iWay Software
An Information Builders Company

iWay

Servlet Application Explorer for BEA WebLogic User's Guide
Version 5 Release 5

DN3501335.1204

Preface

This document explains how to use the 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	*Introducing the Servlet Application Explorer*	Provides an overview of the Servlet Application Explorer and its features.
2	*Creating Targets for Enterprise Information Systems*	Describes how to start the Servlet Application Explorer and create and modify a new target for an Enterprise Information System.
3	*Browsing Metadata and Creating XML Schemas*	Describes how to use the Servlet Application Explorer to browse metadata and create XML schemas.
4	*Creating and Publishing Integration Business Services*	Describes how to use the Servlet Application Explorer to create and publish an Integration Business Service.
5	*Using Event Adapters*	Describes how to use the Servlet Application Explorer to connect to an Enterprise Information System (EIS) and generate events.
6	*Using Integration Business Services Policy-Based Security*	Describes how Integration Business Services Policy-based security works and how to configure it.
A	*Using the WebLogic Workshop to Access Web Services*	Describes how to access Web services created for SAP R/3 BAPIs and RFCs using the WebLogic Workshop.

Documentation Conventions

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

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

Related Publications

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

Customer Support

Do you have questions about the Servlet Application Explorer 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 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.

Help Us to Serve You Better

To help our consultants answer your questions effectively when you call, please provide the following information:

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

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

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

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

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

  - Configuration files (all are applicable):

    .xch files

    config.xml file

    base.xml file

    repository.xml file

    ibserepo.xml file

    .dic files

    .rules files

  - Environment variable settings:

    IWAY55

    IWAY55OEM

    CLASSPATH

    JAVA_HOME

    ACBDIR

    CBDIR (UNIX)

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

User Feedback

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

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

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

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

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

Contents

CHAPTER 1

Introducing the Servlet Application Explorer

Topics:

- Key Features
- Installing and Configuring the Servlet Application Explorer

The Servlet Application Explorer (AE) 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 Enterprise Connector for J2EE Connector Architecture (JCA) or Integration Business Services Engine (iBSE).

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

Key Features

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

The AE uses vendor-provided interfaces and in-depth knowledge of application systems to access and browse business object metadata. Once an object is selected, the AE 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 AE 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 Servlet Application Explorer

Before you install the AE, the target EIS (for example, SAP R/3) must be installed, configured, and available for client access. The AE 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 Servlet Application Explorer, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

CHAPTER 2

Creating Targets for Enterprise Information Systems

Topics:

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

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

Starting the Servlet Application Explorer

Before you can use the Servlet Application Explorer (AE) , 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:

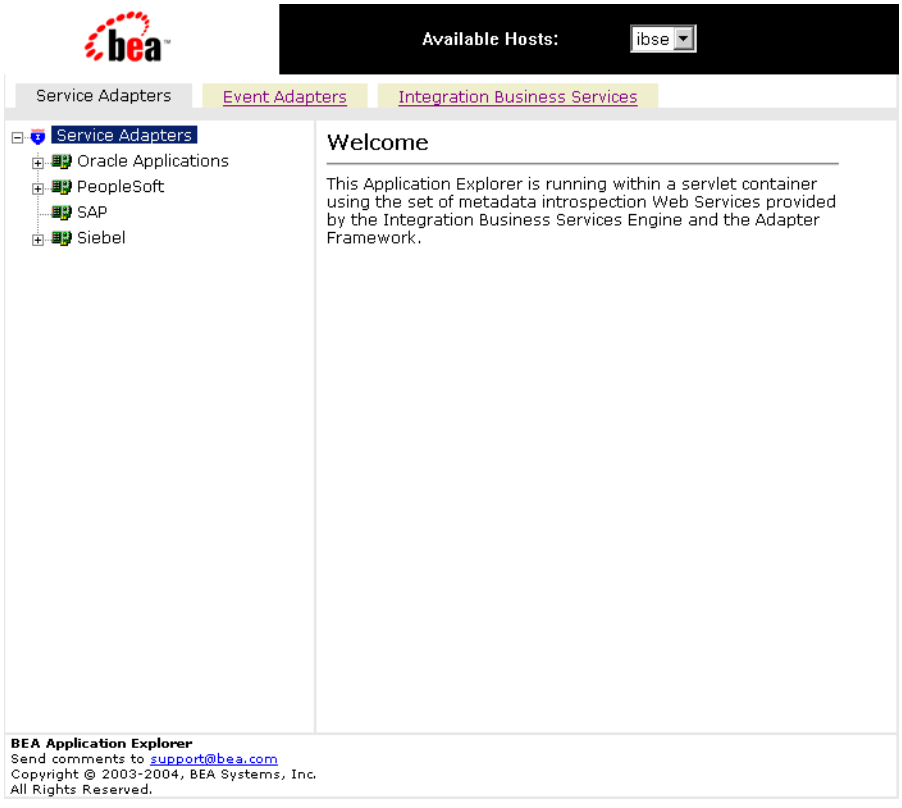
hostname

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 AE, the following window opens.



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

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

Connecting to an Enterprise Information System

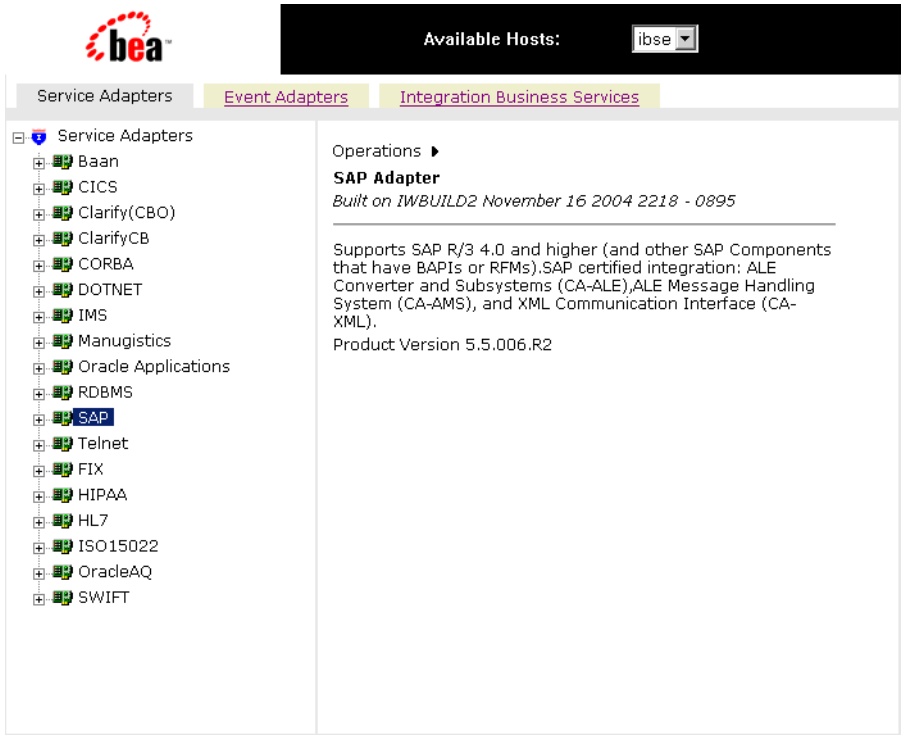
To browse an application system's business objects, you must create a target for the system you intend to use. This target serves as your connection point and is automatically saved after it is created. You must establish a connection to the system every time you start the Application Explorer (AE) or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

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

Procedure How to Create a New Target

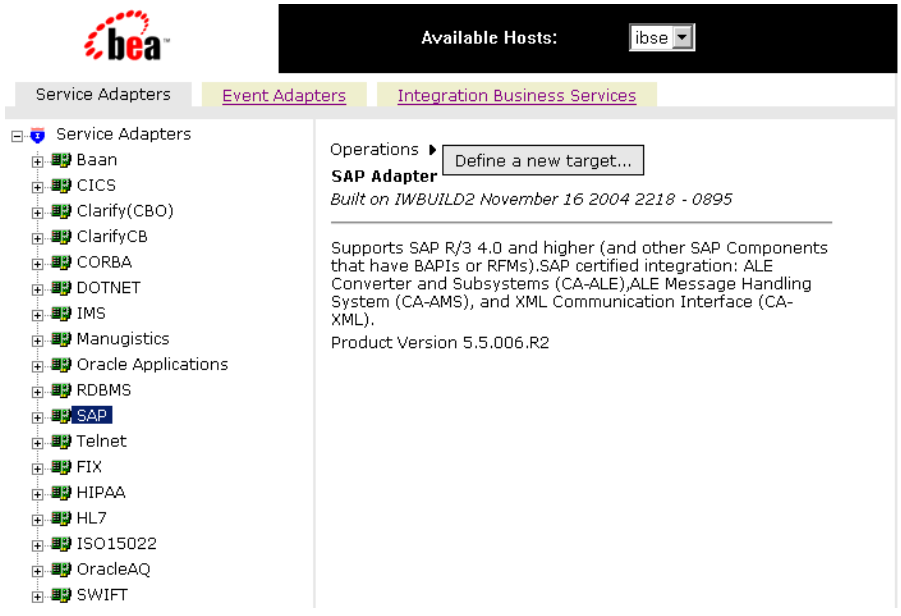
To connect to an application system for the first time, you must create a new target. For example, to create a target for SAP R/3, perform the following steps:

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

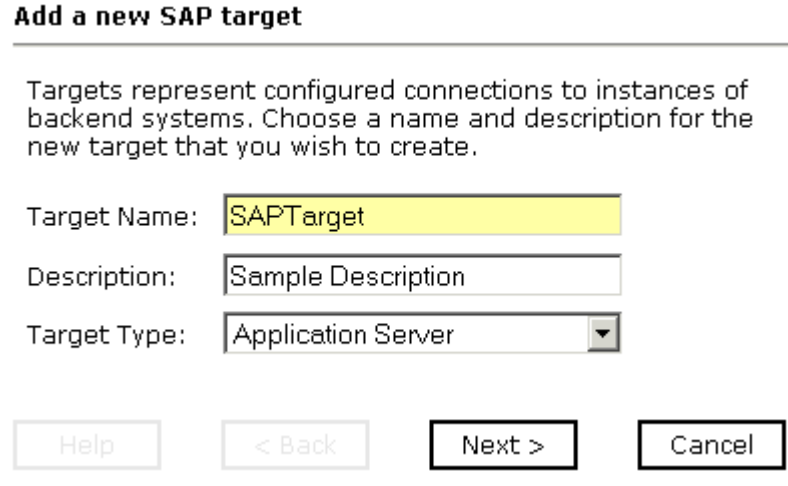


   Descriptive information (for example, title and product version) regarding the iWay Adapter and application system to which you are connecting displays in the right pane.

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



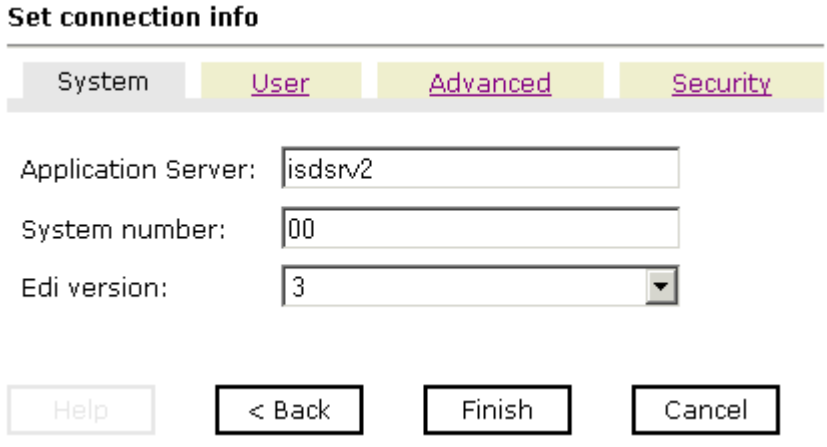
The Add a new SAP target window displays in the right pane.



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*.

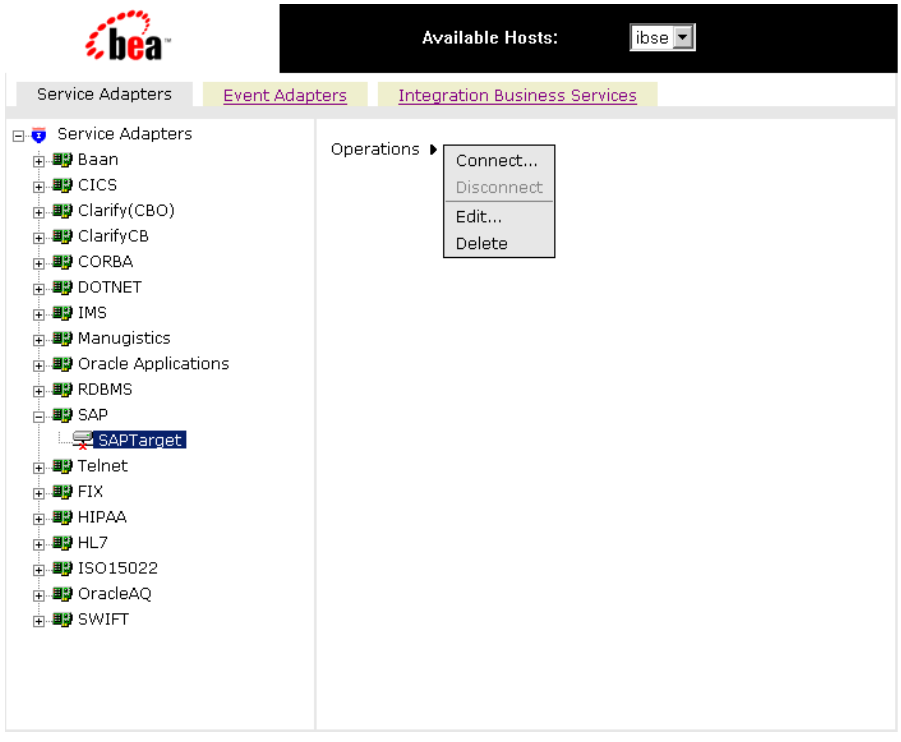
   The target (SAPTarget) is now listed beneath the SAP node in the left pane.

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

Procedure How to Connect to a Target

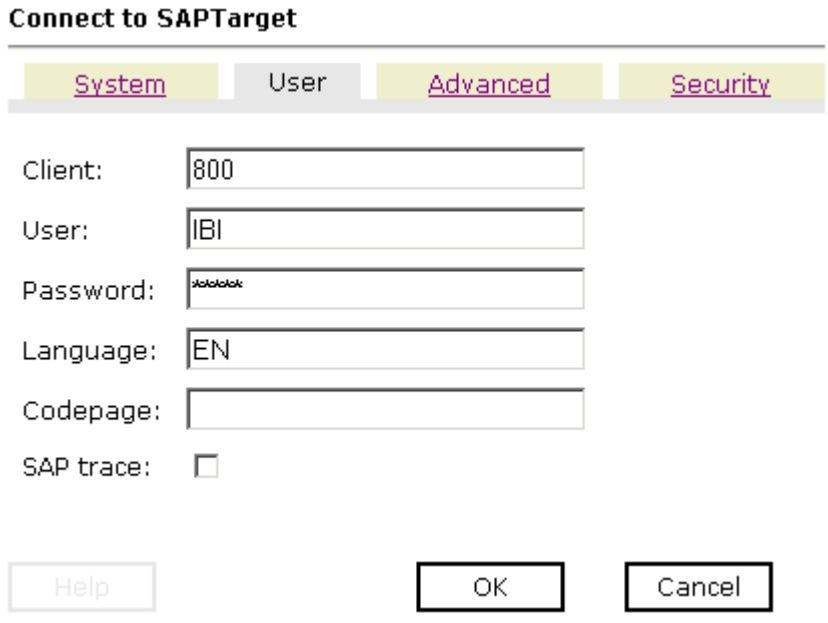
You must use the target you defined to connect to an application. For example, to connect to SAP R/3, perform the following steps.

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



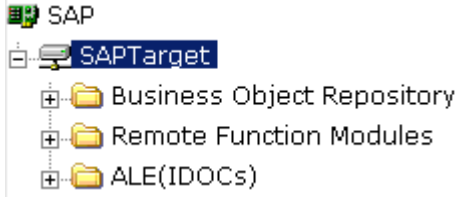
2. Click *Operations* in the right pane and select *Connect*.

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.

