

iWay

iWay Servlet Application Explorer User's Guide
Version 5 Release 5

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

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

Copyright © 2004, by Information Builders, Inc and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Preface

This document explains how to use the iWay Servlet Application Explorer to create XML schemas and Web services for use in conjunction with iWay adapters.

How This Manual Is Organized

This manual includes the following chapters:

Chapter/Appendix		Contents
1	<i>Introducing the iWay Servlet Application Explorer</i>	Provides an overview of the iWay Servlet Application Explorer and its features.
2	<i>Creating Targets for Enterprise Information Systems</i>	Describes how to start the iWay Servlet Application Explorer and create and modify a new target for an Enterprise Information System.
3	<i>Browsing Metadata and Creating XML Schemas</i>	Describes how to use the iWay Servlet Application Explorer to browse metadata and create XML schemas.
4	<i>Creating and Publishing iWay Business Services</i>	Describes how to use the iWay Servlet Application Explorer to create and publish an iWay Business Service.
5	<i>Using iWay Event Adapters</i>	Describes how to use the iWay Servlet Application Explorer (iAE) to connect to an Enterprise Information System (EIS) and generate iWay Events.
A	<i>Using the WebLogic Workshop to Access Web Services</i>	Describes how to access Web services created for SAP R/3 BAPIs and RFCs using the WebLogic Workshop.

Customer Support

Do you have questions about the iWay Servlet Application Explorer?

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 Servlet Application Explorer 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.

Information You Should Have

To help our consultants answer your questions most effectively, be ready to provide the following information when you call:

- Your six-digit site code number (xxxx.xx).
- Your iWay Software configuration:
 - The iWay Software version and release.
- Your application or database release level, including the minor release level (for example, Version 1.2.3...).
- The XML schemas and XML document instances that are used.
- The exact nature of the problem:
 - Are the results or the format incorrect?
 - The error message and return code, if applicable.
 - Is this related to any other problem?
- Has the procedure ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, your security system, communications protocol, or any of the software being used changed?
- Is this problem reproducible? If so, how?
- Have you tried to reproduce your problem in the simplest form possible?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes any opinion you can offer regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Documentation Feedback form on our Web site, <http://www.iwaysoftware.com>.

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

1. Introducing the iWay Servlet Application Explorer	1-1
Key Features	1-2
Installing and Configuring the iWay Servlet Application Explorer	1-2
2. Creating Targets for Enterprise Information Systems	2-1
Starting the iWay Servlet iWay Application Explorer	2-2
Connecting to an Enterprise Information System	2-3
Modifying Targets	2-14
3. Browsing Metadata and Creating XML Schemas	3-1
Viewing Application System Objects	3-2
Creating XML Schemas	3-6
4. Creating and Publishing iWay Business Services	4-1
Understanding iWay Business Services	4-2
Creating iWay Business Services	4-2
Generating WSDL From a Web Service	4-9
5. Using iWay Event Adapters	5-1
Understanding iWay Event Functionality	5-2
Creating Event Ports	5-2
Creating Channels	5-11
A. Using the WebLogic Workshop to Access Web Services	A-1
Using the WebLogic Workshop to Access SAP R/3 BAPIs	A-2
Running the JWSNAME Web Service from WebLogic Workshop for BAPIs	A-8
Using the WebLogic Workshop to Access SAP R/3 RFCs	A-12
Running the JWSNAME Web Service from WebLogic Workshop for RFCs	A-18

CHAPTER 1

Introducing the iWay Servlet Application Explorer

Topics:

- Key Features
- Installing and Configuring the iWay Servlet Application Explorer

The iWay Servlet Application Explorer (iAE) is a Web application running within a servlet container that is accessible through a Web browser.

It uses adapters to create schemas and business services for use with the iWay JCA Connector or iWay Business Services Engine (iBSE).

The following section provides an overview of the iAE and its features.

Key Features

The iWay Servlet Application Explorer (iAE) supports access to several Enterprise Information Systems (EIS). Even though the underlying technology used to access them varies significantly, the iAE user interface varies only slightly to accommodate EIS differences.

The iAE uses vendor-provided interfaces and in-depth knowledge of application systems to access and browse business object metadata. Once an object is selected, the iAE can generate an XML schema or Web service to define the object for use in conjunction with an iWay adapter.

External applications accessing another EIS or database via one of the iWay adapters use either the XML schema or Web service to pass data between the external application and the adapter.

Key features of the iAE include:

- The ability to connect to and explore a variety of application systems.
- Access to application system object metadata.
- A point-and-click process for generating XML schemas and Web services.

Installing and Configuring the iWay Servlet Application Explorer

The iAE is packaged as an archive located in the following directory:

`drive:\iWay55\etc\setup\iwaewar`

The iwaewar file must be deployed through a servlet container or application server (for example, BEA WebLogic).

In addition, the target EIS (for example, SAP R/3) must be installed, configured, and available for client access. The iAE does not need to reside on the same system as the application system being accessed, but network access is required.

For more information on installing and configuring the iWay Servlet Application Explorer, see the *iWay 5.5 Installation and Configuration for BEA WebLogic Version 5 Release 5* documentation.

CHAPTER 2

Creating Targets for Enterprise Information Systems

Topics:

- Starting the iWay Servlet iWay Application Explorer
- Connecting to an Enterprise Information System
- Modifying Targets

This section describes how to start the iWay Servlet Application Explorer and create and modify a new target for an Enterprise Information System (EIS).

Starting the iWay Servlet iWay Application Explorer

Before you can use the iWay Servlet Application Explorer (iAE) , you must start the BEA WebLogic Server. To start the BEA WebLogic Server on Windows, click the *Windows Start menu*, select *Programs, BEA WebLogic Platform 8.1, User Projects, your domain for iWay*, and click *Start Server*. If you are using UNIX or want to start the BEA WebLogic Server from a command line, type the following at the prompt:

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

where:

BEA_HOME

Is the directory where BEA WebLogic is installed.

DOMAIN_NAME

Is the domain you are using for iWay.

Once BEA WebLogic Server is running, enter the following URL in your browser window:

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

where:

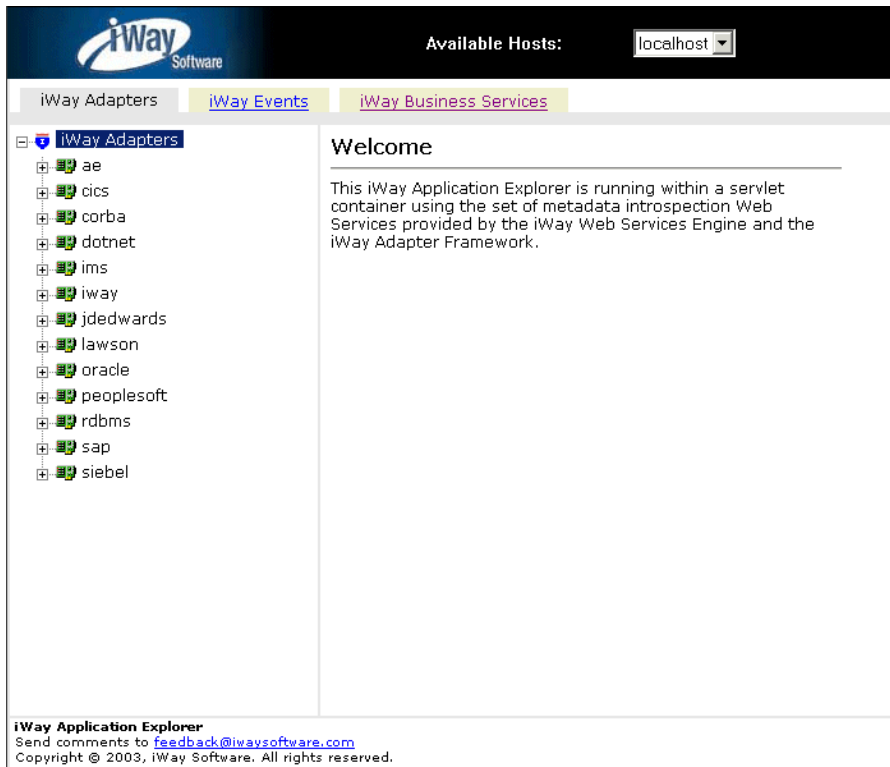
hostname

Is the hostname for your application server.

port

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

When you start the iAE, the following window opens.



The Available Hosts drop-down menu in the upper right determines which iWay JCA Connector or Servlet iBSE instance you can access. For more information on adding instances, see the *iWay 5.5 Installation and Configuration for BEA WebLogic Version 5 Release 5* documentation.

You are now ready to create new targets to Enterprise Information Systems.

Connecting to an Enterprise Information System

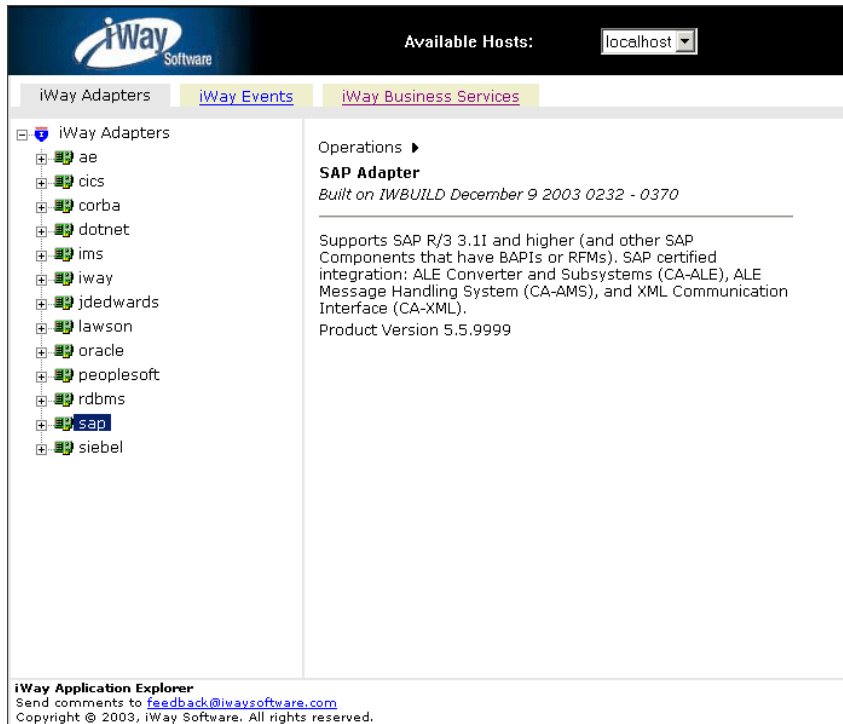
To browse an application system's business objects, you must create a target for the system you intend to use. This target serves as your connection point and is automatically saved after it is created. You must establish a connection to the system every time you start the iWay Application Explorer (iAE) or after you disconnect from the system.

The left pane displays the application systems supported by the iAE, which are based on the iWay adapters you installed and are licensed to use.

Procedure How to Create a New Target

To connect to an application system for the first time, you must create a new target. For example, to create a target for SAP R/3, perform the following steps:

1. Click the **SAP** node in the left pane.

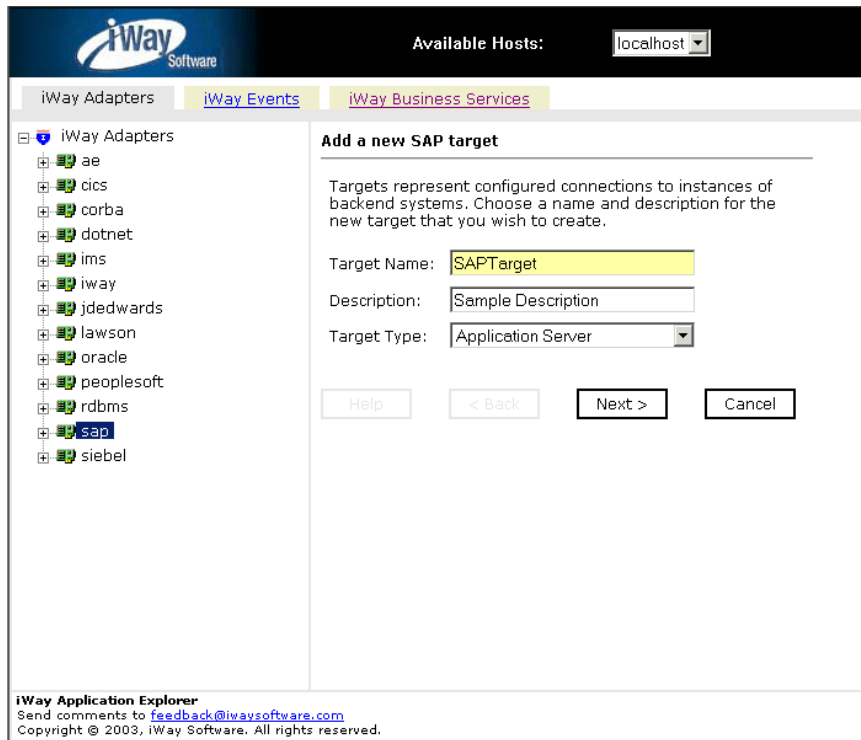


Descriptive information (for example, title and product version) regarding the iWay Adapter and application system to which you are connecting displays in the right pane.

2. Click **Operations** in the right pane and select *Define a new target*.



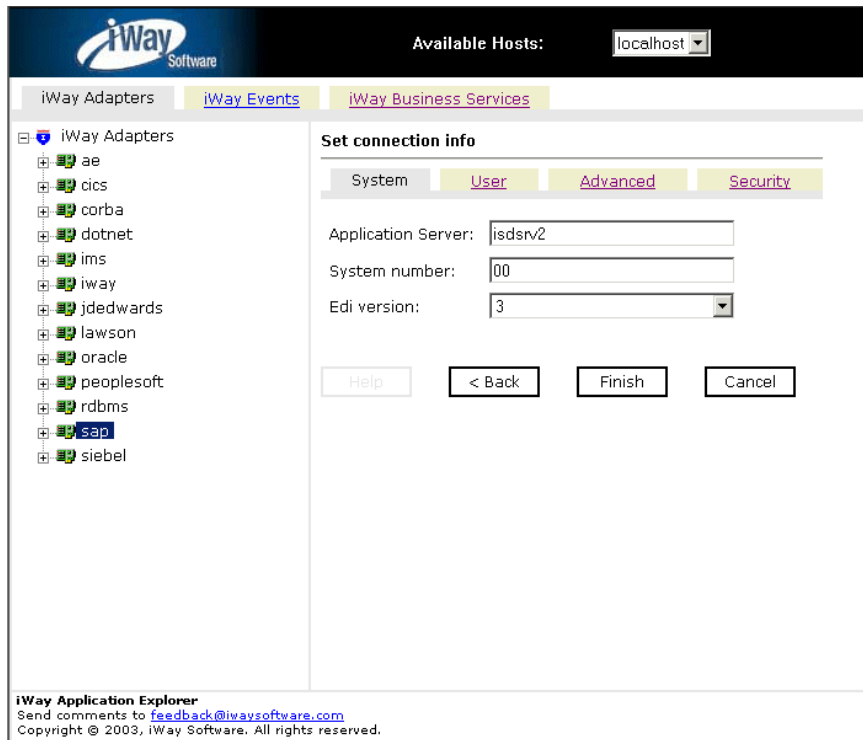
The Add a new SAP target window displays in the right pane.



3. Specify the following information that is specific to the target you are defining:
 - **Target Name.** Enter a descriptive name for the target (for example, SAPTarget).
 - **Target Description.** Enter a brief description for the connection.
 - **Target Type.** Select the type of target you are connecting to from the drop-down list. The default value is *Application Server*.

4. Click *Next*.

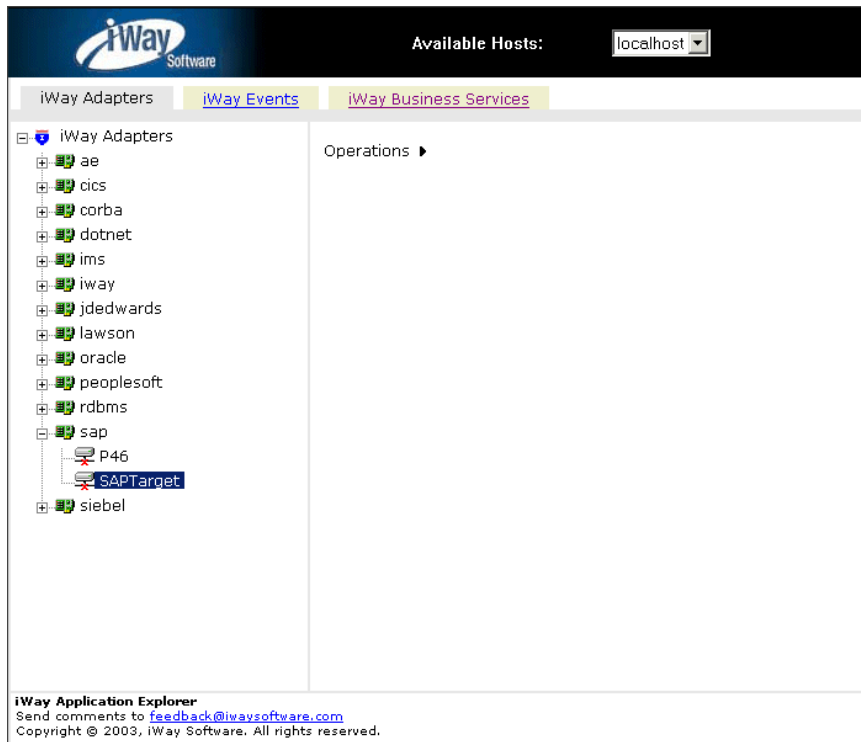
The Set connection info window displays in the right pane.



The various fields and tabs that display in the Set connection info window are specific to each EIS and its application server. For more information, see the iWay Adapter documentation for the specific EIS.

5. Specify the connection information that is specific for the EIS to which you want to connect.
6. Click *Finish*.

The target (SAPTarget) is now listed beneath the SAP node in the left pane.

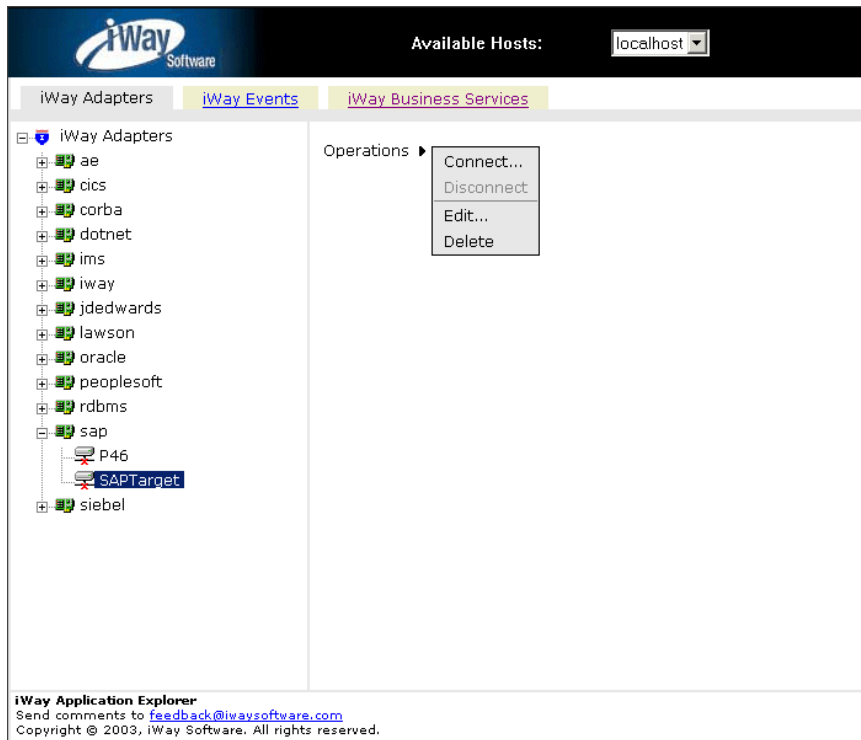


You are now ready to connect to the application target you defined.

Procedure How to Connect to a Target

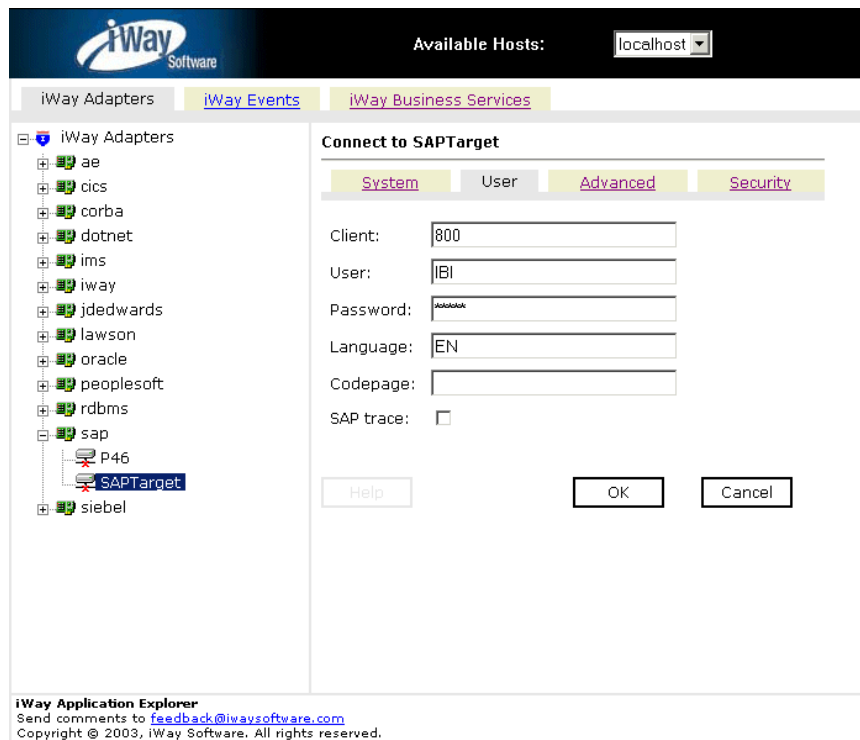
You must use the target you defined to connect to an application. For example, to connect to SAP R/3, perform the following steps.

