IBM WebSphere® Adapter for FlatFiles 6.2.0.0

Quick Start Tutorials

Note: Before using this information and the product it supports, read the information in "Notices" on page 195 .

This edition applies to version 6, release 2, modification 0 of IBM WebSphere Adapter for Flat Files and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

Chapter 1. Introduction

Learning Objectives

After completing a tutorial, you should be able to perform the following tasks:

- Create an adapter project in WebSphere® Integration Developer.
- Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.
- Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.
- Test the module and validate the results.

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.

Software prerequisites

To use these tutorials, you must have the following applications installed:

- WebSphere Integration Developer, version 6.2.
- WebSphere Process Server, version 6.2

Chapter 2. Preparing to run through the tutorial

Configuration prerequisites

Before doing any tutorial testing, complete the following tasks:

- For inbound processing of files, configure a JDBC data source in WebSphere Process Server specifying the name of your database or choose to use the default JDBC data source available with the server.
- For inbound processing of files have event directory and archive directory (if archival of files is required) already created.
- Similarly for outbound processing of files, have output directory created already.

Extracting the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Examples to unzip a sample file say Tutorial1.zip, import it into a workspace in WebSphere Integration Developer. File -> Import -> Project interchange -> Browse to the location of Tutorial1.zip -> Select all of the contents displayed -> Finish.

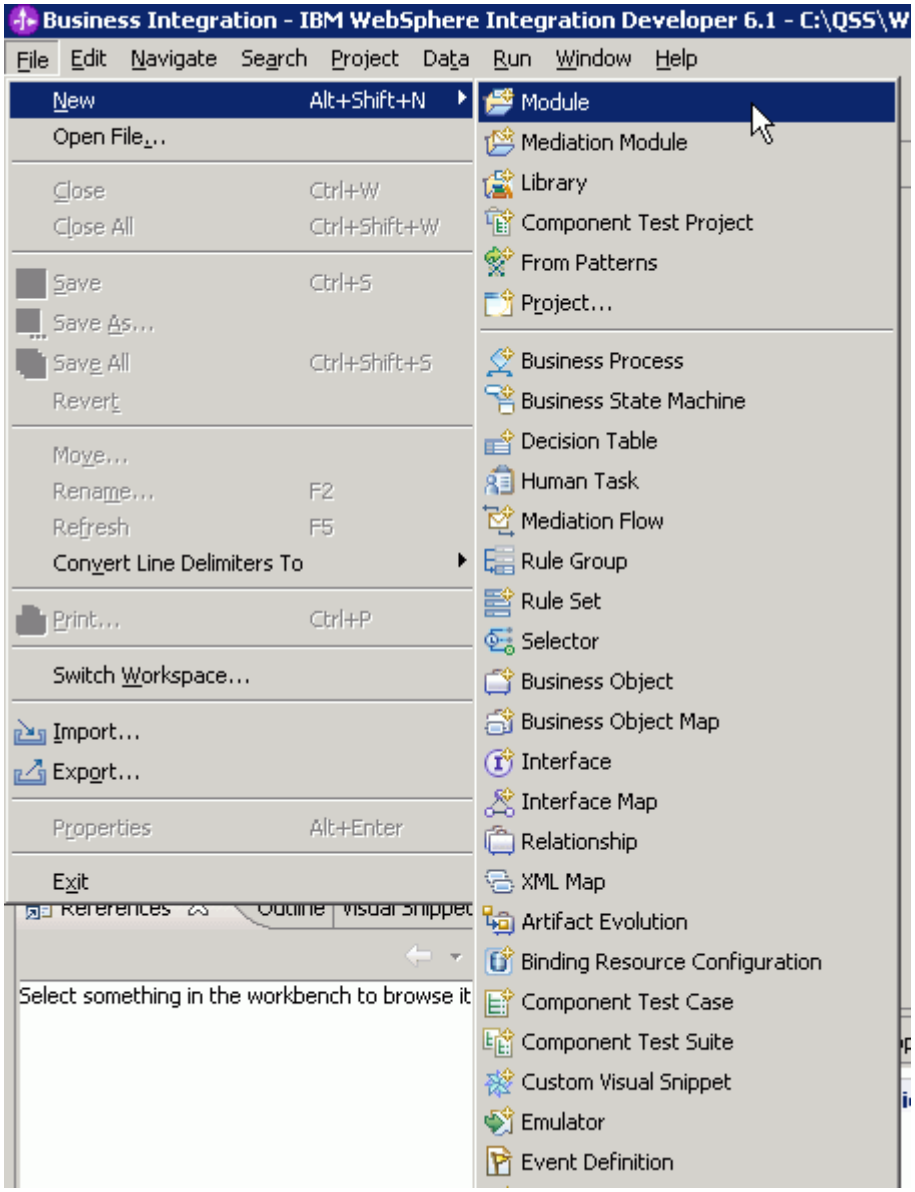
Chapter 3. Tutorial 1: Outbound processing – Retrieving content from file with data transformation

This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.2.0.0 can be used to retrieve content from a file located on local file system. In addition, it demonstrates as to how XML Data handler can be used while retrieving XML content from a file.

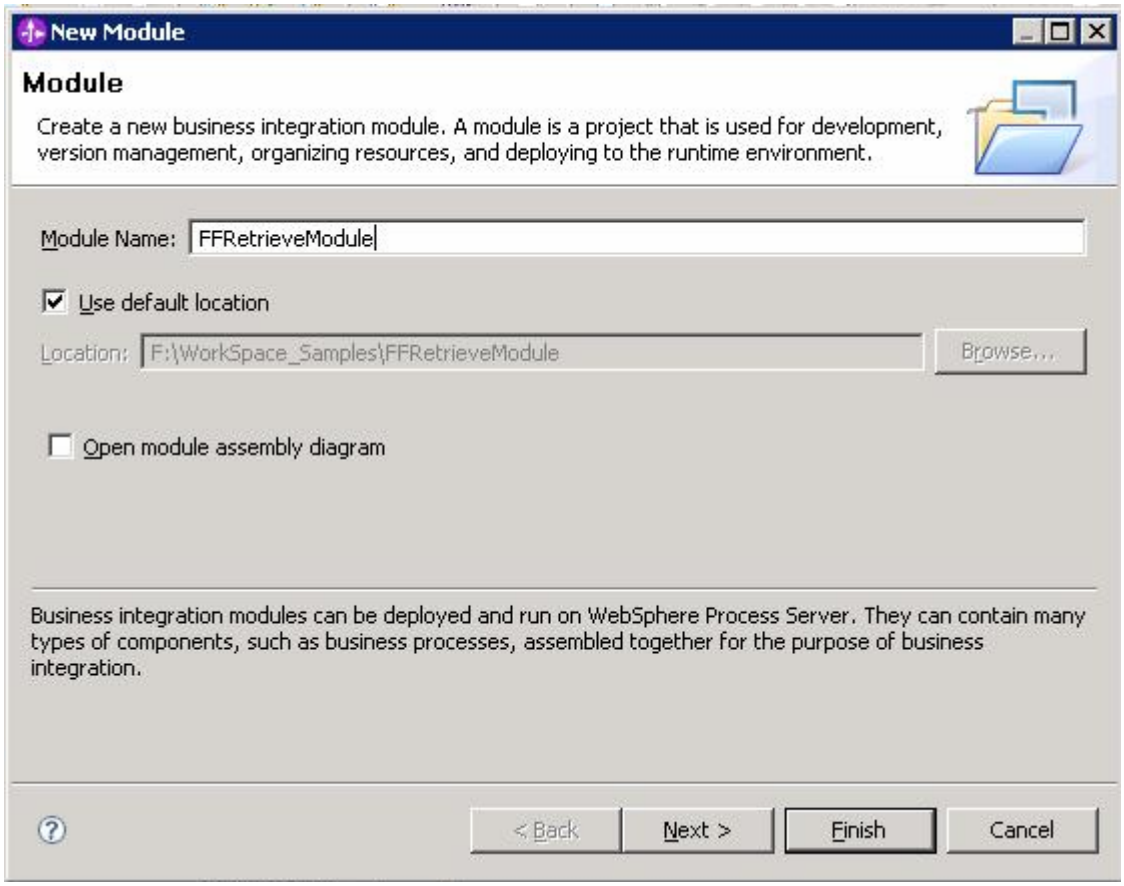
Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

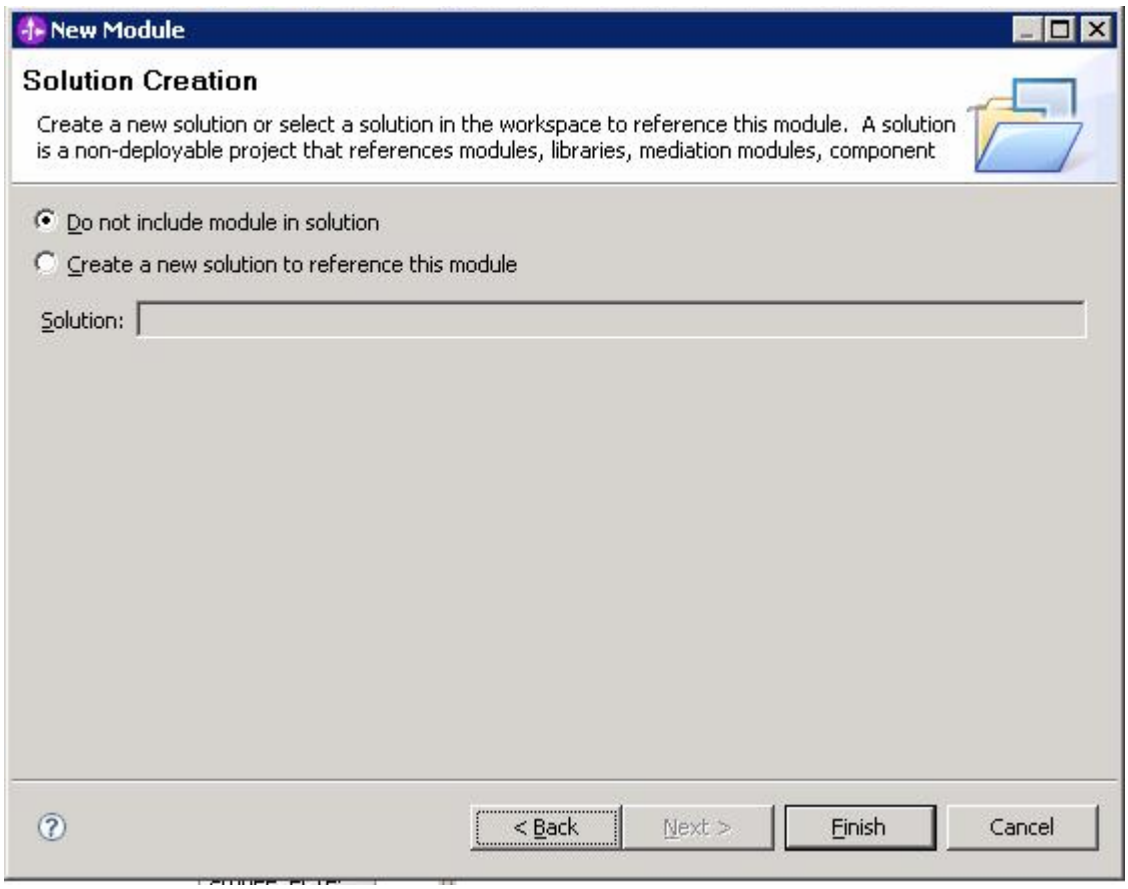
- Create a new module to generate adapter outbound artifact. Go to **File** -> **New** -> **Module.**



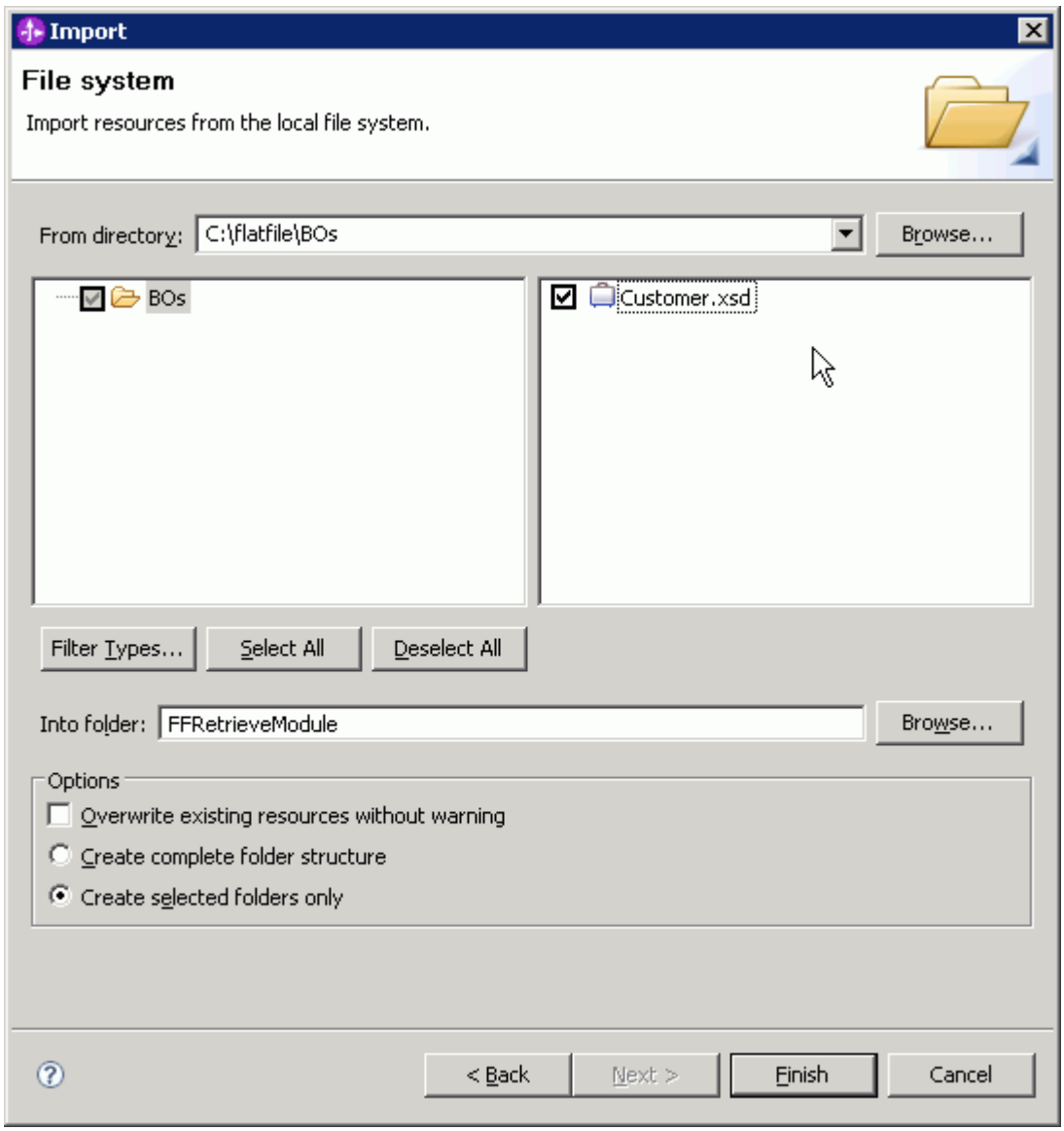
- Enter a desired name for your module. In this tutorial, FFRetrieveModule is set as the value in the **Module Name** field. Click **Next**.



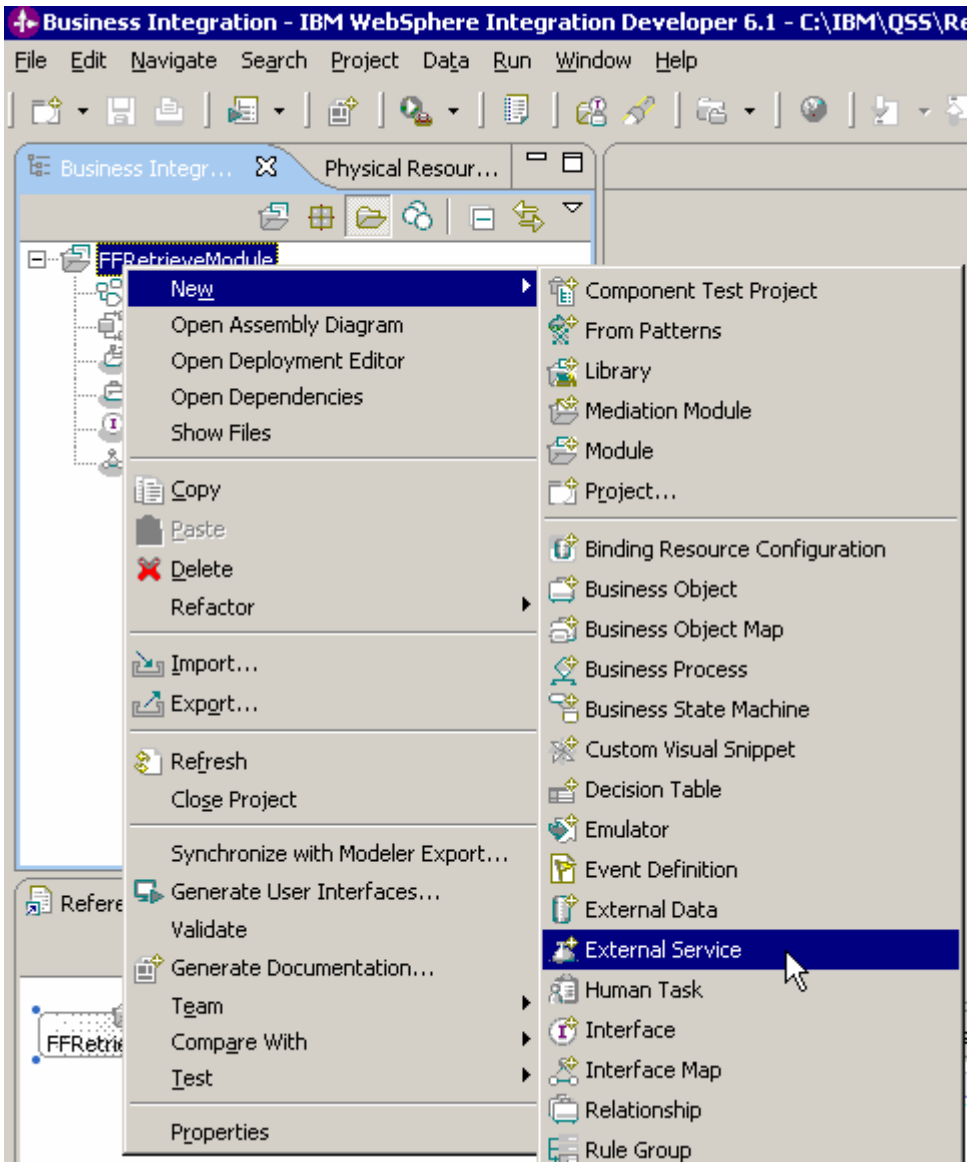
- Select whether you would like to reference this module to an existing solution in workspace. To create your own new solutions select **Do not include module in solution**, and then click on **Finish**.



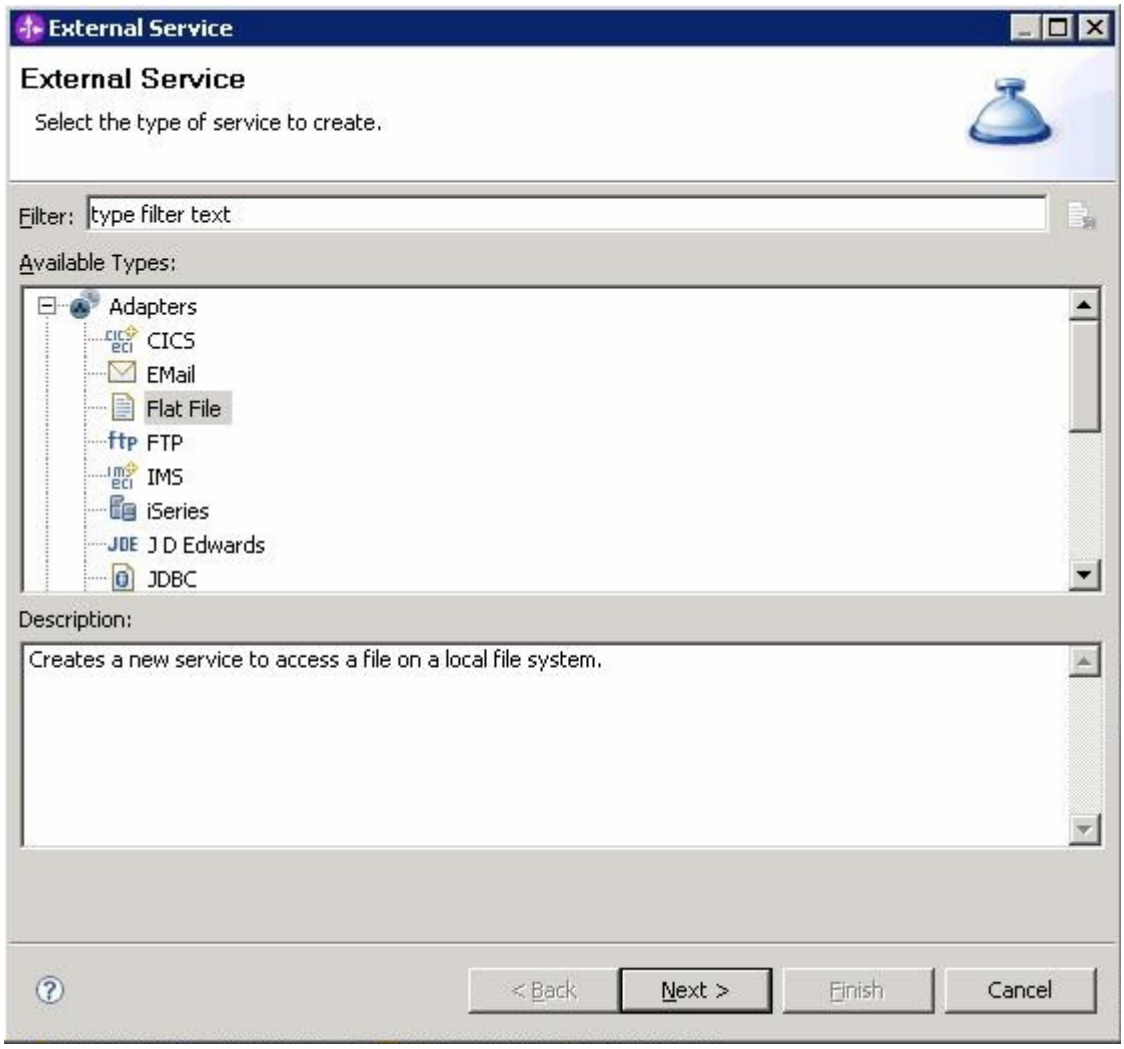
- Import you XSDs into the module created using **File** -> **Import** -> **File system** -> **Browse** to the location of your XSD(s) -> **Select** -> **Finish**. XSDs will be imported into Data type view as seen in Business integration perspective of WebSphere Integration Developer.



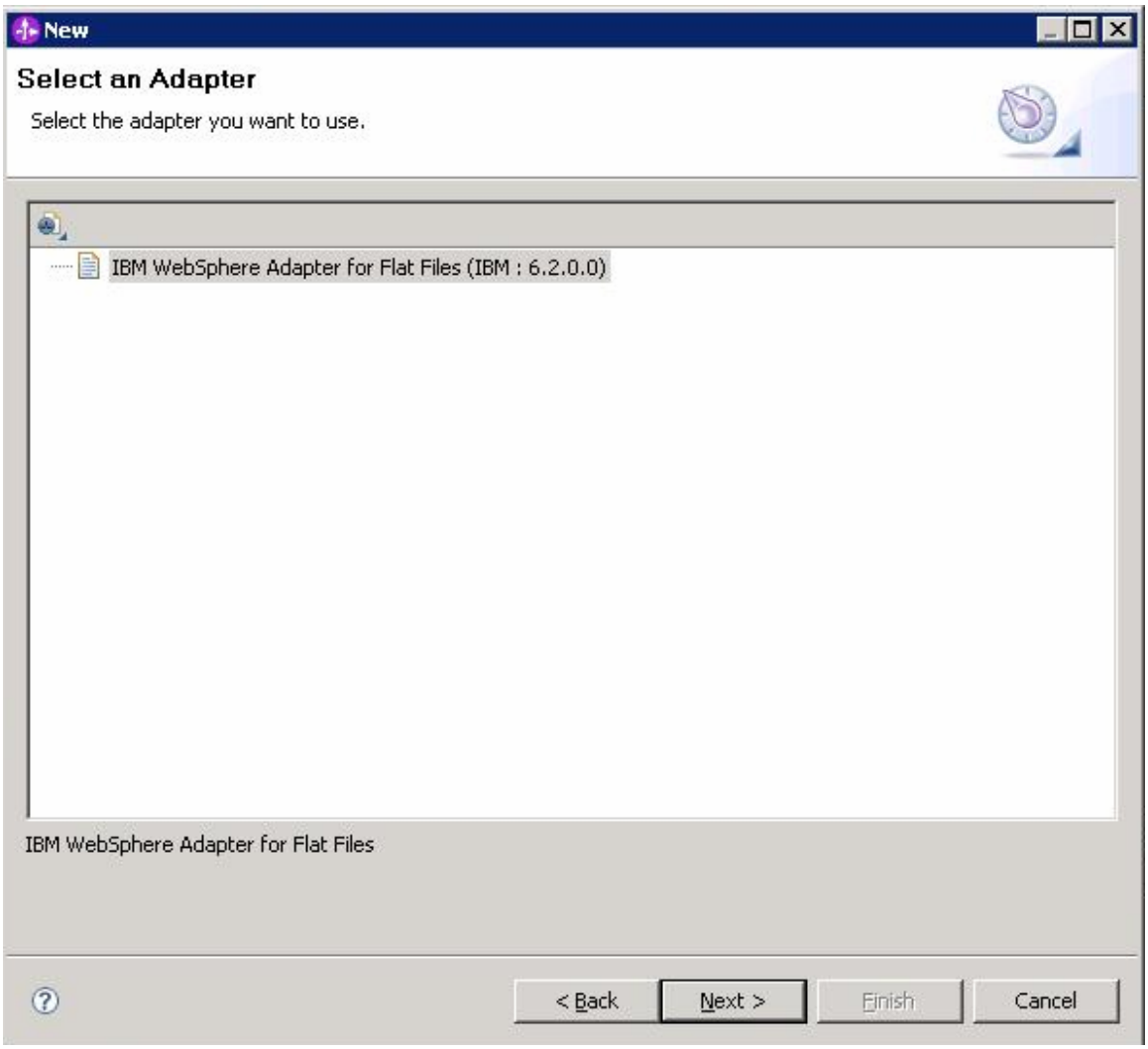
- As displayed in the screen capture below, start the external service wizard by selecting **File** -> **New** -> **External Service**.



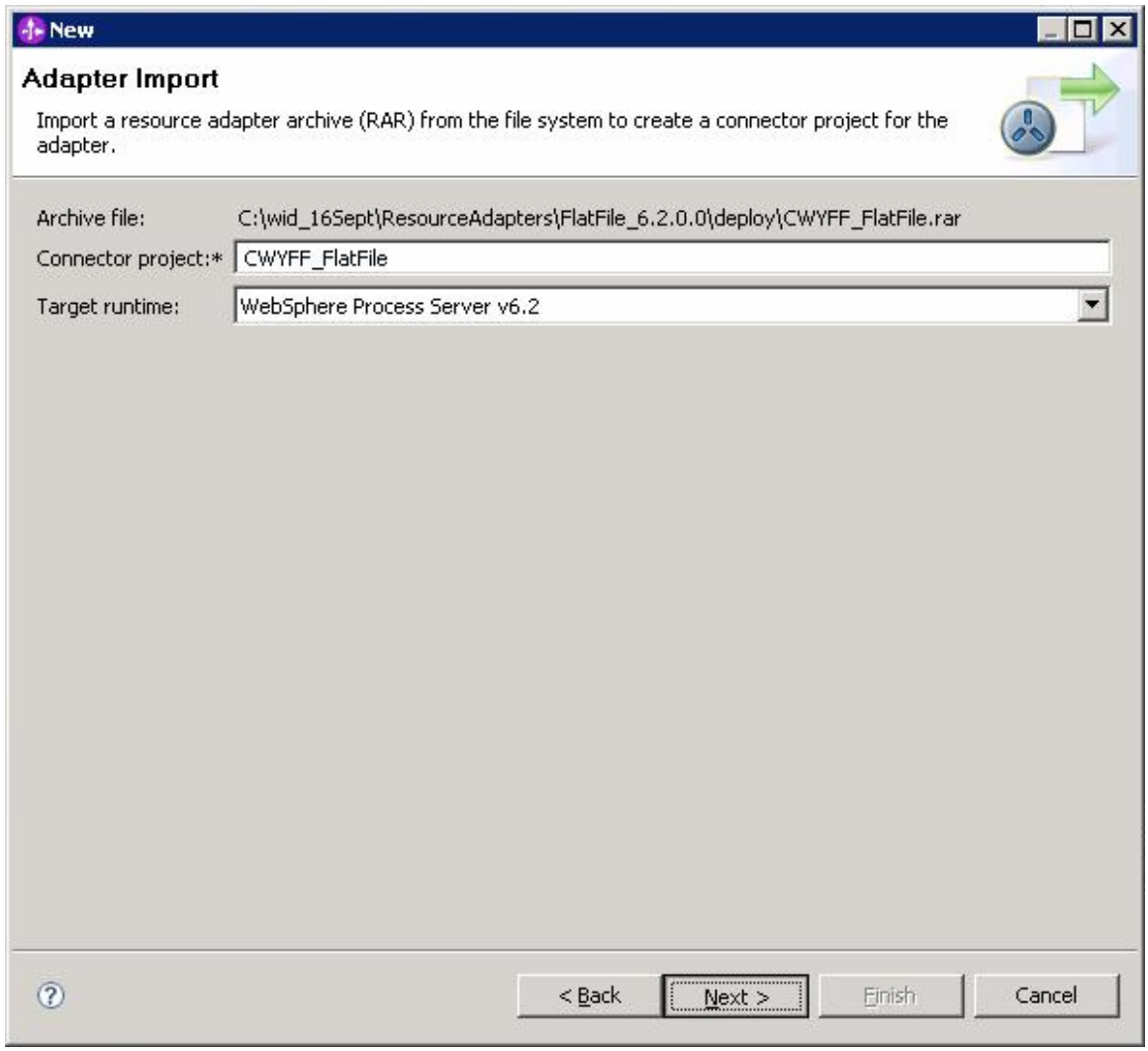
- Expand **Adapters,** select **Flat File** and click **Next**.



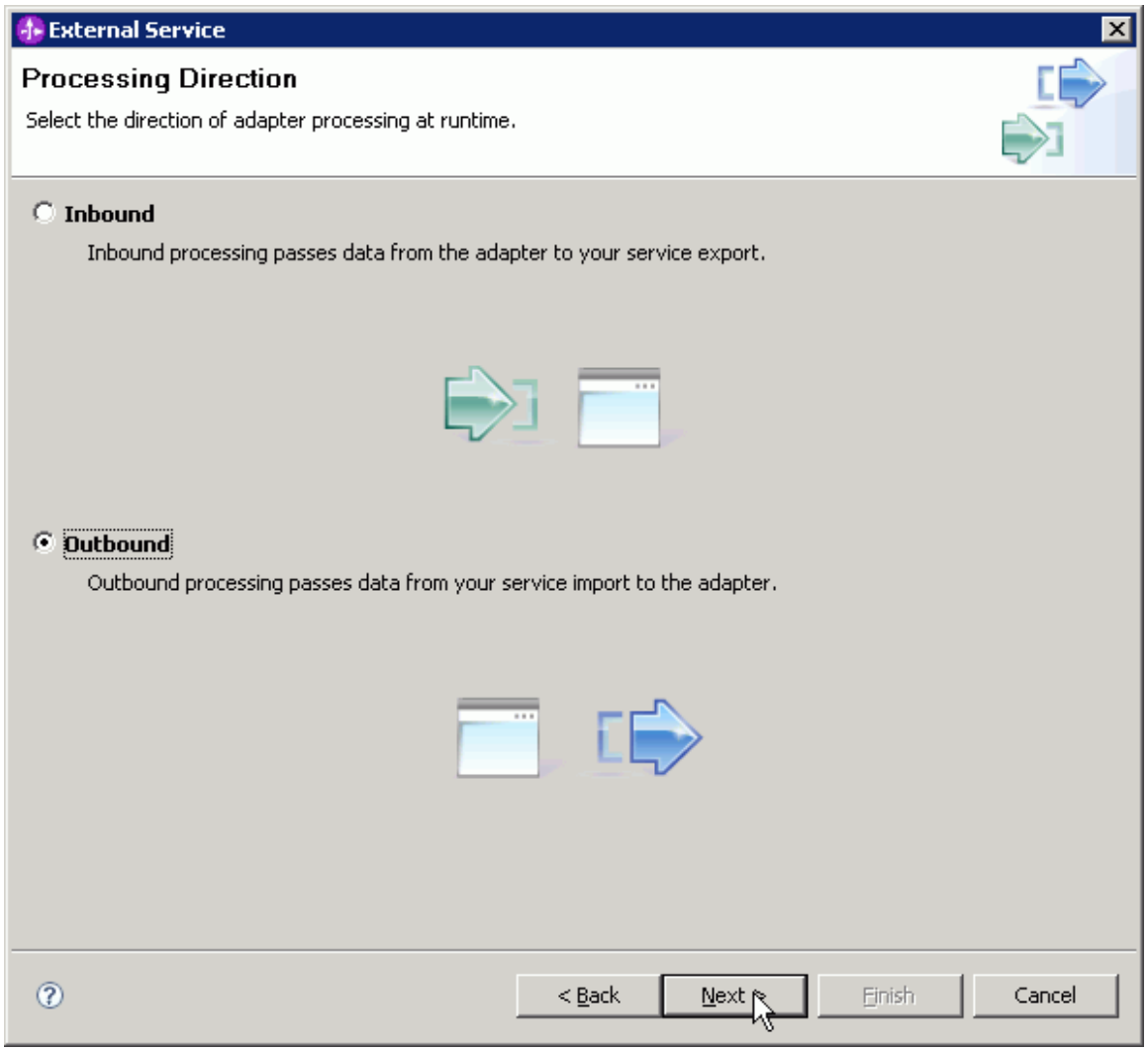
- In the Select an Adapter window, select **IBM WebSphere Adapter for FlatFiles (IBM : 6.2.0.0)** and click **Next**.



- In the Adapter Import window, adapter RAR file will be imported into the module. Leave the default value in the **Connector project** field. Ensure that the Target runtime is set with a value of **WebSphere Process Server v6.2**. Click **Next**.

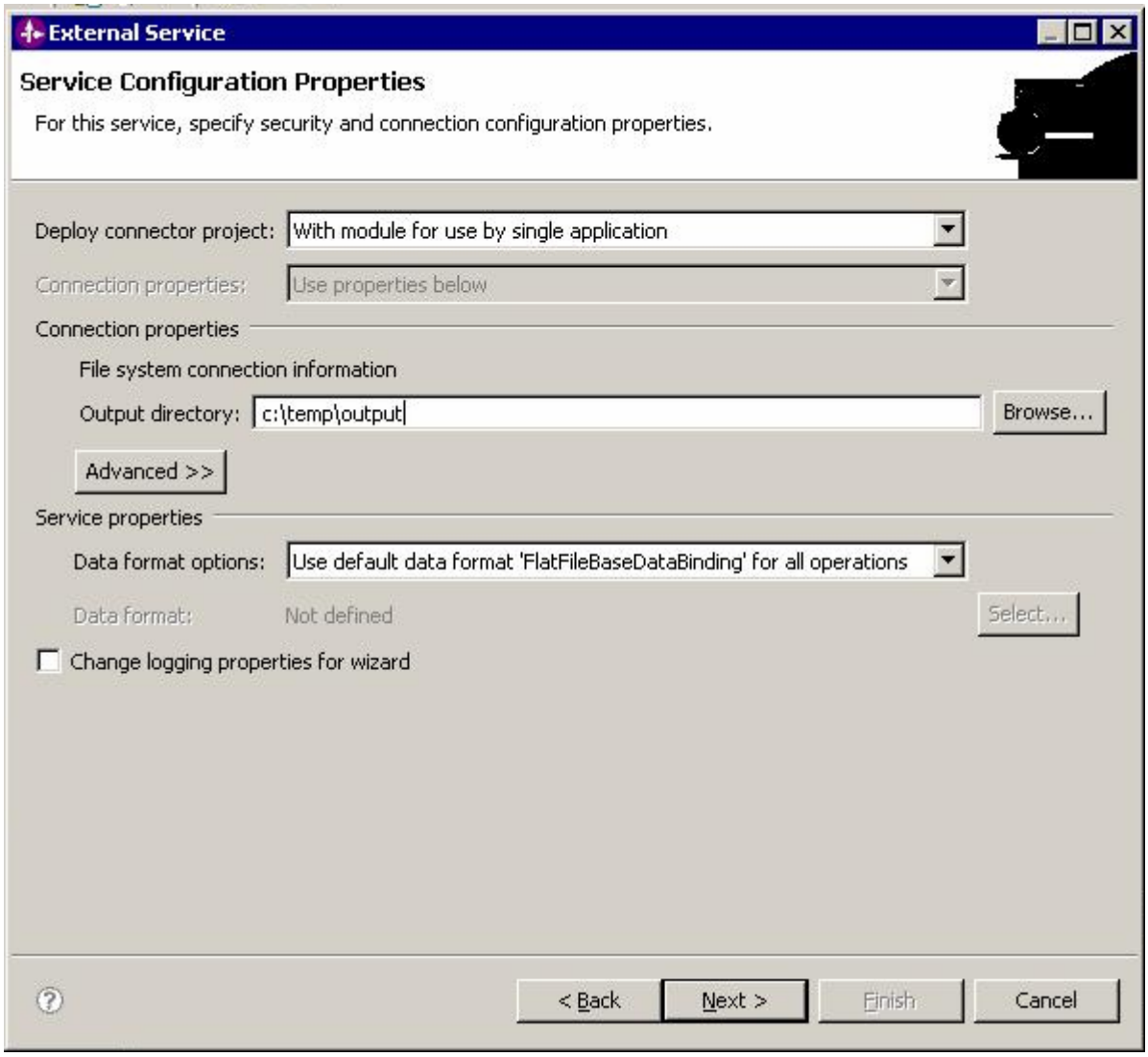


- In the Processing Direction window, select **Outbound** and click **Next**.