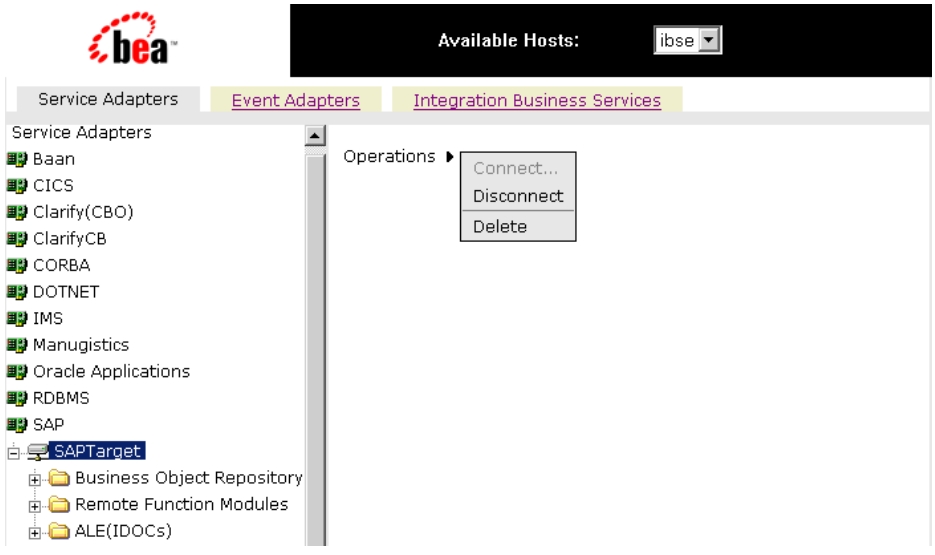


4. Expand the target node to reveal the application system's business objects.

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.

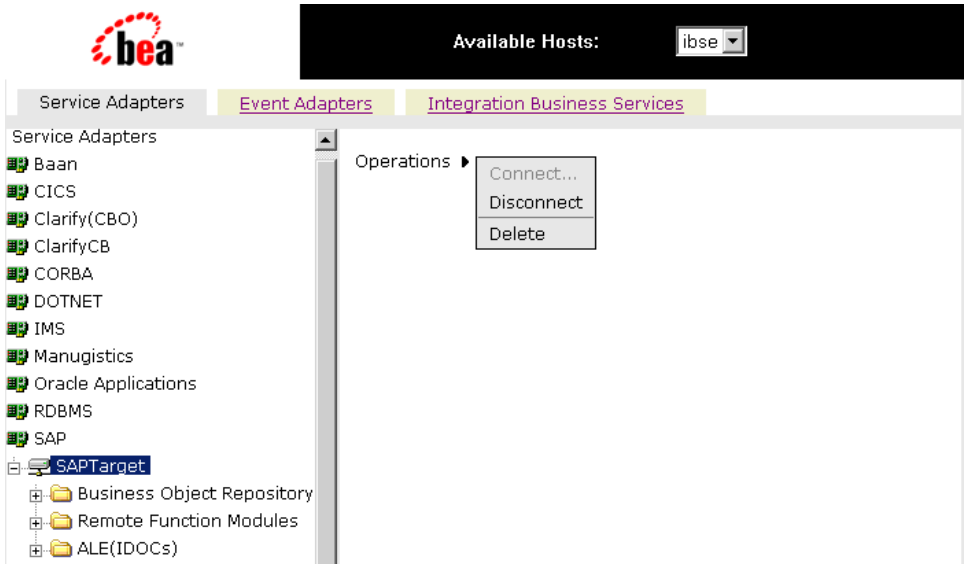
Modifying Targets

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

Procedure How to Edit a Target

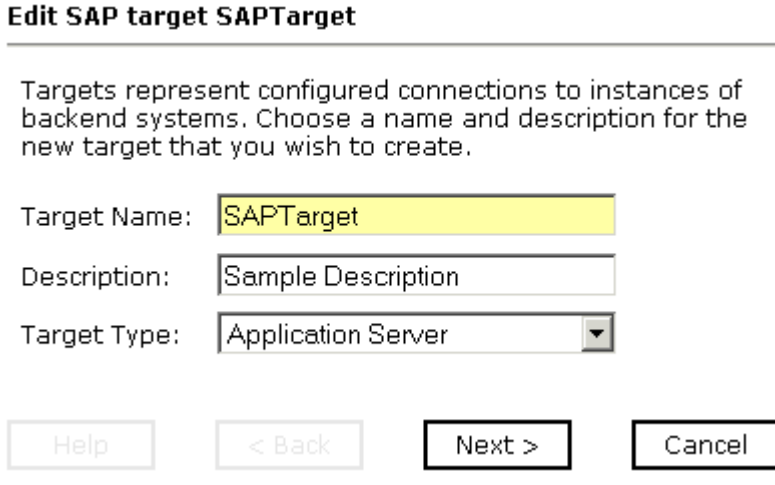
To edit a target in the Servlet Application Explorer:

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.

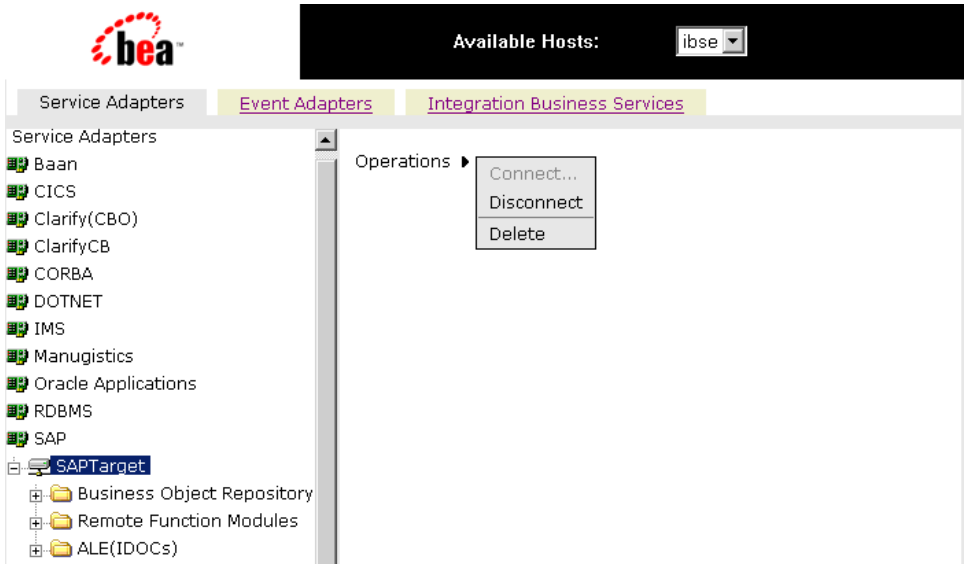


3. Modify the connection information.

Procedure How to Delete a Target

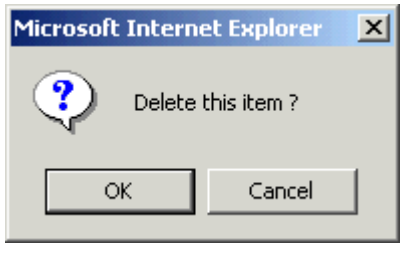
To delete a target in the Servlet Application Explorer:

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 Servlet Application Explorer (AE) 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 AE supports access to several Enterprise Information Systems. Even though the underlying technology used to access them varies significantly, the AE user interface varies only slightly to accomodate EIS differences. In this section, AE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the AE 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 AE with additional systems, see the iWay Adapter documentation for that particular EIS.

Viewing Application System Objects

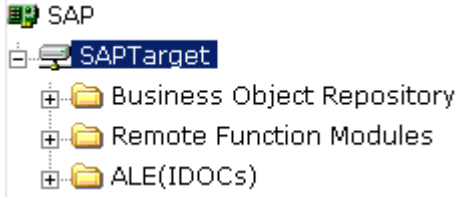
Once you are connected to an EIS, the Servlet Application Explorer allows you to explore and browse business object metadata. For example, the AE enables you to view SAP R/3 BAPI, RFC, and iDOC metadata stored in the SAP Business Object repository.

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).

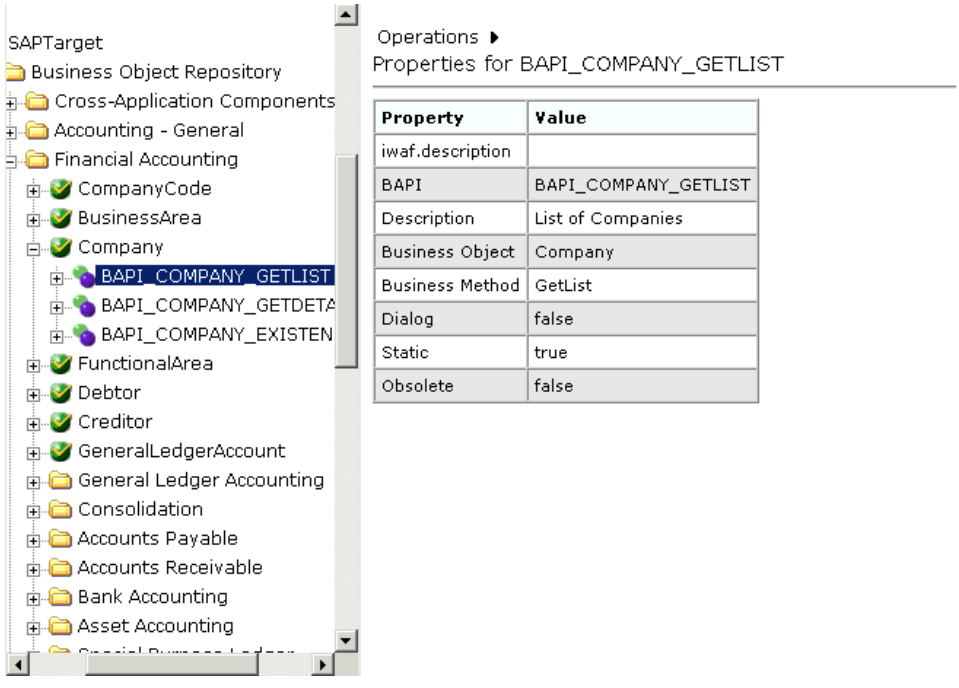
   This expands the target to expose the available system objects.



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.



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

3. Click the icon to the left of the *Financial Accounting* group.

   A list of business objects related to Financial Accounting display.

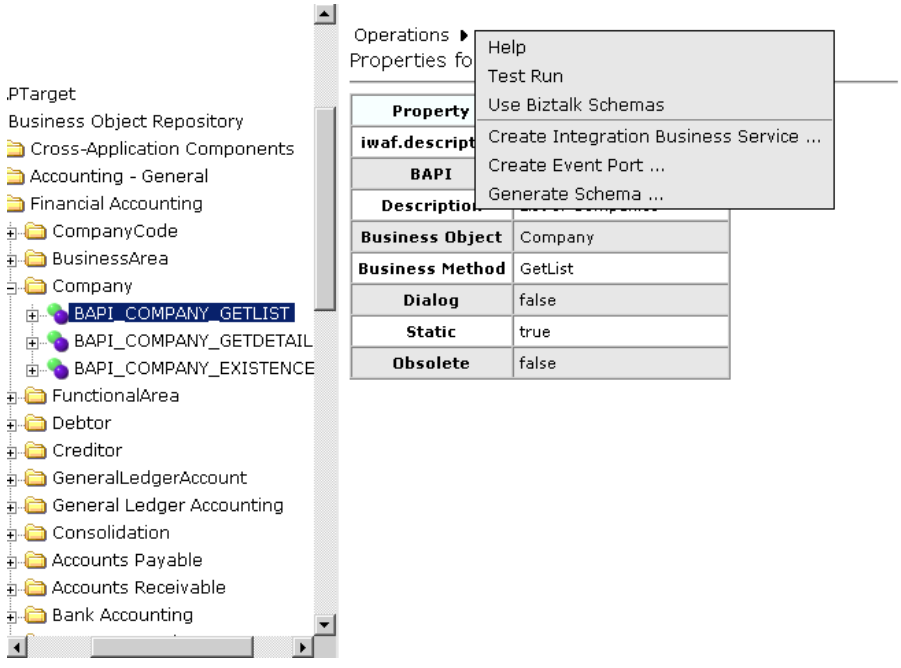
4. Scroll down and click on the icon to the left of the Company business object.

   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.

   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:

- Help.
- Test Run.
- Use Biztalk Schemas.
- Create Integration Business Service.
- Create 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 AE with additional systems, see the iWay Adapter documentation for that particular EIS.

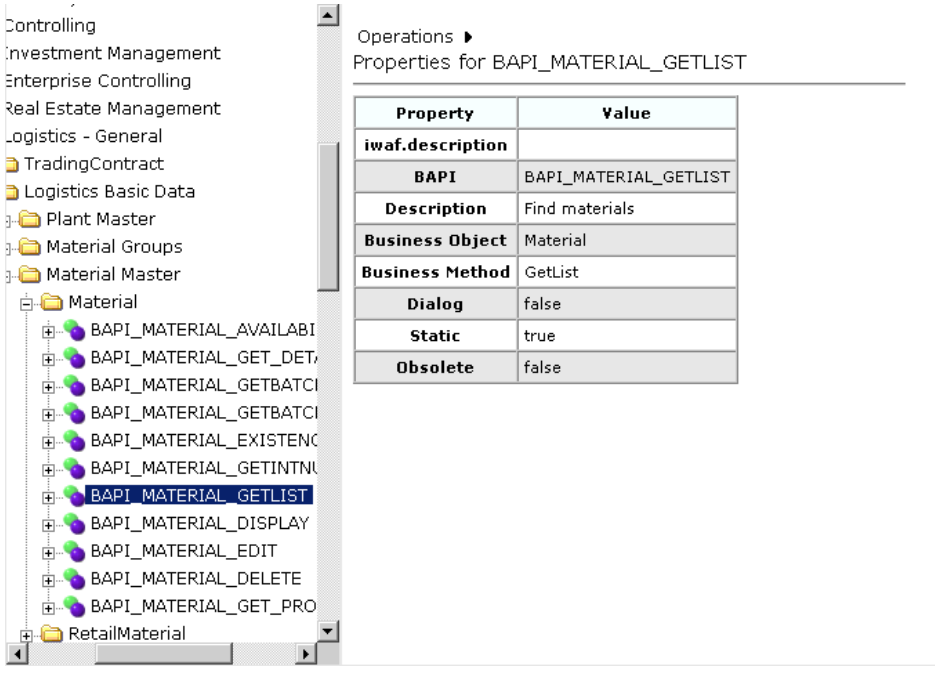
Creating XML Schemas

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

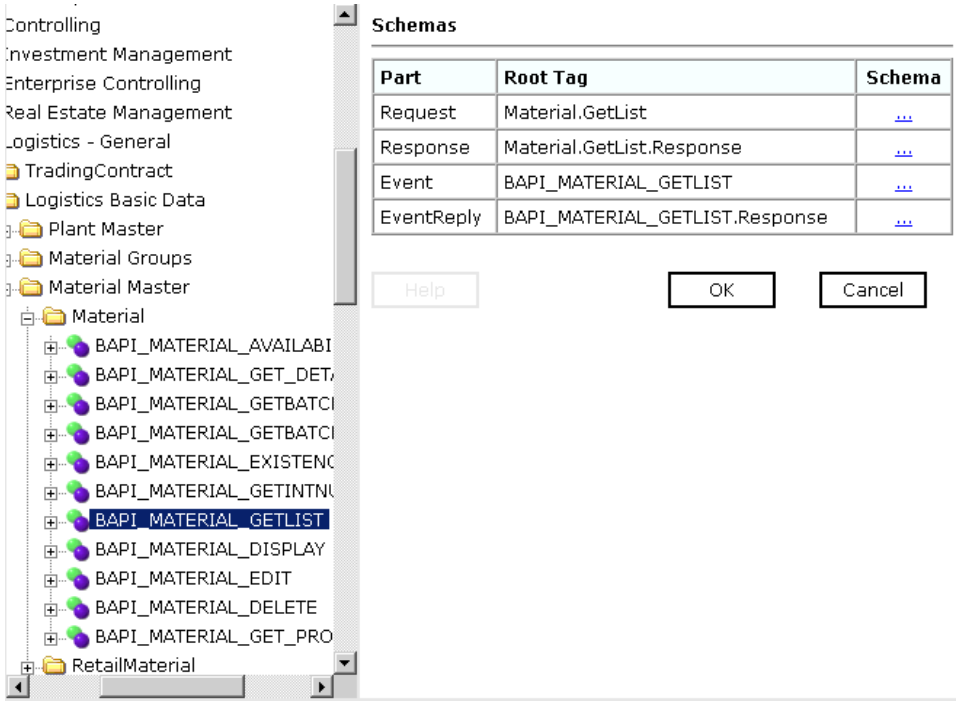
For the SAP BAPI method called BAPI_MATERIAL_GETLIST, perform the following steps to create XML request and response schemas:

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



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

The following graphic shows the Schemas pane that opens on the right. Request, response, and event schemas are created for your business object.