1. Expand the SAP node in the left pane and select the target you defined (for example, SAPTarget).



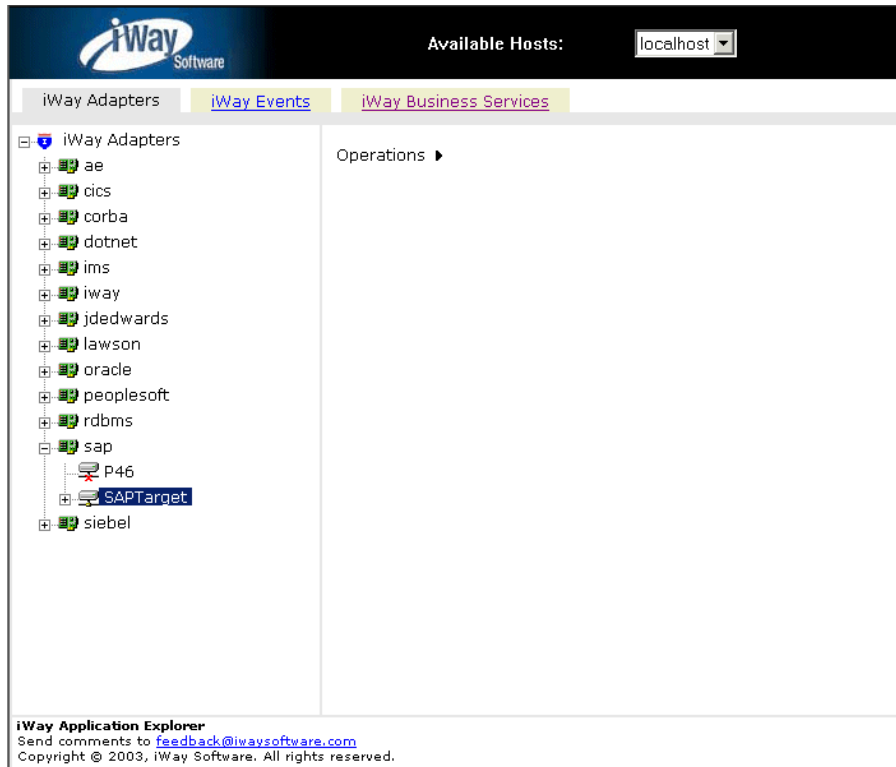
2. Click *Operations* in the right pane and select *Connect*.

The Connect to SAPTarget window opens in the right pane.

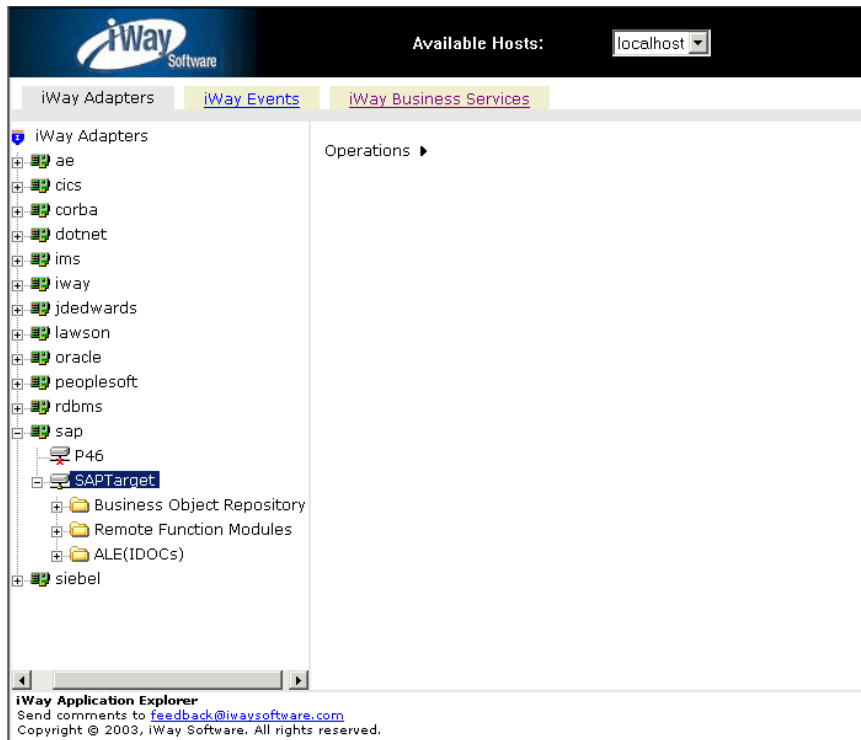


3. Enter a valid password and click *OK*.

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



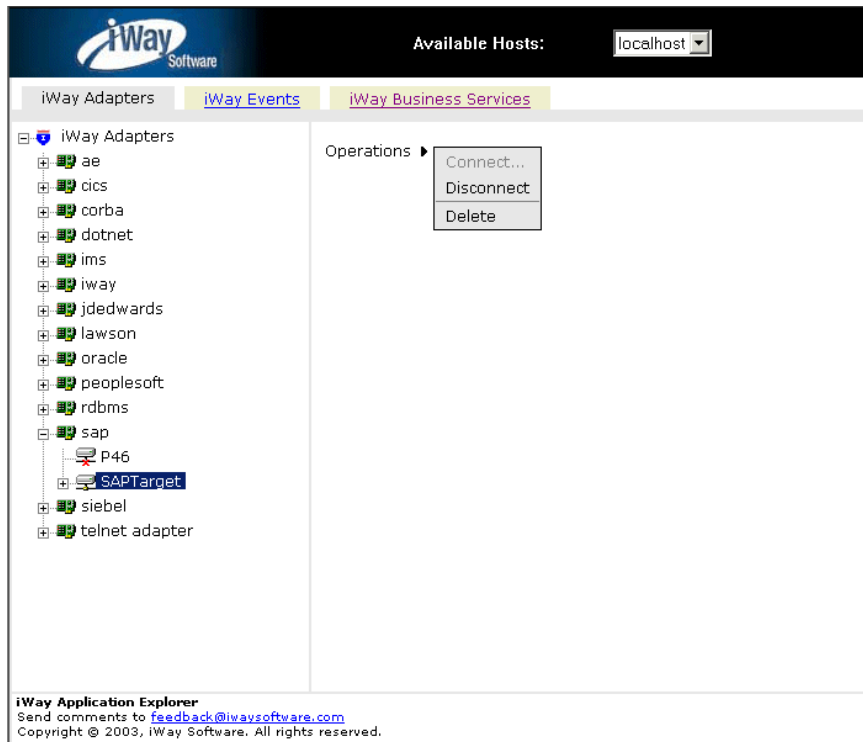
4. Expand the target node to reveal the application system's business objects.



Procedure How to Disconnect From a Target

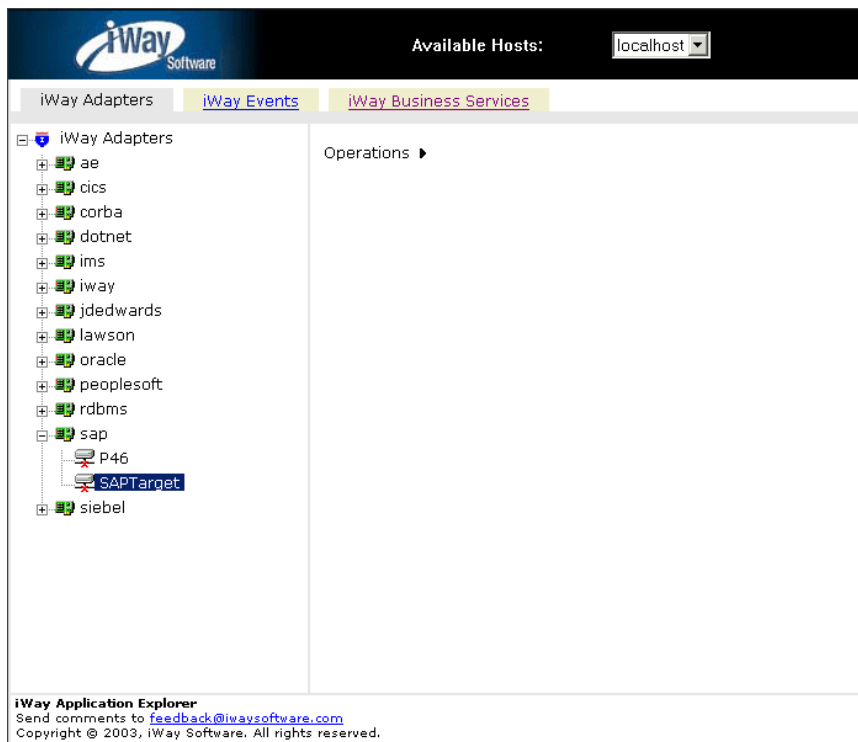
Although you can maintain multiple open connections to different application systems, it is prudent to close connections when they are not being used. Perform the following steps to disconnect from a target.

1. Click the target (for example, SAPTarget) to which you are connected in the left pane.



2. Click *Operations* in the right pane and select *Disconnect*.

Disconnecting from the application system drops the connection, but the node remains. The SAPTarget node in the left pane changes to reflect that a connection was terminated.



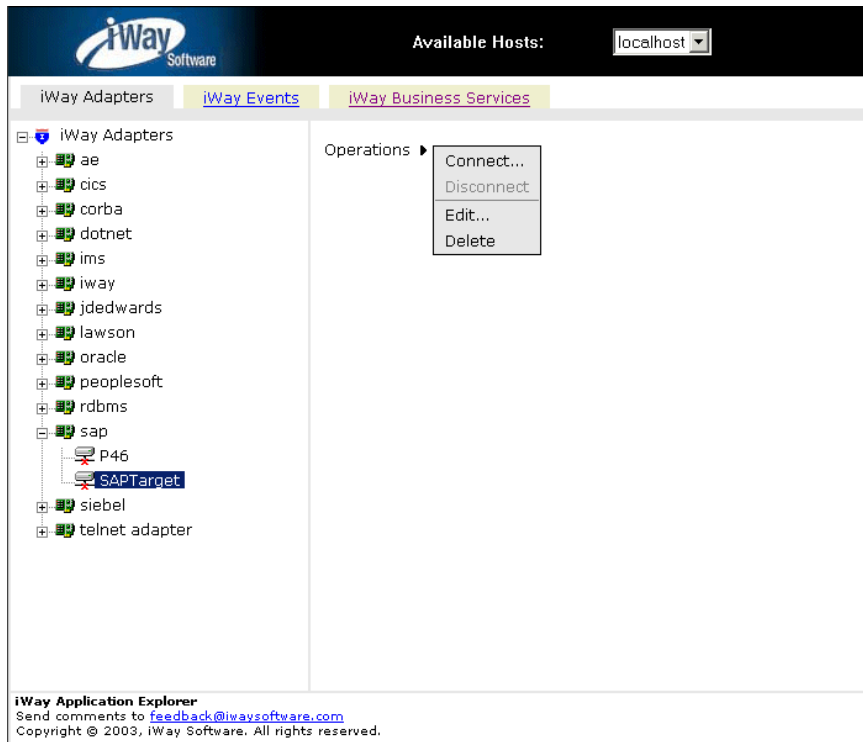
Modifying Targets

After you create a target for an Enterprise Information System using the iWay Servlet Application Explorer, you can edit any information that you provided during the creation process.

Procedure How to Edit a Target

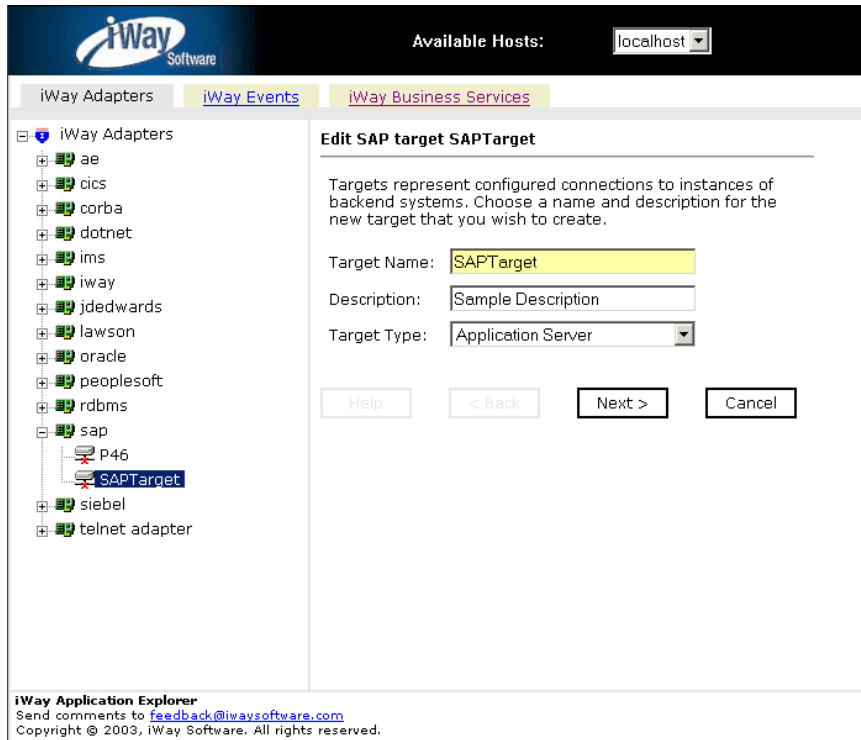
To edit a target in the iWay Servlet Application Explorer:

1. In the left pane, click the target (for example, SAPTarget).



2. In the right pane, click *Operations* and select *Edit*.

The Edit window opens in the right pane.

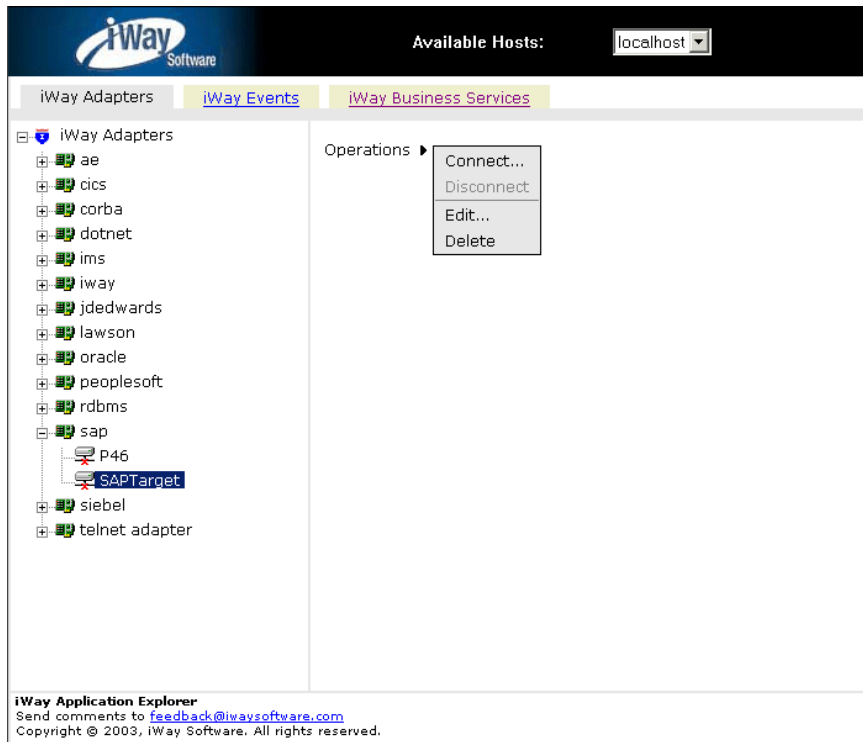


3. Modify the connection information.

Procedure How to Delete a Target

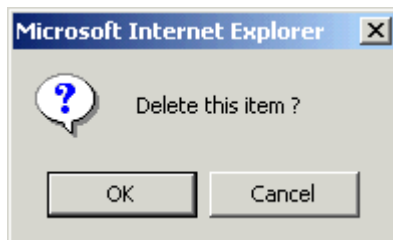
To delete a target in the iWay Servlet Application Explorer:

1. In the left pane, click the target (for example, SAPTarget).



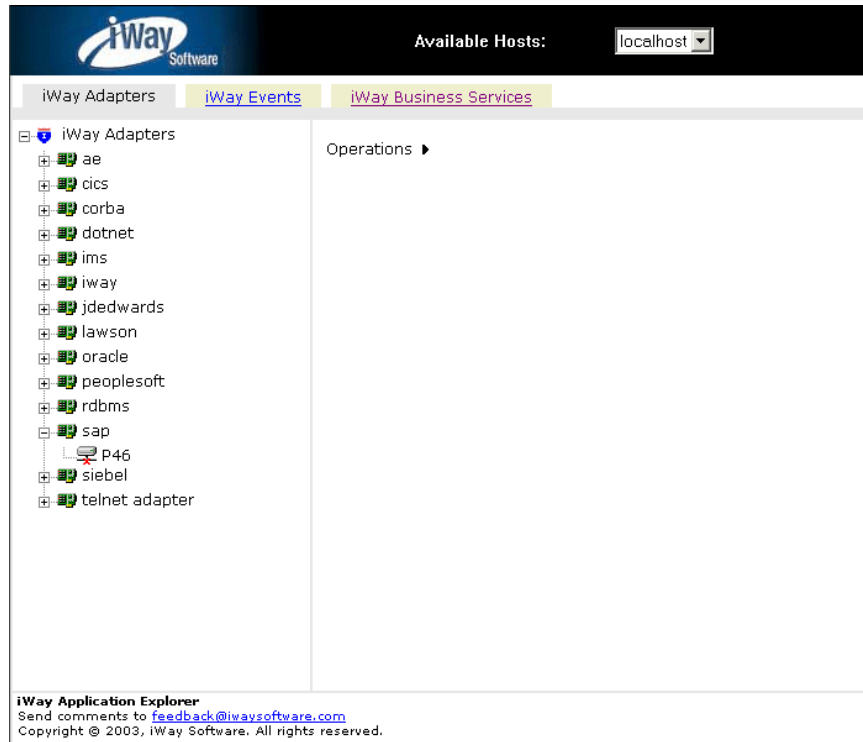
2. In the right pane, click *Operations* and select *Delete*.

The following confirmation dialog box opens.



3. Click *OK* to delete the target you selected.

The SAPTarget node is removed in the left pane.



CHAPTER 3

Browsing Metadata and Creating XML Schemas

Topics:

- Viewing Application System Objects
- Creating XML Schemas

This section describes how to use the iWay Servlet Application Explorer (iAE) to connect to an Enterprise Information System (EIS), browse the business objects within the EIS, and generate XML schemas for the objects that are selected.

The iAE supports access to several Enterprise Information Systems. Even though the underlying technology used to access them varies significantly, the iAE user interface varies only slightly to accommodate EIS differences. In this section, iAE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the iAE and this chapter as reference for use with other systems, depending on the iWay Adapter you are using.

For more specific information on using the iAE with additional systems, see the iWay Adapter documentation for that particular EIS.

Viewing Application System Objects

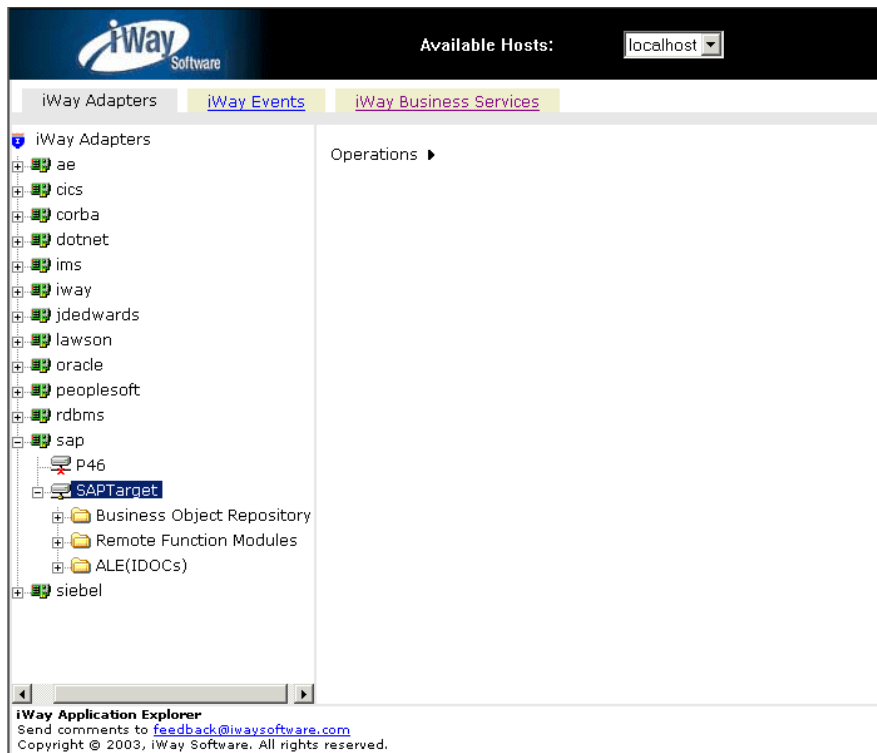
Once you are connected to an EIS, the iWay Servlet Application Explorer allows you to explore and browse business object metadata. For example, the iAE enables you to view SAP R/3 BAPI, RFC, and iDOC metadata stored in the SAP Business Object repository.

Procedure How to View Application System Objects

Perform the following steps to view application system objects.

1. Click on the icon to the left of the target name (for example, SAPTarget).

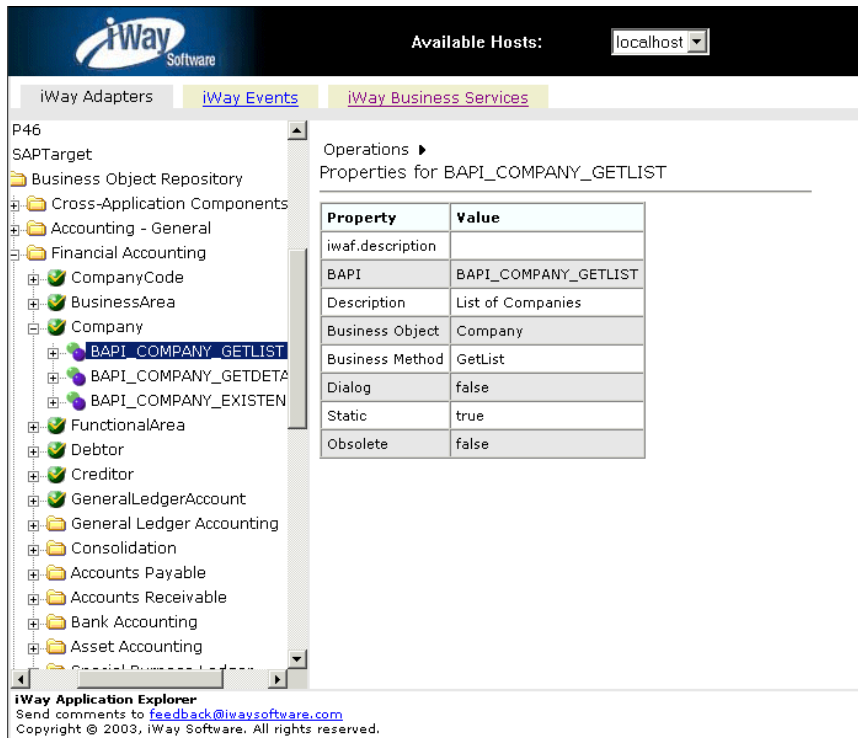
This expands the target to expose the available system objects.