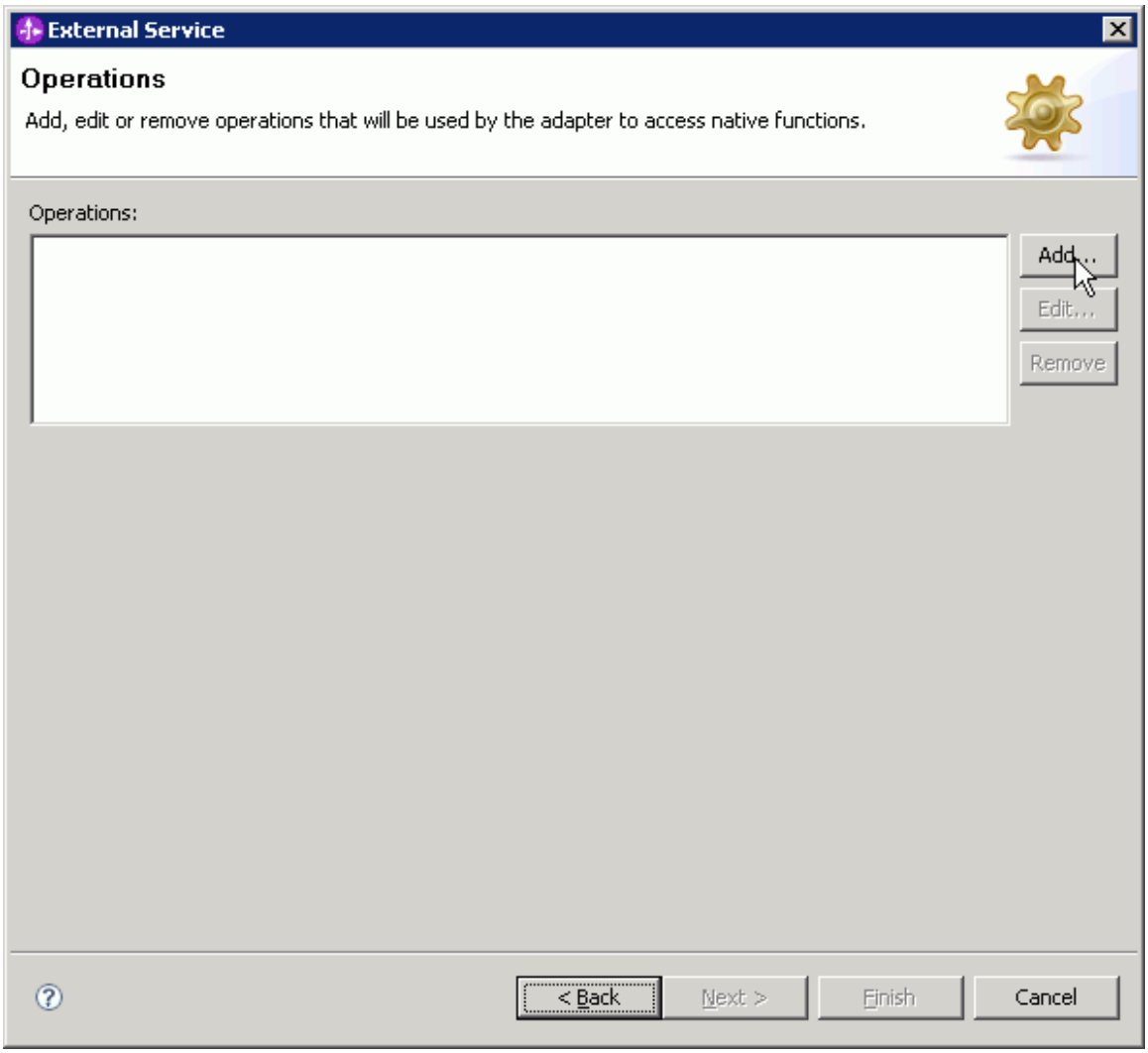


Setting connection properties for the external service wizard

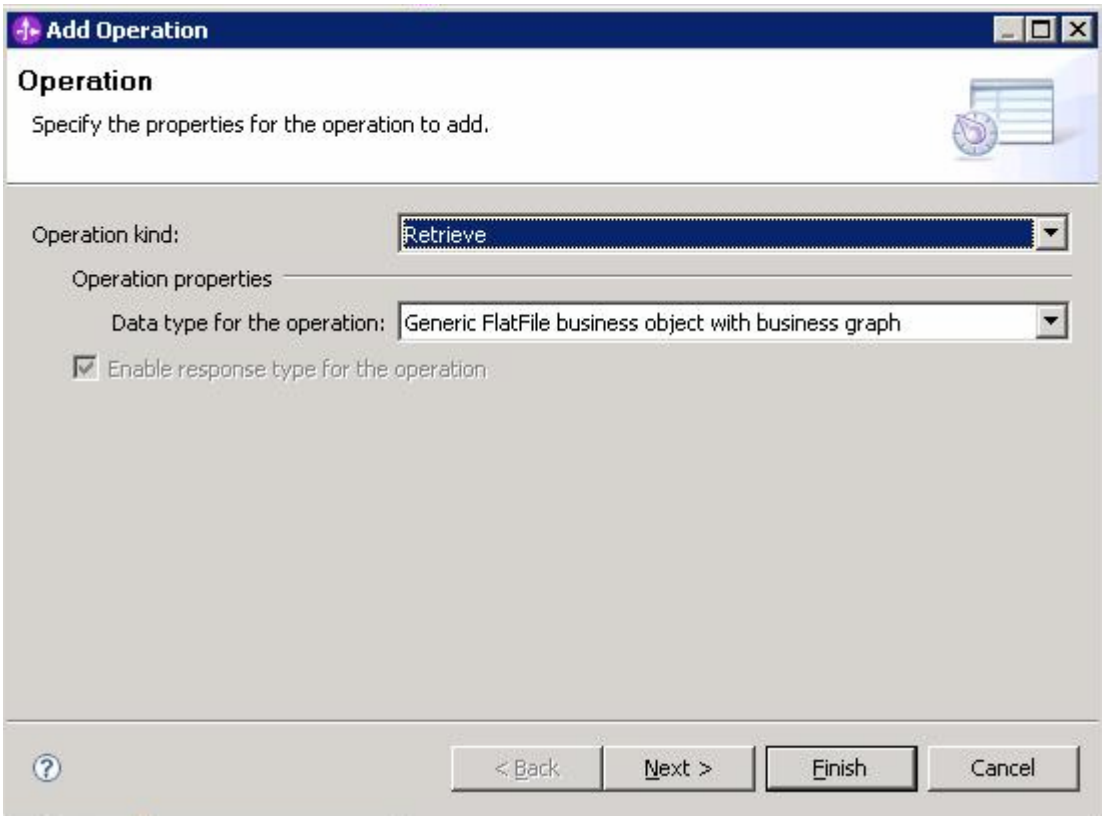
- In the Service Configuration Properties window shown below, you need to specify value for **Output directory** and optionally choose to specify the **Default file name**, **Sequence file** and **Staging directory** at the managed connection factory level. In this particular illustration, skip this step and specify values at record level. Click Next



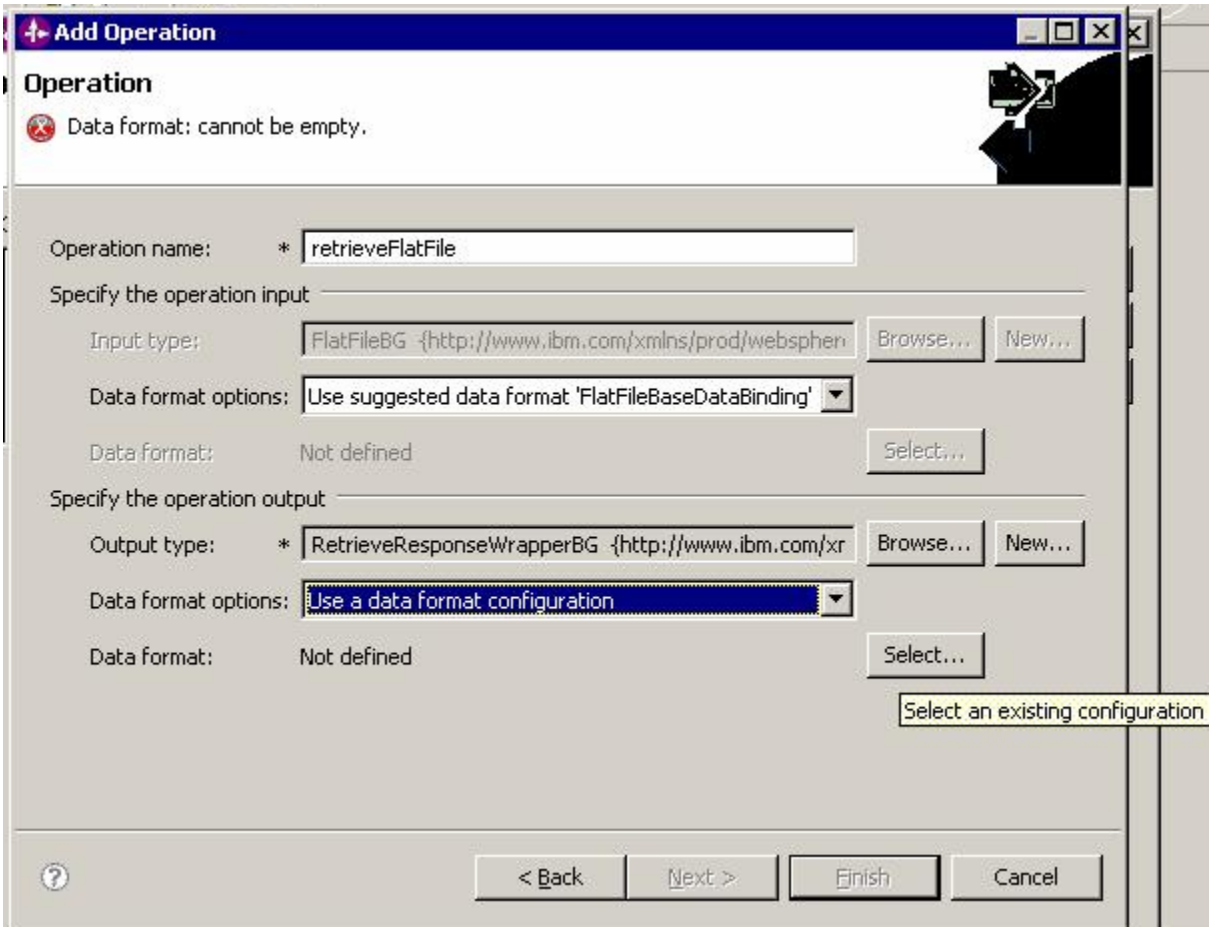
- In the Operations window below, click **Add** to add the operations that you want to perform.



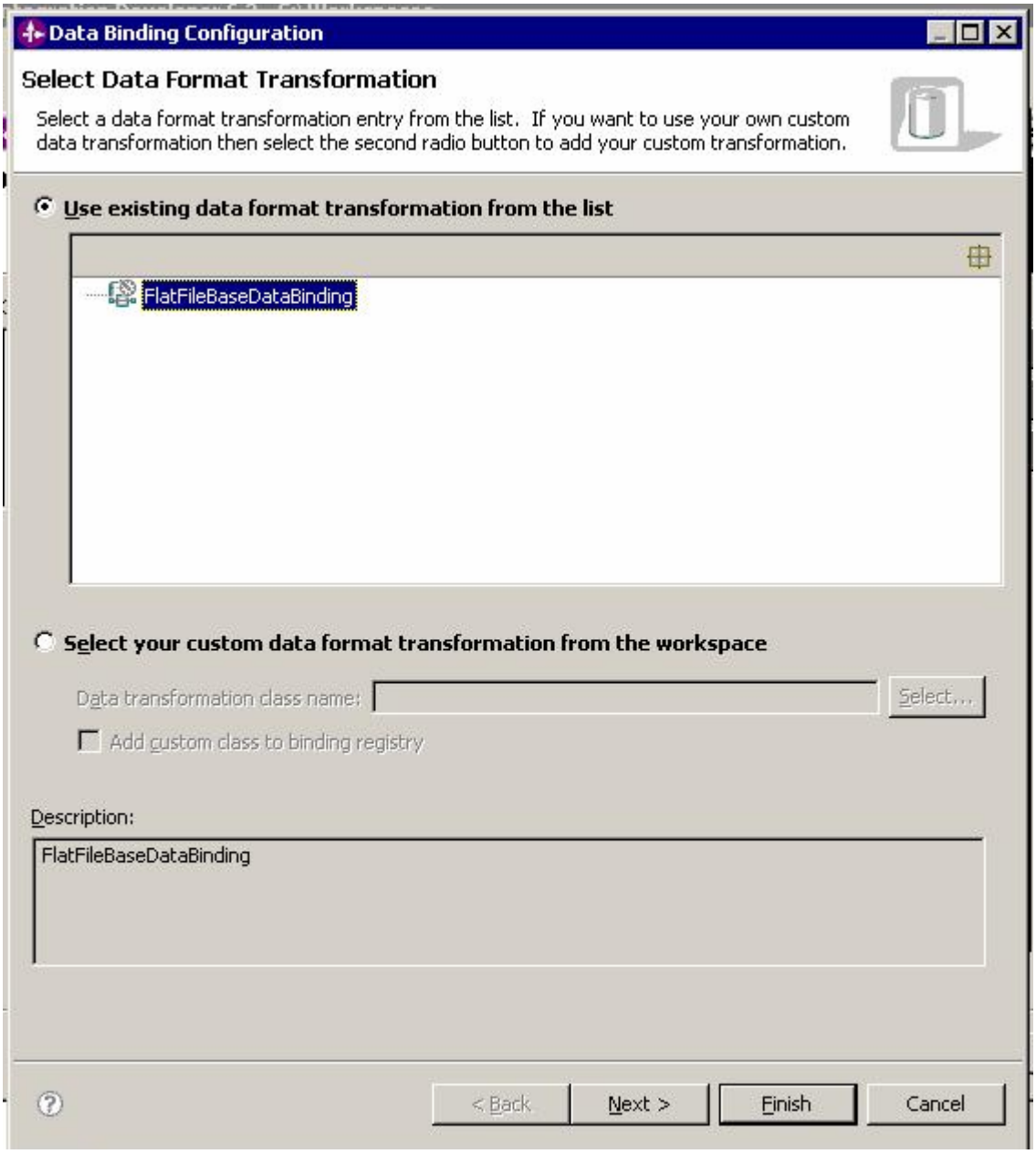
- Choose **Retrieve** from the **Operation kind** list and **Generic FlatFile business object with business graph** from the **Data type for the operation** list and click **Next**.



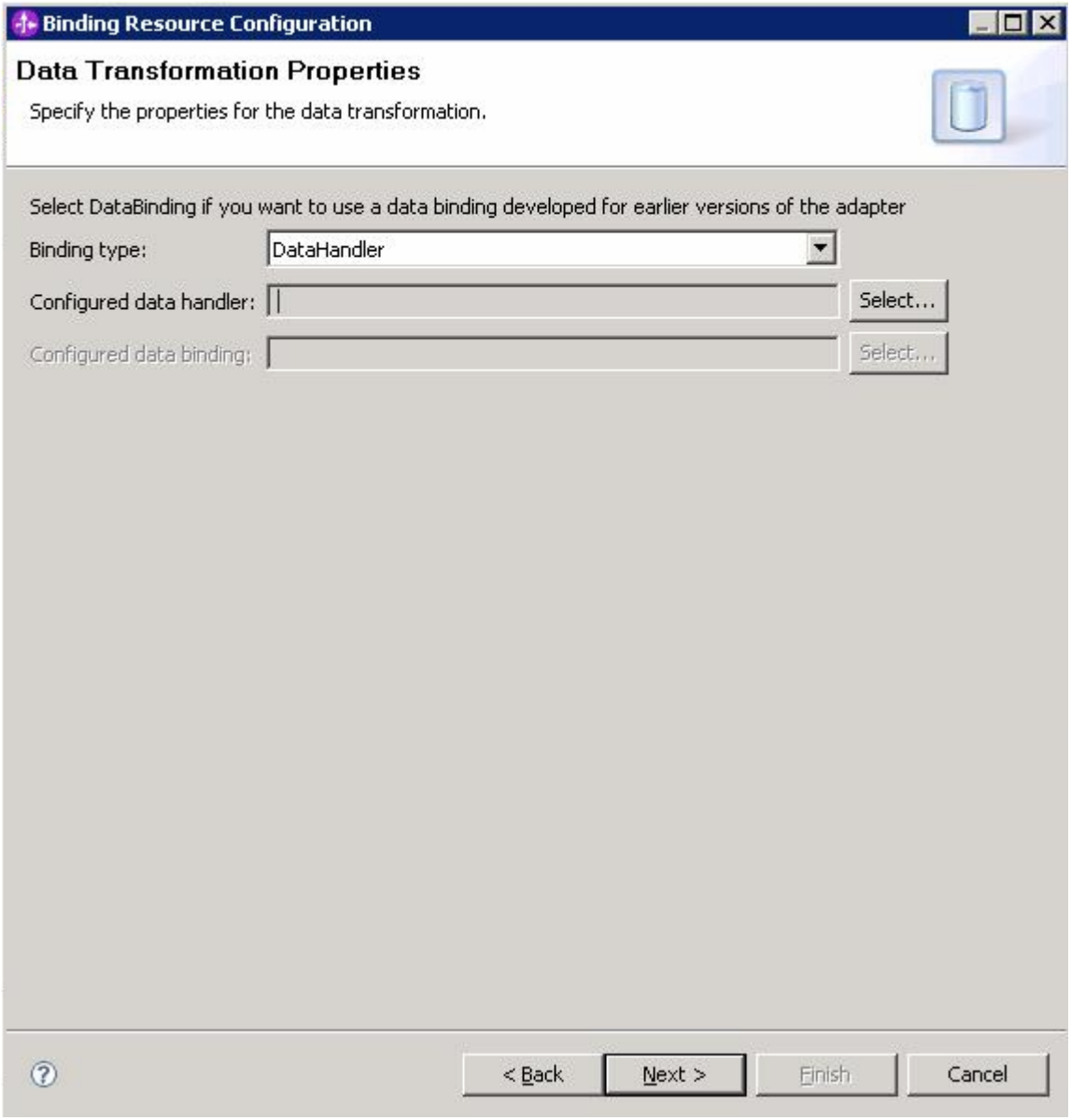
- Next, create a Data binding configuration. In the list box, in alongside **Data format options** label, select **Use a data format configuration**. Click **Select** button alongside **Data format** label.



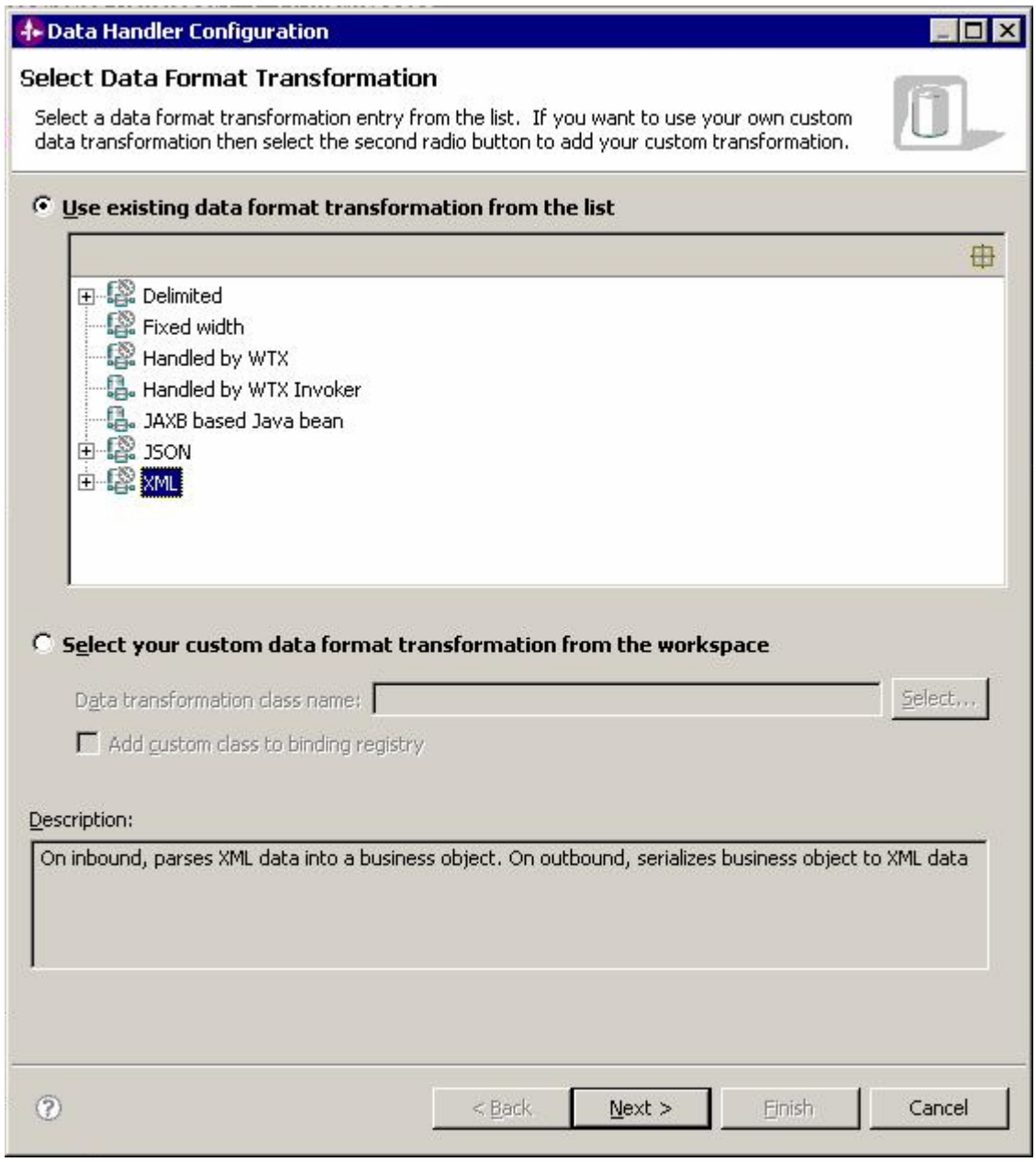
- Following screen capture is displayed next. In this particular illustration, carry out data transformation using FlatFileBaseDataBinding alongwith XML Data handler. Hence choose **FlatFileBaseDataBinding** from the available classes, click **Next**.



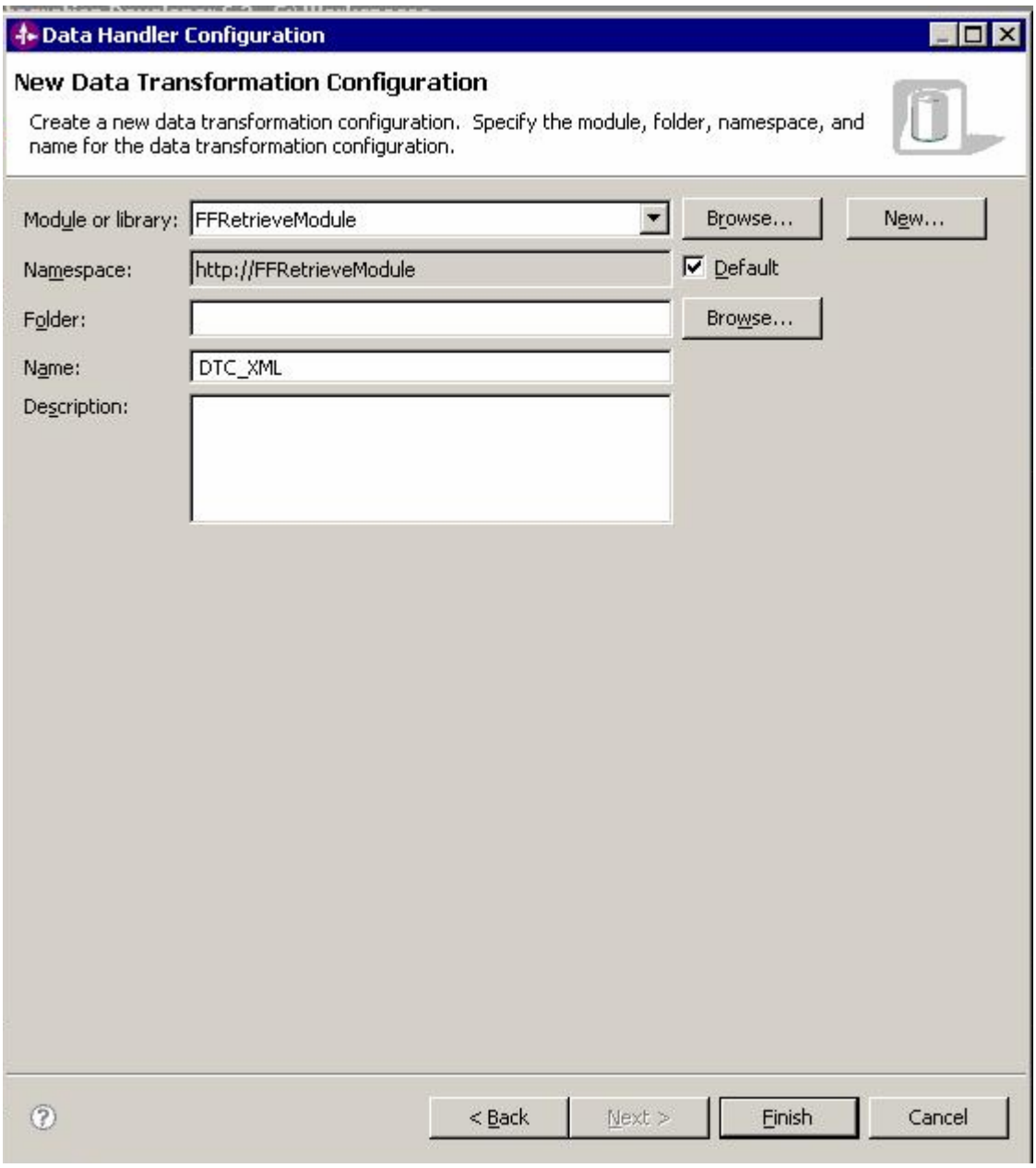
- In the Data Transformation Properties window, leave the **Binding type** value as **DataHandler** and click **Select** to configure the **Datahandler**.



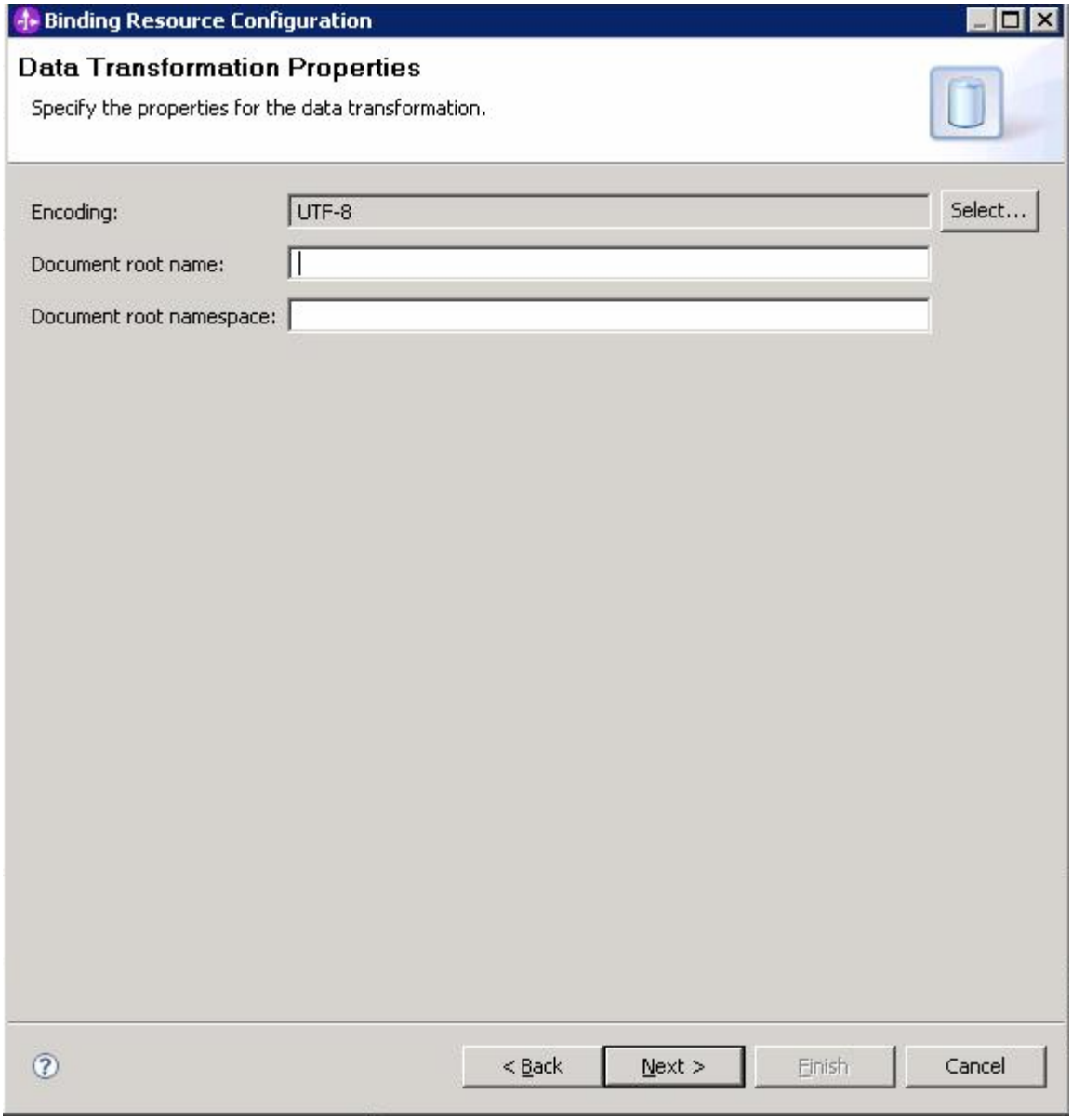
- Select **XML** from the list of datahandlers and then click on **Next**.



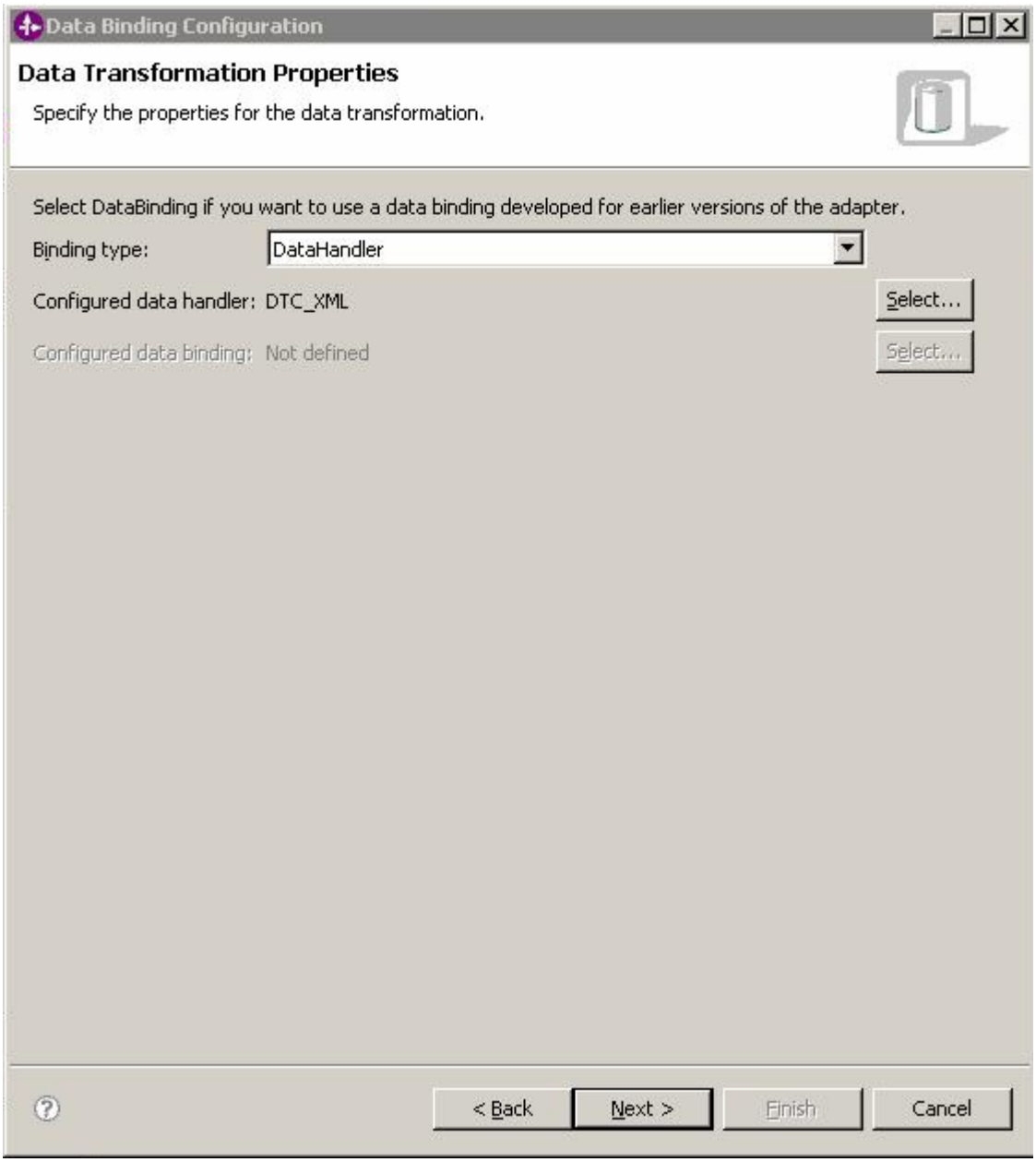
- You will see the New Data Transformation Configuration window as displayed below. Specify a name for the Datahandler (DTC_XML is used as datahandler **Name** in this illustration) and then click **Finish**.



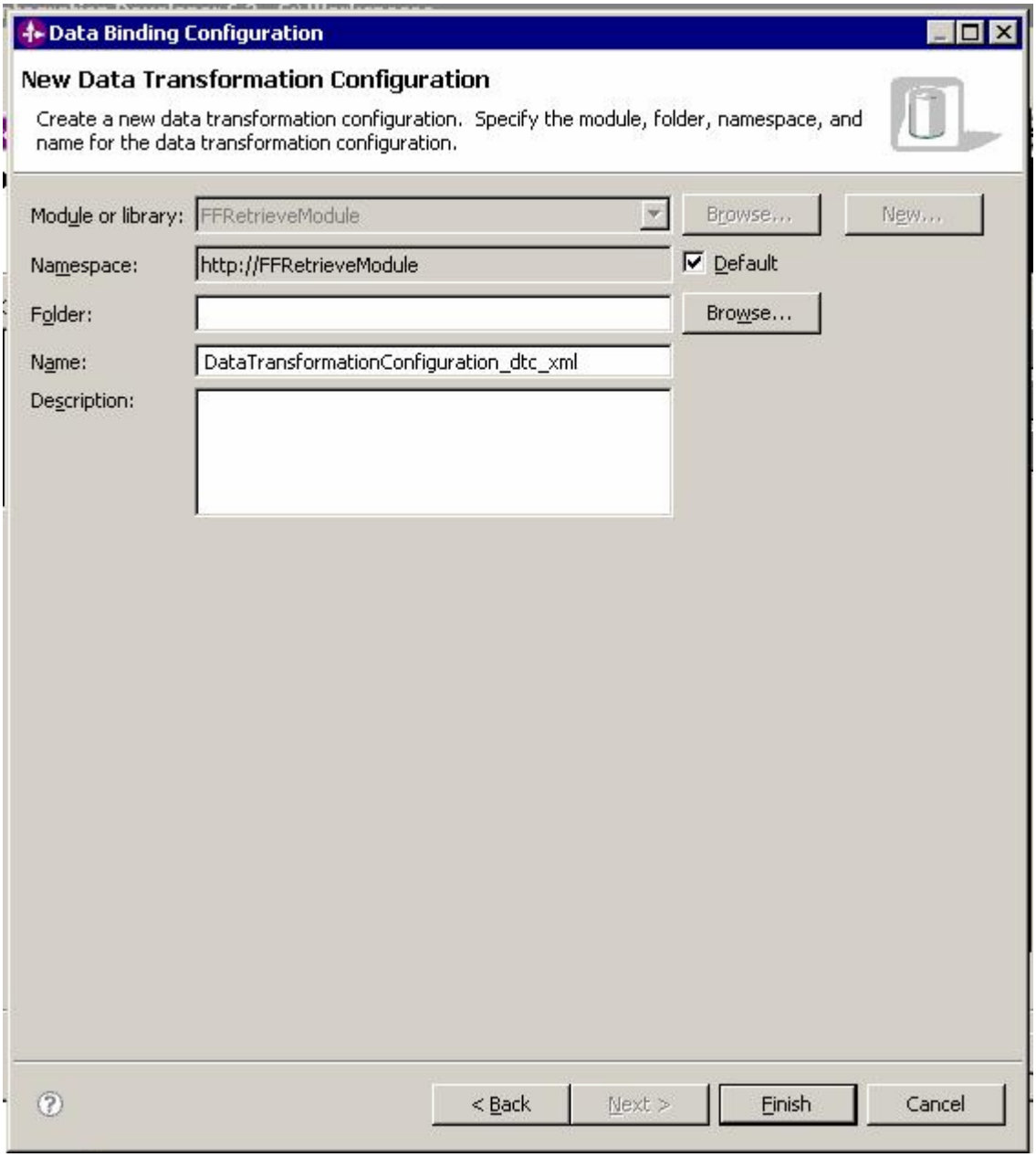
- In the Data Transformation Properties window shown below, leave the default value for **Encoding** as **UTF-8**. Leave the **Document root name** and **Document namespace** fields as blank and click **Next**.



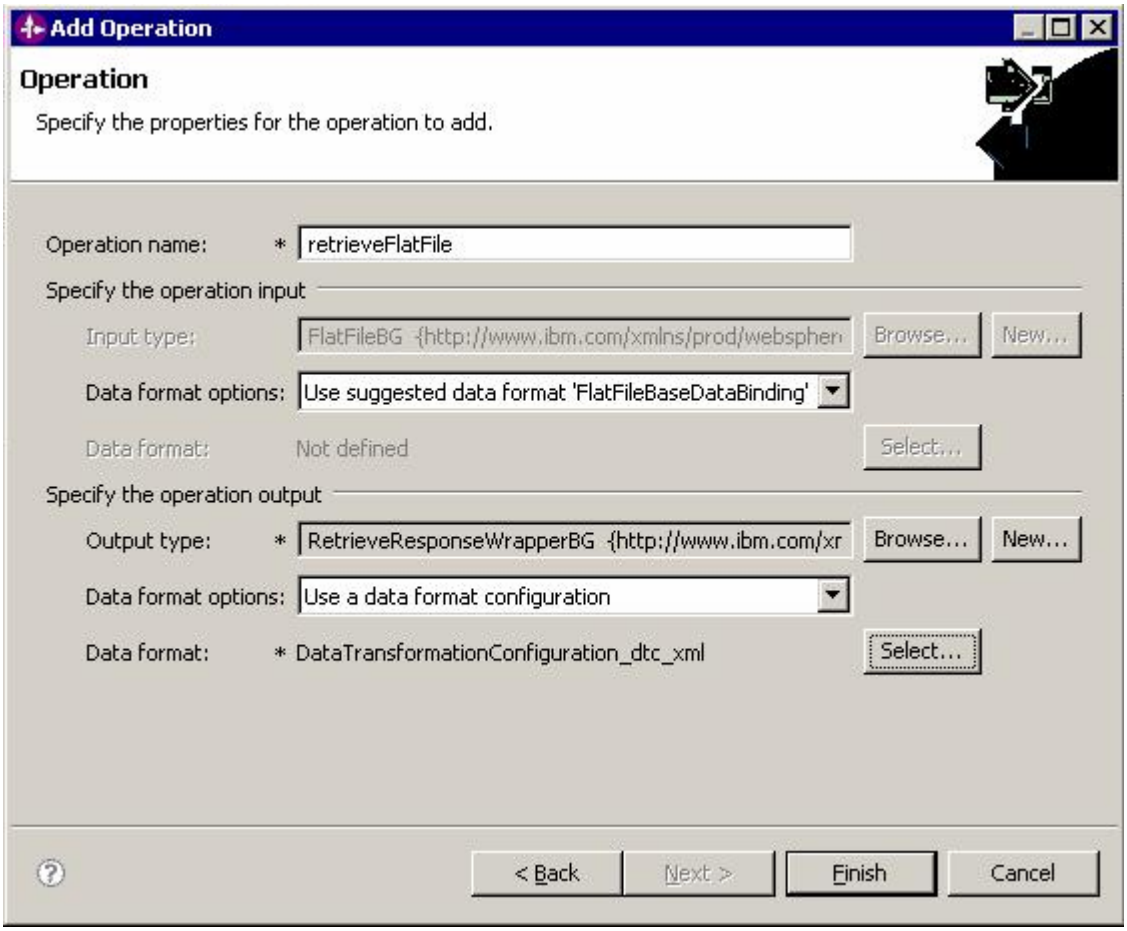
- The data handler is configured as displayed in the window. Click **Next**.



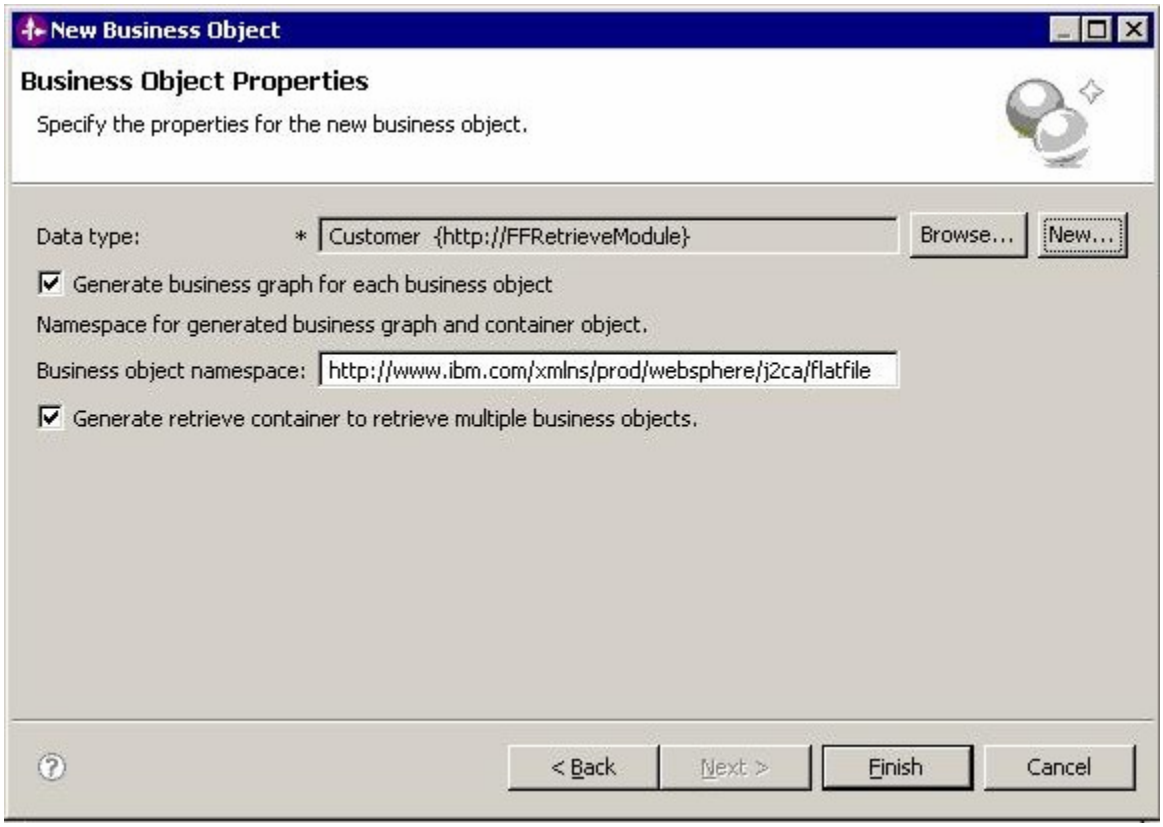
- Specify a name for the Databinding (DataTransformationConfiguration_dtc_xml is the name given in this illustration), and then click **Finish**.