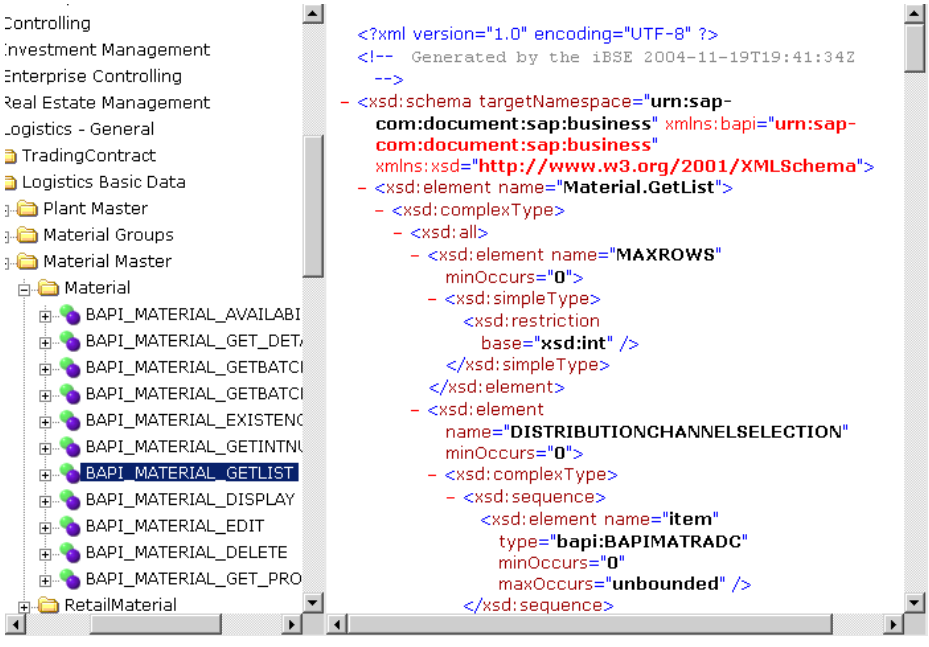


3. From the Schema Support drop-down list select *SAP Interface Repository*.

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

4. 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.



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

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

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

CHAPTER 4

Creating and Publishing Integration Business Services

Topics:

- Understanding Integration Business Services
- Creating Integration Business Services

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

In this section, AE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the AE 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 AE with additional systems, see the iWay Adapter documentation for that particular EIS.

Understanding Integration Business Services

The Servlet Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the iWay Adapter. The Integration Business Services Engine exposes functionality as Web services. It serves as a gateway to hetergeneous 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 service 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 Integration Business Service for the object you wish to use with your adapter.

By default, Integration Business Services that are created through the AE are saved as files using the .ibs extension, which is a proprietary WSDL (Web Services Description Language) format used by iWay Software. To use the Integration Business Service as a resource in a WebLogic workflow, you must rename the .ibs extension to .wsdl. For more information on how to generate WSDL from an Integration Business Service for use with WebLogic, see *Generating WSDL From a Web Service* on page 4-8.

Creating Integration Business Services

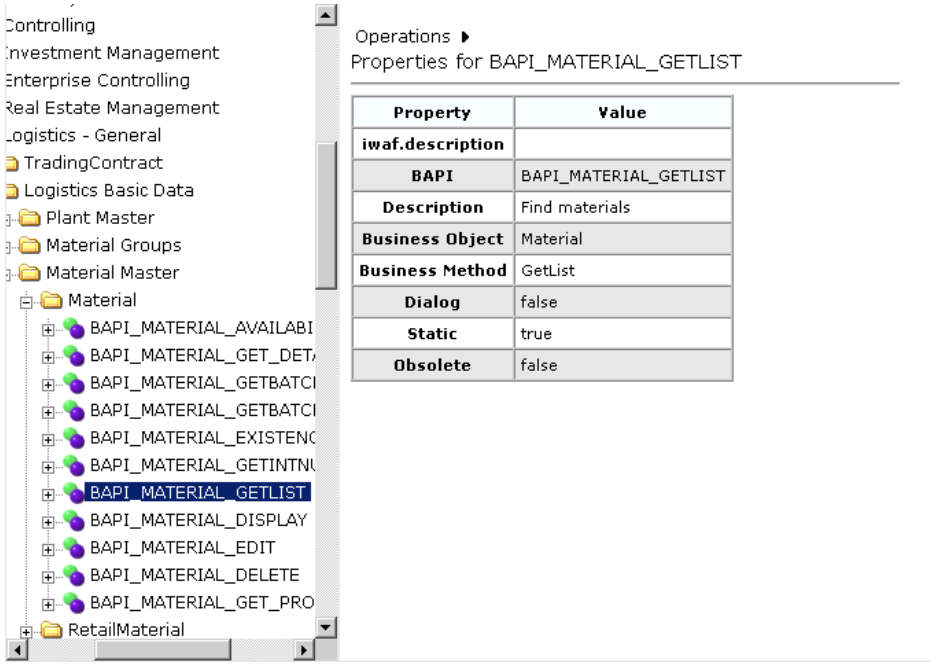
The following section describes how to create Integration Business Services using the Servlet Application Explorer.

Procedure **How to Create an Integration Business Service**

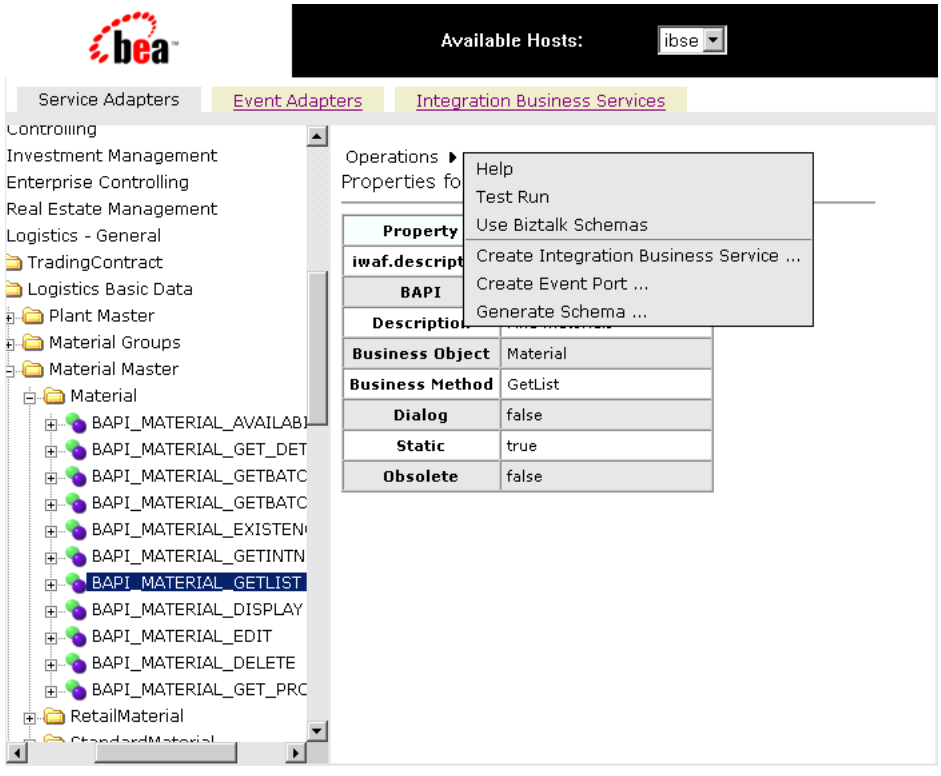
Once you have connected to your application system and created an XML schema for a business object you can create an Integration Business Service for the object.

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 Integration Business Service that will return a list of materials.

1. In the left pane, select the BAPI_MATERIAL_GETLIST method in the Business Object Repository.



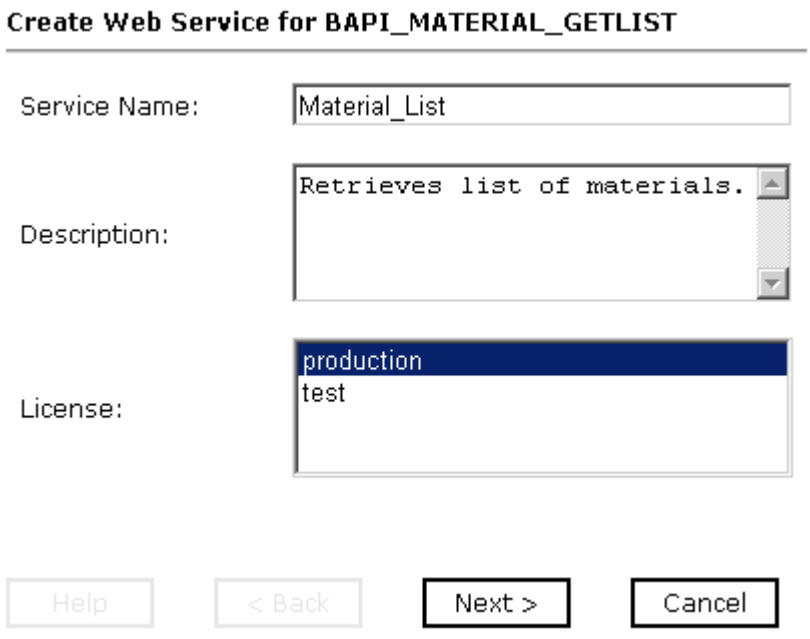
2. Click *Operations* in the right pane and select *Create Integration Business Service*.



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

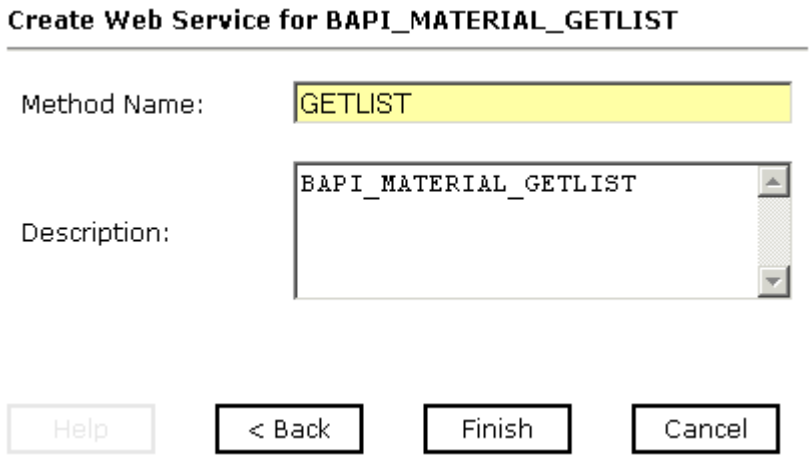
3. Select *create a new service* and click *Next*.

The following window opens in the right pane.



4. Specify the following information that is specific to the Integration Business Service you are defining:
   - **Service Name.** Enter a descriptive name for the Integration Business Service.
   - **Description.** Enter a brief description for the Integration 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 Integration 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 Integration 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 Integration Business Service Input XML* on page 4-16.

9. Click *Invoke*.

The result displays in the right pane.

Generating WSDL From a Web Service

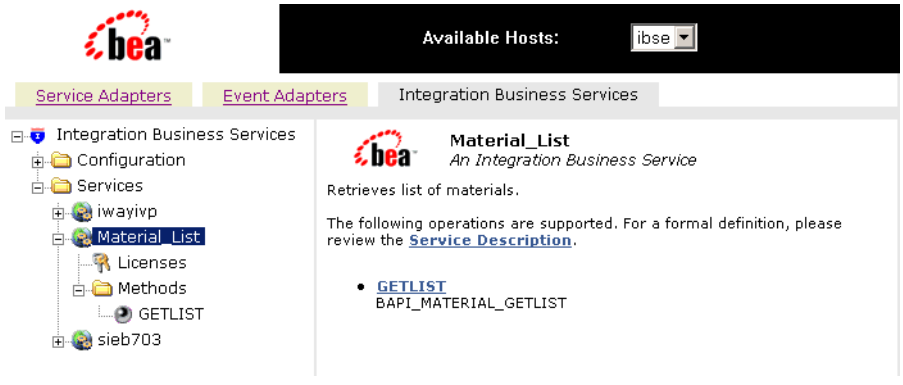
Generating WSDL (Web Services Description Language) from a Web service enables you to make the Web service available to other services within a host server such as the BEA WebLogic Server.

Procedure How to Generate WSDL From a Web Service

To generate WSDL from a Web service:

1. Click the *Integration Business Services* tab.
2. In the left pane, expand the list of services.

   This enables you to view the service for which you want to generate WSDL.



3. Select the service, for example, Material_List.

   The link for the service appears in the right pane.
4. Right-click the *Service Description* link and select *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 Material_List for an SAP R/3 creates a file named Material_List.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="m11: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 Integration 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 Event Adapters

Topics:

- Understanding Event Functionality
- Adding, Modifying, or Deleting a Port
- Adding, Modifying, or Deleting a Channel

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

The AE supports access to several Enterprise Information Systems. Even though the underlying technology used to access them varies significantly, the AE user interface varies only slightly to accomodate EIS differences. In this section, AE functionality is presented using SAP R/3 as an example. Nonetheless, you can use the AE 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 AE with additional systems, see the iWay Adapter documentation for that particular EIS.

Understanding 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.

Applications or functions within SAP may broadcast processing information at predefined points, called events. You must configure an event listener if you are to receive events from SAP. For example, the SAP Business Object "Material" may raise the event status "Material.Assigned" when a material has been created. If you wish to consume this event, you must configure an event listener to capture this event within SAP and transmit the event notification to your system.

To create an event, you must create a port and a channel using Application Explorer.

- **Port**

  A port associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Adding, Modifying, or Deleting a Port on page 5-2*.

- **Channel**

  A channel represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. For more information, see *Adding, Modifying, or Deleting a Channel* on page 5-16.

Adding, Modifying, or Deleting a Port

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

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

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

- MAIL

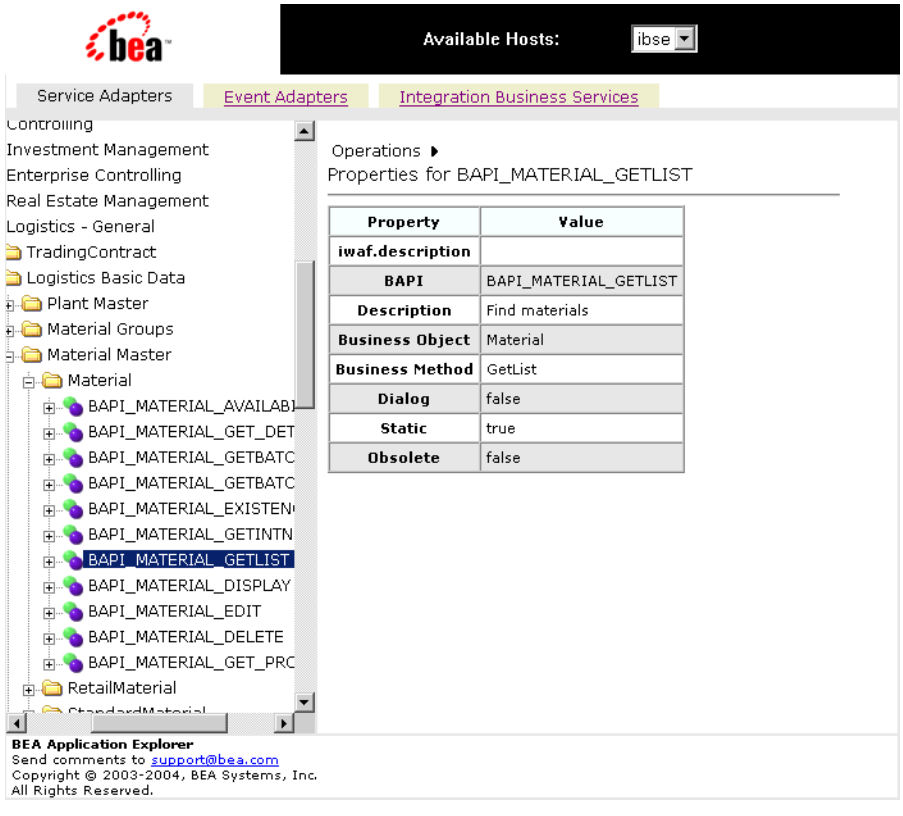
Note: The MAIL disposition option will be supported in a future release.

The following dispositions are available when using Application Explorer in conjunction with a JCA connector implementation.