2. Click on the icon to the left of the repository name to expand the desired SAP R/3 repository node.

For example, click the icon to the left of the repository node named *Business Object Repository*.

A list of business object groups display.



3. Click the icon to the left of the *Financial Accounting* group.
A list of business objects related to Financial Accounting display.
4. Scroll down and click on the icon to the left of the *Company* business object.
A list of BAPI methods related to Company display.

5. Scroll down and click on the icon to the left of the BAPI method named BAPI_COMPANY_GETLIST.

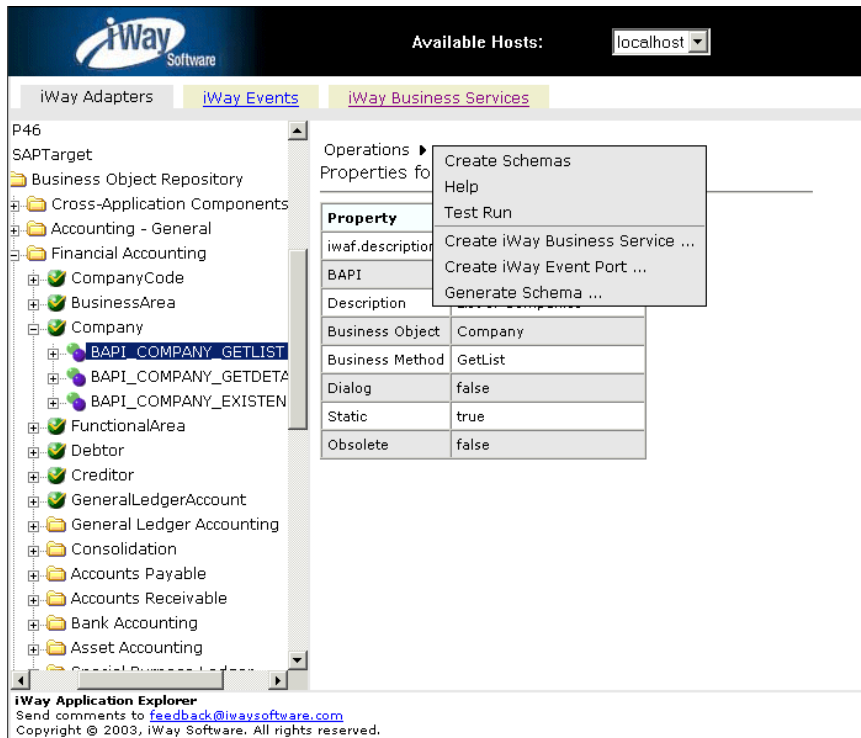
The screenshot shows the iWay Application Explorer interface. The top bar includes the iWay Software logo and an 'Available Hosts' dropdown set to 'localhost'. Below the bar are tabs for 'iWay Adapters', 'iWay Events', and 'iWay Business Services'. The left pane displays a tree structure of application objects. Under 'P46', 'SAPTarget', and 'Business Object Repository', the 'Financial Accounting' folder is expanded, showing sub-objects like 'CompanyCode', 'BusinessArea', 'Company', 'FunctionalArea', 'Debtor', 'Creditor', 'GeneralLedgerAccount', 'General Ledger Accounting', 'Consolidation', 'Accounts Payable', 'Accounts Receivable', 'Bank Accounting', and 'Asset Accounting'. The 'BAPI_COMPANY_GETLIST' method is selected. The right pane shows the 'Operations' section with a sub-section 'Properties for BAPI_COMPANY_GETLIST'. This section contains a table with the following data:

Property	Value
iwaf.description	
BAPI	BAPI_COMPANY_GETLIST
Description	List of Companies
Business Object	Company
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

At the bottom of the window, the text reads: 'iWay Application Explorer', 'Send comments to feedback@iwaysoftware.com', and 'Copyright © 2003, iWay Software. All rights reserved.'

Properties for the BAPI method named BAPI_COMPANY_GETLIST display in a table in the right pane.

6. Click *Operations* in the right pane.



The following options are available from the context menu:

- Create Schemas.
- Help.
- Test Run.
- Create iWay Business Service.
- Create iWay Event Port.
- Generate Schema.

Note: Since each application system's business objects have different properties, the context menu options that are available may vary. For more specific information on using the iAE with additional systems, see the iWay Adapter documentation for that particular EIS.

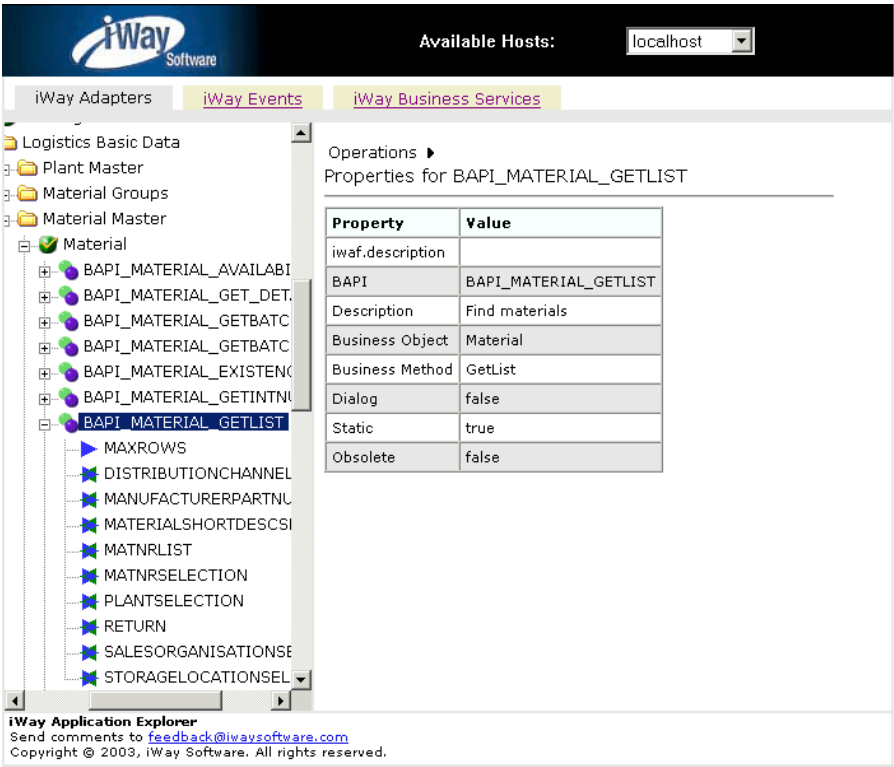
Creating XML Schemas

Once you have browsed the application system business object repository, you can generate an XML request and response schema for the object you wish to use with your adapter.

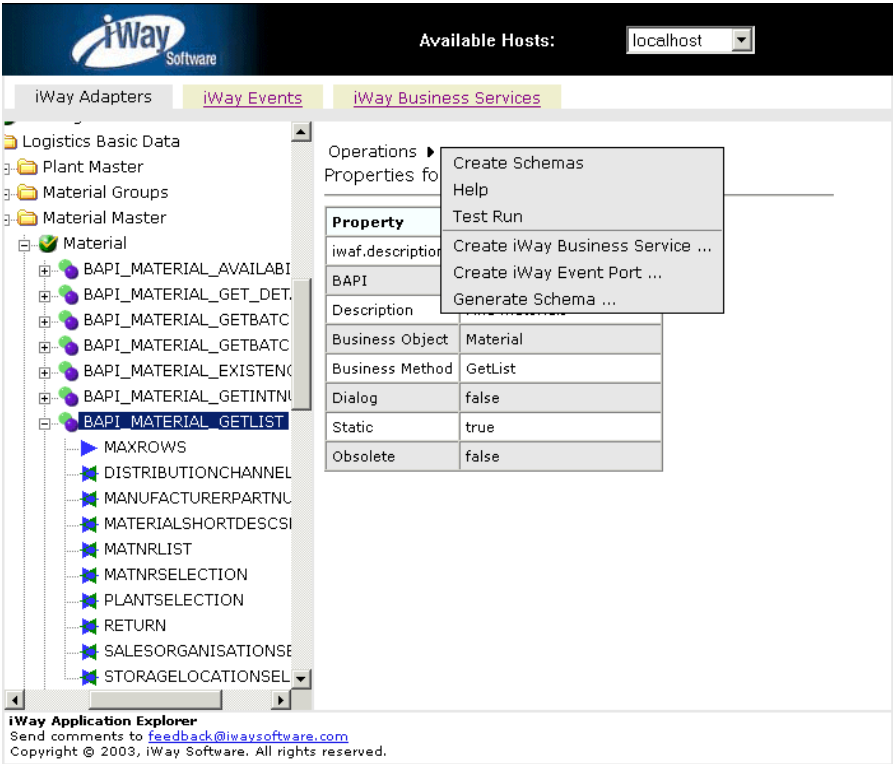
Procedure **How to Create XML Schemas**

For the SAP BAPI method called BAPI_MATERIAL_GETLIST, perform the following steps to create XML request and response schemas:

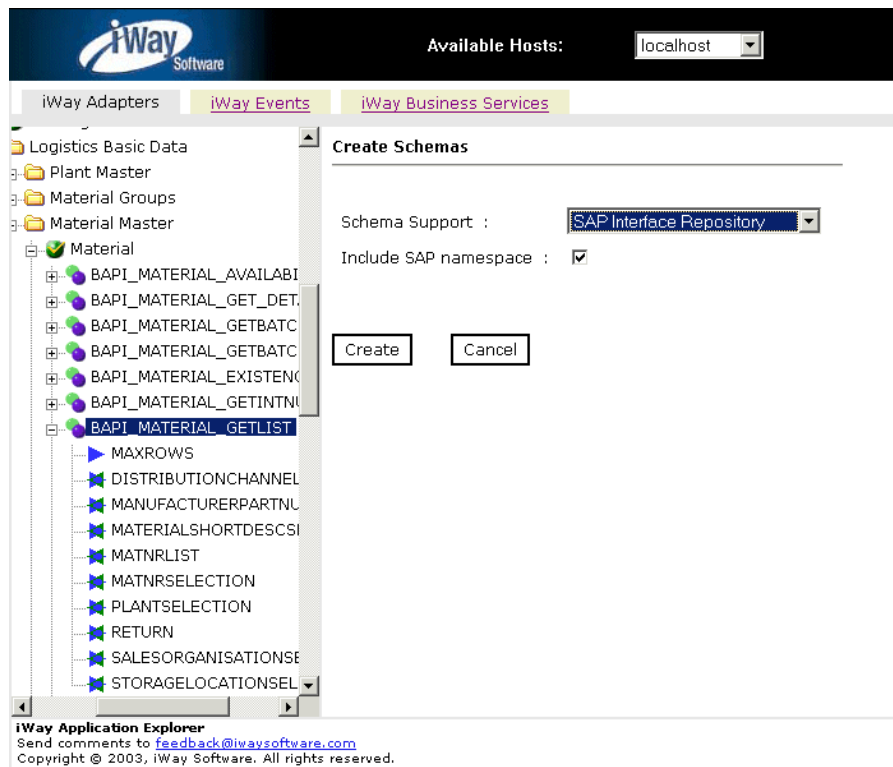
- 1. Select the BAPI_MATERIAL_GETLIST method in the Business Object Repository.



- 2. Click *Operations* in the right pane and select *Create Schemas*.



The Create Schemas window opens in the right pane.



3. From the Schema Support drop-down list select *SAP Interface Repository*.

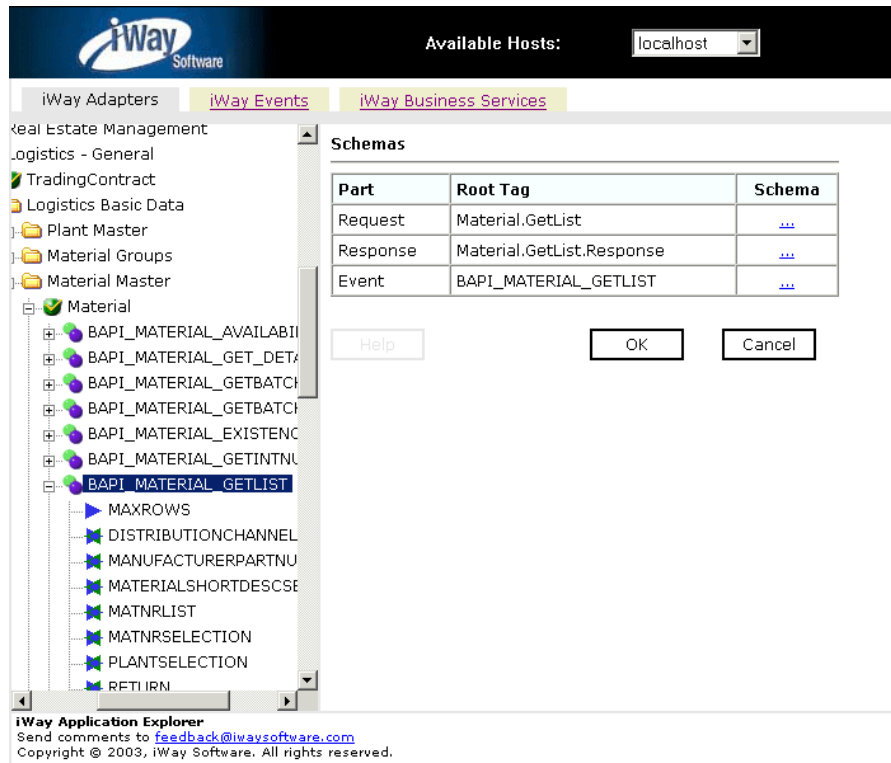
If you are using Microsoft Biztalk Server, you must select *Microsoft Biztalk* from the Schema Support drop-down list.

Note: The *Include SAP namespace* checkbox is selected by default.

4. Click *Create*.

A request, response, and event schema for your business object is created.

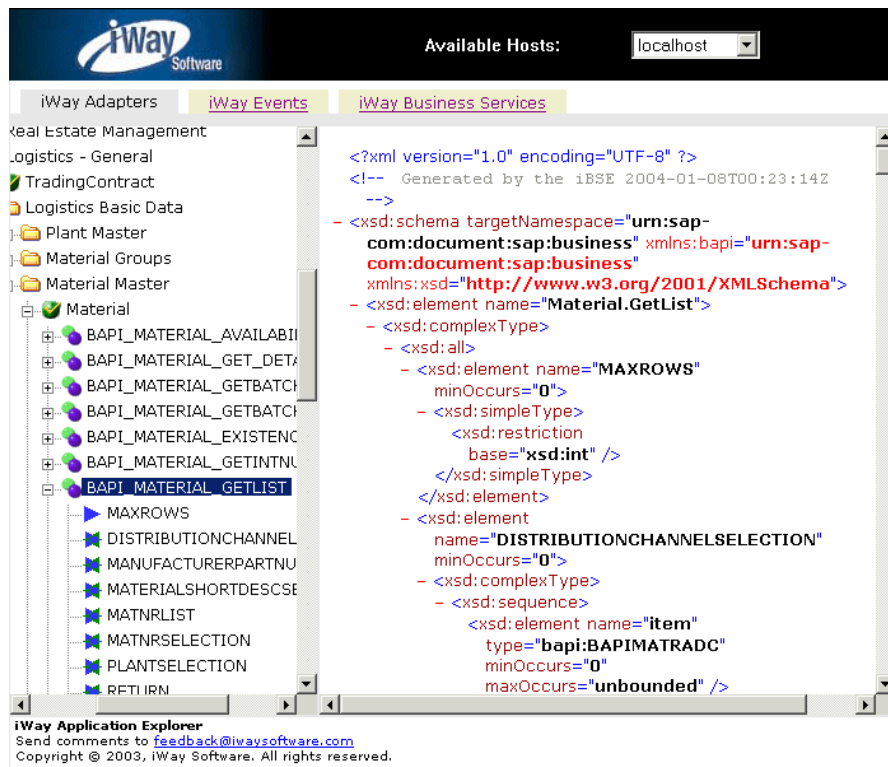
5. To view the schemas, click *Operations* in the right pane and select *Generate Schema*. The Schemas window opens in the right pane.



A table defines the root tag for each schema and provides hyperlinks.

6. Click the hyperlink associated with the type of schema you want to view.

For example, if you click the Request schema, the schema is displayed for you in the right pane.



7. Click the *Back* button on your Web browser to return to the previous window.

Once the schemas are created, you can create iWay Business Services. For more information, see Chapter 4, *Creating and Publishing iWay Business Services*.

You can also create events once the schemas are created. For more information, see Chapter 5, *Using iWay Event Adapters*.

CHAPTER 4

Creating and Publishing iWay Business Services

Topics:

- Understanding iWay Business Services
- Creating iWay Business Services

This section describes how to create and publish an iWay Business Service using the iWay Servlet iWay Application Explorer.

In this section, iAE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the iAE and this chapter as reference for use with other systems, depending on the iWay Adapter you are using.

For more specific information on using the iAE with additional systems, see the iWay Adapter documentation for that particular EIS.

Understanding iWay Business Services

The iWay Servlet iWay Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the iWay 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. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

Once you have browsed the application system business object repository and created an XML schema for the object, you can generate an iWay Business Service for the object you wish to use with your adapter.

Creating iWay Business Services

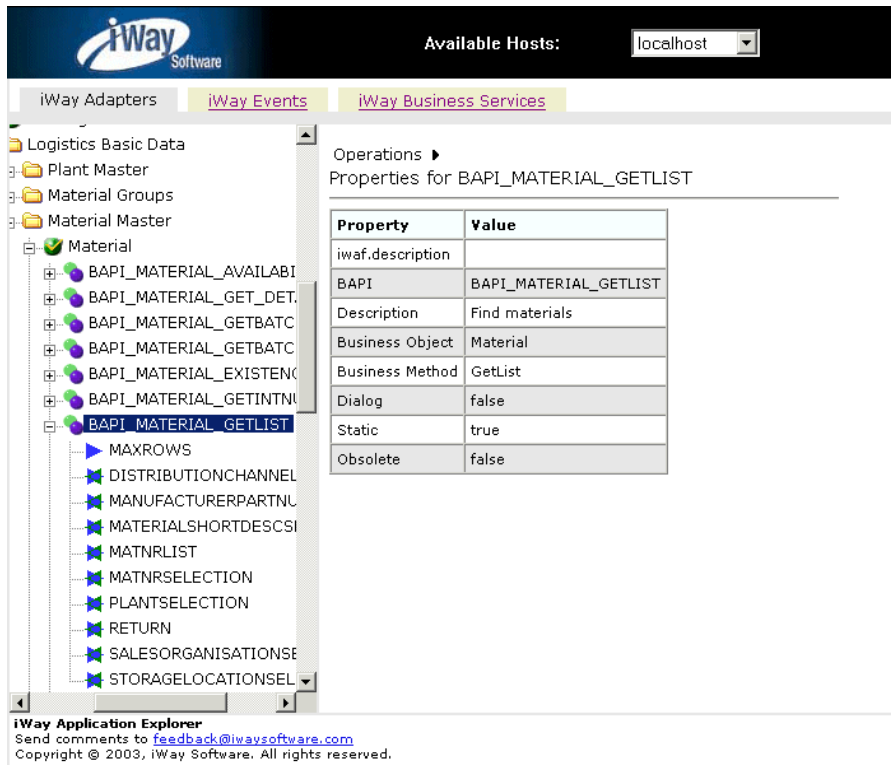
The following section describes how to create iWay Business Services using the iWay Servlet iWay Application Explorer.

Procedure How to Create an iWay Business Service

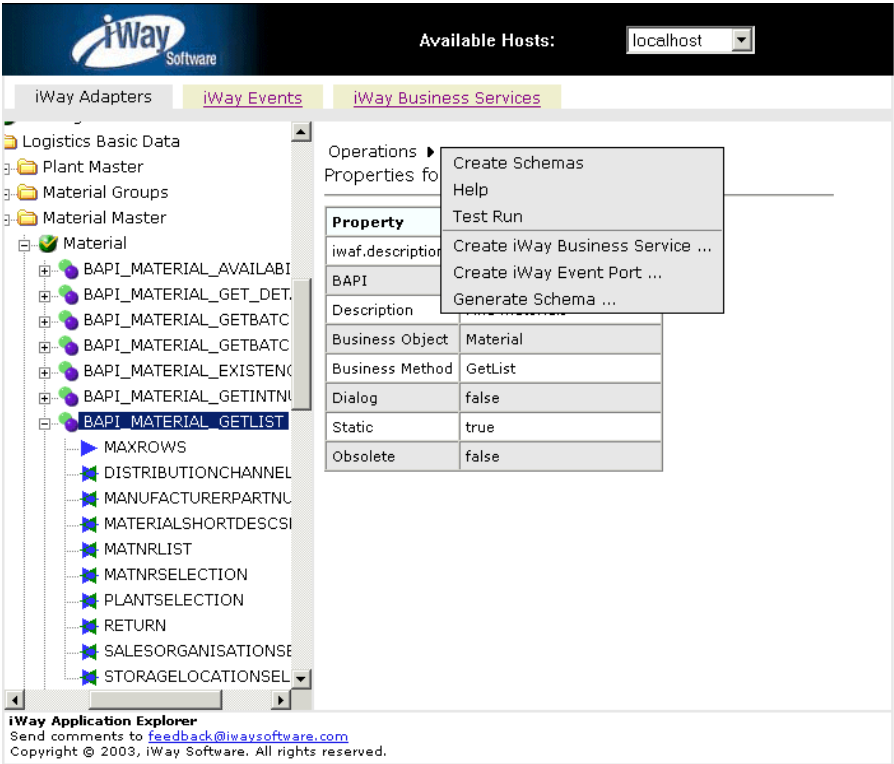
Once you have connected to your application system and created an XML schema for a business object you can create an iWay Business Service for the object.

In the following section, the SAP R/3 BAPI method called BAPI_MATERIAL_GETLIST is used as an example. Perform the following steps to create an iWay Business Service that will return a list of materials.