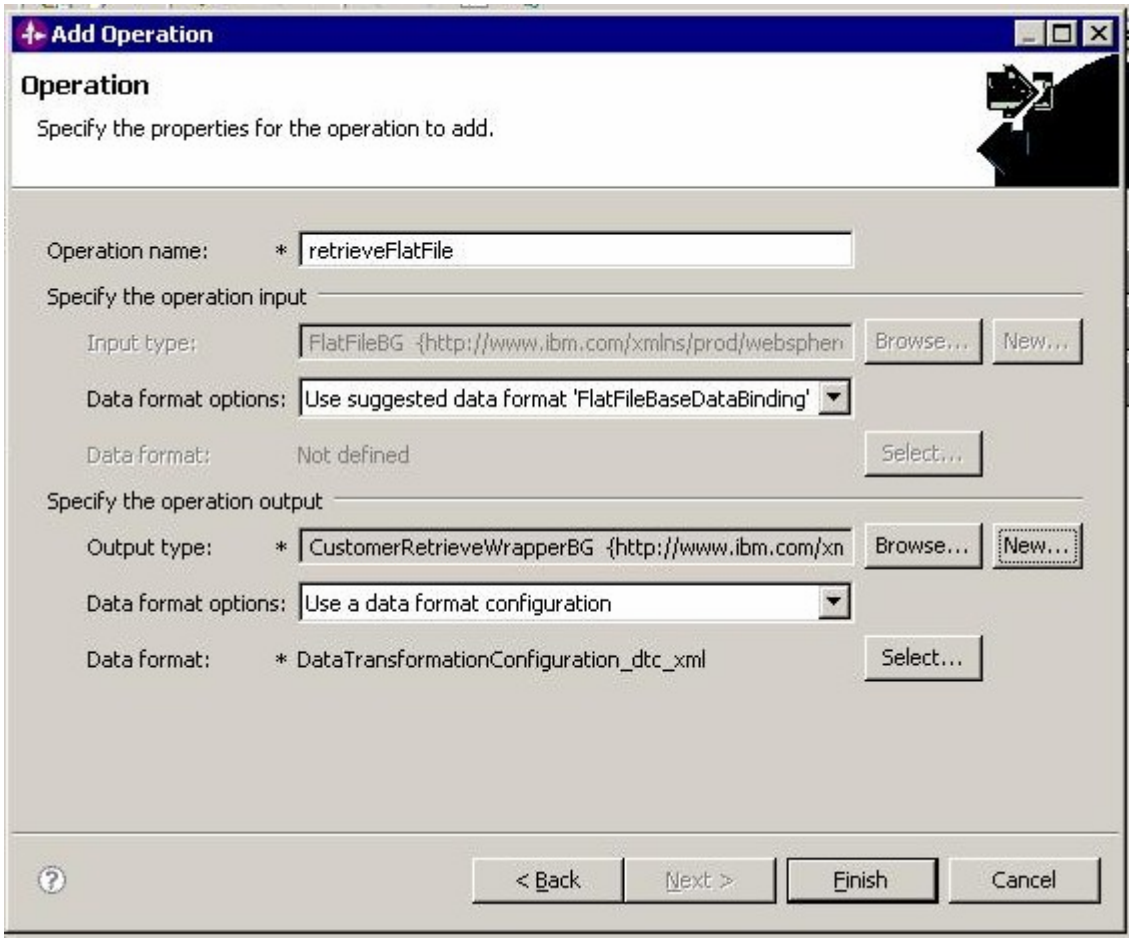
- The following window is displayed. Click **Finish**.



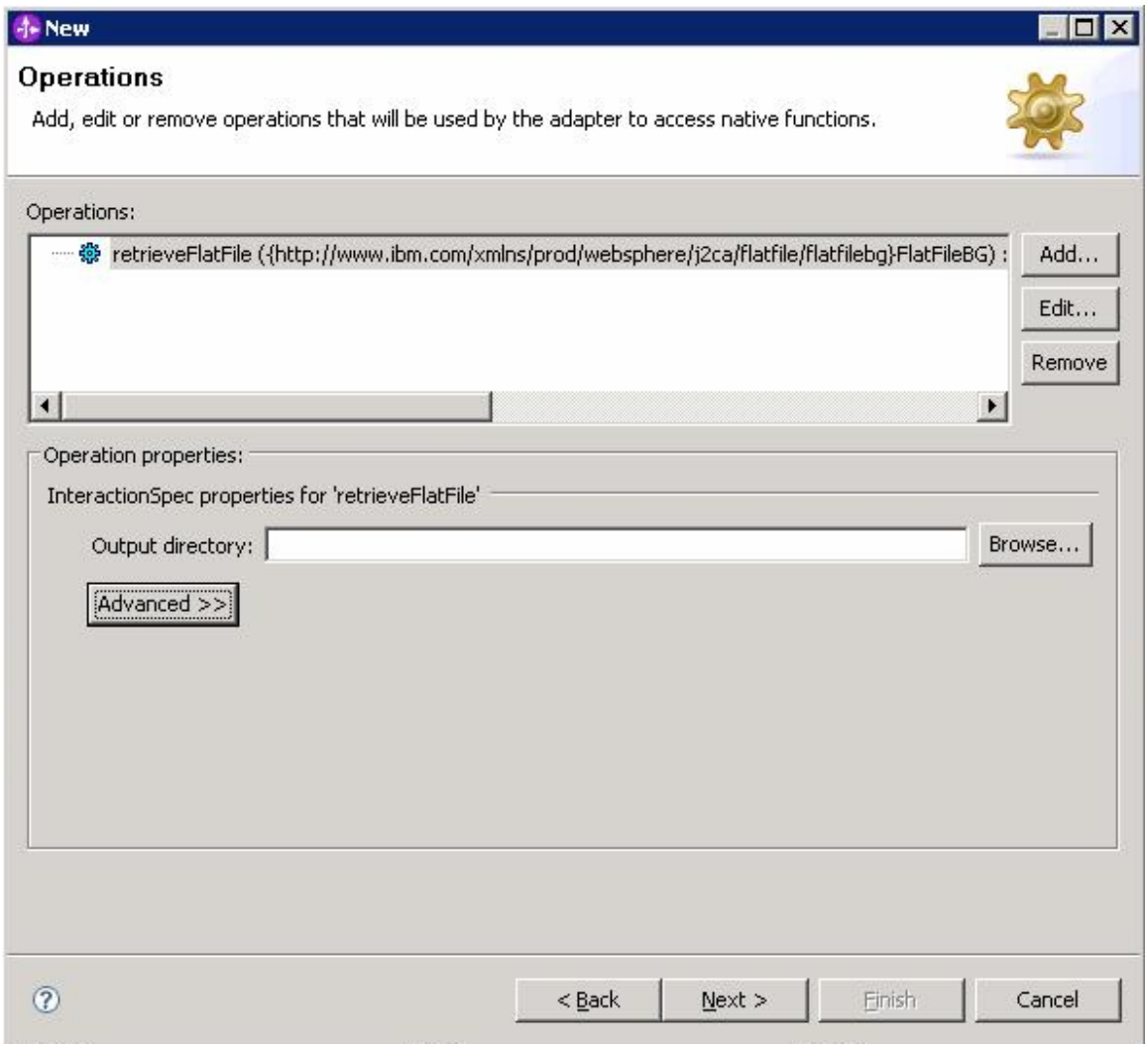
- Now select the **New** button along the Output Type label in order to configure a structured retrieve Response wrapper. Select the check box against the Generate Retrieve Container to retrieve multiple objects label.



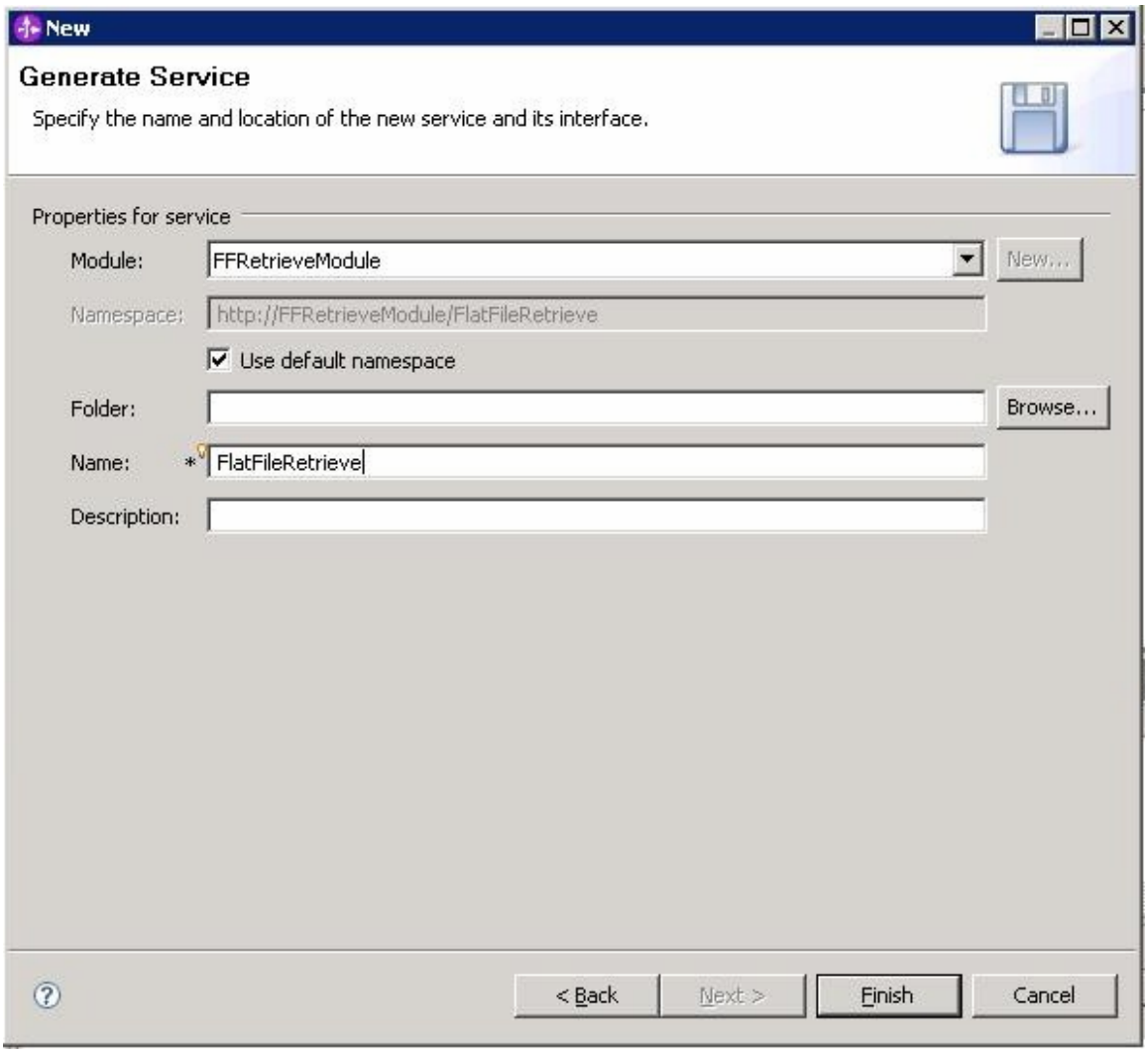
- Select the Finish button .



- The following window is displayed. This shows the interaction spec properties that can be set for the newly added operation. Do not enter any values for any of the fields. Click **Next**.



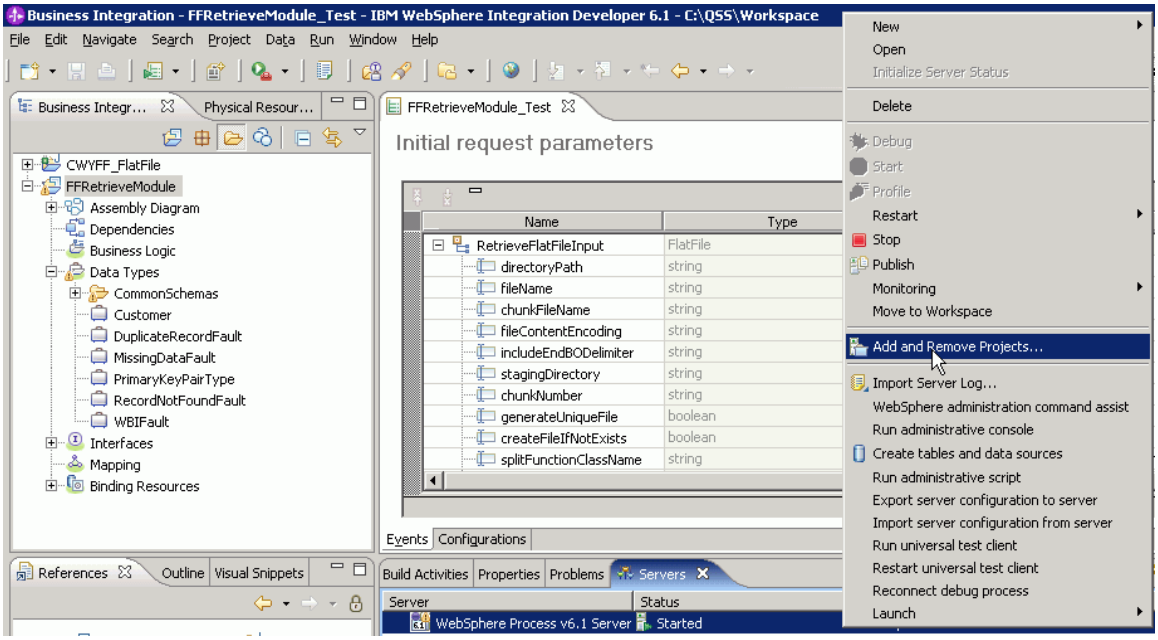
- In the last step of running external service wizard, specify a suitable name (FlatFileRetrieve is the name used in this Illustration) for your adapter interface, or the default name "FlatFileImport" can also be chosen. Click **Finish**.

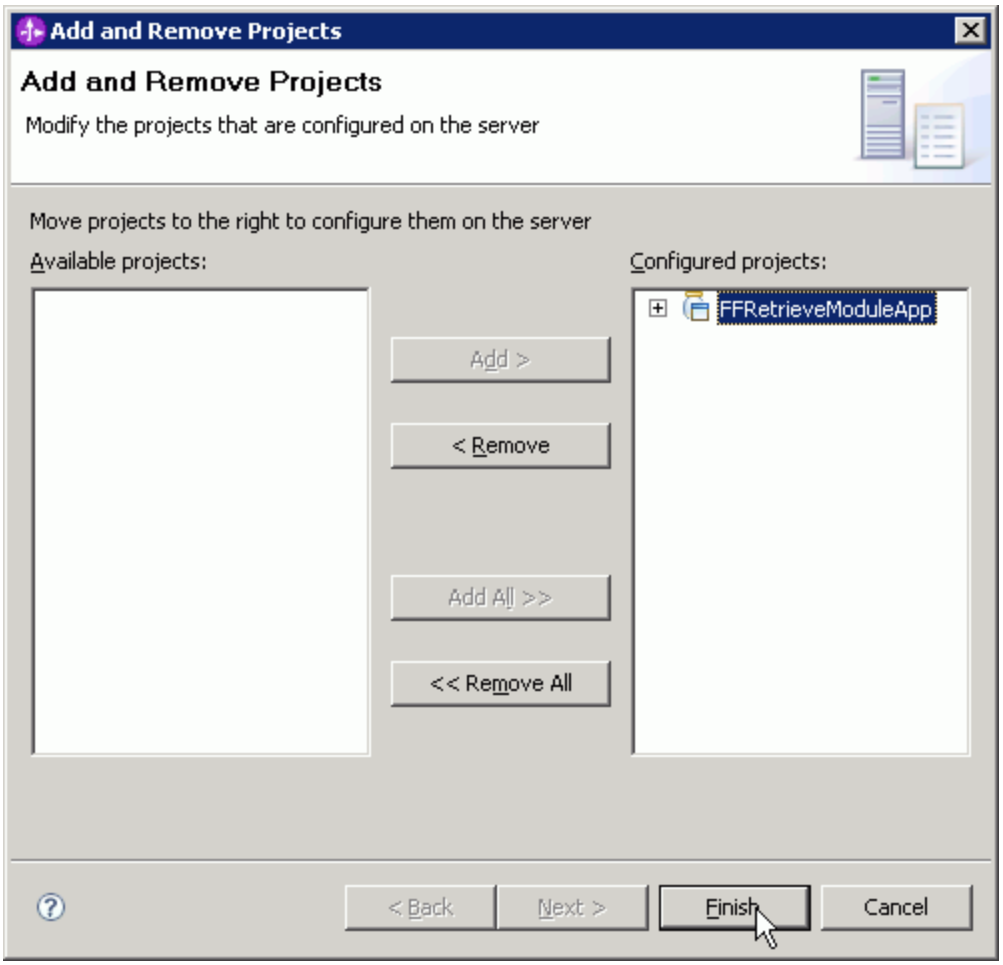


Deploying the module to the test environment

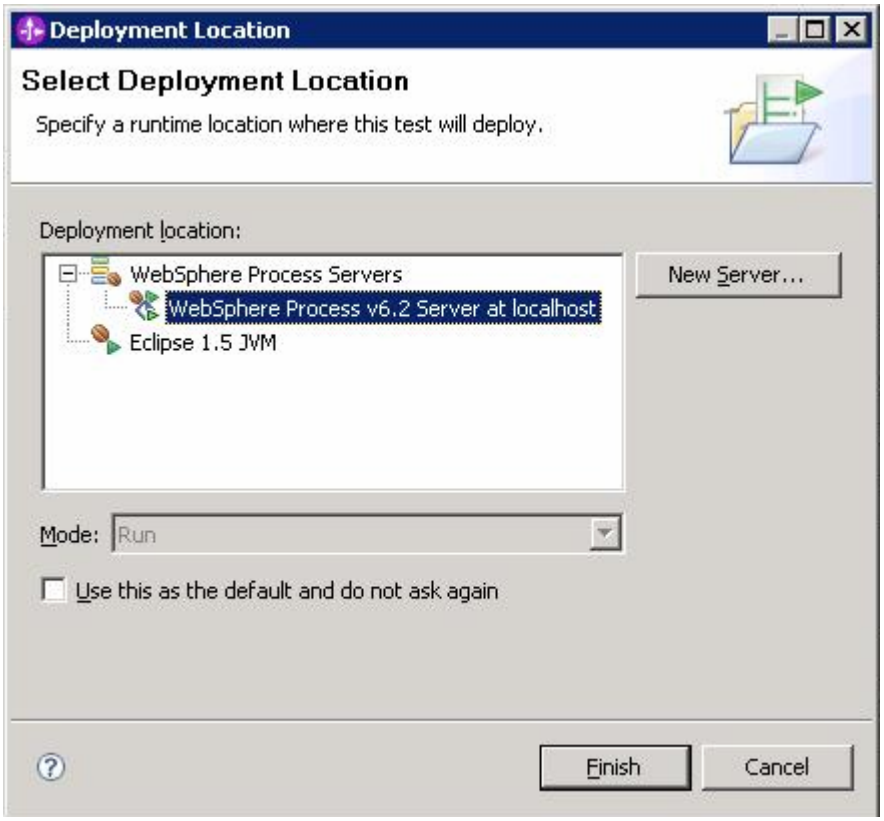
The following steps how how to perform a Retrieve operation with the module you just created.

- Start the WebSphere Process Server.
- Add adapter module to the server. In the Server tab, right click on WebSphere Process Server then click **Add and Remove Projects**.
- From **Available projects** pane, select your adapter module, click **Add >** and click **Finish**.





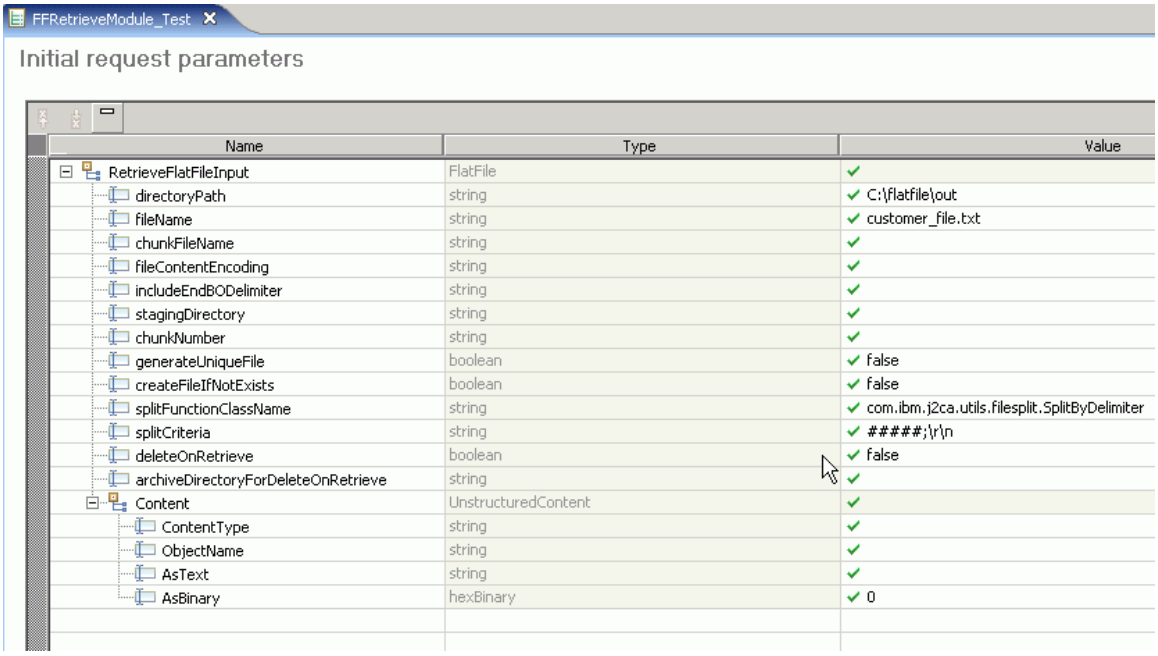
- In the Select Deployment Location window, click Finish.



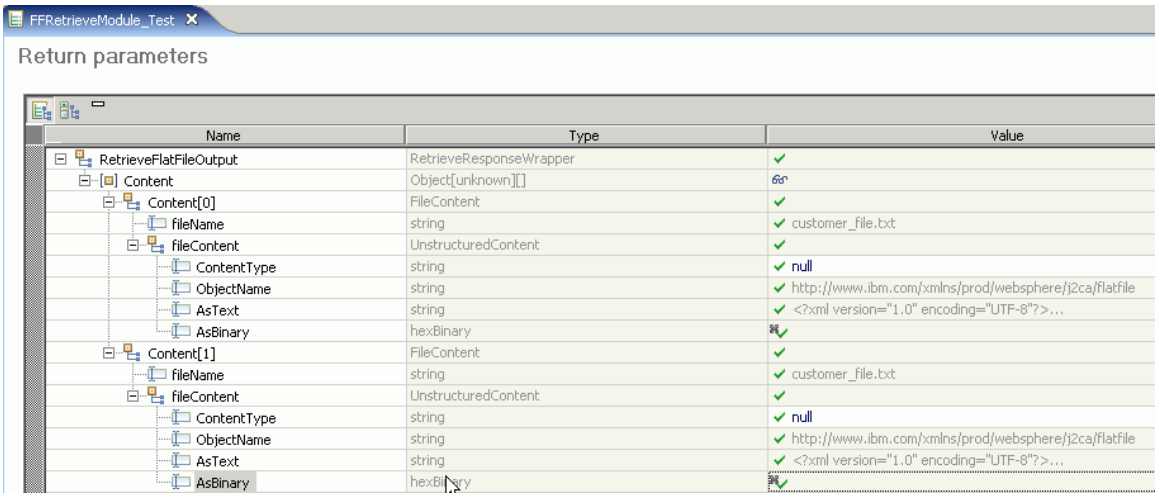
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client:

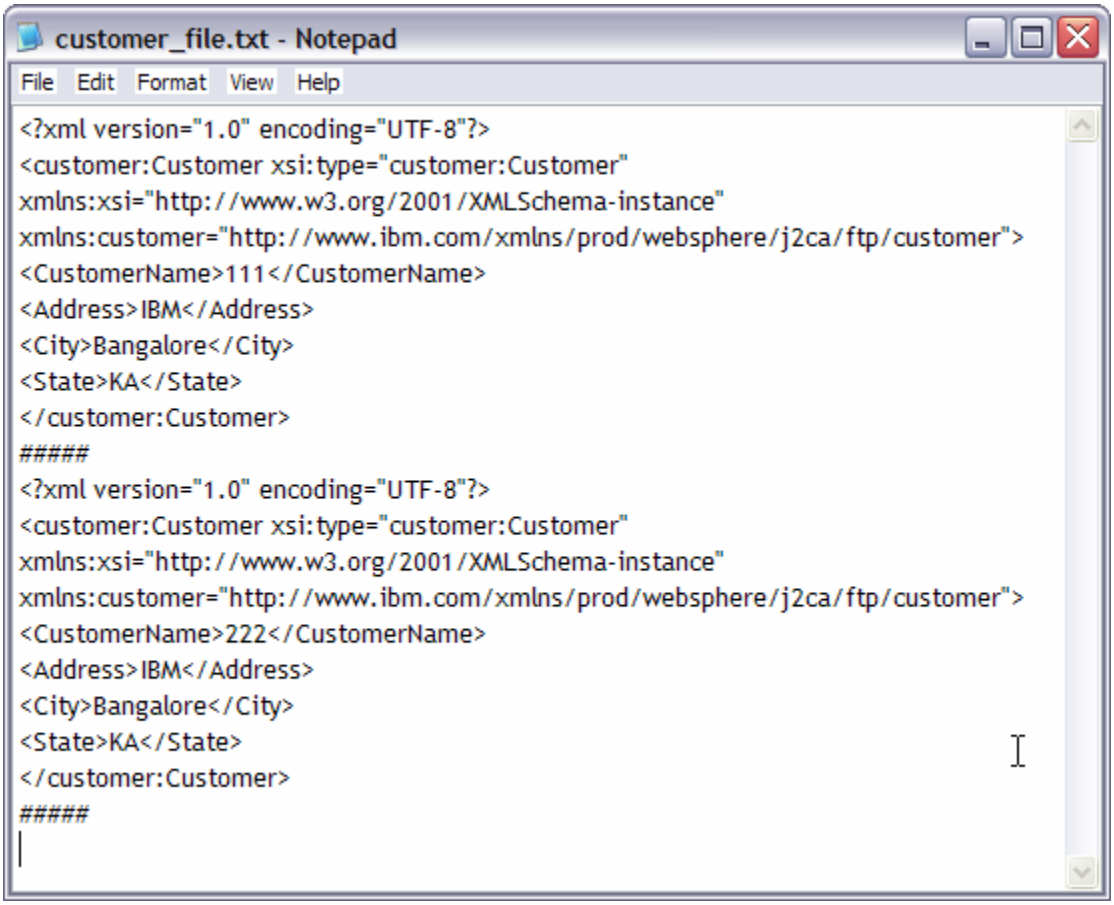
- Right click on the adapter module, FFRetrieveModule then click **Test** -> **Test module**.
- Specify the required parameters to carry out Retrieve operation, i.e., **directoryPath**, **fileName**. Since this tutorial covers XML data transformation during Retrieve operation with splitting configuration, specify **splitCriteria** as well as **splitFunctionClassName**.



- Click Invoke. The following result which is the output of Retrieve operation will be displayed



- A snapshot of the file whose content was retrieved is below.


customer_file.txt - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:customer="http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/customer">
<CustomerName>111</CustomerName>
<Address>IBM</Address>
<City>Bangalore</City>
<State>KA</State>
</customer:Customer>
#####
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:customer="http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/customer">
<CustomerName>222</CustomerName>
<Address>IBM</Address>
<City>Bangalore</City>
<State>KA</State>
</customer:Customer>
#####

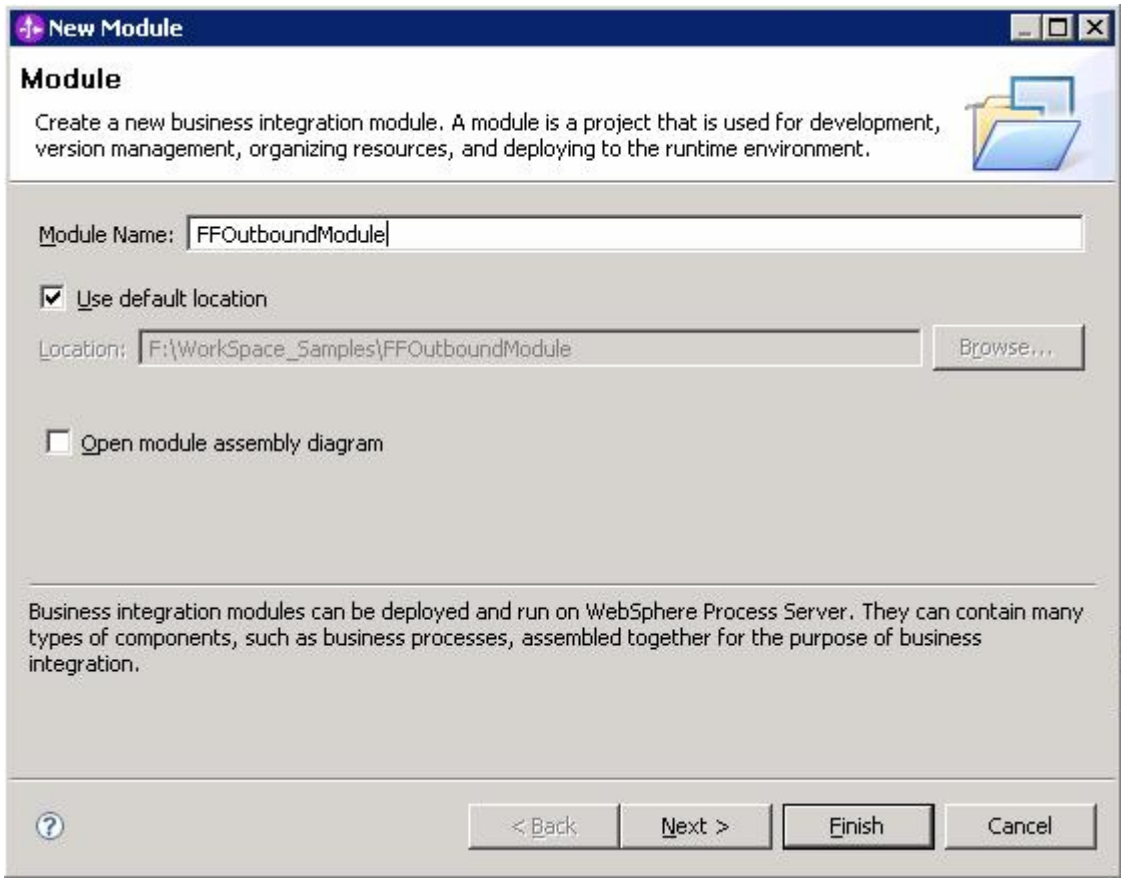
Chapter 4. Tutorial 2: Outbound processing – Demonstration of Sequence file option while creating a file

This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.2.0.0 can be used to create a file by making use of a managed connection factory property, i.e. Sequence file. When the absolute path of a file is specified in Sequence file property, while performing, Create operation, the adapter will sequence the file name to be created. The sequence numbers are serialized into a file specified at Sequence file. Hence sequencing continues even while running multiple sessions of the adapter.

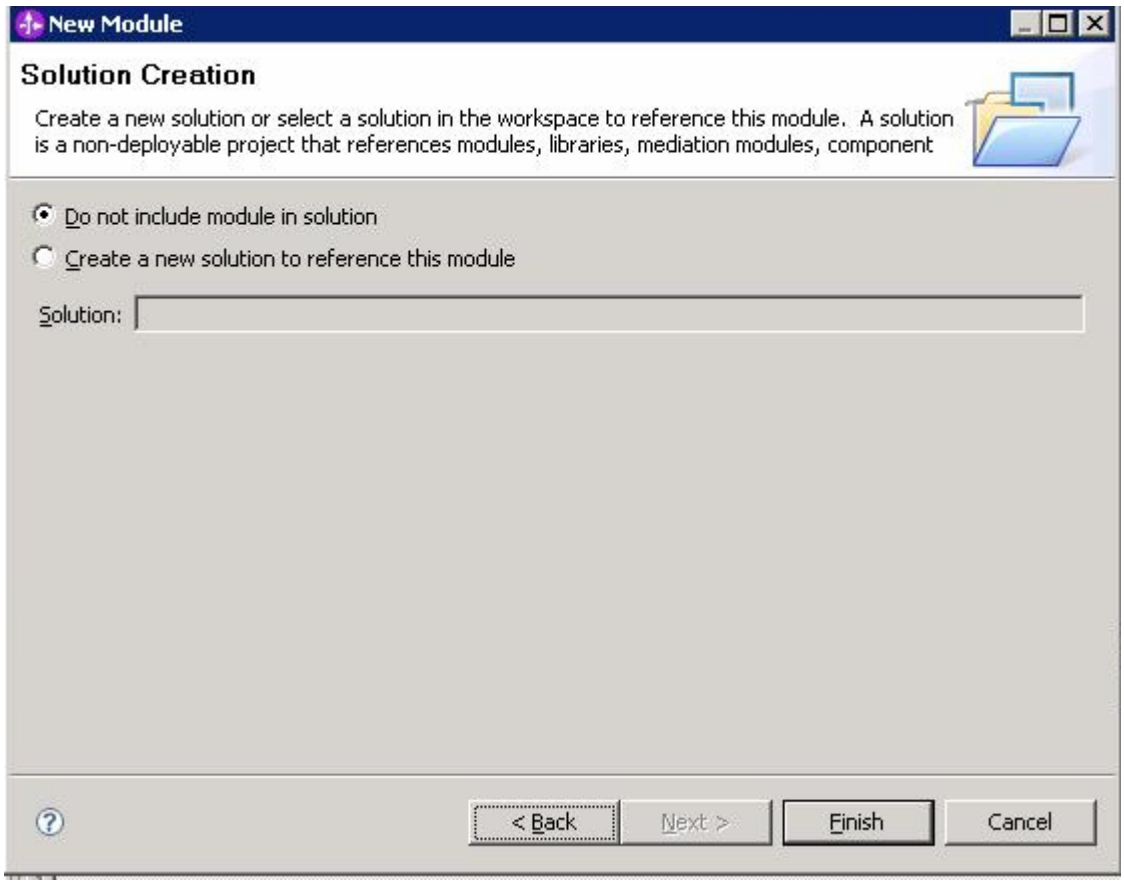
Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

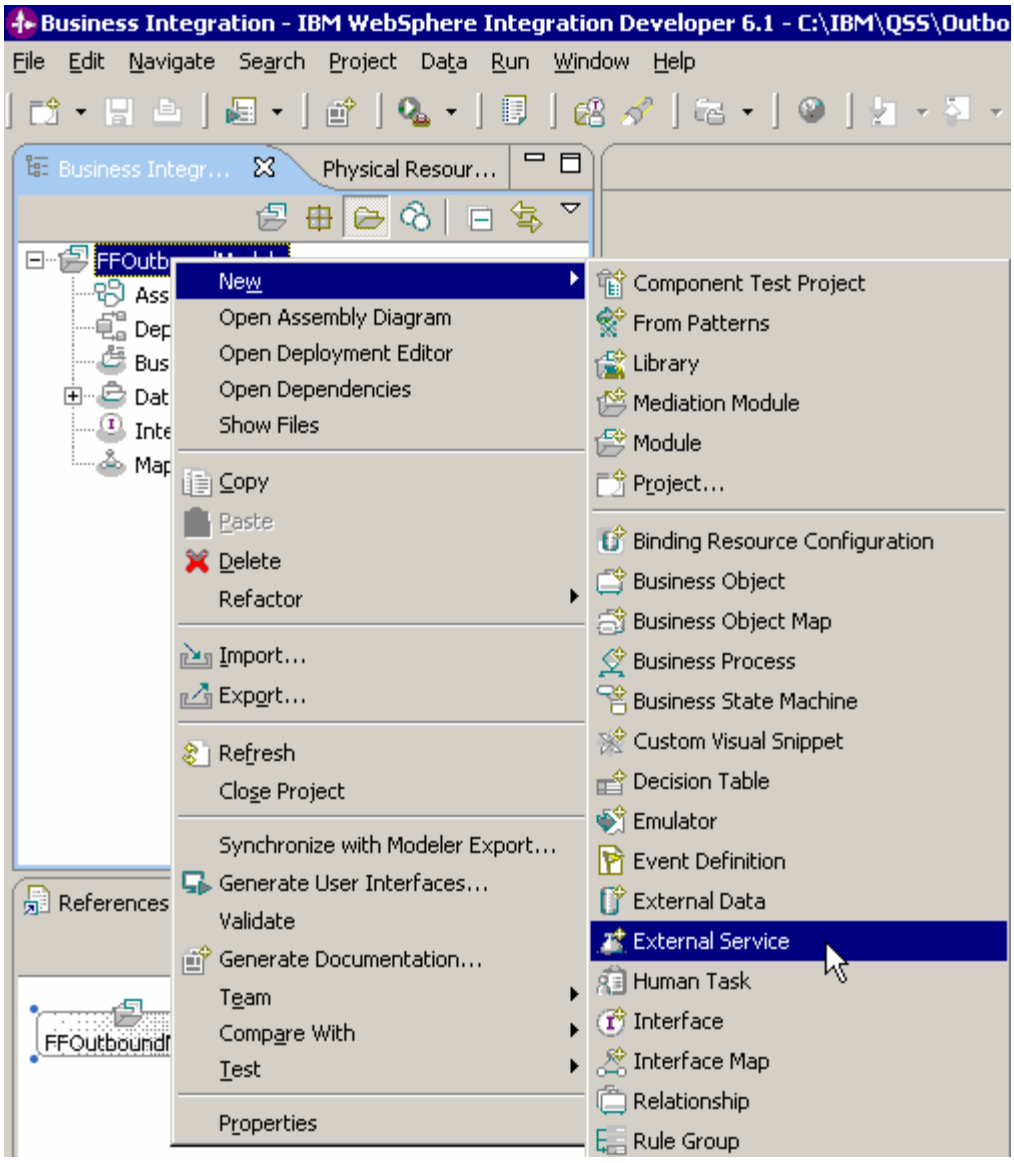
- Create a new module. Go to **File** -> **New** -> **Module**.
- Enter a desired name for your module and click **Next**. This tutorial uses FFOutboundModule as the **Module Name**.