- File
- HTTP
- JMS queue
- MQ Series

Procedure How to Create a Port for the File Disposition

To create a port for the File disposition using Application Explorer:



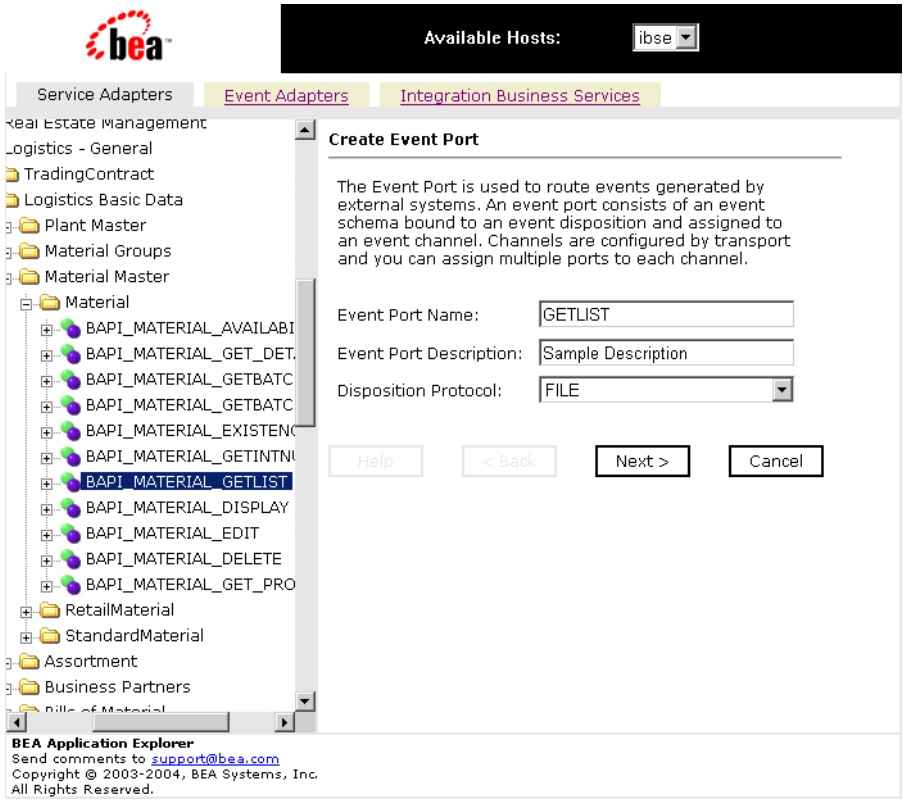
1. Click the *Service Adapters* tab.

2. Select the *BAPI_MATERIAL_GETLIST* method from the Business Object Repository.

   The panel on the right shows a table of properties and values for the method.

3. In the right pane, move the pointer over *Operations* and select *Create Event Port*.

   The following graphic shows the Create Event Port pane that opens on the right, with fields to enter an event port name, an event port description, and a disposition protocol.



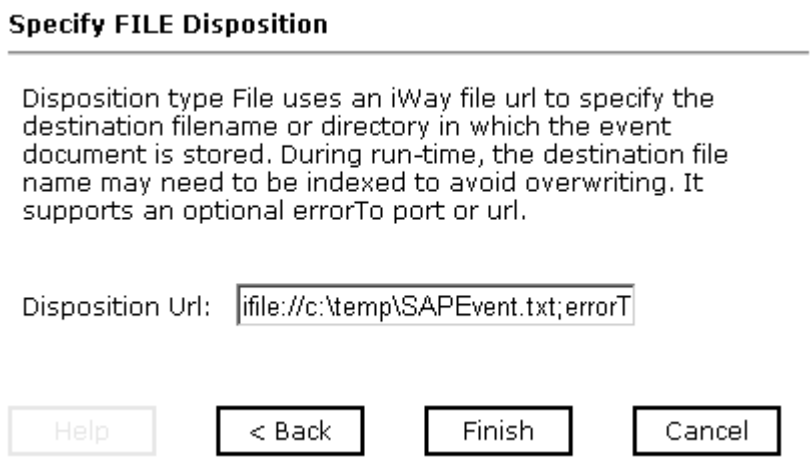
   a. Type a name in the Event Port Name field.

   b. Type a brief description in the Event Port Description field.

   c. Select *FILE* from the Disposition Protocol drop-down list.

4. Click *Next*.

   The following graphic shows the Specify FILE Disposition pane that opens on the right.



5. In the Disposition Url field, type a destination where the event data is written.

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

`ifile://`***location***`;[errorTo=`***errorDest***`]`

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

location

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

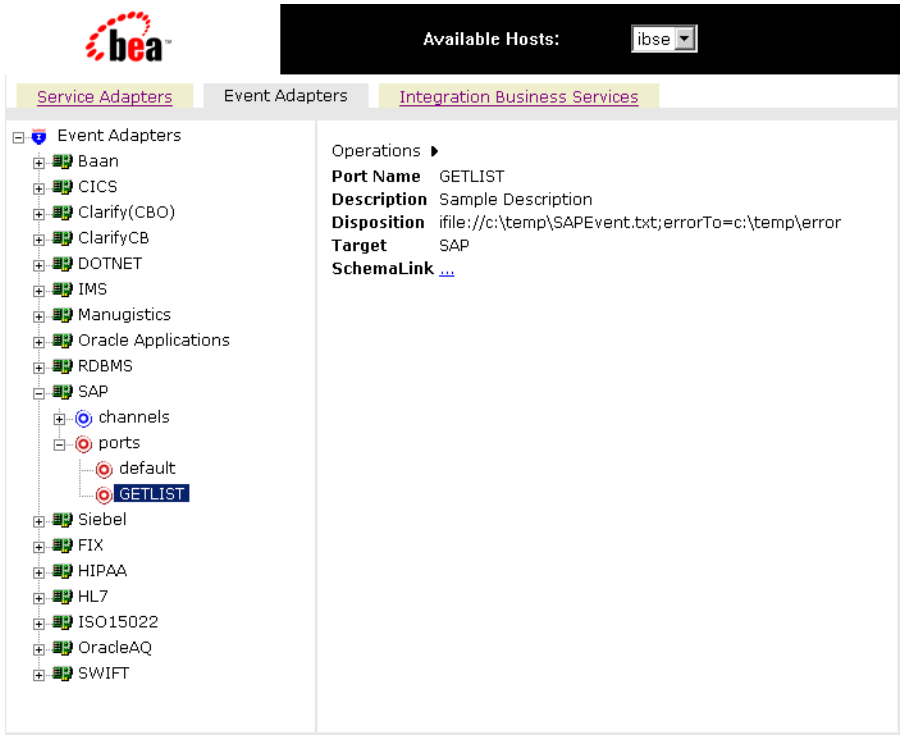
Parameter	Description
location	Full directory path and file name to which the data is written.
errorDest	Location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

For example:

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

6. Click *Finish*.

The Event Adapters tab opens. The event port appears under the ports node in the left pane. In the right pane, a table summarizes all the information associated with the port you created.



7. To view the event schema that was created for the event port, click *SchemaLink*, the only active link in the right pane.

You are now ready to associate the event port for File with a channel. For more information, see *Adding, Modifying, or Deleting a Channel* on page 5-16.

Procedure How to Create a Port for the iBSE Disposition

The iBSE disposition allows an event to launch an Integration Business Service method. To create a port for an iBSE disposition using Application Explorer:

1. Click the *Event Adapters* tab.

The Event Adapters window opens.

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

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

The Create New Port window opens in the right pane.

a. Type a name for the event port and provide a brief description.

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

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

`ibse:`*svcName*`.`*mthName*`;[responseTo=`*responseTo*`];[errorTo=`*errorDest*`]`

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

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

5. Click *OK*.

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

Procedure How to Create a Port for the MSMQ Disposition

The MSMQ disposition supports public and private queues. To create a port for an MSMQ disposition using Application Explorer:

1. Click the *Event Adapters* tab.

The Event Adapters window opens.

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

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

The Create New Port window opens in the right pane.

a. Type a name for the event port and provide a brief description.

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

c. In the Disposition field, enter an MSMQ destination in the format:

```
msmq://host/private$/qName;[errorTo=errorDest]
```

The following table lists and defines the dispostion parameters for MSMQ.

Parameter	Description
host	The name of the host on which the Microsoft Queuing system runs.
queueType	The type of queue. For private queues, enter Private$. Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
qName	The name of the queue in which messages are placed.
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

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

Procedure How to Create a Port for the JMS Queue Disposition

The JMS queue disposition allows an event to be enqueued to a JMS queue. To create a port for a JMS queue disposition using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port window opens in the right pane.

   a. Type a name for the event port and provide a brief description.

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

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

When pointing Application Explorer to an **iBSE** deployment, use the following format:

```
jmsq:myQueueName@myQueueFac;jndiurl=[myurl];jndifactory=[myfactory
];user=[user];password=[xxx];errorTo=[pre-defined port name or
another disposition url]
```

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

```
jms:jmsqueue@jmsfactory;jndiurl=;jndifactory=;
```

The following table defines the parameters for the disposition.

Parameter	**Description**
queue	JNDI name of a queue to which events are emitted.
Connection Factory	A resource that contains information about the JMS Server. The WebLogic connection factory is: `javax.jms.QueueConnectionFactory`
jndiurl	The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property, `java.naming.provider.url` For BEA WebLogic Server this is `t3://`***host:port*** where: ***host*** Is the machine name where WebLogic Server is installed. ***port*** Is the port on which WebLogic server is listening. The default port if not changed at installation is 7001.
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`
user	A valid user name required to access a JMS server.
password	A valid password required to access a JMS server.

Parameter	Description
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

5. Click *OK*.

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

Procedure How to Create a Port for the SOAP Disposition

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

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

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

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port window opens in the right pane.

   a. Type a name for the event port and provide a brief description.

   b. From the Disposition Protocol drop-down list, select *SOAP*.