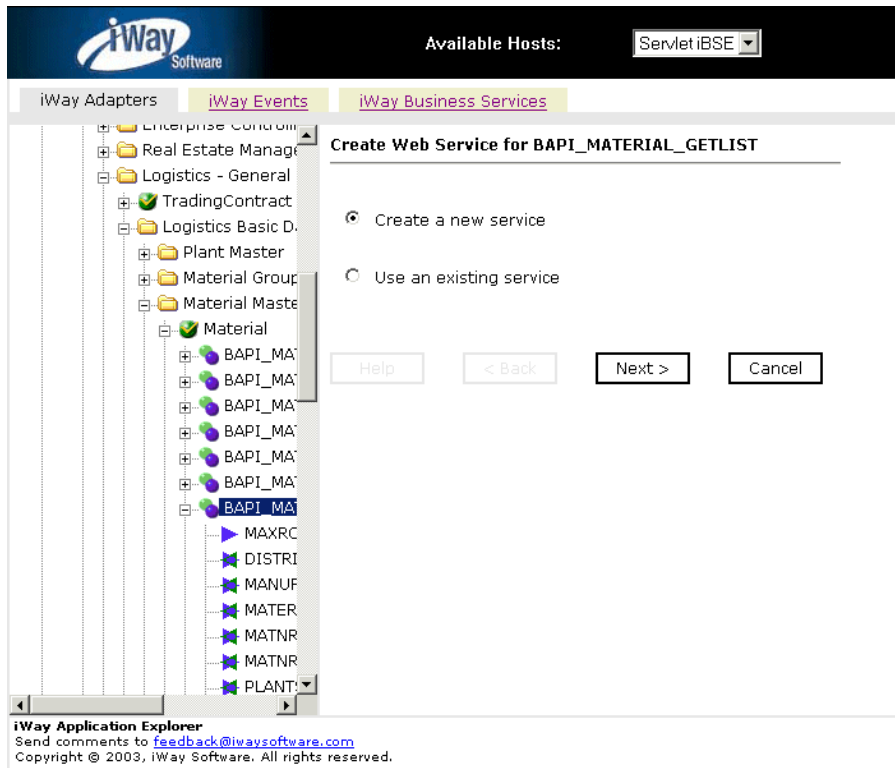
1. Select the BAPI_MATERIAL_GETLIST method in the Business Object Repository.



2. Click *Operations* in the right pane and select *Create iWay Business Service*.

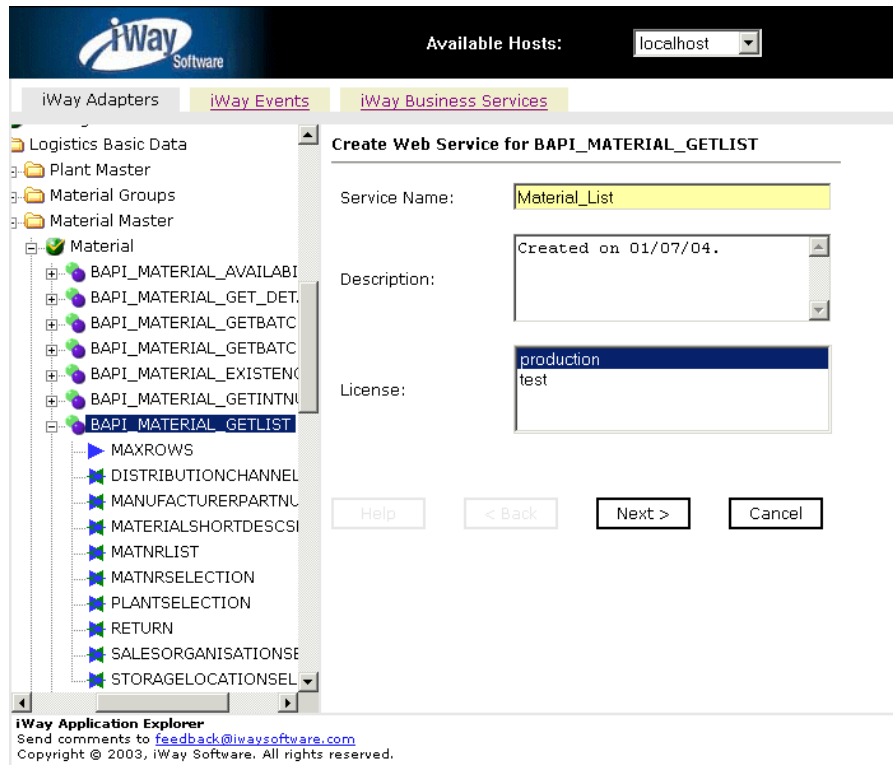


The Create Web Service for BAPI_COMPANYCODE_GETLIST window opens in the right pane.



3. Select *create a new service* and click *Next*.

The following window opens in the right pane.

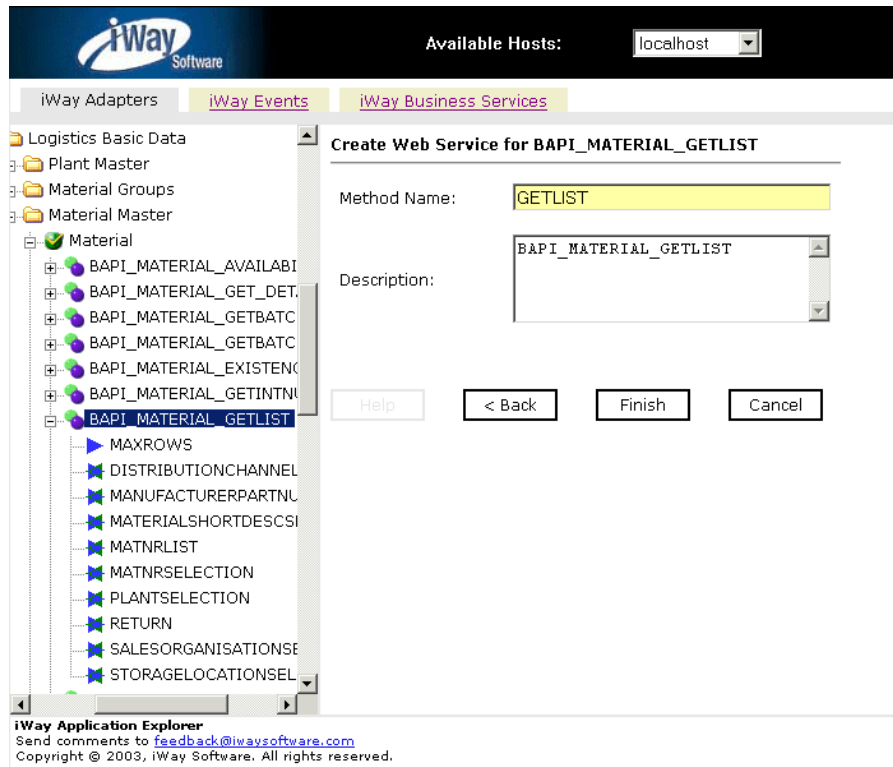


4. Specify the following information that is specific to the iWay Business Service you are defining:

- **Service Name.** Enter a descriptive name for the iWay Business Service.
- **Description.** Enter a brief description for the iWay Business Service.
- **License.** Select the license definition you want to use.

5. Click Next.

The following window opens in the right pane.



6. Specify the following information that is specific to the iWay Business Service you are defining:
 - **Method Name.** Enter a descriptive name for the method.
 - **Description.** Enter a brief description for the method.
7. Click *Finish*.

The iWay Business Services Engine tab opens.

The left pane lists all the available services that have been created. The Material_List service node is expanded for you and the GETLIST method is automatically selected.

The test window for the GETLIST method opens in the right pane.

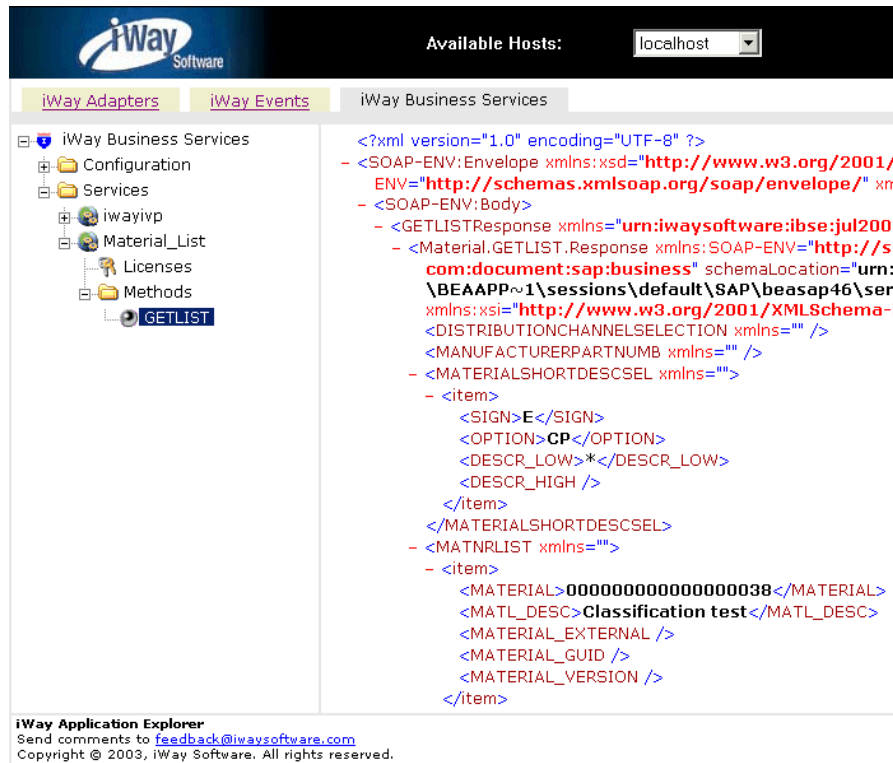


8. Enter a sample XML document that will query the service in the input xml field.

Note: To use the identical sample input XML that is shown in this example, see *Sample iWay Business Service Input XML* on page 4-17.

9. Click *Invoke*.

The result displays in the right pane.



Generating WSDL From a Web Service

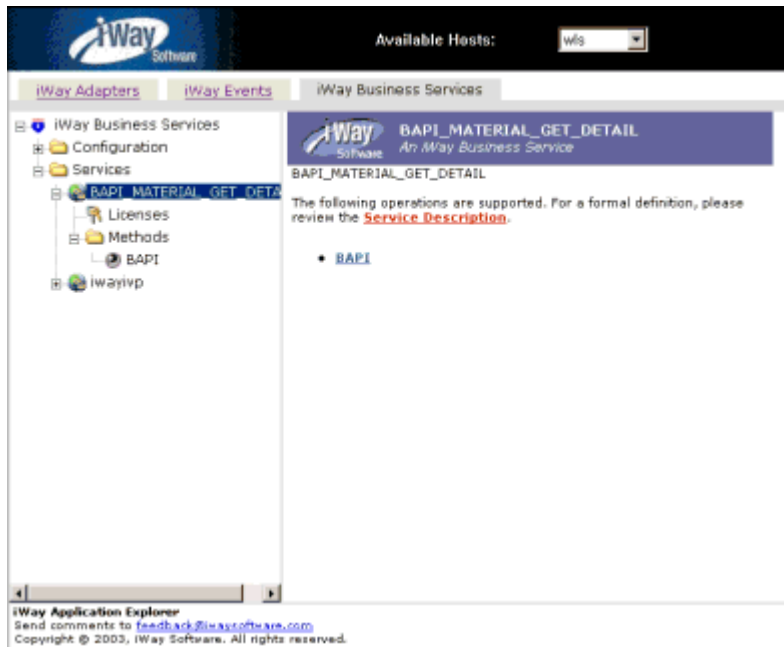
Generating WSDL (Web Services Description Language) from a Web service enables you to make the Web service available to other services within a host server such as the BEA WebLogic Server.

Procedure How to Generate WSDL From a Web Service

To generate WSDL from a Web service:

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

The link for the service appears in the right pane.



4. Right-click the Service Description link and choose *Save Target As* from the pop-up menu.
5. Choose a location for the file and add a .wsdl file extension.
6. Click *Save*.

For example, saving a Web service called BAPI_MATERIAL_GET_DETAIL for a SAP R/3 creates a file named BAPI_MATERIAL_GET_DETAIL.wsdl.

Note: The file extension must be .wsdl.

The following is an example of a WSDL file for a Web service called BAPI_MATERIAL_GET_DETAIL.

```
<definitions xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI:response"
xmlns:tns="urn:schemas-iwaysoftware-com:iwse"
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:BAPI:response"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:BAPI"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"><types><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="ibsinfo"><xs:complexType><xs:sequence><xs:element type="xs:string"
name="service"/><xs:element type="xs:string" name="method"/><xs:element
type="xs:string" name="license"/><xs:element type="xs:string"
minOccurs="0" name="disposition"/><xs:element type="xs:string"
minOccurs="0" name="Username"/><xs:element type="xs:string" minOccurs="0"
name="Password"/><xs:element type="xs:string" minOccurs="0"
name="language"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="adapterexception"><xs:complexType><xs:sequence><xs:element
type="xs:string"
name="error"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI"
targetNamespace="urn:iwaysoftware:ibse:jul2003:BAPI"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:BAPI"
elementFormDefault="qualified"><xs:element
name="BAPI"><xs:complexType><xs:sequence><xs:element
name="BAPI_MATERIAL_GET_DETAIL"><xs:complexType><xs:all><xs:element
minOccurs="1" name="MATERIAL"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="18"/></xs:restriction></xs:simpleType>
</xs:element><xs:element minOccurs="0"
name="PLANT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="4"/></xs:restriction></xs:simpleType>
</xs:element><xs:element minOccurs="0"
name="VALUATIONAREA"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="4"/></xs:restriction></xs:simpleType>
</xs:element><xs:element minOccurs="0"
name="VALUATIONTYPE"><xs:simpleType><xs:restriction
```

```

base="xs:string"><xs:maxLength
value="10"/></xs:restriction></xs:simpleType></xs:element></xs:all></xs:c
omplexType></xs:element></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI:response"
targetNamespace="urn:iwaysoftware:ibse:jul2003:BAPI:response"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:BAPI:response"
elementFormDefault="qualified"><xs:element
name="BAPIResponse"><xs:complexType><xs:sequence><xs:element
name="BAPI_MATERIAL_GET_DETAIL.Response"><xs:complexType><xs:all><xs:elem
ent type="rfc:BAPIMATDOC" minOccurs="0"
name="MATERIALPLANTDATA"/><xs:element type="rfc:BAPIMATDOBEW"
minOccurs="0" name="MATERIALVALUATIONDATA"/><xs:element
type="rfc:BAPIMATDOA" minOccurs="0"
name="MATERIAL_GENERAL_DATA"/><xs:element type="rfc:BAPIRETURN"
minOccurs="0" name="RETURN"/></xs:all></xs:complexType></xs:element>
</xs:sequence><xs:attribute type="xs:string" use="required"
name="cid"/></xs:complexType>
</xs:element><xs:complexType
name="BAPIMATDOC"><xs:sequence><xs:element
name="PUR_GROUP"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="ISSUE_UNIT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType></xs:element></xs:sequence>
</xs:complexType><xs:complexType
name="BAPIMATDOBEW"><xs:sequence><xs:element
name="PRICE_CTRL"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="1"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="MOVING_PR"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="23"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="STD_PRICE"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="23"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="PRICE_UNIT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="5"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="CURRENCY"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="5"/></xs:restriction></xs:simpleType>

```

```

        </xs:element><xs:element
name="CURRENCY_ISO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType></xs:element></xs:sequence>
</xs:complexType><xs:complexType
name="BAPIMATDOA"><xs:sequence><xs:element
name="MATL_DESC"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="40"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="OLD_MAT_NO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="18"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="MATL_TYPE"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="4"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="IND_SECTOR"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="1"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="DIVISION"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="MATL_GROUP"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="9"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="PROD_HIER"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="18"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="BASIC_MATL"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="14"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="STD_DESCR"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="18"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="LAB_DESIGN"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
</xs:element><xs:element
name="PROD_MEMO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength

```

```
value="18"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="PAGEFORMAT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="4"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="CONTAINER"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="STOR_CONDS"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="TEMP_CONDS"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="BASE_UOM"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="EAN_UPC"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="18"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="EAN_CAT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="SIZE_DIM"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="32"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="GROSS_WT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="NET_WEIGHT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="UNIT_OF_WT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="VOLUME"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
```

```

value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="VOLUMEUNIT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="LENGTH"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="WIDTH"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="HEIGHT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="13"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="UNIT_DIM"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="MANU_MAT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="40"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="MFR_NO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="10"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="BASE_UOM_ISO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="UNIT_OF_WT_ISO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="VOLUMEUNIT_ISO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="UNIT_DIM_ISO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="3"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="CREATED_ON"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength

```

```

value="8"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="CREATED_BY"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="12"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="LAST_CHNGE"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="8"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="CHANGED_BY"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="12"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="MATL_CAT"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="2"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="EMPTIESBOM"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="1"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="BASIC_MATL_NEW"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="48"/></xs:restriction></xs:simpleType></xs:element></xs:sequence>
    </xs:complexType><xs:complexType
name="BAPIRETURN"><xs:sequence><xs:element
name="TYPE"><xs:simpleType><xs:restriction base="xs:string"><xs:maxLength
value="1"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="CODE"><xs:simpleType><xs:restriction base="xs:string"><xs:maxLength
value="5"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="MESSAGE"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="220"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="LOG_NO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="20"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="LOG_MSG_NO"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="6"/></xs:restriction></xs:simpleType>
    </xs:element><xs:element
name="MESSAGE_V1"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="50"/></xs:restriction></xs:simpleType>

```

```

        </xs:element><xs:element
name="MESSAGE_V2"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="50"/></xs:restriction></xs:simpleType>
        </xs:element><xs:element
name="MESSAGE_V3"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="50"/></xs:restriction></xs:simpleType>
        </xs:element><xs:element
name="MESSAGE_V4"><xs:simpleType><xs:restriction
base="xs:string"><xs:maxLength
value="50"/></xs:restriction></xs:simpleType></xs:element></xs:sequence><
/xs:complexType></xs:schema>
    </types><message name="BAPIIn"><part element="m1:BAPI"
name="parameters"/>
    </message><message name="BAPIOut"><part element="m1:BAPIResponse"
name="parameters"/>
    </message><message name="BAPI_MATERIAL_GET_DETAILHeader"><part
element="tns:ibsinfo" name="header"/>
    </message><message name="AdapterException"><part
element="tns:adapterexception" name="fault"/>
    </message><portType name="BAPI_MATERIAL_GET_DETAILSoap"><operation
name="BAPI"><documentation/><input message="tns:BAPIIn"/><output
message="tns:BAPIOut"/><fault message="tns:AdapterException"
name="AdapterExceptionFault"/></operation>
    </portType><binding type="tns:BAPI_MATERIAL_GET_DETAILSoap"
name="BAPI_MATERIAL_GET_DETAILSoap"><soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/><operation
name="BAPI"><soap:operation style="document"
soapAction="BAPI_MATERIAL_GET_DETAIL.BAPIRequest@test@"/><input><soap:bo
dy use="literal"/><soap:header part="header"
message="tns:BAPI_MATERIAL_GET_DETAILHeader" use="literal"/>
    </input><output><soap:body use="literal"/>
    </output><fault name="AdapterExceptionFault"><soap:fault
use="literal" name="AdapterExceptionFault"/></fault></operation>
    </binding><service
name="BAPI_MATERIAL_GET_DETAIL"><documentation>BAPI_MATERIAL_GET_DETAIL</
documentation><port binding="tns:BAPI_MATERIAL_GET_DETAILSoap"
name="BAPI_MATERIAL_GET_DETAILSoap1"><soap:address
location="http://GERBER-2K.ibi.com:7001/ibse/IBSEServlet/XDSOAPRouter"/><
/port></service></definitions>

```

For more information on using WSDL in the BEA WebLogic Workshop, including an example, see Appendix A, *Using the WebLogic Workshop to Access Web Services*.