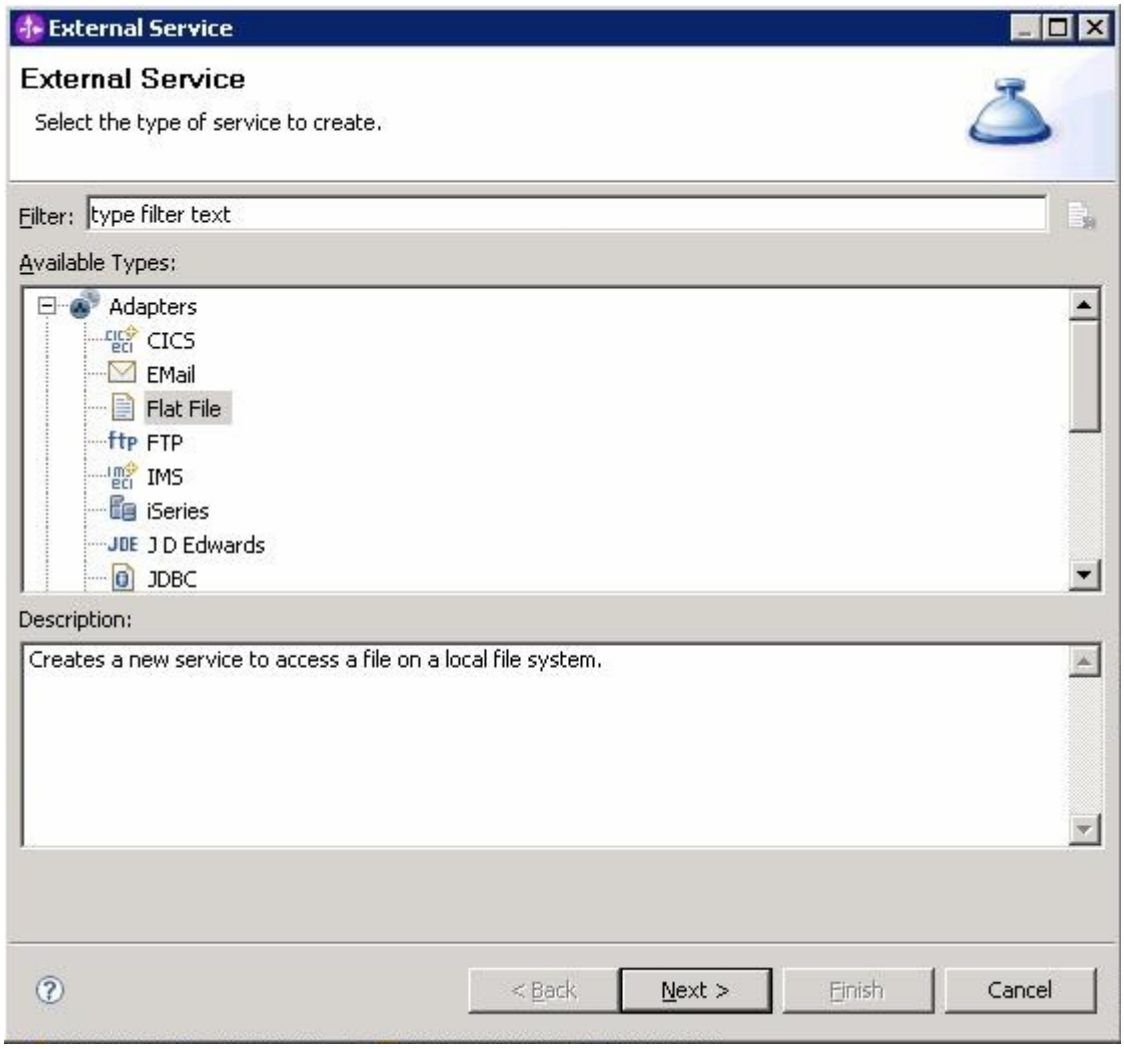
- Select **Do no include module in the session** if you do not want to have an existing solution and click **Next**.



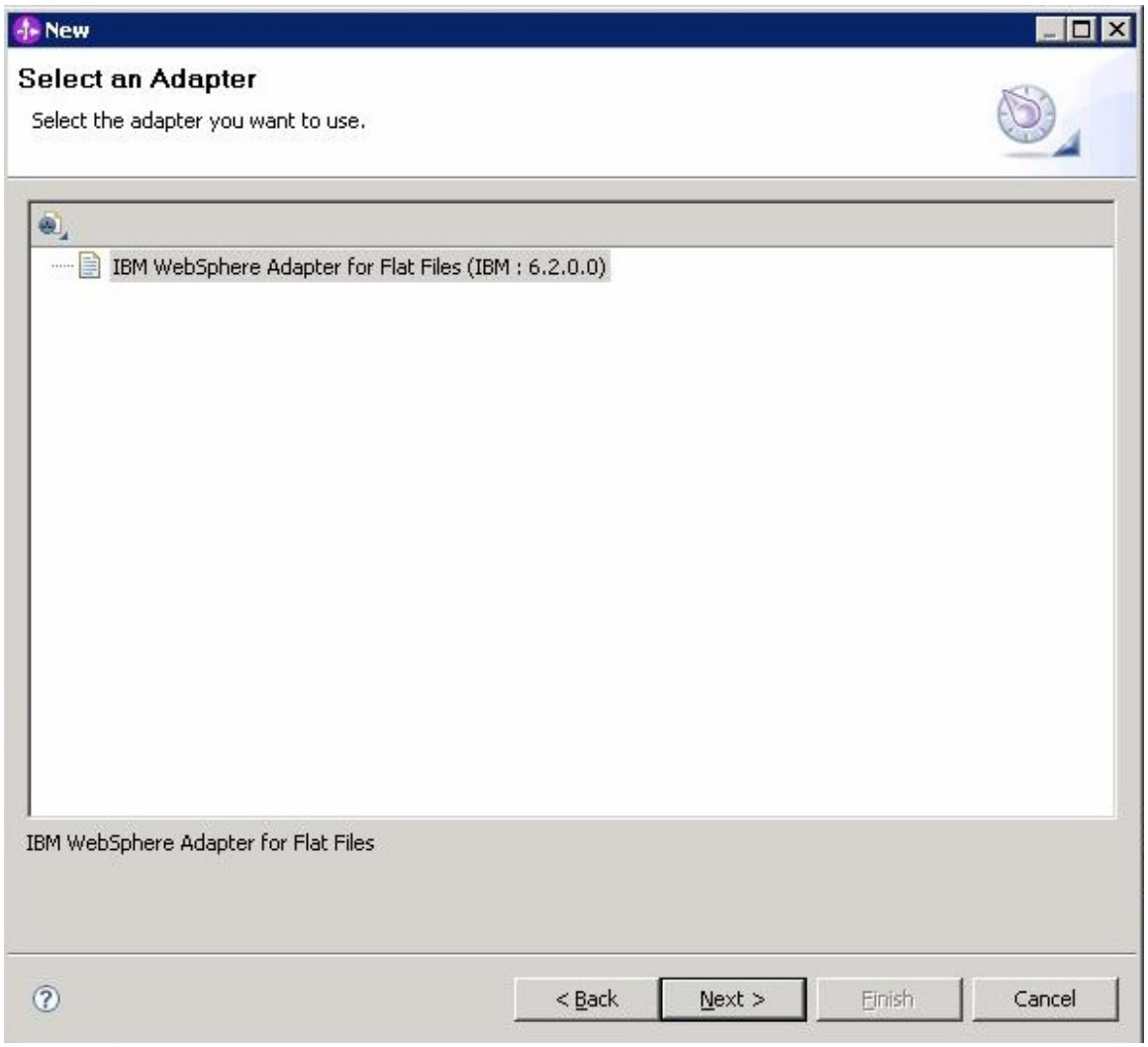
- Start the external service wizard by choosing **File** -> **New** -> **External Service**.



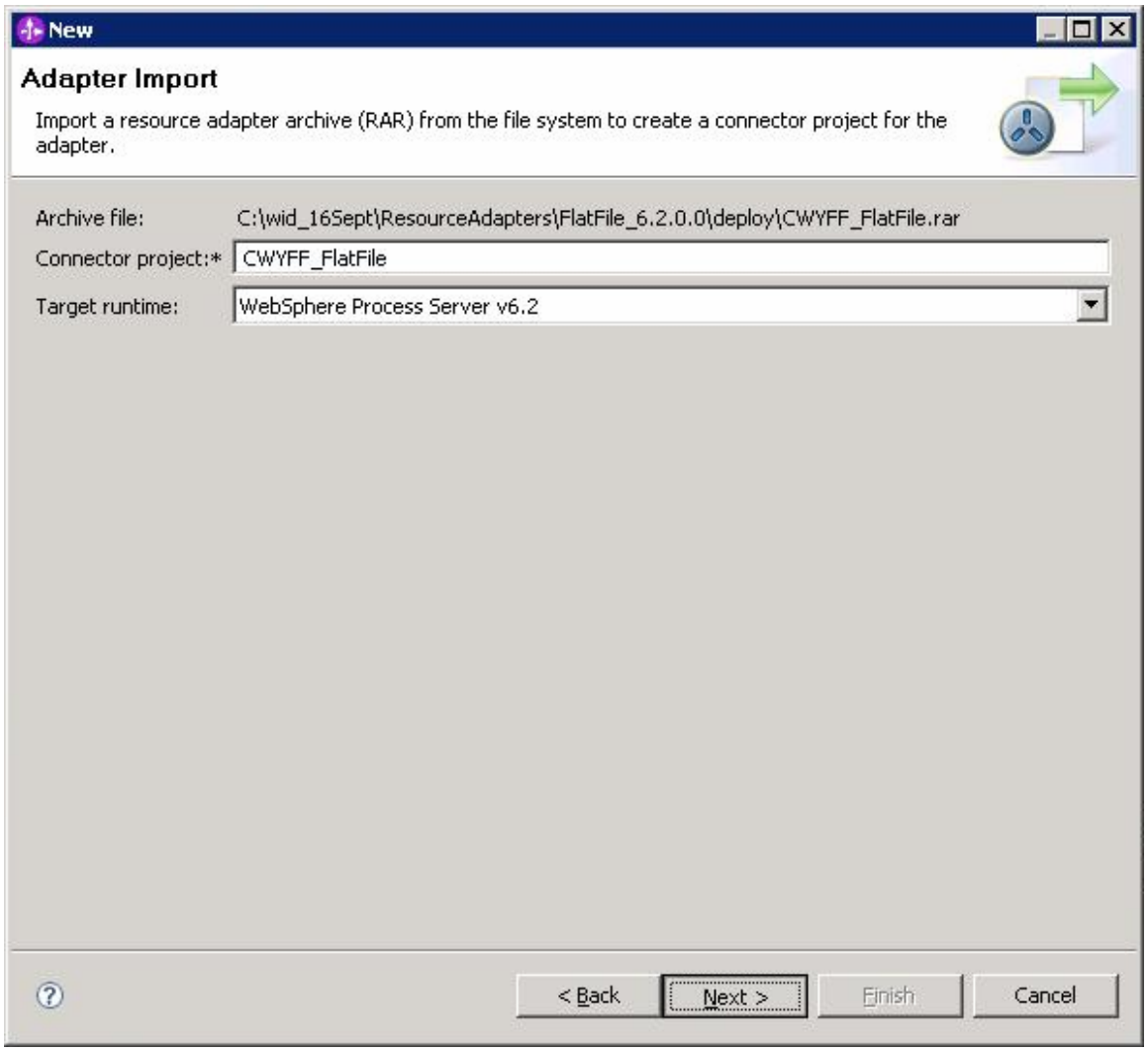
- Expand **Adapters**, select **Flat File** and then click **Next**.



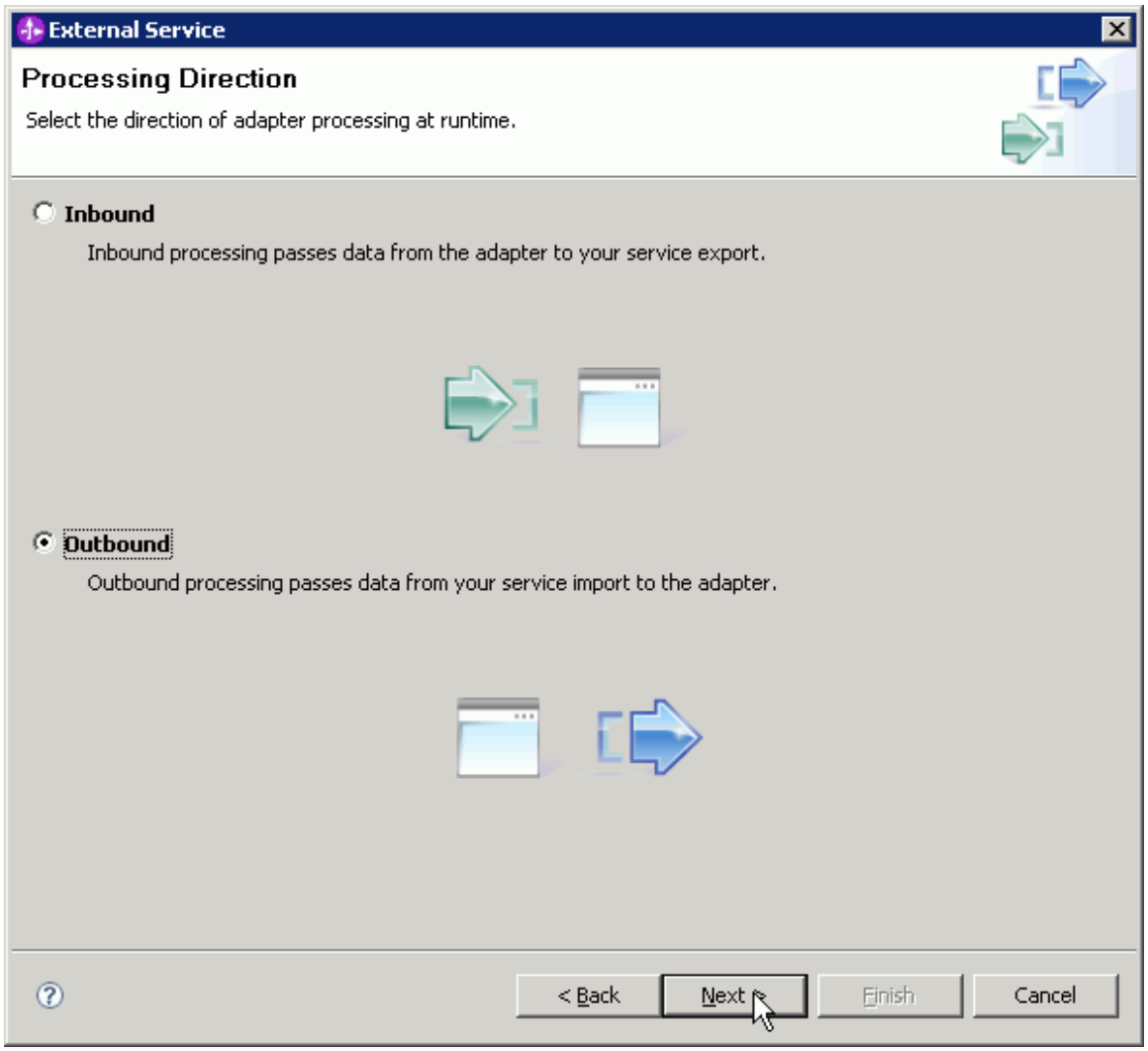
- In the Select an Adapter window, select **IBM WebSphere Adapter for FlatFiles (IBM : 6.2.0.0)**.



- In the Adapter Import window, adapter RAR file will be imported into the module. Leave the default value set in the **Connector project** field. Ensure that the **Target runtime** is set to **WebSphere Process Server v6.2**. Click **Next**.

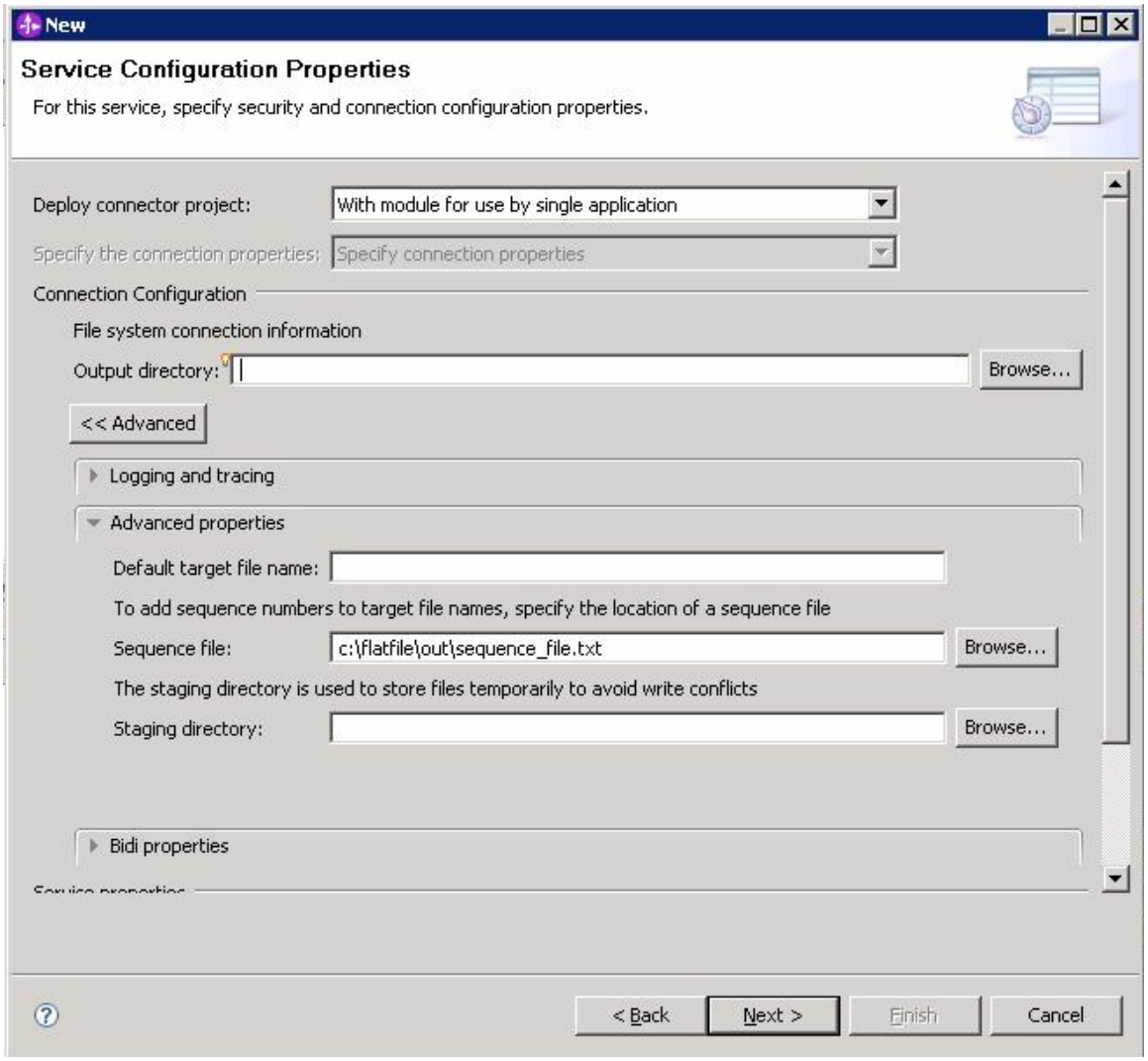


- To proceed with Create operation, choose **Outbound** in the Processing Direction window and click **Next**.

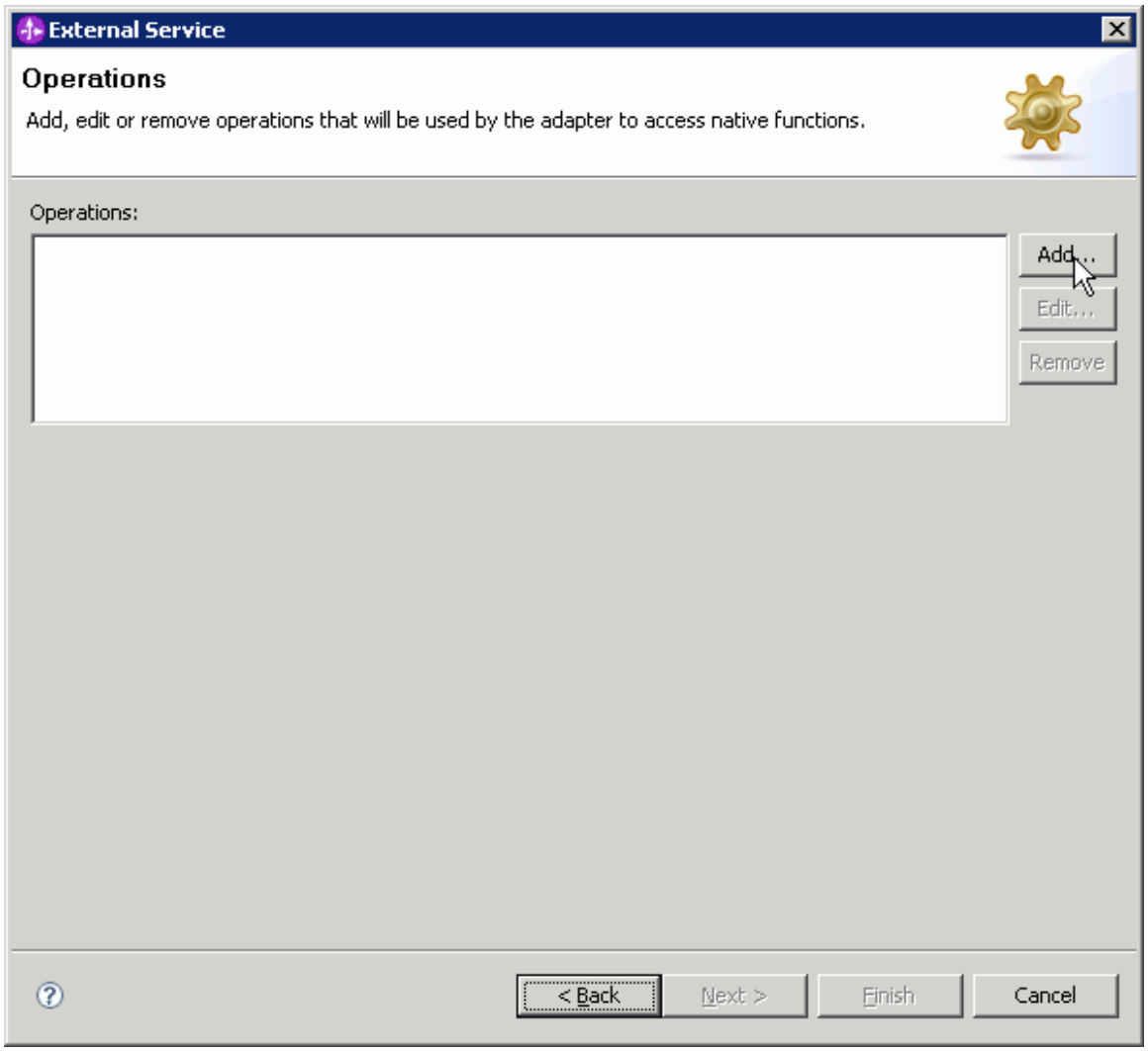


Setting properties for the external service wizard

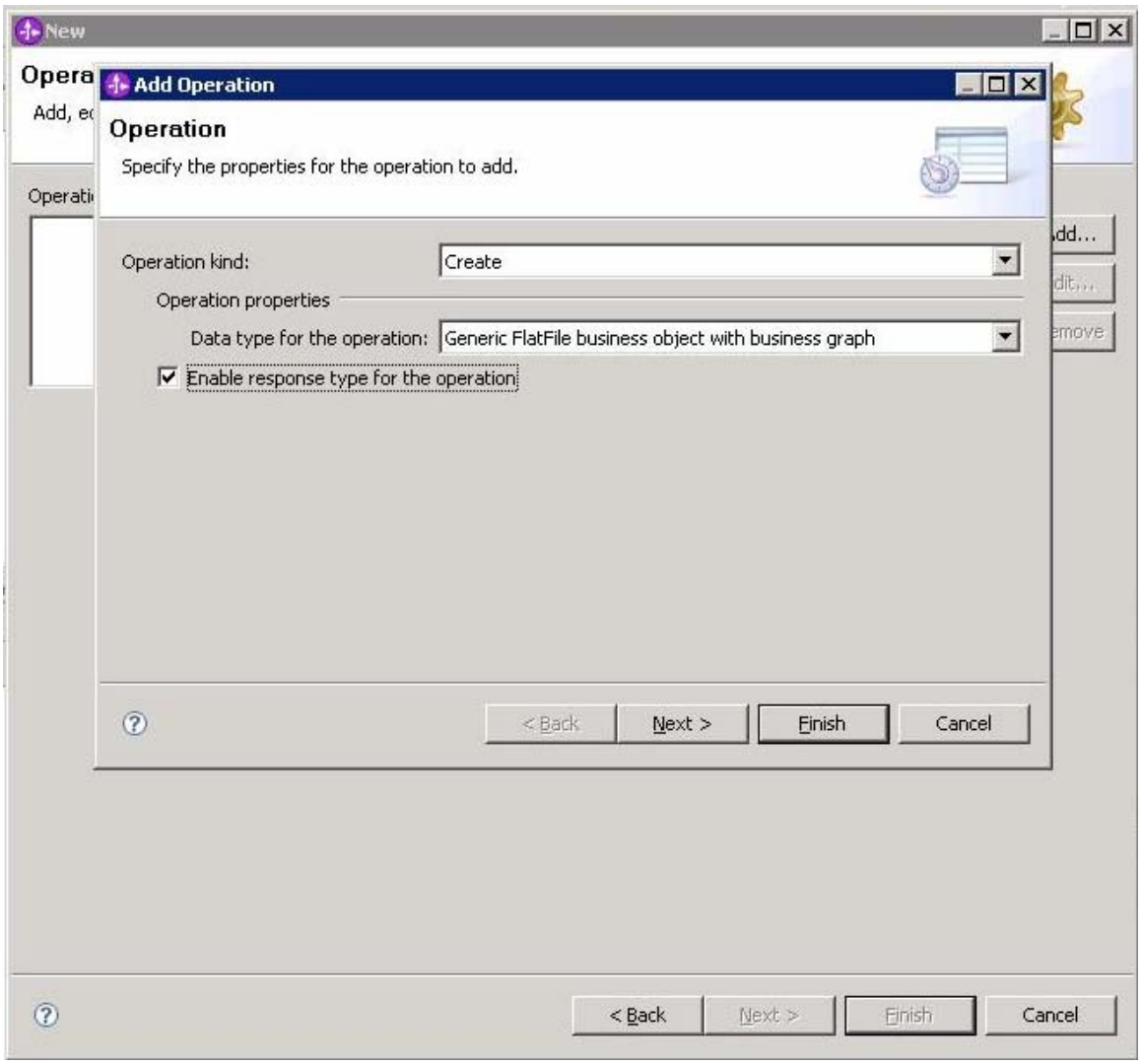
- In the screen capture shown below, specify a value for **Sequence file** property.



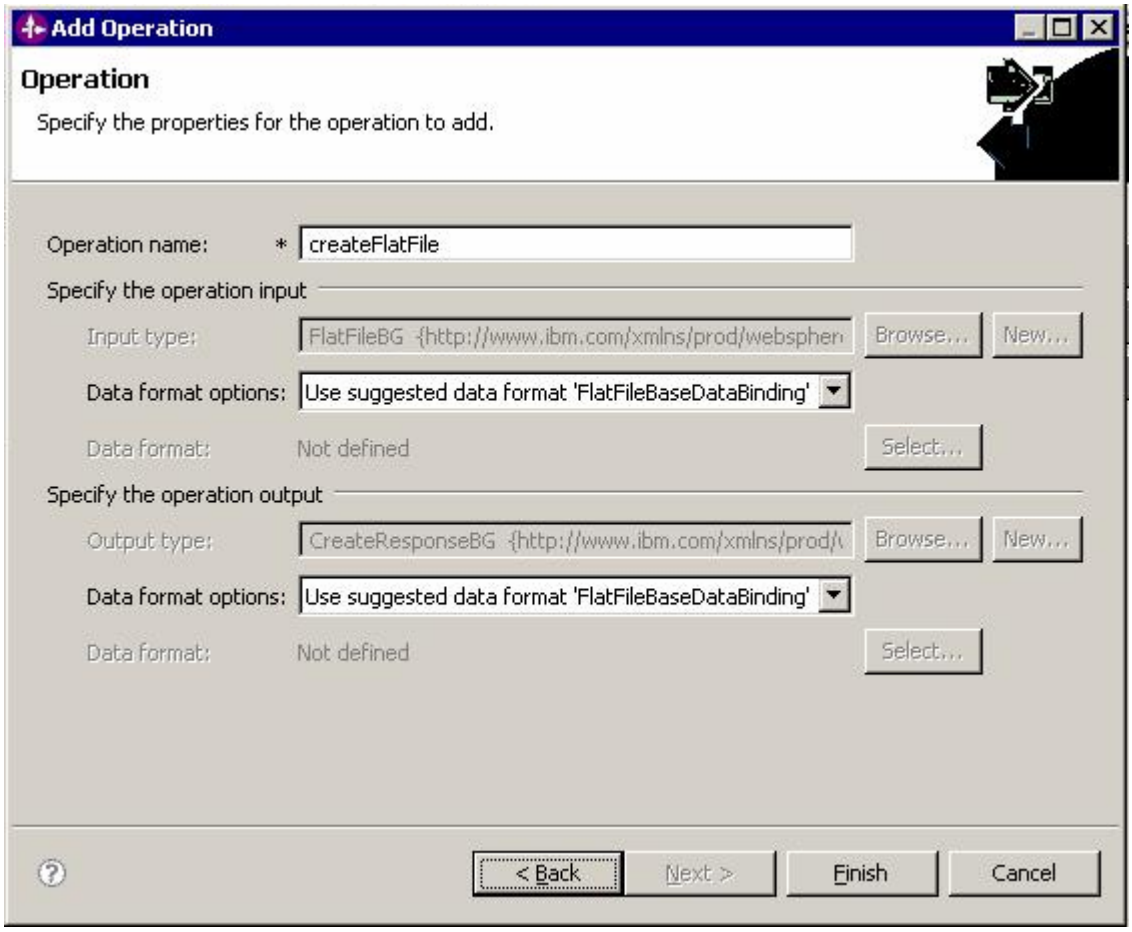
- Next, in the Operations window, click **Add** to add the operations that you want to perform.



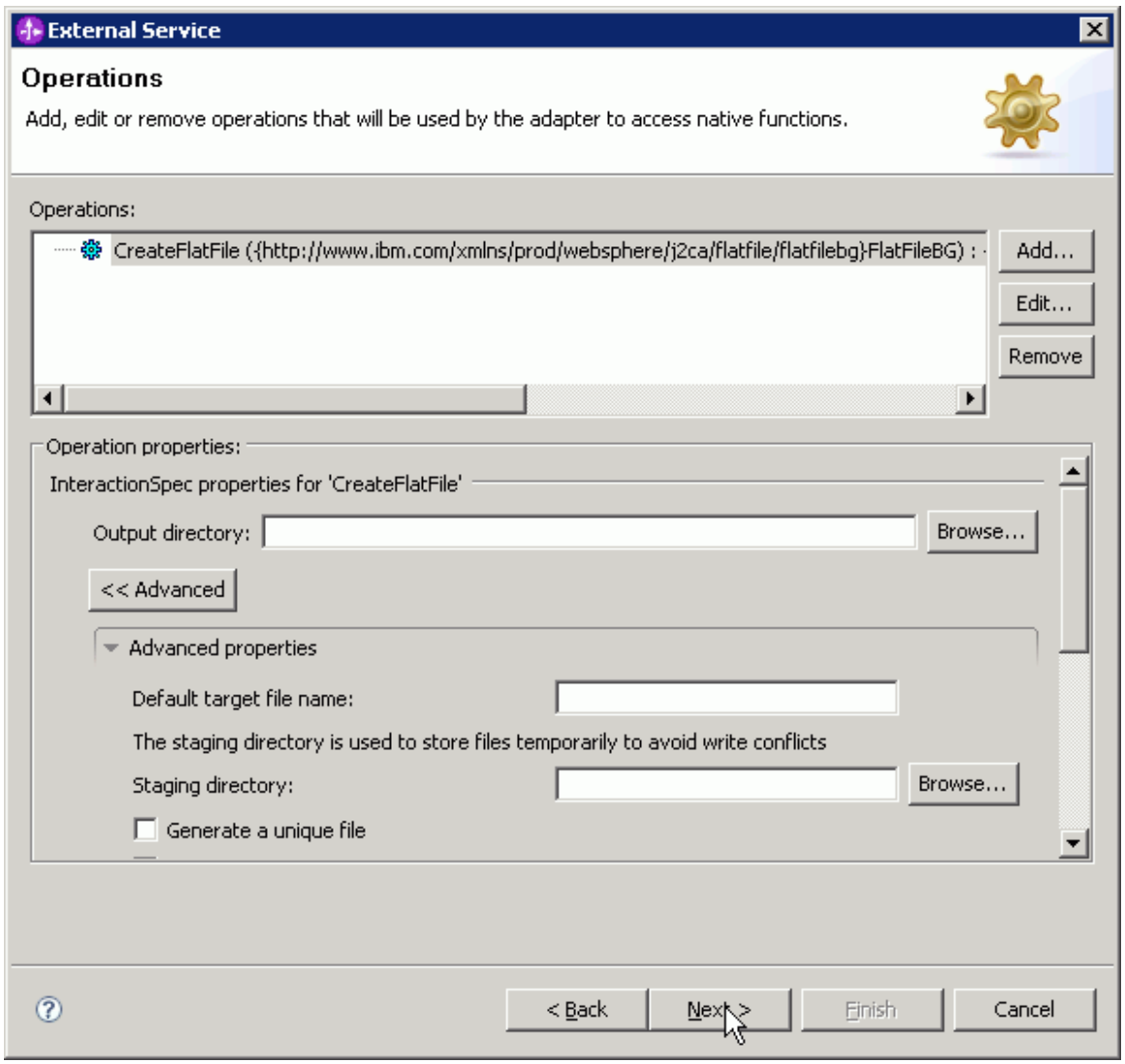
- Choose **Create** from the **Operation kind** list and **Generic FlatFile business object** from the **Data type for the operation** list. Click on **Enable response type for the operation** if a business object needs to be returned after performing a create operation and click **Next**.



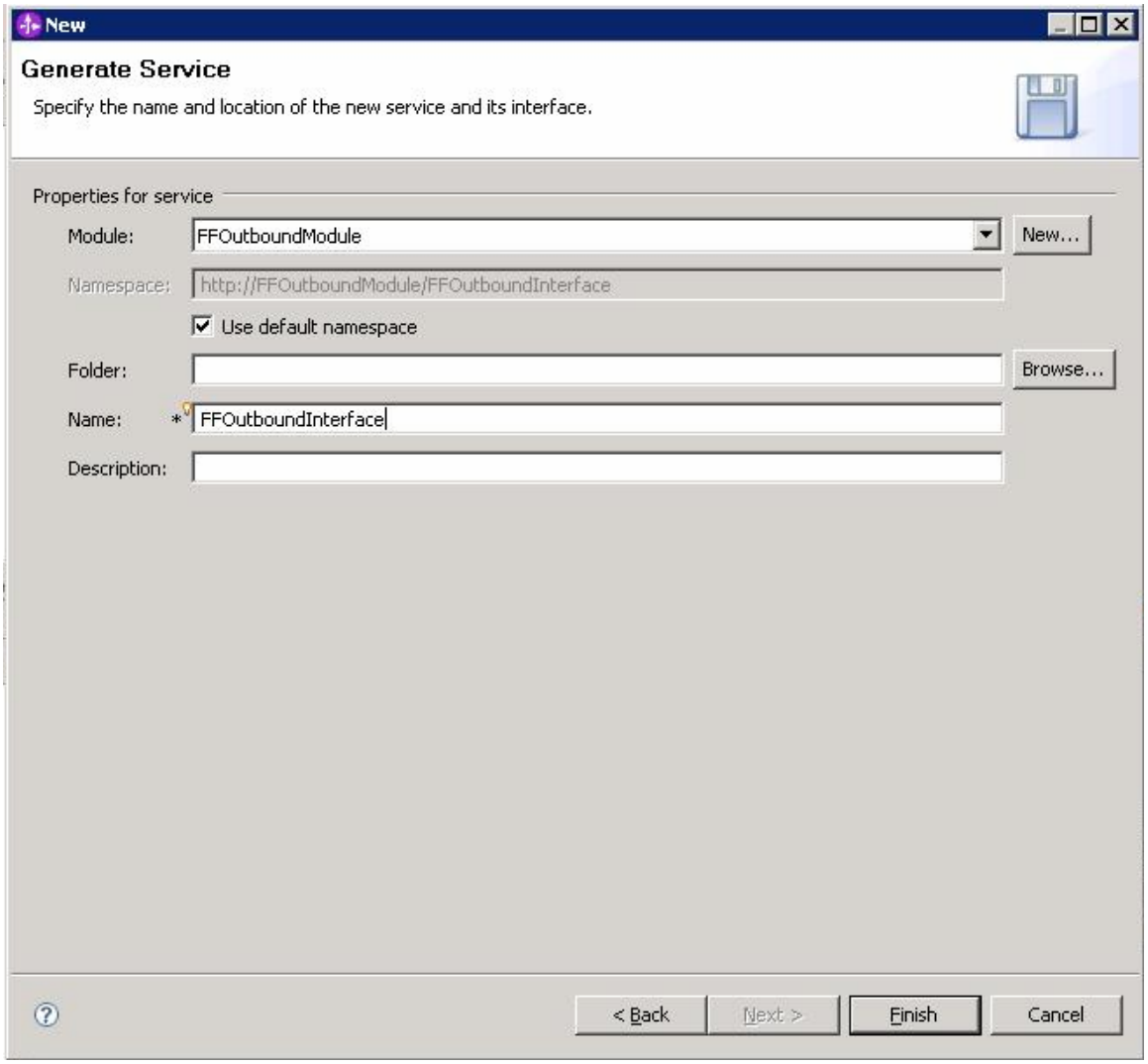
- There is no need to configure a Data handler for the Data binding being created as this is a pass-through scenario without any sort of data transformation. Hence, the default data binding configuration available is used. The **Input type** (i.e., wrapper FlatFileBG) and the **Output type** (CreateResponseBG) have been created without user intervention in case of pass-through scenario. Click **Finish**.



- In the next window, the Interaction spec properties can be specified as shown below. No values are set at the interaction spec level. All values will be set at record level in this tutorial. Click **Next**.



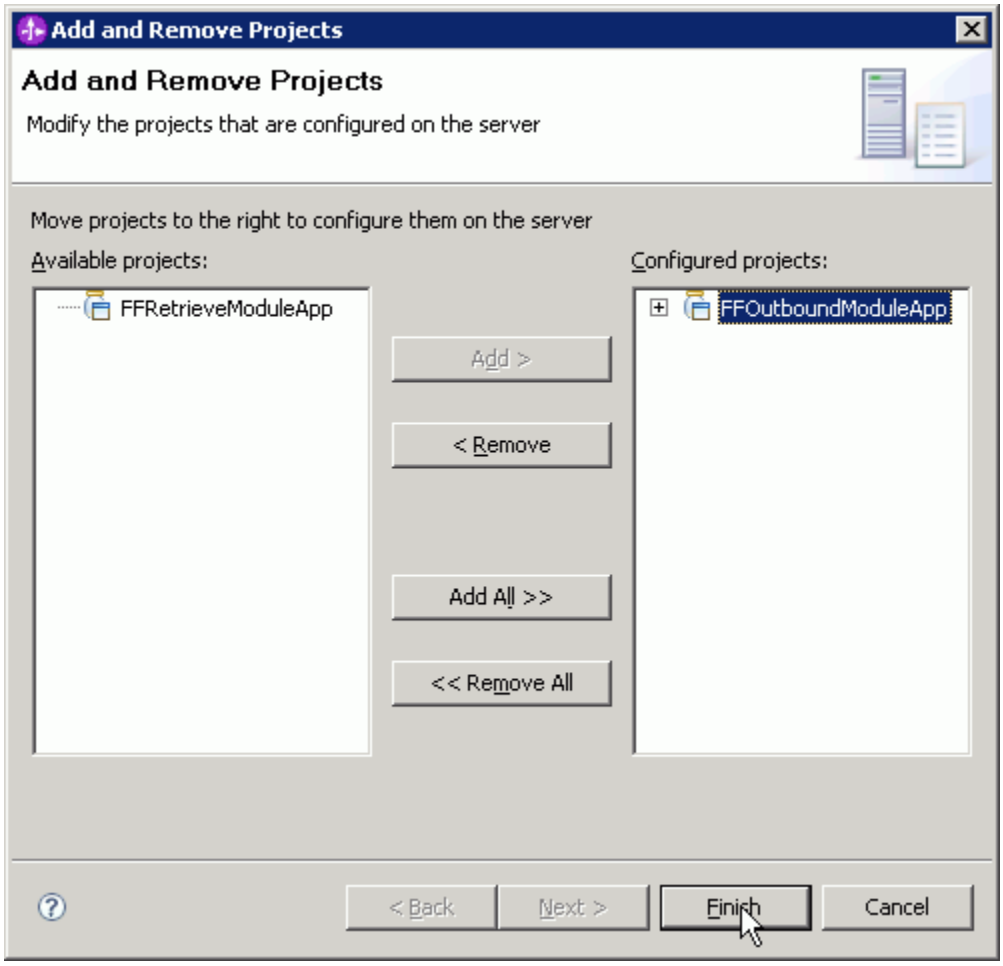
- In the last step of running external service wizard, specify a suitable name for your adapter interface. In this tutorial FFOutboundInterface is set as the value in the **Name** field. Click **Finish**.