   c. In the Disposition field, enter an SOAP destination, using the following format:

```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table defines the parameters for the disposition.

Parameter	Description

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

soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.
method	The Web service method you are using. This value can be found in the WSDL file.
namespace	The XML namespace you are using. This value can be found in the WSDL file.
responseTo	The location to which responses are posted, which can be a predefined port name or another URL. Optional. The URL must be complete, including the protocol.
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

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

Procedure How to Create a Port for the HTTP Disposition

The HTTP disposition uses an HTTP URL to specify a HTTP end point to which the event document is posted. To create a port for an HTTP disposition using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

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

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port window opens in the right pane.

   a. Type a name for the event port and provide a brief description.

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

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

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

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

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

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

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

Parameter	Description
url	The URL target for the post operation.
respDest	The location to which responses are posted. A predefined port name or another full URL. Optional. A predefined port name or another another disposition URL. The URL must be complete, including the protocol.
host	The name of the host on which the Web server resides.
port	The port number on which the Web server is listening.
uri	The universal resource identifier that completes the URL specification.

5. Click *OK*.

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

Procedure How to Create a Port for the MQ Series Disposition

The MQ Series disposition allows an event to be enqueued to an MQ Series queue. Both queue manager and queue name may be specified. To create a port for an MQ Series disposition using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.

2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port window opens in the right pane.

   a. Type a name for the event port and provide a brief description.

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

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

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

      ```
      mqseries:/qManager/qName;host=[hostname];port=[port];channel=[chan
      nelname];errorTo=[pre-defined port name or another disposition
      url]
      ```

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

      ```
      mq:qmanager@respqueue;host=;port=;channel=
      ```

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

Parameter	Description
qManager	Is the name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	The host on which the MQ Server is located (for the MQ Client only).

Parameter	Description
port	The number to connect to an MQ Server queue manager (for the MQ client only).
channel	The case-sensitive name of the channel that connects with the remote MQ Server queue manager (for the MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. This can be a predefined port name or another full URL. Optional.

5. Click *OK*.

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

Editing an Event Port

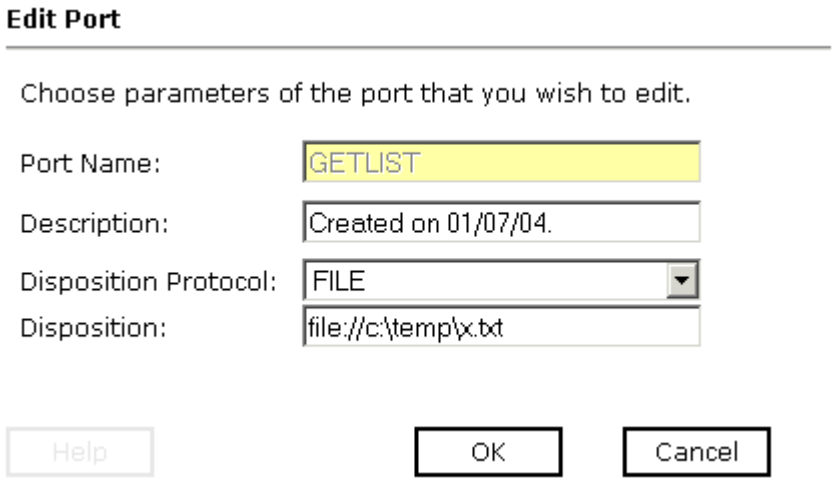
You can edit an existing event port.

Procedure How to Edit an Event Port

To edit an event port:

1. Select the event port you want to edit.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

The following graphic shows the Edit Port pane that opens on the right, with fields to enter a port name, a description, a disposition protocol, and a disposition of the port.



3. Make any required changes to the event port configuration fields and click *OK*.

Deleting an Event Port

You can delete an existing event port.

Procedure How to Delete an Event Port

To delete an event port:

1. Select the event port you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete* from the dialog box that appears.

   The following confirmation dialog box opens, asking whether to delete the event port.

3. To delete the event port you selected, click *OK*.

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

Adding, Modifying, or Deleting a Channel

The following topics describe how to create, modify, or remove a channel for your event adapter.

Creating a Channel

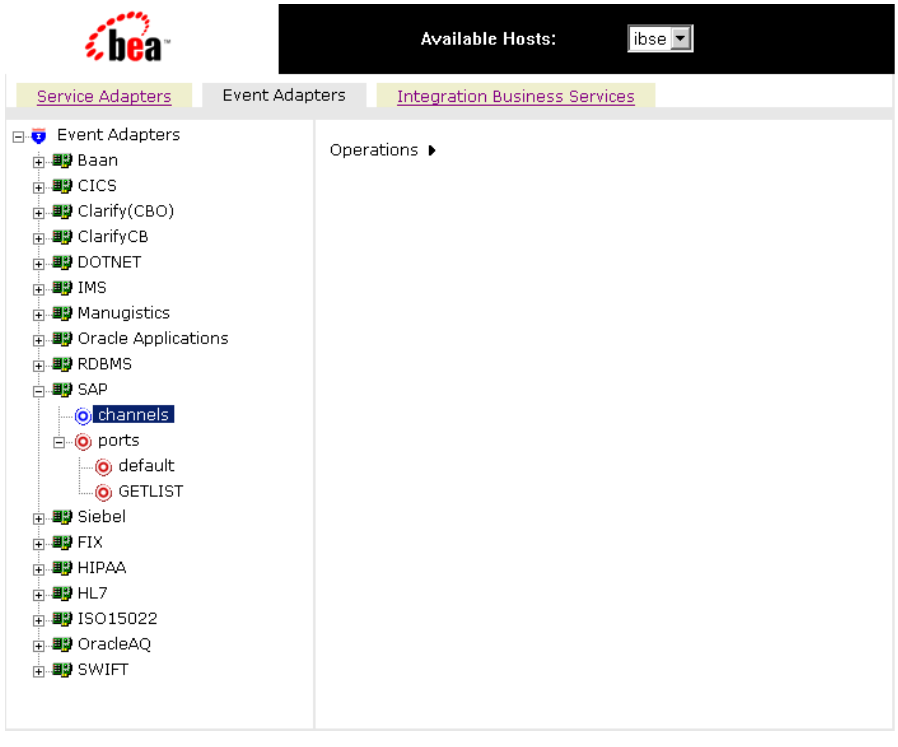
All defined event ports must be associated with a channel. You can create a channel using Servlet Application Explorer. The following procedure also describes how to start or stop a channel.

Procedure How to Create a Channel

To create a channel using Application Explorer:

1. Click the *Event Adapters* tab.

   The Event Adapters window opens.



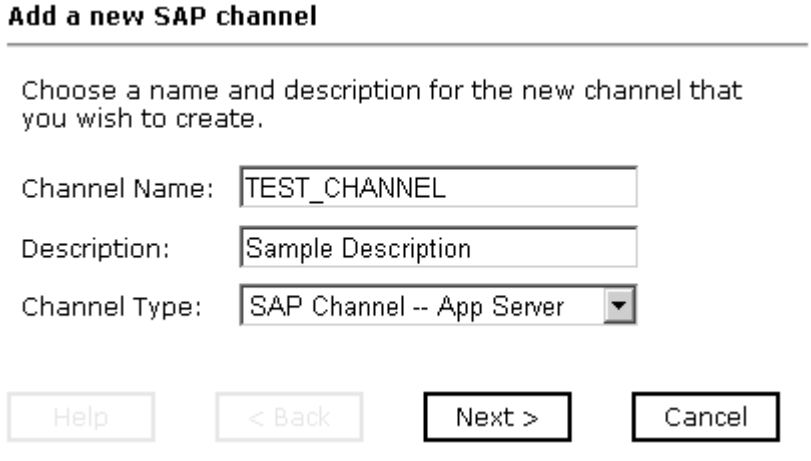
   The list of adapters that support events appears in the left pane.

2. Expand the Event Adapter node, for example, SAP.

   The ports and channels nodes appear in the left pane.

3. Click the *channels* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new channel* from the dialog box that appears.

   The following graphic shows the Add a new SAP channel pane that opens on the right, with fields to enter a channel name, a description, and a channel type.

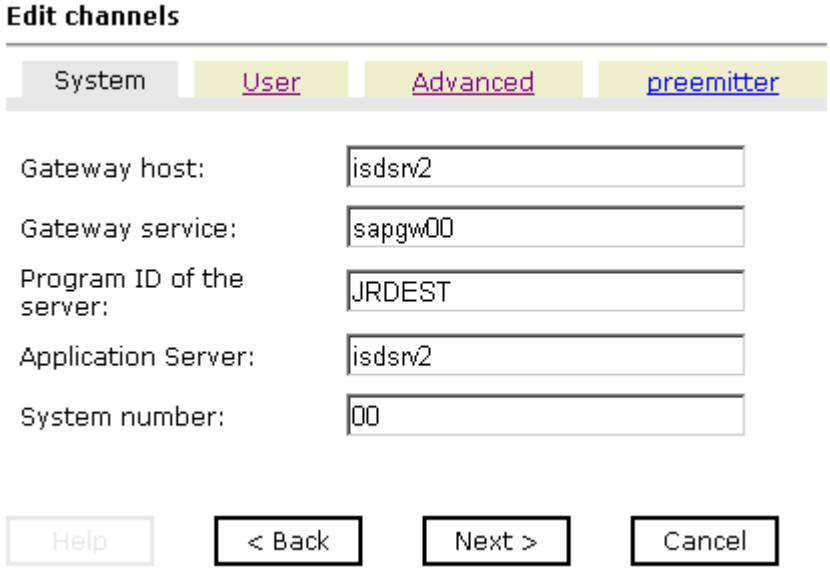


   a. In the Channel Name field, type a name, for example, TEST_CHANNEL.
   b. In the Description field, type a brief description.

c. From the Channel Type drop-down list, select *SAP Channel -- App Server*.

5. Click *Next*.

The following graphic shows the Edit channels pane that opens on the right, with fields to enter a gateway host, a gateway service, the program ID of the server, an application server, and a system number for the channel.

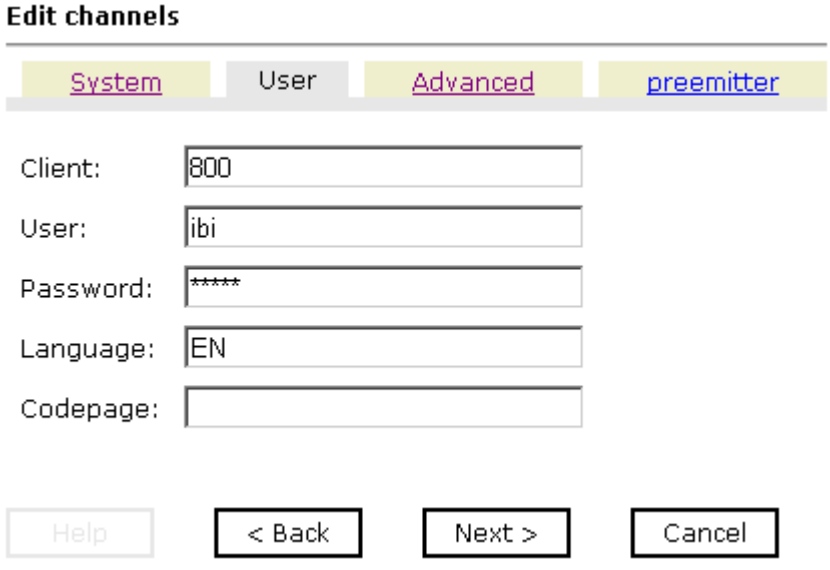


6. On the System tab, enter the information that is specific to your SAP system.

Note: The program ID of the server is case sensitive.

7. Click the *User* tab.

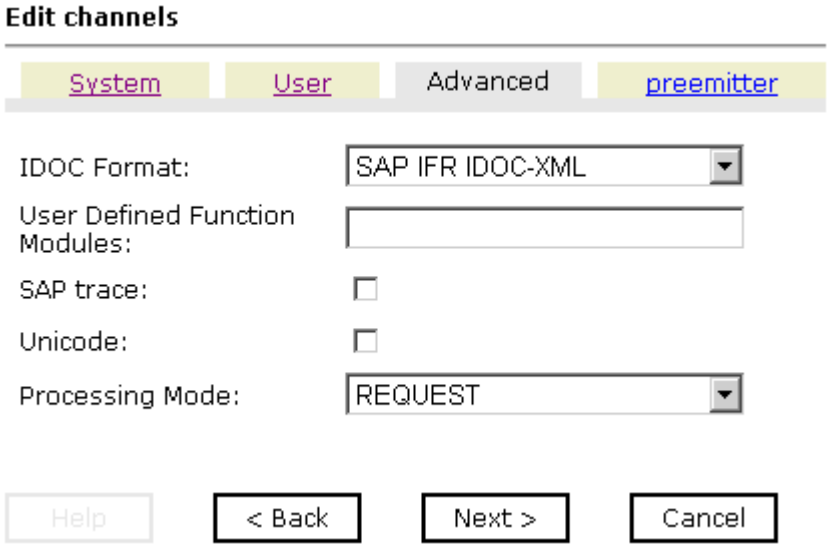
The following graphic shows the User tab that opens, with fields to enter a client, a user, a password, a language, and a code page.



8. Enter the user information that is specific to your SAP system.

9. Click the *Advanced* tab.

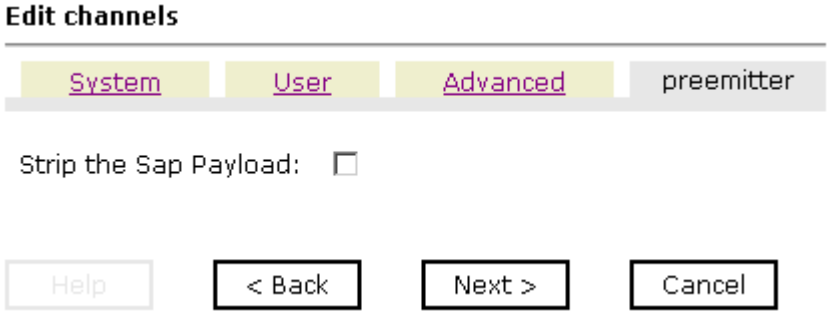
The following graphic shows the *Advanced* tab that opens, with fields to enter an IDoc format, any user defined function modules, check boxes to enable SAP traces or Unicode encoding of the event data, and a pull down menu with options for synchronous event processing.



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

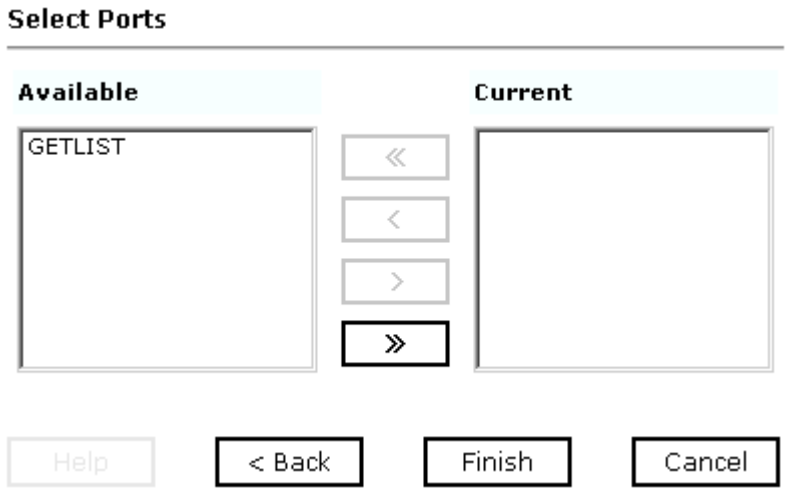
11. Click the *preemitter* tab.

The following graphic shows the *preemitter* tab that opens, with a checkbox that enables you to strip the SAP payload of an event document.



12. Click *Next*.

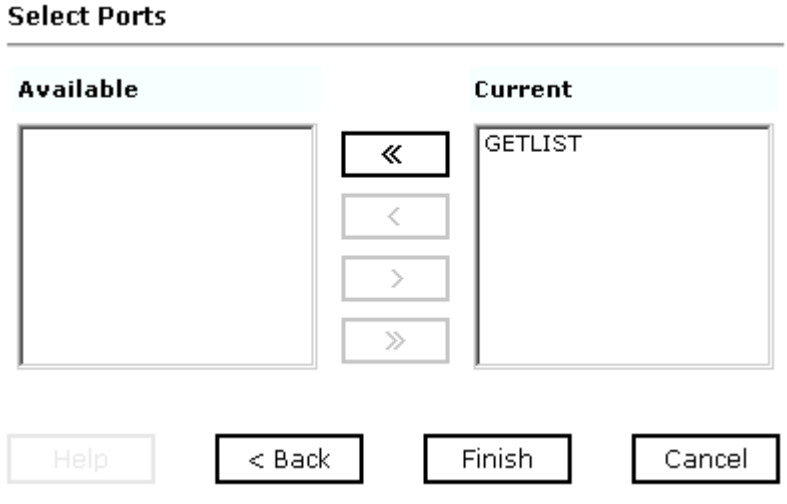
The following graphic shows the Select Ports pane that opens where you can move ports between the Current and Available list. To return to the previous screen, you can click the Back button, or to escape you can click the Cancel button.



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

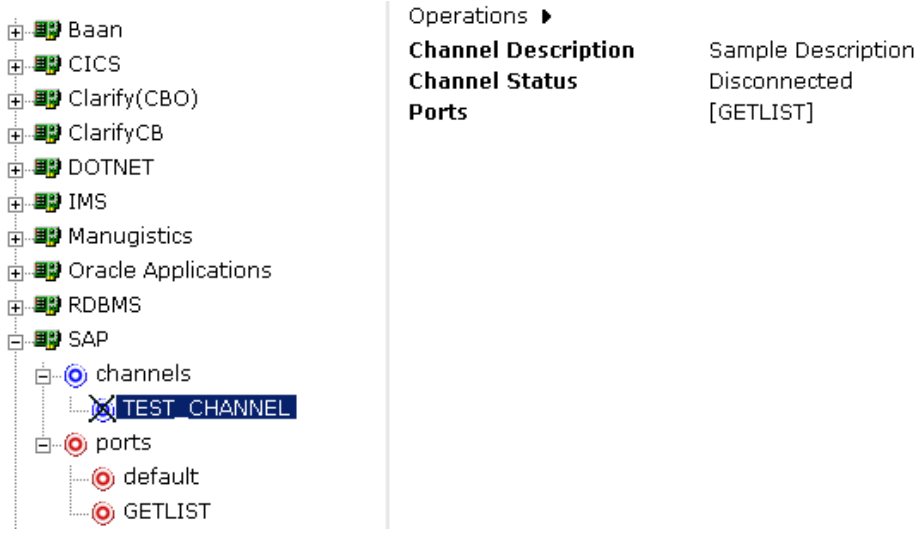
b. To transfer the port to the list of available ports, click the single right > arrow button or to associate all event ports, click the double right >> arrow button.

The port appears in the list of available ports.



13. Click *Finish*.

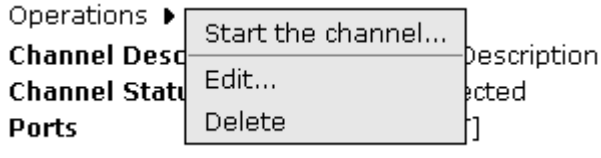
The summary pane opens on the right.



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

The channel also appears under the channels node in the left pane. An X through the icon indicates that the channel is currently disconnected.

You must start the channel to activate your event configuration.



14. In the right pane, move the pointer over *Operations* and select *Start the channel*.

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

15. To stop the channel at any time, move the pointer over *Operations* and select *Stop the channel*.

Modifying a Channel

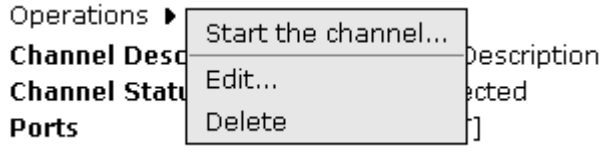
You can edit an existing channel.

Procedure How to Edit a Channel

To edit an existing channel:

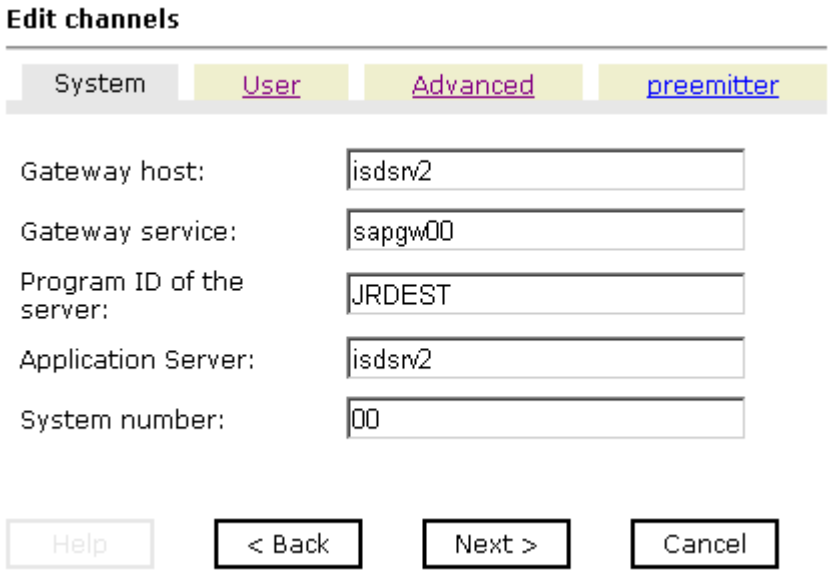
1. In the left pane, select the channel you want to edit.

   In the following graphic, the right pane shows the operations, channel description, channel status and ports of the channel.



2. In the right pane, move the pointer over *Operations* and select *Edit* from the dialog box that appears.

The following graphic shows the Edit channels pane that opens on the right, with fields to enter a gateway host, a gateway service, the program ID of the server, an application server, and the system number of the channel.



3. Make any required changes to the channel configuration fields and click *Finish*.

Deleting a Channel

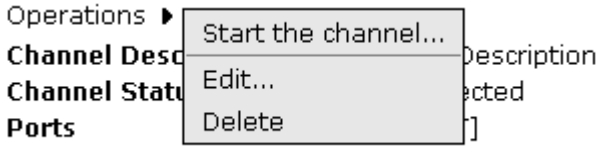
You can remove an existing channel.

Procedure How to Delete a Channel

To delete an existing channel:

1. In the left pane, select the channel you want to delete.

   The right pane displays the operations menu, the channel description, the channel status, and the ports connected to the channel.



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

   A confirmation dialog box opens asking if you want to delete this item.

3. To delete the channel you selected, click *OK*.

   The channel disappears from the list in the left pane.

CHAPTER 6

Using Integration Business Services Policy-Based Security

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

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

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

Integration Business Services Policy-Based Security

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

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

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

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

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

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

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

Configuring Integration Business Services Policy-Based Security

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Servlet Application Explorer. For more information, see *How to Create a User to Associate With a Policy* on page 6-3 or *How to Create a Group to Associate With a Policy* on page 6-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *How to Create an Execution Policy* on page 6-7.

You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to individual Integration Business Services. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 6-10.

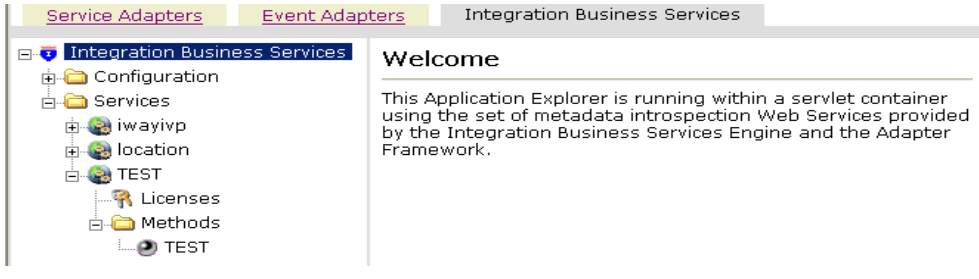
Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

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



   a. Click the *Integration Business Services* tab.

   b. Expand the *Configuration* node.

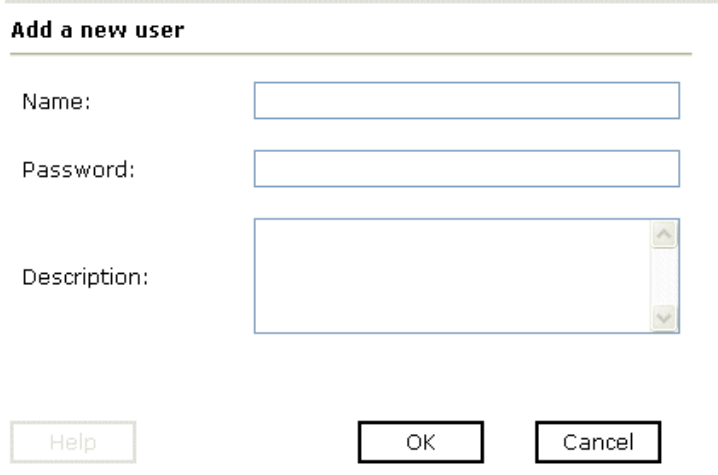
   c. Expand the *Security* node.

   d. Expand the *Users and Groups* node.

   e. Select *Users*.

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

The following image shows the Add a new user pane that opens with fields where you enter a user name, a password, and a description of the user. To escape without making changes, you can click the Cancel button.



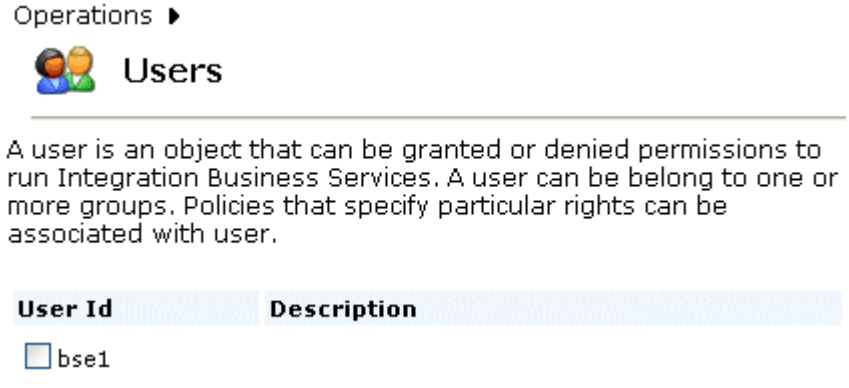
a. In the Name field, type a user ID.

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

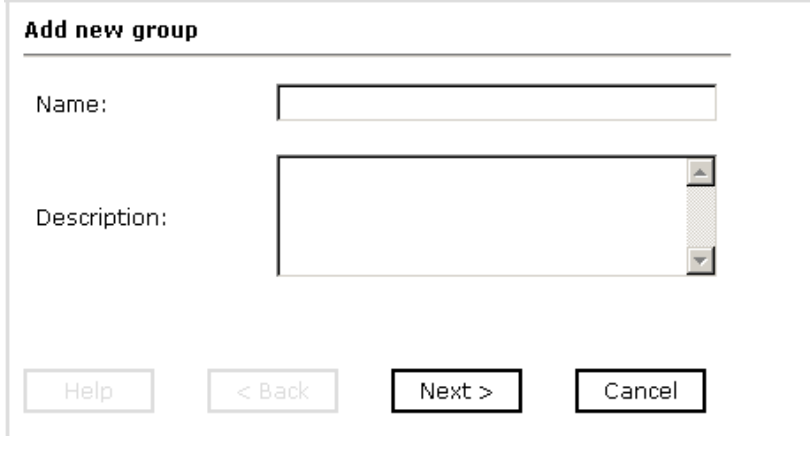


Procedure How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

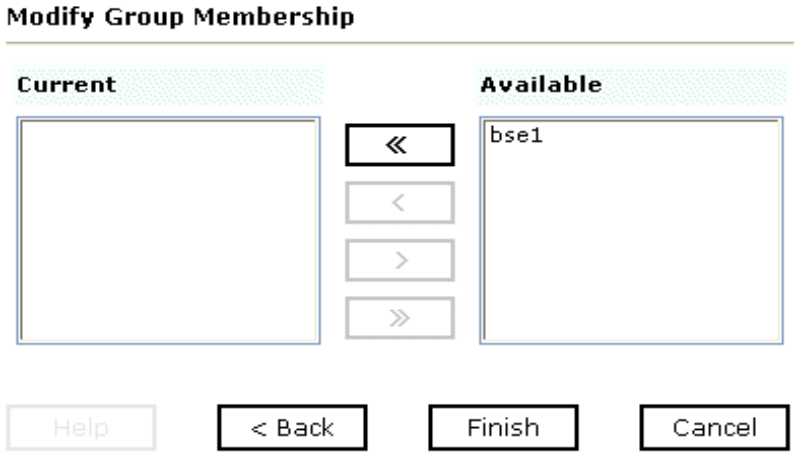
1. Open *Servlet Application Explorer*.
   a. Click the *Integration Business Services* tab.
   b. Expand the *Configuration* node.
   c. Expand the *Security* node.
   d. Expand the *Users and Groups* node.
   e. Select *Groups*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

   The following image shows the Add new group pane that opens with fields where you enter a name and a description for the group. To continue, you click the Next button or to escape without making any changes, you click the Cancel button.



   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*.

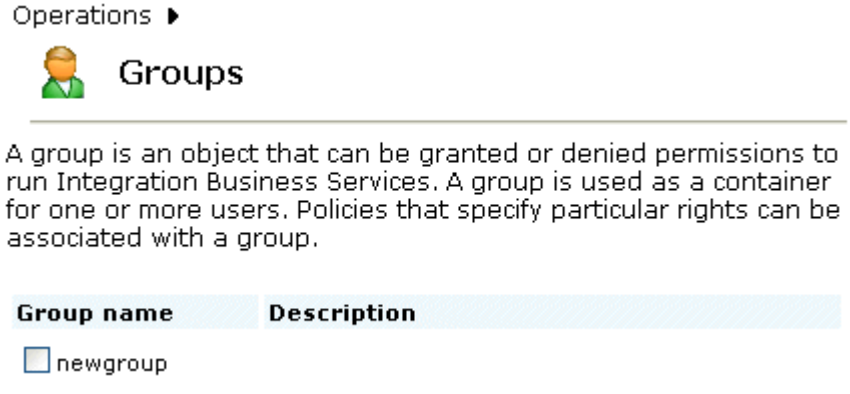
The following image shows the Modify Group Membership pane that opens where you can move users to or from a group by moving them between the Current and Available lists and then clicking the Finish button. To return to the previous screen, you can click the Back button or to escape, you can click the Cancel button.



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 new group added to the configuration. It includes a definition of a group and the group name and description.

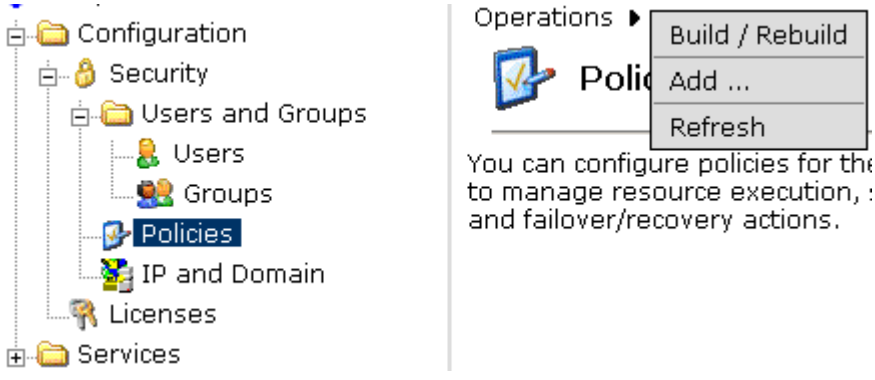


Procedure How to Create an Execution Policy

To create an execution policy:

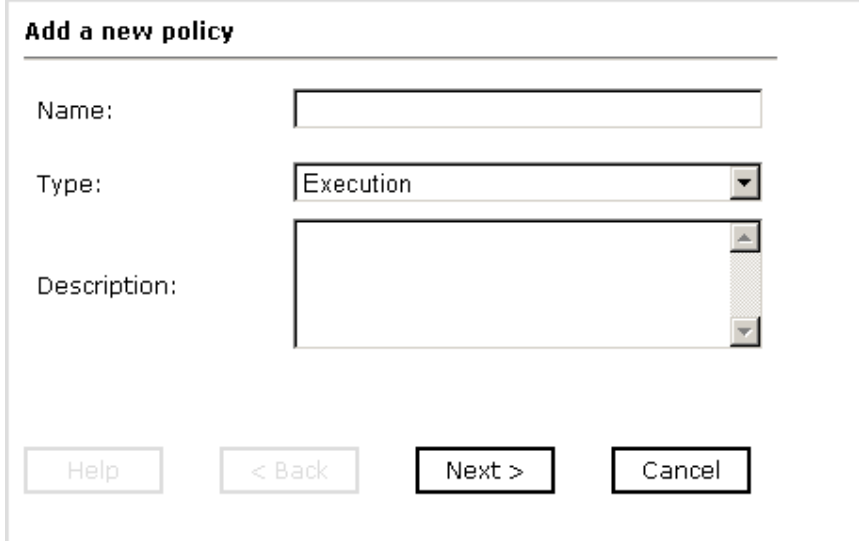
1. Open *Servlet Application Explorer*.
   a. Click the *Integration Business Services* tab.
   b. Expand the *Configuration* node.
   c. Select *Policies*.

   The following image shows the Policies pane on the right where you can apply a policy. The Operations menu becomes available with three options, including the Add option.



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

The following image shows the Add a new policy pane that opens, with fields for the name, type, and description of the policy. To continue, you click the Next button or to escape without making changes, you click the Cancel button.



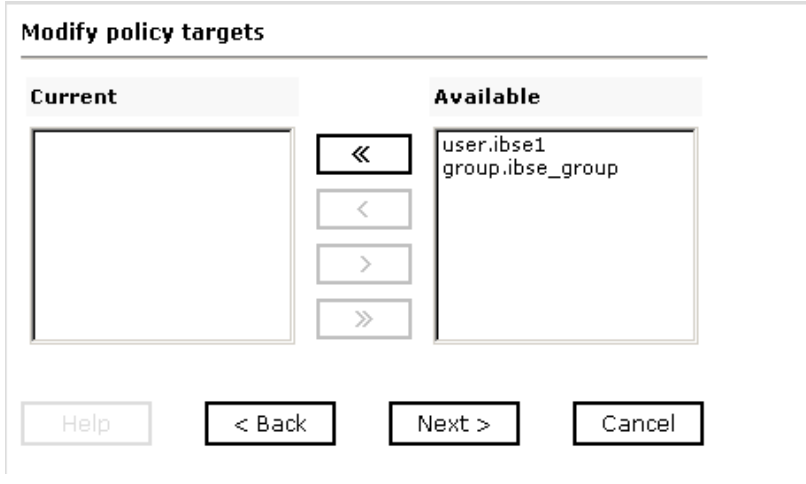
a. In the Name field, type a a name for the policy.

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 with a list of current and available targets and arrow buttons to move targets from one list to the other. The pane includes a Back button to return to the previous screen, a Next button to continue to the next screen, or a Cancel button to escape from the pane.

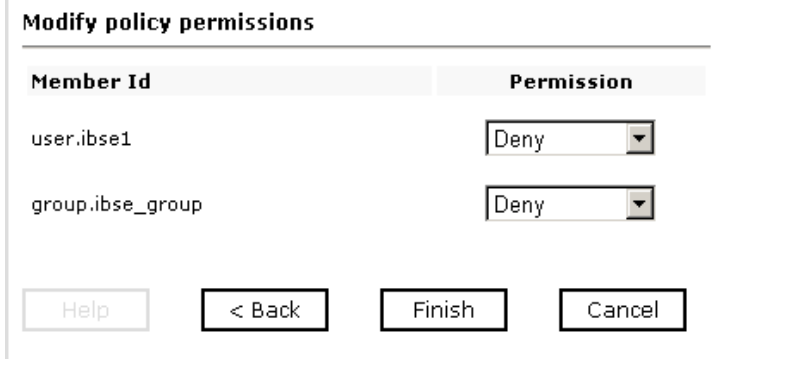


4. Select a minimum of one user or group from the Available pane.

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.

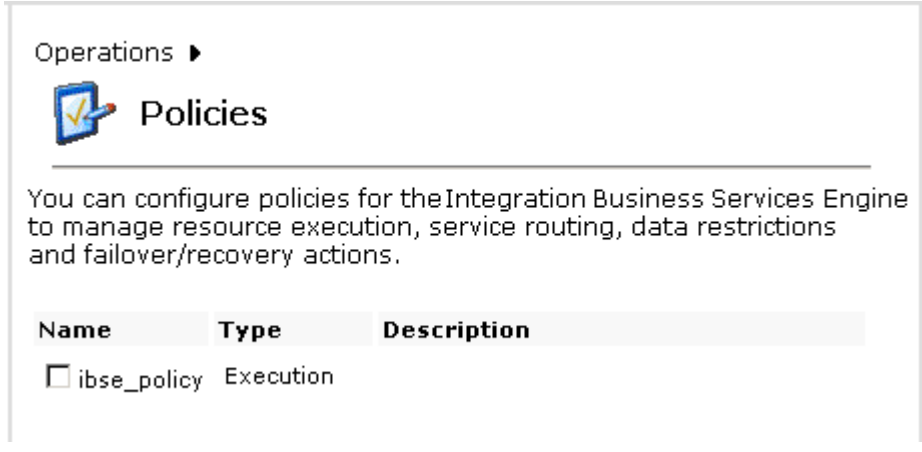
5. Click *Next*.

The following image shows the Modify policy permissions pane that opens with drop-down lists where you can select to grant or deny permission to members and then click a button to finish. Alternately, you can click the Back button to return to the previous screen or the Cancel button to escape from the screen without making changes.



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

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

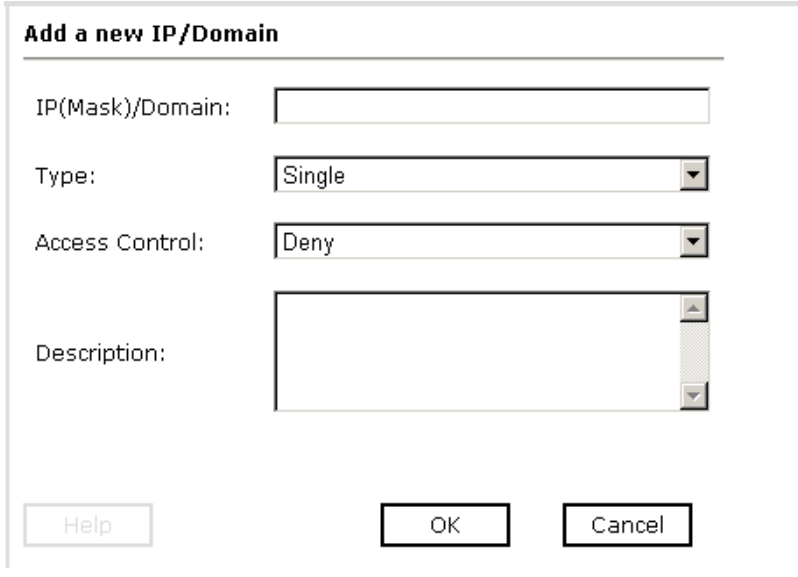


Procedure How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

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

The following image shows the Add a new IP/Domain pane that opens where you enter information for the IP/Domain in four fields. To escape, you can click the Cancel button. You must select a type of restriction from a drop-down list before you can enter information in the IP(Mask)/Domain field.



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 summarizes your configuration including the domain name, whether access is granted or denied, and a description (optional).


Operations
IP and Domain
You can configure the Integration Business Services Engine to use policies that control access from a single IP address, a group of IP addresses, or all addresses within a particular domain.
IP(Mask) / Domain
Access
Description
test
Deny

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

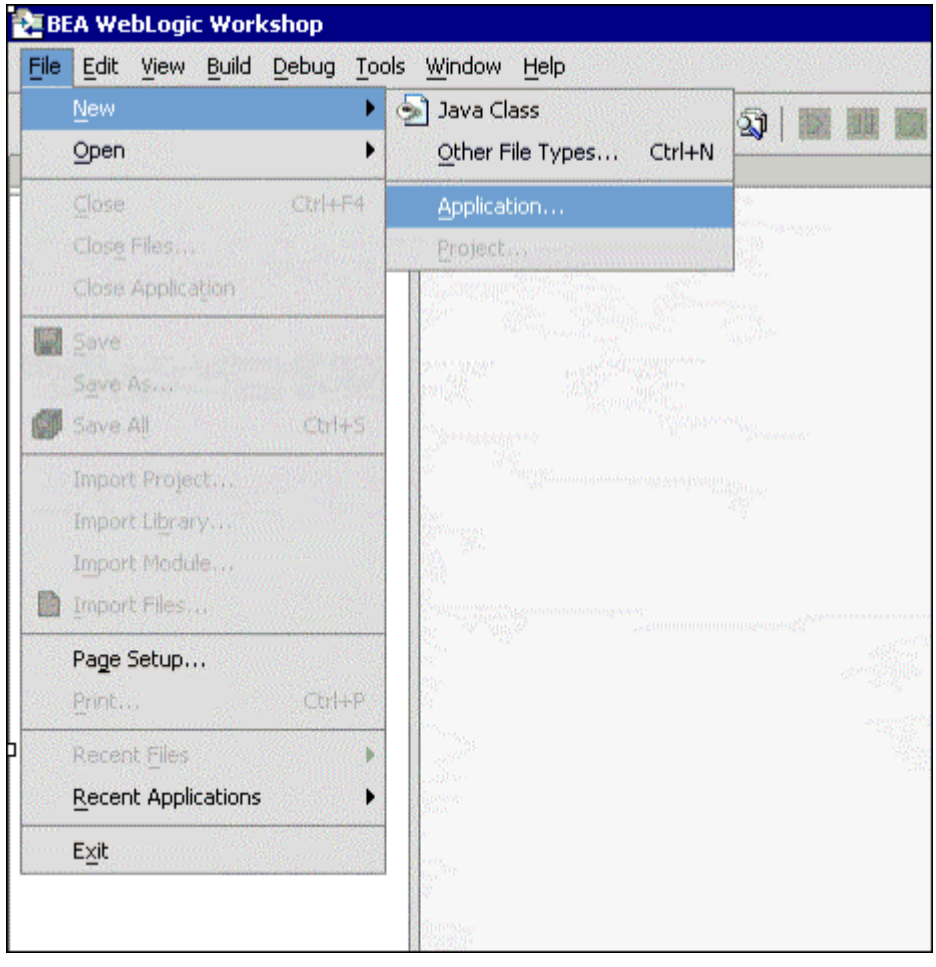
WebLogic Workshop provides a framework for building Web services. The Web services that you build with WebLogic Workshop are enterprise-class services, and WebLogic Workshop provides simple controls for connecting to your enterprise resources. At the same time, WebLogic Workshop simplifies the process 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 procedures 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 Integration Business Services*.

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

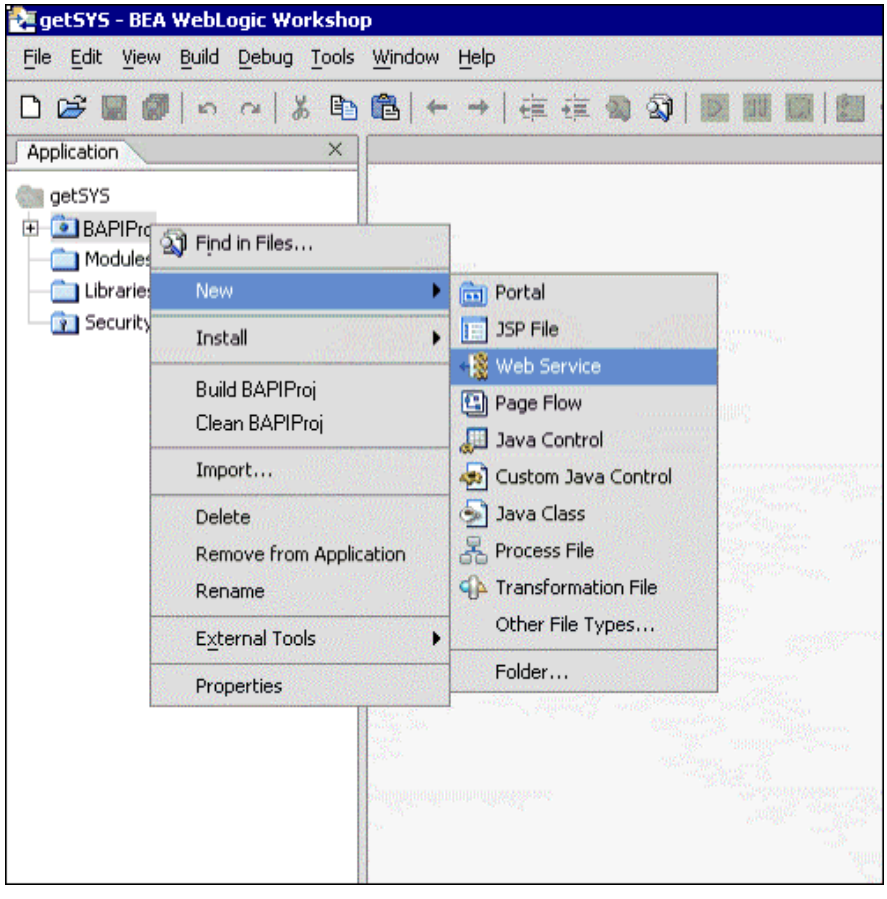
BEA WebLogic Workshop opens.



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 *BAPIProj* 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*.

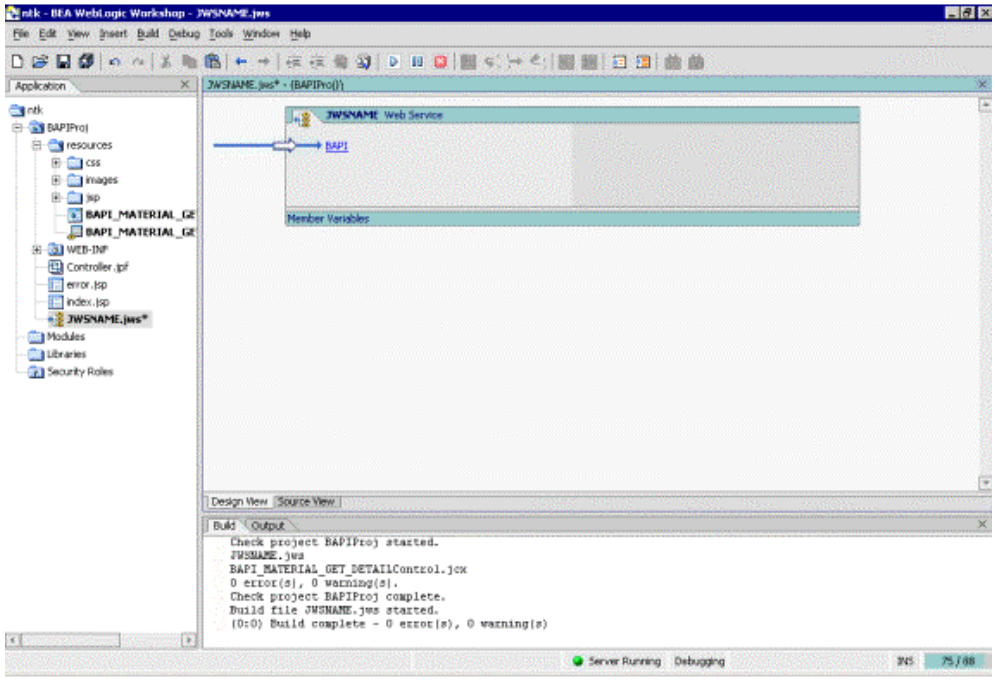
The design view window opens.

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*.

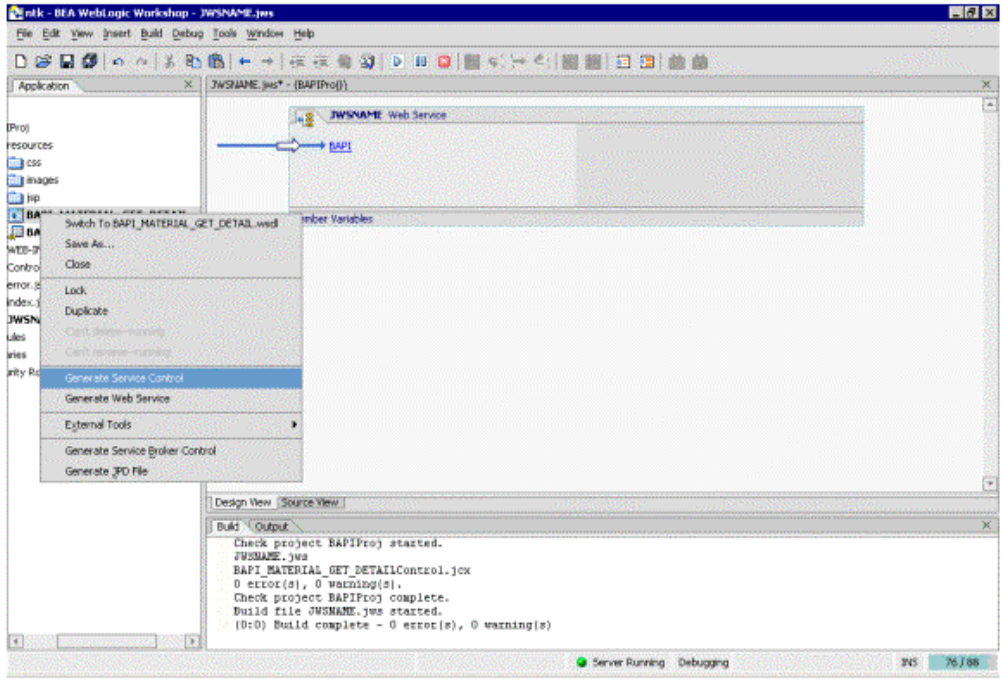
b. In the space provided, replace *method1* with *BAPI*, and press *Enter*.



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

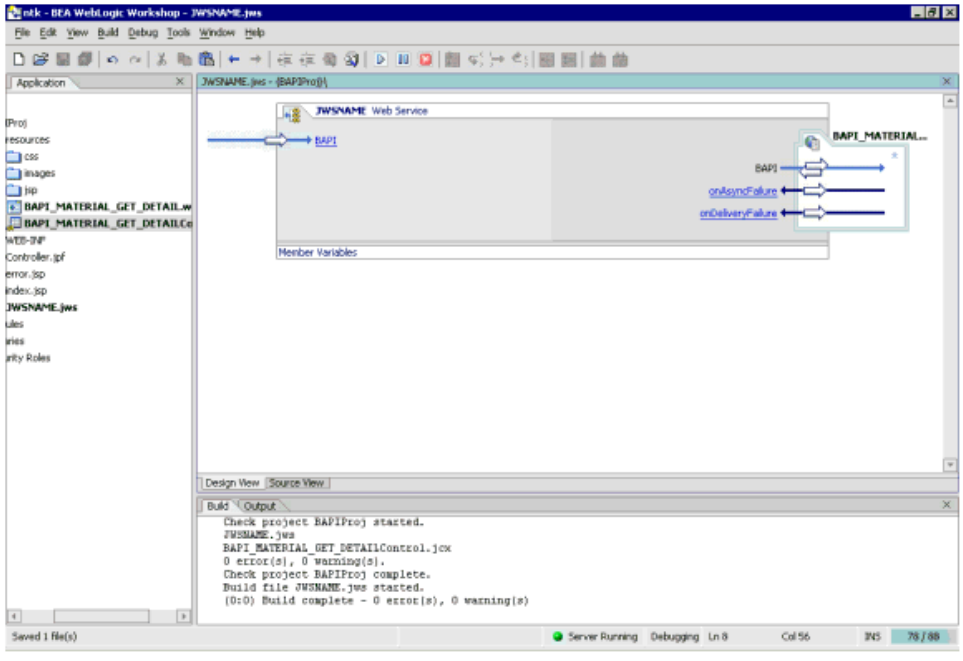
10. Import the BAPI_MATERIAL_GET_DETAIL.*WSDL* .

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



11. To generate a Java Control file, right-click the BAPI_MATERIAL_GET_DETAIL.*wsdl* 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);
  }
}
```

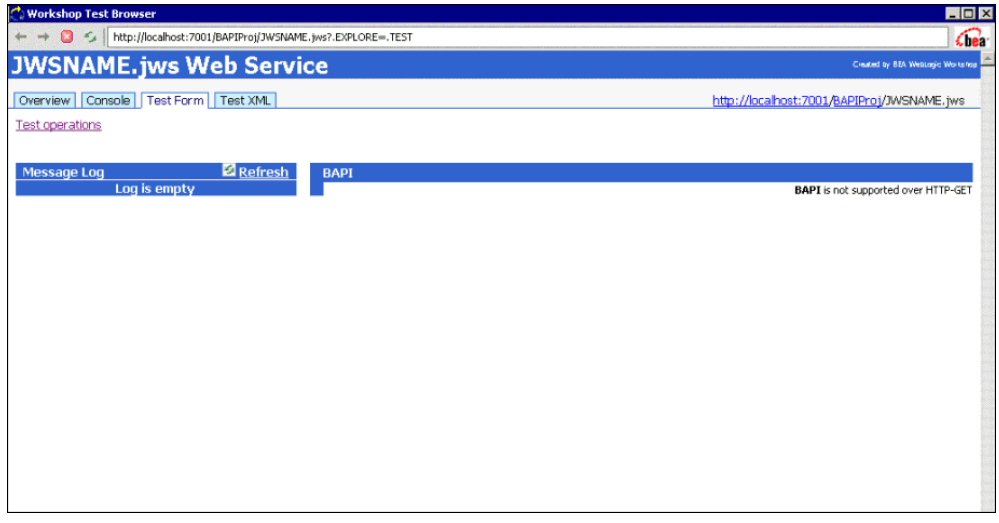
Running the JWSNAME Web Service from WebLogic Workshop for BAPIs