Reference **Sample iWay Business Service Input XML**

The following input XML retrieves a list of materials using the SAP R/3 BAPI_MATERIAL_GETLIST method.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!-- Sample XML file generated by XMLSPY v5 rel. 3 U
(http://www.xmlspy.com)
-->
- <Material.GETLIST xmlns="urn:sap-com:document:sap:business"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:sap-com:document:sap:business
C:\PROGRA~1\BEASYS~1\BEAAPP~1\sessions\default\SAP\beasap46\service_BAPI_
MATERIAL_GETLIST.xsd">
  <MAXROWS>1000</MAXROWS>
- <DISTRIBUTIONCHANNELSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <DISTR_CHAN_LOW />
  <DISTR_CHAN_HIGH />
</item>
</DISTRIBUTIONCHANNELSELECTION>
- <MANUFACTURERPARTNUMB>
- <item>
  <MANU_MAT />
  <MFR_NO />
</item>
</MANUFACTURERPARTNUMB>
- <MATERIALSHORTDESCSEL>
- <item>
  <SIGN />
  <OPTION />
  <DESCR_LOW />
  <DESCR_HIGH />
</item>
</MATERIALSHORTDESCSEL>
- <MATNRLIST>
- <item>
  <MATERIAL />
  <MATL_DESC />
  <MATERIAL_EXTERNAL />
  <MATERIAL_GUID />
  <MATERIAL_VERSION />
</item>
</MATNRLIST>
- <MATNRSELECTION>
- <item>
  <SIGN>E</SIGN>
  <OPTION>BT</OPTION>
  <MATNR_LOW>1000</MATNR_LOW>
  <MATNR_HIGH>1010</MATNR_HIGH>
</item>
```



```

    </MATNRSELECTION>
- <PLANTSELECTION>
- <item>
    <SIGN />
    <OPTION />
    <PLANT_LOW />
    <PLANT_HIGH />
  </item>
</PLANTSELECTION>
- <RETURN>
- <item>
    <TYPE />
    <ID />
    <NUMBER />
    <MESSAGE />
    <LOG_NO />
    <LOG_MSG_NO />
    <MESSAGE_V1 />
    <MESSAGE_V2 />
    <MESSAGE_V3 />
    <MESSAGE_V4 />
    <PARAMETER />
    <ROW>0</ROW>
    <FIELD />
    <SYSTEM />
  </item>
</RETURN>
- <SALESORGANISATIONSELECTION>
- <item>
    <SIGN />
    <OPTION />
    <SALESORG_LOW />
    <SALESORG_HIGH />
  </item>
</SALESORGANISATIONSELECTION>
- <STORAGELOCATIONSELECT>
- <item>
    <SIGN />
    <OPTION />
    <STLOC_LOW />
    <STLOC_HIGH />
  </item>
</STORAGELOCATIONSELECT>
</Material.GETLIST>

```

CHAPTER 5

Using iWay Event Adapters

Topics:

- Understanding iWay Event Functionality
- Creating Event Ports
- Creating Channels

This section describes how to use the iWay Servlet iWay Application Explorer (iAE) to connect to an Enterprise Information System (EIS) and generate iWay Events.

The iAE supports access to several Enterprise Information Systems. Even though the underlying technology used to access them varies significantly, the iAE user interface varies only slightly to accommodate EIS differences. In this section, iAE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the iAE and this chapter as reference for use with other systems, depending on the iWay Adapter you are using.

For more specific information on using the iAE with additional systems, see the iWay Adapter documentation for that particular EIS.

Understanding iWay Event Functionality

Events are generated as a result of activity on an application system. You can use events to trigger an action in your application. For example, SAP R/3 may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event.

Once you create a connection to your application system, you can add events using the iWay Servlet iWay Application Explorer. To create an iWay Event, you must:

1. Create a **Port**.

A port is used to associate a particular business object exposed by the iWay Adapter with a particular disposition. The port disposition defines the endpoint of the event consumption and is an attribute of the channel provided by the container. For more information, see *Creating Event Ports* on page 5-2.

2. Create a **Channel**.

A channel represents configured connections to particular instances of backend systems. A channel is used to bind one or more event ports to a particular listener managed by the iWay Adapter. For more information, see *How to Create a Channel* on page 5-12.

Creating Event Ports

The following section describes how to create an arbitrary or specific event port using the iWay Servlet iWay Application Explorer. In addition, information on how to modify and delete each type of event port is provided.

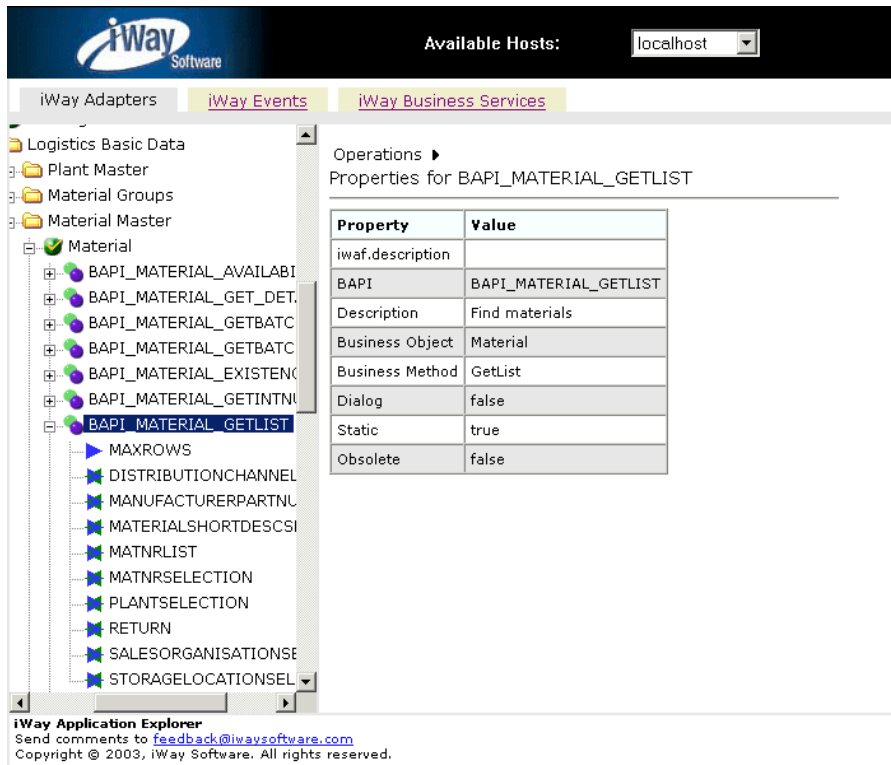
Procedure How to Create an Event Port

Perform the following steps to create a specific event port using the iWay Servlet iWay Application Explorer.

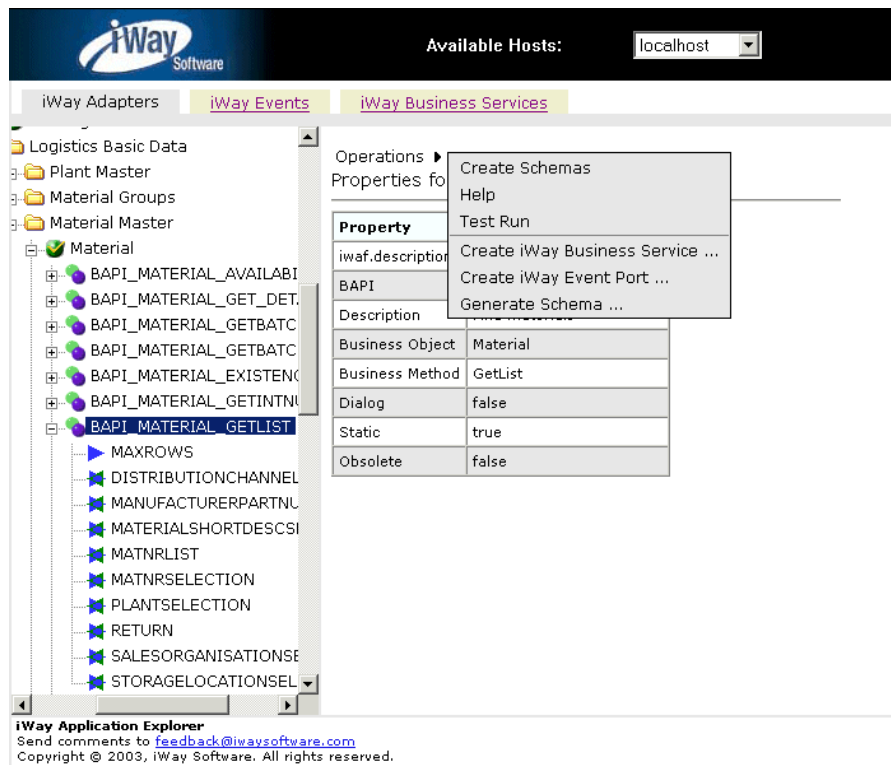
1. Click the *iWay Adapters* tab.

The iWay Adapters window opens.

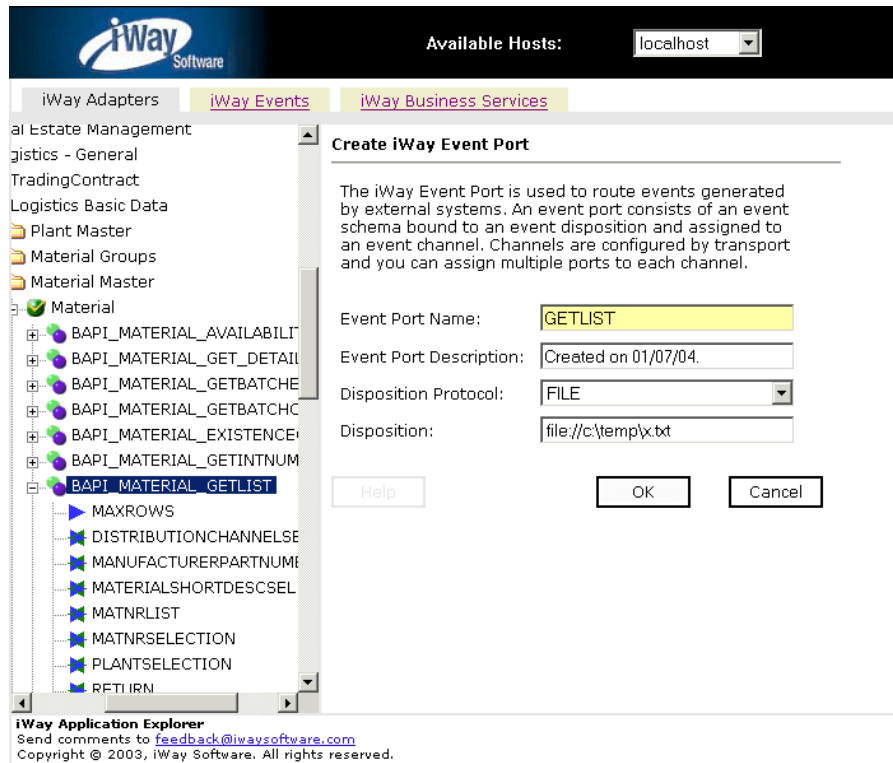
2. Select the BAPI_MATERIAL_GETLIST method in the Business Object Repository.



3. Click *Operations* in the right pane and select *Create iWay Event Port*.



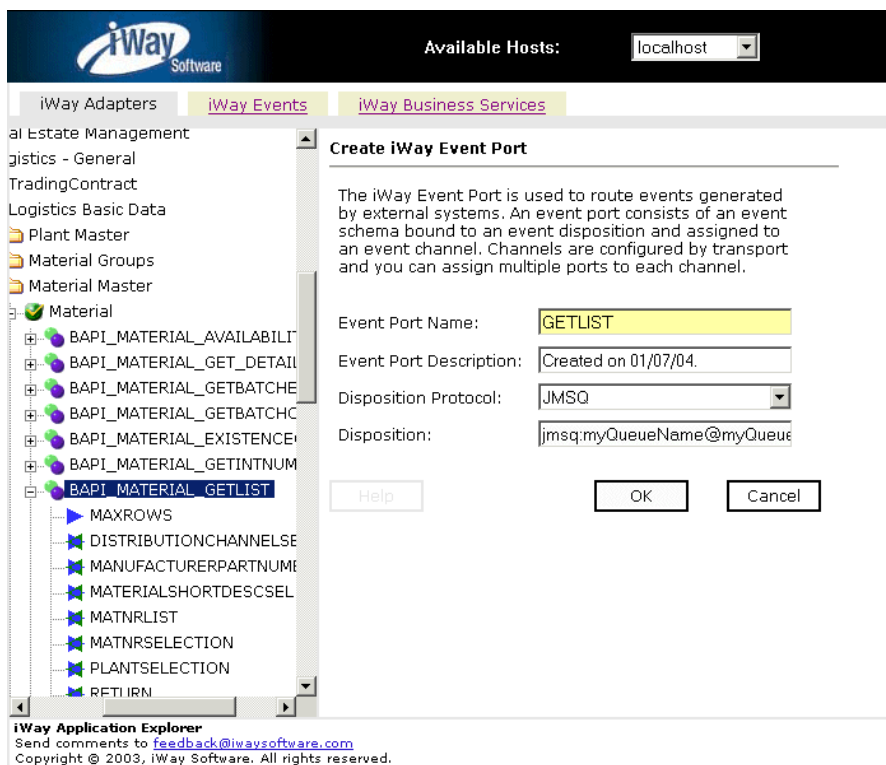
The Create iWay Event Port window opens in the right pane.



4. Enter an event port name and a brief description.
5. Select a disposition protocol (for example, File) in the drop-down list.
6. In the Disposition field, enter a destination where the event data will be written. For example,

```
file:///c:/temp/x.txt
```

To route the output of an SAP R/3 business function to a JMS Queue hosted by BEA WebLogic Server, use the JMS protocol.



For example, select *JMSQ* in the Disposition Protocol drop-down list. In the Disposition field, enter a JMS destination in the form of:

```
jmsq:myQueueName@javax.jms.QueueConnectionFactory;jndiurl=t3://localhost:7001;jndifactory=weblogic.jndi.WLInitialContextFactory
```

where:

Queue

Is the JNDI name of a queue to which events are emitted.

ConnectionFactory

Is a resource which contains information about the JMS Server. The WebLogic connection factory is: `javax.jms.QueueConnectionFactory`

jndiurl

Is the URL of the Weblogic Server: `t3://host:port`

Host is the machine name where the Weblogic Server is installed.

Port is the port on which Weblogic server is listening. The default port if not changed at installation is 7001.

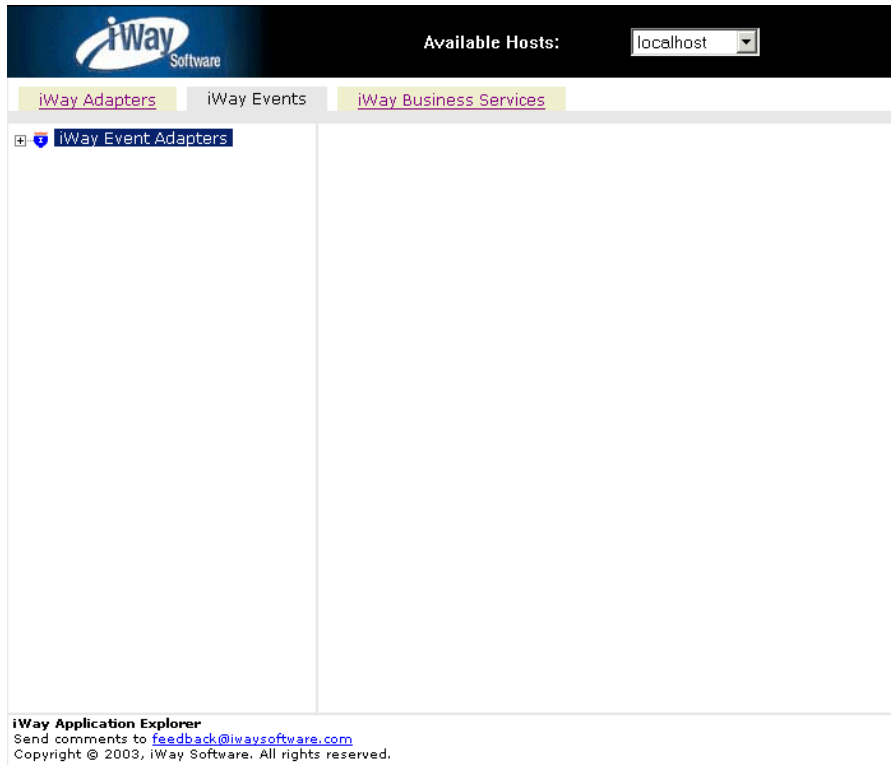
`jndifactory`

Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For WebLogic Server, the WebLogic factory is:

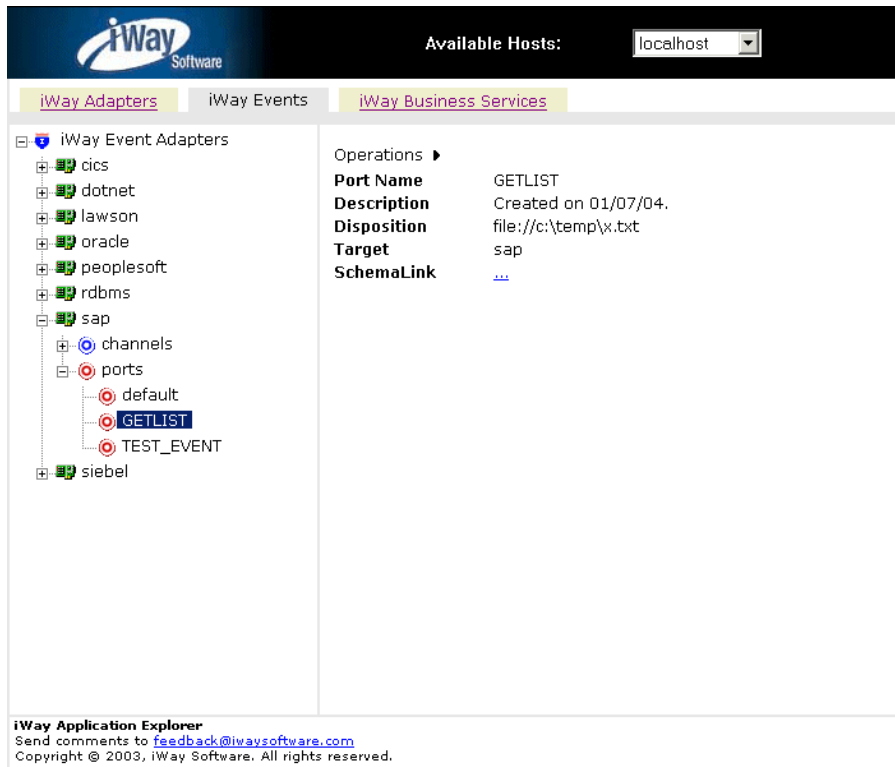
`weblogic.jndi.WLInitialContextFactory`

7. Click OK.

The iWay Events tab opens in the right pane.



8. Expand the iWay Event Adapters node.



A table summarizes all the information associated with the event port you created in the right pane. The event port is also listed under the ports node in the left pane.

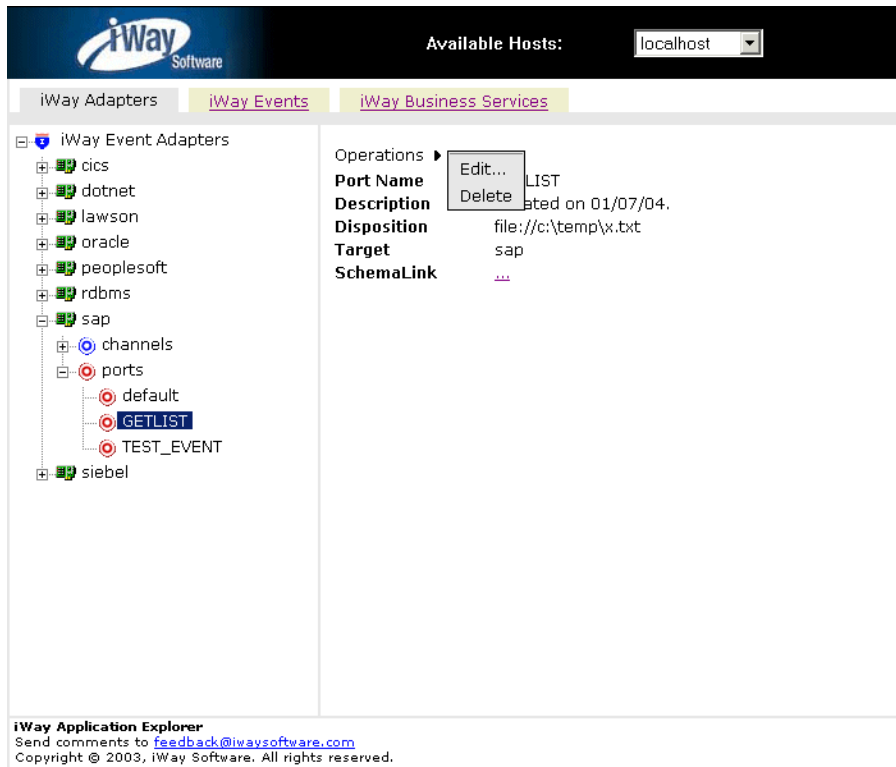
9. Click *Schema* in the right pane to view the event schema that has been created for the event port.

You are now ready to associate the event port with a channel.

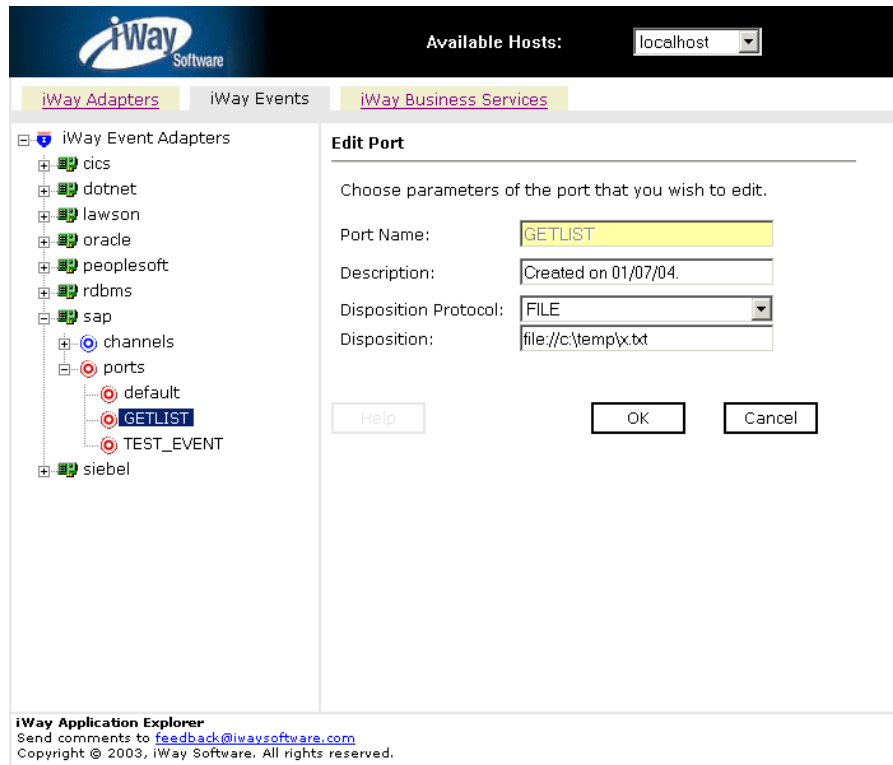
Procedure How to Edit an Event Port

Perform the following steps to edit an existing event port.

1. Select the event port you want to edit in the left pane or in the right pane.



2. Click *Operations* in the right pane and select *Edit*.
The Edit Port window opens in the right pane.



3. Make any necessary changes to the event port configuration windows and click **OK** once you are complete.

Procedure How to Delete an Event Port

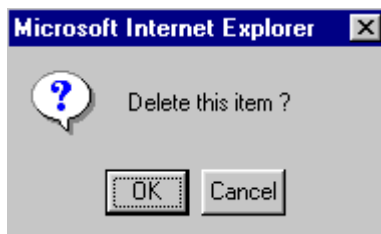
Perform the following steps to delete an existing event port.

1. Select the event port you want to delete in the left pane or in the right pane.



2. Click *Operations* in the right pane and select *Delete*.

The following confirmation dialog box opens.



3. Click *OK* to delete the event port you selected.

The event port is removed from the list in the left pane.

Creating Channels

The following section describes how to create a channel for your iWay Event. All defined event ports must be associated with a channel.

Procedure How to Create a Channel

Perform the following steps to create a channel using the iWay Servlet iWay Application Explorer.

1. Click the *iWay Events* tab.

This iWay Event Adapters window opens:



The list of iWay Adapters in the left pane support events.

2. Expand the iWay Adapter node (for example, SAP).

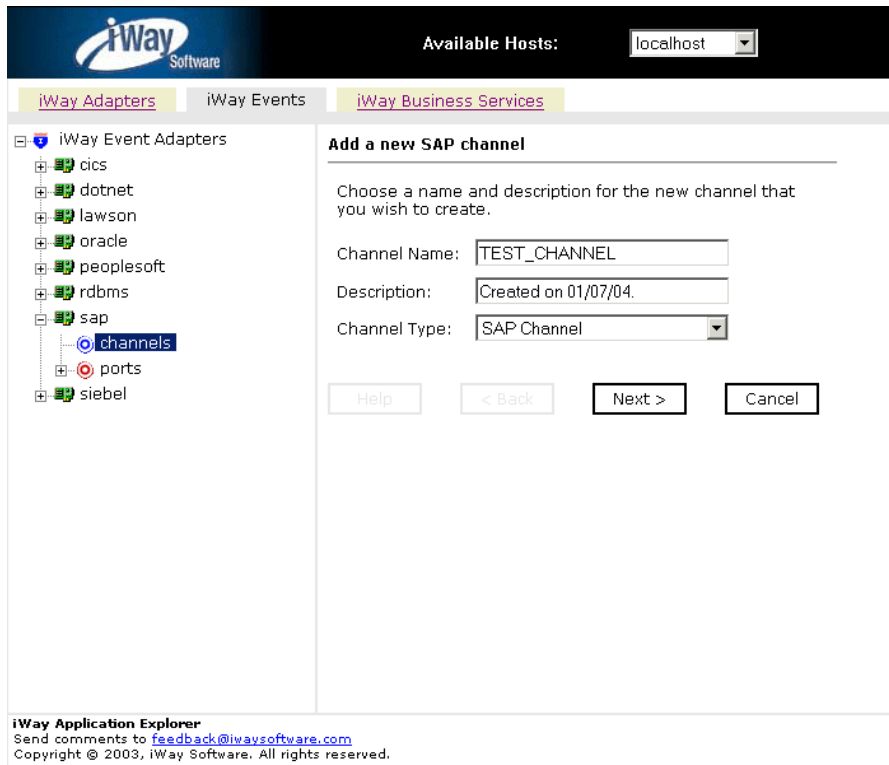
The ports and channels nodes are listed in the left pane.

3. Click the *channels* node.



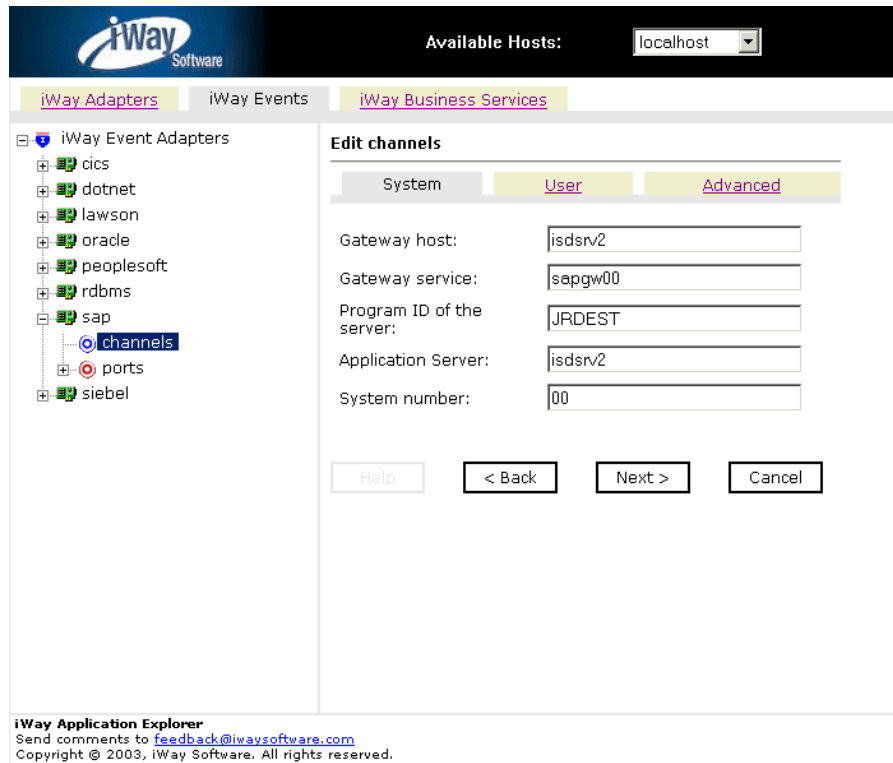
4. Click *Operations* in the right pane and select *Add a new channel*.

The Add a new channel window opens in the right pane.



5. Enter a name for the event port (for example, TEST_CHANNEL) and a brief description.
6. Select a channel type from the drop-down list.
7. Click *Next*.

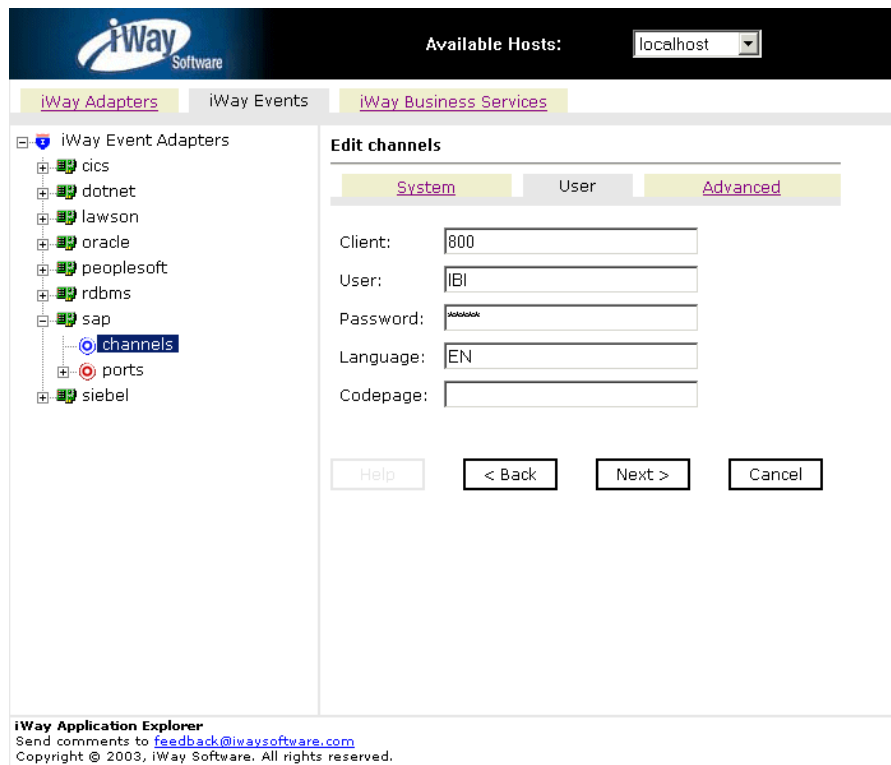
The Edit channels window opens in the right pane.



8. Enter the system information that is specific to your EIS.

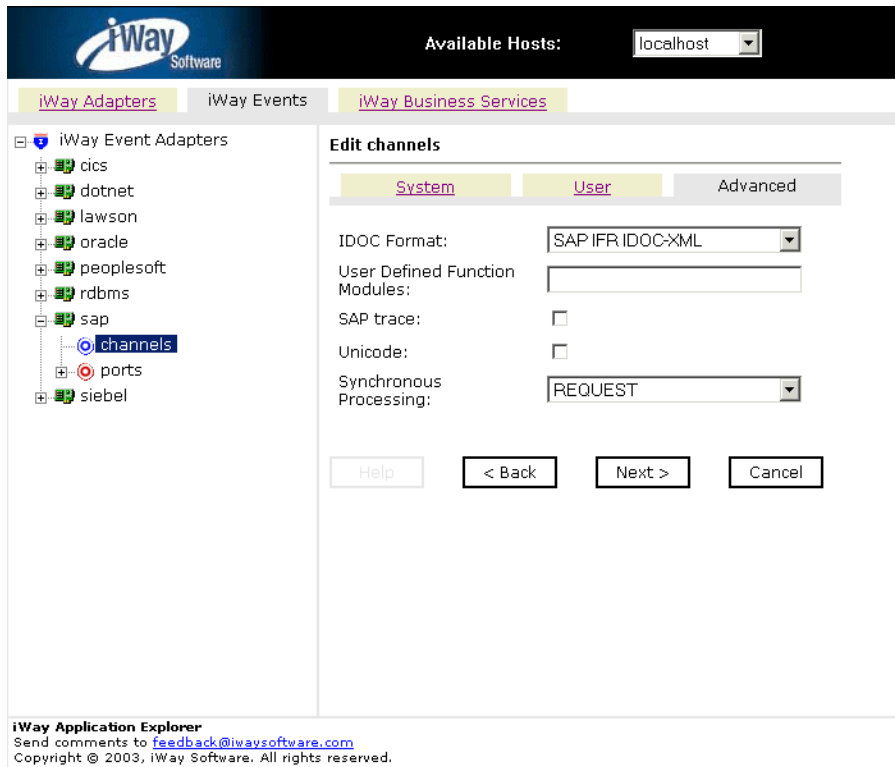
Note: For more specific information on system, user, and advanced values that apply to your EIS, see the iWay Adapter documentation for that particular EIS.

9. Click the *User* tab.



10. Enter the user information that is specific to your EIS.

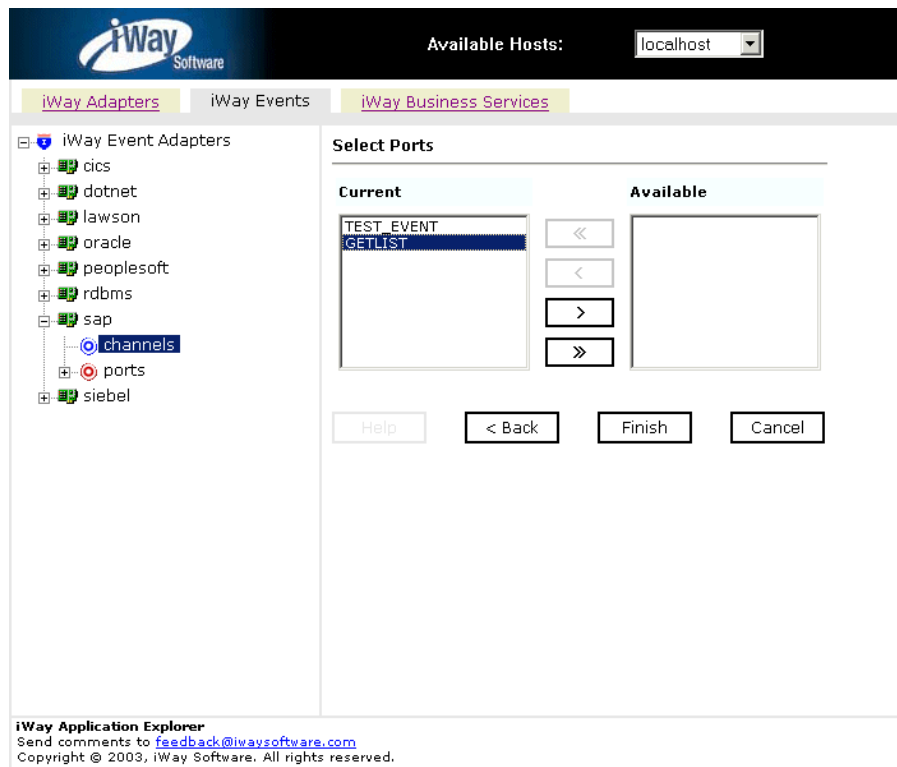
11. Click the *Advanced* tab.



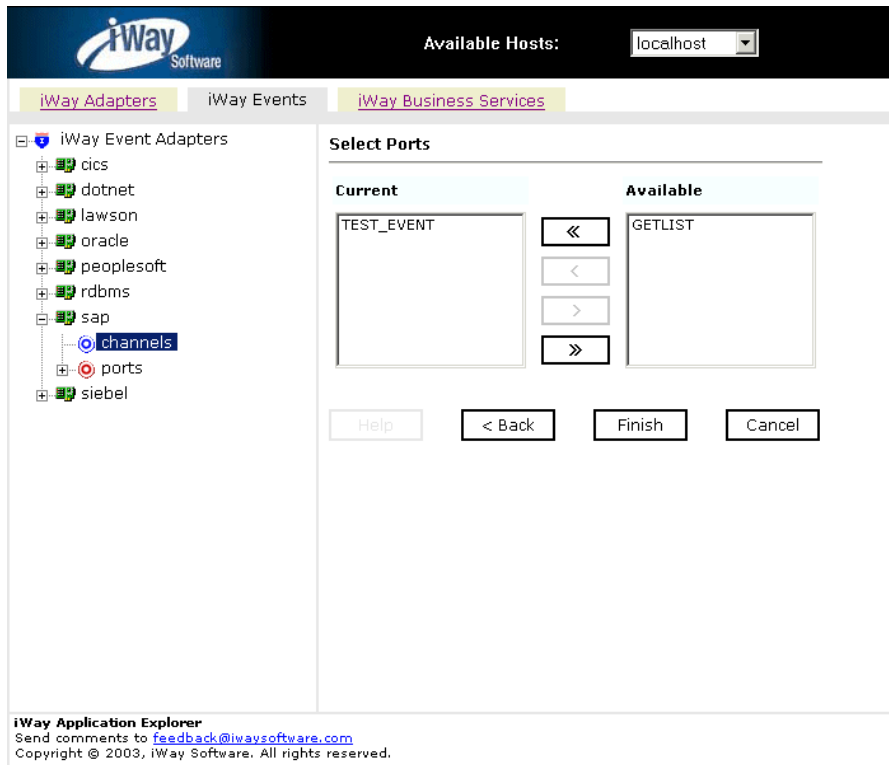
12. Specify any additional information or criteria for the channel you are creating.

13. Click *Next*.

The Select Ports window opens in the right pane.



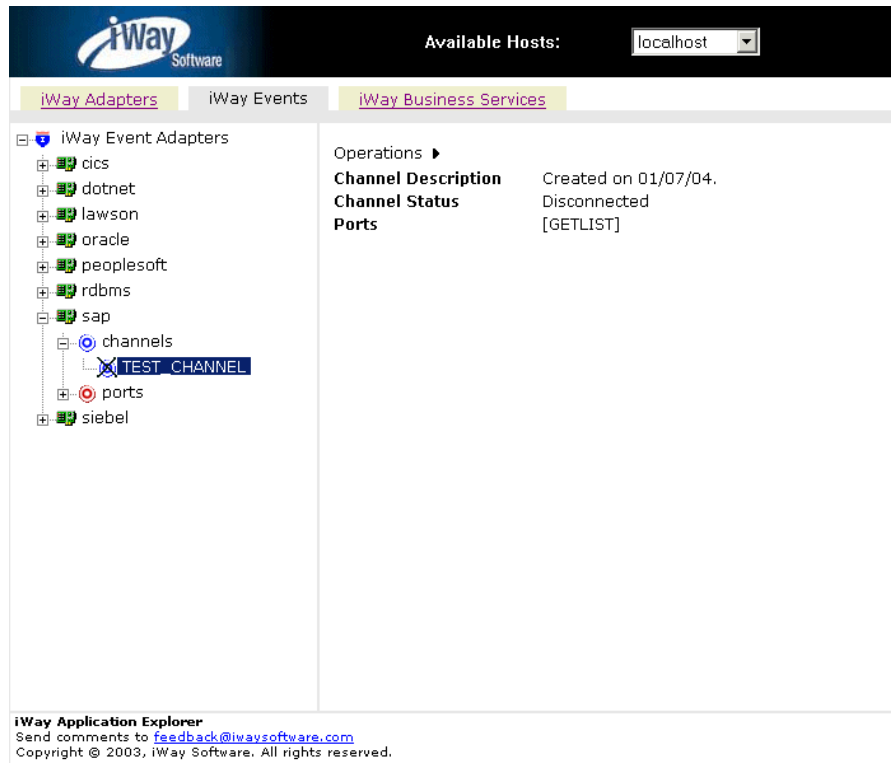
14. Select an event port from the list of current ports and click the single right > arrow button to transfer it to the available list.



To associate all the event ports click the double right >> arrow button.

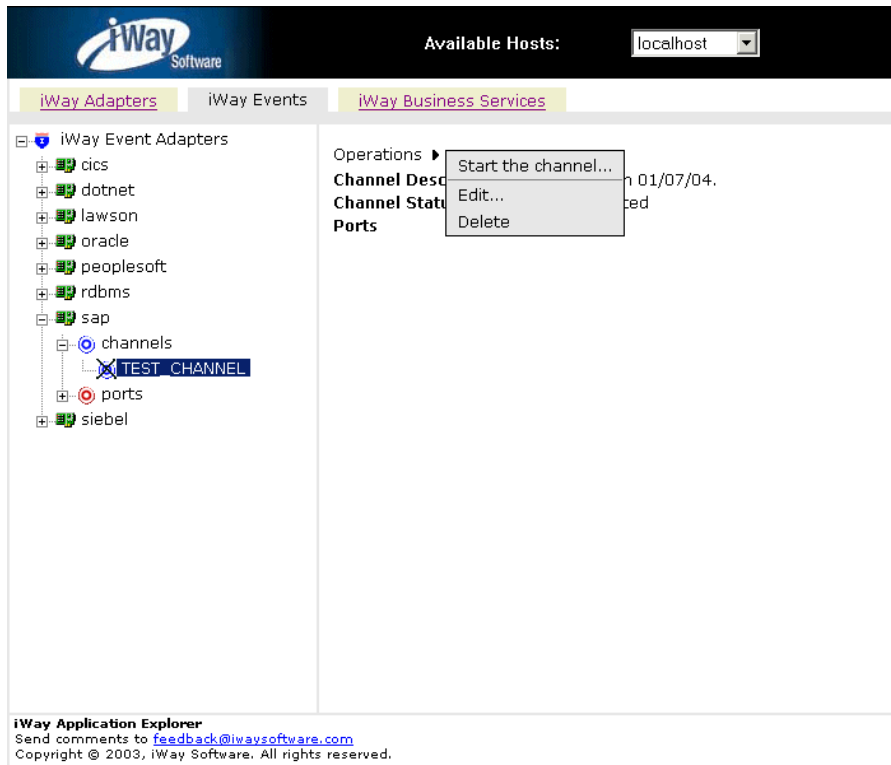
15. Click *Finish*.

The summary window opens in the right pane.



A summary provides the channel description, channel status, and available ports. All the information is associated with the channel you created. The channel is also listed under the channels node in the left pane. An X through the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

16. Click *Operations* in the right pane and select *Start the channel*.



The channel you created is now active and the X through the icon in the left pane is removed.

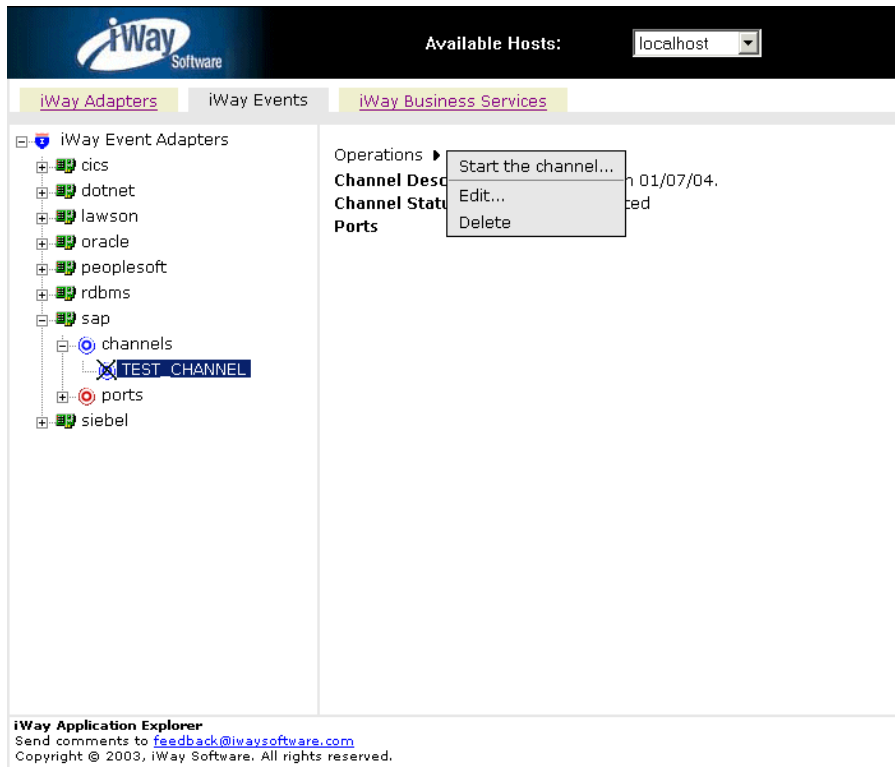


If you want to stop the channel at any time, click *Operations* in the right pane and select *Stop the channel*.

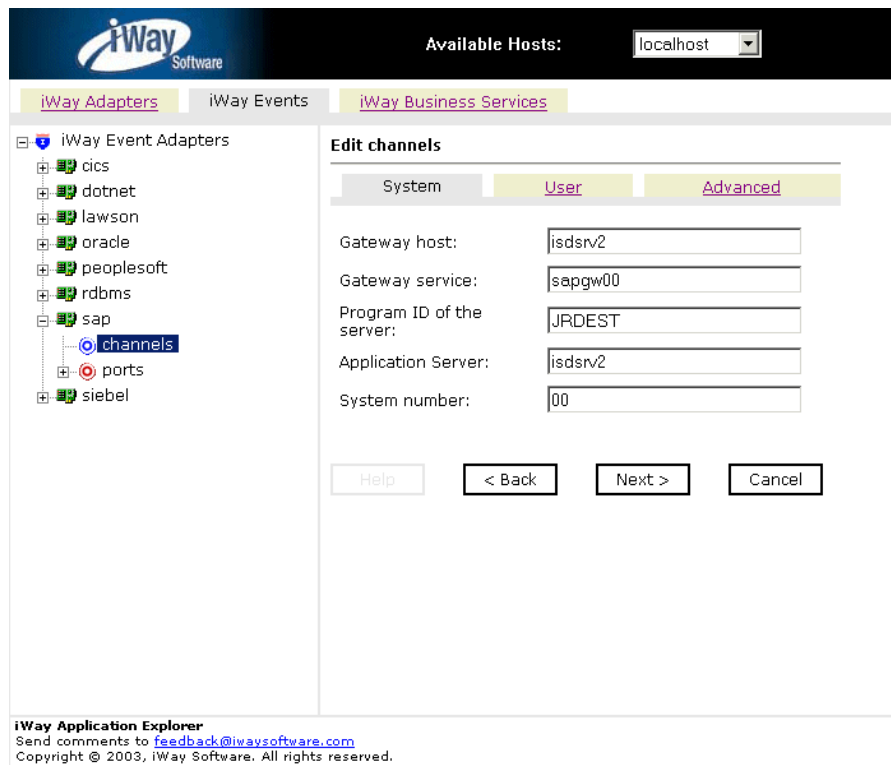
Procedure How to Edit a Channel

Perform the following steps to edit an existing channel.

1. Select the channel you want to edit in the left pane.



2. Click *Operations* in the right pane and select *Edit*.
The Edit channels window opens in the right pane.

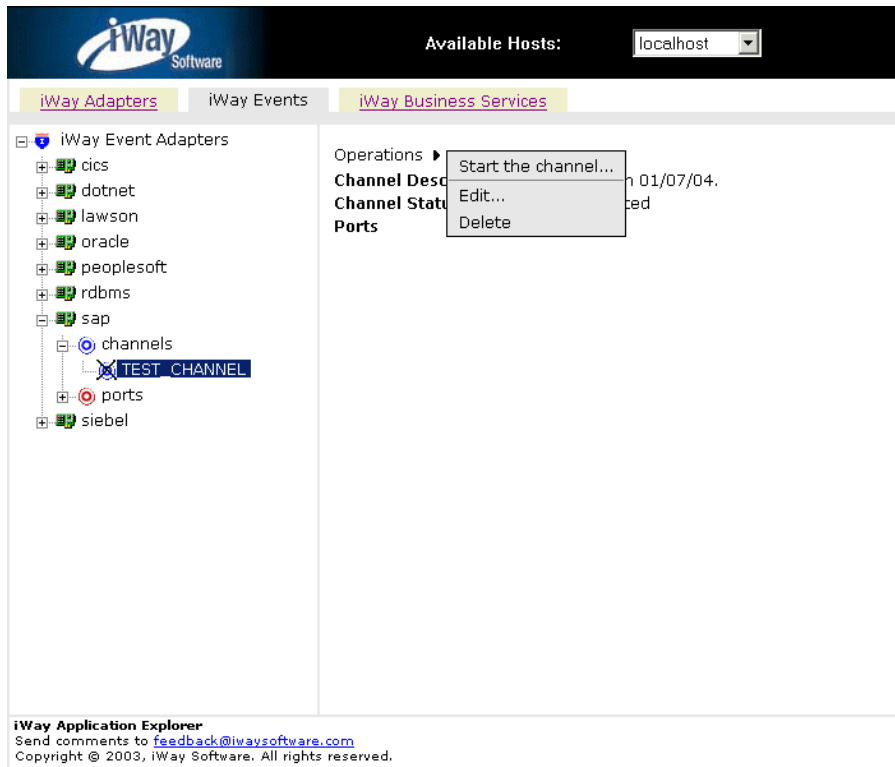


3. Make any necessary changes to the channel configuration windows and click *Finish* once you are complete.

Procedure How to Delete a Channel

Perform the following steps to delete an existing channel.

1. Select the channel you want to delete in the left pane.



2. Click *Operations* in the right pane and select *Delete*.

The following confirmation dialog box opens.



3. Click *OK* to delete the channel you selected.

The channel is removed from the list in the left pane.

APPENDIX A

Using the WebLogic Workshop to Access Web Services

Topics:

- Using the WebLogic Workshop to Access SAP R/3 BAPIs
- Running the JWSNAME Web Service from WebLogic Workshop for BAPIs
- Using the WebLogic Workshop to Access SAP R/3 RFCs
- Running the JWSNAME Web Service from WebLogic Workshop for RFCs

This section describes how to access Web services created for SAP R/3 BAPIs and RFCs using the WebLogic Workshop.

Using the WebLogic Workshop to Access SAP R/3 BAPIs

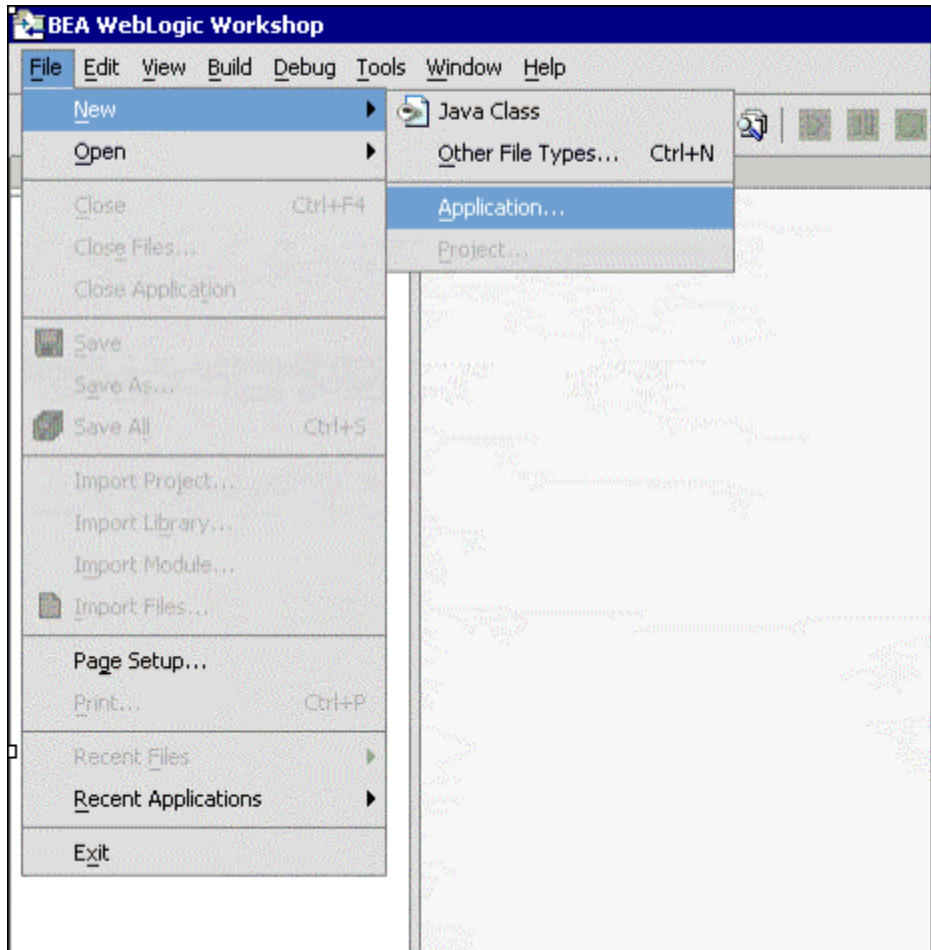
WebLogic Workshop provides a framework for building Web services. The Web services that you build with WebLogic Workshop are enterprise-class services, and WebLogic Workshop provides simple controls for connecting to your enterprise resources. At the same time, WebLogic Workshop simplifies the process of creating Web services by insulating developers from the low-level implementation details that have 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 Access SAP R/3 BAPIs

This procedure assumes you have already created and tested a Web service using iWay Application Explorer. It also assumes you have created the WSDL used to access the service. For more information on creating Web services, see Chapter 4, *Creating and Publishing iWay Business Services*.

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

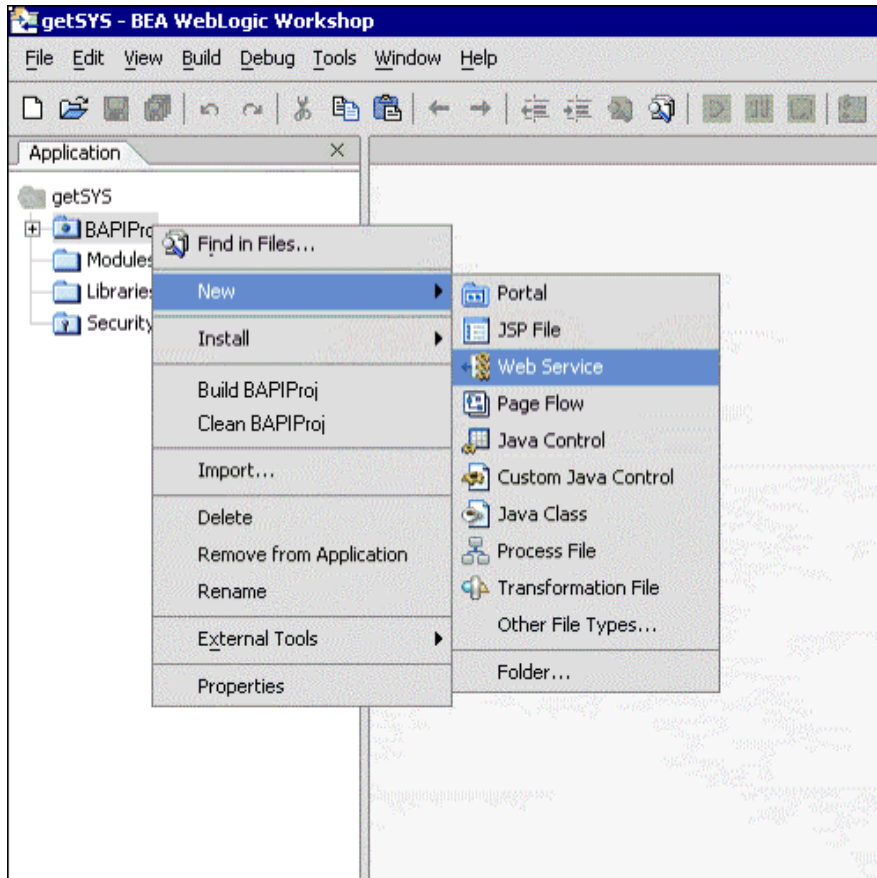
BEA WebLogic Workshop opens.



2. Create a new application.
 - a. From the File menu, select *New* and then, *Application*.
 - b. In the upper-left pane, select all and then, select *Empty Application*.
 - c. In the directory field, type *C:\IWAYSrv*.
 - d. Click *Create*.
3. In the Application tab, right-click the *IWAYSrv* folder and select *New Project*.
4. In the upper-left pane, select all and then, select *Web Project*.
5. In the name field, type *BAPProj* and click *Create*.

The code for a Web service is contained within a JWS (Java Web Service) file. A JWS file is a JAVA file in that it contains code for a Java class. However, because a file with a JWS extension contains the implementation code intended specifically for a Web service class, the extension gives it special meaning in the context of the WebLogic Server.

The New Web Service dialog box opens.

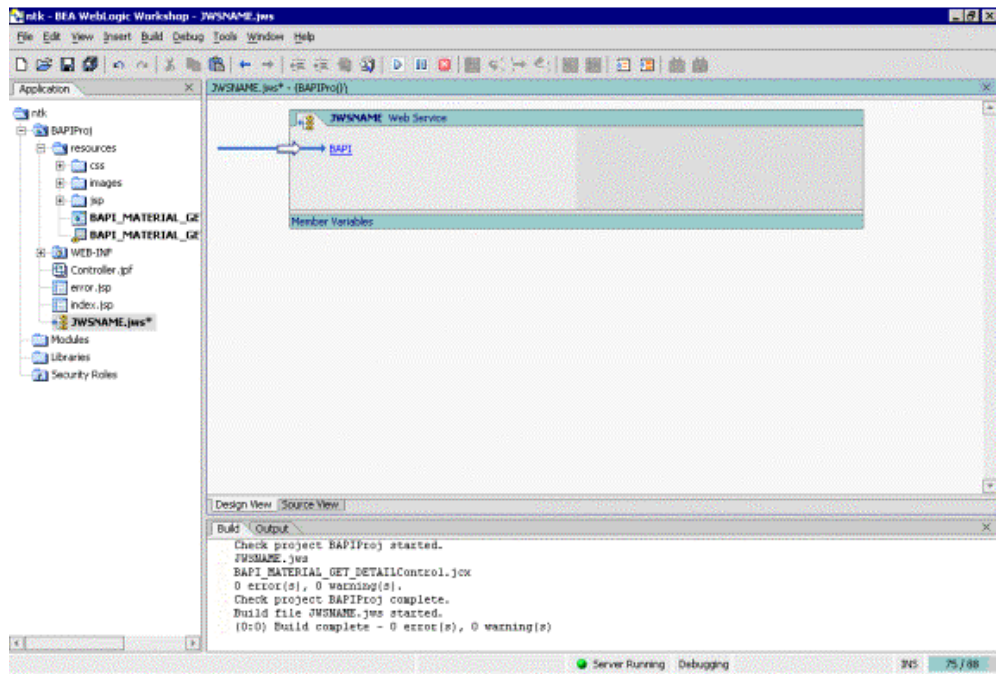


6. In the Application tab, right-click the *BAPIProj* folder.
 - a. Select *New*.
 - b. Select *Web Service*.
7. In the upper-left pane, select all and then, select *Web Service* in the right pane.
 - a. In the name field, type *JWSNAME.jws*.
 - b. Click *Create*.

The design view window opens.

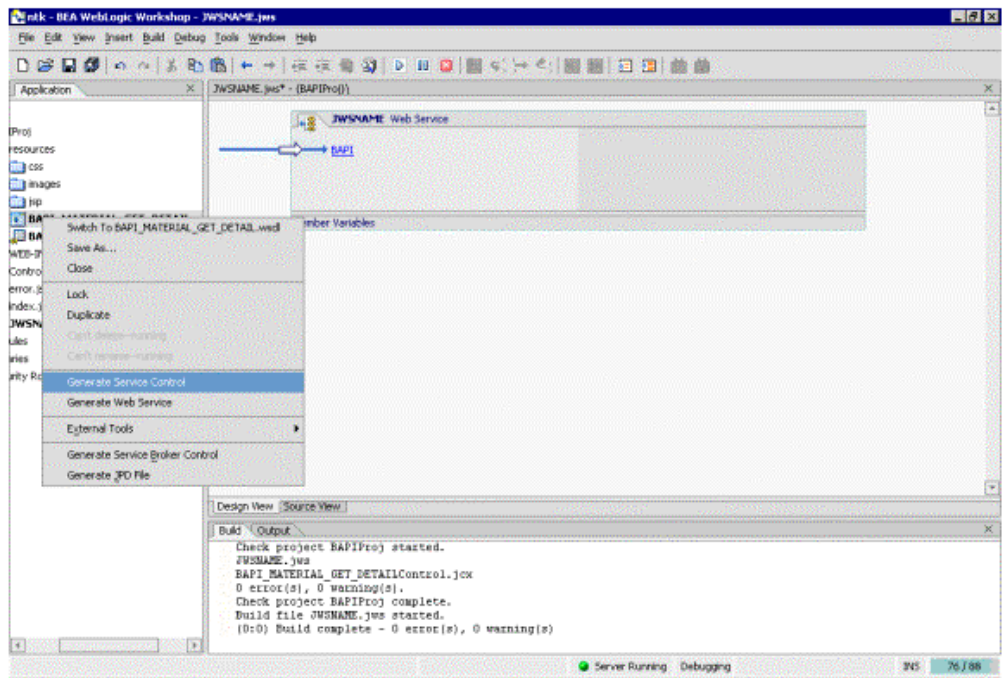
Web services expose their functionality through methods that clients invoke when they want to request something from the Web service. In this case, clients invoke a method to call the BAPI_MATERIAL_GET_DETAIL Control that is exposed later in this procedure.

8. If it is not selected already, click the *Design View* tab.
 - a. From the Insert menu, select *Method*.
 - b. In the space provided, replace *method1* with *BAPI*, and press *Enter*.



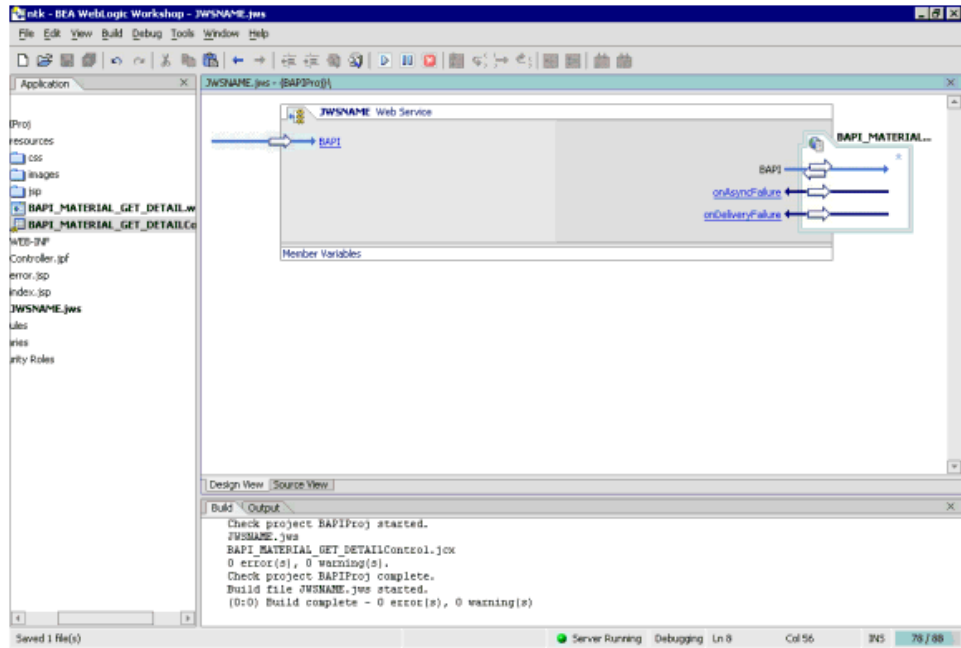
9. Right-click the *resources* sub-folder project and select *Import*.
10. Import the BAPI_MATERIAL_GET_DETAIL.WSDL .

For more information on creating a WSDL file, see Chapter 4, *Creating and Publishing iWay Business Services*.



11. To generate a Java Control file, right-click the BAPI_MATERIAL_GET_DETAIL.wsd/ file and select *Generate Service Control*.

- 12.** Drag the BAPI_MATERIAL_GET_DETAIL.jcx file onto the JWSNAME Web service as follows:



- 13.** Click the *Source View* tab to modify the source code and call the iWay BAPI_MATERIAL_GET_DETAIL Web service.

- a.** Add the following code to the source view:

```
public void
BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input)
{
    BAPI_MATERIAL_GET_DETAILControl.BAPI(input)
}
```

- b.** To save your current work, press *Control + S*.

The resulting Java code should look similar to the following:

```
import resources.BAPI_MATERIAL_GET_DETAILControl;

public class JWSNAME implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.BAPI_MATERIAL_GET_DETAILControl
    BAPI_MATERIAL_GET_DETAILControl;

    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void
    BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input )
    {
        BAPI_MATERIAL_GET_DETAILControl.BAPI(input);
    }
}
```

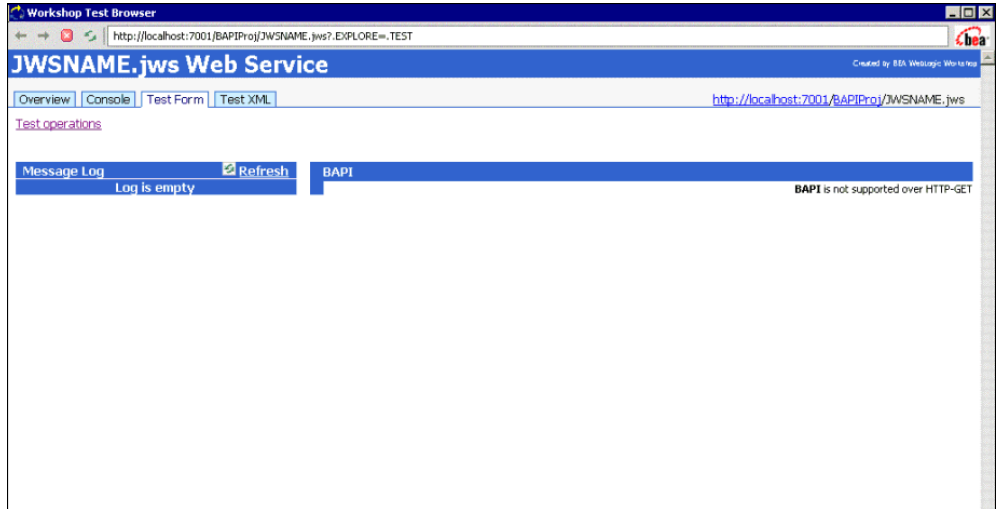
Running the JWSNAME Web Service from WebLogic Workshop for BAPIs

When you create a new Web service tutorial application, you must ensure that WebLogic Server is running while you build your Web service. You can confirm whether WebLogic Server is running by looking at the status bar at the bottom of WebLogic Workshop. If WebLogic Server is running, a green ball appears. If WebLogic Server is not running, a red ball appears. If you see the red ball in the status bar, then start WebLogic Server, as described in the following procedure.

Procedure How to Start WebLogic Server

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

The following test window opens.



4. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
5. Replace the string XML input with the following:


```
<?xml version="1.0" encoding="UTF-8"?>
  <BAPI_MATERIAL_GET_DETAIL>
    <MATERIAL>P-100</MATERIAL>
  </BAPI_MATERIAL_GET_DETAIL/>
```
6. Click the *BAPI* button to submit the request.

After the SOAP request is sent to the iWay Business Services Engine (iBSE), the following response is returned:

The screenshot displays the Workshop Test Browser interface for the JWSNAME.jws Web Service. The browser window shows the URL `http://localhost:7001/BAPITest/JWSNAME.jws?_EXPLORE=&TESTXML&LOGENTRY=1`. The service is titled "JWSNAME.jws Web Service" and includes tabs for Overview, Console, Test Form, and Test XML. The Test XML tab is active, showing the SOAP request and response.

Message Log

- BAP1
- BAP1_MATERIAL_GET_DETAILControlBAP1

External Service Request

Submitted at Thursday, January 8, 2004 2:49:18 AM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2001/soap" xmlns:instance="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:BAP1 xmlns:ns="urn:iwaysoftware:ibse:jul2003:BAP1">
      <ns:BAP1_MATERIAL_GET_DETAIL>
        <ns:MATERIAL>P-100</ns:MATERIAL>
        </ns:BAP1_MATERIAL_GET_DETAIL>
      </ns:BAP1>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

External Service Response

Submitted at Thursday, January 8, 2004 2:49:19 AM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://www.w3.org/2001/soap"
xmlns:instance="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <BAP1Response xmlns="urn:iwaysoftware:ibse:jul2003:BAP1" response=">
      <BAP1_MATERIAL_GET_DETAIL_Response xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:soap="http://www.w3.org/2001/soap" xmlns:instance="http://www.w3.org/2001/XMLSchema"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <MATERIALPLANTDATA>
          <PLANT>P-100</PLANT>
          <ISSUE_UNIT>
            <ISSUE_UNIT>
              <MATERIALPLANTDATA>
                <PRICE_CTRL>
                  <PRICE_CTRL>
                    <MOVING_PRICE>
                      <STD_PRICE>
                        <PRICE_UNIT>
                          <CURRENCY>
                            <CURRENCY_ISO>
                              <MATERIALVALUATIONDATA>
                                <MATERIAL_GENERAL_DATA>
                                  <MATERIAL_DESC>Pump PRECISION 100</MATERIAL_DESC>
                                  <OLD_MAT_NO>
                                    <OLD_MAT_NO>
```

The previous sample is a very simple example of calling an iWay Web service.