Deploying the module to the test environment

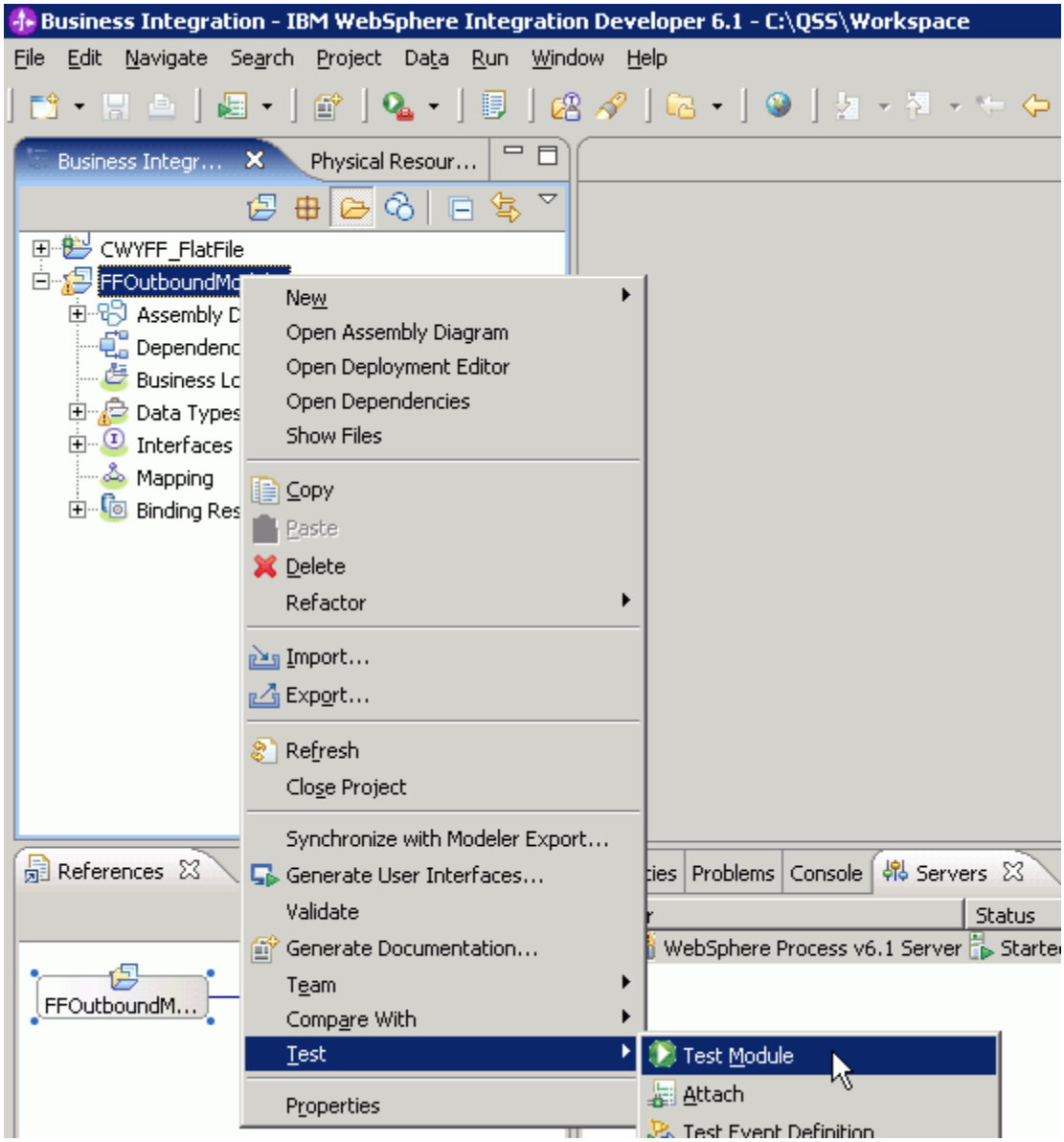
The following steps show how to perform a Retrieve operation with the module you just created.

- Start the WebSphere Process Server
- Add adapter module to the server. In the Server tab, right click on WebSphere Process Server then click **Add and Remove Projects**.
- From the **Available projects** pane, select your adapter module, click **Add >** and click **Finish**.

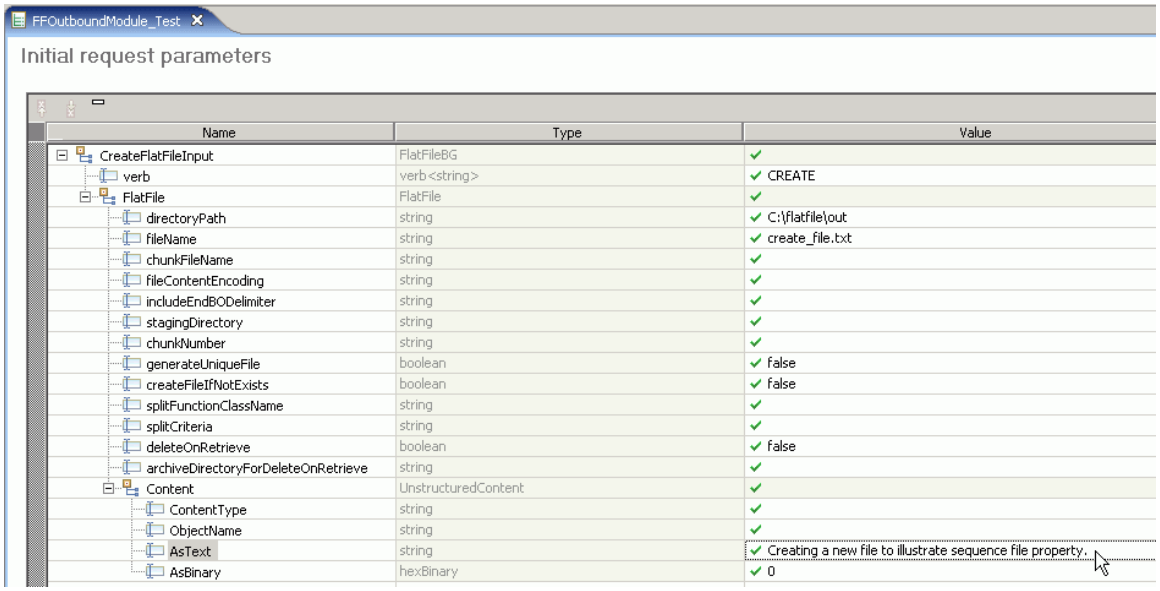


Testing the assembled adapter application

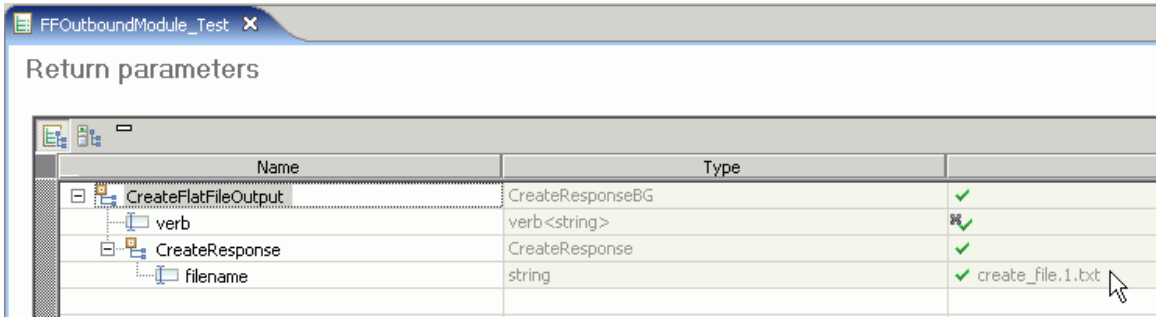
- Right click on the adapter module, FFOutboundModule then click **Test** -> **Test module**.



- Specify the required parameters to carry out Create operation, i.e., **directoryPath**, **fileName**. And content to be written to the file in **AsText** field.



- Click Invoke. The following business object which is the output of Create operation will be displayed. It contains a filename field which carries the value of the name of the file created during Create operation.



Chapter 5. Tutorial 3: Outbound processing – Demonstration of Append operation with XML data transformation

This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.2.0.0 can be used to append content to an existing file while performing data transformation using XML Data handler. Essentially this tutorial will guide us through configuration of Data handler for a Data binding. Though only Append operation will be illustrated here, the same is applicable to Create, Overwrite operations as well.

Configuring the adapter for outbound processing

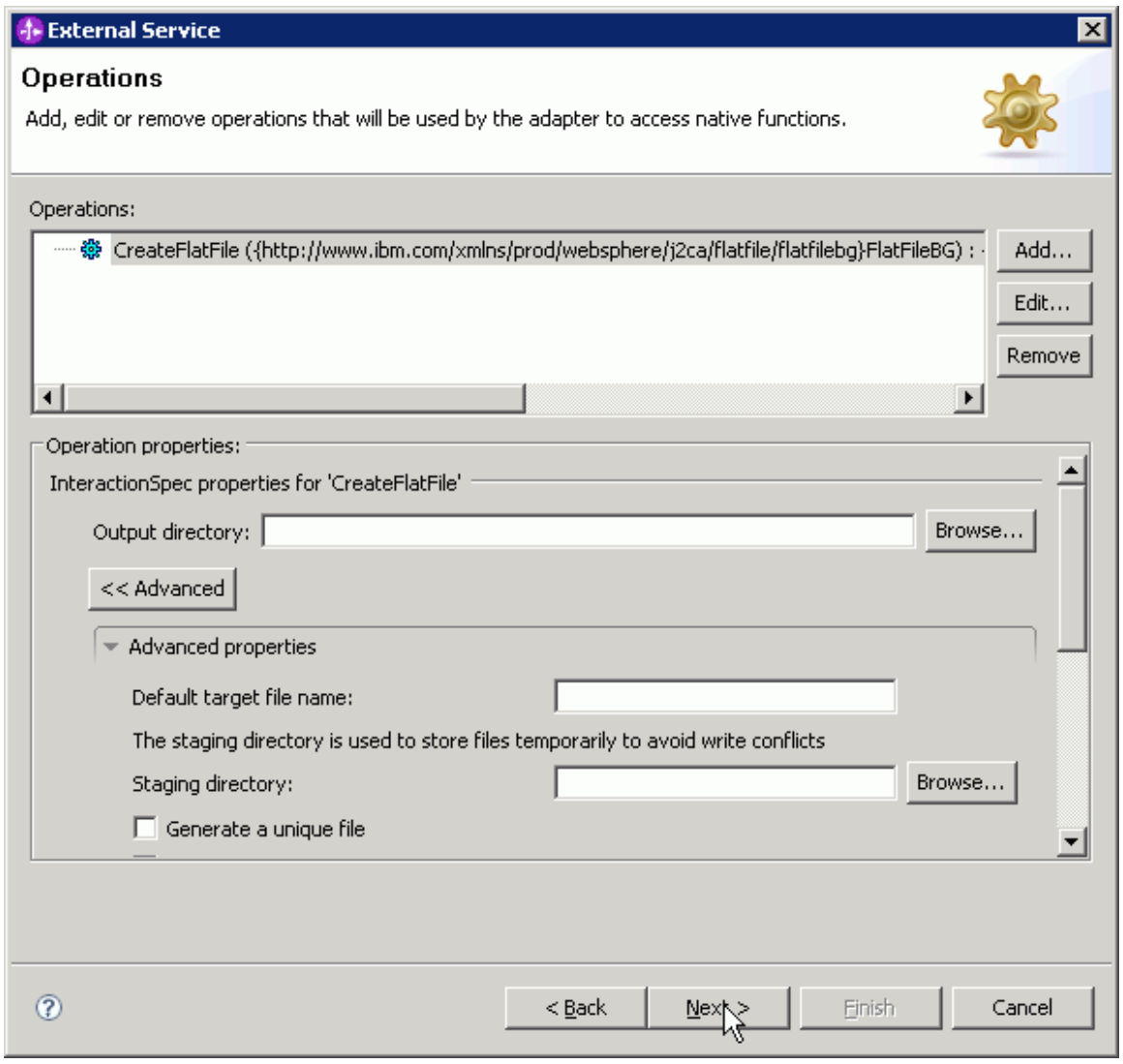
Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial. Follow the steps given in Configuring the adapter for outbound processing in tutorial 2 to proceed with outbound Append operation.

Also after creation of the module import your XSD into the module. File -> Import -> File system -> Browse to the location of your XSD(s) -> Select -> Finish. XSDs will be imported into Data type view as seen in Business integration perspective of WebSphere Integration Developer.

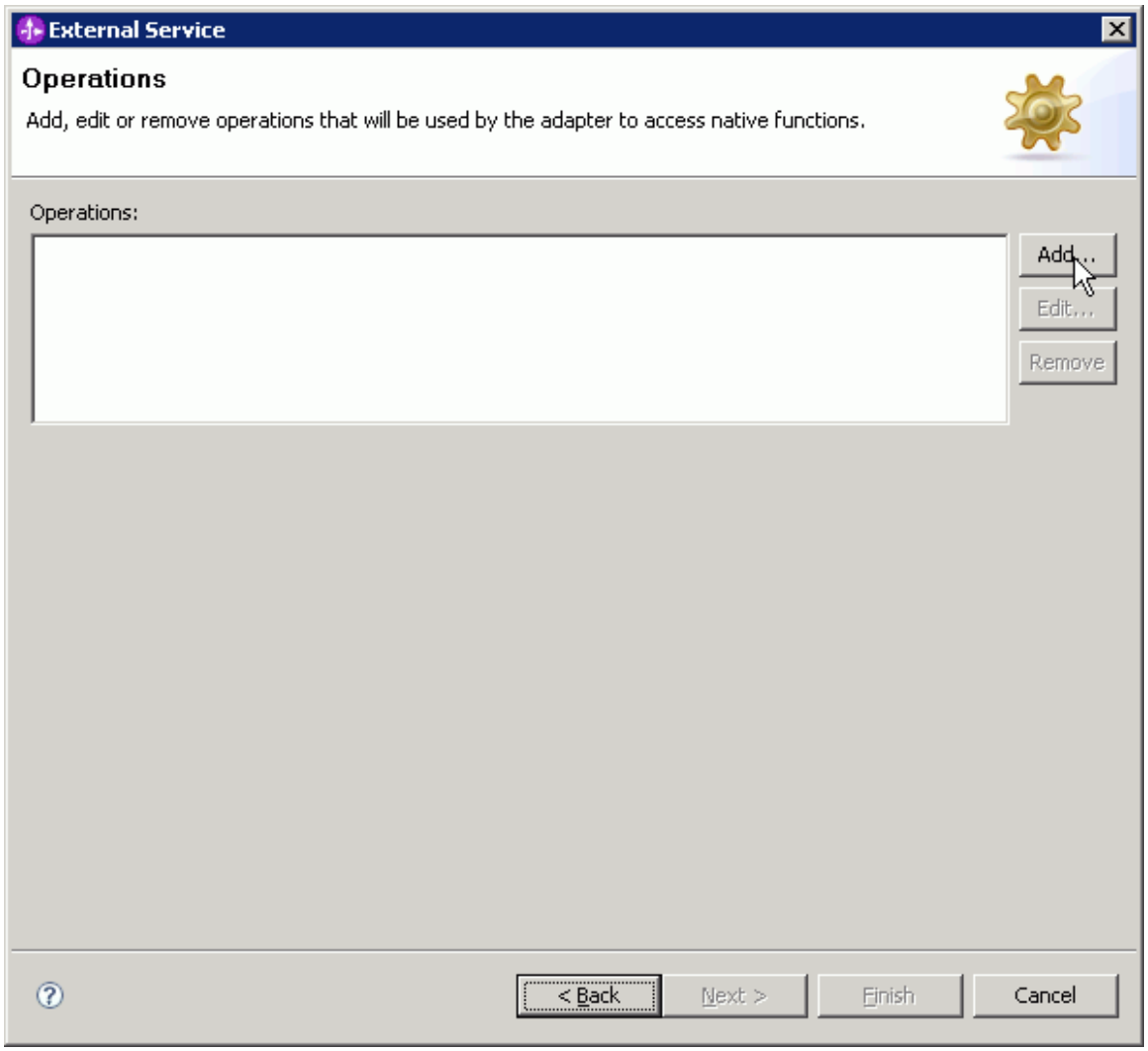
Setting properties for the external service wizard

Since you already created an outbound module for Create operation in tutorial 2, let us add another operation, i.e. Append to the same existing module (FFOutboundModule).

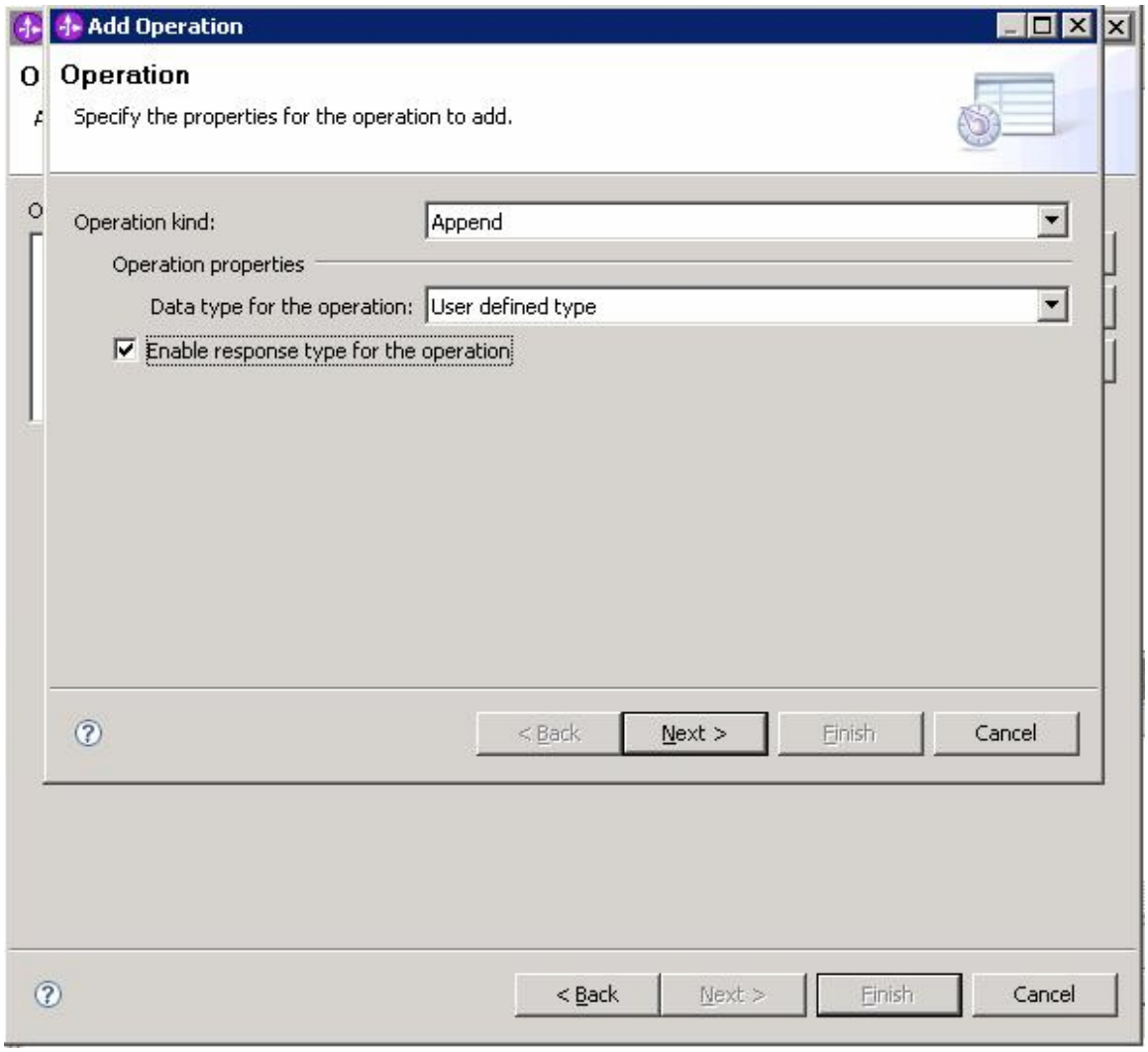
- To add another operation to an existing module, after reaching the window where interaction spec properties are displayed, click **Add**.



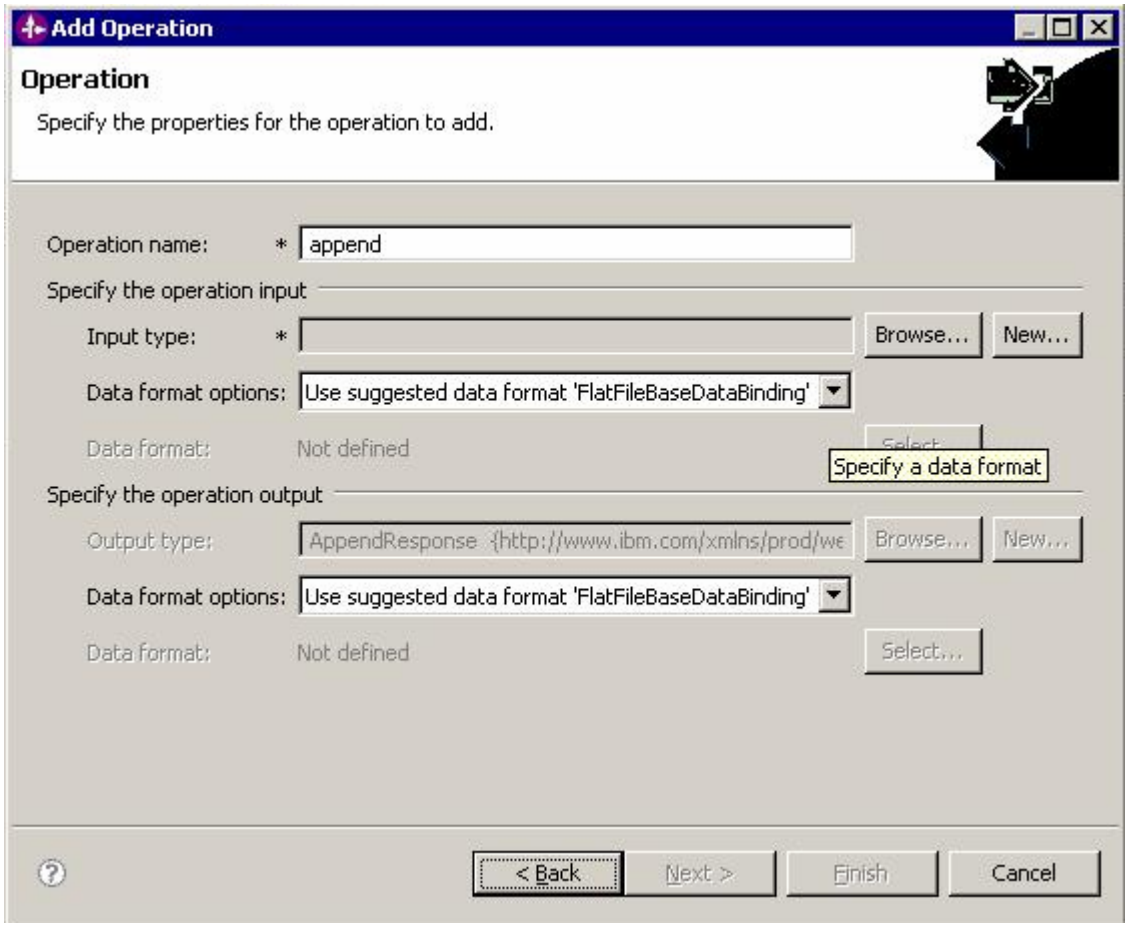
- Next, in the Operations window, click **Add** to add the operations that you want to perform.



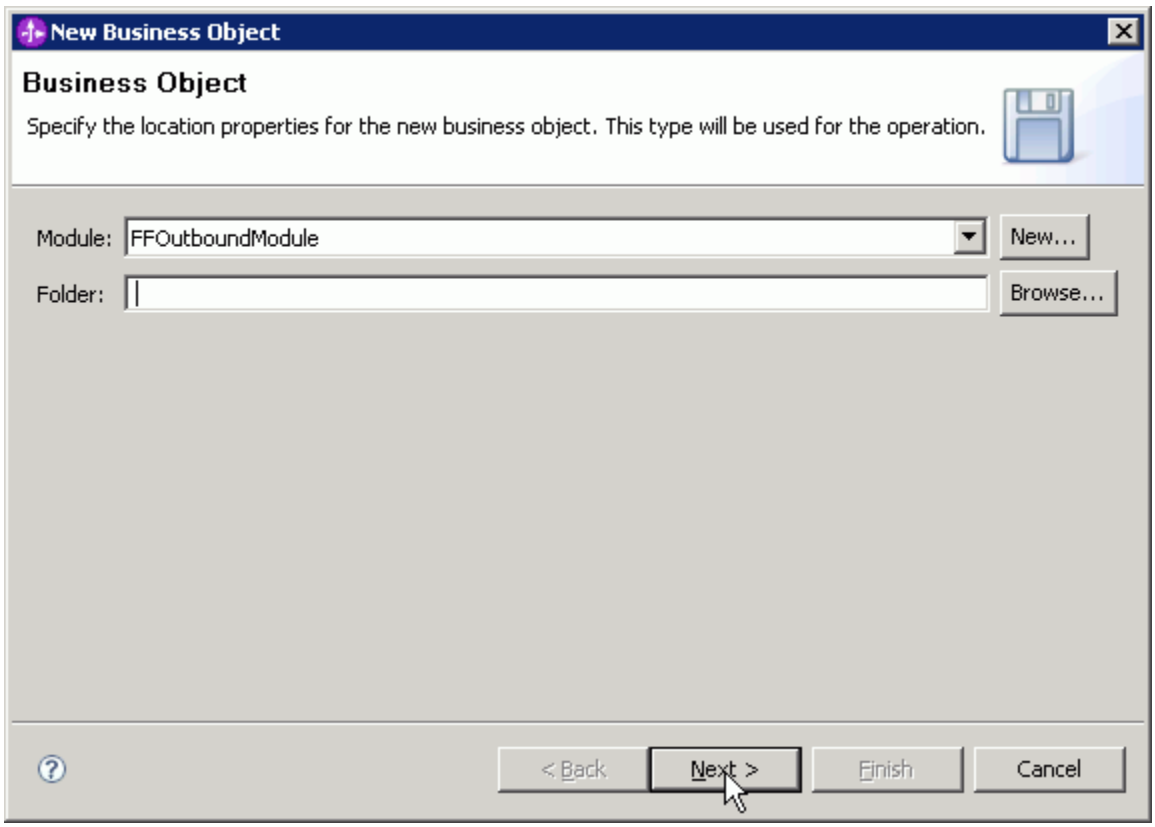
- Choose **Append** in the **Operation kind** list box and **User defined type** (for data-transformation scenario) in **Data type for the operation**. Select the **Enable response type for the operation** if a business object needs to be returned after performing a create operation and click **Next**.



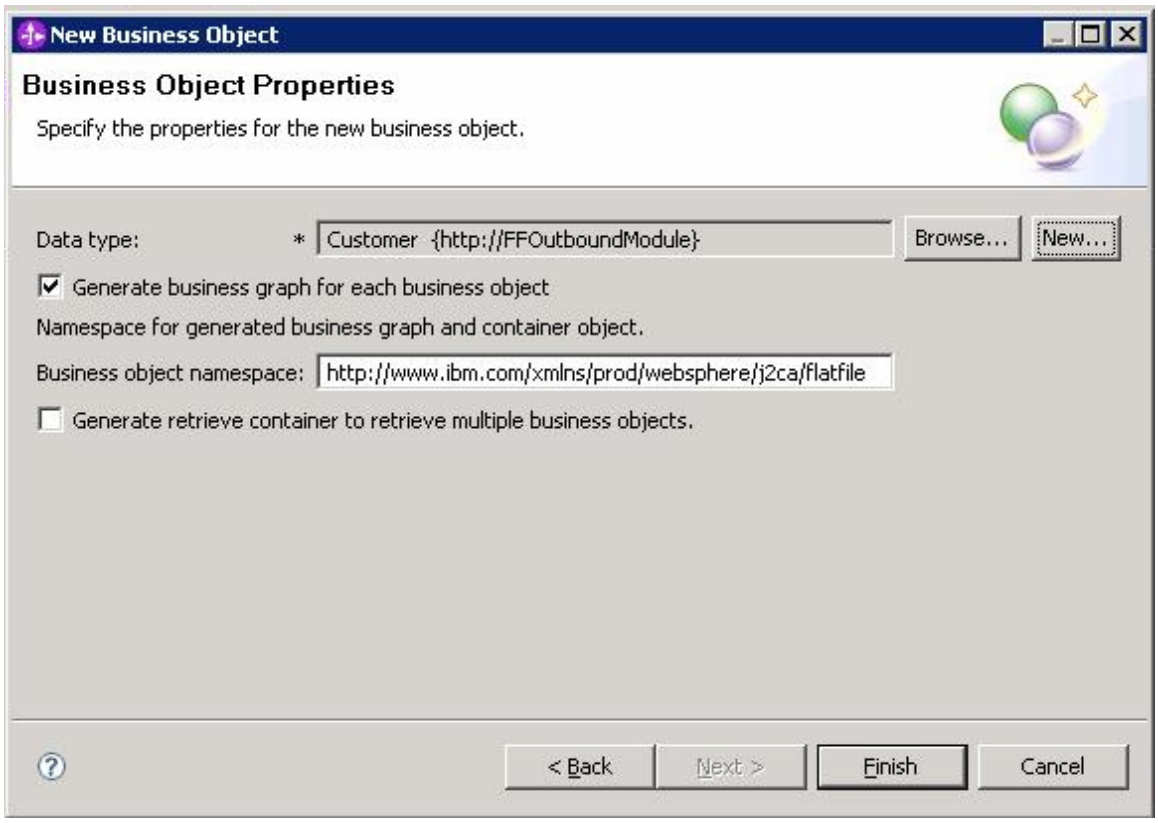
- Next configure an input type. In this example, select wrapper business graph as input type. Click **New** in **Input type** label.



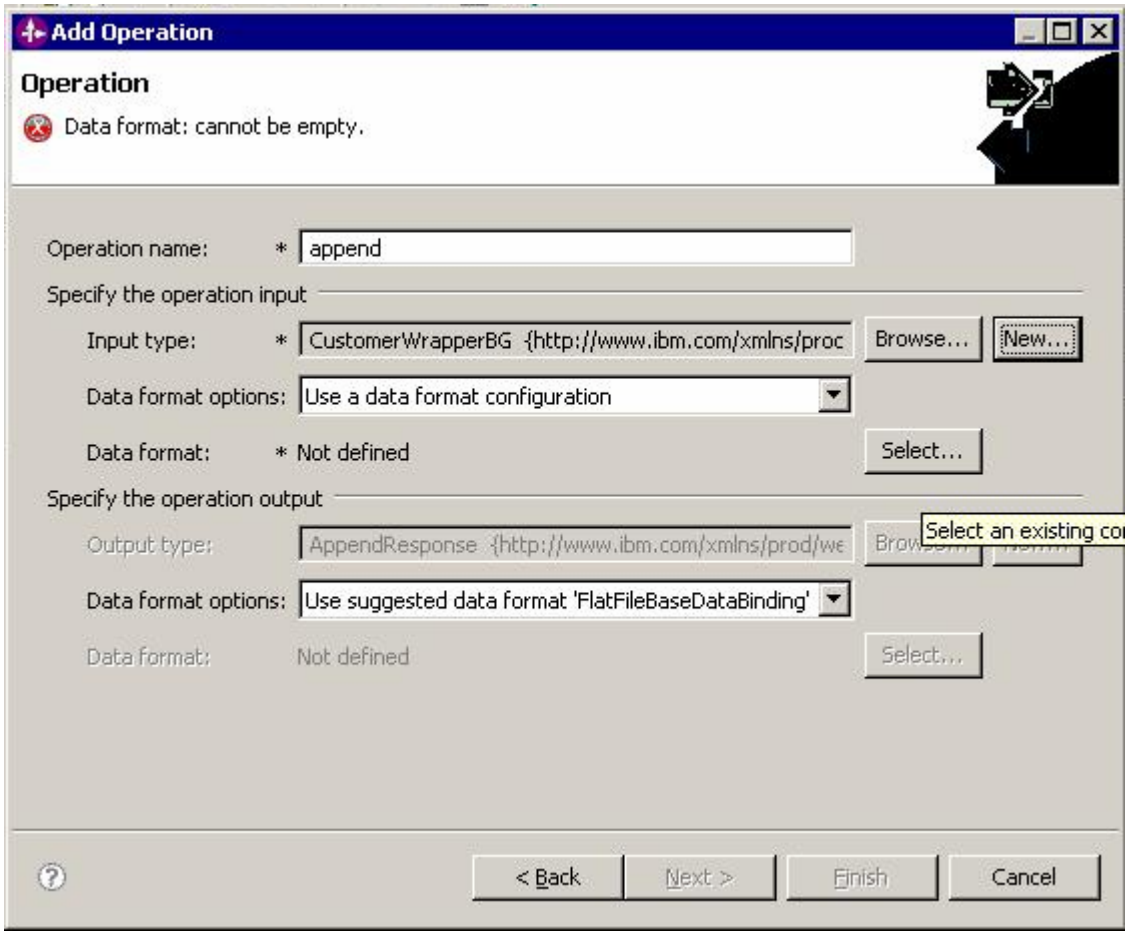
- In the following window, leave default values for **Module** and **Folder** fields and click **Next**.



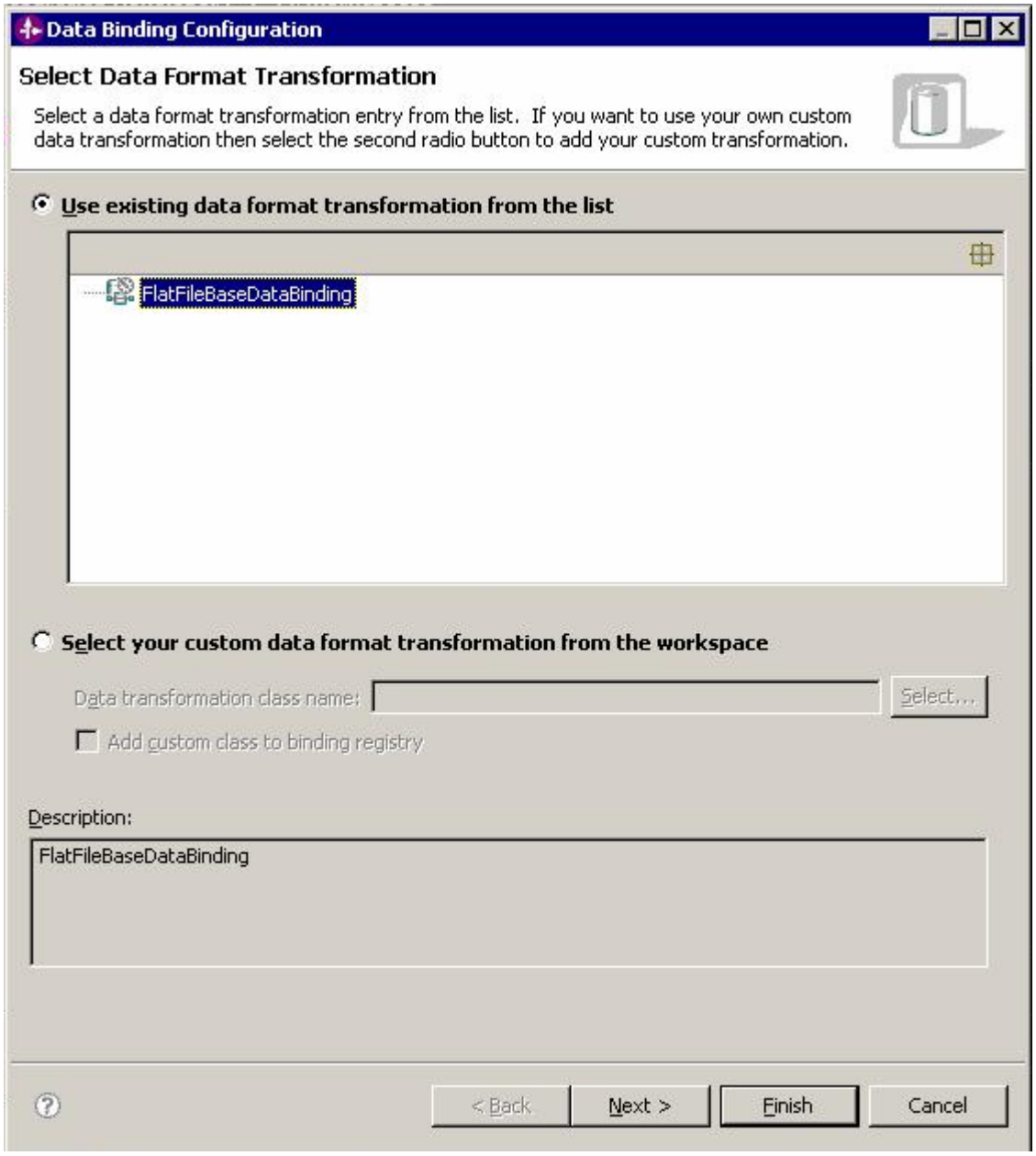
- **Browse** and choose Customer business object imported earlier on into the module (FFOutboundModule). Check on **Generate business graph for each business object** to generate wrapper (CustomerWrapperBG). Click **Finish**.

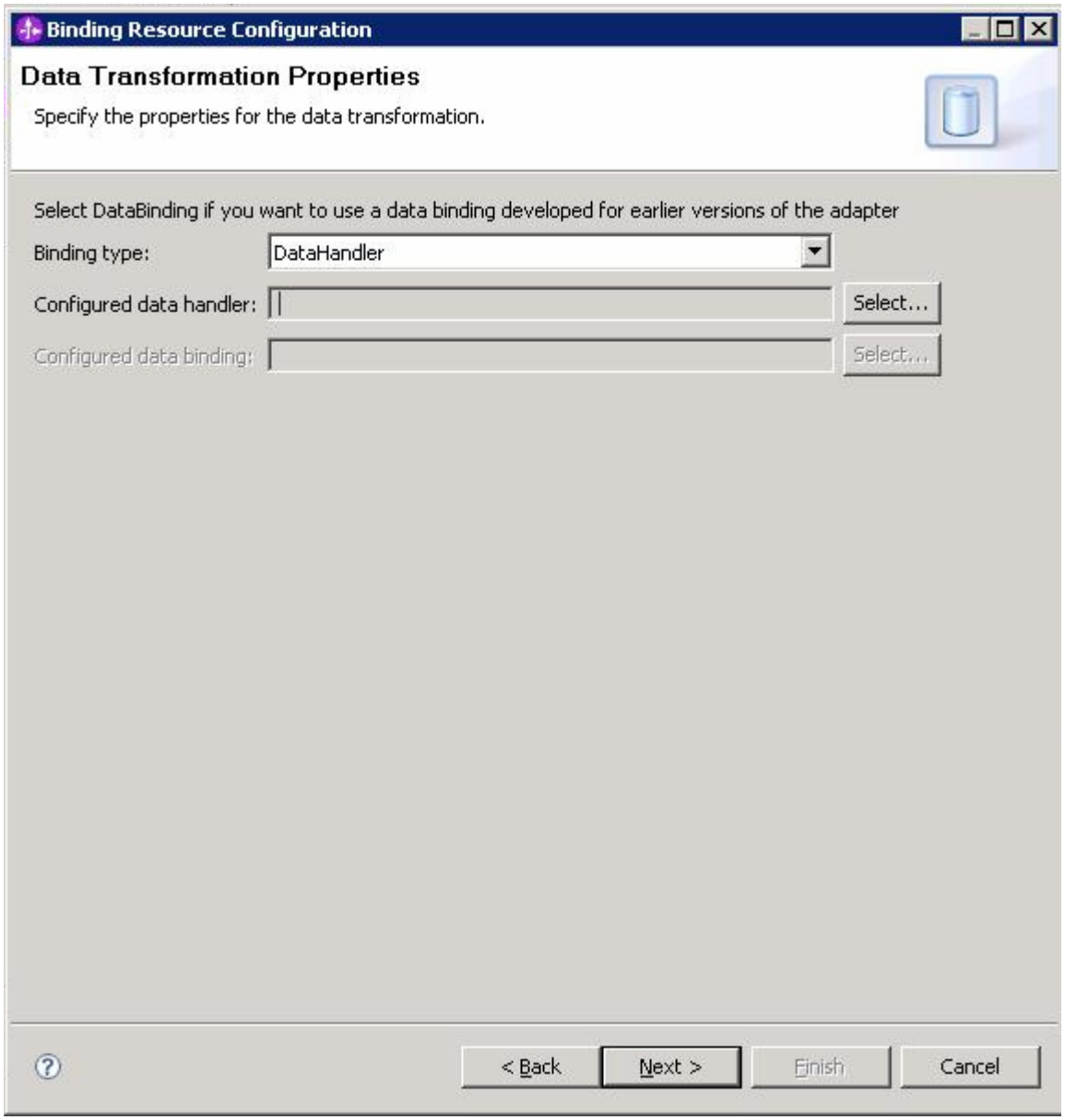


- Next, create Data binding configuration. Choose **Use a data binding configuration** in **Data binding** list box. Click **Select** alongside **Data binding configuration** label.



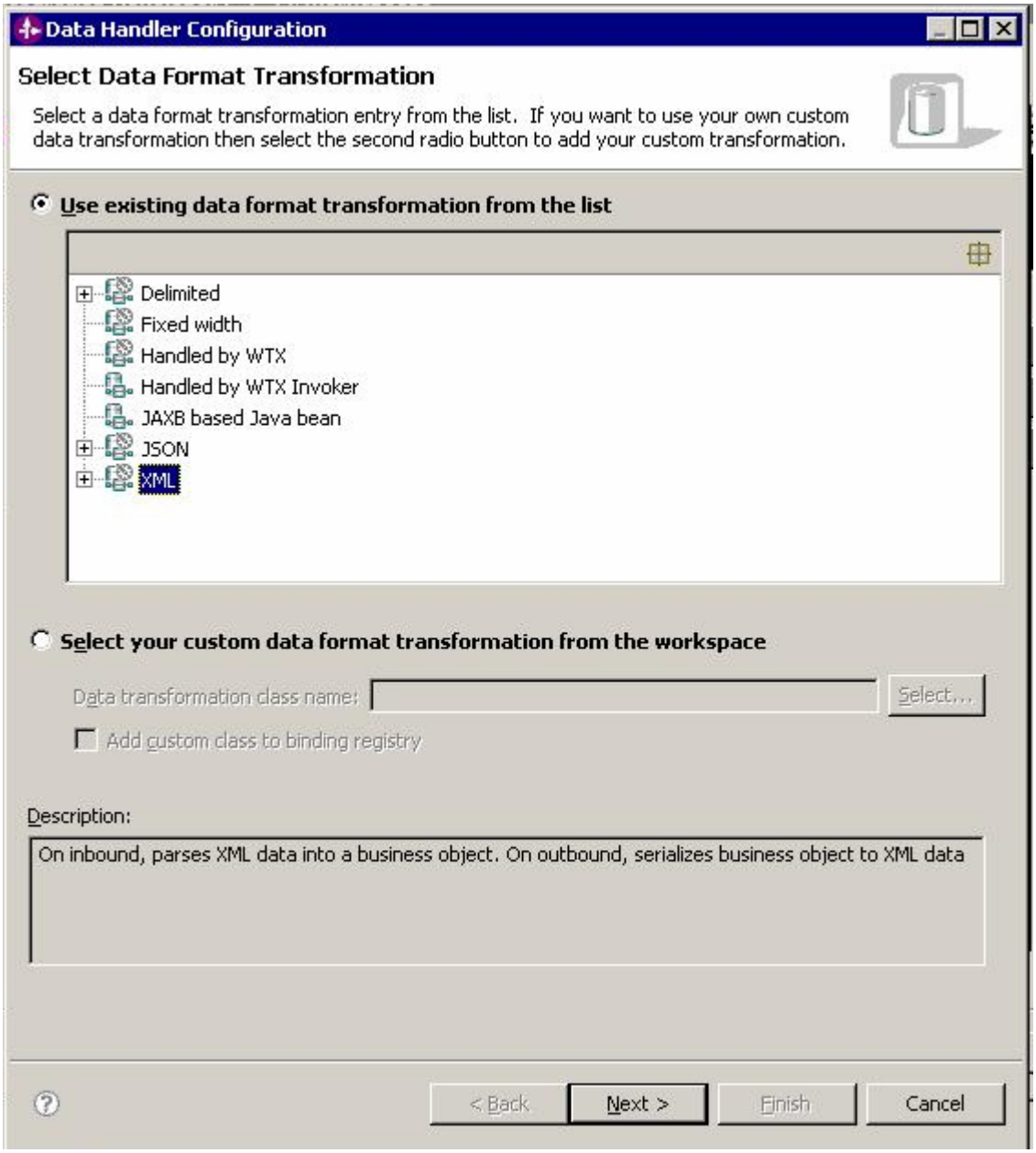
- Select **FlatFileBaseDataBinding** configuration and Click **Next**.


Data Binding Configuration
Select Data Format Transformation
Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.
Use existing data format transformation from the list
FlatFileBaseDataBinding
Select your custom data format transformation from the workspace
Data transformation class name:
Select...
Add custom class to binding registry
Description:
FlatFileBaseDataBinding
< Back
Next >
Finish
Cancel

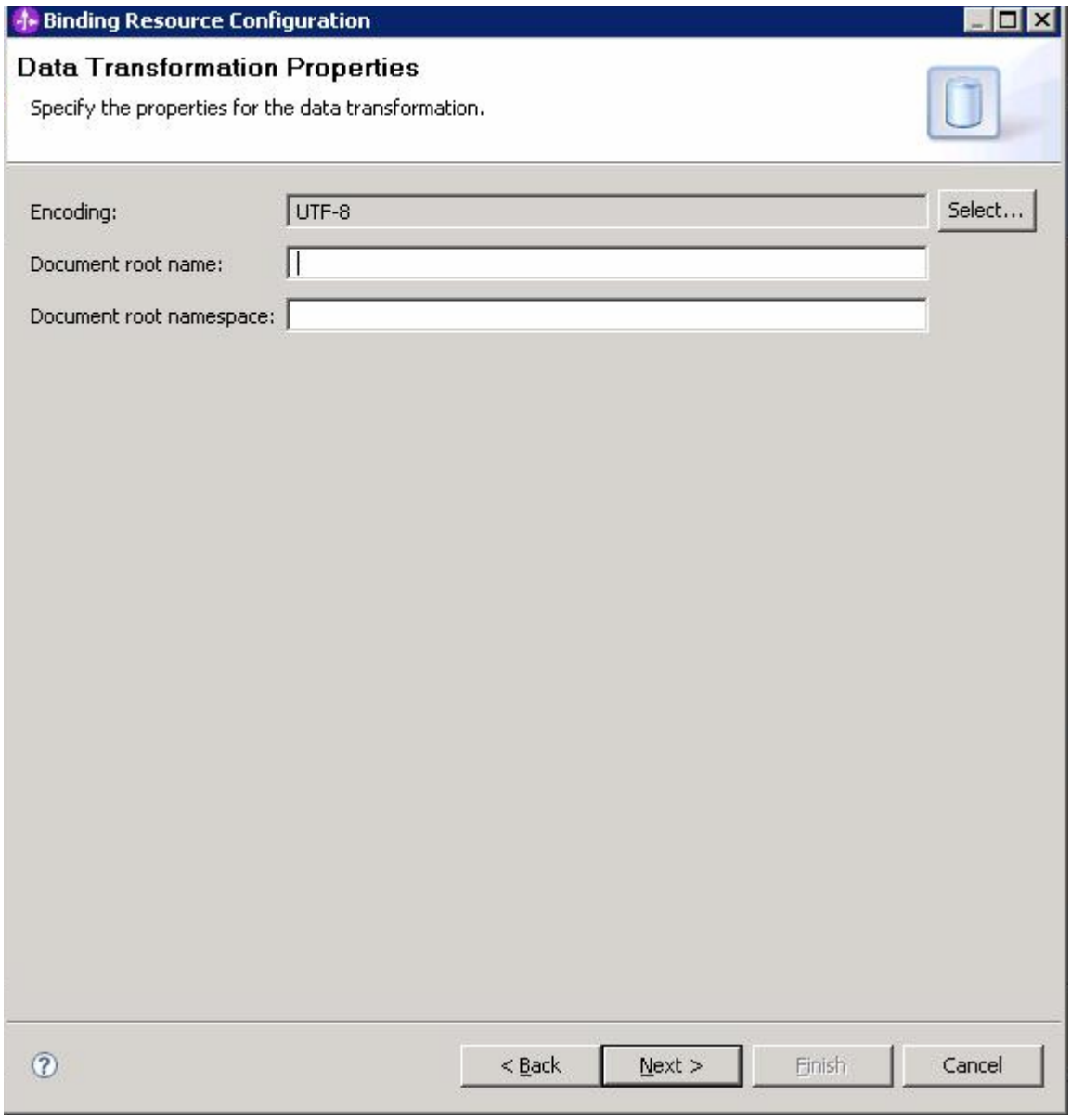


- For the Data binding being created, to configure a Data handler, click **Select** alongside **Configured data handler** label.

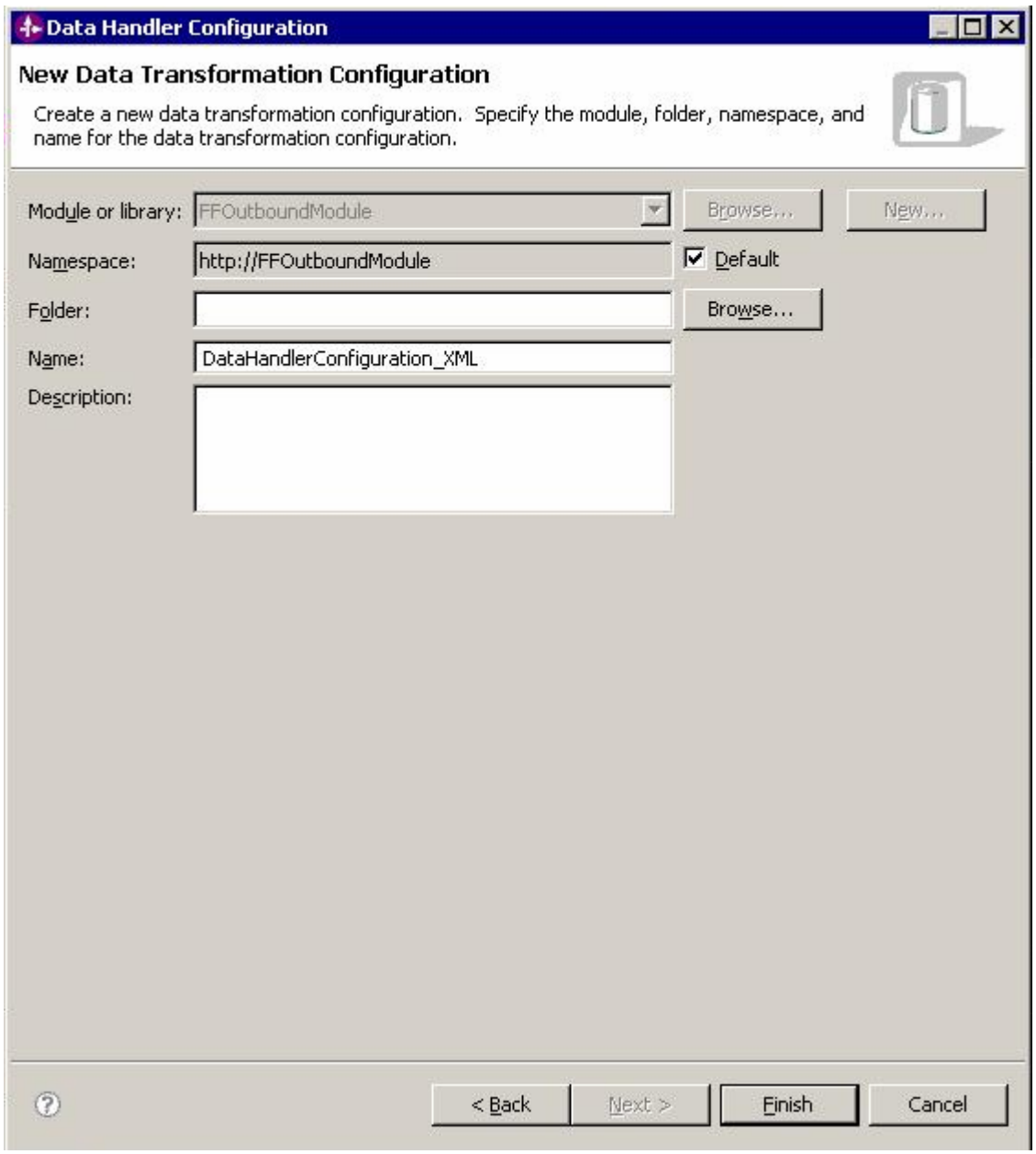
- In the Select data format transformation window, select the **XML** and then click **Next**.



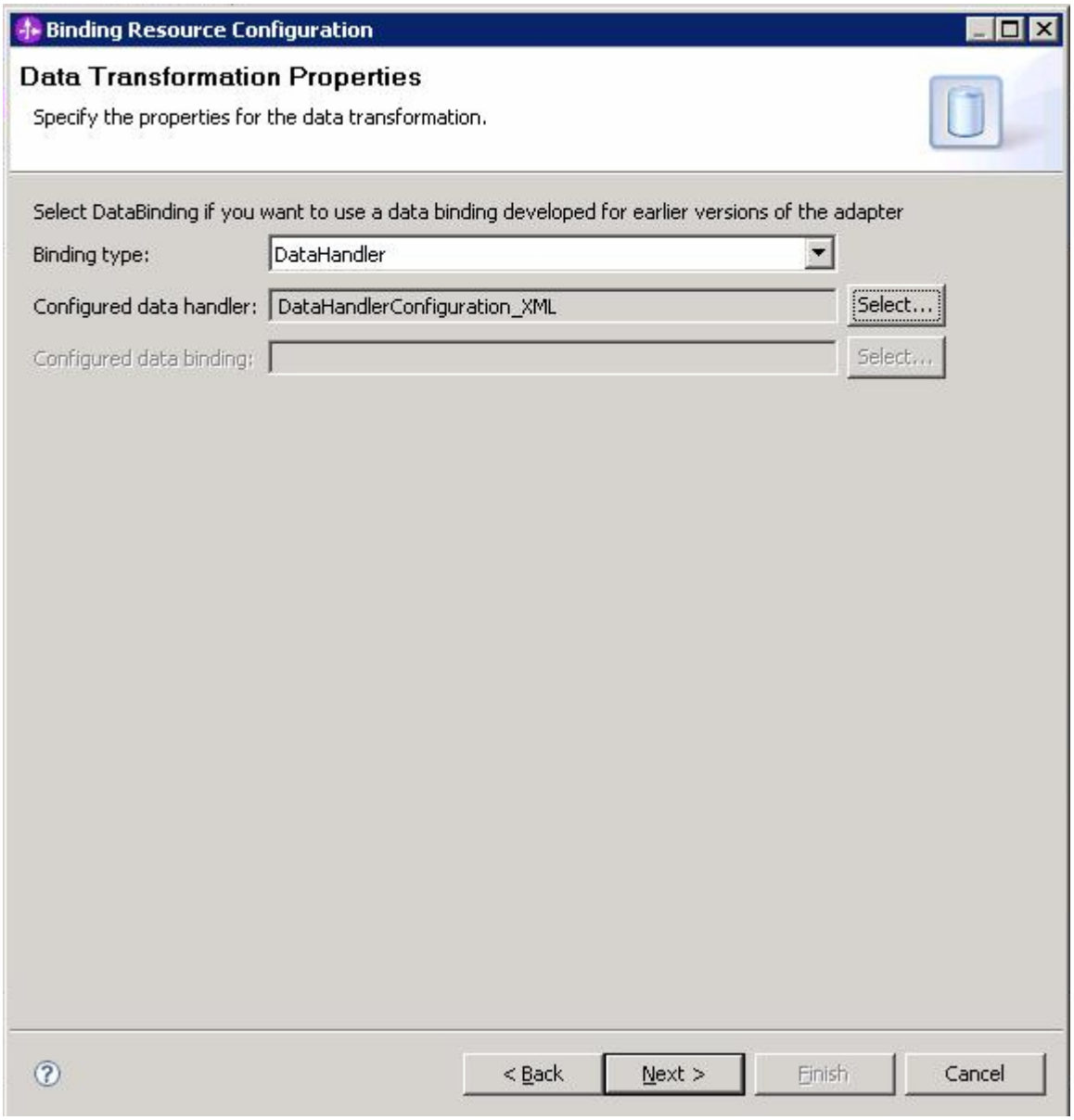
- In the Data Transformation Properties window, specify the **Encoding**, and optionally the Document root name and Document namespace. Leave them as blank for this tutorial. Click **Next**.



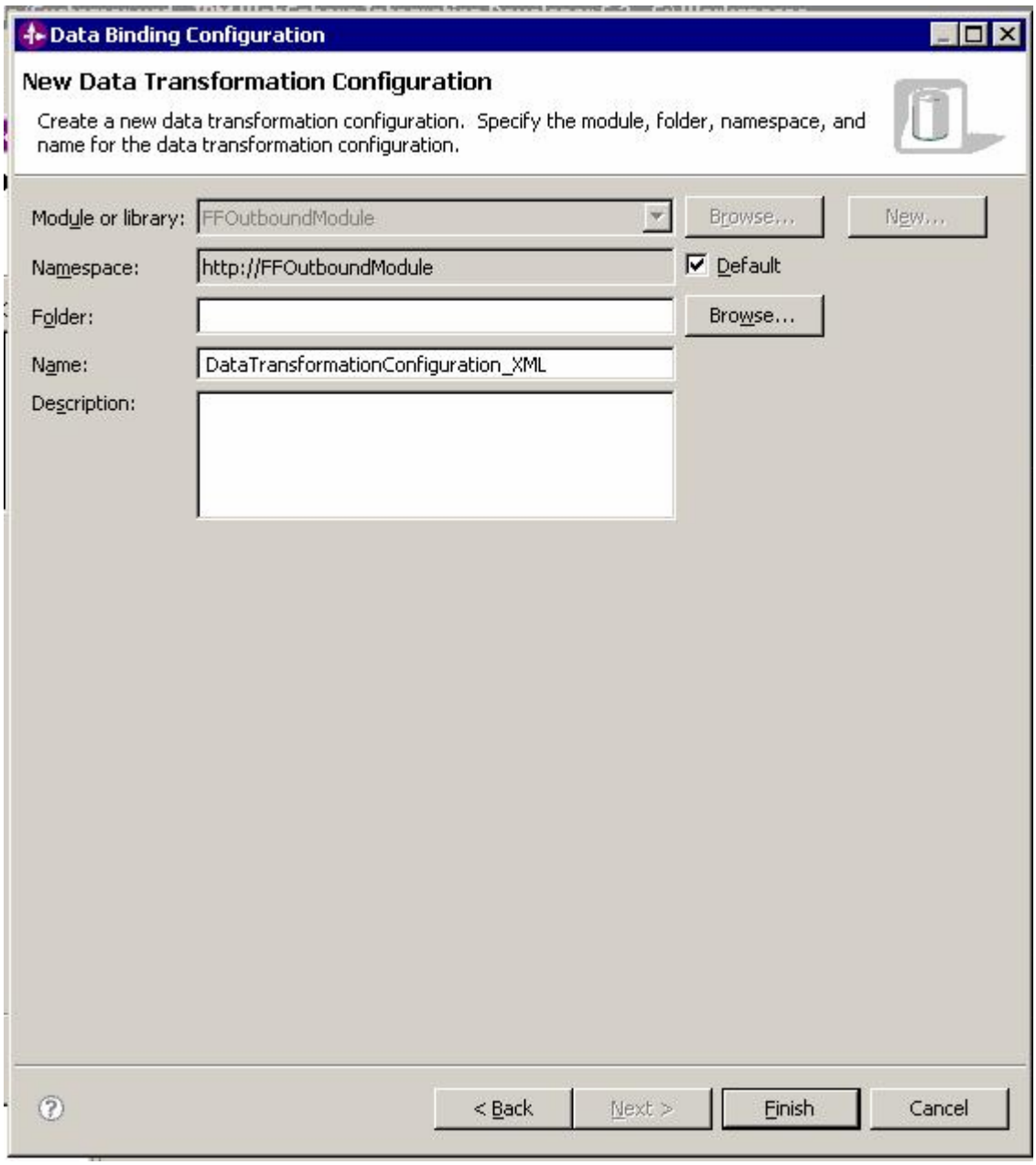
- Specify a unique name (DataHandlerConfiguration_XML is the name specified in this illustration) for this Datahandler configuration and click **Finish**.



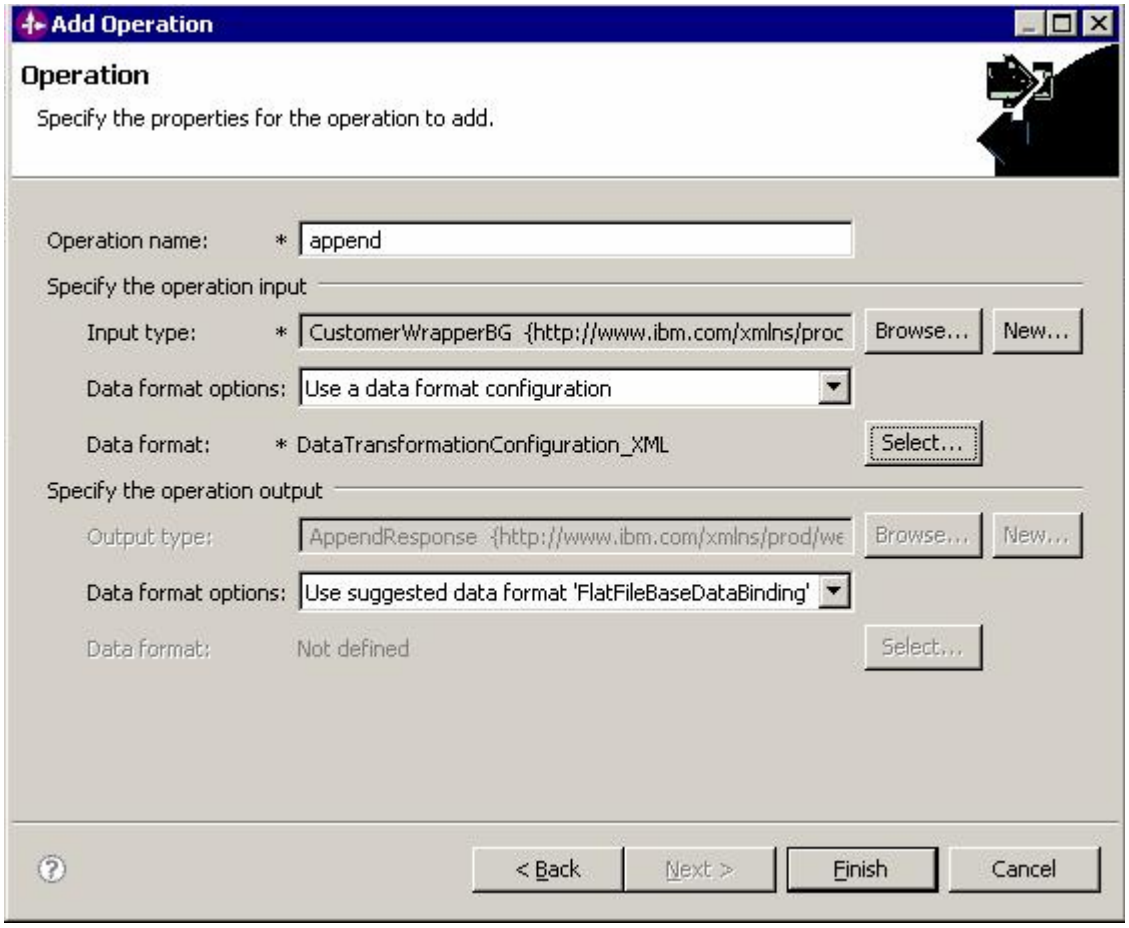
- The Data Transformation Properties window displays the configured data handler. Click **Next**.

- 

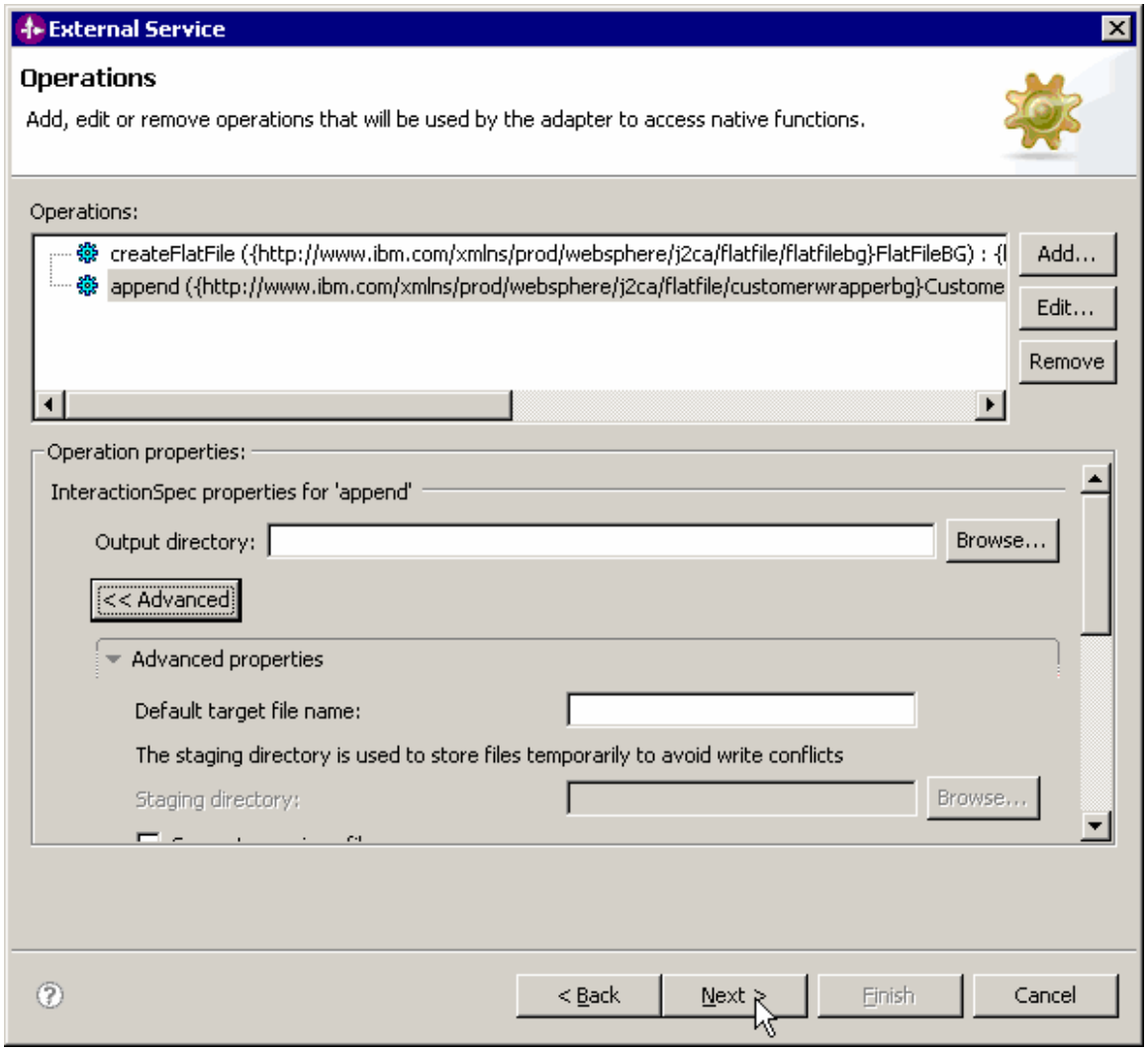
- In the New Data Transformation Configuration window, give a name to the Data binding configuration (DataTransformationConfiguration_XML is given in this illustration) and click **Finish**.



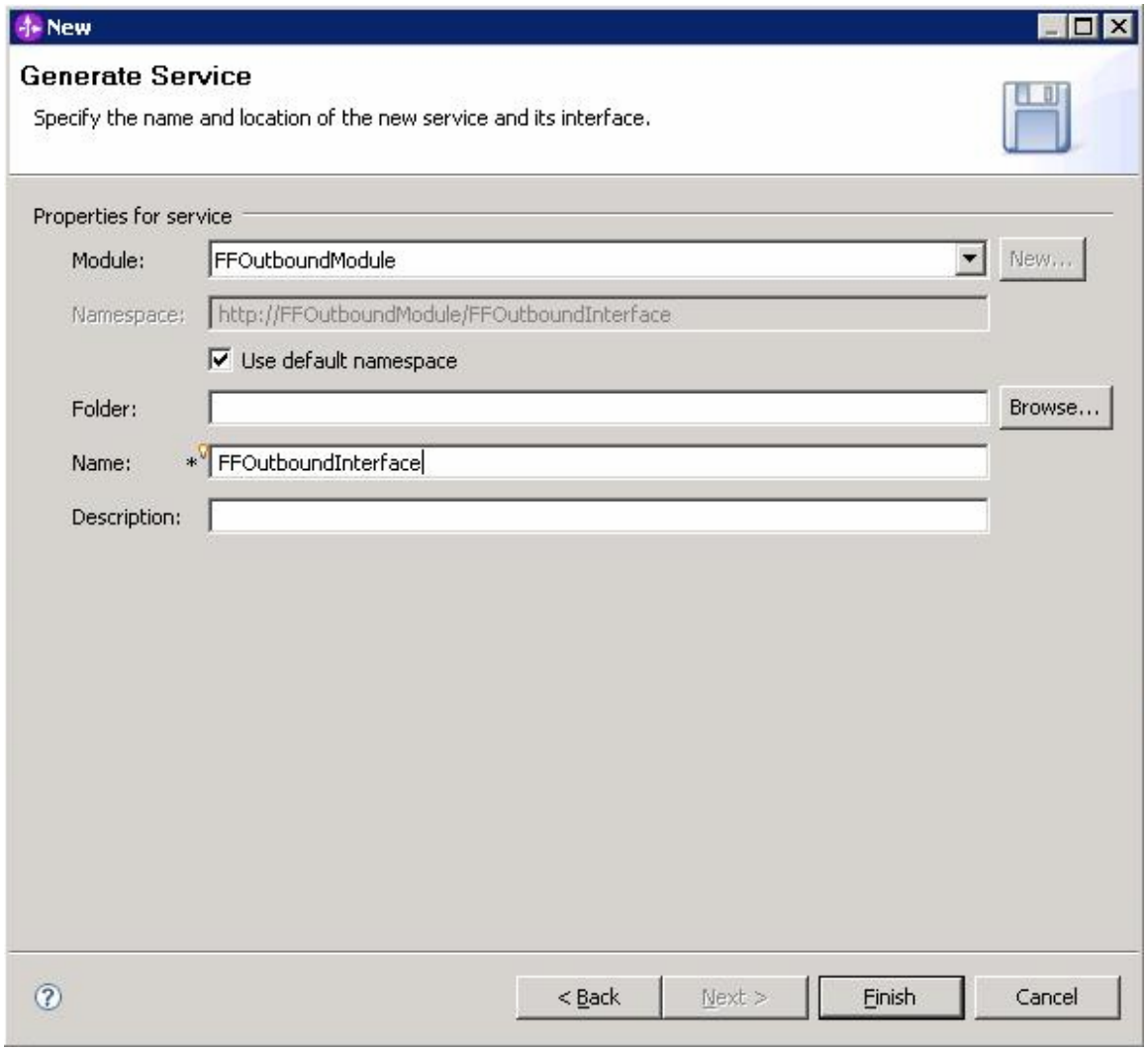
- You will see the Data Binding Configuration for the input Type on this window. Click **Finish** to exit Operation window.



- In the next window, the Interaction spec properties can be specified as shown below. No values are set at the interaction spec level. All values will be set at record level in this tutorial. Click **Next**.



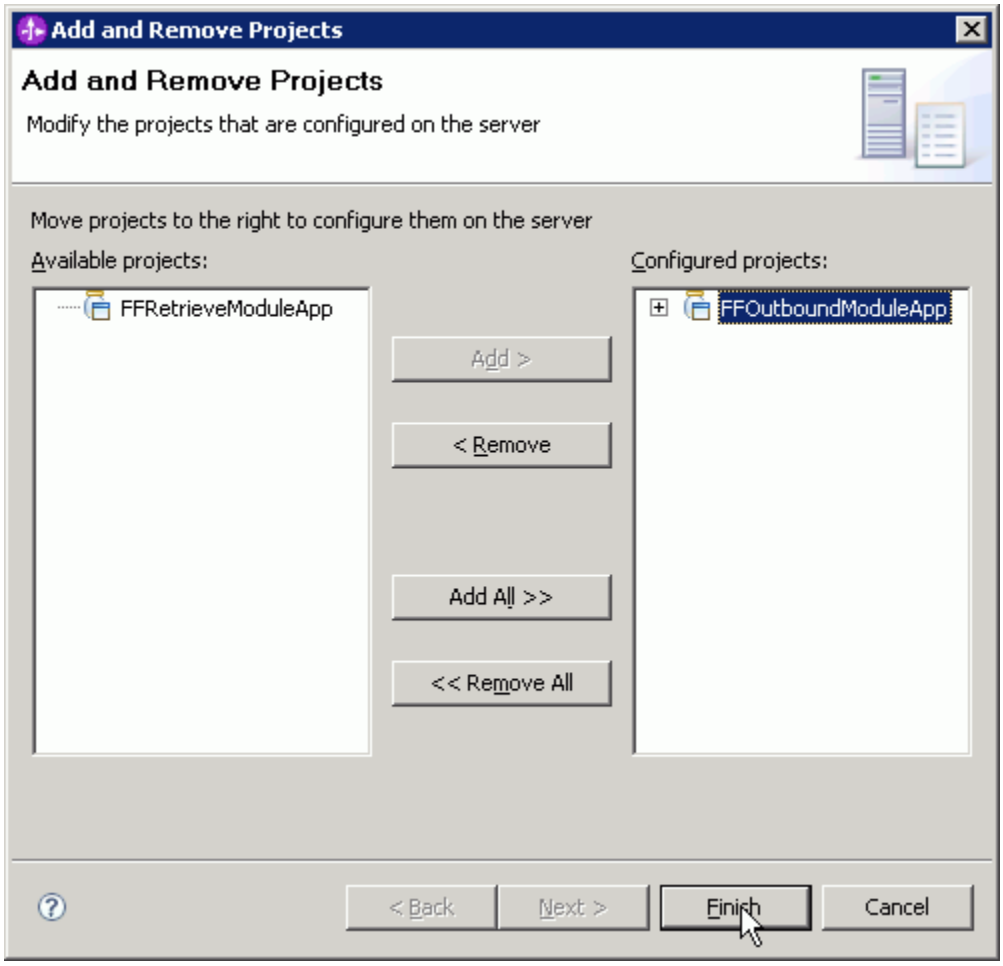
- In the last step of running External service wizard, specify a suitable name for your adapter interface. In this illustration, use the default value specified. Click **Finish**.



Deploying the module to the test environment

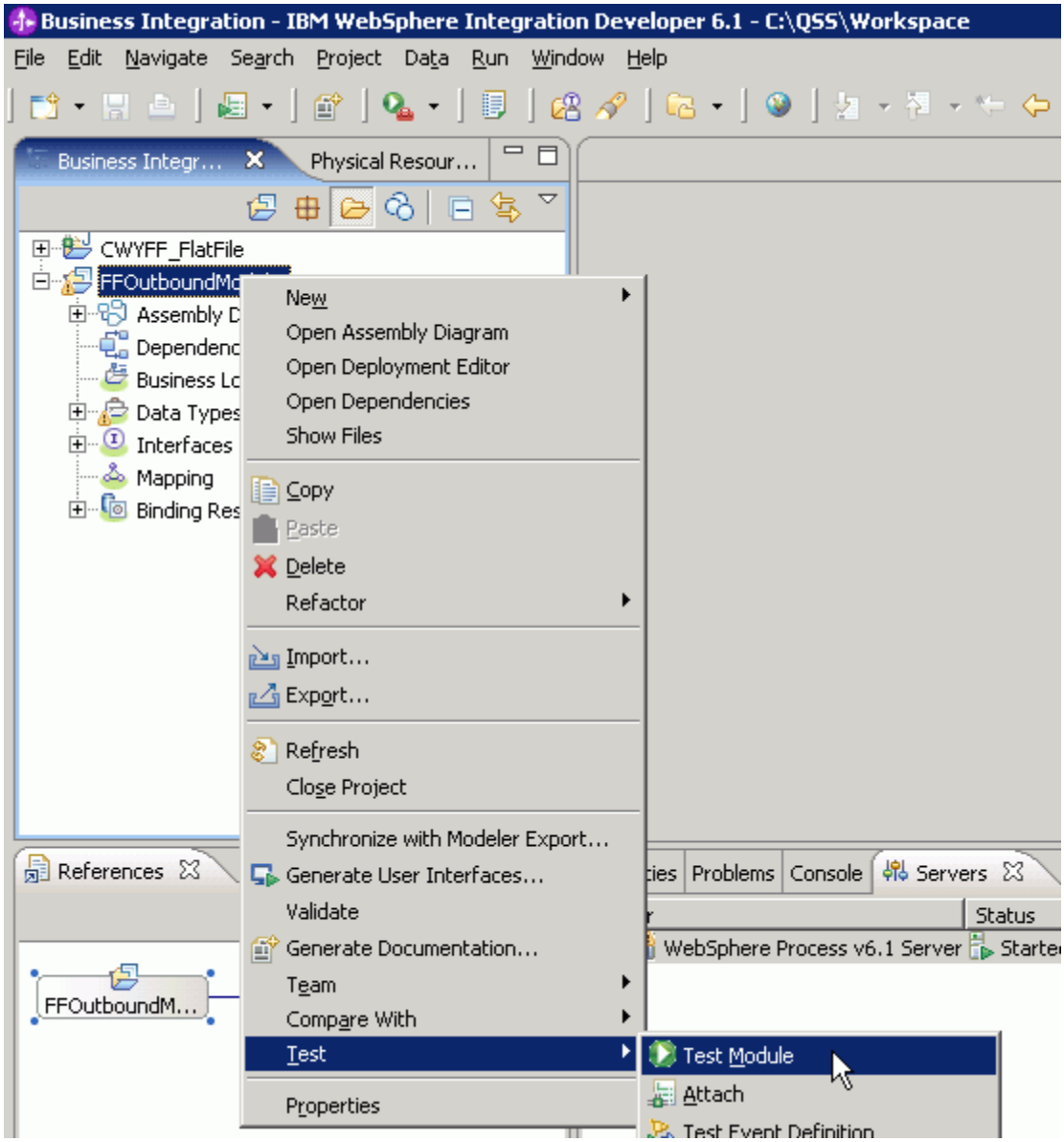
The following steps show how to perform Retrieve operation with the module you just created.

- Start the WebSphere Process Server.
- Add adapter module to the server. In the Server tab, right click on WebSphere Process Server then click **Add and Remove Projects**.
- From the **Available projects** pane, select your adapter module, click **Add >** and click **Finish**.

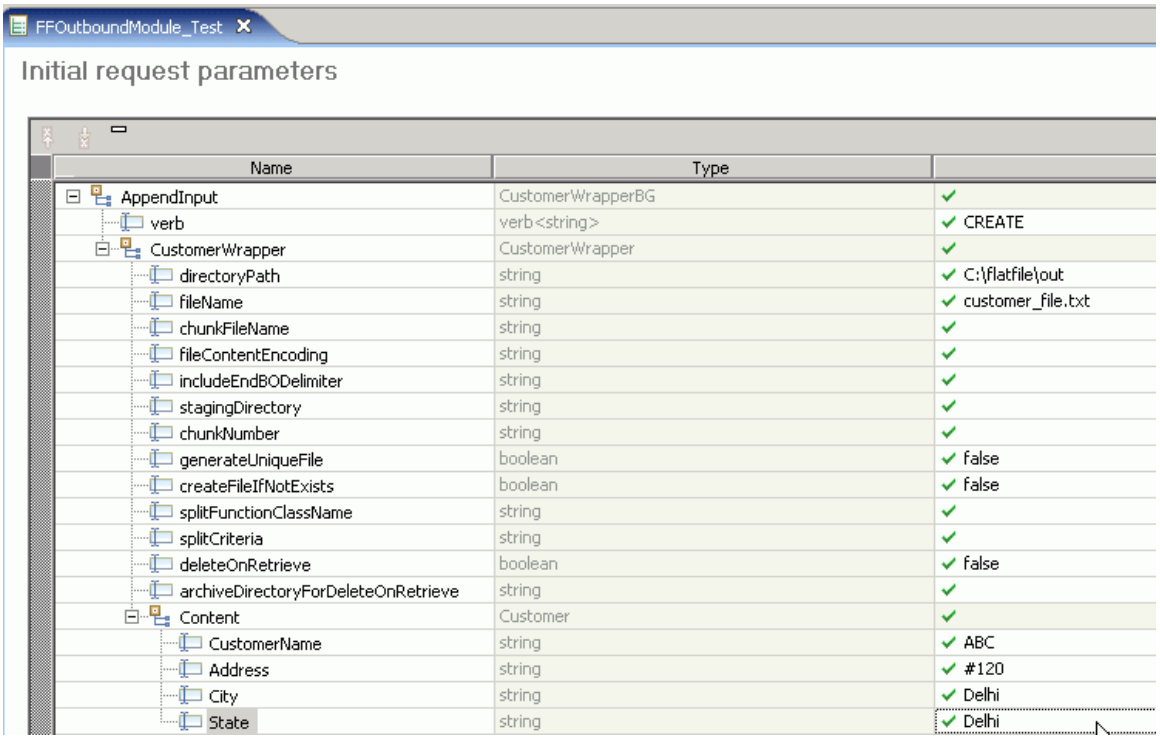


Testing the assembled adapter application

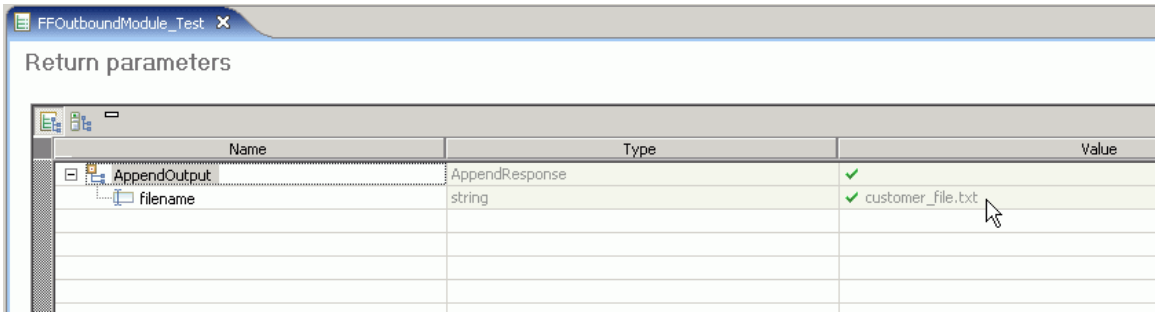
- Right click on the adapter module, FFOutboundModule then click **Test** -> **Test module**.



- Specify the required parameters to carry out Append operation, i.e., **directoryPath**, **fileName** (which already exists). And content to be appended to the file.



- Click Invoke. The following business graph which is the output of Append operation will be displayed. It contains a filename field which carries the value of the name of the file created during Create operation. Append operation was carried out successfully and the desired content has been appended to the specified file.



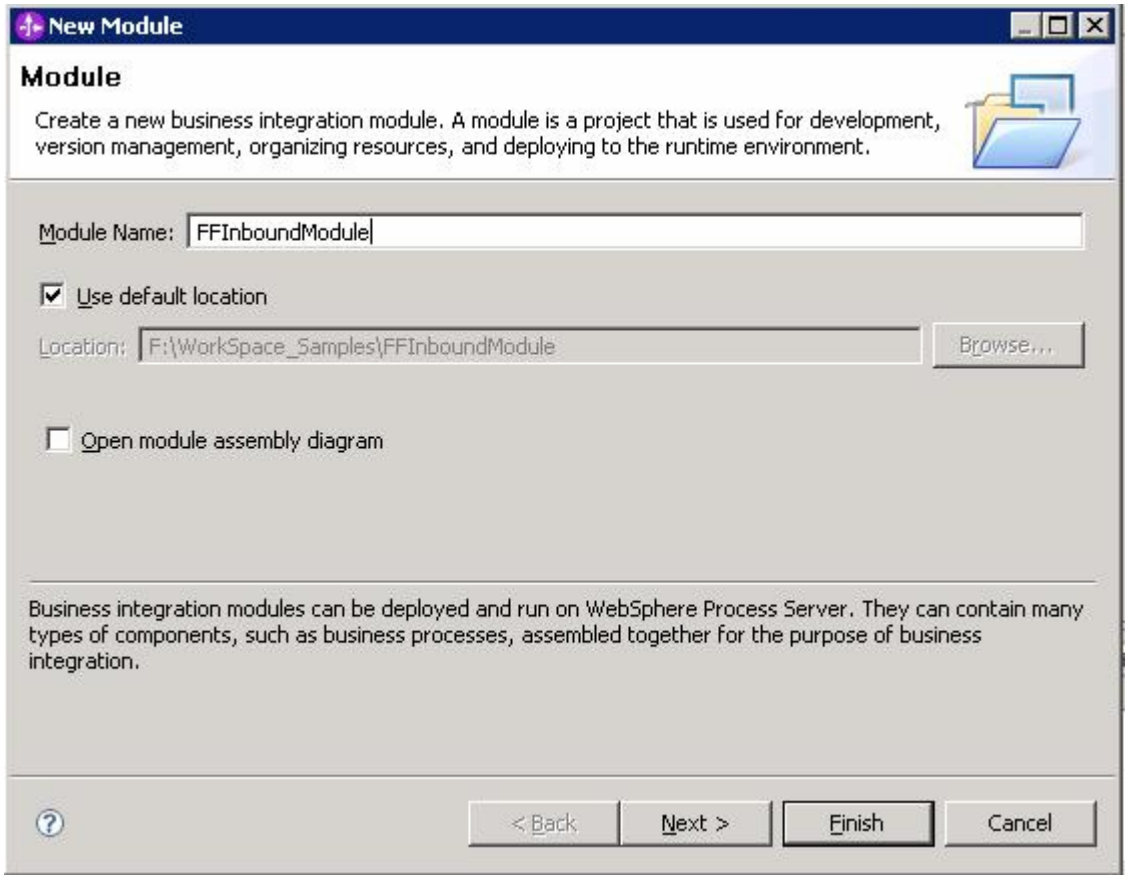
Chapter 6. Tutorial 4: Inbound processing – Polling of a sample file without data transformation

This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.2.0.0 can be used to poll a file to send a business object to the end-point. The delivered event/business object in this example will be of type Unstructured content.

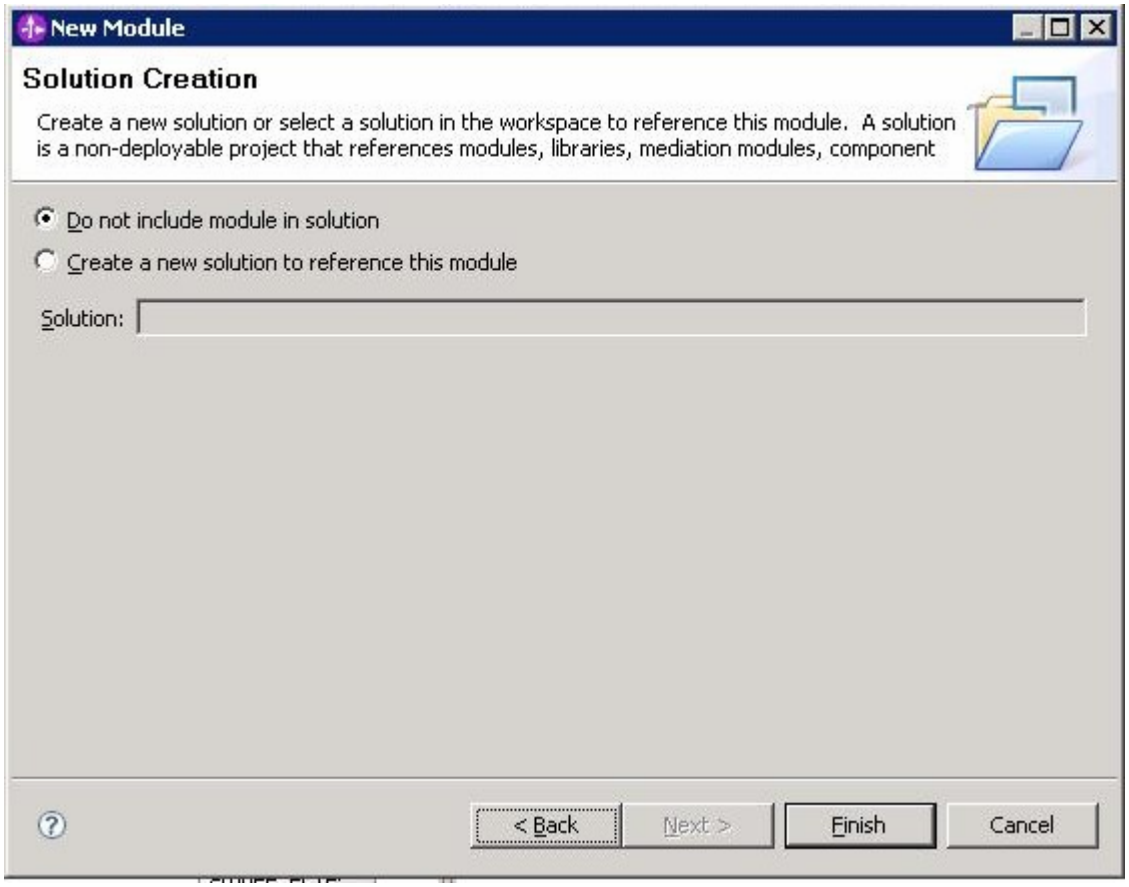
Configuring the adapter for inbound processing

Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

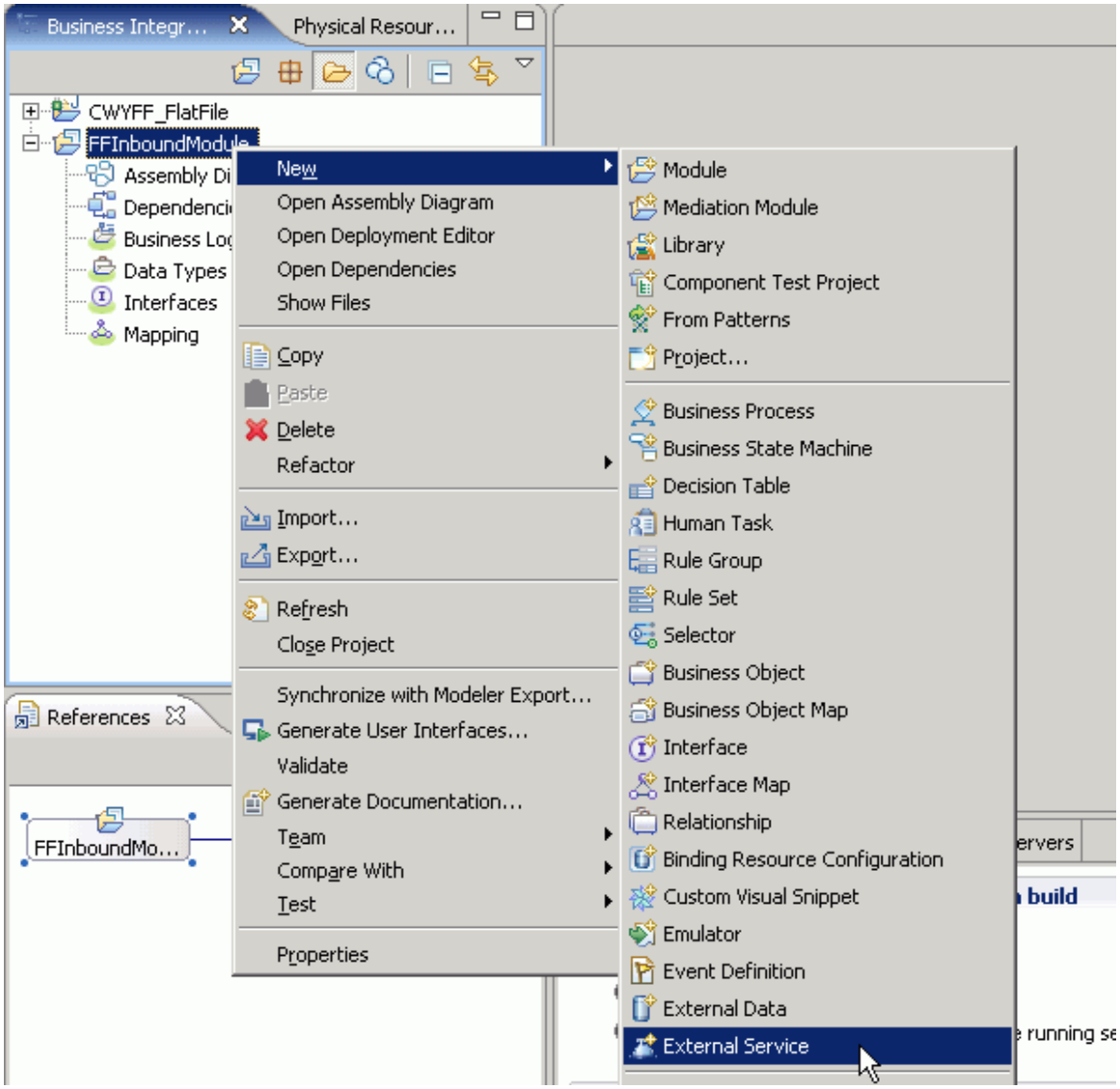
- Create a new module. Go to **File** -> **New** -> **Module.**
- Enter a desired name for your module. This tutorial uses FFInboundModule as the **Module name**.



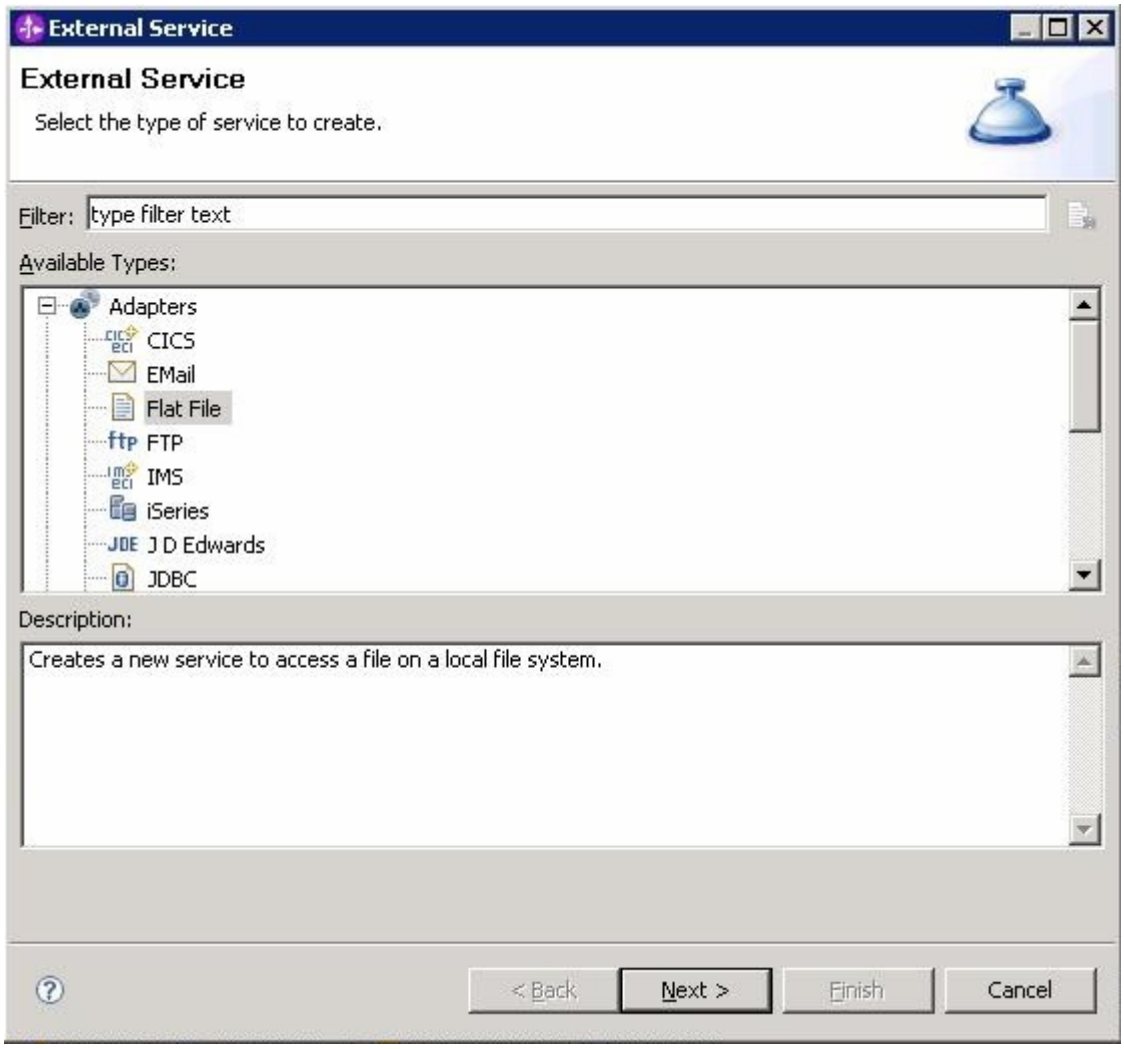
- Leave Do not include module in solution selected and click Next.



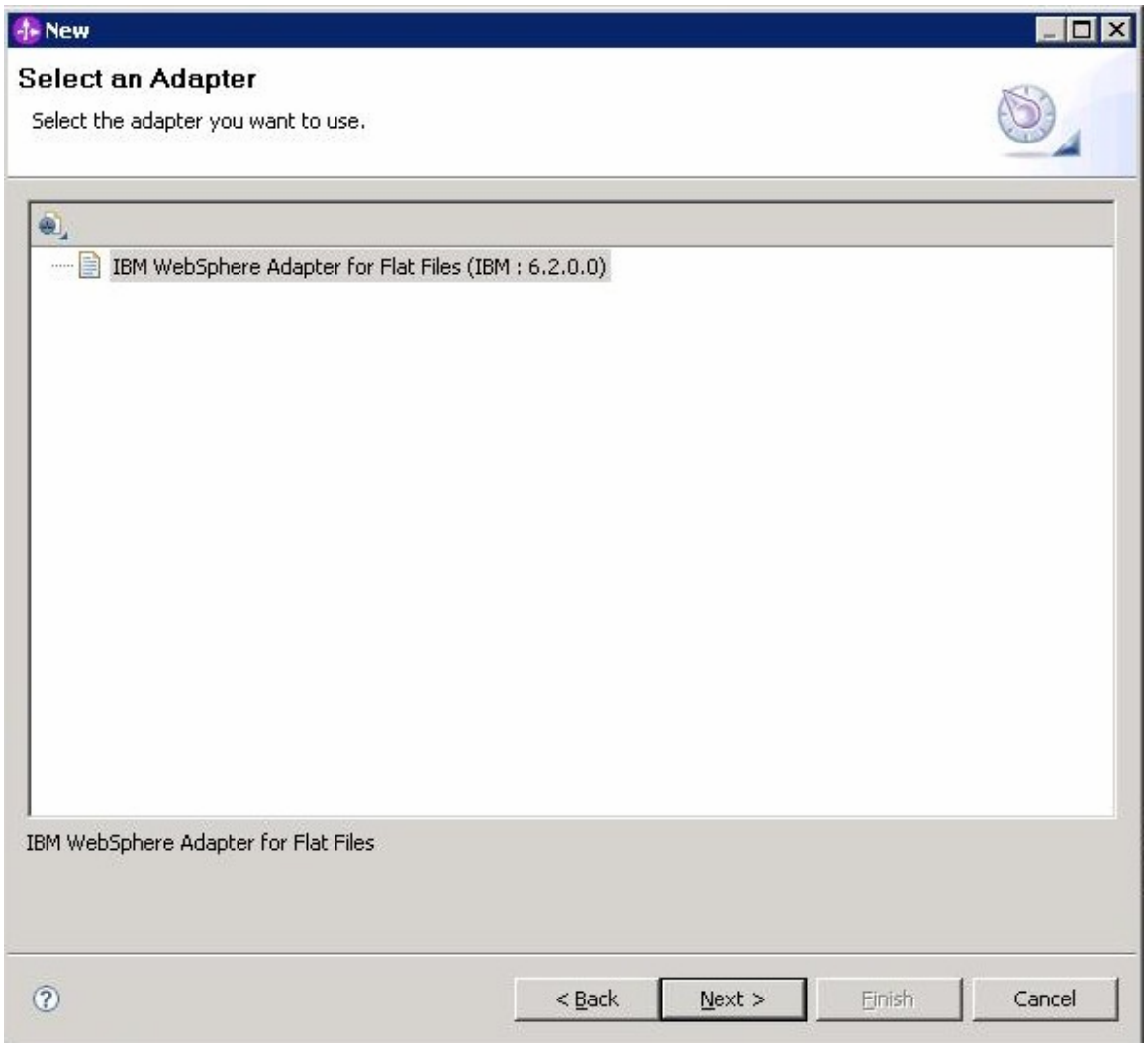
- Start the WebSphere FlatFiles adapter external service wizard by choosing: **File** -> **New** -> **External Service**.



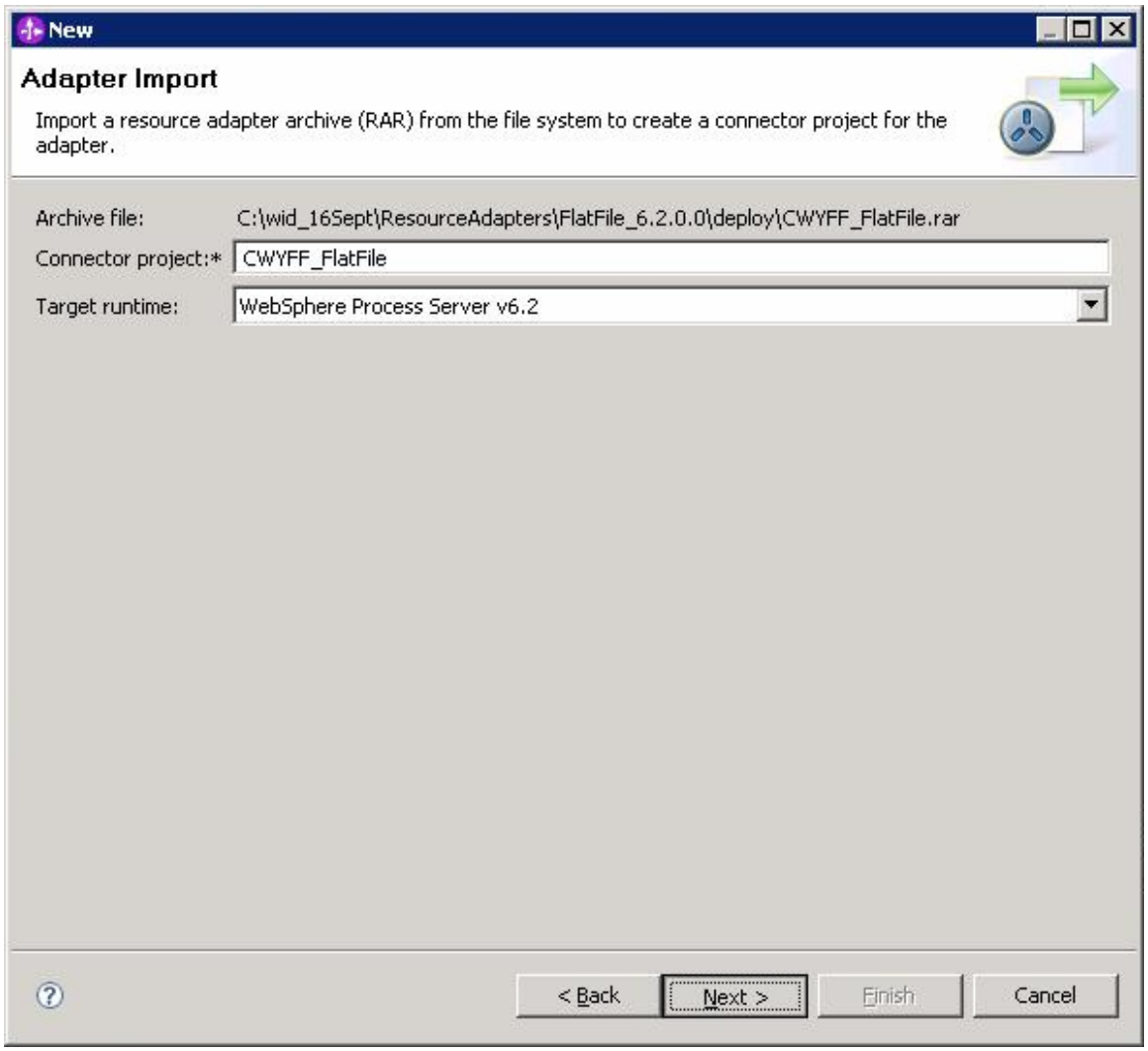
- Expand **Adapters**, select **Flat File** and then click **Next**.



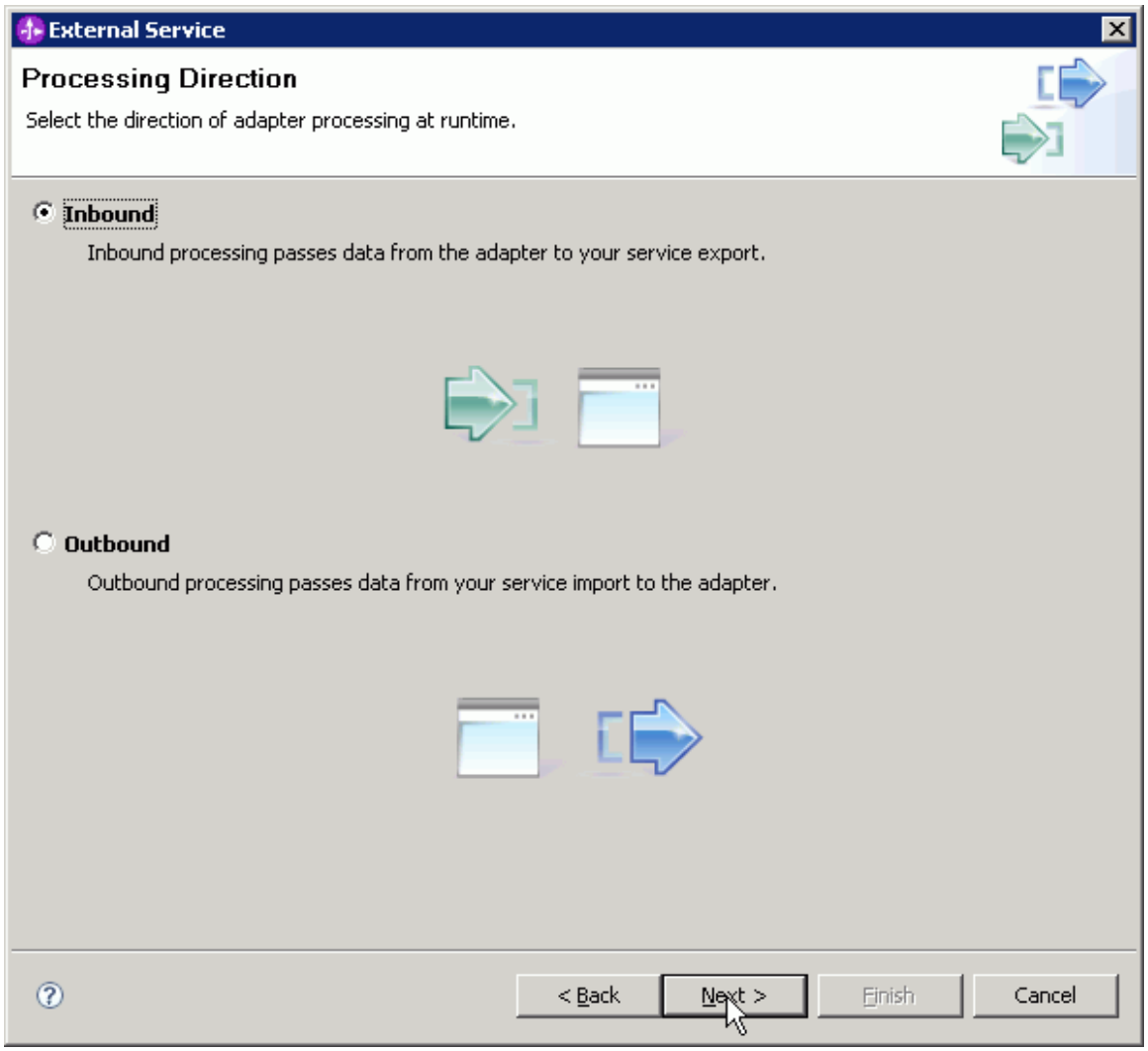
- In the Select an Adapter window, select **IBM WebSphere Adapter for FlatFiles (IBM : 6.2.0.0)**. Click **Next**. The adapter RAR file will be imported into the module.



- In the Adapter Import window, leave the default value in the **Connector project** field. Ensure that the **Target runtime** is set to **WebSphere Process Server v6.2**. Click **Finish**.

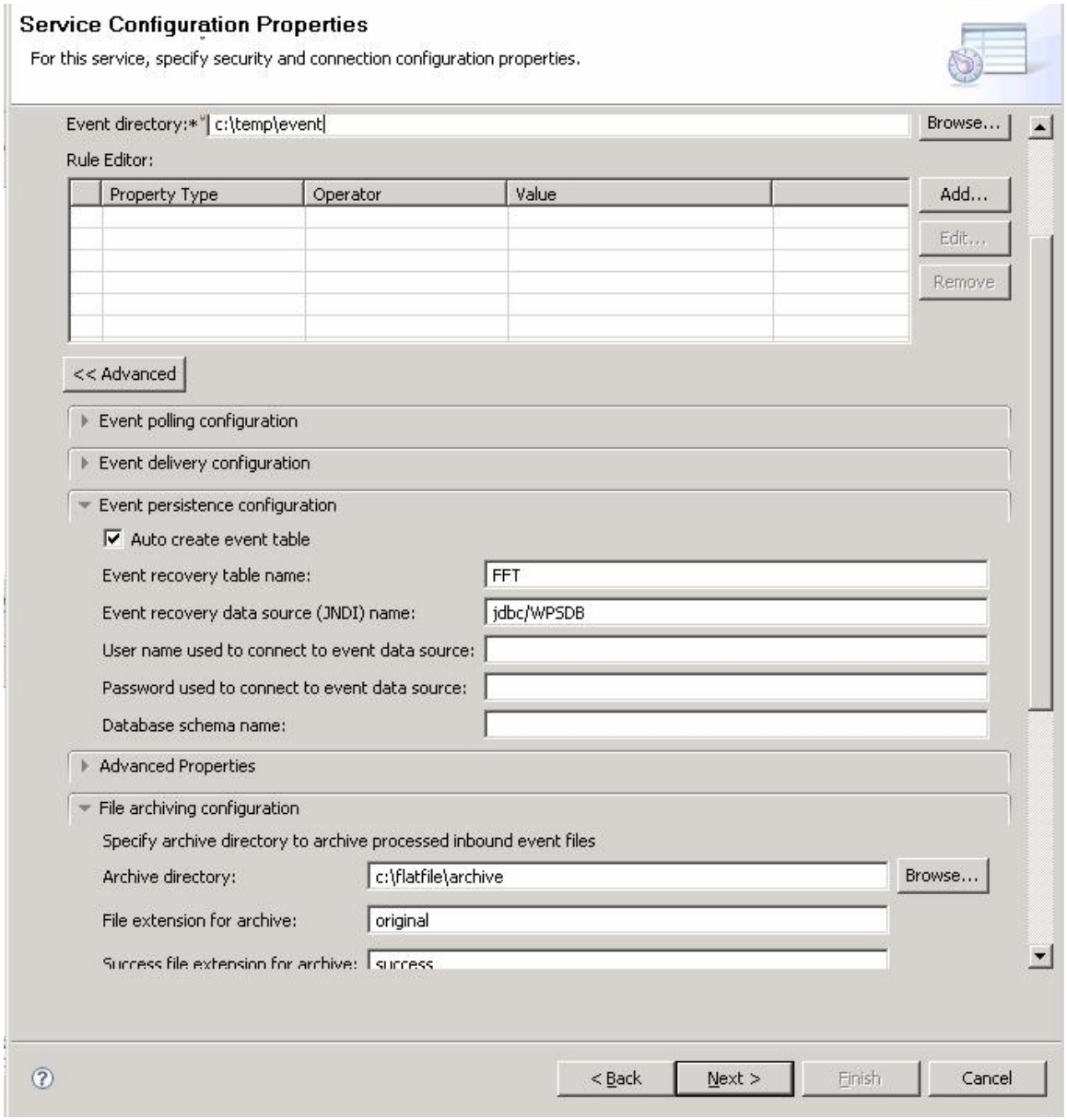


- In the Processing Direction window, select **Inbound** and click **Next**.

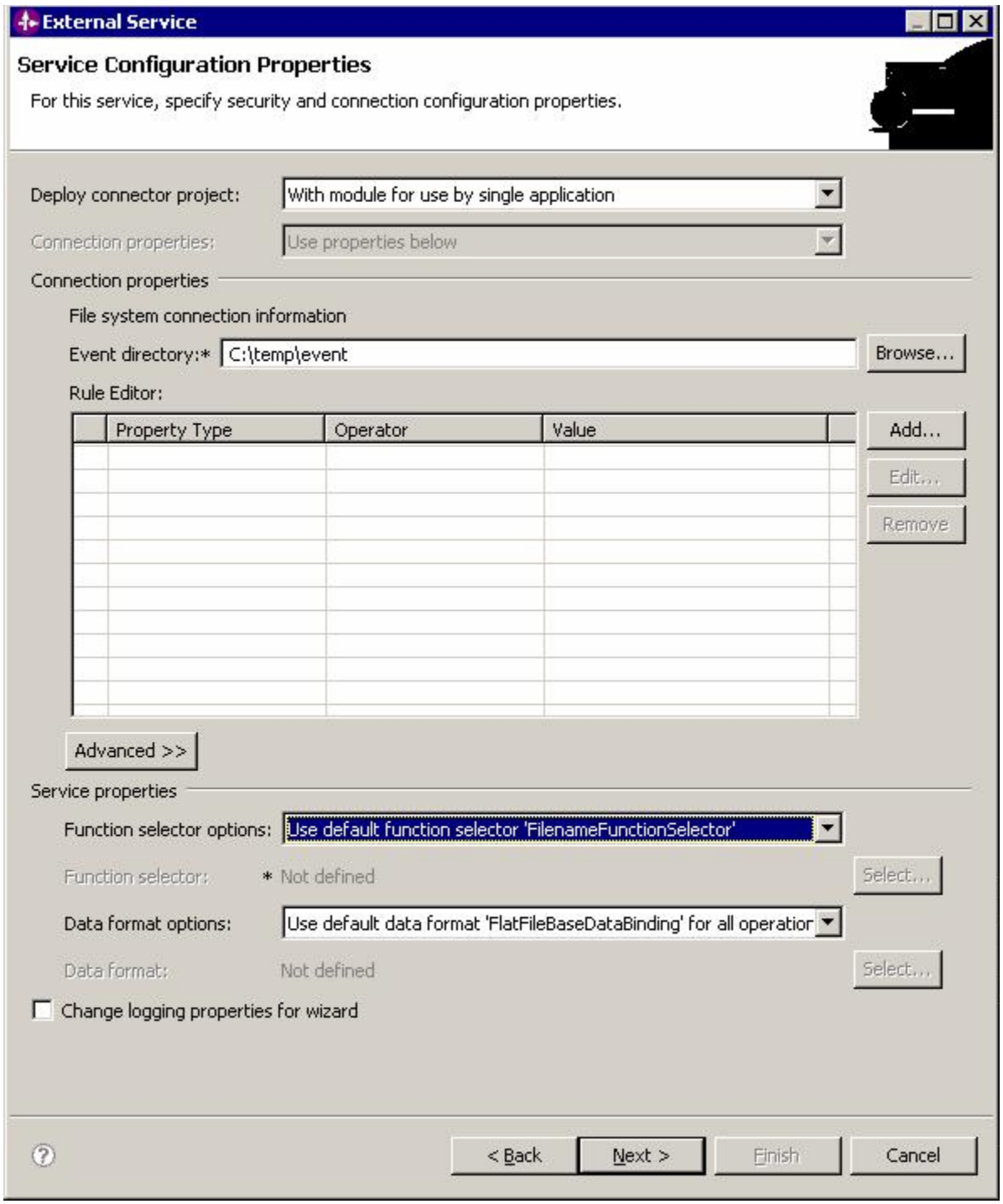


Setting properties for the external service wizard

- In the Service Configuration Properties window shown below, specify values for **Event directory** and **Archive directory. Event recovery table name** and **Event recovery data source (JNDI) name** can be left blank in which case in-memory processing of file is carried out.



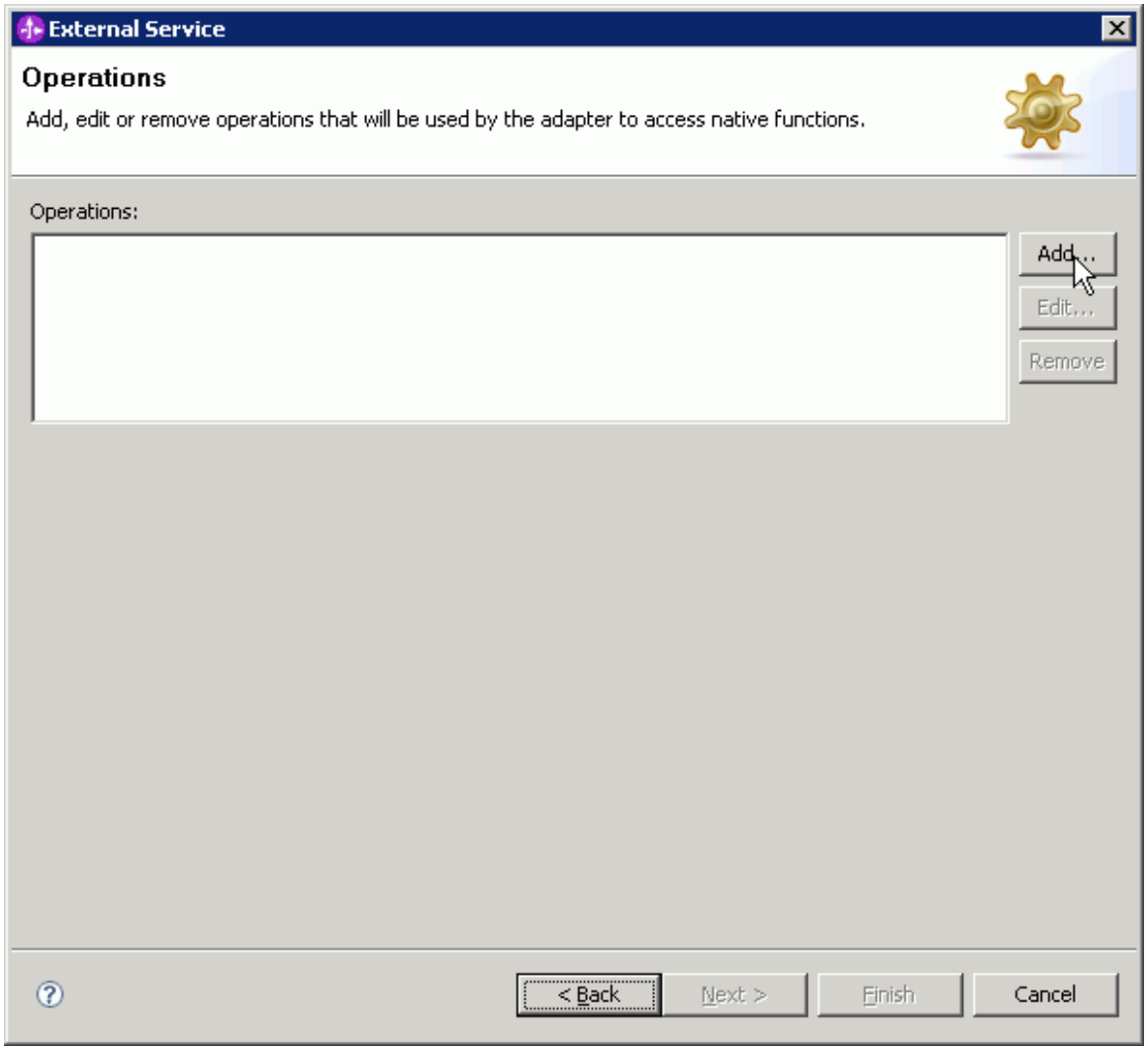
- Configure the function selector which determines which method is to be called at end-point during event delivery. In **Function Selector** list box choose **Use a function selector configuration**. Click **Select** button alongside the label **Function selector configuration**.