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.

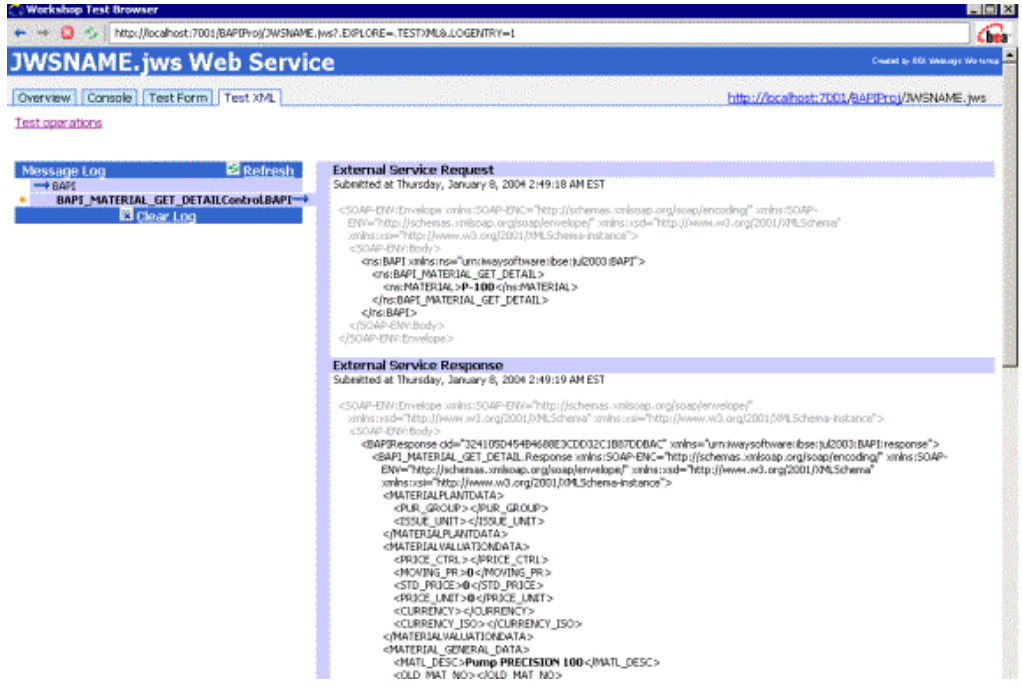


4. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
5. Replace the string XML input with the following:

```
<?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 Integration Business Services Engine (iBSE), the following response is returned:



The previous sample is a very simple example of calling a 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 puposes.

```
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

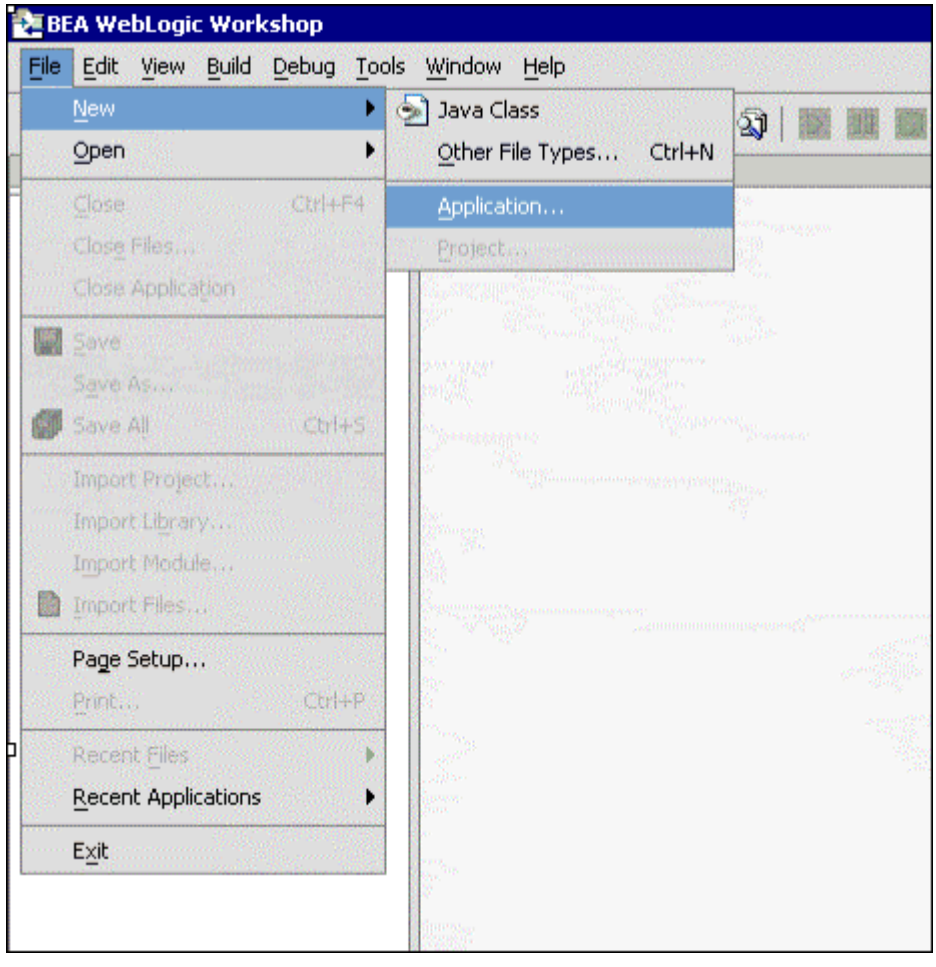
WebLogic Workshop provides a framework for building Web services. The Web services that you build with WebLogic Workshop are enterprise-class services, and WebLogic Workshop provides simple controls for connecting to your enterprise resources. At the same time, WebLogic Workshop simplifies the process 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 procedures 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 Integration Business Services*.

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

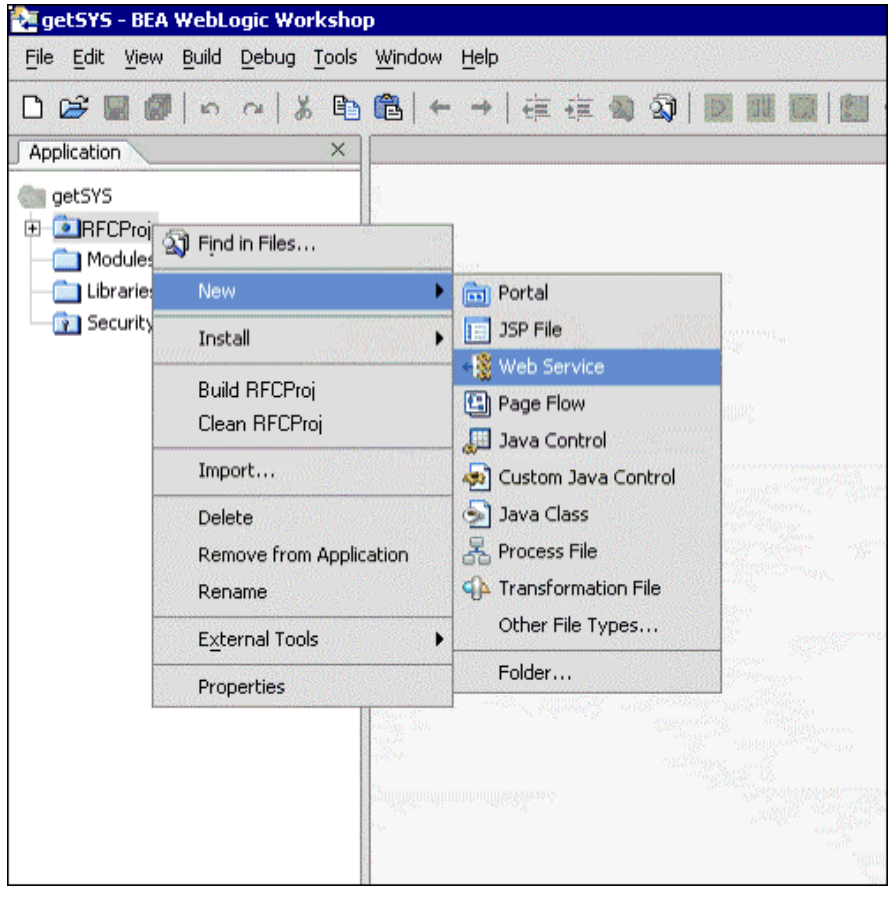
BEA WebLogic Workshop opens.



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*.

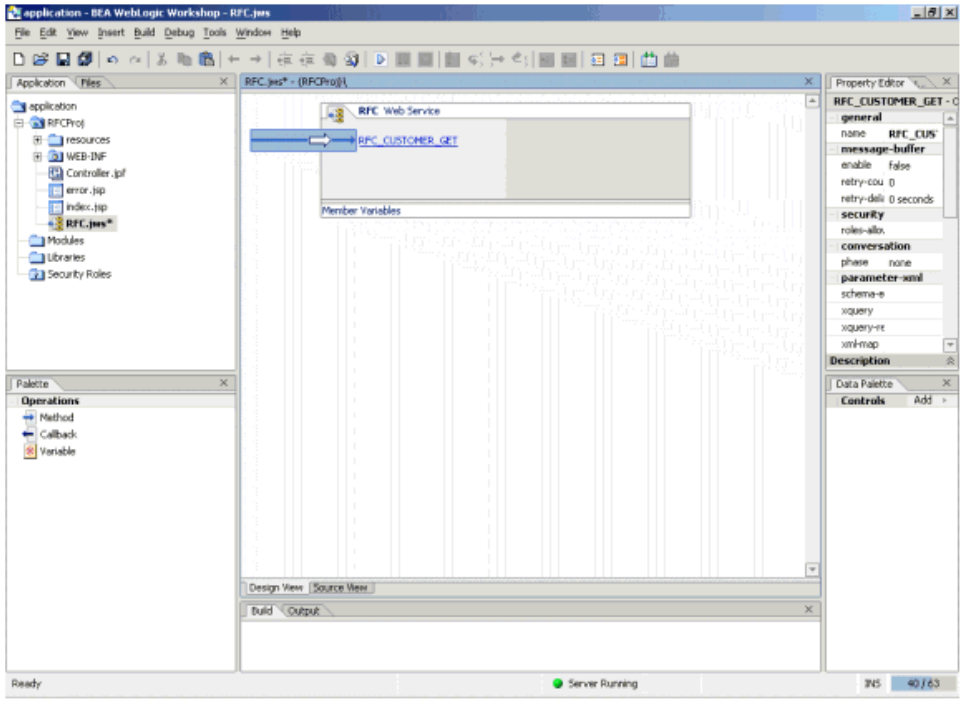
The design view window opens.

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*.

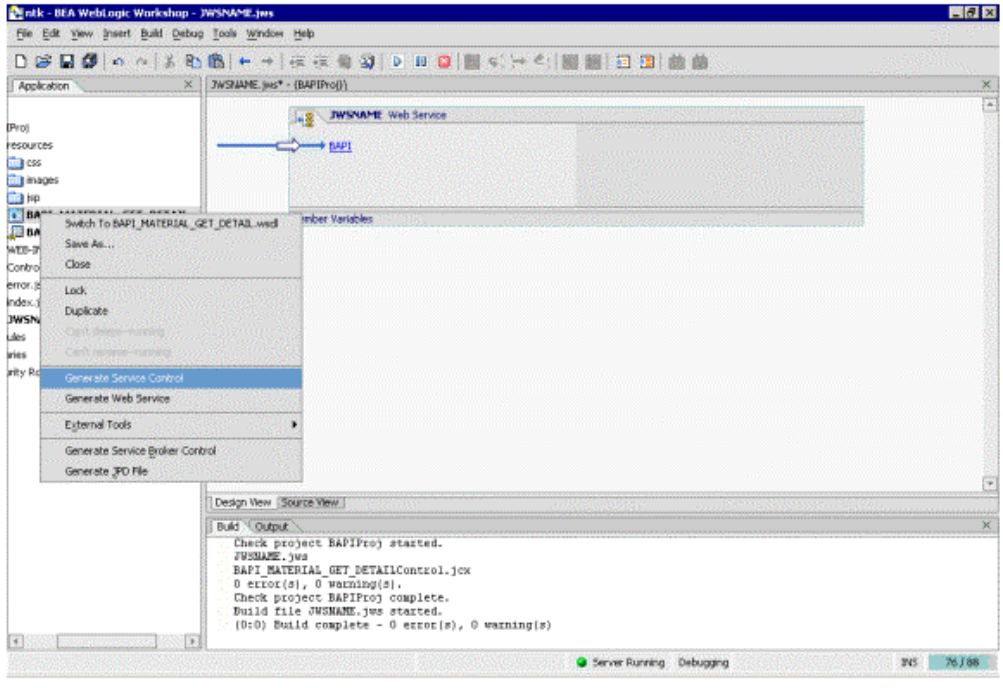
b. In the space provided, replace *method1* with *RFC_CUSTOMER_GET*, and press *Enter*.



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

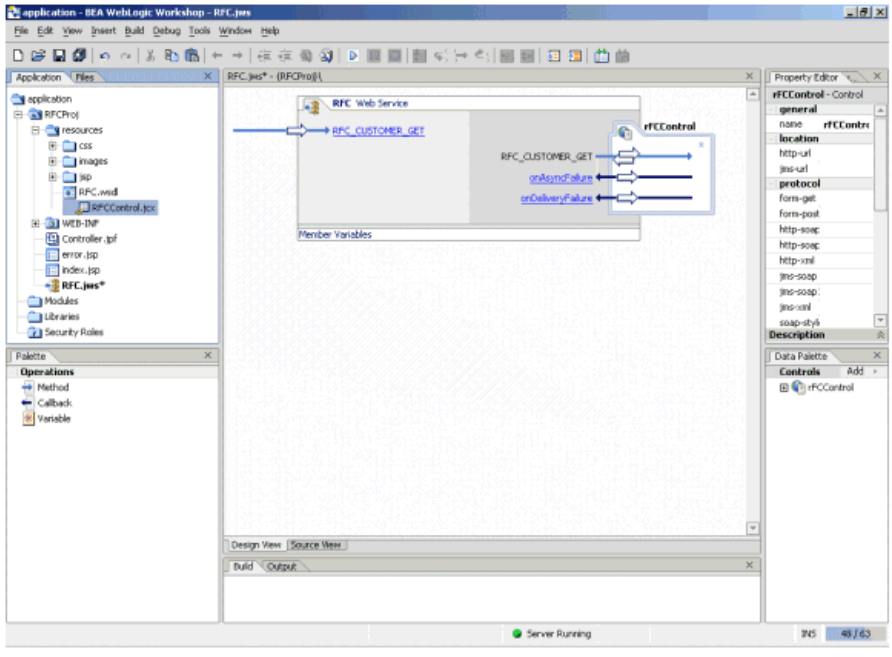
10. Import the *RFC.WSDL*.

For more information on creating a WSDL file, see Chapter 4, *Creating and Publishing Integration 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 *RFC.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(RFCControl.RFC_CUSTOMER_GET input)
 {
   RFCControl.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);
  }
}
```

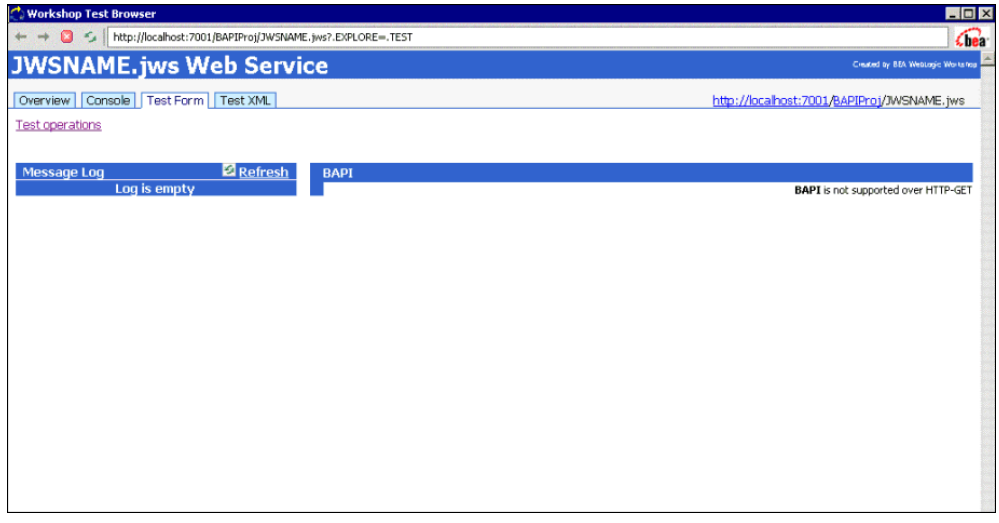
Running the JWSNAME Web Service from WebLogic Workshop for RFCs

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.

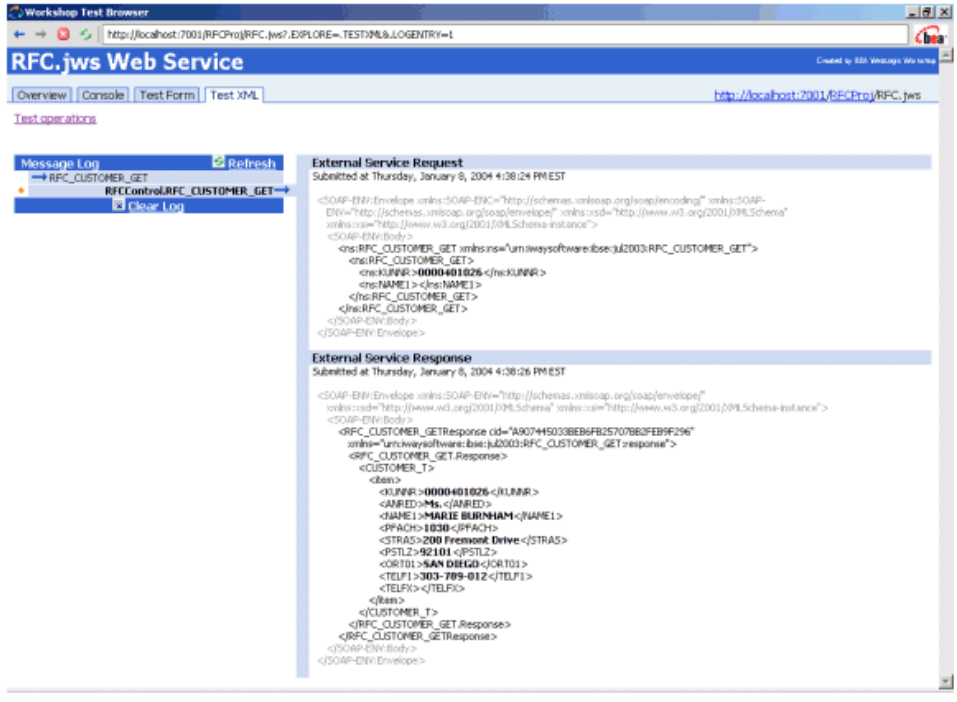


4. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
5. Replace the string XML input with the following:

```
<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 Integration Business Services Engine (iBSE), the following response is returned:



The previous sample is a very simple example of calling a 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 puposes.

```
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
```

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**