You may want to perform more complex operations in your workflow. The following code represents sample Java code used to calculate the execution time of the Web service. You can do similar coding for benchmarking or other purposes.

```
import resources.BAPI_MATERIAL_GET_DETAILControl;

import java.io.*;
import java.lang.*;
import java.util.*;

public class JWSNAME implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.BAPI_MATERIAL_GET_DETAILControl
    BAPI_MATERIAL_GET_DETAILControl;

    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void
    BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input)
    throws Exception    {

        File outFile=new File("RESULTS.txt"); //creating an output file
        FileWriter out=new FileWriter(outFile); //creating a fileWriter for
        the output file

        long diff=0; //used to store the execution time

        Calendar cal_start=Calendar.getInstance(TimeZone.getTimeZone("EST"));
        //creating a start calendar
        System.out.println("<<<< start: "+ cal_start.getTimeInMillis());
        //Display the start time of execution to the WEBLOGIC CONSOLE

        BAPI_MATERIAL_GET_DETAILControl.BAPI(input);

        Calendar
        cal_end=Calendar.getInstance(TimeZone.getTimeZone("EST")); //create end
        calendar

        System.out.println("<<<< end: "+ cal_end.getTimeInMillis());
        Display the end time of execution to the WEBLOGIC CONSOLE
    }
}
```

```
        diff=cal_end.getTimeInMillis()-cal_start.getTimeInMillis();
//Calculating the difference (execution time)
        System.out.println("<<<< EXECUTION time in Milliseconds:" +diff);
//Displaying the execution time to the WEBLOGIC Console

//writing to file
        out.write( "start time: "+ cal_start.getTimeInMillis()+"\n");
        out.write("end time: "+cal_end.getTimeInMillis()+"\n");
        out.write("execution time : "+diff+"\n");

        out.close(); //closing file

    }
}
```

The results of the execution are saved in a file as follows:

```
start time: 1073598362655
end time: 1073598362775
execution time : 120
```

Using the WebLogic Workshop to Access SAP R/3 RFCs

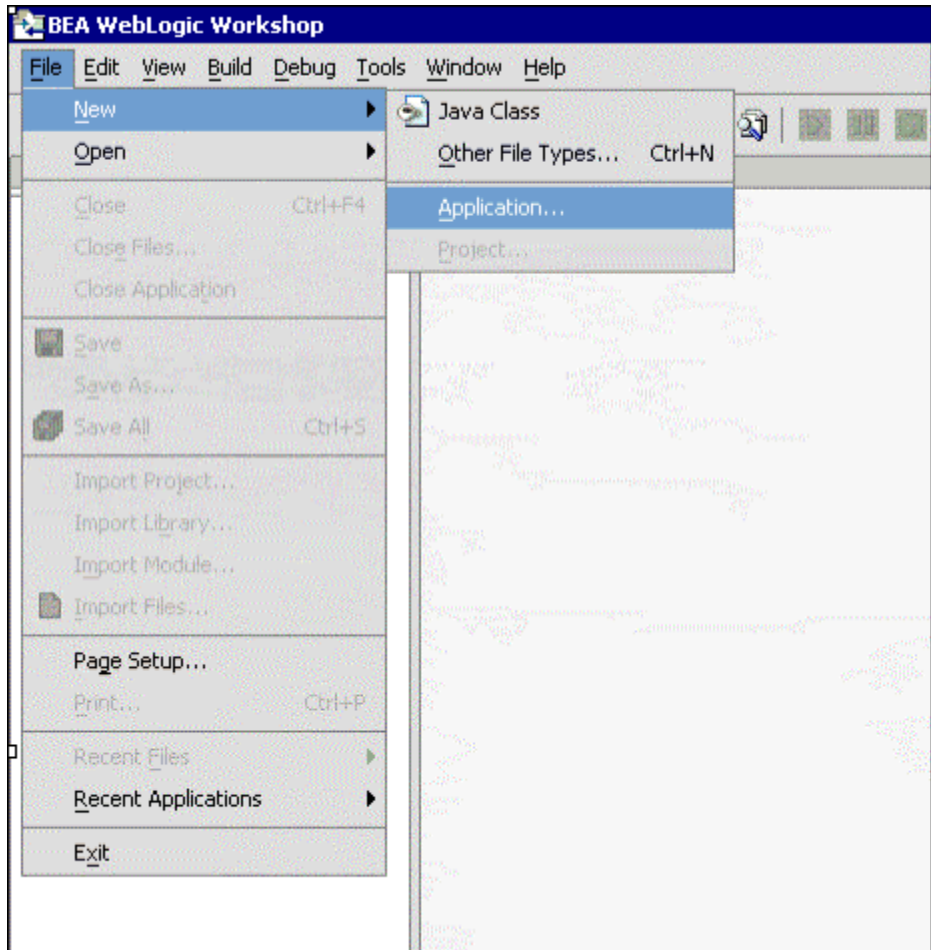
WebLogic Workshop provides a framework for building Web services. The Web services that you build with WebLogic Workshop are enterprise-class services, and WebLogic Workshop provides simple controls for connecting to your enterprise resources. At the same time, WebLogic Workshop simplifies the process of creating Web services by insulating developers from the low-level implementation details that have 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 Access SAP R/3 RFCs

This procedure assumes you have already created and tested a Web service using iWay Application Explorer. It also assumes you have created the WSDL used to access the service. For more information on creating Web services, see Chapter 4, *Creating and Publishing iWay Business Services*.

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

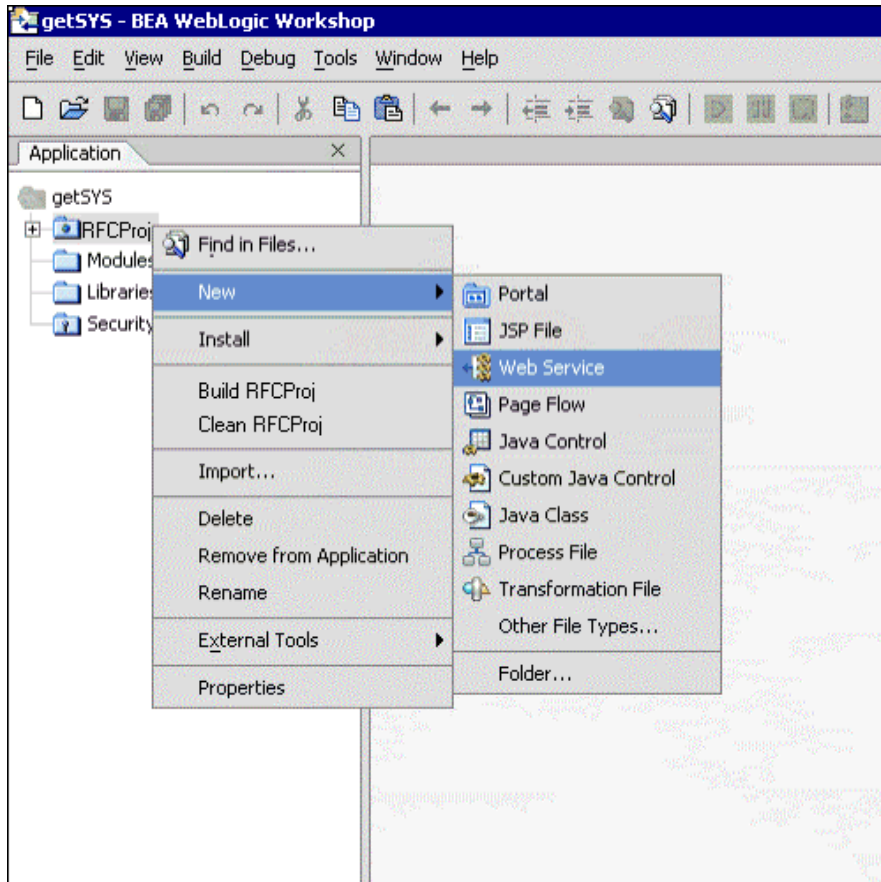
BEA WebLogic Workshop opens.



2. Create a new application.
 - a. From the File menu, select *New* and then, *Application*.
 - b. In the upper-left pane, select all and then, select *Empty Application*.
 - c. In the directory field, type *C:\IWAYSrv*.
 - d. Click *Create*.
3. In the Application tab, right-click the *IWAYSrv* folder and select *New Project*.
4. In the upper-left pane, select all and then, select *Web Project*.
5. In the name field, type *RFCProj* and click *Create*.

The code for a Web service is contained within a JWS (Java Web Service) file. A JWS file is a JAVA file in that it contains code for a Java class. However, because a file with a JWS extension contains the implementation code intended specifically for a Web service class, the extension gives it special meaning in the context of the WebLogic Server.

The New Web Service dialog box opens.

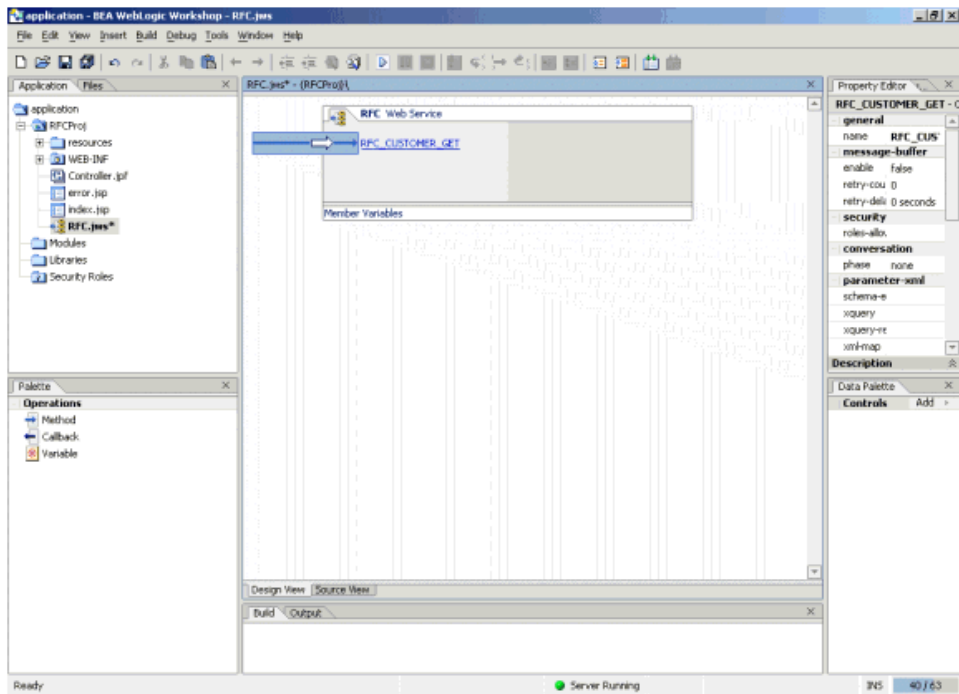


6. In the Application tab, right-click the *RFCProj* folder.
 - a. Select *New*.
 - b. Select *Web Service*.
7. In the upper-left pane, select all and then, select *Web Service* in the right pane.
 - a. In the name field, type *RFC.jws*.
 - b. Click *Create*.

The design view window opens.

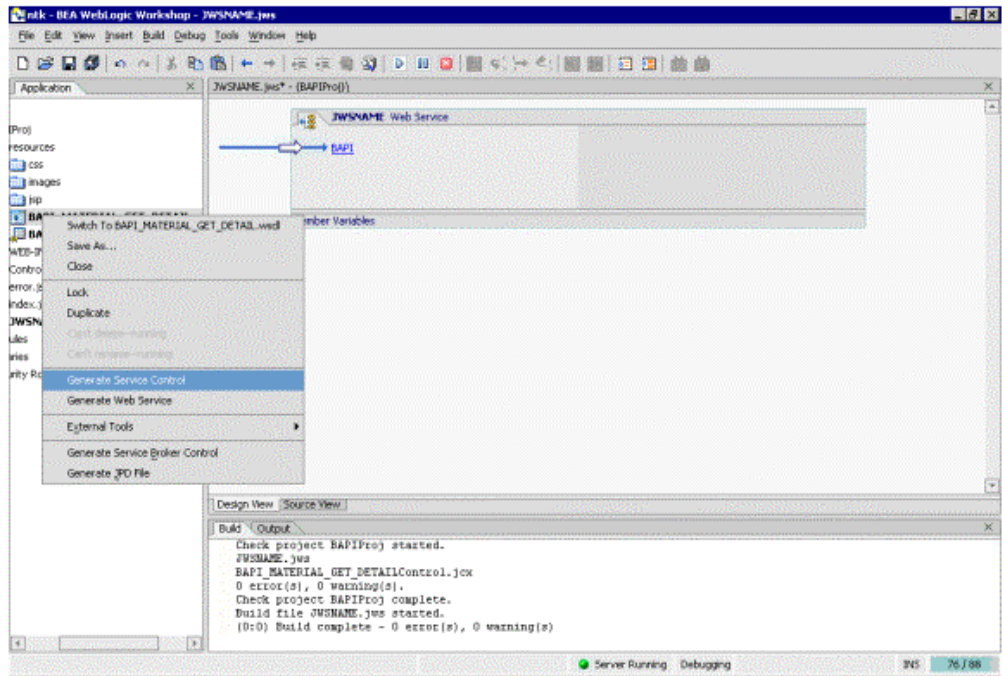
Web services expose their functionality through methods that clients invoke when they want to request something from the Web service. In this case, clients invoke a method to call the `RFC_CUSTOMER_GET` Control that is exposed later in this procedure.

8. If it is not selected already, click the *Design View* tab.
 - a. From the Insert menu, select *Method*.
 - b. In the space provided, replace *method1* with `RFC_CUSTOMER_GET`, and press *Enter*.



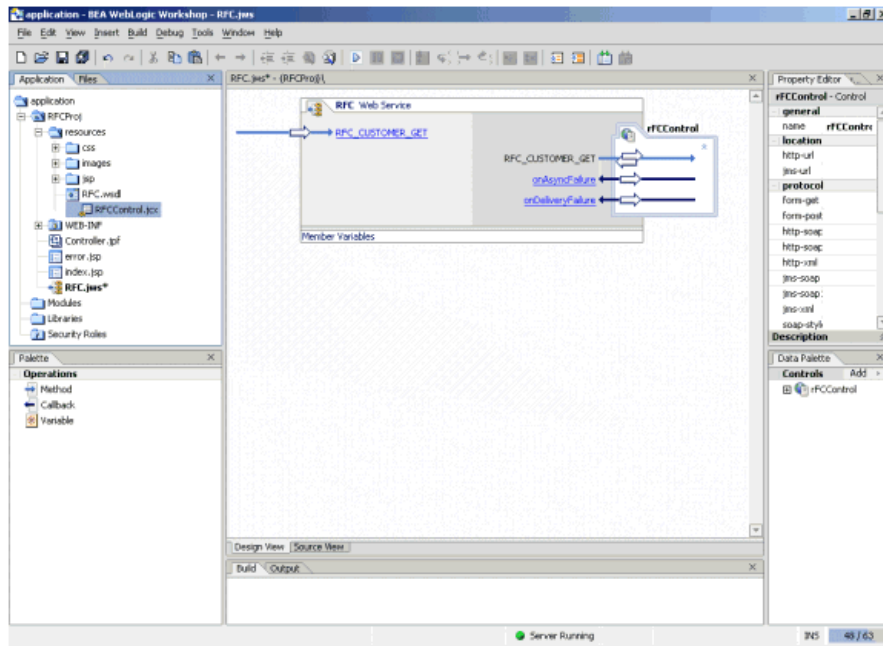
9. Right-click the *resources* sub-folder project and select *Import*.
10. Import the *RFC.WSDL*.

For more information on creating a WSDL file, see Chapter 4, *Creating and Publishing iWay Business Services*.



11. To generate a Java Control file, right-click the *RFC_CUSTOMER_GET.wsdl* file and select *Generate Service Control*.

12. Drag the *RFControl.jcx* file onto the JWSNAME Web service as follows:



13. Click the *Source View* tab to modify the source code and call the iWay RFC Web service.

a. Add the following code to the source view:

```
public void RFC_CUSTOMER_GET(RFControl.RFC_CUSTOMER_GET input)
{
    RFControl.RFC_CUSTOMER_GET(input)
}
```

b. To save your current work, press *Control + S*.

The resulting Java code should look similar to the following:

```
import resources.RFCControl;

public class RFC implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.RFCControl RFCControl;

    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void RFC_CUSTOMER_GET(RFCControl.RFC_CUSTOMER_GET input )
    {
        RFCControl.RFC_CUSTOMER_GET(input);
    }
}
```

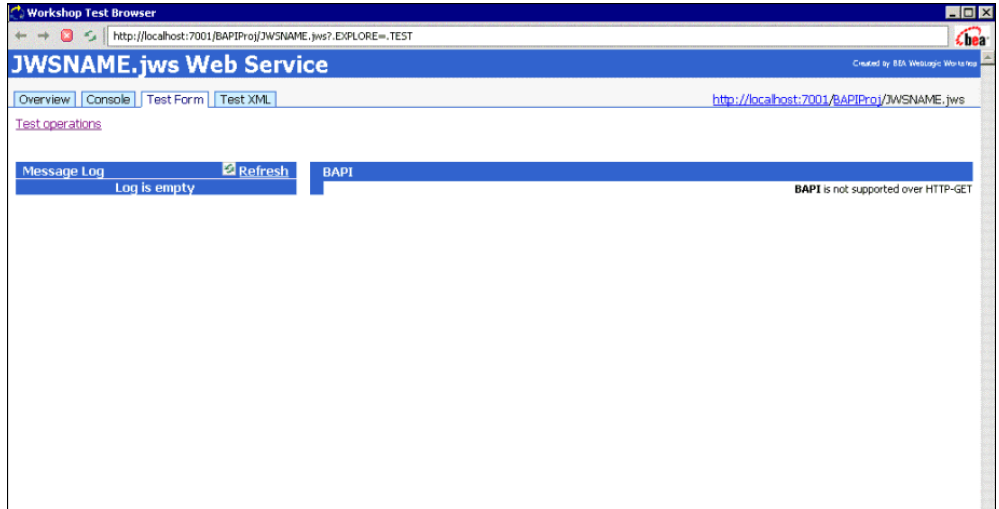
Running the JWSNAME Web Service from WebLogic Workshop for RFCs

When you create a new Web service tutorial application, you must ensure that WebLogic Server is running while you build your Web service. You can confirm whether WebLogic Server is running by looking at the status bar at the bottom of WebLogic Workshop. If WebLogic Server is running, a green ball appears. If WebLogic Server is not running, a red ball appears. If you see the red ball in the status bar, then start WebLogic Server, as described in the following procedure.

Procedure How to Start WebLogic Server

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

The following test window opens.



4. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
5. Replace the string XML input with the following:


```
<RFC_CUSTOMER_GET xmlns="http://www.openuri.org/">
  <input>
    <KUNNR>0000401026</KUNNR>
    <NAME1></NAME1>
  </input>
</RFC_CUSTOMER_GET>
```
6. Click the *RFC_CUSTOMER_GET* button to submit the request.

After the SOAP request is sent to the iWay Business Services Engine (iBSE), the following response is returned:

The screenshot displays the Workshop Test Browser interface for the RFC.jws Web Service. The browser window shows the URL `http://localhost:7001/RFCPro/RFC.jws?_EXPLORE=&_TESTXML&_LOGENTRY=1`. The service is titled "RFC.jws Web Service" and is created by iB3, iWay Software. The interface includes tabs for Overview, Console, Test Form, and Test XML. The Test XML tab is active, showing the SOAP request and response.

Message Log

- RFC_CUSTOMER_GET
- RFCControlRFC_CUSTOMER_GET
- Clear Log

External Service Request

Submitted at Thursday, January 8, 2004 4:38:24 PM EST

```
<?XML:ENVELOPE xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:si="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:RFC_CUSTOMER_GET xmlns:ns="urn:iwaysoftware:ibse:jul2003:RFC_CUSTOMER_GET">
      <ns:RFC_CUSTOMER_GET>
        <ns:KUNNR>0000401026</ns:KUNNR>
        <ns:NAME1></ns:NAME1>
        </ns:RFC_CUSTOMER_GET>
      </ns:RFC_CUSTOMER_GET>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

External Service Response

Submitted at Thursday, January 8, 2004 4:38:26 PM EST

```
<?XML:ENVELOPE xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:si="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <RFC_CUSTOMER_GETResponse id="A907445038EB6FB25707B60FEB9F296"
      xmlns="urn:iwaysoftware:ibse:jul2003:RFC_CUSTOMER_GET:response">
      <RFC_CUSTOMER_GET:Response>
        <CUSTOMER_T>
          <bean>
            <KUNNR>0000401026</KUNNR>
            <ANRED>Ms</ANRED>
            <NAME1>MARIE BURHAM</NAME1>
            <PFACH>1030</PFACH>
            <STRAS>200 Fremont Drive</STRAS>
            <PSTL2>92101</PSTL2>
            <ORT01>SAN DIEGO</ORT01>
            <TELF1>303-789-012</TELF1>
            <TELF2></TELF2>
          </bean>
        </CUSTOMER_T>
      </RFC_CUSTOMER_GET:Response>
    </RFC_CUSTOMER_GET:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


The previous sample is a very simple example of calling an iWay Web service.

You may want to perform more complex operations in your workflow. The following code represents sample Java code used to calculate the execution time of the Web service. You can do similar coding for benchmarking or other purposes.

```
import resources.RFCControl;

import java.io.*;
import java.lang.*;
import java.util.*;

public class RFC implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.RFCControl RFCControl;

    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void RFC_CUSTOMER_GET(RFCControl.RFC_CUSTOMER_GET input)
    throws Exception    {

        File outFile=new File("RESULTS.txt"); //creating an output file
        FileWriter out=new FileWriter(outFile); //creating a fileWriter for
        the output file

        long diff=0; //used to store the execution time

        Calendar cal_start=Calendar.getInstance(TimeZone.getTimeZone("EST"));
        //creating a start calendar
        System.out.println("<<<< start: "+ cal_start.getTimeInMillis());
        //Display the start time of execution to the WEBLOGIC CONSOLE

        RFCControl.RFC_CUSTOMER_GET(input);

        Calendar
        cal_end=Calendar.getInstance(TimeZone.getTimeZone("EST")); //create end
        calendar

        System.out.println("<<<< end: "+ cal_end.getTimeInMillis());
        Display the end time of execution to the WEBLOGIC CONSOLE
        diff=cal_end.getTimeInMillis()-cal_start.getTimeInMillis();
        //Calculating the difference (execution time)
```

```
        System.out.println("<<<< EXECUTION time in Milliseconds:" +diff);  
//Displaying the execution time to the WEBLOGIC Console  
  
//writing to file  
        out.write( "start time: "+ cal_start.getTimeInMillis()+"\n");  
        out.write("end time: "+cal_end.getTimeInMillis()+"\n");  
        out.write("execution time : "+diff+"\n");  
  
        out.close(); //closing file  
  
    }  
}
```

The results of the execution are saved in a file as follows:

```
start time: 1073598362650  
end time: 1073598362775  
execution time : 125
```

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:

