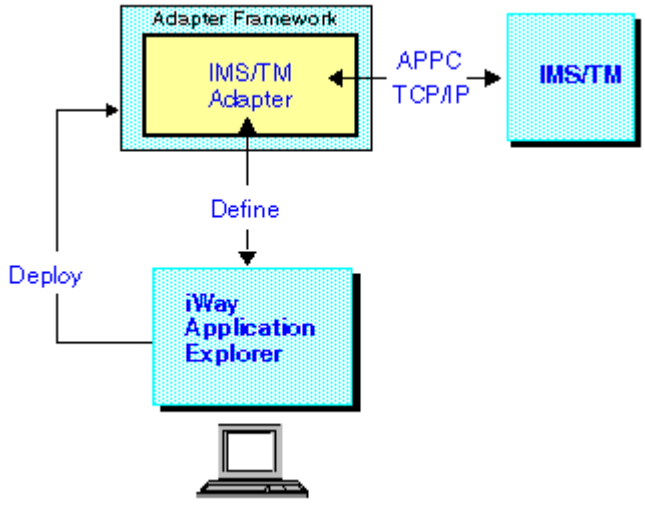
This section describes the iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic. The adapter supports automatic transaction invocation, message transformation, and error recovery. The adapter enables applications to call IMS/TM transactions and to work with the native features and syntax of IMS/TM.

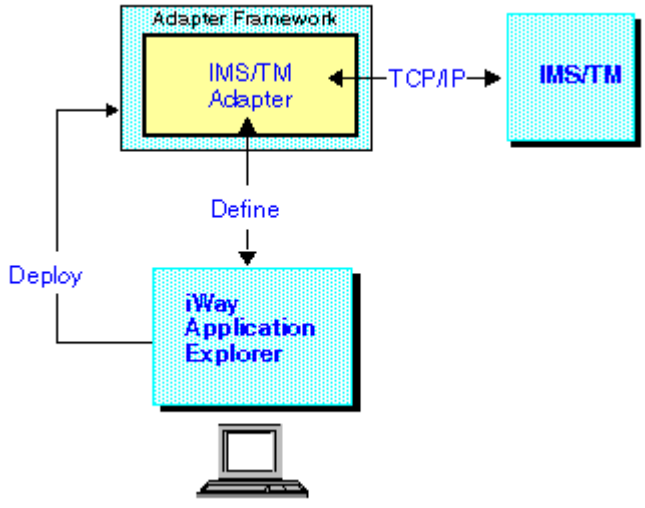
Overview of the Adapter

The adapter enables you to execute IMS/TM transactions. The advantages of the adapter include the following:

- No modification required to existing IMS/TM transactions.
- No installation of new code required on IMS/TM.
- Adapter processing performed off of the mainframe.
- Configuration by metadata—no coding required.
- Support for older versions of IMS/TM.
- Support for IMS/TM transactions.

The following diagram illustrates the framework for executing IMS/TM programs with the iWay Application Explorer and the iWay Transaction Adapter for IMS/TM.



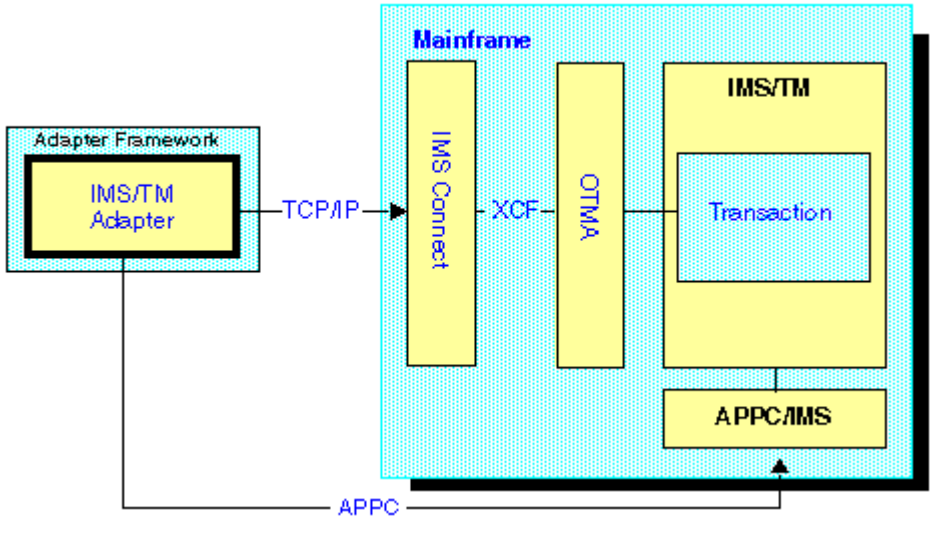


The following bidirectional scenarios are supported by the adapter:

- IMS/TM services
- IMS/TM events - Contact Customer Support Services

The iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic

The following diagram illustrates the adapter connection to the mainframe via TCP/IP.



The adapter is the component that connects to IMS/TM. It is hosted in a container that supports events. The adapter enables the following functions:

- Connecting to IMS/TM.
- Executing IMS/TM transactions.
- Mapping XML messages to and from IMS/TM data structures.

The adapter enables you to invoke an IMS/TM transaction by sending a request and retrieving the response.

The adapter uses IMS Connect, available with IMS Version 7.1 and higher, to execute IMS/TM transactions from a TCP/IP client.

At design time, you describe the request and response messages by mapping them to Cobol File Descriptions. You communicate with IMS/TM through either TCP/IP or APPC.

IMS/TM Transactions

The two kinds of IMS/TM transactions are:

- Non-conversational.
- Conversational: when a user interacts with a terminal screen (3270).

Because the adapter can execute only non-conversational transactions, this distinction is important.

To execute 3270 conversational programs, you require a screen scraper such as the iWay Adapter for 3270. For many years IMS/TM applications were structured so that the business processing, as opposed to the screen dialogue, was in non-conversational transactions. Therefore, in many cases, executing a non-conversational transaction is recommended for application integration.

Software Requirements for the Adapter

The following are the software requirements for the adapter:

- OS/390 v2.6 or higher, or z/OS.

For TCP/IP:

- IMS Version 5 or higher.
- IMS Connect and Open Transaction Manager Access (OTMA) installed and configured on the remote IMS/TM system.

For APPC Communications:

- IMS Version 5 or higher.
- IMS/APPC.

- LU6.2 sessions created within the IMS/TM region.
- When the adapter is running on a non-mainframe platform, APPC communications requires that an SNA server (or the AIX/UNIX equivalent) be available to connect to APPC/IMS.

Deployment Information for the Adapter

The iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic works with iWay Application Explorer in conjunction with one of the following components:

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

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

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

Deployment Information Roadmap

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

Deployed Component	For more information, see
iWay Application Explorer	Chapters 2 and 3 and Appendix A of this guide *iWay Installation and Configuration for BEA WebLogic* *iWay Servlet Application Explorer for BEA WebLogic User's Guide*
Integration Business Services Engine (iBSE)	*iWay Installation and Configuration for BEA WebLogic*

Deployed Component	For more information, see
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	*iWay Connector for JCA for BEA WebLogic User's Guide* *iWay Installation and Configuration for BEA WebLogic*

The Integration Business Services Engine

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

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

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

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

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

The iWay Enterprise Connector for J2EE Connector Architecture

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

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

CHAPTER 2

Configuring the Adapter

Topics:

- Starting Servlet Application Explorer
- Configuring a Connection to IMS/TM
- Managing a Connection to IMS/TM

At design time, you use Application Explorer to create the configuration and metadata the adapter requires at run time. This section describes how to configure a connection to IMS/TM.

Starting Servlet Application Explorer

The server must be started where Servlet 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 Exploreris installed.

   port

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

   Application Explorer opens.

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

   - Service Adapters, where you create and manage connections to IMS/TM.
   - Event Adapters, where you configure event listening.
   - iWay Business Services, where you create and view business services.

   The left pane of the window contains an expandable list of adapter nodes (based on the 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 IMS/TM.

Configuring a Connection to IMS/TM

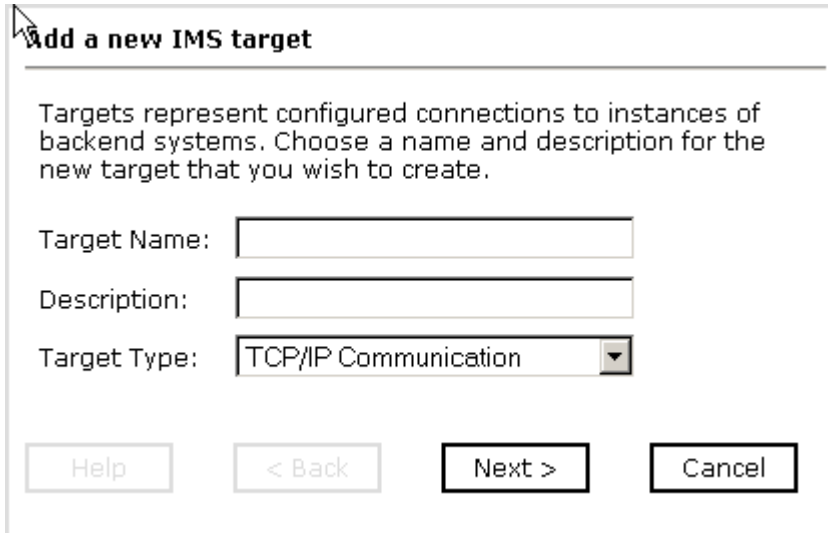
To access an adapter, you must configure a connection to that adapter. After the connection is created, it is automatically saved. You must establish a connection to the system every time you start Application Explorer or after disconnecting.

You can connect to IMS/TM using TCP/IP Communication or APPC Communication logon options.

Procedure: How to Configure a Connection to IMS/TM

1. In the left pane, expand the *Service Adapters* node in Application Explorer.
2. Click the *IMS* node.
3. In the right pane, move your pointer over *Operations* and select *Define a new target*.

   The Add a new IMS target dialog box opens in the right pane, as shown in the following image, where you enter the Target Name and Description, and select the Target Type from a drop-down list.



   a. In the Target Name field, type a name for the connection (for example, TCPIP_Connection).

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

   b. In the Description field, type a description for the target name you just created (for example, Connection using TCPIP option).

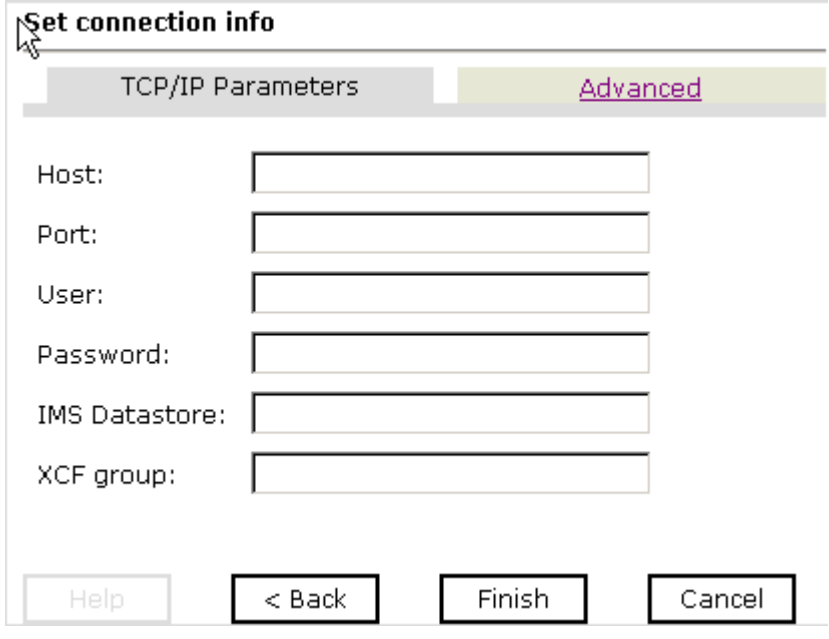
   c. In the Target Type drop-down list, select the type of target (for example, *TCP/IP Communication*).

4. Click *Next*.

The connection name is verified for the system. If you entered an invalid instance name, a new Input dialog box opens and prompts you for an instance name again.

- If you selected TCP/IP Connection as the type of target, proceed to Step 5.
- If you selected APPC Connection as the type of target, proceed to Step 7.

The following image shows the Set connection info dialog box which opens to the TCP/IP Parameters tab window containing six text fields for entry. The following step describes the entries needed.



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

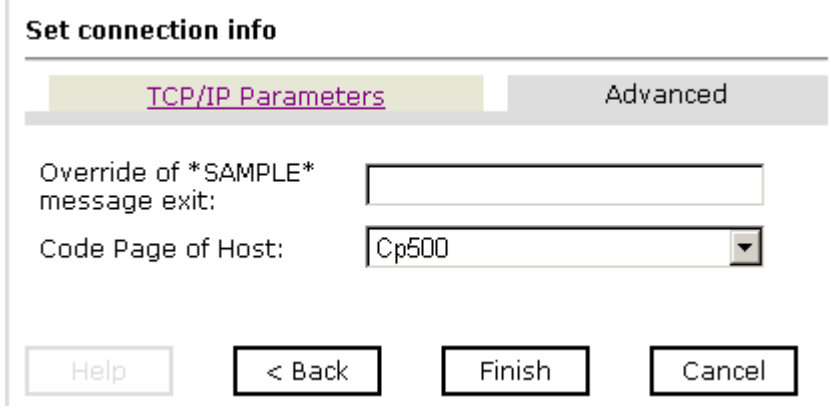
5. Enter the TCP/IP parameters to configure a new connection to IMS/TM.

You can obtain this information from the IMS/TM Systems Administrator. This information should be the same for all programs in a single IMS/TM system.

The following table lists and describes the TCP/IP parameters:

Parameter	Description
Host	Host name, or IP address, for the computer where IMS/TM is running.
Port	Port number on which IMS Connect is listening.
User	Valid user ID for IMS/TM.
Password	Valid password associated with the IMS/TM user ID.
IMS Datastore	Name of the IMS/TM datastore.
XCF Group	Name of the XCF group.

6. If you click the Advanced tab, the following parameters appear, as shown in the following image.



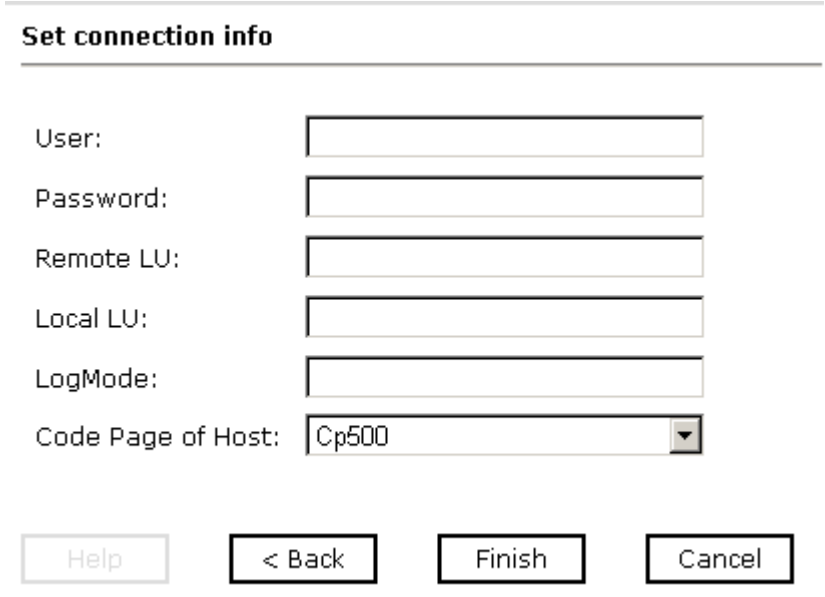
a. Enter the Advanced parameters. The following table lists and describes the Advanced parameters:

Parameter	Description
Override of *SAMPLE* message exit	Specify a message exit if you want to override the sample message exit.
Code Page of Host	Select the codepage from the drop-down menu. Cp500 is the default value.

b. Proceed to Step 8.

7. Enter the APPC parameters to configure a new connection to IMS/TM

The Set connection info dialog box for APPC opens in the right pane. The following image shows the Set connection info dialog box containing six text fields for entry. The following step describes the entries needed.



You can obtain this information from the IMS/TM Systems Administrator. This information should be the same for all programs in a single IMS/TM system.

The following table lists and describes the APPC parameters:

Parameter	Description
User ID	Valid user ID for IMS/TM.
Password	Valid password associated with the IMS/TM user ID.
Remote LU	LU of APPC/IMS.
Local LU	LU of the SNA access point to which you have access, for example, SNA server.
logMode	Log mode of APPC/IMS.
Code Page of Host	Select the codepage from the drop-down menu. Cp500 is the default value.

For more information, see Appendix B, *Running the Adapter Using LU6.2 Communication*.

8. Click *Finish*.

   The newly created connection, TCPIP_Connection, appears as a node under the IMS service adapter. The configuration information is stored in the repository for the configuration you defined at installation time.

Connecting to TCP/IP

You can connect to a defined IMS/TM target (for example, TCPIP_Connection) from Application Explorer.

Procedure: How to Connect to a Defined IMS/TM Target

1. Expand the *Service Adapters* node.
2. Expand the *IMS* node.
3. Click the target name (for example, TCPIP_Connection) under the IMS node.
4. Move your pointer over *Operations* and select *Connect*.

   The Connect to TCPIP_Connection dialog box opens, populated with values you entered for the connection parameters.

5. Verify your connection parameters. If required, provide the password and then click *OK*.

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

Managing a Connection to IMS/TM

To manage IMS/TM connections, you can:

- Disconnect from a connection that is not currently in use.

  Although you can maintain multiple open connections to different transaction processing systems, it is recommended to disconnect from connections not in use.

- Edit a connection to change its properties.
- Delete a connection that is no longer required.

Procedure: How to Disconnect From a Connection to IMS/TM

1. Expand the *Service Adapters* node.
2. Expand the *IMS* node.
3. Click the connection, for example, *TCPIP_Connection*, move your pointer over *Operations*, and select *Disconnect*.

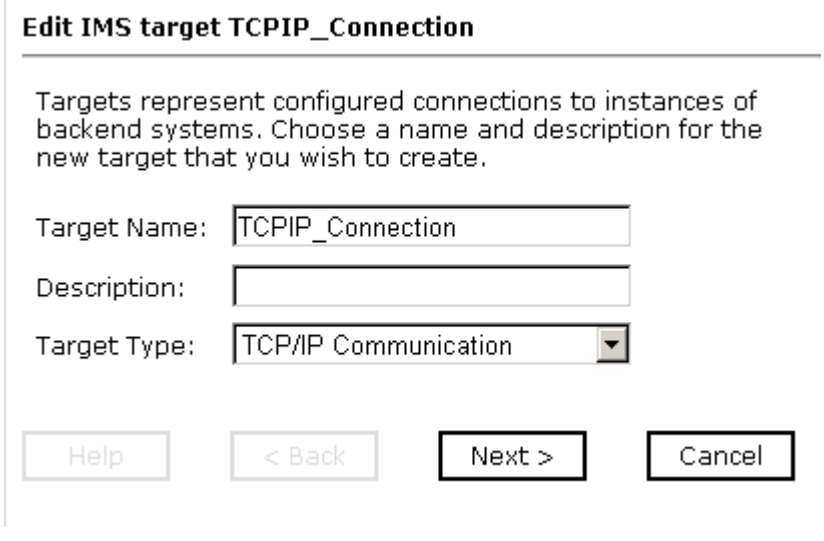
   Disconnecting from IMS drops the connection with IMS, but the node remains.

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

Procedure: How to Edit a Connection to IMS/TM

1. In the left pane of Application Explorer, expand the *Service Adapters* node.
2. Expand the *IMS* node and select the defined target (for example, *TCPIP_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 needed and then click *Next*.

   The Set connection info dialog box opens in the right pane containing the TCP/IP Parameters and Advanced tabs.

5. Modify the information as needed and then click *Finish*.

Procedure: How to Delete a Connection to IMS/TM

1. Expand the *Service Adapters* node.
2. Expand the *IMS* node.
3. Click the connection, for example, *TCPIP_Connection*, move your pointer over *Operations*, and select *Delete*.

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

4. Click *OK*.

   The node disappears from the list of available connections.

CHAPTER 3

Designing the Adapter

Topics:

- Creating an Adapter Transaction
- Creating Schemas for an Adapter Transaction
- Understanding iWay Business Services

Application Explorer is a Web application running within a servlet container and is accessible through a browser. It enables the adapter to explore metadata and create XML schemas.

The following topics describe how to use Application Explorer to create IMS/TM transactions and generate request and response XML schemas for new or existing transactions. These schemas are used to represent a transaction for integration with external systems.

In addition, this section provides information on how to use the generated schemas to create iWay Business Services, which expose functionality as Web services.

Creating an Adapter Transaction

After you create a connection to IMS/TM, you can add adapter transactions using Application Explorer. A single IMS/TM connection may be associated with multiple transactions. Each transaction represents one service offered by IMS/TM and consists of a program and its metadata.

A generic transaction is automatically added and represents IMS/TM services whose data will not be mapped to XML. You can use a generic transaction for transactions that accept no input and for transactions that return no output or when it is acceptable to return a non-formatted answer set.

For example, the IMS transaction PART connects to IMS/TM and returns PART information on successful adapter installation and configuration. One request and response schema is applicable for this generic transaction. The request schema for the generic transaction is in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

Using the generic transaction, the XML request document that is received must include the name of the program to be called in the <Transaction> element. The payload to be sent as the IMS segment must be in the <message> tag, which can be a maximum of 32,500 bytes.

The generic response schema is constructed from the data received from IMS/TM. If the <message> element has more than 80 bytes, the received IMS segment is split into 80-byte messages. Illegal XML characters ('<', '/', and '&') are converted to XML entities. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<IMS>
<Transaction tpname="PART" noreply="NO">
<message>
<message>*</message>
</message>
</Transaction>
</IMS>
```

For transactions that require input and output and a formatted response, which is usually the case, you must add your own adapter transactions, as described in *Create an Adapter Transaction* on page 3-4. XML request messages must specify the transaction to use in the location attribute of the <Transaction> tag. For example, if you create an IMS/TM transaction called PART, the location is "IMS/Transactions/PART".

To view a sample generic request or response schema or for information about specifying a transaction to use in the location attribute of the <Transaction> tag, see Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

Cobol Descriptions for Defining IMS/TM Transaction Output

The adapter uses Cobol descriptions to properly create an XML structure of the message that is returned from the IMS/TM transaction. For transactions that return one or more messages (for example, a message that has its own message layout), the adapter transforms the message into the proper structure based on Cobol descriptions.

Because Message Format Services (MFS) is not used for IMS/TM transactions that the adapter can execute, such as PART, you must create or use existing Cobol descriptions.

If the application that executes the adapter requires proper formatting of each returned message from the transaction:

- You must examine all possible outputs of the transaction so you can create Cobol descriptions correctly.

  As a reference, it helps to use the Cobol description of the output message or the MFS message output descriptor area for the transaction to create the Cobol descriptions.

- You must supply Cobol descriptions of each type of output using Application Explorer when configuring the transaction.

  Note: For generic transactions where the format of the output is of no consequence, you are not required to supply Cobol descriptions.

  For the PART transaction, you require three Cobol descriptions. A RECTYPE field in each description is used by the adapter to determine which of the three messages is being returned by the adapter.

Sample Transaction PART

IBM supplies the PART transaction with an IMS system. This document uses the PART transaction for illustration purposes and as a reference for the adapter. The PART transaction accepts an input part number with a length of 17 characters or less. Based on what is passed to the PART transaction, an answer set is returned from the DP21PART database.

- If a part number is passed and found in IMS, the transaction returns detail information.
- If an asterisk ('*') is passed in the request, the transaction returns all part numbers in the database with their descriptions.
- If the part number is not found in IMS, the transaction, "PART NOT FOUND," is returned.

The PART transaction is an example of a transaction that returns multiple answer sets. Three different answer sets are returned based on what is passed in the request. The adapter enables you to create a response schema that contains different possible return messages.

Sample request documents, including sample response schemas for the PART transaction, are in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*. You specify the output as explained in *Creating an Adapter Transaction* on page 3-2. You must know the field in the Cobol description that can be used as a record type and the value of that field. You specify the value of the field to create the appropriate response schema.

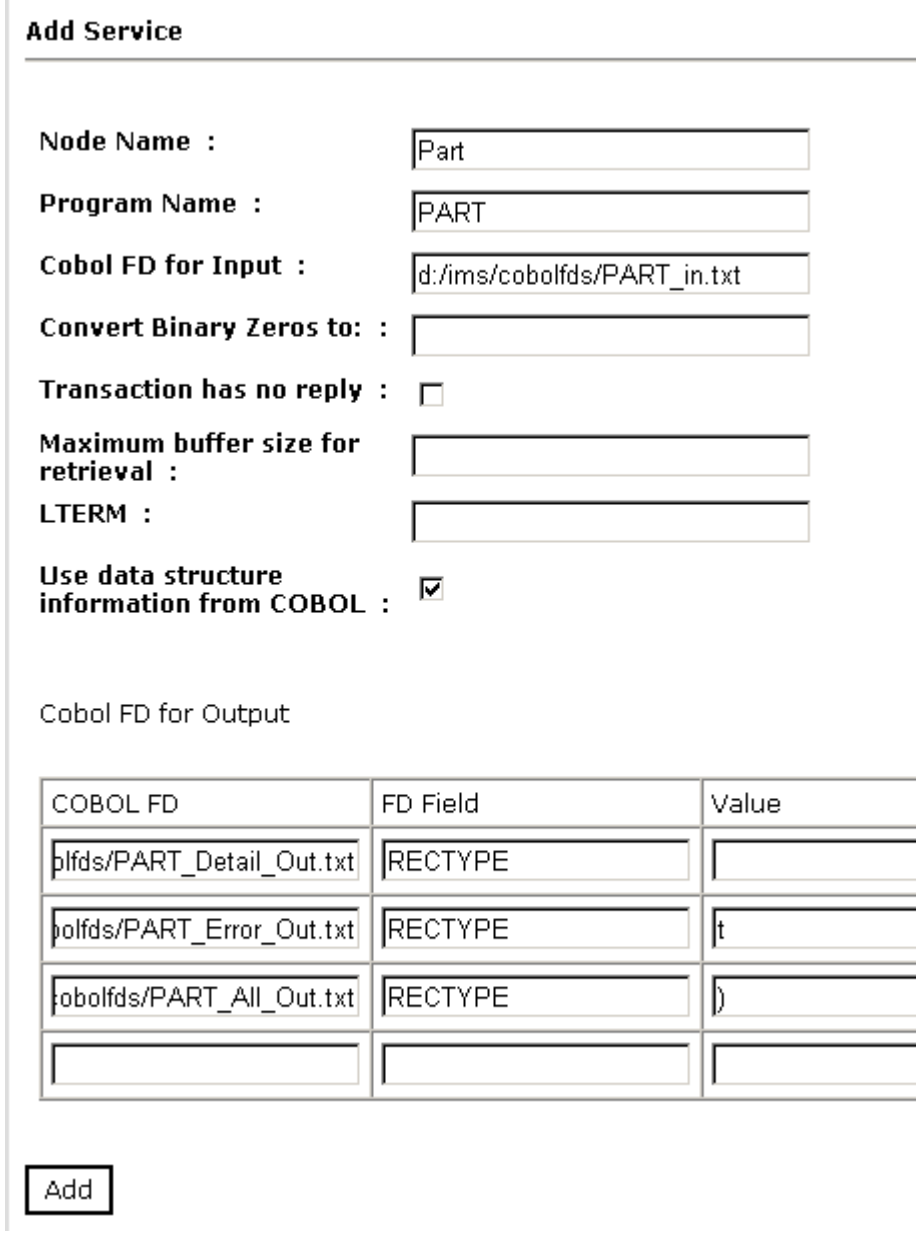
The previous is also true for events to determine the layout returned from IMS when you configure an IMS event. If you must configure an IMS event, contact Customer Support Services.

Procedure: How to Create an Adapter Transaction

1. Expand the *Service Adapters* node.
2. Expand the *IMS* node and connect to an IMS target (for example, TCPIP_Connection).
3. Expand the node to which you connected.

   The Transaction node appears under the connected node.
4. Click *Transactions*, move your pointer over *Operations*, and select *Add Service*.

The Add Service dialog box opens in the right pane. The following image shows the Add Service pane containing parameters that will enable you to map the Cobol descriptions for the IMS/TM transaction.



5. To map the Cobol descriptions for the IMS/TM transaction, type values for the parameters, as defined in the following table.

Field	Description
Name	Name to describe the adapter transaction you create. The name, for example, IMS_Transaction, appears under the Transactions node for the current connection. The name to use in the <Transaction location="..."> attribute.
Transaction Name	Name of the transaction to be called in IMS/TM, for example, PART. The PART input FD is shown in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.
Cobol FD for Input	Location of the Cobol description that describes the input parameters of the IMS transaction to execute. Converted by the adapter to an XML schema that the adapter uses to map from XML to the format required by IMS/TM at run time.
Convert non printable char to	Character to convert binary zeros to in output.
Transaction has no reply	Select this check box when you do not want to wait for a response from the program.
Maximum buffer size for retrieval	Maximum buffer size for an answer set.
LTerm	Logical terminal. Blank is the default unless specified by the user.
Use data structure information from Cobol	When this parameter is checked, the adapter creates request and response schemas that reflect Cobol group levels (for example, 05, 10, 20, and so on). The Cobol grouping will be reflected in the XML request and response schemas. You must check this parameter when Cobol input or output descriptions contain the Cobol OCCURS statement. When this parameter is checked and the program COMMAREA contains an OCCURS statement, the output Cobol definition must also contain an OCCURS statement. Do not “flatten out” the Cobol output description, since the adapter relies on the number of OCCURS when formulating the program’s output.

<table>
<tr><th>Field</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">Cobol FD for Output - contains the following columns:
• COBOL FD
• FD Field
• Value</td><td colspan="2">Path that corresponds to the message you want returned from the IMS/TM transaction.

If the transaction can return multiple types of messages for each output Cobol description, enter the Cobol description field and value to determine the schema that is created and used for a particular message.

Application Explorer creates the schema to use for a particular message based on the contents of a field that is returned. For example, a program called PART populates the field called RECTYPE. Depending on program logic, Application Explorer creates the correct response schema.</td></tr>
<tr><td>Value in RECTYPE Field

space ‘ ’

parenthesis ‘)’

‘t’</td><td>Cobol Description

PART_Detail_Out

PART_All_Out

PART_Error_Out</td></tr>
<tr><td colspan="2">The PART_Detail_Out, PART_All_Out, and PART_Error_Out Cobol descriptions appear in Appendix C, Sample Requests, Schemas, and Cobol File Descriptions.</td></tr>
</table>

Note: You must transfer the Cobol descriptions to a location accessible to Application Explorer. For the correct Cobol descriptions to use for the program, contact your IMS/TM Administrator or application developer.

6. Click *Add*.

   The new IMS/TM transaction is added, for example, IMS_Transaction under the Transactions node for the current connection.

Cobol Descriptions for Input and Output Communications

IMS Connect Considerations: Every time IMS Connect returns data (that is, an IMS Segment) it prefixes the data with LL, which represents the length of the returned data, and ZZ, which is a reserved field of binary zeros. The LL is called the IRM_LEN and is the length of the IRM structure, while ZZ is called IRM_RSV and is a reserved field. The format of the IRM structure is LLZZDATA, where LL is the total length of this segment, ZZ are binary zeros, and DATA is the IMS trancode followed by transaction data.

Important: The adapter does not require these prefixes in the Cobol input and output definitions. Only describe the data portion in the Cobol input and output definitions.

Example: Cobol Descriptions

You must use the following syntax for binary, packed, and float fields for the Cobol descriptions for the adapter transaction input and output formats.

For a binary field:

```
05  BINARY-FIELD          PIC S9(n) COMP.
```

For a packed-decimal field:

```
05  PACKED-FIELD          PIC S9(n) COMP-3.
```

For a single-float field:

```
05  FLOAT-SINGLE          COMP-1.
```

For a double-float field:

```
05  FLOAT-DOUBLE          COMP-2.
```

Note: Underscores are not supported in Cobol descriptions.

Creating Schemas for an Adapter Transaction

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 the adapter is used with an iBSE configuration, Application Explorer stores the schemas under a \schemas subdirectory of the iWay home directory, for example,

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

where:

TCPIP_Connection

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

When the adapter is used with a JCA configuration, Application Explorer stores the schemas under a \schemas subdirectory of the iWay home directory, for example,

```
C:\Program Files\iWay55\config\base\schemas\IMS\TCPIP_Connection
```

where:

TCPIP_Connection

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

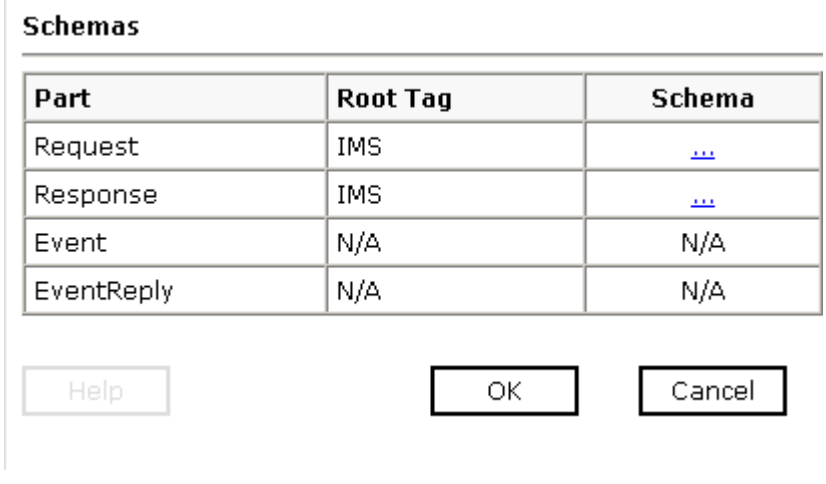
Procedure: How to Create Schemas for an Adapter Transaction

The following procedure describes how to create request and response schemas for an adapter transaction.

1. In the left pane, select the transaction for which you want to generate schemas.
2. In the right pane, move the pointer over Operations and select *Generate Schema*.

   Note: The adapter generates the schemas for the selected Cobol descriptions and associates them with the transaction. The schemas generated for the sample Cobol descriptions appear inAppendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

   The Schemas table appears in the right pane. The following image shows the Schemas window which is a table of four rows and three columns, Part, Root Tag and Schema. Only the first two rows are applicable, Request and Response under the Part column that have IMS as the value located under the Root Tag column and clickable ellipsis under the Schema column.



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

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

3. Click *OK* to exit the Schemas window.

Understanding iWay Business Services

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

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

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

Creating a Web Service

After you connect to your application system and create an XML schema for a transaction, you can create a Web service. The following procedure describes how to create a Web service using Application Explorer.

Procedure: How to Create a Web Service

1. Click the *Service Adapters* tab.

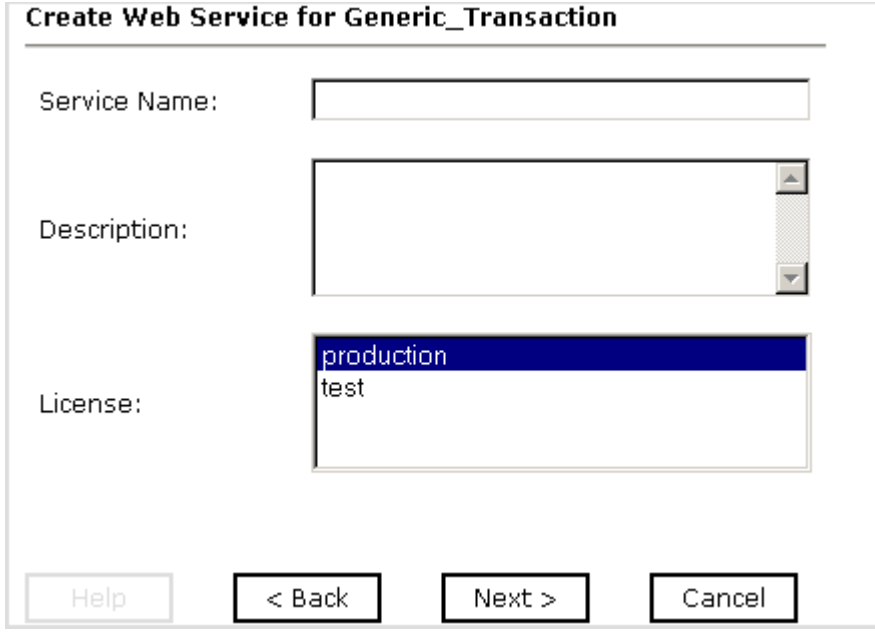
   The Service Adapters window opens.

2. In the left pane, expand the *IMS* node.
3. Connect to an IMS target (for example, TCPIP_Connection).
4. Expand the node to which you connected.

   The Transaction node appears under the connected node.

5. Click *Transactions* and then select the transaction for which you want to create a Web service.
6. In the right pane, move your cursor over *Operations* and select *Create Integration Business Service*.

The Create Web Service dialog box opens in the right pane. The following image shows the Create Web Service for Generic_Transaction dialog box which contains three fields for entry.



7. Enter information that is specific to the Web service you are defining.

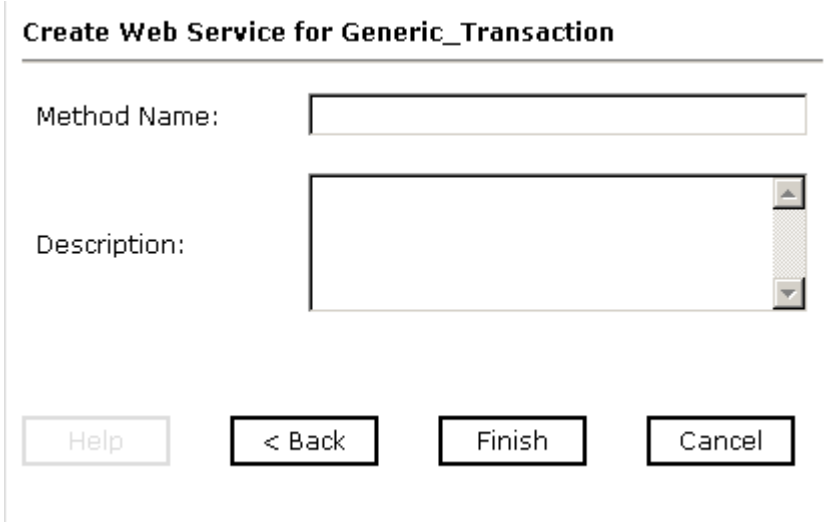
a. In the Service Name field, type a descriptive name for the Web service.

b. In the Description field, type a brief description for the Web service (optional).

c. 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.

8. Click *Next*.

Another dialog box with the Method Name and Description fields opens. The following image shows another Create Web Service for Generic_Transaction dialog box that have two fields for entry.



9. Enter information that is specific to the method you are defining.

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

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

10. Click *Finish*.

The Integration Business Services Engine tab opens. The Web service is created and published to the Integration Business Services Engine. Application Explorer displays the newly created Web Service under the iWay Business Services folder.

Testing the Web 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 the Web Service

1. If you are not on the iWay Business Services tab of Application Explorer, click the tab to access business services.
2. If it is not expanded, expand the 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.
6. In the input xml field, either enter a sample XML document that queries the service, for example,

```
<?xml version="1.0" encoding="UTF-8" ?>
<IMS>
 <Transaction location="/IMS/Transaction/PART">
   <message>AN960C10</message>
 </Transaction>
</IMS>
```

   or browse to the location of an XML instance and click *Upload*.
7. Click *Invoke*.

   The result appears 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

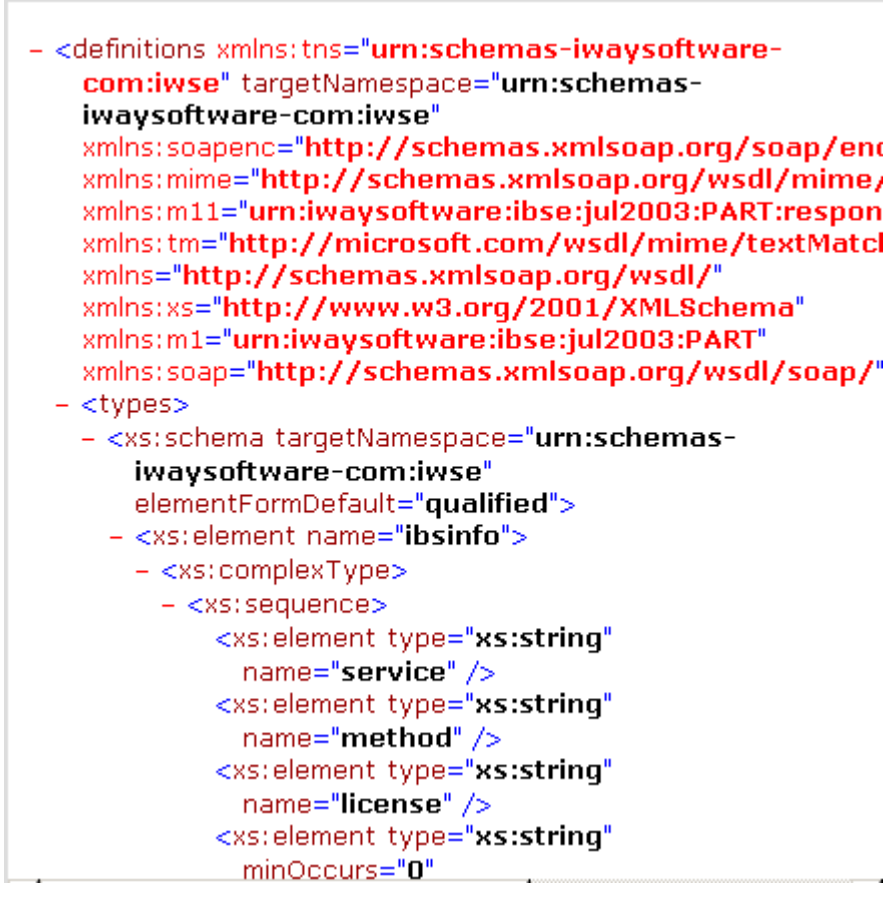
1. If you are not already in the iWay Business Services tab, click the tab to access business services.
2. In the left pane, expand the list of services to display the Web service for which you want to generate WSDL.
3. Click the Web service.

   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*.

Example: Viewing WSDL Generated from a Web Service

After generating a WSDL file from the PART.ibs serialized object, the PART.wsdl file looks similar to the following image which is a sample XML file.



```
- <definitions xmlns:tns="urn:schemas-iwaysoftware-
    com:iwse" targetNamespace="urn:schemas-
    iwaysoftware-com:iwse"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/enc
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/
    xmlns:m11="urn:iwaysoftware:ibse:jul2003:PART:respon
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatcl
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:m1="urn:iwaysoftware:ibse:jul2003:PART"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  - <types>
    - <xs:schema targetNamespace="urn:schemas-
        iwaysoftware-com:iwse"
        elementFormDefault="qualified">
      - <xs:element name="ibsinfo">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element type="xs:string"
                name="service" />
              <xs:element type="xs:string"
                name="method" />
              <xs:element type="xs:string"
                name="license" />
              <xs:element type="xs:string"
                minOccurs="0"
```

Identity Propagation

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

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

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

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

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

CHAPTER 4

Using Web Services Policy-Based Security

Topics:

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

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

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

iWay Business Services Policy-Based Security

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

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

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

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

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

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

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

Configuring iWay Business Services Policy-Based Security

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

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *Create an Execution Policy* on page 4-8.

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

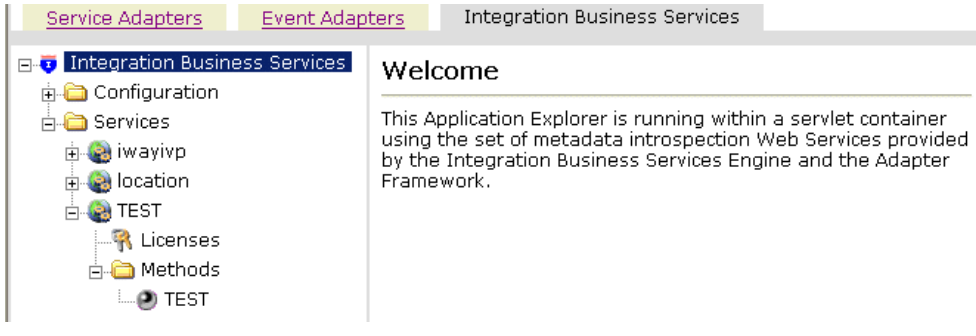
Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

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

To create a user to associate with a policy:

1. Open *Servlet Application Explorer.*

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



   a. Click the *iWay Business Services* tab.

   b. Expand the *Configuration* node.

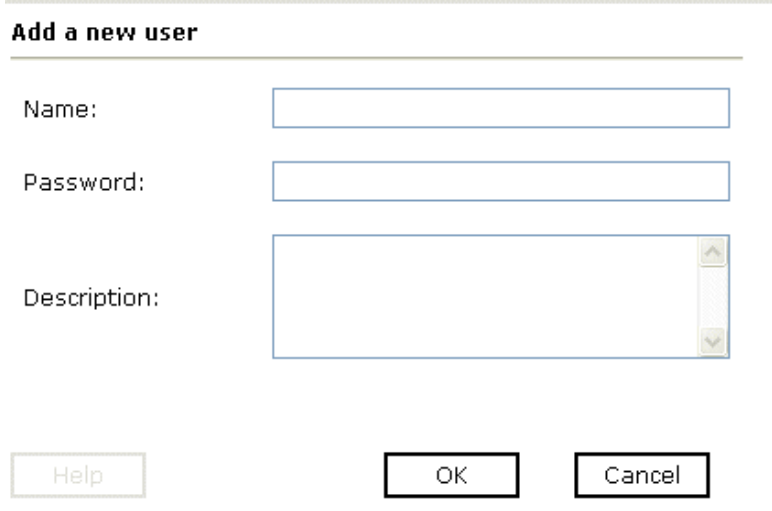
   c. Expand the *Security* node.

   d. Expand the *Users and Groups* node.

e. Select *Users*.

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

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



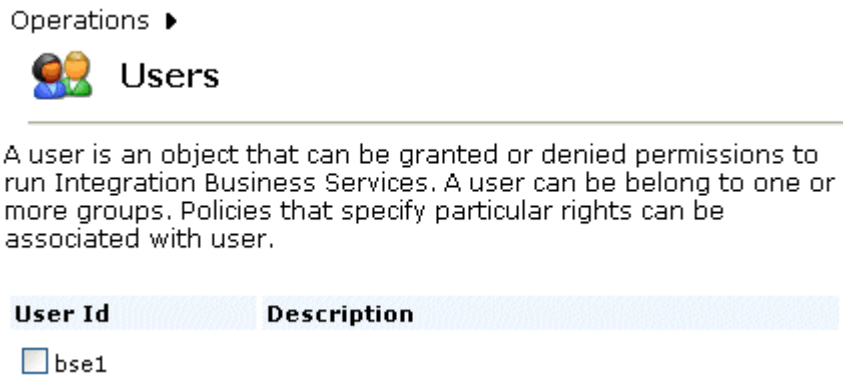
a. In the Name field, type a user ID.

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

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

3. Click *OK*.

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

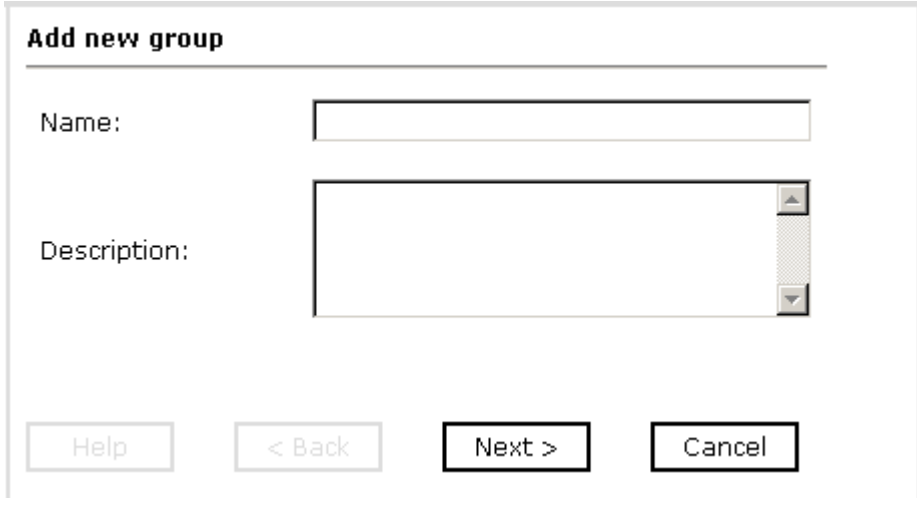


Procedure: How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

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

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

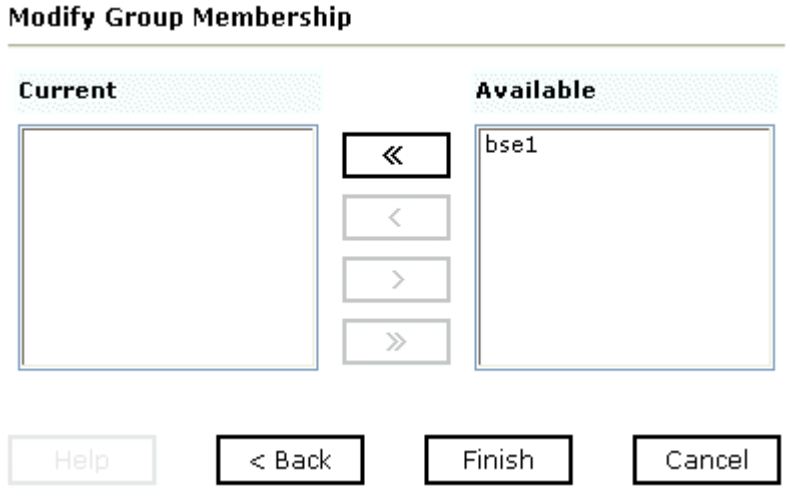


a. In the Name field, type a a name for the group.

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

3. Click *Next*.

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

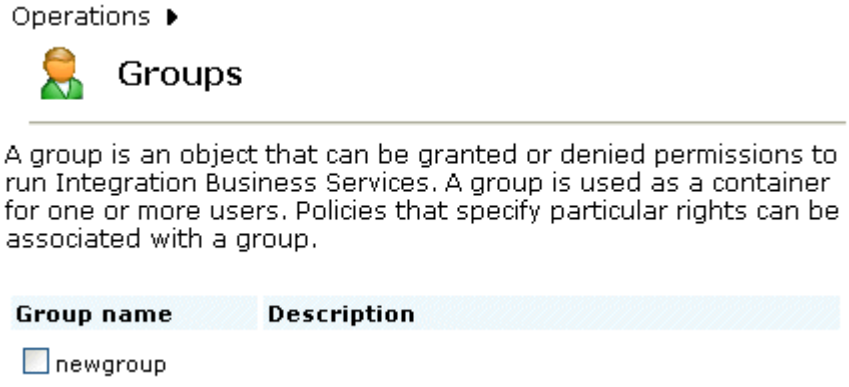


You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

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

The new group is added.

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

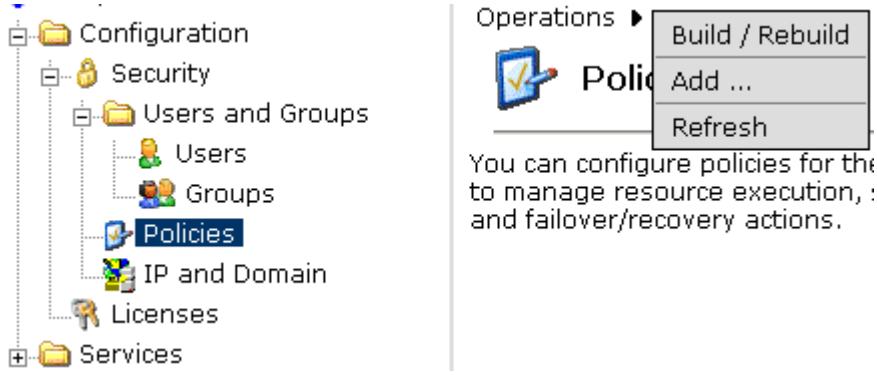


Procedure: How to Create an Execution Policy

To create an execution policy:

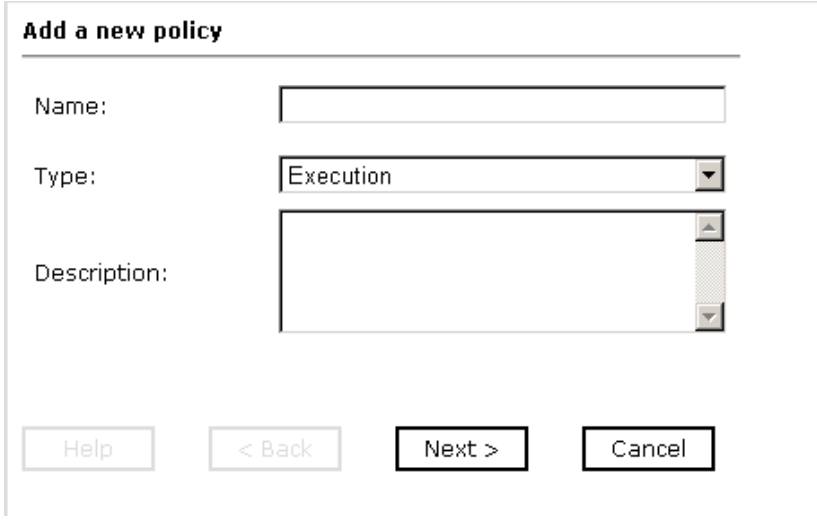
1. Open *Servlet Application Explorer*.
   a. Click the *iWay Business Services* tab.
   b. Expand the *Configuration* node.
   c. Select *Policies*.

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



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

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



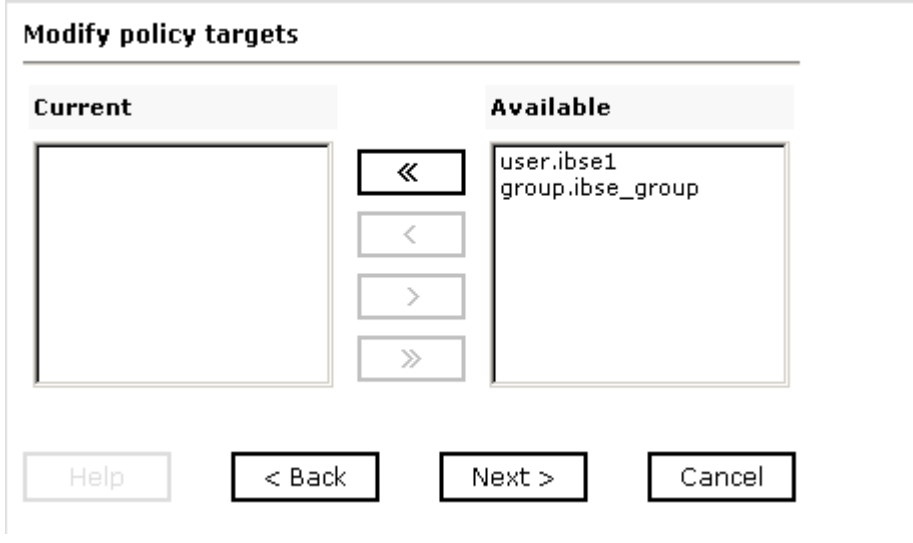
   a. In the Name field, type a a name for the policy.

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

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

3. Click *Next*.

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

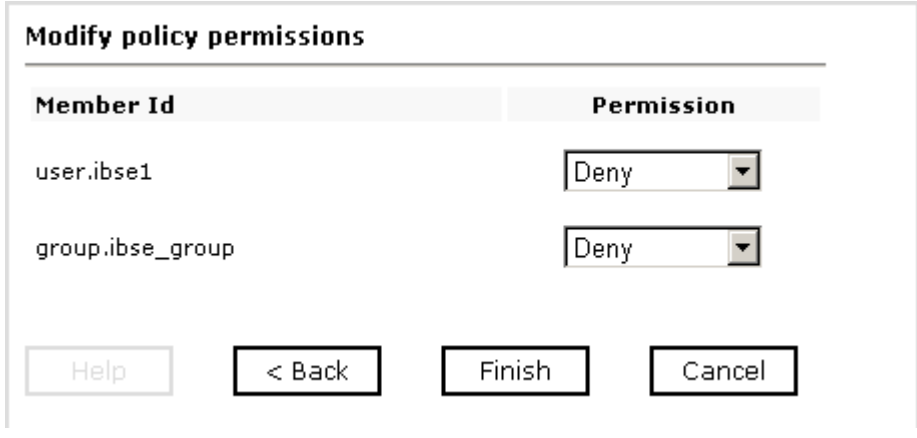


4. Select a minimum of one user or group from the Available pane.

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

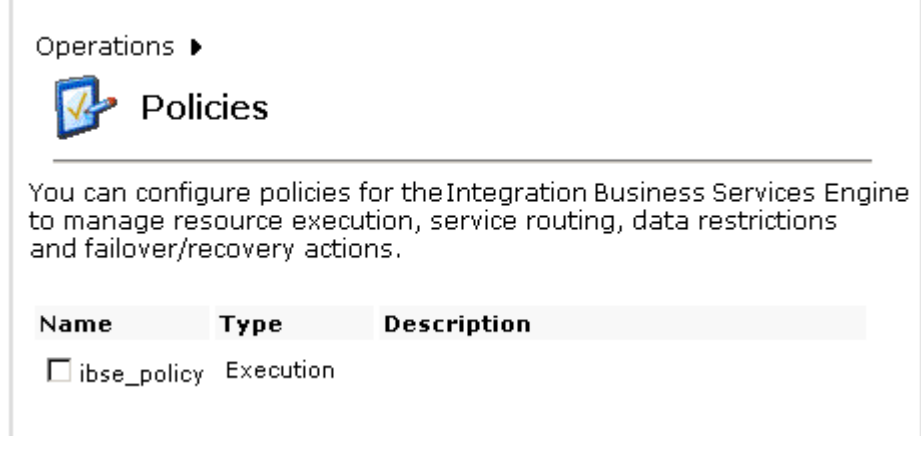
5. Click *Next*.

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



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

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

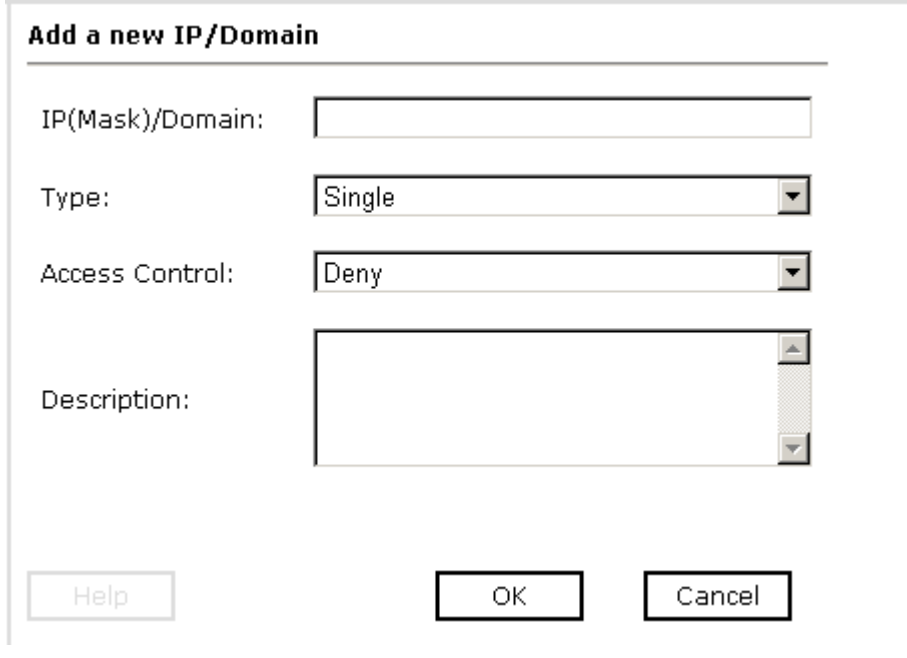


Procedure: How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

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

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



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

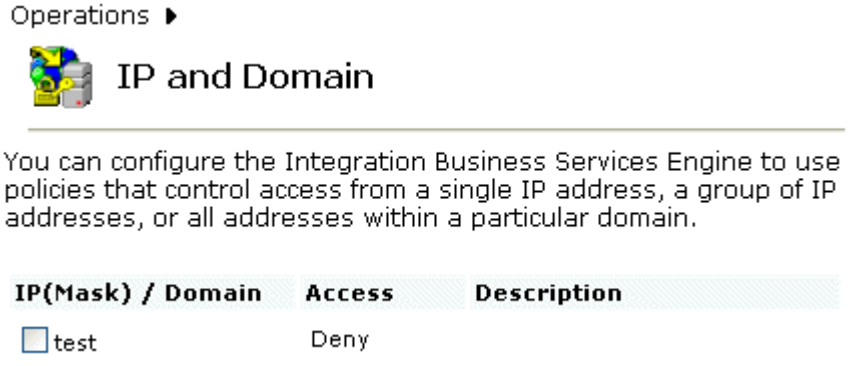
      If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

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

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

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

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



CHAPTER 5

Management and Monitoring

Topics:

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

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

Managing and Monitoring Services and Events Using iBSE

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

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

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

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

   where:

   localhost

   Is the machine where the application server is running.

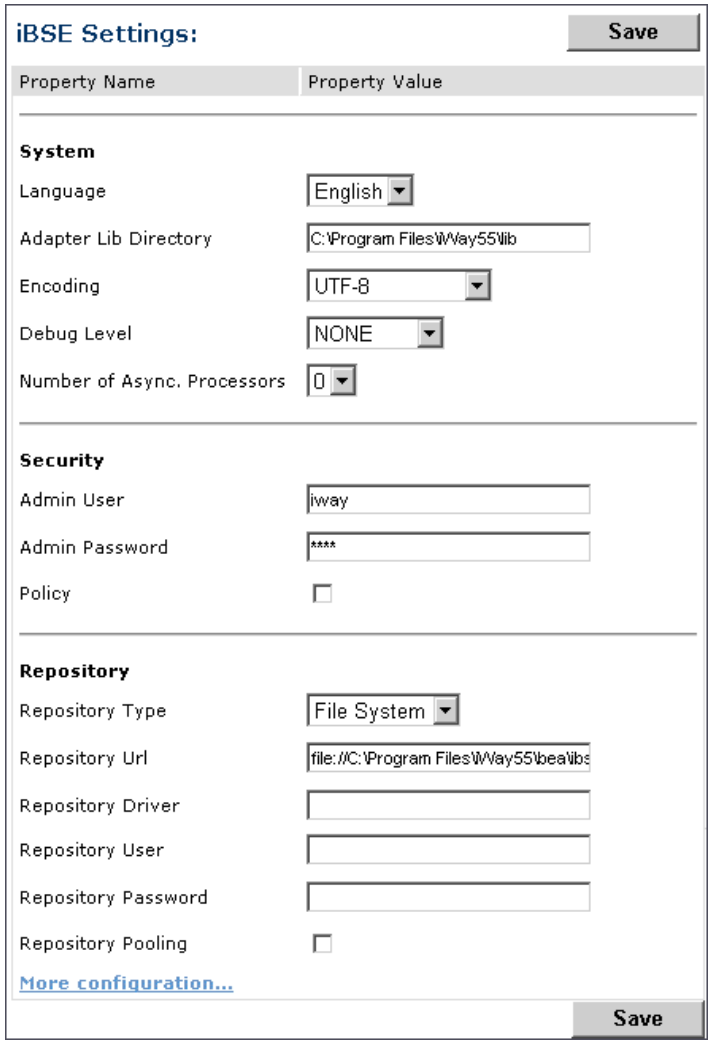
   port

   Is the HTTP port for the application server.

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

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

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



3. Click *More configuration*.

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

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

where:

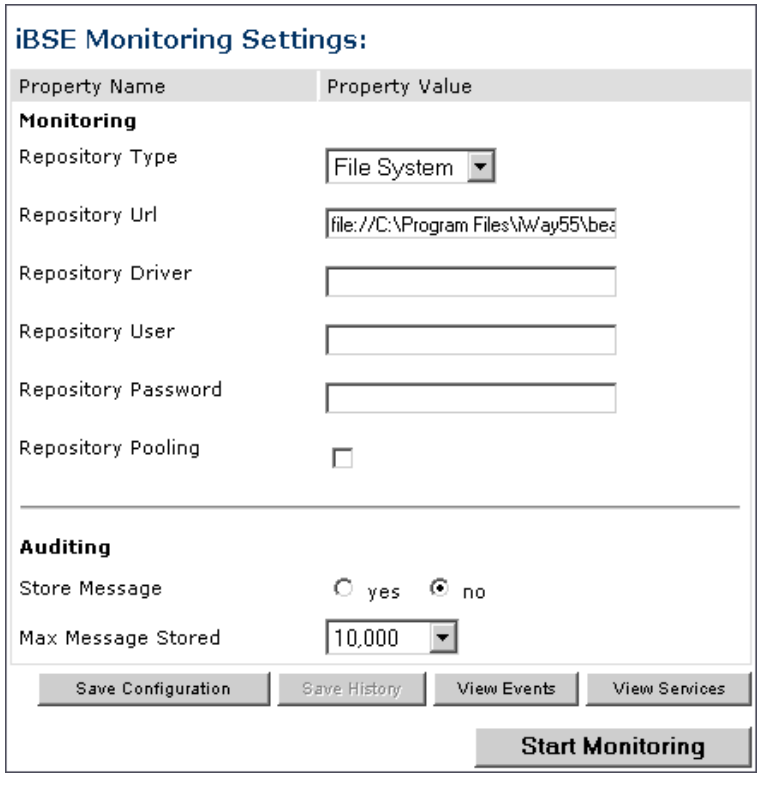
localhost

Is the machine where the application server is running.

port

Is the HTTP port for the application server.

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



a. In the Monitoring pane, from the Repository Type drop-down list, select the type of repository you are using.

b. To connect to the database in the Repository Url field, type a JDBC URL.

c. To connect to the database in the Repository Driver field, type a JDBC Class.

d. To access the monitoring repository database, type a user ID and password.

e. To enable pooling, click the *Repository Pooling* check box.

f. In the Auditing pane, select *yes* if you want to store messages.

This option is disabled by default.

Note: You must start and then, stop monitoring to enable this option.

g. Select the maximum number of messages you want to store.

By default, 10,000 is selected.

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

h. Click *Save Configuration*.

4. Click *Start Monitoring*.

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

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

Procedure: How to Monitor Services

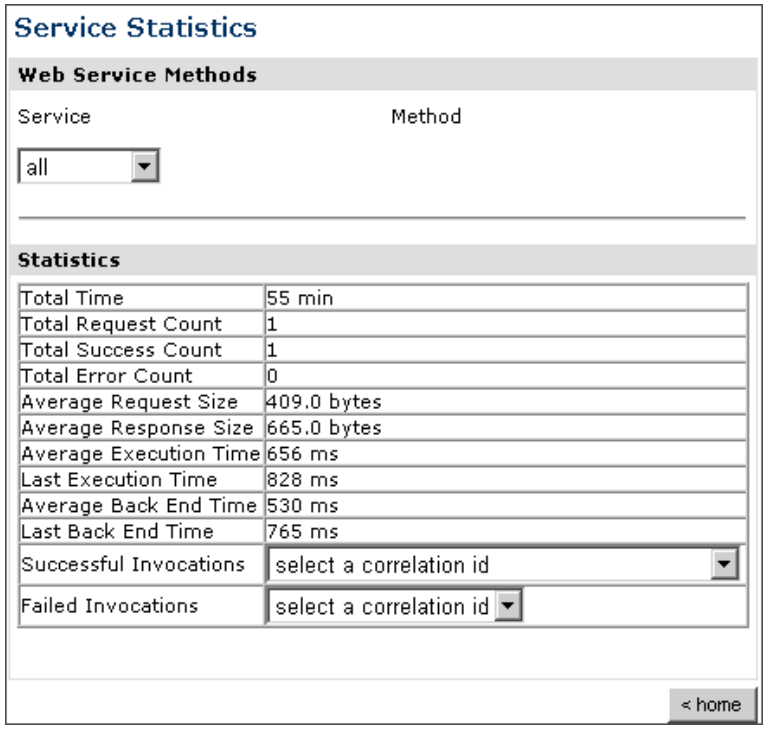
To monitor services:

1. Ensure that your BEA WebLogic Server is started.

2. From the iBSE Monitoring Settings window, click *Start Monitoring*.

3. Click *View Services*.

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



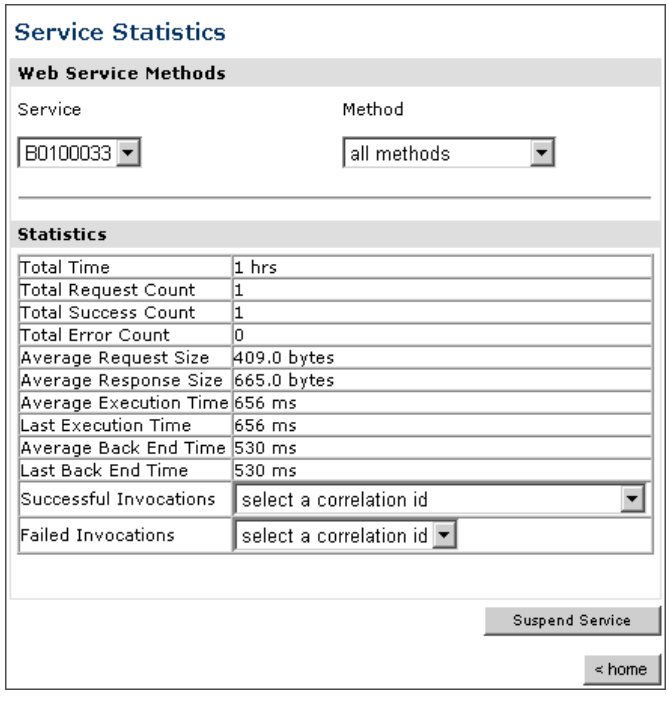
The system level summary provides services statistics at a system level.

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

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

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

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

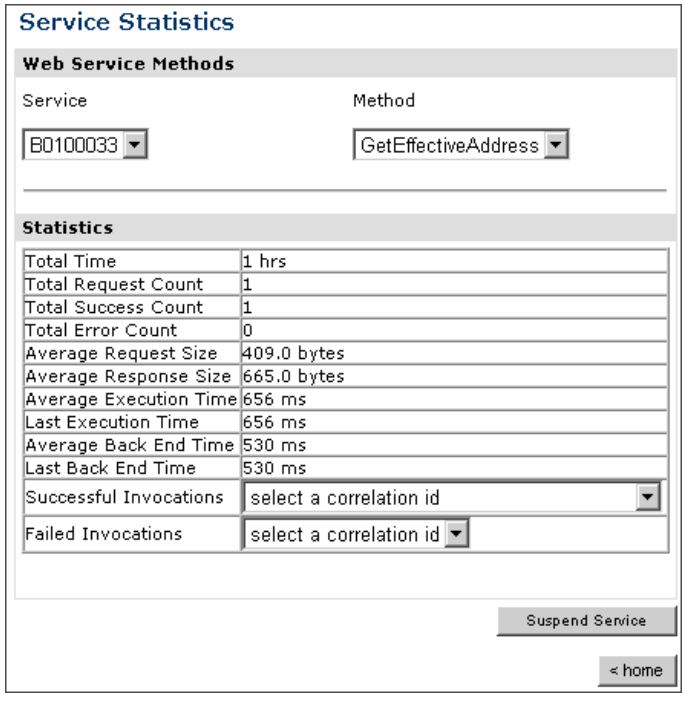


a. To stop a service at any time, click *Suspend Service*.

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

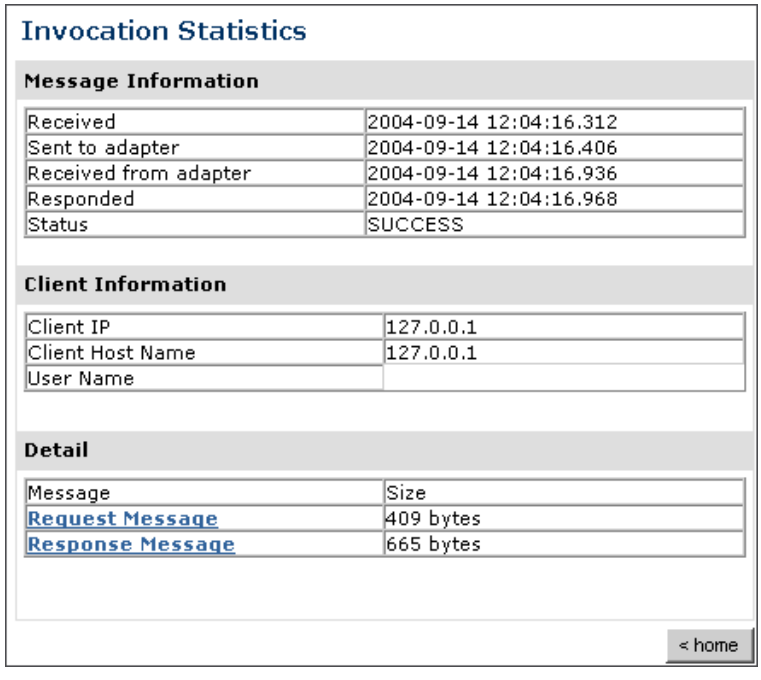
5. Select a method for the service from the Method drop-down list.

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



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

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



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

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

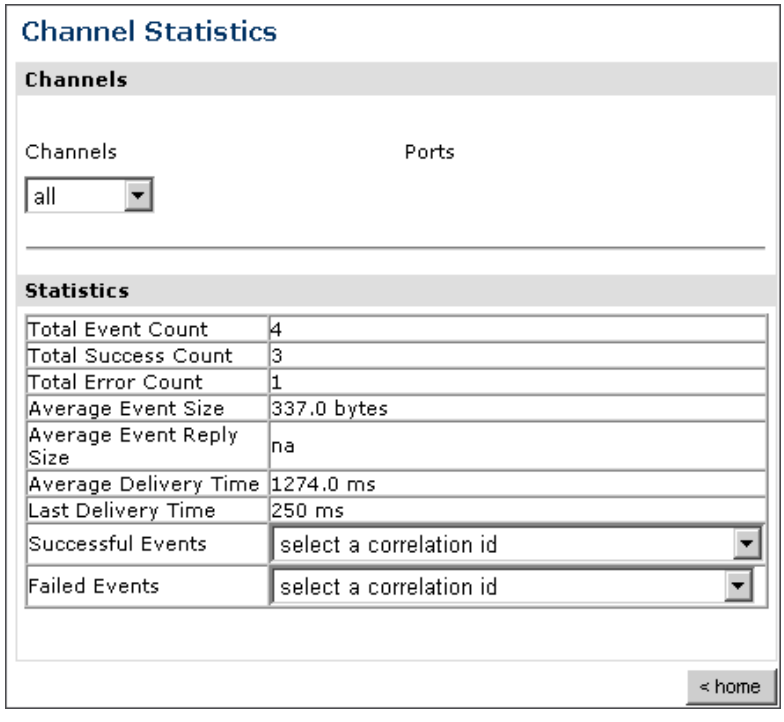
8. To return to the iBSE Monitoring Settings window, click *home*.

Procedure: How to Monitor Events

To monitor events:

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

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



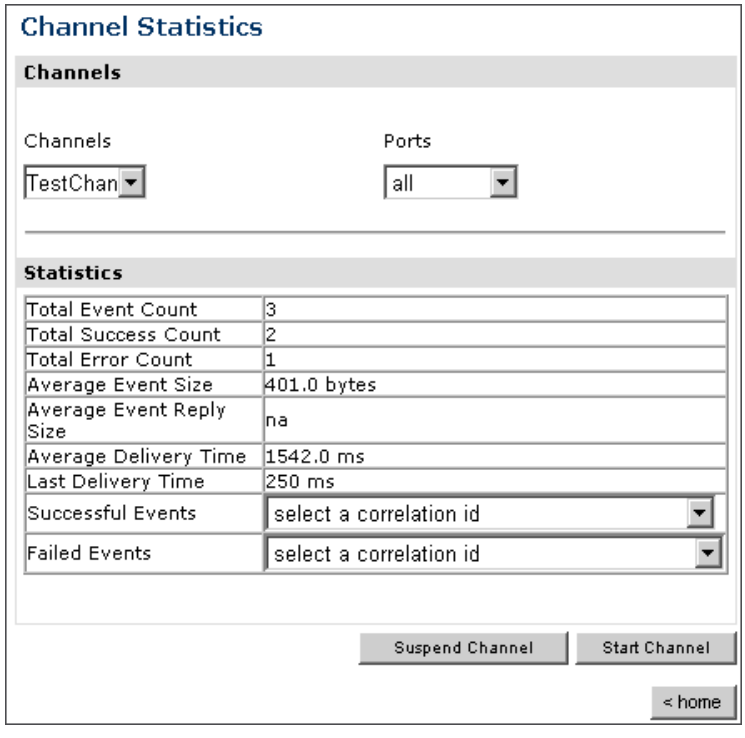
The system level summary provides event statistics at a system level.

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

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

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

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

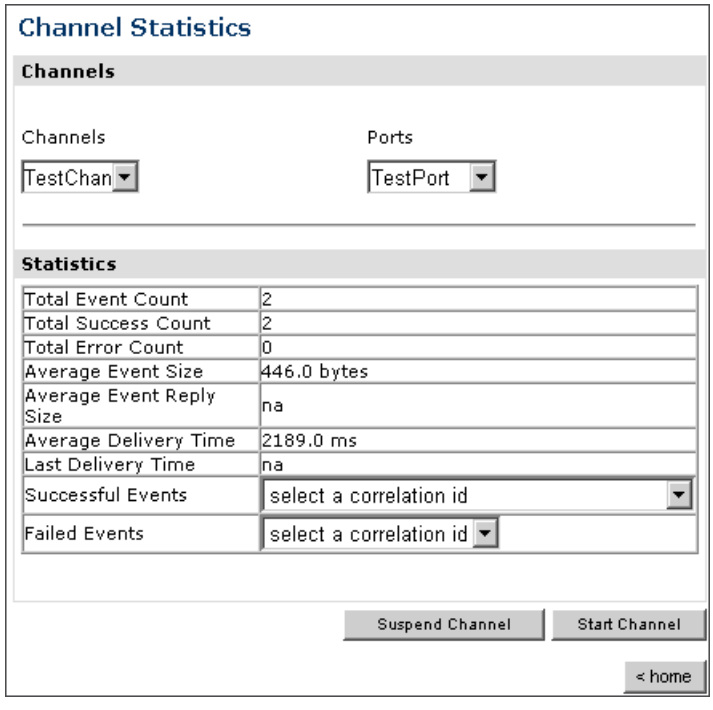


a. To stop a channel at any time, click *Suspend Channel*.

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

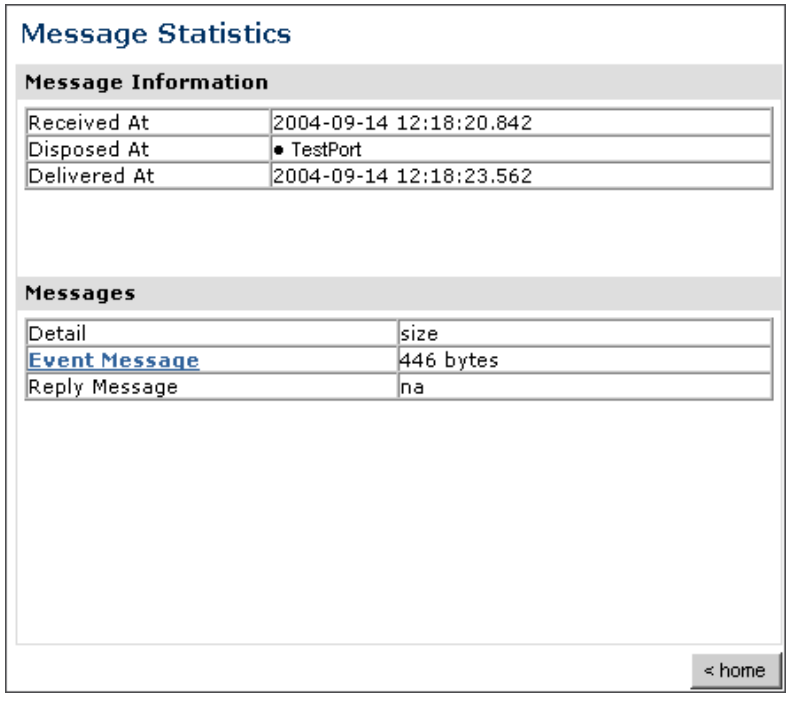
5. From the Ports drop-down list, select a port for the channel.

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



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

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



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

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

Managing and Monitoring Services and Events Using the JCA Test Tool

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

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

To manage and monitor services using the JCA Test Tool:

1. Open a Web browser to:

   `http://localhost:port/iwjcaivp`

   where:

   localhost

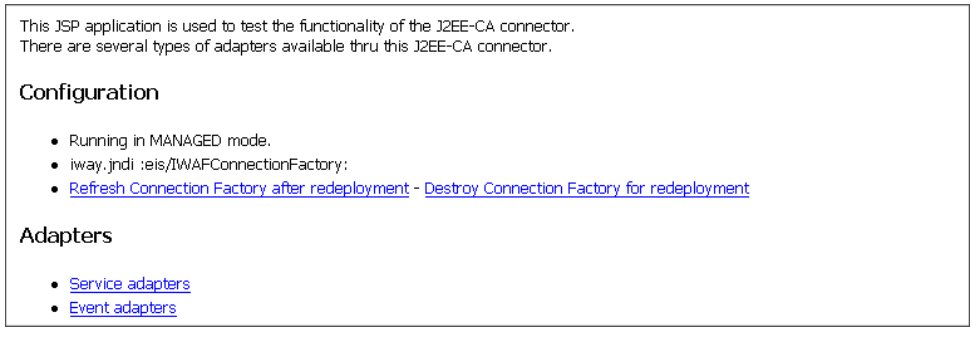
   Is the name of the machine where your application server is running.

   port

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

   `http://localhost:7001/iwjcaivp`

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



   The JCA Test Tool runs in managed mode by default.

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

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

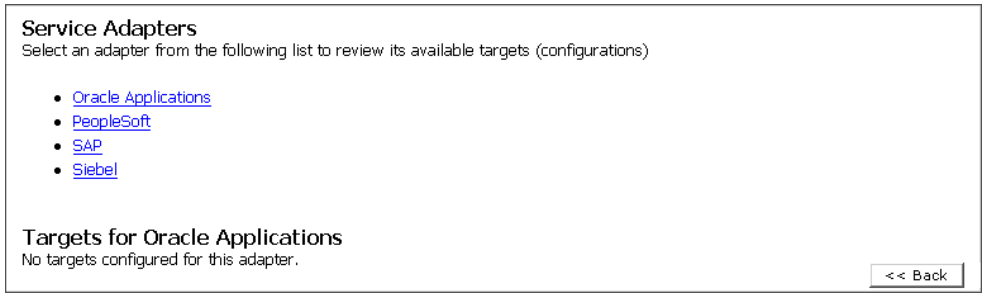
a. Click *Destroy Connection Factory for redeployment*.

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

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

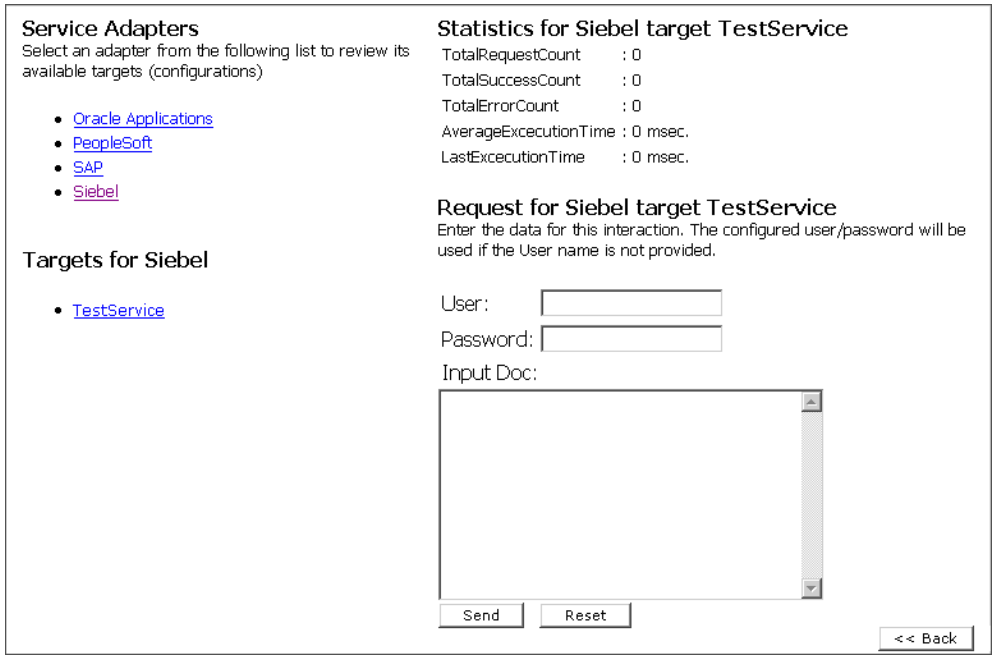
3. Click *Service adapters*.

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



4. Select a service adapter to monitor.

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



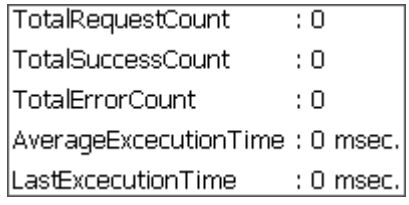
a. Click the desired target for your service adapter.

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

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

5. Click *Send*.

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



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

To manage and monitor events using the JCA Test Tool:

1. Open a Web browser to:

   `http://localhost:port/iwjcaivp`

   where:

   localhost

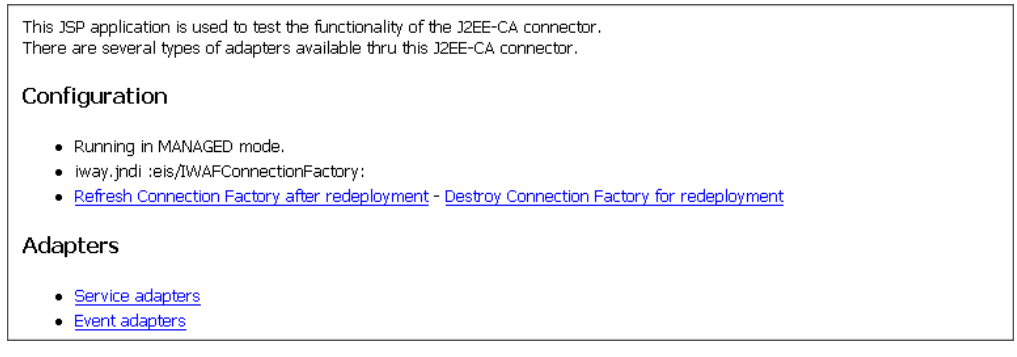
   Is the name of the machine where your application server is running.

   port

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

   ```
   http://localhost:7001/iwjcaivp
   ```

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



   The JCA Test Tool runs in managed mode by default.

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

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

   a. Click *Destroy Connection Factory for redeployment*.

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

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

3. Click *Event adapters*.

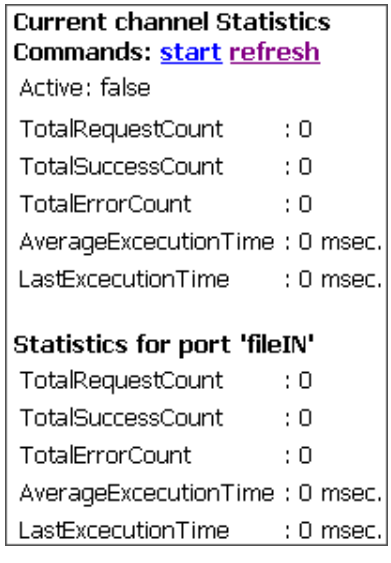
   The Event Adapters page opens.

4. Select the event adapter to monitor.

5. Click the desired channel for your event adapter.

6. Click *start*.

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



Setting Engine Log Levels

The following section describes how to set engine log levels for Servlet iBSE and JCA. For more information, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

Procedure: How to Enable Tracing for Servlet iBSE

To enable tracing for Servlet iBSE:

1. Open the Servlet iBSE configuration page at:

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

   where:

   localhost

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

   port

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

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

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

The default location for the trace information on Windows is:

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

Procedure: How to Enable Tracing for JCA

To enable tracing for JCA:

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

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

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

   For example:

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

A directory in the configuration directory contains the logs.

a. Review the logs generated by your application server.

b. Leave the remainder of the previous file unchanged.

3. Save the file and exit the editor.

4. Redeploy the connector.

Configuring Connection Pool Sizes

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

Procedure: How to Configure Connection Pool Sizes

To configure connection pool sizes:

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

2. Locate and change the following setting:

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

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

3. Save the file and exit the editor.

4. Redeploy the connector.

Migrating Repositories

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

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

File Repositories

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

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

where:

drive

Is the location of your iWay 5.5 installation.

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

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

iBSE Repositories

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

Note: The following procedure allows you to migrate only Web services. If migrating event handling information is one of your requirements, you must migrate at the database level. For more information, see *Migrating Event Handling Configurations* on page 5-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

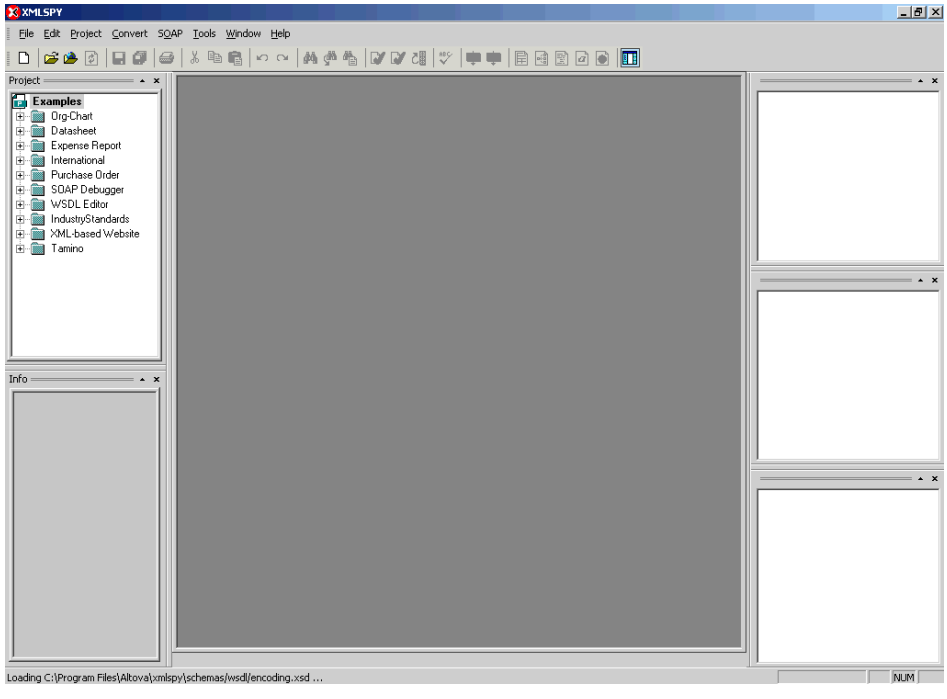
To migrate an iBSE repository that is configured for Oracle:

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

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

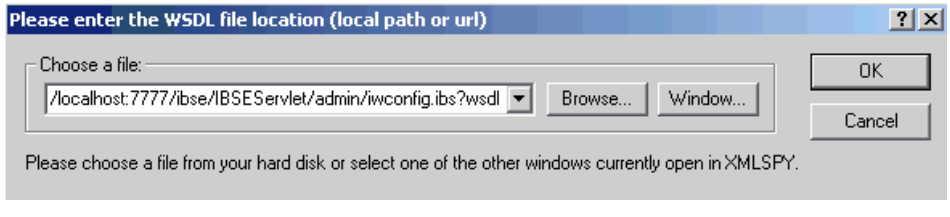
2. Open a third party XML editor, for example, XMLSPY.

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



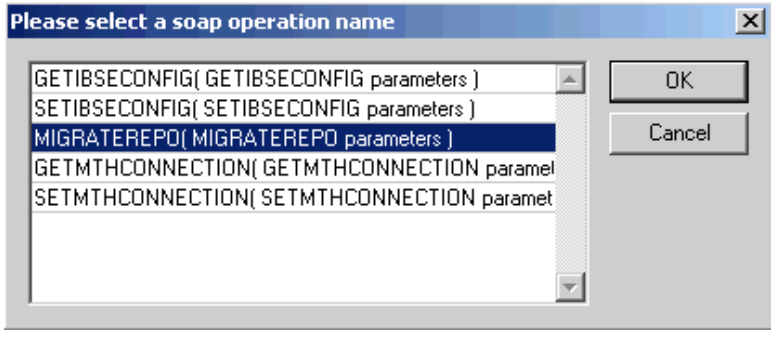
3. From the SOAP menu, select *Create new SOAP request*.

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



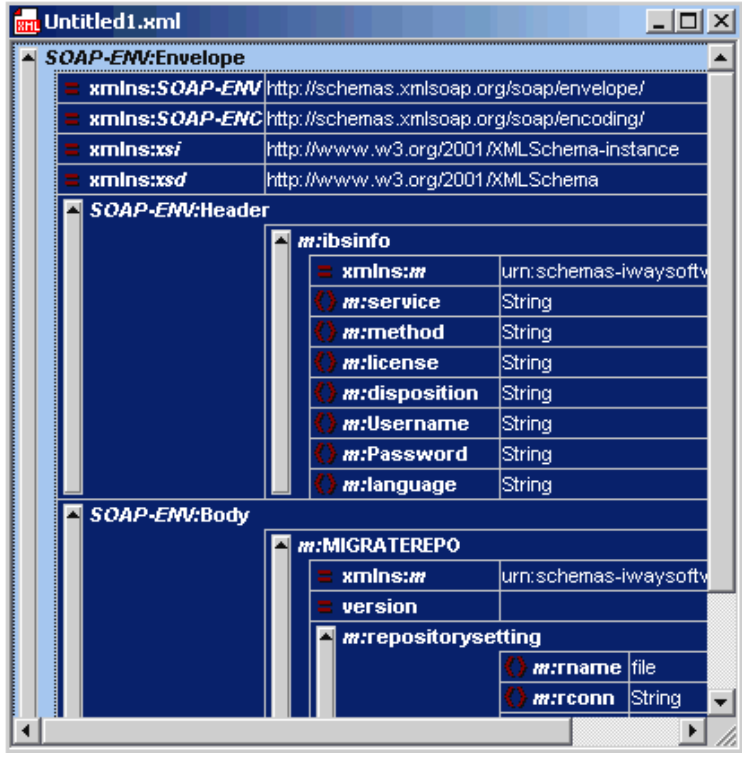
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

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



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

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



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

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

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

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

9. Locate the following section:

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

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

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

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

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

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

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

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

10. Perform one of the following migration options.

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

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

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

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

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

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

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

JCA Repositories

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

Procedure: How to Migrate a JCA Repository

To migrate a JCA repository:

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

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

2. Locate and copy the *repository.xml* file.

3. Place this file in a new JCA configuration directory to migrate the existing repository.

   Your JCA repository migrates to the new JCA configuration directory.

Migrating Event Handling Configurations

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

Procedure How to Migrate a Microsoft SQL Server 2000 Repository

To migrate a Microsoft SQL Server 2000 repository:

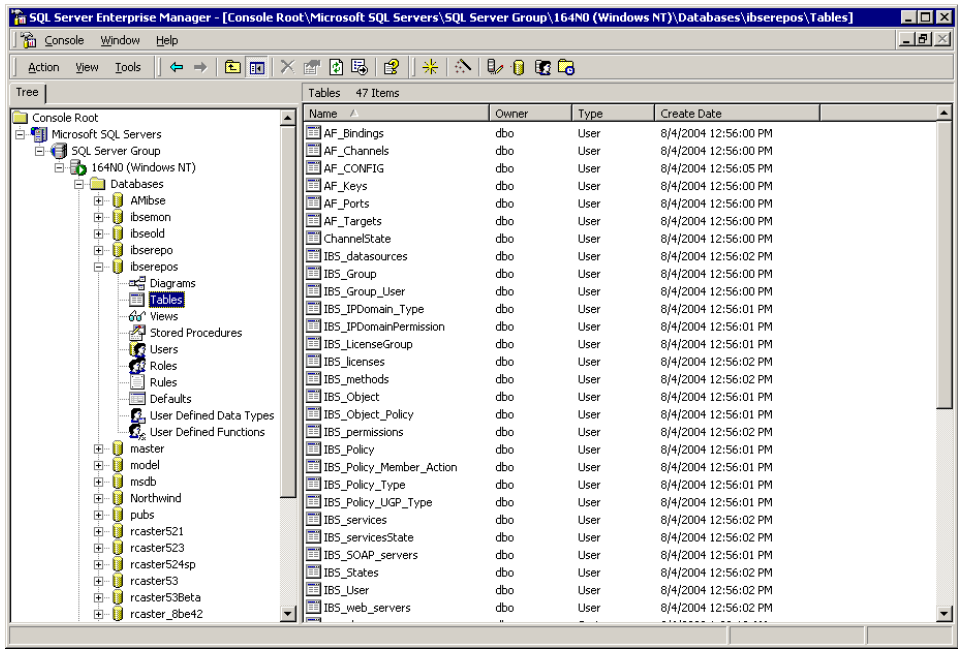
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

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

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

   ```
   iwse.sql
   ```

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



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

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

Procedure How to Migrate an Oracle Repository

To migrate an Oracle repository:

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

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

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

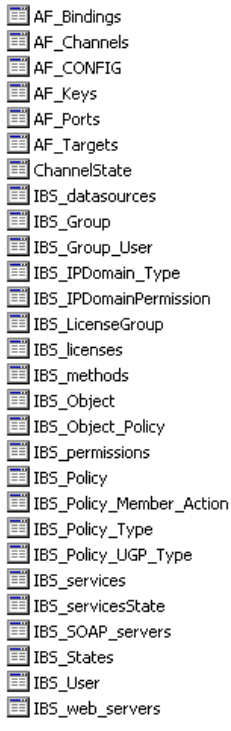
   For Oracle 8:

   ```
   iwse.ora
   ```

For Oracle 9:

```
iwse.ora9
```

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



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

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

Procedure How to Migrate a Sybase Repository

To migrate a Sybase repository:

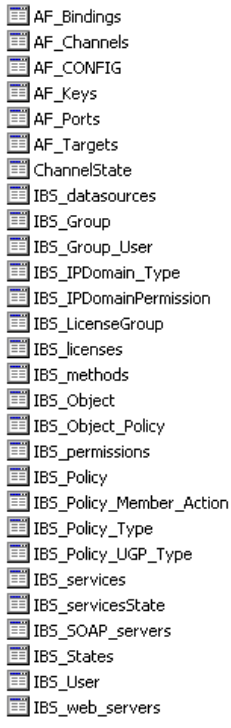
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

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

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

```
sybase-iwse.sql
```

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



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

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

Procedure How to Migrate a DB2 Repository

To migrate a DB2 repository:

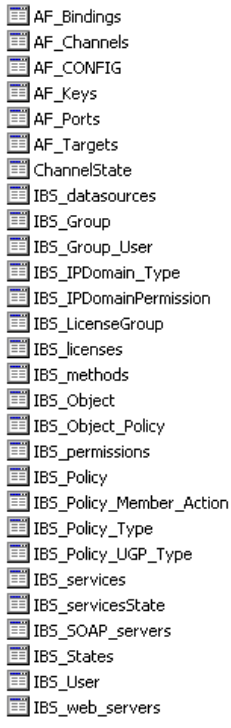
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

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

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

   ```
   db2-iwse.sql
   ```

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



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

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

Exporting or Importing Targets

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

Procedure: How to Export a Target

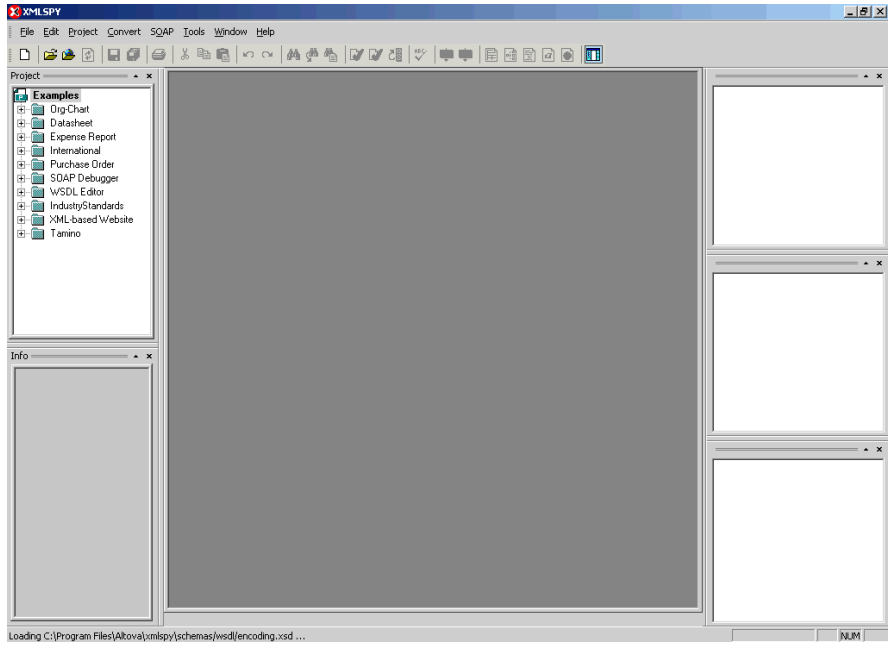
To export a target:

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

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

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

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



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

The WSDL file location dialog box opens.

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

5. Click *OK*.

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

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

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

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

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

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

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

9. Locate the following section:

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

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

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

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

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

Procedure: How to Import a Target

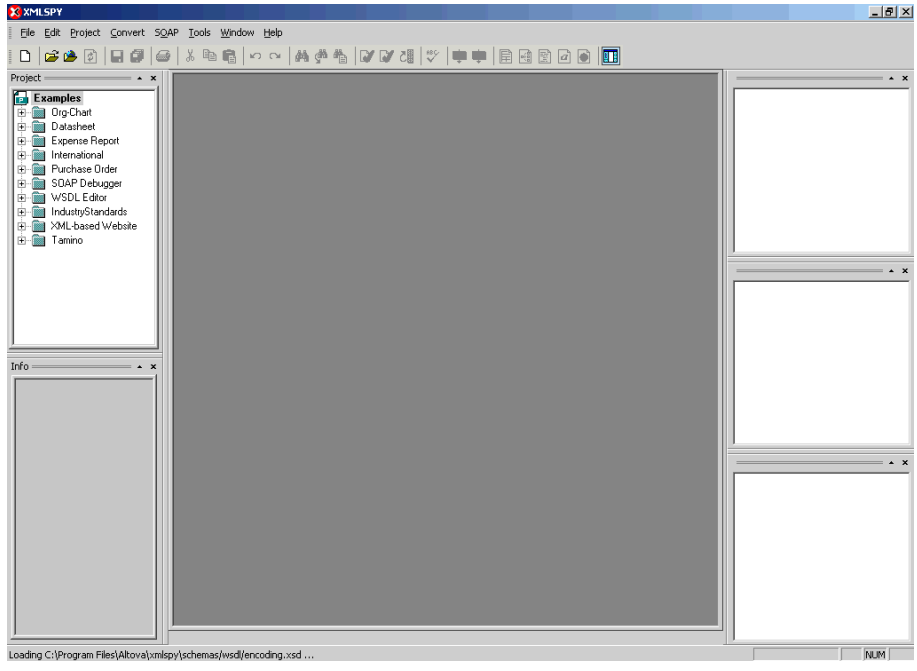
To import a target:

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

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

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

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



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

The WSDL file location dialog box opens.

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

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

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

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

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

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

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

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

8. Locate the following section:

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

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

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

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

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

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

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

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

Retrieving or Updating Web Service Method Connection Information

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

Procedure: How to Retrieve Web Service Method Connection Information

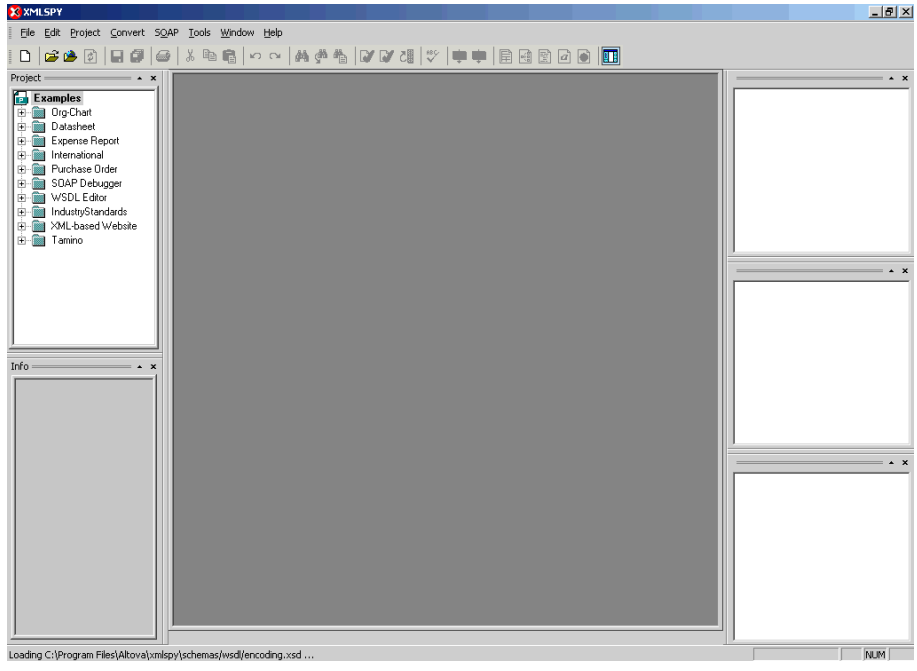
To retrieve Web service method connection information:

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

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

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

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



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

The WSDL file location dialog box opens.

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

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

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

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

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

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

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

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

8. Locate the following section:

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

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

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

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

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

Procedure: How to Update Web Service Method Connection Information

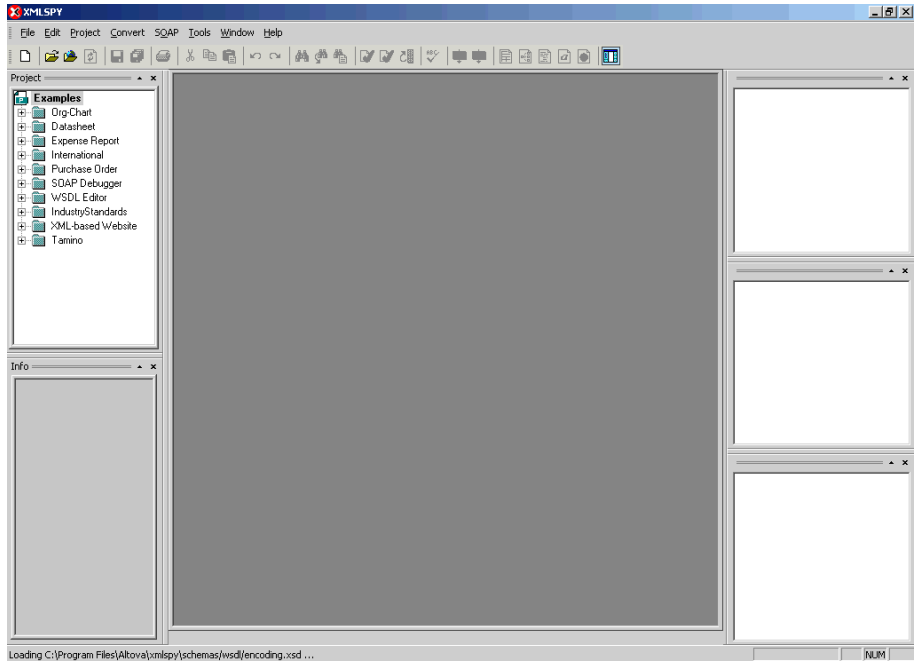
To update Web service method connection information:

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

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

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

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



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

The WSDL file location dialog box opens.

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

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

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

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

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

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

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

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

8. Locate the following section:

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

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

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

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

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

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

Starting or Stopping a Channel Programmatically

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

Procedure: How to Start a Channel Programmatically

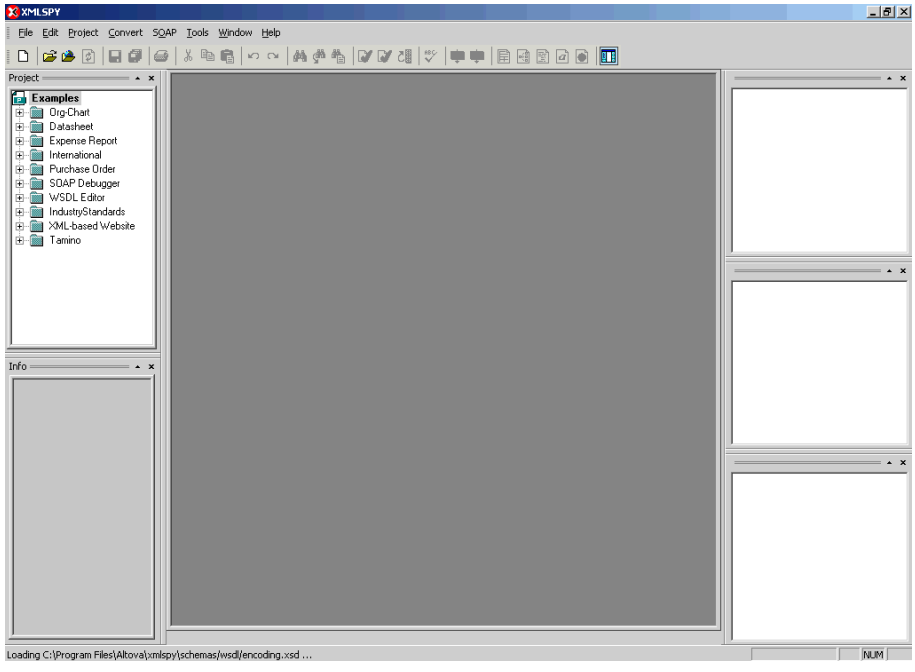
To start a channel programmatically:

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

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

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

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

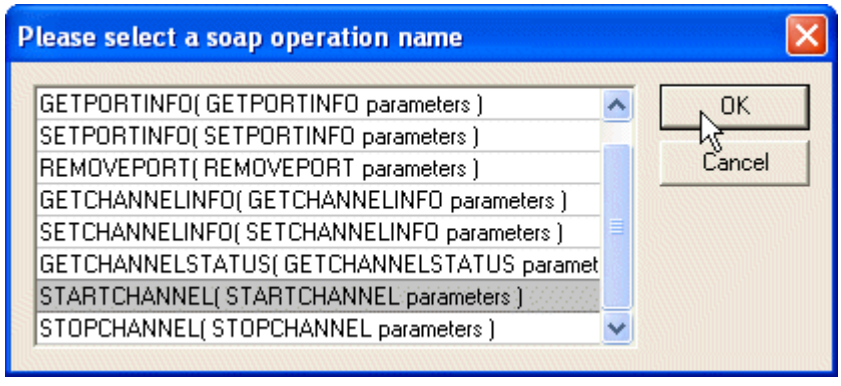


3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

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

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



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

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

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

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

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

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

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STARTCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STARTCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the Channel you want to start.

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

Procedure: How to Stop a Channel Programmatically

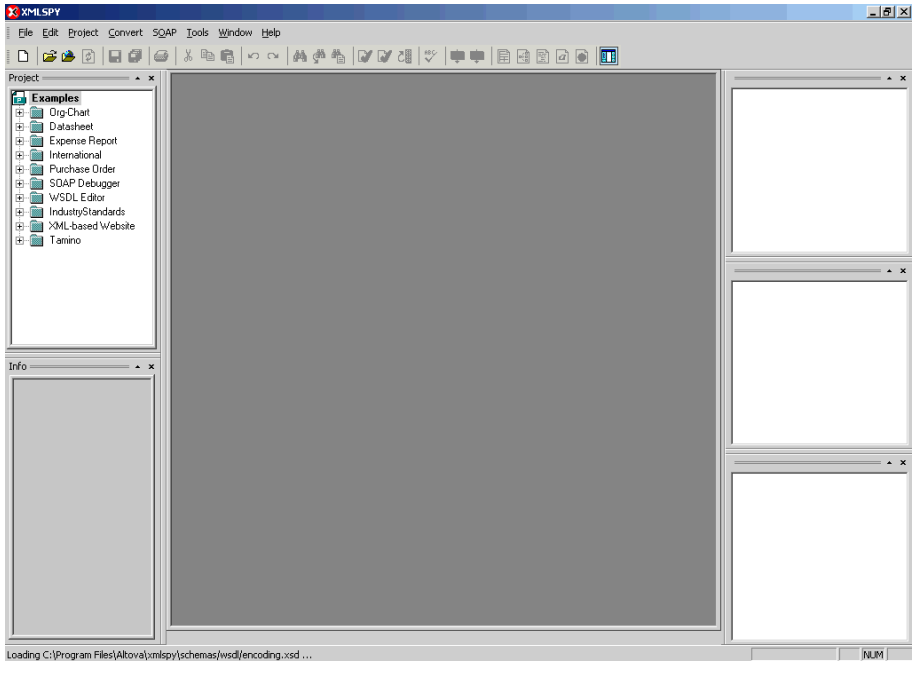
To stop a channel programmatically:

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

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

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

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

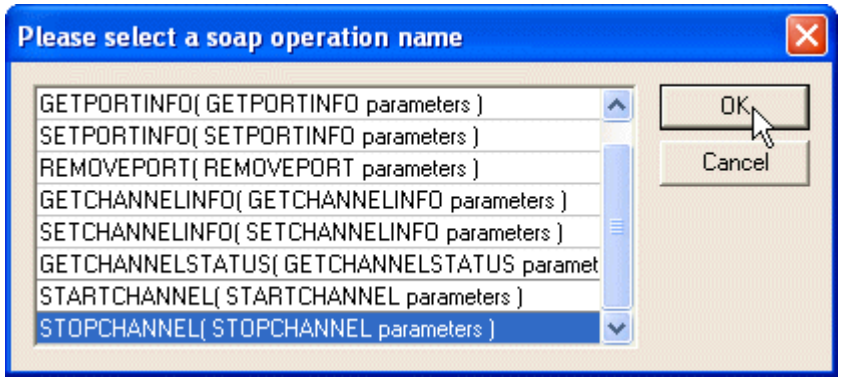


3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

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

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



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

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

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

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

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

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

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STOPCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STOPCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the Channel you want to stop.

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

CHAPTER 6

Using WebLogic Workshop

Topics:

- Using WebLogic Workshop to Create and Access a Web Service
- Running the Process Definition From WebLogic Workshop

This section describes how to create and access a Web service using WebLogic Workshop.

Using WebLogic Workshop to Create and Access a Web Service

WebLogic Workshop provides a framework for building Web services that are enterprise-class services. It provides simple controls for connecting to your enterprise resources.

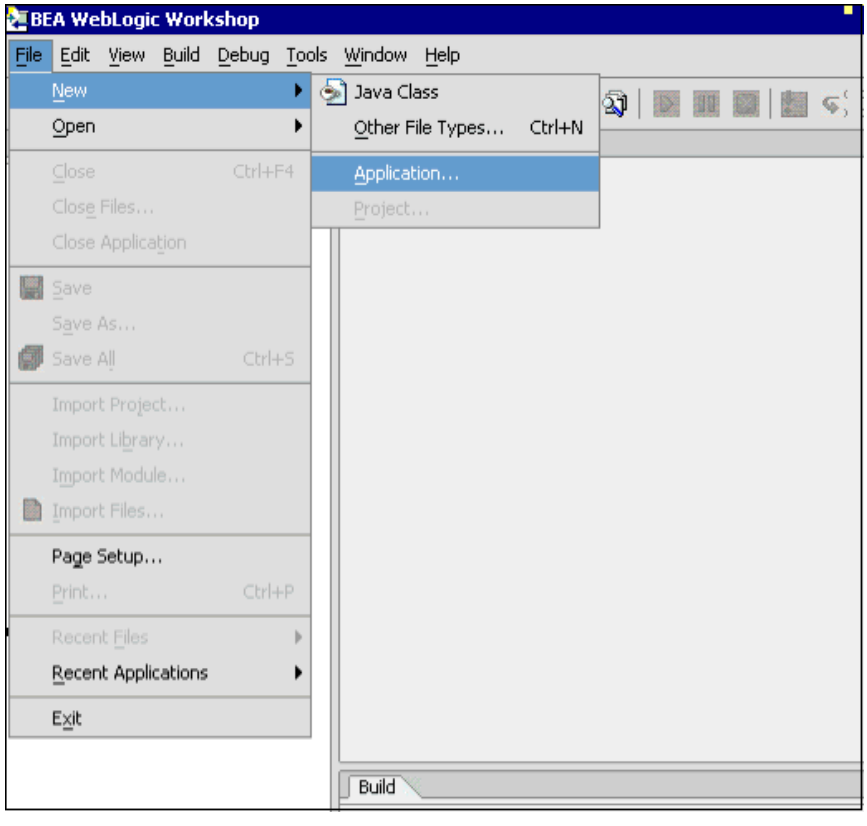
At the same time, WebLogic Workshop simplifies the process of creating Web services by insulating developers from the low-level implementation details that traditionally made Web service development the domain of sophisticated J2EE™ developers. With WebLogic Workshop, you can build powerful Web services whether you are an application developer or a J2EE expert.

Procedure How to Create an Application

To create an application:

1. From the Start menu, choose *Programs*, *BEA WebLogic Platform 8.1*, and then *WebLogic Workshop 8.1*.

   WebLogic Workshop opens.



2. To create a new application, select *New* from the File menu and then, *Application*.

   The New Application dialog box opens.

   a. In the upper-left pane, select *All*.

   b. In the upper-right pane, select *Default Application*.

   c. In the Directory field, type *C:\IWAYSRV.*

3. Click *Create*.

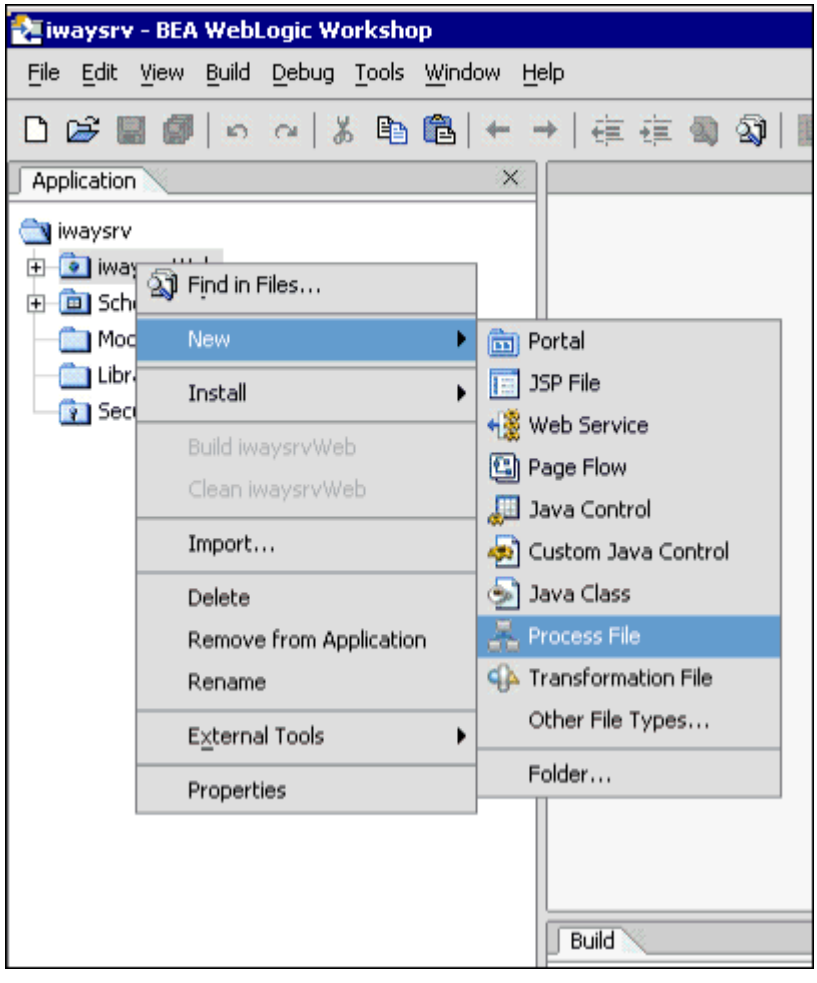
   A new Workshop application with a Web project and schema project is created. You can add additional projects later.

Creating a Schema for a Process Definition

The code for a process file resides within a Java™ Process Definition (JPD) file. A JPD file is considered to be a JAVA file in that it contains code for a Java class. However, because a file with a JPD extension contains the implementation code intended specifically for a process definition class, the extension gives it special meaning in the context of WebLogic Server.

Procedure How to Create a Schema

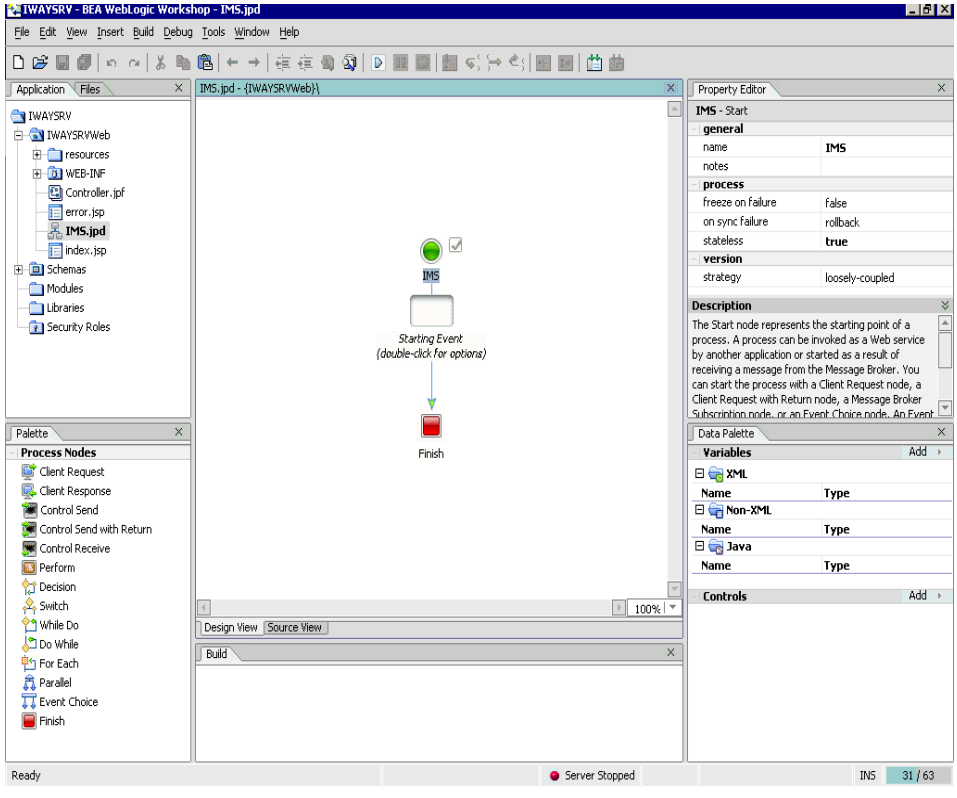
To create a schema to be used by the process definition:



1. In the Application tab, right-click the *iwaysrvWeb* folder.
   a. Select *New*.
   b. Then, select *Process File*.
2. In the File name field, type *IMS.jpd*.

3. Click *Create*.

A window similar to the following opens.



a. If not already selected, click the *Application* tab.

b. To import the PART.wsdl file previously saved, right-click the *Schemas* folder and select *Import*.

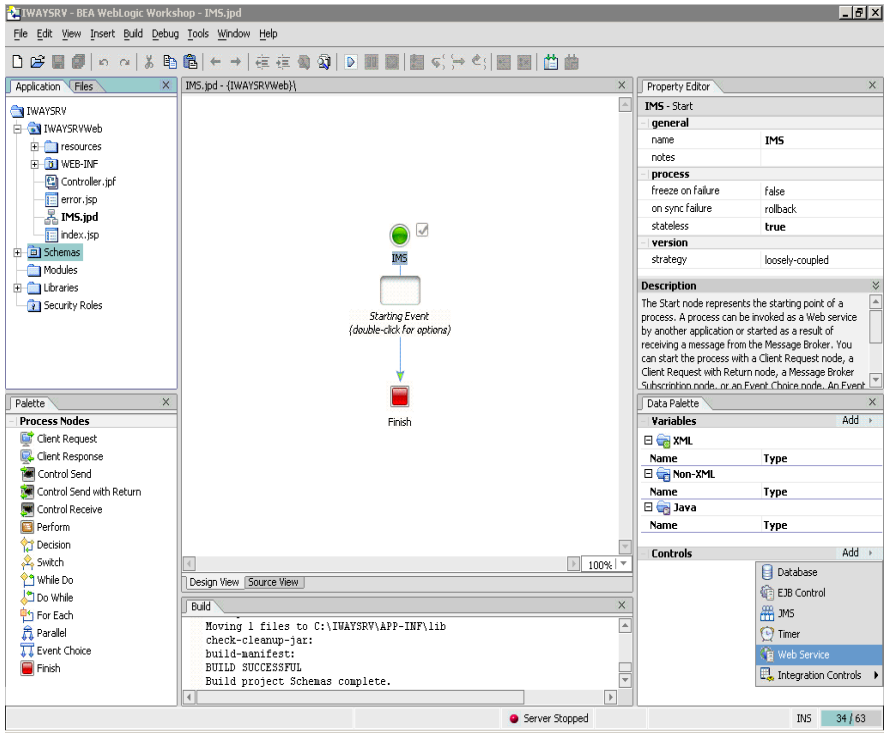
The Import Files window opens.

4. In the Import Files window, navigate to the location where you have saved the PART.wsdl file, for example, C:\IWAYSRV.

a. Highlight the *PART.wsdl* file.

b. Click *Import*.

WebLogic Workshop creates schemas to be used by the process definition as shown in the following illustration.



Creating and Inserting a Control for a Web Service

To access a Web service, you must first create and insert a control. A Web service control makes is easy to access an external Web service from a WebLogic Workshop application.

Procedure How to Create a Control for a Web Service

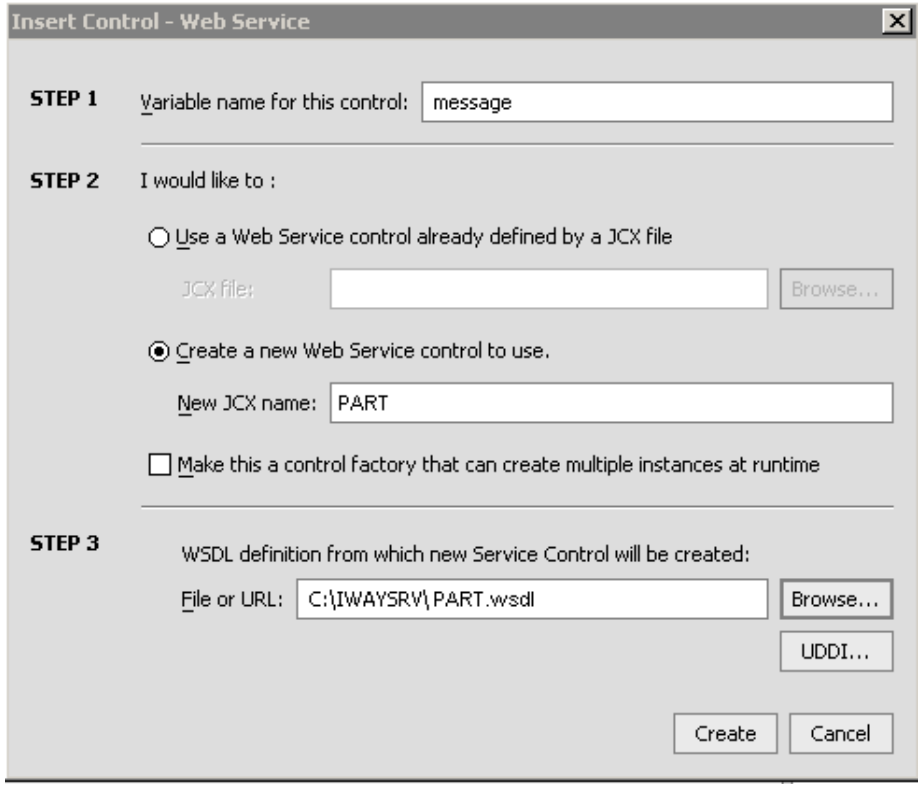
To create and insert a control for a Web service:

1. In the Controls pane, click *Add*.

   The Control menu opens.

2. Select *Web Service*.

The Insert Control – Web Service window opens.



a. Type a name for the control in the *Variable name* field, for example, message.

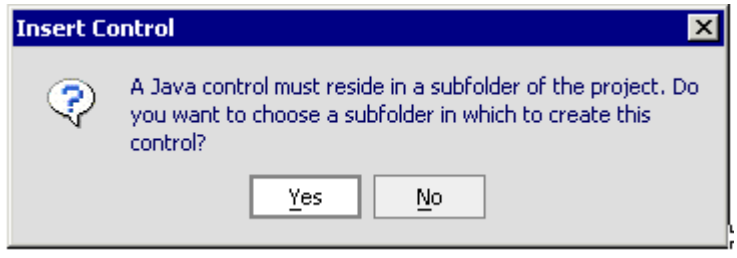
b. Select the *Create a new Web Service control to use* option button.

c. Type a name for the JCX, for example, PART.

d. Enter (or browse to) the location of the saved wsdl, for example, C:\IWAYSRV\PART.wsdl.

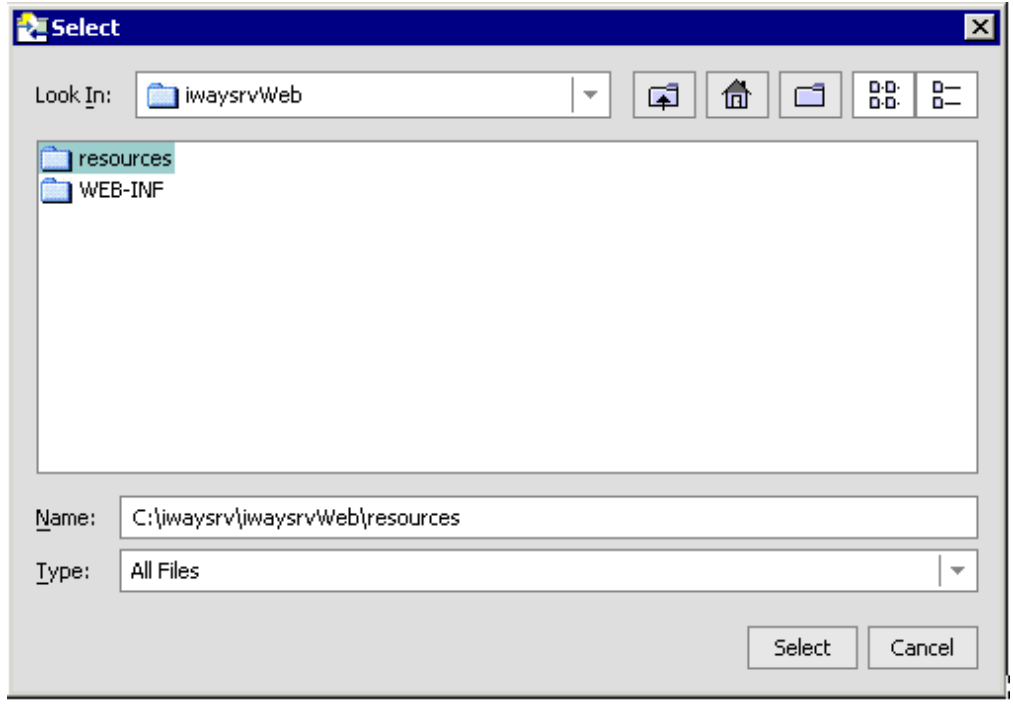
3. Click *Create*.

   An Insert Control message appears.



4. To choose a subfolder for the control, click *Yes*.

   The system prompts you to choose a subfolder for the Web service.



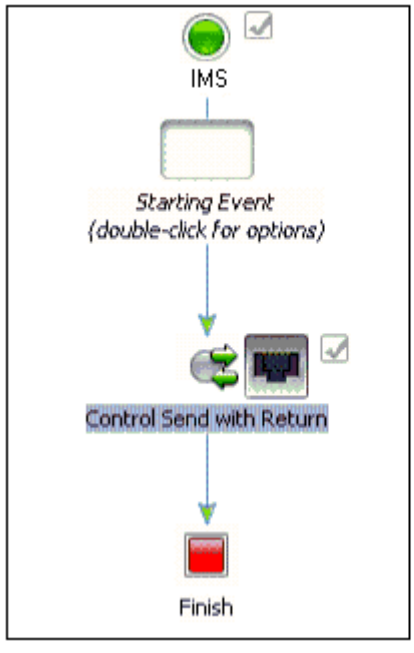
5. Highlight the *resources* subfolder and click *Select*.

   The Web service is created in the resources subfolder.

Procedure How to Insert a Control

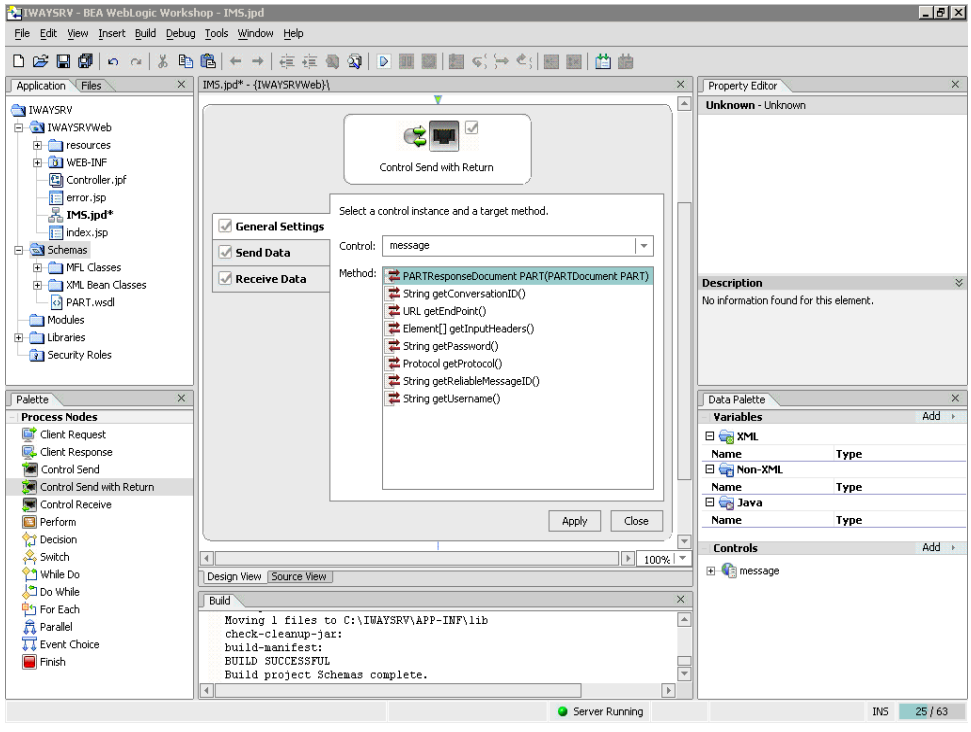
To insert a control:

1. Select the Palette pane.
2. Drag and drop the *Control Send with Return* node onto the process as follows:



3. Double-click the *Control Send with Return* node.

   The following window opens.



   a. From the list of available controls in the middle pane, select *message*.

   b. For the method, select *PARTResponseDocument*.

4. Click *Apply* and then, *Close*.

   Note: This control was created in the previous procedure.

Creating an Input Variable and an Output Variable

You must create an input and output variable for your business process.

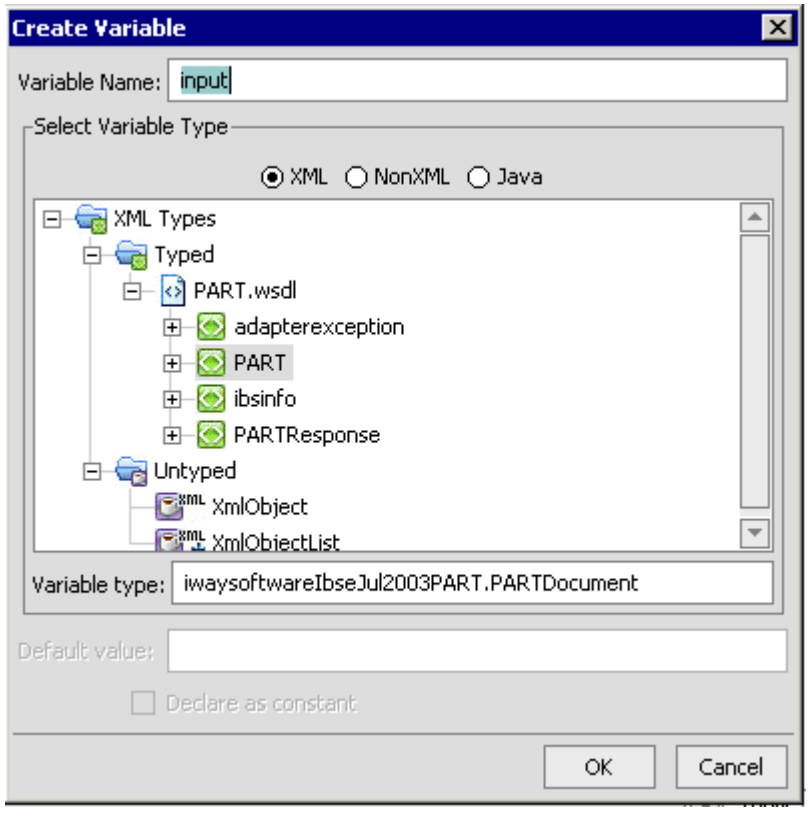
Procedure How to Create a Variable

To create a variable:

1. Double-click the *Control Send with Return* node.
2. Select *Send Data*.

3. In the *Select variables to assign* area, select *Create new variable*.

   The Create Variable dialog box opens.



   a. For the Send data, type a name, *input*, in the *Variable Name* field.

   b. Select variable type, *xml,* and click *OK*.

      The dialog box closes.

4. In the middle pane, select *Receive Data*.

5. From the *Select variables to assign* area, select *Create new variable*.

   The Create Variable dialog box opens.

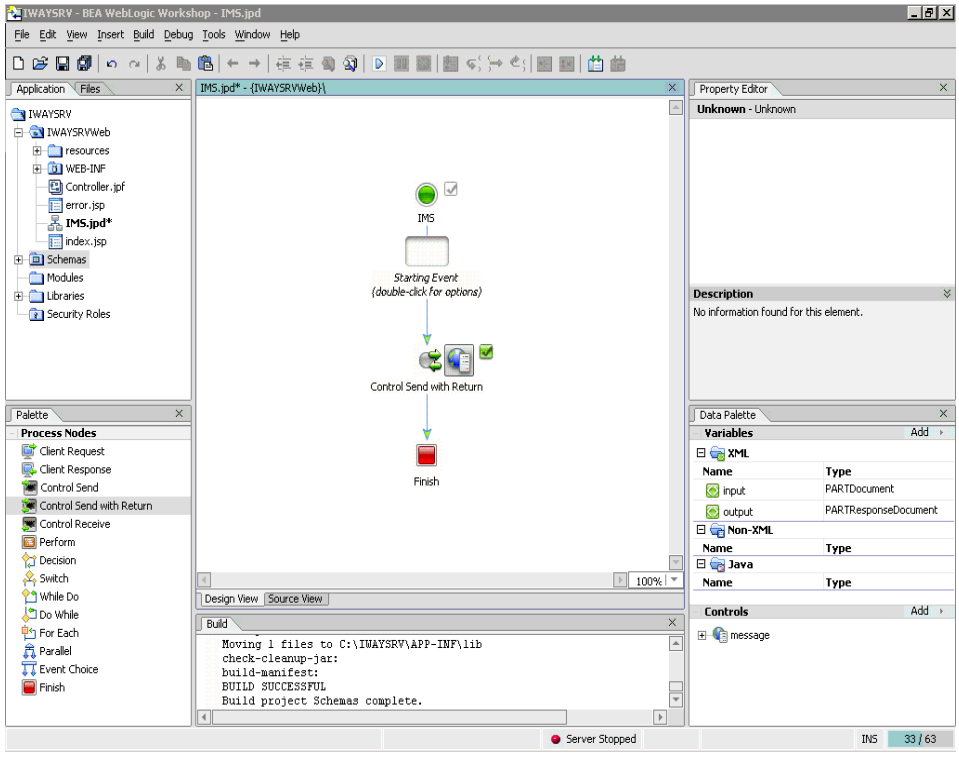
6. For the Receive data, create a new output variable.

   a. Type a name, *output*, in the *Variable Name* field.

   b. Select variable type, *xml,* and click *OK*.

   c. Click *Apply*, and then, click *Close*.

If you successfully configured the input and output variables, green arrows appear next to the *Control Receive* node as shown in the following illustration.



Configuring Parameters to Start an Event

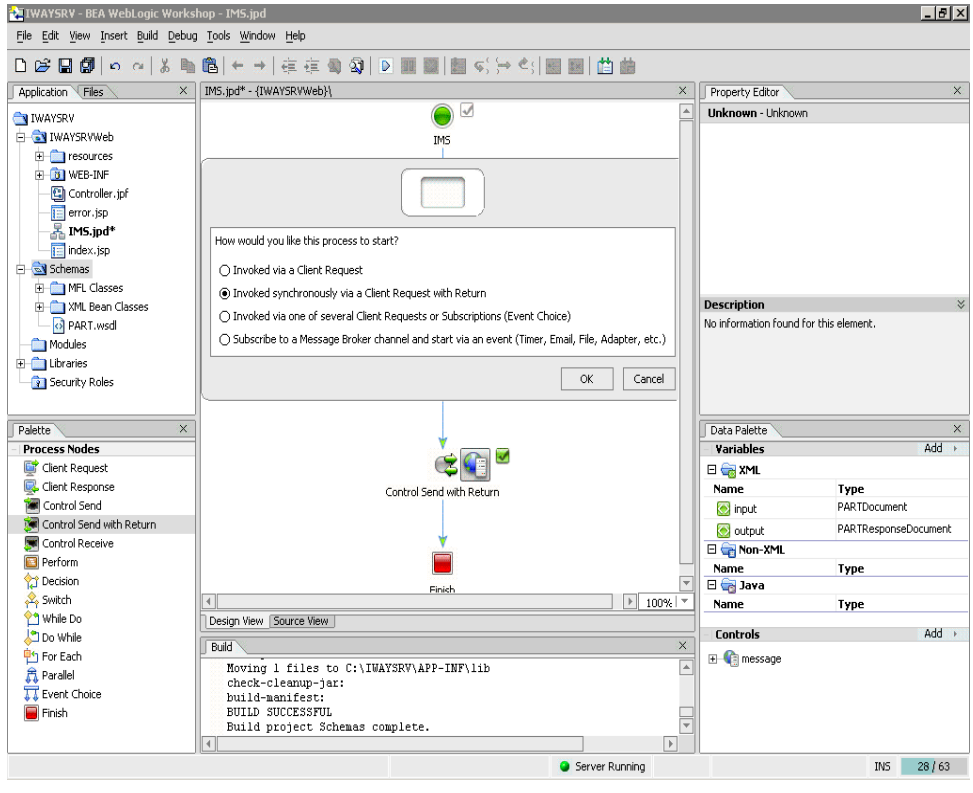
The first action in the business process is specified at the Start node. You specify how the business process is started at runtime by defining a Starting Event.

Procedure How to Configure Parameters to Start an Event

To configure parameters to start an event:

1. In the middle pane, double-click the *Starting Event* node.

   A pane opens and asks how you would like the process to start.

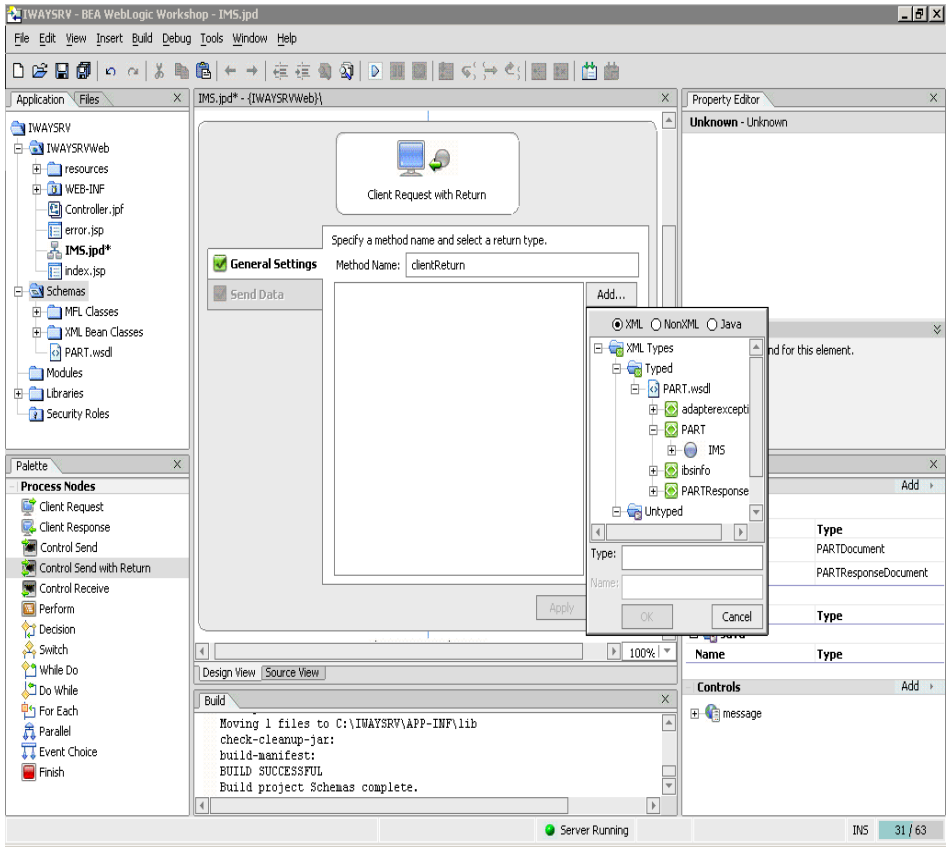


2. Select the *Invoked synchronously via a Client Request with Return* option button and click *OK*.

3. Double-click the *Client Request with Return* node.

4. Click *Add*.

A pane opens where you can specify a method name and select a return type. An additional pane opens where you can view the hierarchy.



a. Expand the *PART.wsdl* tree.

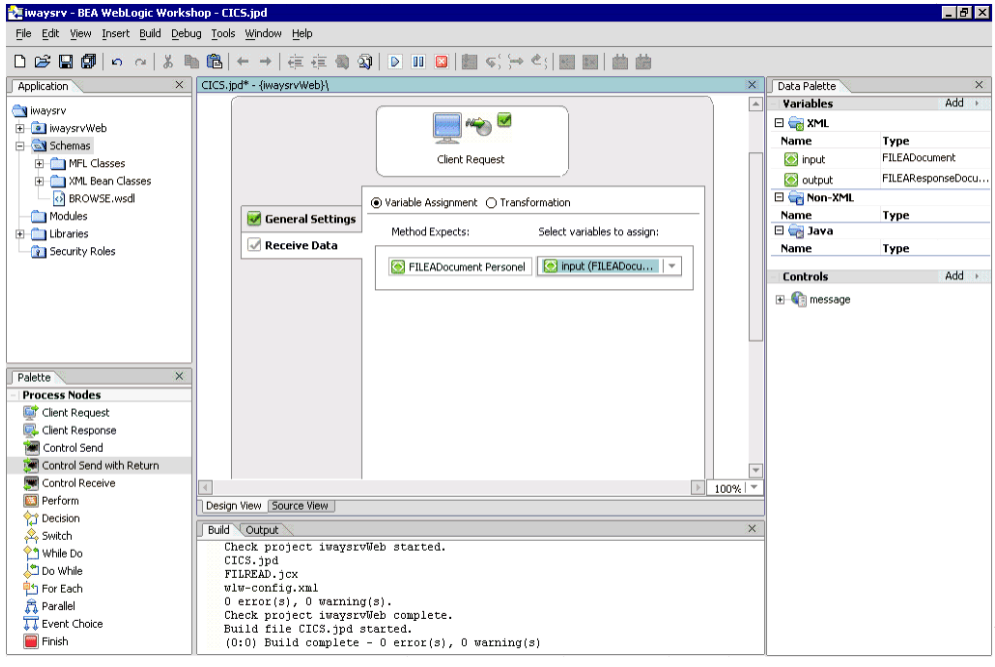
b. Select *PART* from the tree.

You can provide or modify the name of the variable to use.

The default value is x0.

5. Click *OK*.

   You can now map the input variable to the output variable.



6. In the middle pane, select *Receive Data*.
7. From the *Select variables to assign* drop-down list, select the input variable.
8. To activate these changes, click *Apply* and then, click *Close*.

Running the Process Definition From WebLogic Workshop

After you create a new process definition, you must ensure that WebLogic Server is running while you build your Web service. You can see whether or not WebLogic Server is running by viewing the status bar in WebLogic Workshop.

- If WebLogic Server is running, a green ball appears with the message, Server Running.
- If WebLogic Server is not running, a red ball appears with the message, Server Stopped.

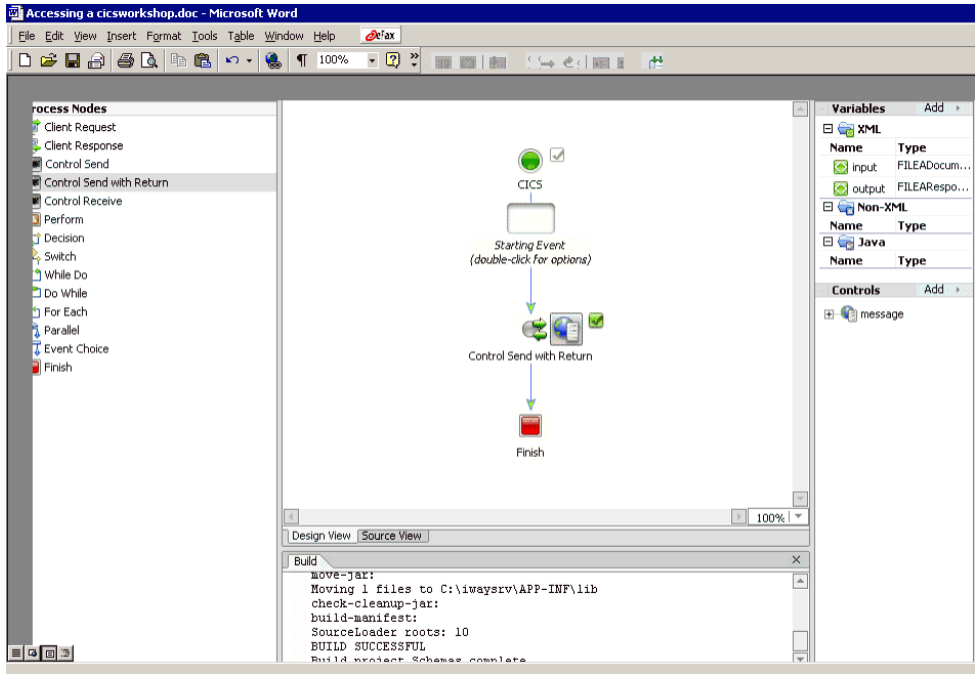
If you see the red ball in the status bar, follow the instructions in the following procedure to start WebLogic Server.

Procedure How to Start WebLogic Server and Deploy the Application to WebLogic

To start WebLogic Server:

1. From the Tools menu, choose *WebLogic Server* and then, *Start WebLogic Server*.
2. To deploy the application to WebLogic, select *Tools* and then, *Deploy Application*.
3. To start the application, click the *Start* button in the toolbar.

   The following test window opens.



4. Click the *Test SOAP* tab to test the XML stream to pass to the Web service.
5. Replace the following:

   ```
   location="string"
   ```

   with

   ```
   location="/IMS/Transaction/PART"
   ```

6. Modify the following:

   ```
   <urn:NUMB>anyType</urn:NUMB>
   ```

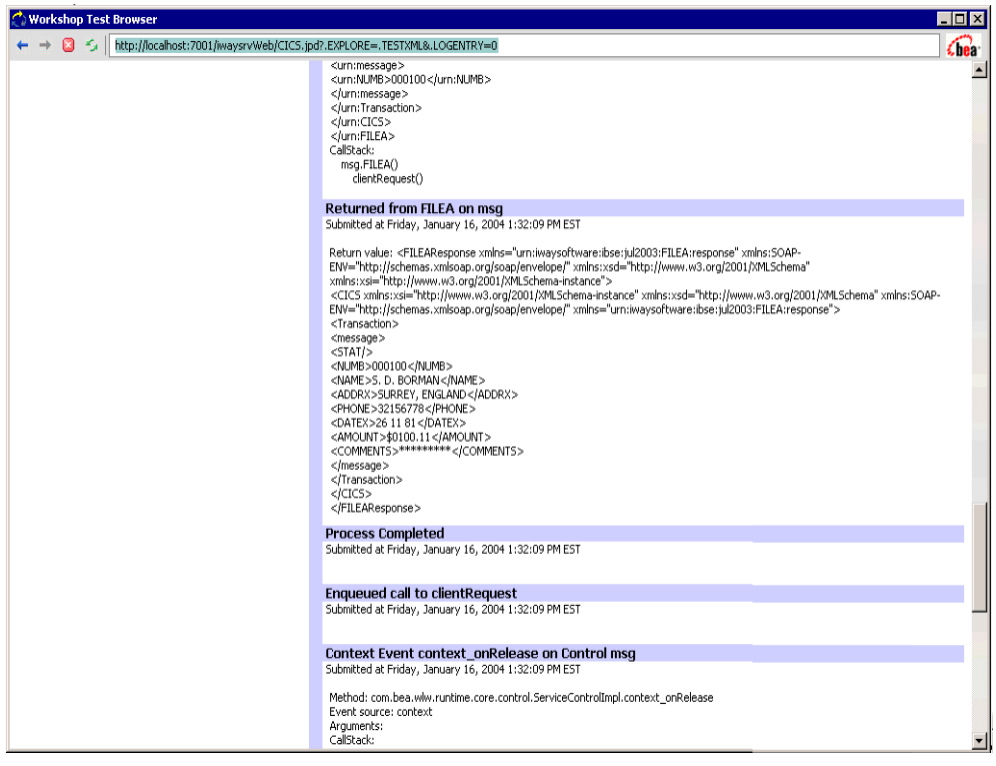
   to

   ```
   <urn:NUMB>000100</urn:NUMB>
   ```

7. To submit the request, click *clientResponse*.

8. To view the output, scroll to the middle of the screen.

The output should look similar to the following.



The output appears under the *Returned from PART under msg* heading.

Note: It may take a few seconds to display the output. You can click the refresh button while waiting for the response.

APPENDIX A

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

Topics:

- Starting Application Explorer in WebLogic Workshop
- Configuring a Connection to IMS/TM
- Managing a Connection to IMS/TM
- Creating an Adapter Transaction
- Creating Schemas for an Adapter Transaction
- Understanding iWay Business Services
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an iWay Extensible CCI Control to a BEA WebLogic Workshop Application

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

The following topics describe how to use Application Explorer to create IMS/TM transactions and generate request and response XML schemas for new or existing transactions. These schemas are used to represent a transaction for integration with external systems. In addition, this section provides information on how to create Web services.

Starting Application Explorer in WebLogic Workshop

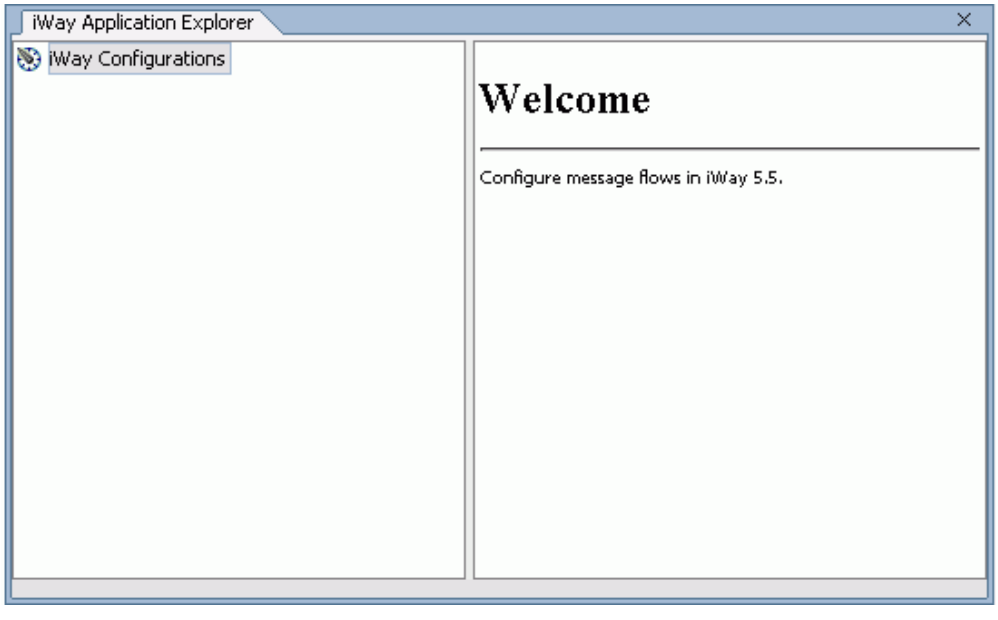
You can use Application Explorer with an iBSE or a JCA configuration. Before you can use Application Explorer, you must start BEA WebLogic Server.

Procedure How to Start Application Explorer

1. Start BEA WebLogic Workshop.
2. Verify 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 in BEA WebLogic Workshop.



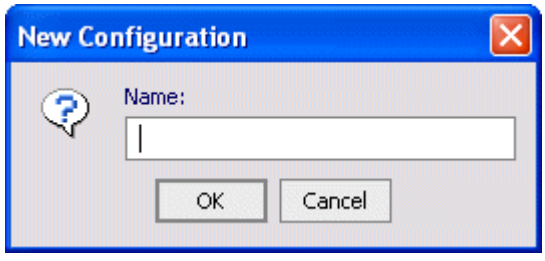
You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

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:

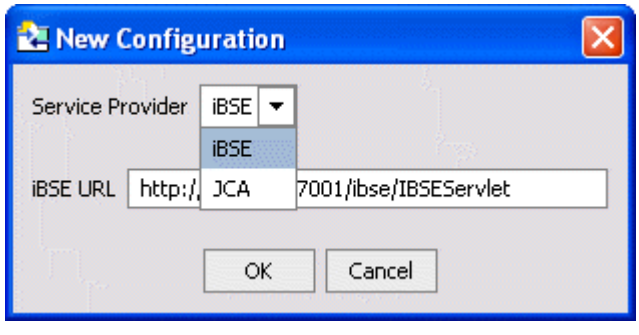
1. Right-click *iWay Configurations* and select *New*.

   The New Configuration dialog box opens:



2. Enter a name for the new configuration (for example, IMS) 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,

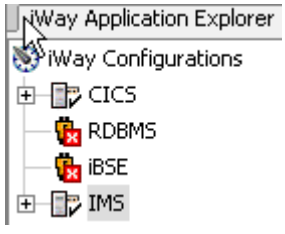
```
C:\Program Files\iWay55
```

where:

iWay55

Is the full path to your iWay installation.

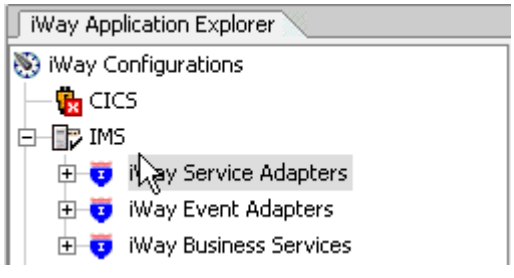
A node representing the new configuration appears under the iWay Configurations node. The right pane provides details of the configuration you created.



Procedure How to Connect to a New Configuration

Right-click the configuration to which you want to connect (for example, IMS), 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 IMS.

Configuring a Connection to IMS/TM

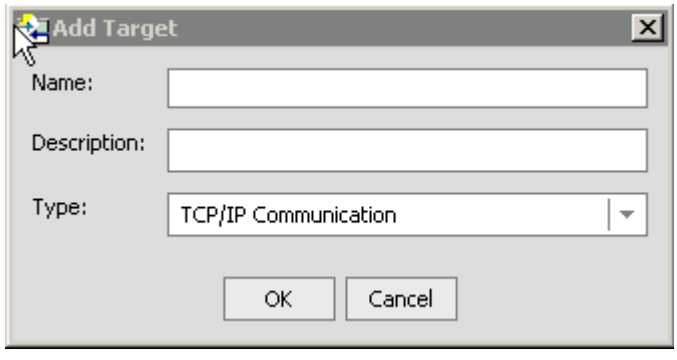
To access an IMS/TM adapter, you must configure a connection to that adapter. After the connection is created, it is automatically saved. You must establish a connection to the system every time you start Application Explorer or after disconnecting.

You can connect to IMS/TM using TCP/IP Communication or APPC Communication logon options.

Procedure How to Configure a Connection to IMS/TM

1. Expand the *iWay Service Adapters* node in Application Explorer.
2. Expand the *IMS* node.
3. Right-click the IMS node and select *Add Target*.

   The Add Target dialog box opens:



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

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

   b. In the Description field, type a description for the target name you just created (for example, *Connection using TCPIP option*).

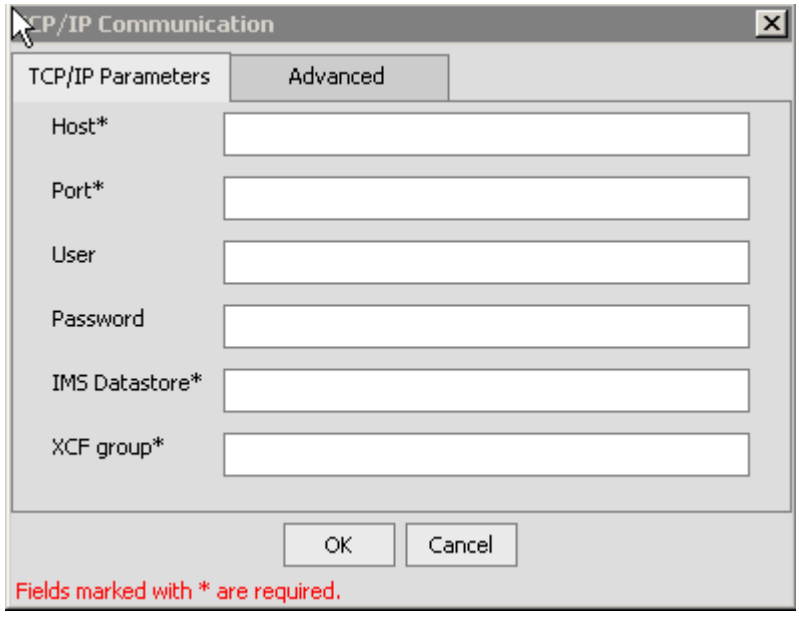
   c. In the Type drop-down list, select the type of target (for example, *TCP/IP Communication*).

4. Click *OK*.

   The connection name is verified for the system. If you entered an invalid instance name, a new Input dialog box opens and prompts you for an instance name again.

   - If you selected TCP/IP Connection as the type of target, proceed to step 5.
   - If you selected APPC Connection as the type of target, proceed to step 6.

5. The TCP/IP Communication dialog box opens:



Note: The IMS/TM connection parameters are consistent with those found in your IMS/TM system. For more information on parameter values that are specific to your IMS/TM configuration, consult your IMS/TM system administrator. This information should be the same for all programs in a single IMS/TM system.

a. Enter the parameters to configure a new connection to IMS/TM.

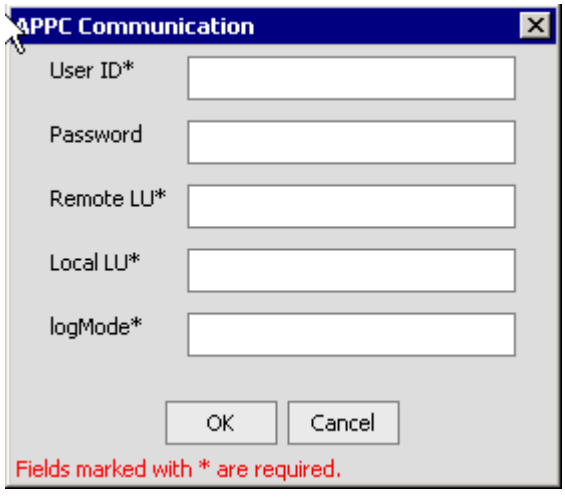
The following table lists and describes the parameters:

Parameter	Description
Host	Host name, or IP address, for the computer where IMS/TM is running.
Port IMS Connect is listening on	Port number on which IMS Connect is listening.
User	Valid user ID for IMS/TM.
Password	Valid password associated with the IMS/TM user ID.
IMS Datastore	Name of the IMS/TM datastore.
XCF Group	Name of the XCF group.

Override of message exit	Defaults to SAMPLE (*SAMPLE*).

b. Proceed to step 7.

6. The APPC Communication dialog box opens:



Note: The IMS/TM connection parameters are consistent with those found in your IMS/TM system. For more information on parameter values that are specific to your IMS/TM configuration, consult your IMS/TM system administrator. This information should be the same for all programs in a single IMS/TM system.

Enter the following parameters to make a new connection to IMS/TM.

The following table lists and describes the parameters.

Parameter	Description
User ID	Valid user ID for IMS/TM.
Password	Valid password associated with the IMS/TM user ID.
Remote LU	LU of APPC/IMS.
Local LU	LU of the SNA access point to which you have access, for example, SNA server.
logMode	Log mode of APPC/IMS.

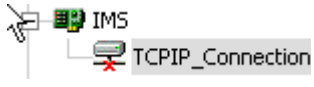
For more information, see Appendix B, *Running the Adapter Using LU6.2 Communication*.

7. Click *Finish*.

   The newly created connection (for example, TCPIP_Connection) appears as a node under the IMS service adapter. The configuration information is stored in the repository for the configuration you defined at installation time.

Procedure How to Connect to a Defined IMS/TM Target

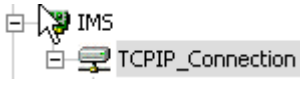
1. Expand the *iWay Service Adapters* node.
2. Expand the *IMS* node.
3. Click the target name (for example, TCPIP_Connection) under the IMS node.



   The Connection dialog box opens, populated with values you entered for the connection parameters.

4. Verify your connection parameters. If required, provide the password.
5. Right-click the target name and select *Connect*.

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



Managing a Connection to IMS/TM

To manage IMS/TM connections, you can:

- Disconnect from a connection that is not currently in use.

  Although you can maintain multiple open connections to different transaction processing systems, it is recommended to disconnect from connections not in use.

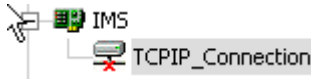
- Edit a connection to change its properties.
- Delete a connection that is no longer required.

Procedure How to Disconnect From a Connection to IMS/TM

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

   Disconnecting from IMS/TM drops the connection with IMS/TM, but the node remains.

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



Procedure How to Edit a Connection to IMS/TM

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

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

Procedure How to Delete a Connection to IMS/TM

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

   The node disappears from the list of available connections.

Creating an Adapter Transaction

After you create a connection to IMS/TM, you can add adapter transactions using Application Explorer. A single IMS/TM connection may be associated with multiple transactions. Each transaction represents one service offered by IMS/TM and consists of a program and its metadata.

A generic transaction is automatically added and represents IMS/TM services whose data will not be mapped to XML. You can use a generic transaction for transactions that accept no input and for transactions that return no output or when it is acceptable to return a non-formatted answer set.

For example, the IMS transaction PART connects to IMS/TM and returns PART information on successful adapter installation and configuration. One request and response schema is applicable for this generic transaction. The request schema for the generic transaction is in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

Using the generic transaction, the XML request document that is received must include the name of the program to be called in the <Transaction> element. The payload to be sent as the IMS segment must be in the <message> tag, which can be a maximum of 32,500 bytes.

The generic response schema is constructed from the data received from IMS. If the <message> element has more than 80 bytes, the received IMS segment is split into 80-byte messages. Illegal XML characters ('<', '/', and '&') are converted to XML entities.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<IMS>
<Transaction tpname="PART" noreply="NO">
<message>
<message>*</message>
</message>
</Transaction>
</IMS>
```

For transactions that require input and output and a formatted response, which is usually the case, you must add your own adapter transactions, as described in *How to Create an Adapter Transaction* on page A-12. XML request messages must specify the transaction to use in the location attribute of the <Transaction> tag. For example, if you create an IMS/TM transaction called PART, the location is "IMS/Transactions/PART".

To view a sample generic request or response schema or for information about specifying a transaction to use in the location attribute of the <Transaction> tag, see Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

Cobol Descriptions for Defining IMS/TM Transaction Output

The adapter uses Cobol descriptions to properly create an XML structure of the message that is returned from the IMS/TM transaction. For transactions that return one or more messages (for example, an error message that has its own message layout), the adapter transforms the message into the proper structure based on Cobol descriptions.

Because Message Format Services (MFS) is not used for IMS/TM transactions that the adapter can execute, such as PART, you must create or use existing Cobol descriptions.

If the application that executes the adapter requires proper formatting of each returned message from the transaction:

- You must examine all possible outputs of the transaction so you can create Cobol descriptions correctly.

  As a reference, it helps to use the Cobol description of the output message or the MFS message output descriptor area for the transaction to create the Cobol descriptions.

- You must supply Cobol descriptions of each type of output using Application Explorer when configuring the transaction.

  Note: For generic transactions where the format of the output is of no consequence, you are not required to supply Cobol descriptions.

  For the PART transaction, you require three Cobol descriptions. A RECTYPE field in each description is used by the adapter to determine which of the three messages is being returned by the adapter.

Sample Transaction PART

IBM supplies the PART transaction with an IMS system. This document uses the PART transaction for illustration purposes and as a reference for the adapter. The PART transaction accepts an input part number with a length of 17 characters or less. Based on what is passed to the PART transaction, an answer set is returned from the DP21PART database.

- If a part number is passed and found in IMS, the transaction returns detail information.
- If an asterisk ('*') is passed in the request, the transaction returns all part numbers in the database with their descriptions.
- If the part number is not found in IMS, the transaction, "PART NOT FOUND," is returned.

The PART transaction is an example of a transaction that returns multiple answer sets. Three different answer sets are returned based on what is passed in the request. The adapter enables you to create a response schema that contains different possible return messages.

Sample request documents, including sample response schemas for the PART transaction, are in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*. You specify the output as explained in *Creating an Adapter Transaction* on page A-10. You must know the field in the Cobol description that can be used as a record type and the value of that field. You specify the value of the field to create the appropriate response schema.

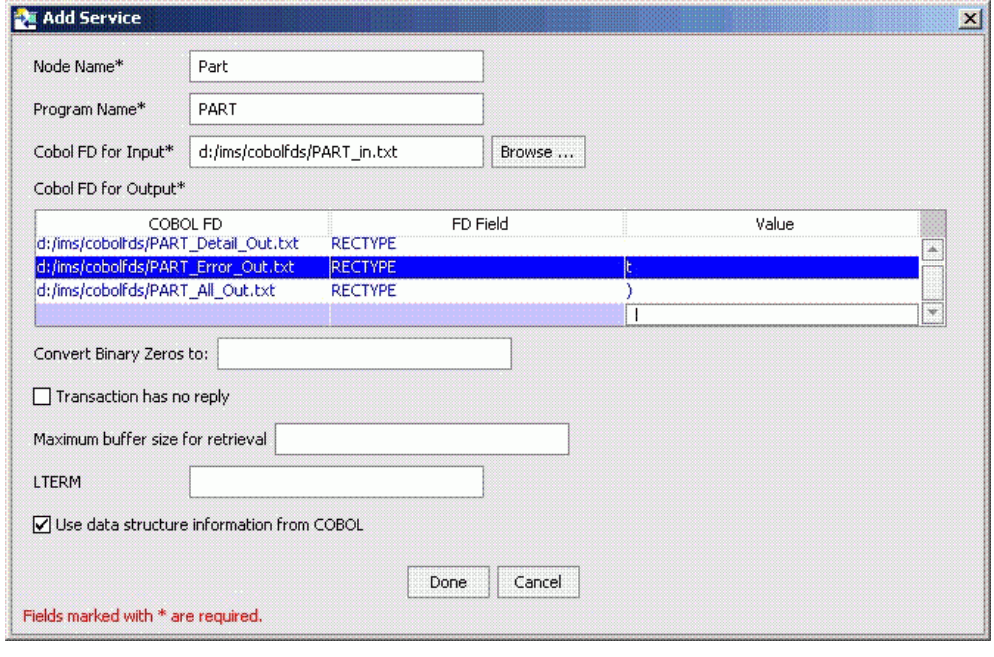
The previous is also true for events to determine the layout returned from IMS when you configure an IMS event. If you must configure an IMS event, contact Customer Support Services.

Procedure How to Create an Adapter Transaction

1. Expand the *iWay Service Adapters* node.
2. Expand the *IMS* node and connect to a IMS/TM target (for example, TCPIP_Connection).
3. Expand the node to which you connected.

   The Transaction node appears under the connected node.
4. Right-click *Transactions* and select *Add Service*.

   The Add Service dialog box opens:



5. To map the Cobol descriptions for the IMS/TM transaction, enter values for the following parameters. Fields with an asterisk are required.

<table>
<tr><th>Field</th><th colspan="2">Description</th></tr>
<tr><td>Node Name*</td><td colspan="2">Name to describe the adapter transaction you create. The name, for example, IMS_Transaction, appears under the Transactions node for the current connection. The name to use in the <Transaction location="..."> attribute.</td></tr>
<tr><td>Program Name*</td><td colspan="2">Name of the transaction to be called in IMS/TM (for example, PART). The PART input FD is shown in Appendix C, Sample Requests, Schemas, and Cobol File Descriptions.</td></tr>
<tr><td>Cobol FD for Input</td><td colspan="2">Location of the Cobol description that describes the input parameters of the IMS transaction to execute.

Converted by the adapter to an XML schema that the adapter uses to map from XML to the format required by IMS/TM at run time.</td></tr>
<tr><td rowspan="3">Cobol FD for Output
• COBOL FD
• FD Field
• Value</td><td colspan="2">Path that corresponds to the message you want returned from the IMS/TM transaction.

If the transaction can return multiple types of messages for each output Cobol description, enter the Cobol description field and value to determine the schema to use for a particular message.

Application Explorer creates the schema to use for a particular message based on the contents of a field that is returned. For example, a program called PART populates the field called RECTYPE. Depending on program logic, Application Explorer creates the correct response schema.</td></tr>
<tr><td>Value in RECTYPE Field

space ‘ ‘

parenthesis ‘)’

‘t’</td><td>Cobol Description

PART_Detail_Out

PART_All_Out

PART_Error_Out</td></tr>
<tr><td colspan="2">The PART_Detail_Out, PART_All_Out, and PART_Error_Out Cobol descriptions appear in Appendix C, Sample Requests, Schemas, and Cobol File Descriptions.</td></tr>
<tr><td>Convert non printable char to</td><td colspan="2">Character to convert binary zeros to in output.</td></tr>
</table>

Field	Description
Transaction has no reply	Select when you do not want to wait for a response from the program.
Maximum buffer size for retrieval	Maximum buffer size for an answer set.
LTerm	Logical terminal. Blank is the default unless specified by the user.
Use data structure information from Cobol	When this parameter is checked, the adapter creates request and response schemas that reflect Cobol group levels (for example, 05, 10, 20, and so on). The Cobol grouping will be reflected in the XML request and response schemas. You must check this parameter when Cobol input or output descriptions contain the Cobol OCCURS statement. When this parameter is checked and the program COMMAREA contains an OCCURS statement, the output Cobol definition must also contain an OCCURS statement. Do not "flatten out" the Cobol output description, since the adapter relies on the number of OCCURS when formulating the program's output.

Note: You must transfer the Cobol descriptions to a location accessible to Application Explorer. For the correct Cobol descriptions to use for the program, contact your IMS/TM Administrator or application developer.

6. Click *Add*.

   The new IMS/TM transaction is added, for example, IMS_Transaction under the Transactions node for the current connection.

Cobol Descriptions for Input and Output Communications

IMS Connect Considerations: Every time IMS Connect returns data (that is, an IMS Segment) it prefixes the data with LL, which represents the length of the returned data, and ZZ, which is a reserved field of binary zeros. The LL is called the IRM_LEN and is the length of the IRM structure, while ZZ is called IRM_RSV and is a reserved field. The format of the IRM structure is LLZZDATA, where LL is the total length of this segment, ZZ are binary zeros, and DATA is the IMS trancode followed by transaction data.

Important: The Adapter for IMS/TM does not require these prefixes in the Cobol input and output definitions. Only describe the data portion in the Cobol input and output definitions.

Reference Syntax For Cobol Descriptions

You must use the following syntax for binary, packed, and float fields for the Cobol descriptions for the adapter transaction input and output formats:

For a binary field:

```
05  BINARY-FIELD          PIC S9(n) COMP.
```

For a packed-decimal field:

```
05  PACKED-FIELD          PIC S9(n) COMP-3.
```

For a single-float field:

```
05  FLOAT-SINGLE          COMP-1.
```

For a double-float field:

```
05  FLOAT-DOUBLE          COMP-2.
```

Note: Underscores are not supported in Cobol descriptions.

Creating Schemas for an Adapter Transaction

When you select an adapter transaction in Application Explorer, the adapter automatically generates the schemas for the selected Cobol descriptions and associates them with the transaction. To view the schema information, select the transaction node and then click the *Request Schema* or *Response Schema* tab. The schemas generated for the sample Cobol descriptions appear in Appendix C, *Sample Requests, Schemas, and Cobol File Descriptions*.

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 the adapter is used with an iBSE configuration, Application Explorer stores the schemas under a \schemas subdirectory of the iWay home directory, for example,

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

where:

TCPIP_Connection

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

When the adapter is used with a JCA configuration, Application Explorer stores the schemas under a \schemas subdirectory of the iWay home directory, for example,

```
C:\Program Files\iWay55\config\base\schemas\IMS\TCPIP_Connection
```

where:

TCPIP_Connection

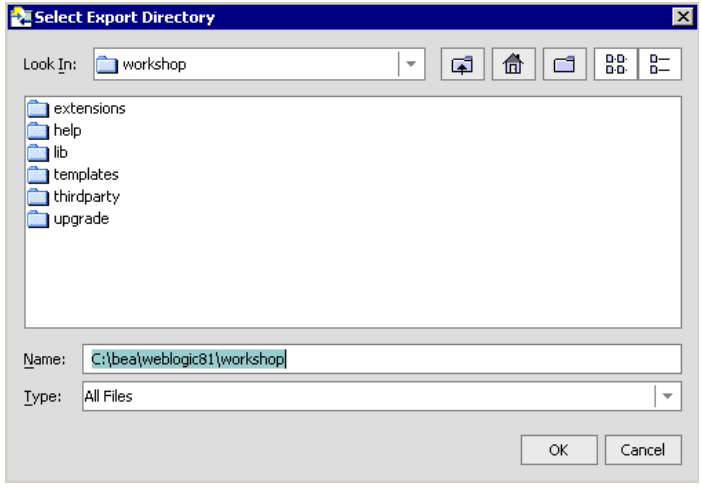
Is the name of the connection to the IMS/TM system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

The following screen shows the Request Schema for the Part transaction:

The following screen shows the Response Schema for the Part transaction:

Procedure How to Export a Schema

1. If you have not already done so, connect to the target from which you want to export a schema (for example, TCPIP_Connection).
2. Right-click the transaction from which you want to export a schema, and select *Export Schema*.
3. The Select Export Directory dialog box opens:



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

Note: To view a sample schema, see Chapter C, *Sample Requests, Schemas, and Cobol File Descriptions*.

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

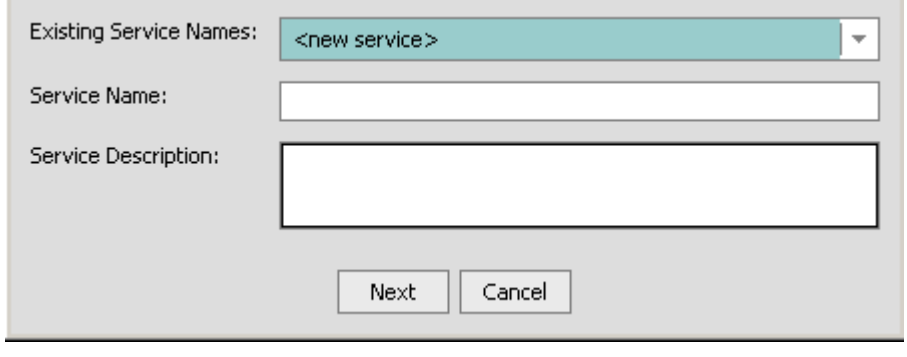
Creating a Web Service

After you connect to your application system and create an XML schema for a transaction, you can create a Web service. The following procedure describes how to create a Web service for IMS/TM using Application Explorer.

Procedure How to Create a Web Service

1. Expand the *iWay Service Adapters* node.
2. Expand the *IMS* node.
3. If you have not already done so, connect to the target for which you want to create a Web service (for example, TCPIP_Connection).
4. Right-click the transaction for which you want to create a Web service, and select *Create iWay Business Service*.

   The service information dialog box opens:

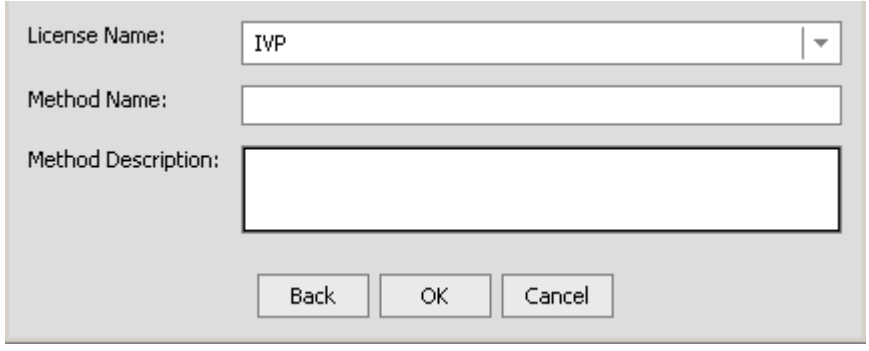


5. Choose whether to create a new service or use an existing service.
   a. Select either a new service or an existing service from the Existing Service Names drop-down box.
   b. Specify a service name if you are creating a new service.

      This name identifies the Web service in the list of services under the iWay Business Services node.
   c. Provide a description for the service.

6. Click *Next*.

The license and method dialog box appears:



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 descriptive name for the method.

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

7. Click *OK*.

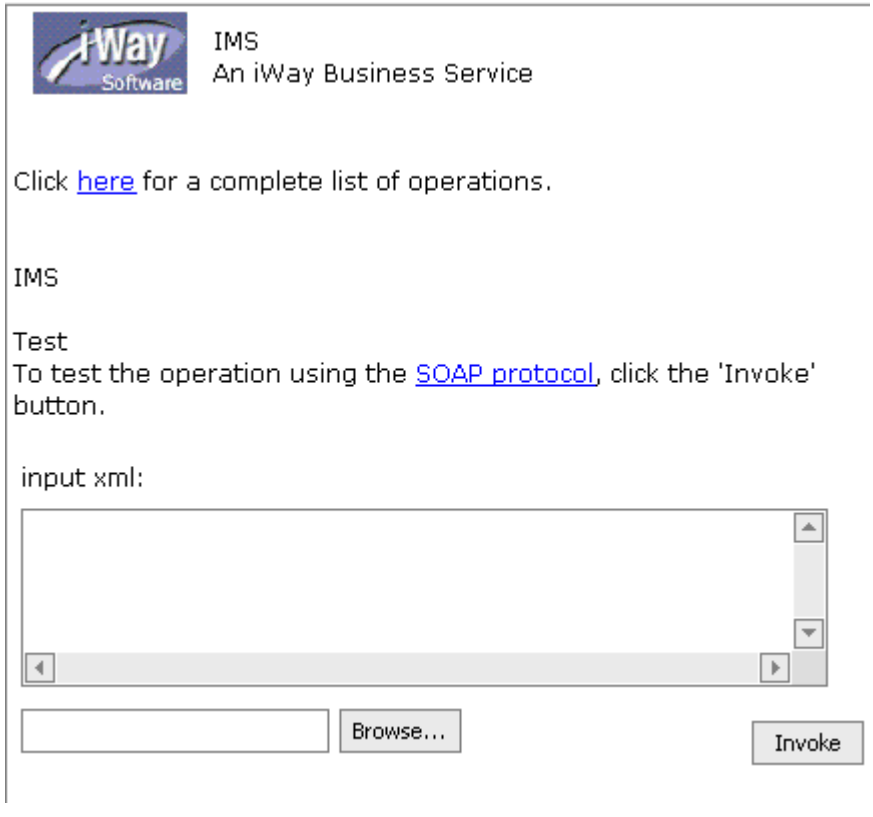
Application Explorer expands the iWay Business Services node in the left pane to show the newly created Web service and presents a test input area in the right pane.

Testing the Web Service

When the Web service has been created, use the test tool to ensure that it functions properly.

Procedure How to Test the Web Service

Upon creating a Web service (for example, IMS), the following test input area appears:



1. In the input xml field, either enter a sample XML document that queries the service, for example,

```
<?xml version="1.0" encoding="UTF-8" ?>
<IMS>
 <Transaction location="/IMS/Transaction/PART">
   <message>AN960C10</message>
 </Transaction>
</IMS>
```

   or browse to the location of an XML instance and click *Open*.

2. Click *Invoke*.

   The result appears 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 a *.wsdl* file extension.
5. Click *Save*.

Identity Propagation

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

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

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

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

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

Adding a Control for an iWay Resource in BEA WebLogic Workshop

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

Adding a Web Service Control to a BEA WebLogic Workshop Application

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

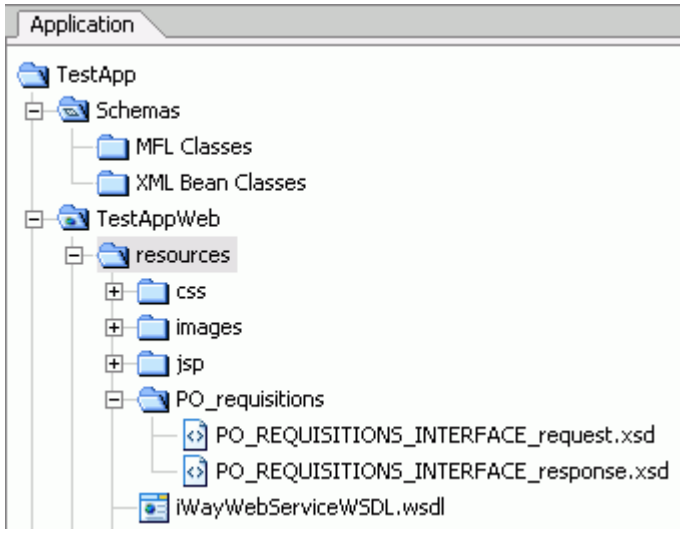
For more information on exporting a WSDL file, see *How to Generate WSDL From a Web Service* on page A-22.

Procedure How to Add a Web Service Control

To add a Web service control:

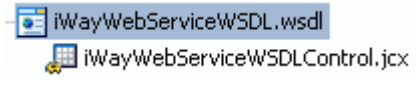
1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

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



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

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



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

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

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

Overview

The extensible iWay CCI control provides:

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

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

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

where:

ResponseDataType

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

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

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

VariableName

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

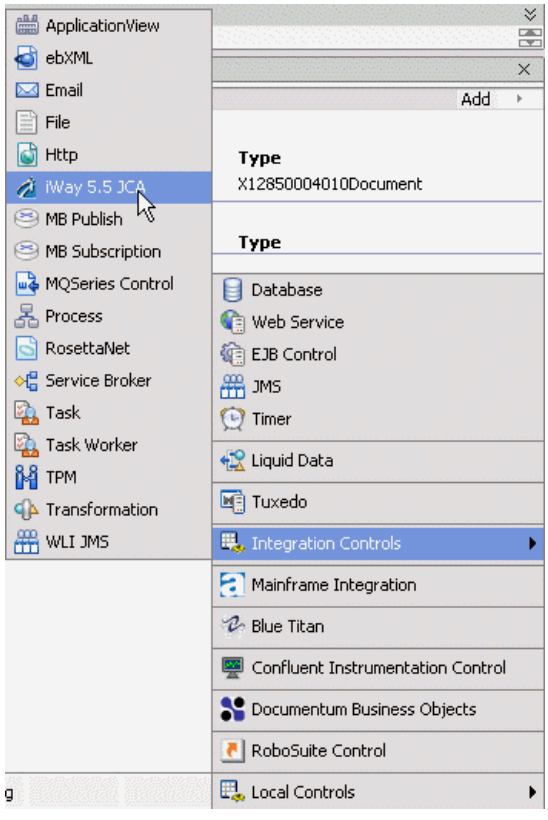
In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control* on page A-26.

You can also use dynamic class casting to specify schema-based input or output XmlObjects to be cast into a pure XmlObject as a service method, which is expected by the CCI control. For more information, see *Using Dynamic Class Casting* on page A-32.

Example Defining a Control Using the Extensible CCI Control

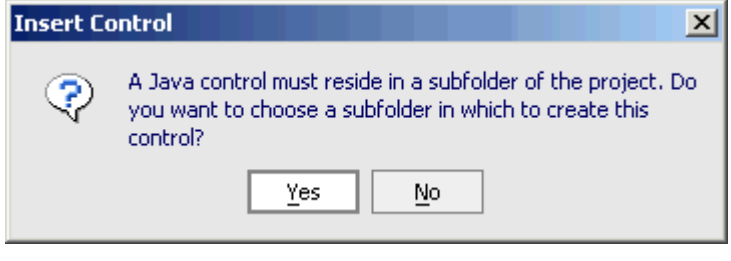
The following sample JCX demonstrates how to define a control for IMS using the extensible CCI control in BEA WebLogic Workshop.

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



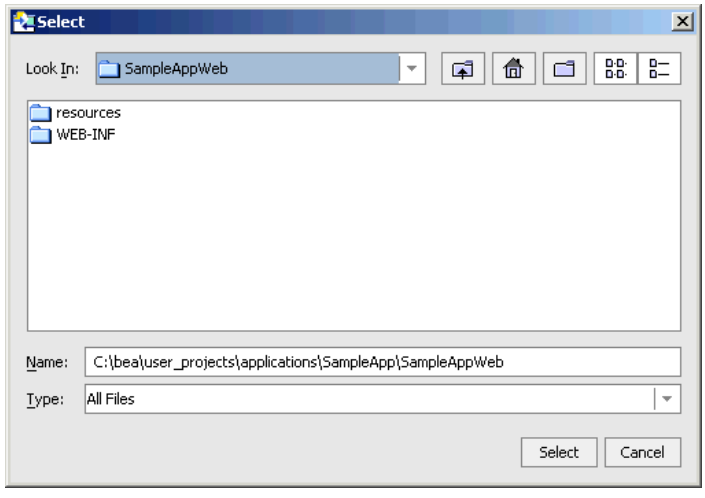
2. Click *Add* from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

   The Insert Control message box opens.



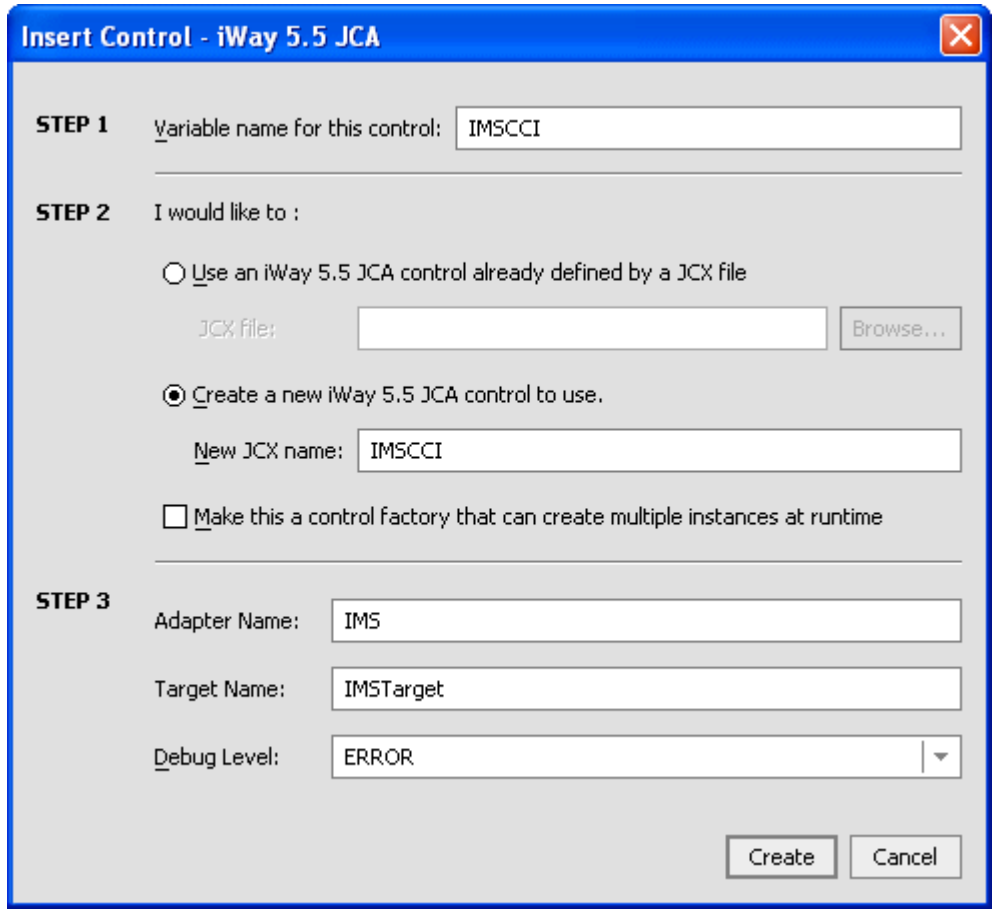
3. Click *Yes*.

The Select dialog box opens.



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

The Insert Control - iWay 5.5 JCA dialog box opens.



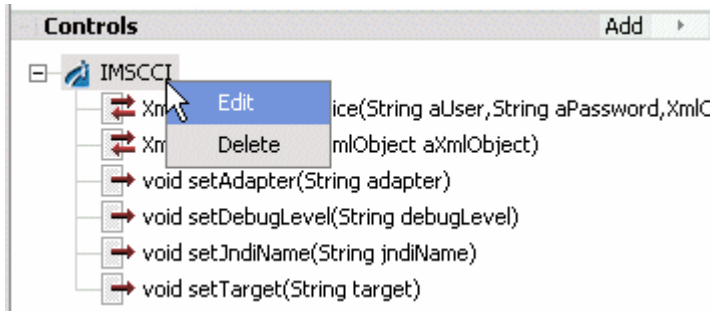
a. Provide a variable name for the control.

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

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

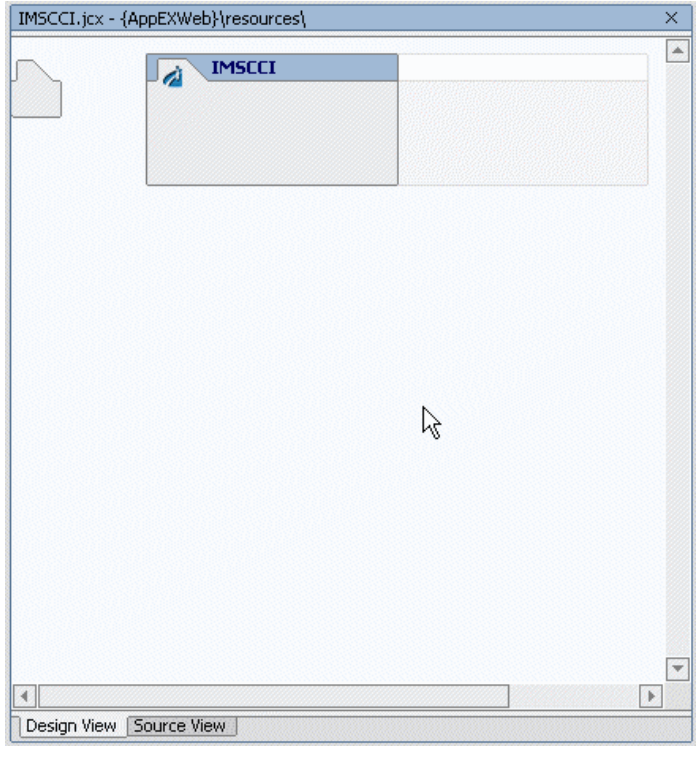
5. Click *Create*.

A new JCX file is created.



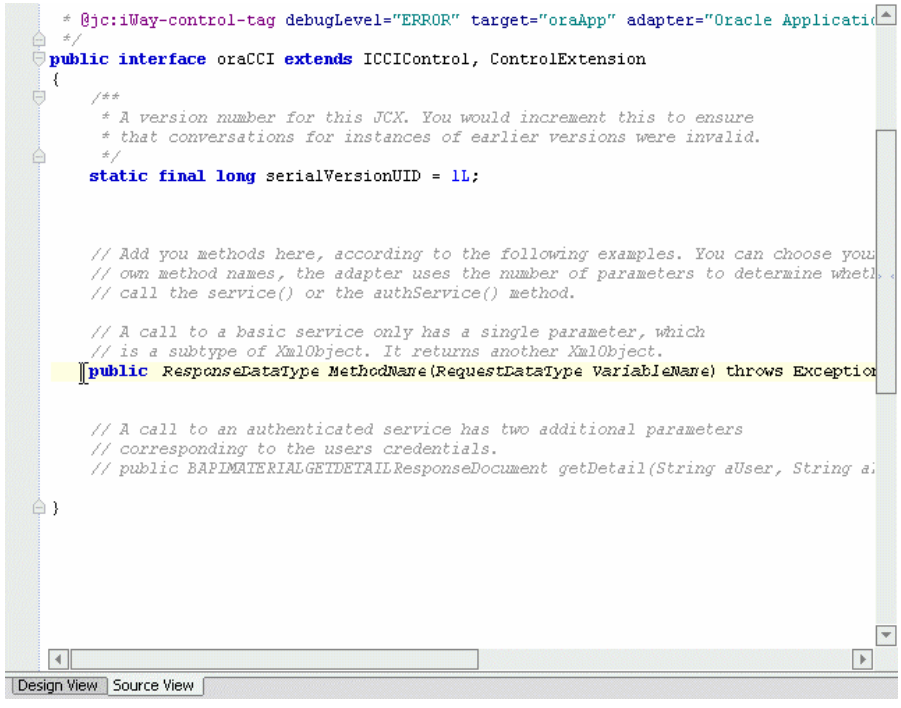
6. Right-click the control, for example, IMSCCI, and select *Edit*.

The Design View for the control opens.



7. Click the *Source View* tab.

The Source View for the control opens.



Perform the following steps:

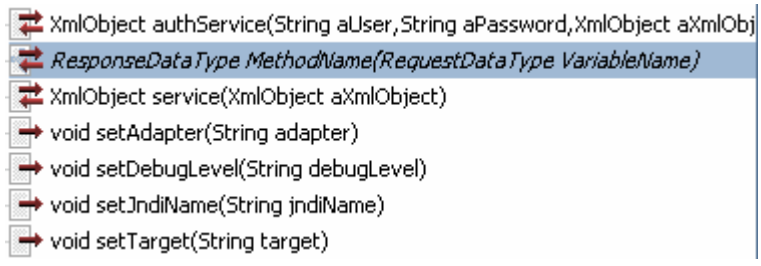
a. Uncomment the public class definition.

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

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

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

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



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

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

Using the Extensible CCI Control

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

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

For example, instead of the generic XmlObject:

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

you call:

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

where:

ResponseDataType

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

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

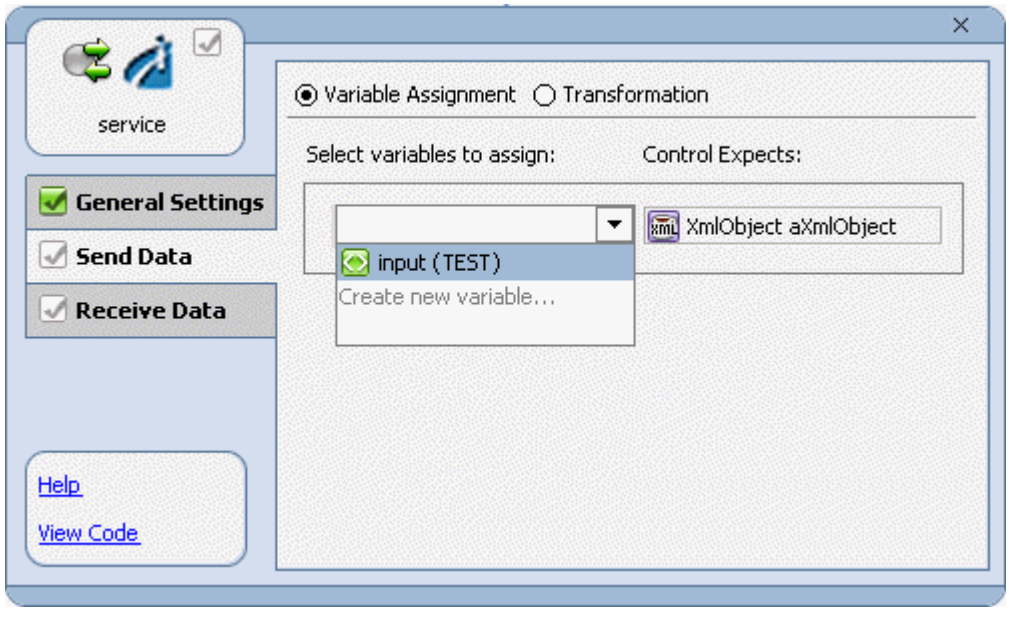
Is the XML Bean Class value that is generated from the request schema.

VariableName

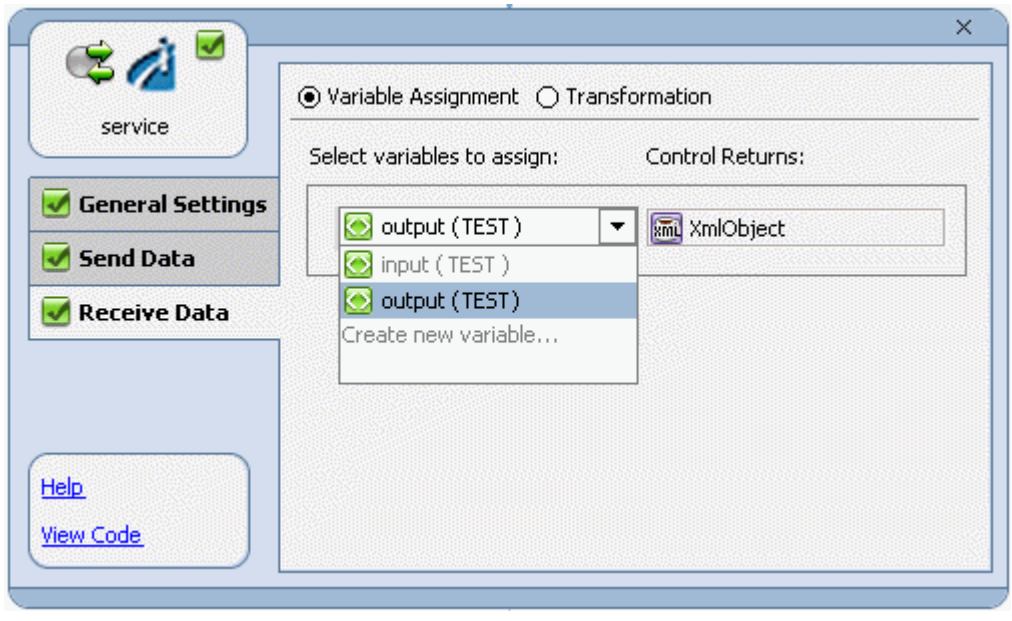
Is the request variable that stores the request document, which is used as input by the extensible CCI control.

Example Using Dynamic Class Casting

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



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



APPENDIX B

Running the Adapter Using LU6.2 Communication

Topics:

- MVS OS/390 APPC Communication
- Microsoft SNA Server Communication
- Application Runtime Requirements

This appendix contains technical information that you can use as a guide to ensure LU6.2 communication to the IMS/TM region.

MVS OS/390 APPC Communication

The following topics contain technical information that you can use as a guide to ensure LU6.2 APPC communication to the IMS/TM region.

Use the following information for conversation with VTAM and IMS/TM administrators.

LU6.2 Setup on MVS

To correctly set the APPC/MVS LU definition, use the following VTAM APPC settings as a guide:

```
000083              IM8CAPPC APPL AUTH=(ACQ),                    X
000084              ACBNAME=IM8CAPPC,                            X
000085              DMINWNL=5,                                   X
000086              DMINWNR=5,                                   X
000087              DSESLIM=10,                                  X
000088              EAS=100,                                     X
000089              PARSESS=YES,                                 X
000090              APPC=YES,                                    X
000091              MODETAB=APPCLM,                              X
000092              DLOGMOD=LU62APPC,                            X
000093              SECACPT=ALREADYV

** IMS/TM Remote LU = IM8CAPPC
```

* VTAM Mode table entry for APPC sessions.

```
LU62APPC  MODEENT LOGMODE=LU62APPC,                              X
          FMPROF=X'13',TSPROF=X'07',                             X
          PRIPROT=X'B0',SECPROT=X'B0',                           X
          COMPROT=X'50B5',RUSIZES=X'8585',                       X
          PSERVIC=X'060200000000000000002F00',                   X
          TYPE=X'00'
```

Procedure: How to Verify that the Remote LU is Active

Use the following commands as a guide. Your VTAM or IMS/TM Administrator may have alternate ways to verify that the Remote LU is active.

1. Issue the following command to view the MVS log:

   ```
   =S;LOG
   ```

2. Look for the outstanding message for the IMS/TM region:

   ```
   4200000 IBI1     14.21.04 STC02021 *79 DFS996I *IMS READY*  IM8C
   ```

3. Reply to the outstanding message and display APPC:

   ```
   /79,/DIS APPC
   ```

4. In the IMS/TM Started Task, look for the display output. An example of the active LU at iWay Software is:

   ```
   /DIS APPC LINE 001 PTERM 001.
   .    IMSLU             .APPC-CONV SECURITY STATUS   DESIRED  GRNAME
   TYPE LINE
       USIBINET.IM8CAPPC          0 FULL     ENABLED  ENABLED  ........
   BASE LINE 0
       *04036/142317* LINE 001 PTERM 001.
   ```

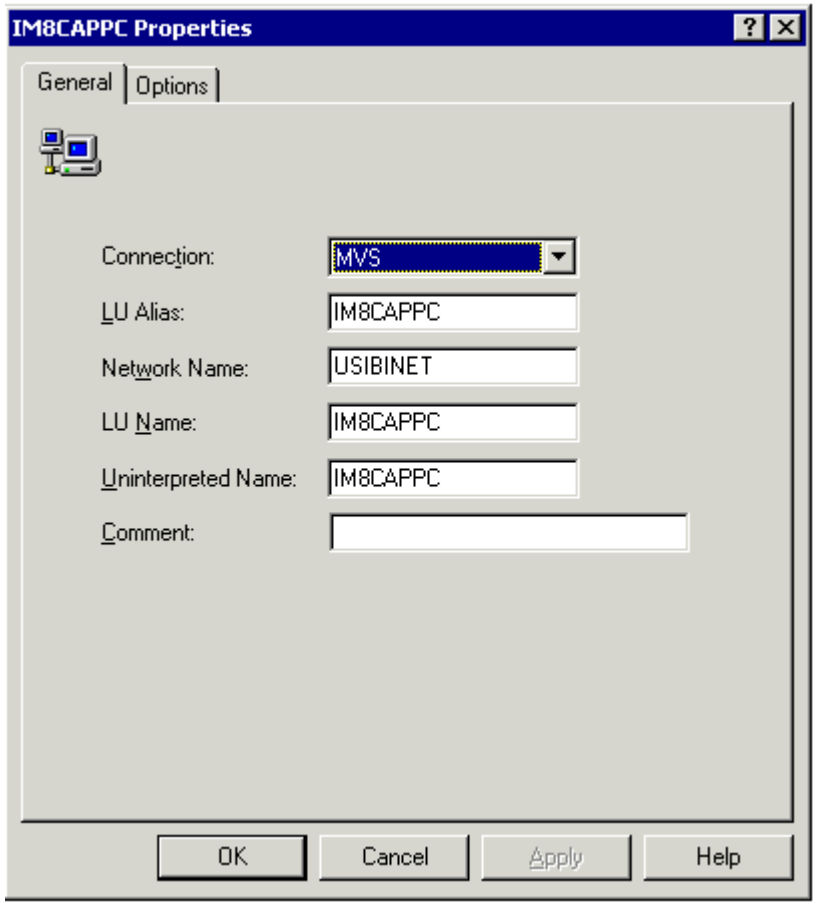
Microsoft SNA Server Communication

This following topics contain technical information that you can use as a guide to ensure Microsoft SNA Server communication to the IMS/TM region.

LU6.2 Setup on a Windows SNA Server

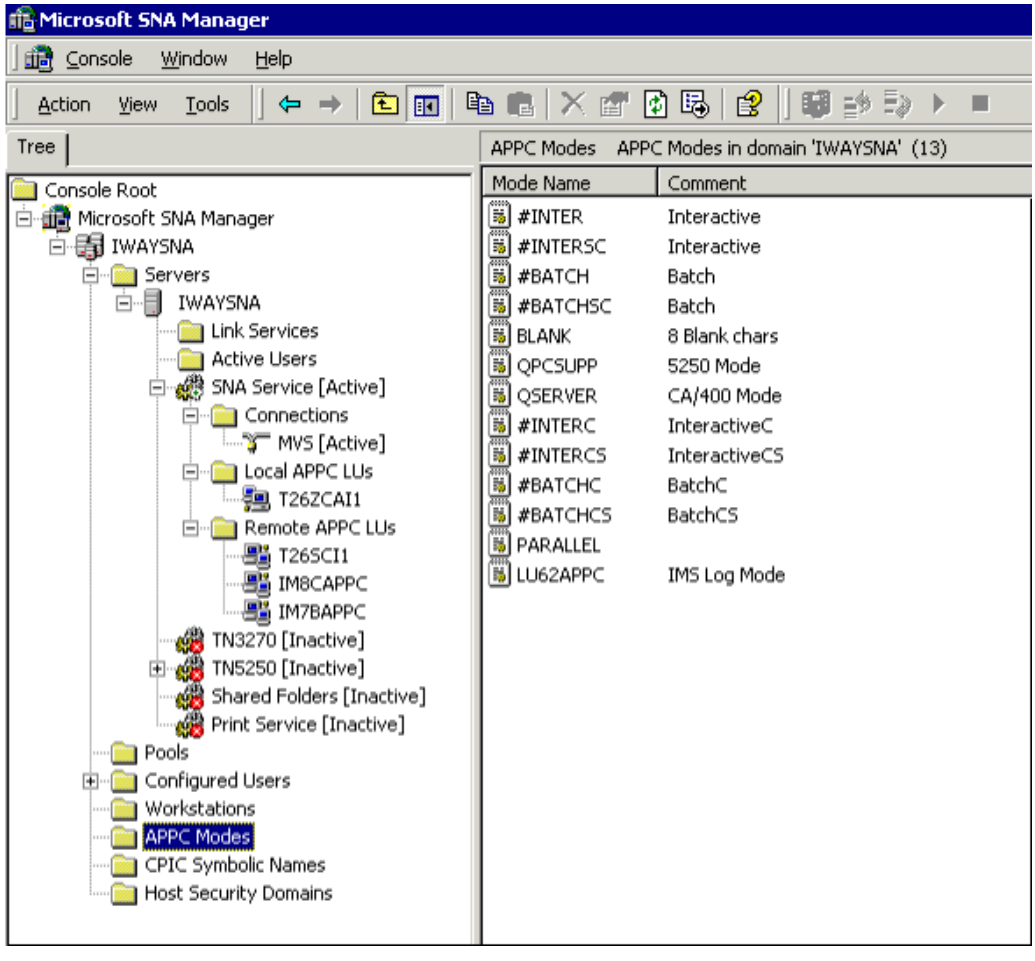
Verify that the SNA Service is active. You must create one or more remote LUs on the SNA Server with equivalent names for the remote LUs for the IMS/TM region(s).

The following illustration displays the Remote LU, IM8CAPPC, and the SNA Manager properties for the Remote LU.



The following image displays the SNA Manager on a Windows NT platform with remote LUs and an active SNA Service.

In this case, two remote LUs for the IMS/TM region have been created, one to access the IMS Version 7.1 region and the other to access the IMS Version 8 region at iWay Software. The remote LU to access the IMS Version 8 region is IM8CAPPC. The SNA Server Local LU is T26ZCAI1.



Application Runtime Requirements

The following are runtime requirements for applications that invoke the IMS/TM Adapter using SNA Services. The adapter engine, iBSE, or any application server must include the required SNA dlls.

WebLogic Server

For WebLogic Server only:

- Install WebLogic Server on the Windows platform where the SNA server is running.
- Verify that the directory containing the MS SNA .dll is included in the system PATH by issuing the SET PATH command at a Windows cmd prompt.

  If the SNA directory is not included in the PATH, set the PATH to include the WIN APPC libraries and add the following to the StartWeblogic.cmd file for the WebLogic Server domain:

  For example:

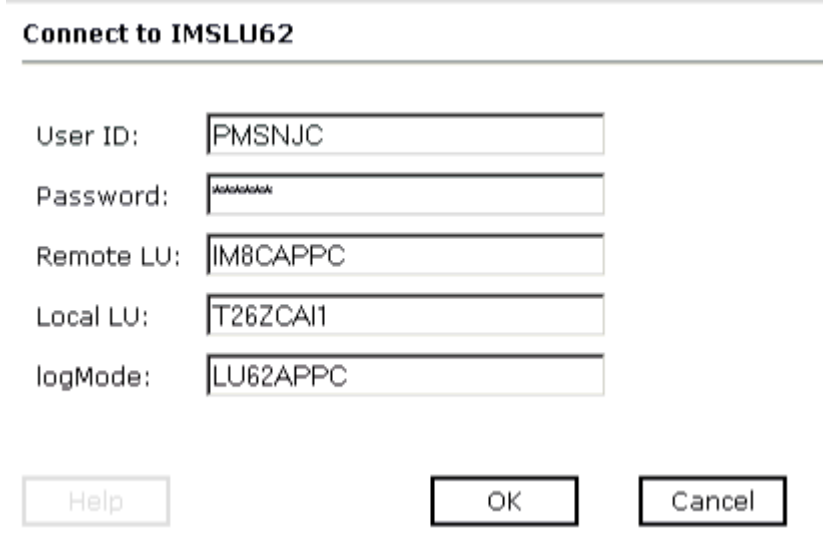
  ```
  SET PATH=C:\Program Files\Host Integration Server\system;C:\program
  files\iway55\lib;%PATH%;
  ```

  Note: Confirm that the directories are echoed back in the WebLogic Server log.

The following image displays the connection information using Application Explorer. Note that the remote LU is the IMS/TM Remote LU (IM8CAPPC) which matches the Remote LU on the SNA server.

The Local LU (T26ZCAI1) matches the SNA server local LU.

LU62APPC logmode is used for the connection. Verify that you specify a valid logmode for the IMS/TM region (VTAM mode table entry for APPC sessions).



APPENDIX C

Sample Requests, Schemas, and Cobol File Descriptions

Topics:

- Request Document to Run PART as a Generic Transaction
- Request Schema for Generic Transaction PART
- Response Schema for Generic Transaction PART
- Request Documents for Adapter Transaction PART
- Request Schema for Adapter Transaction PART
- Response Schema for Adapter Transaction PART
- Sample Cobol File Descriptions for PART

After you create a connection to IMS/TM, you can add IMS/TM transactions using Application Explorer. The generic transaction always is added automatically and represents IMS/TM services whose data will not be mapped to XML.

The following topics illustrate the request and response documents for the transaction, PART. The Cobol descriptions that were used as input for the sample IMS/TM transactions also are shown.

Request Document to Run PART as a Generic Transaction

```
<?xml version="1.0" encoding="UTF-8" ?>
<IMS>
  <Transaction tpname="PART" noreply= "NO">
    <message>
      <message>*</message>
    </message>
  </Transaction>
</IMS>
```

Request Schema for Generic Transaction PART

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-11-04T16:29:41Z -->
<xs:schema xml:lang="en" targetNamespace="urn:iwaysoftware:IMS/
Transactions/Generic_Transaction_Request"
attributeFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/
XMLSchema" elementFormDefault="qualified">
   <xs:element name="IMS">
      <xs:complexType>
         <xs:all>
            <xs:element name="Transaction">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="message">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element minOccurs="1" name="message"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:all>
                  <xs:attribute type="xs:string"
                  use="optional" name="hexConv"/>
                  <xs:attribute type="xs:integer"
                  use="optional" name="bufferLimit"/>
                  <xs:attribute type="xs:boolean"
                  use="optional" name="noRepl"/>
                  <xs:attribute type="xs:string"
                  use="required" name="tpname"/>
                  <xs:attribute type="xs:string"
                  use="optional" name="lterm"/>
               </xs:complexType>
            </xs:element>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Response Schema for Generic Transaction PART

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-11-04T16:29:41Z -->
<xs:schema xml:lang="en"
targetNamespace="urn:iwaysoftware:IMS/Transactions/
Generic_Transaction_Response" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="IMS">
      <xs:complexType>
         <xs:all>
            <xs:element name="Transaction">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="message">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element minOccurs="1" name="message"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:all>
               </xs:complexType>
            </xs:element>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Request Documents for Adapter Transaction PART

The following are the sample XML request documents to run the transaction, PART:

PART_All_Request.xml

```
<IMS>
 <Transaction location="IMS/Transactions/Part">
  <message>
   <MESSAGE>*</MESSAGE>
  </message>
 </Transaction>
</IMS>
```

PART_Detail_Request.xml

```
<IMS>
 <Transaction location="IMS/Transactions/Part">
  <message>
   <MESSAGE>PartNumberHere</MESSAGE>
  </message>
 </Transaction>
</IMS>
```

PART_Error_Request.xml

```
<IMS>
 <Transaction location="IMS/Transactions/Part">
  <message>
   <MESSAGE>WillNotFind</MESSAGE>
  </message>
 </Transaction>
</IMS>
```

Request Schema for Adapter Transaction PART

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-11-04T16:44:28Z -->
<xs:schema xml:lang="en" targetNamespace="urn:iwaysoftware:IMS/
Transactions/Part_Request" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="IMS">
      <xs:complexType>
         <xs:all>
            <xs:element name="Transaction">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="message">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element minOccurs="1" name="MESSAGE"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:all>
                  <xs:attribute type="xs:string" use="required"
                  fixed="IMS/Transactions/Part" name="location"/>
               </xs:complexType>
            </xs:element>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Response Schema for Adapter Transaction PART

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-11-04T16:44:28Z -->
<xs:schema xml:lang="en" targetNamespace="urn:iwaysoftware:IMS/
Transactions/Part_Response" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="IMS">
      <xs:complexType>
         <xs:all>
            <xs:element name="Transaction">
               <xs:complexType>
                  <xs:choice>
                     <xs:element name="message1">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element minOccurs="1"
                              name="FILL" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="RECTYPE" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL1" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="PARTNUMBER" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL2" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="DESCRIPTION" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL3" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="PROCCODE" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL4" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="INVCODE" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL5" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="MAKEDEPT" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL6" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="PREVNO" maxOccurs="1"/>
                              <xs:element minOccurs="1"
                              name="FILL7" maxOccurs="1"/>
                              <xs:element minOccurs="1"
```

```
                                    name="MAKETIME" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="FILL8" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="CCODE" maxOccurs="1"/>
                                 </xs:sequence>
                              </xs:complexType>
                           </xs:element>
                           <xs:element name="message2">
                              <xs:complexType>
                                 <xs:sequence>
                                    <xs:element minOccurs="1"
                                    name="FILL" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="RECTYPE" maxOccurs="1"/>
                                 </xs:sequence>
                              </xs:complexType>
                           </xs:element>
                           <xs:element name="message3">
                              <xs:complexType>
                                 <xs:sequence>
                                    <xs:element minOccurs="1"
                                    name="REC-NUM" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="RECTYPE" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="PART-NUM" maxOccurs="1"/>
                                    <xs:element minOccurs="1"
                                    name="PART-DESC" maxOccurs="1"/>
                                 </xs:sequence>
                              </xs:complexType>
                           </xs:element>
                        </xs:choice>
                     </xs:complexType>
                  </xs:element>
               </xs:all>
            </xs:complexType>
         </xs:element>
      </xs:schema>
```

Sample Cobol File Descriptions for PART

The following sample Cobol File Description is used as input for the IMS/TM transactions in Chapter 3, *Designing the Adapter*.

PART_IN.cbl

```
01  PARTIN.
    05  MESSAGE                PIC X(80)   VALUE SPACE.
```

PART_All_Out.cbl

```
01  PARTALL.
    05  FILL                   PIC X(3).
    05  RECTYPE                PIC X(1).
    05  MESSAGE                PIC X(76).
```

PART_Detail_Out.cbl

```
01  PARTDETAIL.
    05  FILL                   PIC X(3)   VALUE SPACE.
    05  RECTYPE                PIC X(1)   VALUE SPACE.
    05  FILL1                  PIC X(22)  VALUE SPACE.
    05  PARTNUMBER             PIC X(12)  VALUE SPACE.
    05  FILL2                  PIC X(18)  VALUE SPACE.
    05  DESCRIPTION            PIC X(20)  VALUE SPACE.
    05  FILL3                  PIC X(26)  VALUE SPACE.
    05  PROCCODE               PIC X(12)  VALUE SPACE.
    05  FILL4                  PIC X(18)  VALUE SPACE.
    05  INVCODE                PIC X(8)   VALUE SPACE.
    05  FILL5                  PIC X(26)  VALUE SPACE.
    05  MAKEDEPT               PIC X(12)  VALUE SPACE.
    05  FILL6                  PIC X(18)  VALUE SPACE.
    05  PREVNO                 PIC X(8)   VALUE SPACE.
    05  FILL7                  PIC X(26)  VALUE SPACE.
    05  MAKETIME               PIC X(12)  VALUE SPACE.
    05  FILL8                  PIC X(18)  VALUE SPACE.
    05  CCODE                  PIC X(8)   VALUE SPACE.
```

PART_Error_Out.cbl

```
01  PARTERROR.
    05  FILL                   PIC X(3)   VALUE SPACE.
    05  RECTYPE                PIC X(1)   VALUE SPACE.
    05  MESSAGE                PIC X(46)  VALUE SPACE.
```

APPENDIX D

Debugging and Troubleshooting

Topic:

- Troubleshooting

This appendix includes tips and techniques for debugging the adapter.

Troubleshooting

Certain situations may cause the adapter to return error messages. This section describes error messages, their possible causes, and their solutions:

```
Unable to process request:java.lang.IllegalStateException: Error reading
segment ret: 4 res: Datastore not found
```

Accompanied by:

```
IMS Connect Message:
Message failed, ORIGIN=6686 _35775912 to DESTID=IMS8A ; R=4, S=NF
```

Possible Cause: Datastore is not active.

Solution: Start or cycle imsconnect job.

Reader Comments

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

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

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

Name:_____

Company:_____

Address:_____

Telephone:_____Date:_____

E-mail:_____

Comments:

Reader Comments

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

iWay Transaction Adapter for IMS/TM for Java Technologies for BEA WebLogic User's Guide
DN3501332.0205