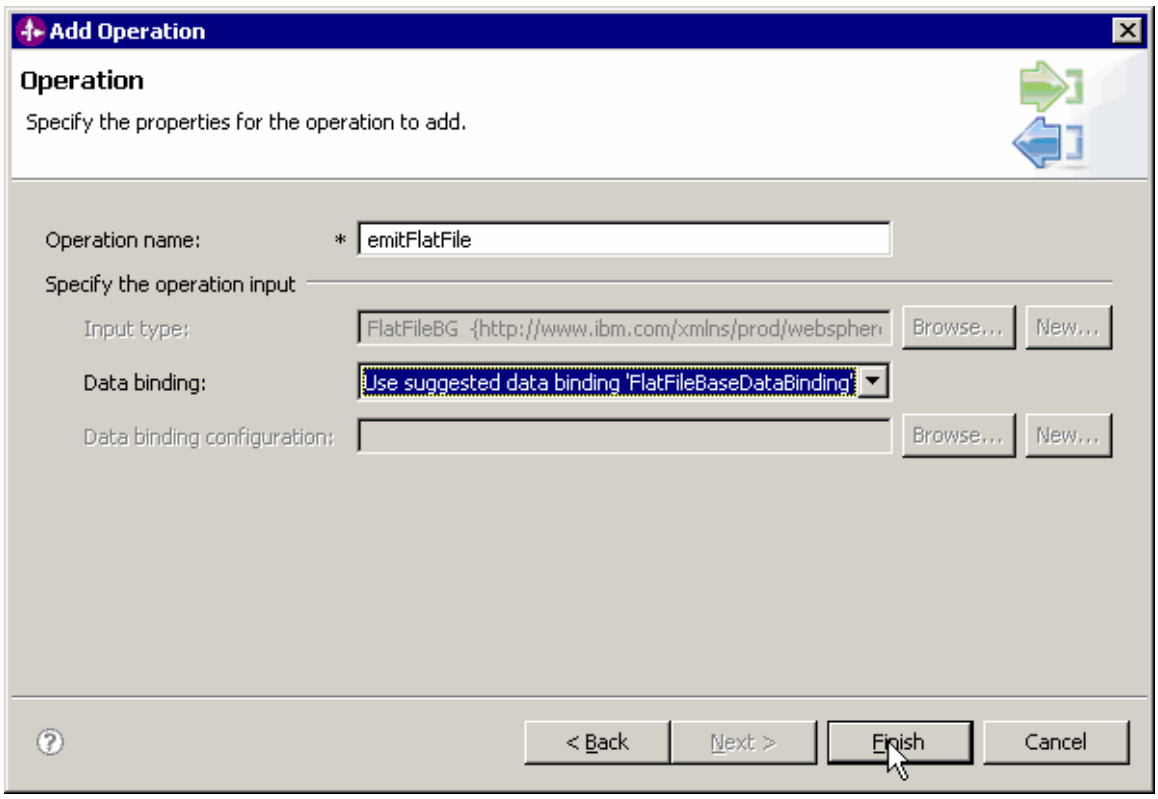
- Select the Use default function selector 'FilenameFunctionSelector'. Click Next.

- After selecting the Default Function selector in the screen capture below, click **Next**

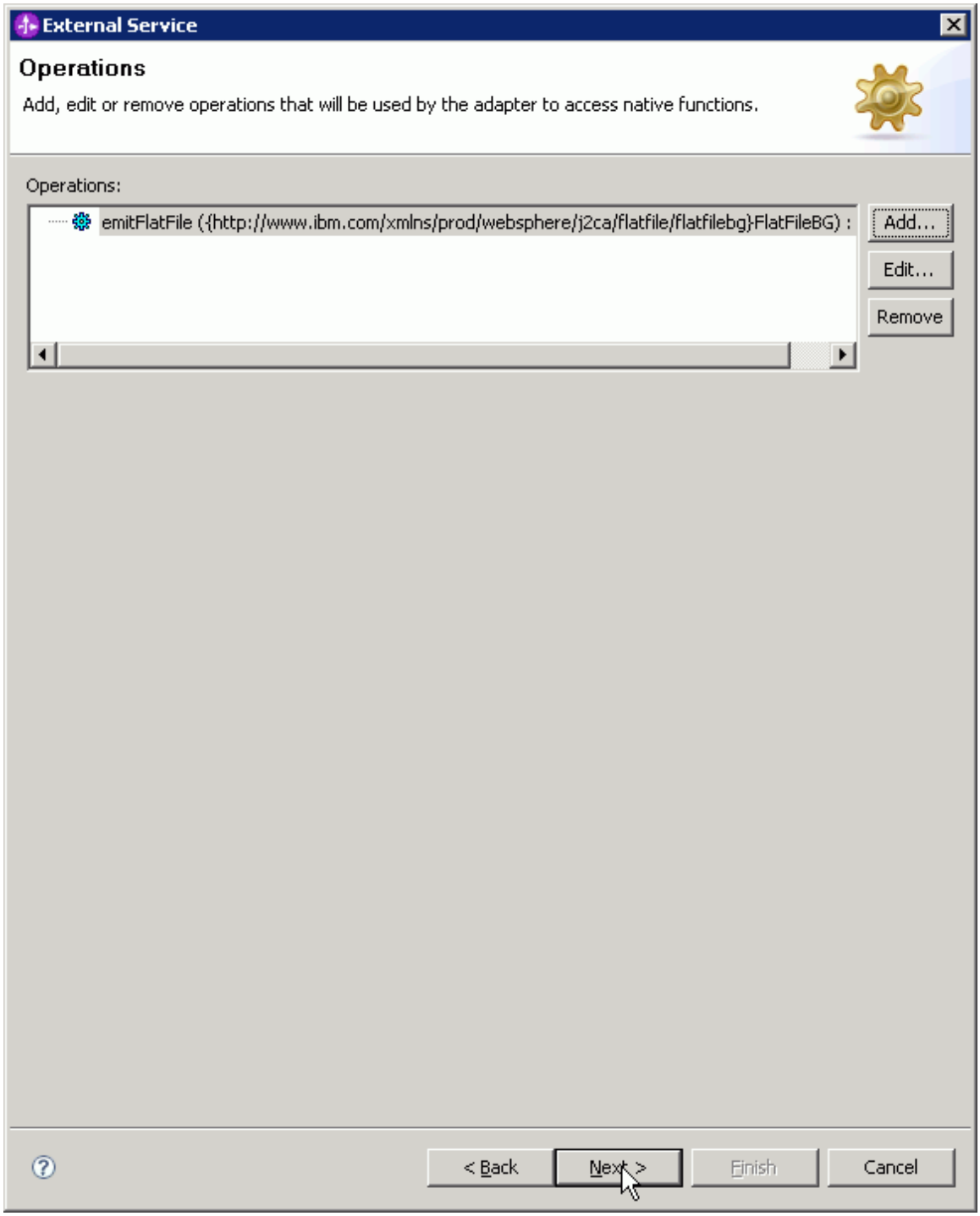
- Next, in the Operations window, click **Add** to add the operations that you want to perform.



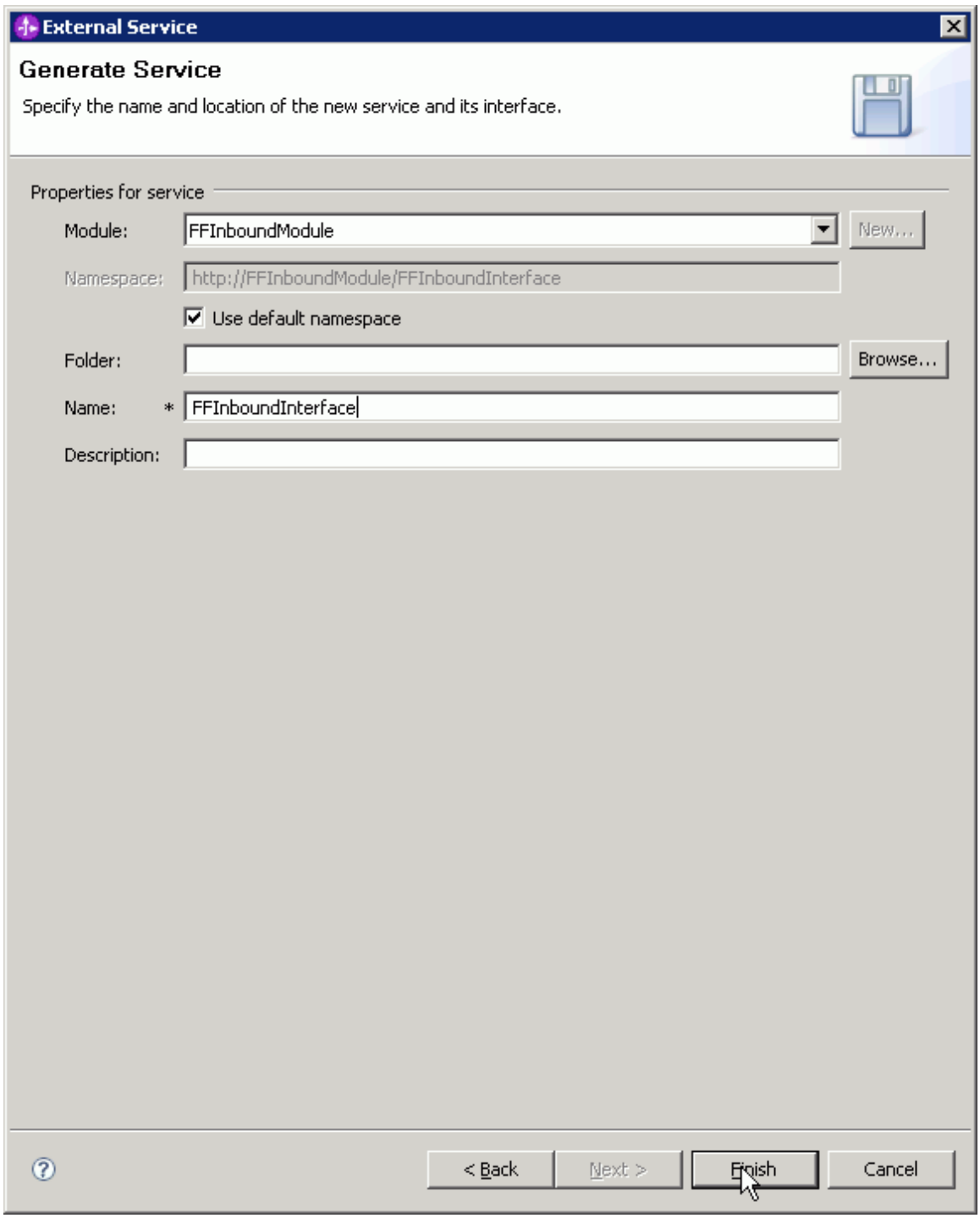
- The only operation available in case of inbound is **emitFlatFile**. Next, create a Data binding configuration. Choose the **Use suggested data binding FlatFileBaseDataBinding,** which is the default. Click **Finish**.



- Click **Next** in the screen capture that follows.



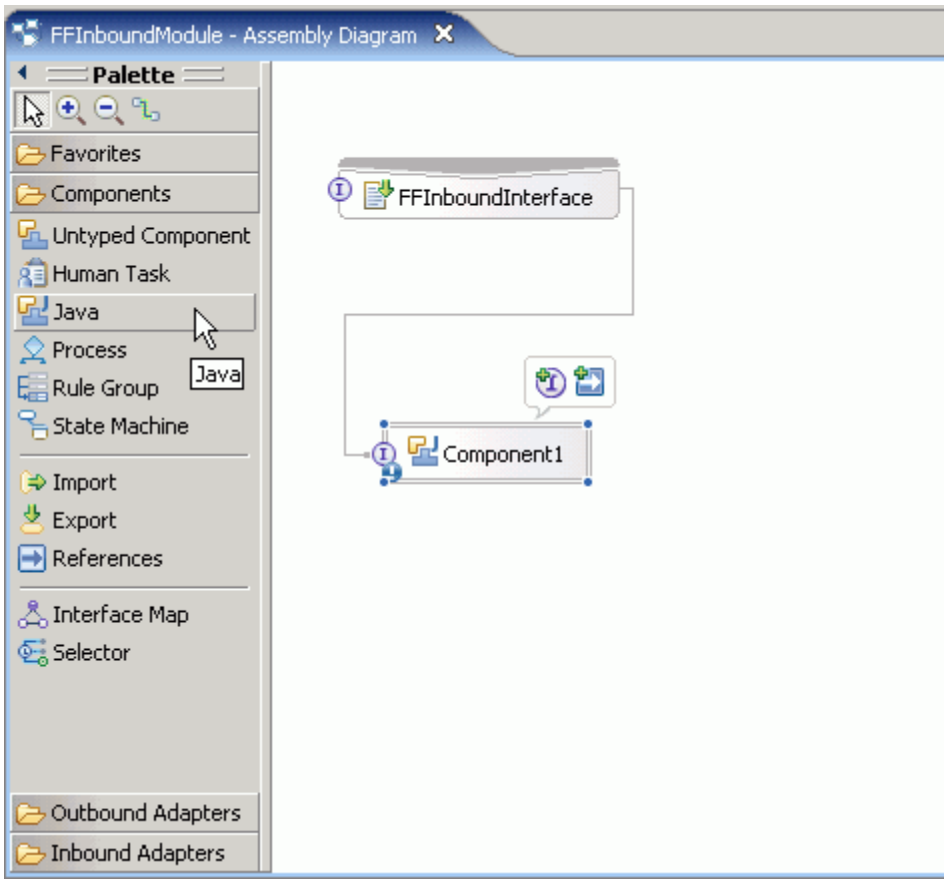
- In the last step of running external service wizard, specify a suitable name for your adapter interface. In this tutorial, leave the default value as FFInboundInterface for the **Name** field. Click **Finish**.



Generating business object definitions and related artifacts

- Generate a component of Java™ type and draw a wire from **FFInboundInterface** to **Component1.**

- Implement the Java component by right clicking Component1 and clicking on **Generate Implementation**. You can choose to have the implementation belong to default package or any other package.



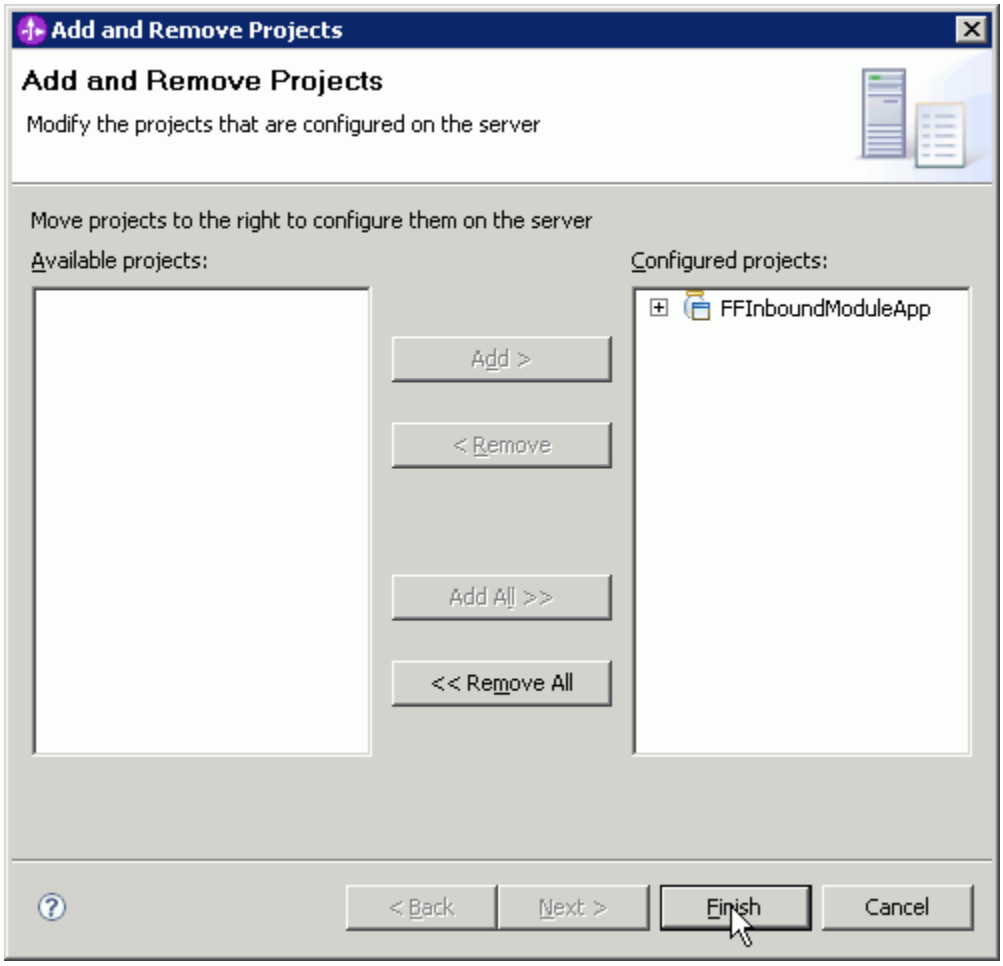
- A window containing a code snippet of the generated implementation of Component1 is shown below. Customize the implementation of the method emitFlatFile as shown in the screen capture below to test the module.

- 

Deploying the module to the test environment

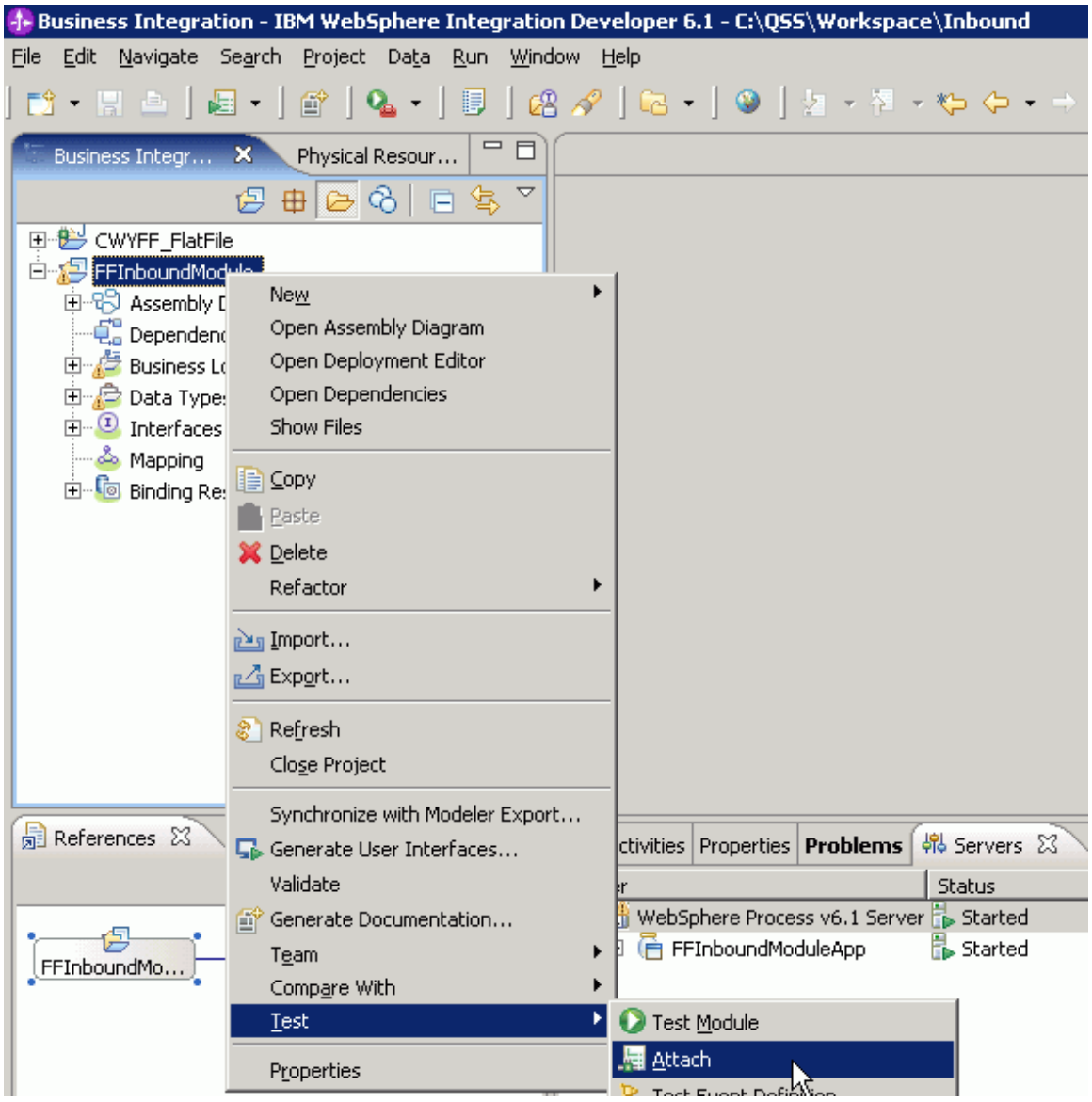
Let us proceed on to performing Inbound operation with the module thus created.

- Start the WebSphere Process Server
- Add adapter module to the server. In the **Server** tab, right click on **WebSphere Process Server** then select **Add and Remove Projects**
- From the **Available projects** pane, select your adapter module, click **Add >** and click **Finish**.



Testing the assembled adapter application

- Right click on the adapter module, **FFInboundModule** then select **Test** -> **Attach.**
- Copy a sample event file to the specified Event directory.
- The relevant business object will be delivered to the end-point.
- This can be verified by checking for the end-point messages in System.Out file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer.


Business Integration - IBM WebSphere Integration Developer 6.1 - C:\QSS\Workspace\Inbound
File Edit Navigate Search Project Data Run Window Help
Business Integr...
Physical Resour...
CWYFF_FlatFile
FFInboundMod
Assembly D
Dependenc
Business Lc
Data Type
Interfaces
Mapping
Binding Re
New
Open Assembly Diagram
Open Deployment Editor
Open Dependencies
Show Files
Copy
Paste
Delete
Refactor
Import...
Export...
Refresh
Close Project
Synchronize with Modeler Export...
Generate User Interfaces...
Validate
Generate Documentation...
Team
Compare With
Test
Properties
Test Module
Attach
References
FFInboundMo...
ctivities
Properties
Problems
Servers
Status
WebSphere Process v6.1 Server
Started
FFInboundModuleApp
Started

Chapter 7. Tutorial 5: Inbound processing – Polling of a sample file with XML data transformation

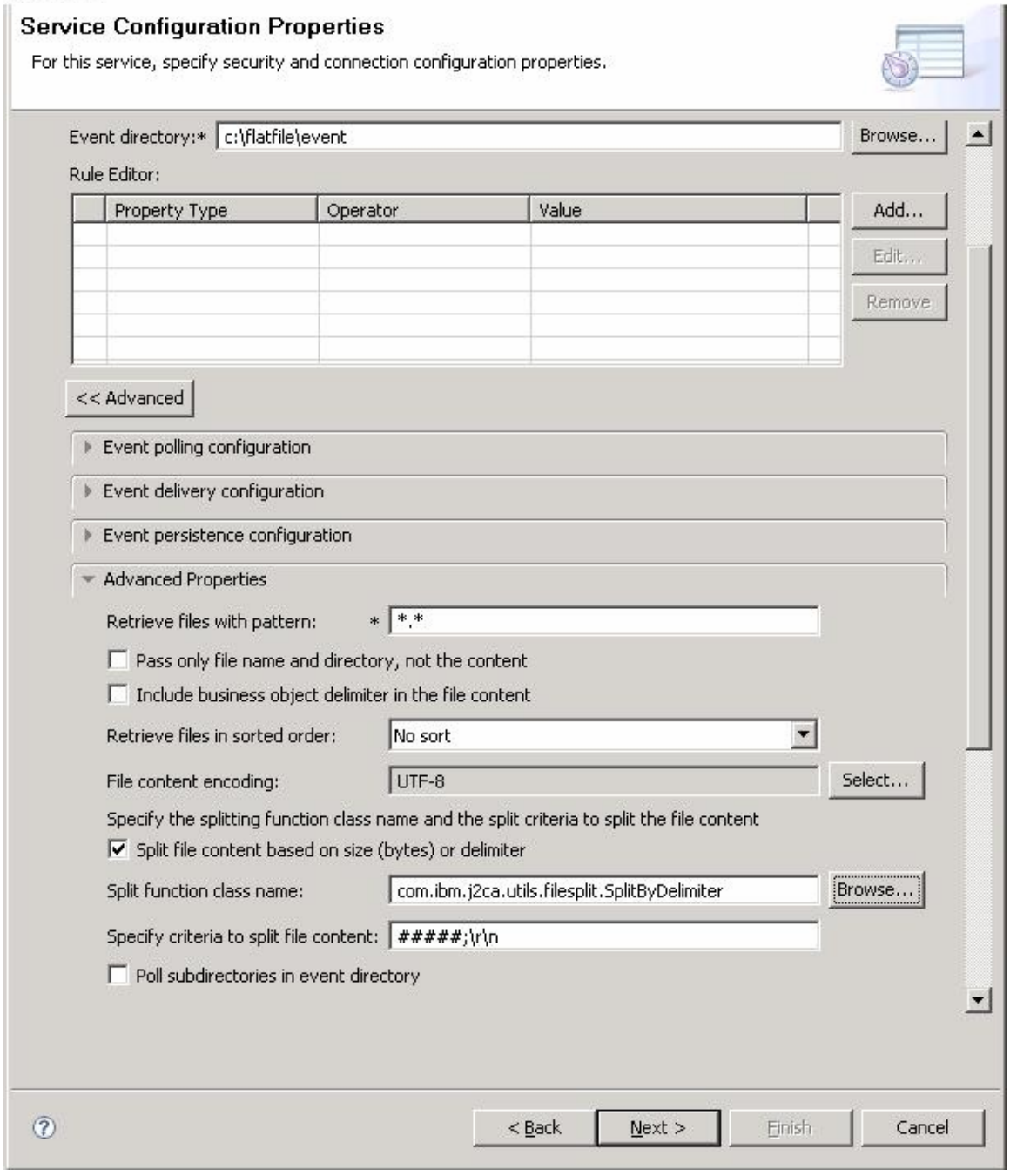
This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.2.0.0 can be used to poll a file to send a business object to the end-point. The delivered event/business object in this example will go through data transformation with XML Data Handler.

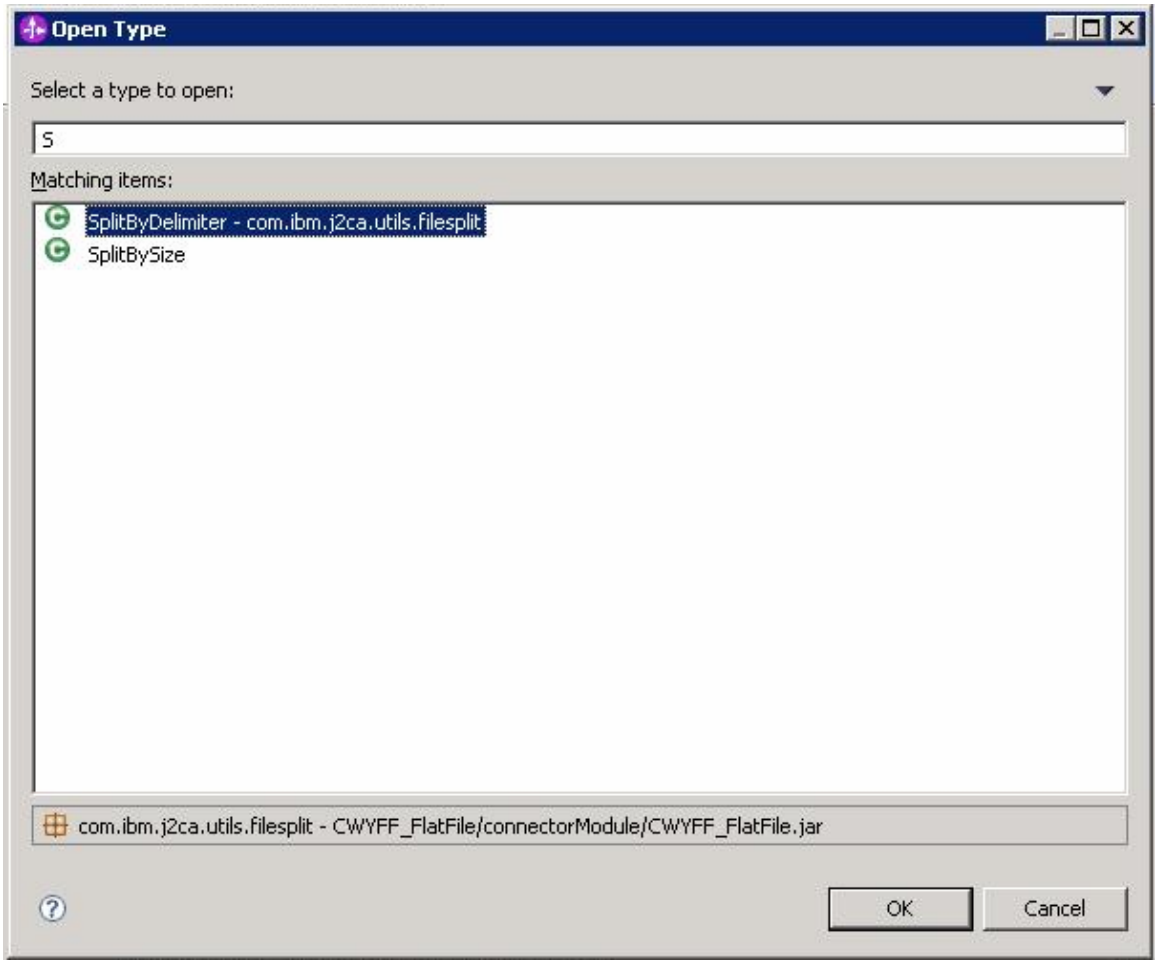
Configuring the adapter for inbound processing

Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial. Follow the steps given in the topic 'Configuring the adapter for inbound processing' in tutorial 1 to proceed with inbound processing with XML data transformation. Additionally, the content of the files passing through this sample will be chunked using SplitByDelimiter which looks for delimiter pattern in the file and chunks it accordingly.

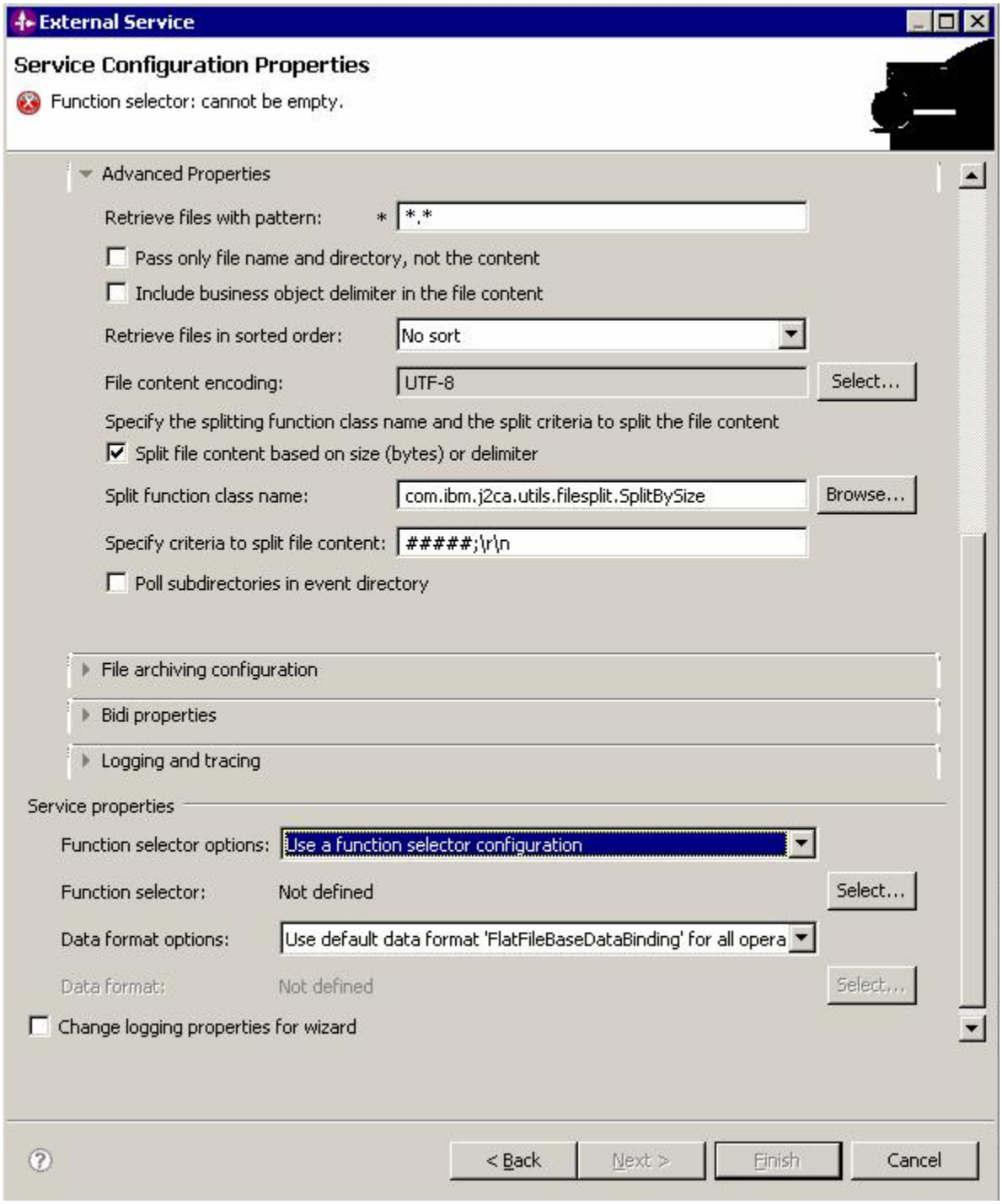
Setting properties for the external service wizard

- In the screen capture shown below, specify values for **Event directory** and **Archive directory**. **Event recovery table name** and **Event recovery data source (JNDI) name** can be left blank in which case in-memory processing of file is carried out.
- Set the splitting configuration for the inbound mode of processing. A splitting implementation available with the adapter, i.e. SplitByDelimiter is being used in this illustration. To view the implementations available with the adapter, click on **Browse** button alongside **Split function class name** label and type '?' as shown in screen capture below. A list will be displayed containing the implementations of Splitting class provided with the adapter.

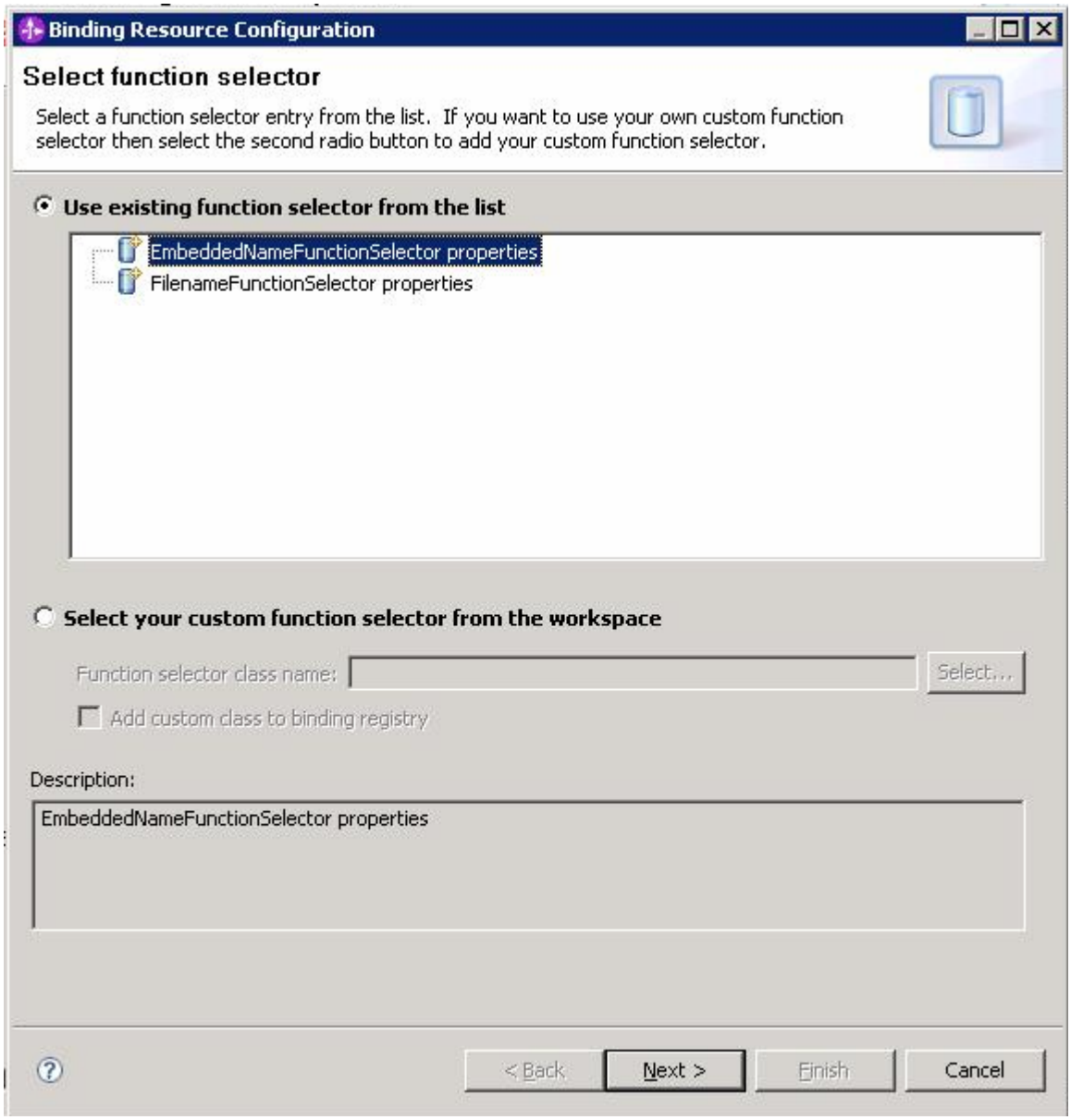

Service Configuration Properties
For this service, specify security and connection configuration properties.
Event directory:*
c:\flatfile\event
Browse...
Rule Editor:
Property Type
Operator
Value
Add...
Edit...
Remove
<< Advanced
Event polling configuration
Event delivery configuration
Event persistence configuration
Advanced Properties
Retrieve files with pattern:
.
Pass only file name and directory, not the content
Include business object delimiter in the file content
Retrieve files in sorted order:
No sort
File content encoding:
UTF-8
Select...
Specify the splitting function class name and the split criteria to split the file content
Split file content based on size (bytes) or delimiter
Split function class name:
com.ibm.j2ca.utils.filesplit.SplitByDelimiter
Browse...
Specify criteria to split file content:
#####;\r\n
Poll subdirectories in event directory
< Back
Next >
Finish
Cancel



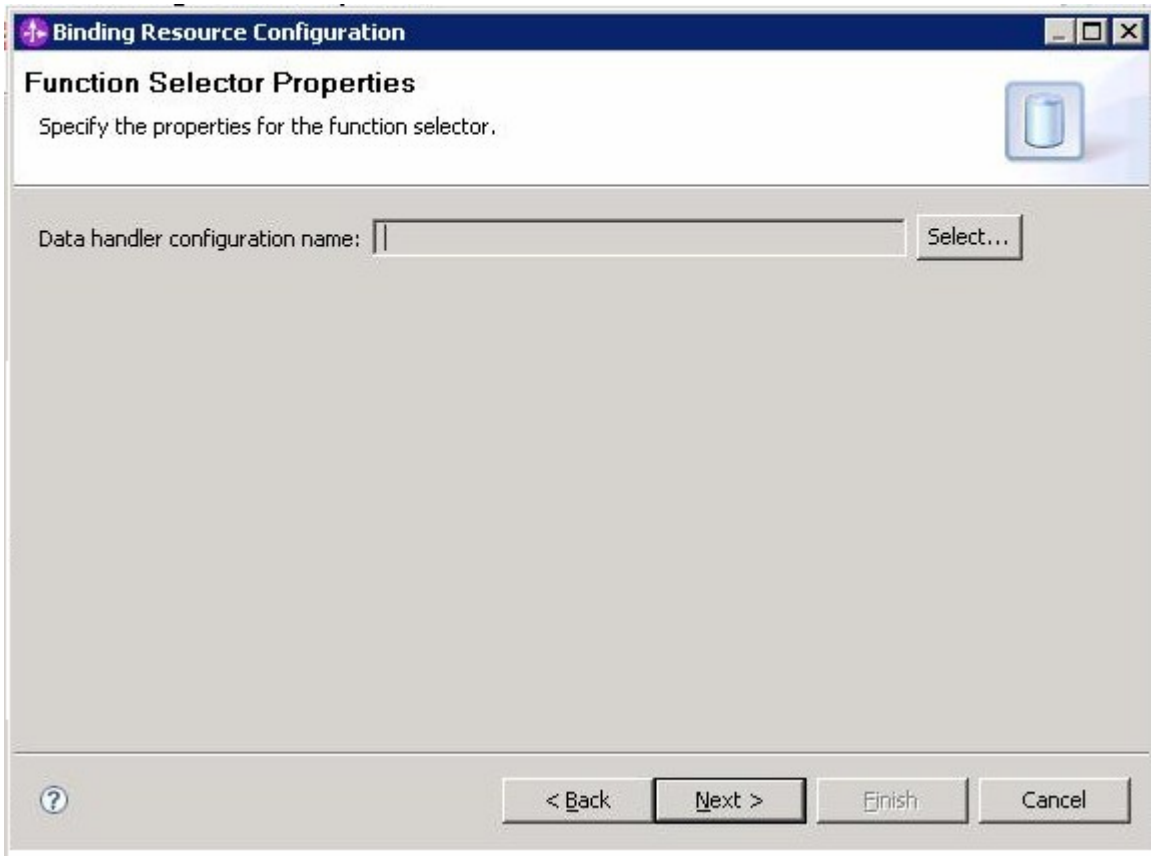
- Configure the function selector which determines which method is to be called at end-point during event delivery. In **Function Selector** list box choose **Use a function selector configuration**. Click **Select** button alongside the label **Function selector configuration**.
- In the screen capture that follows, click **Next**.



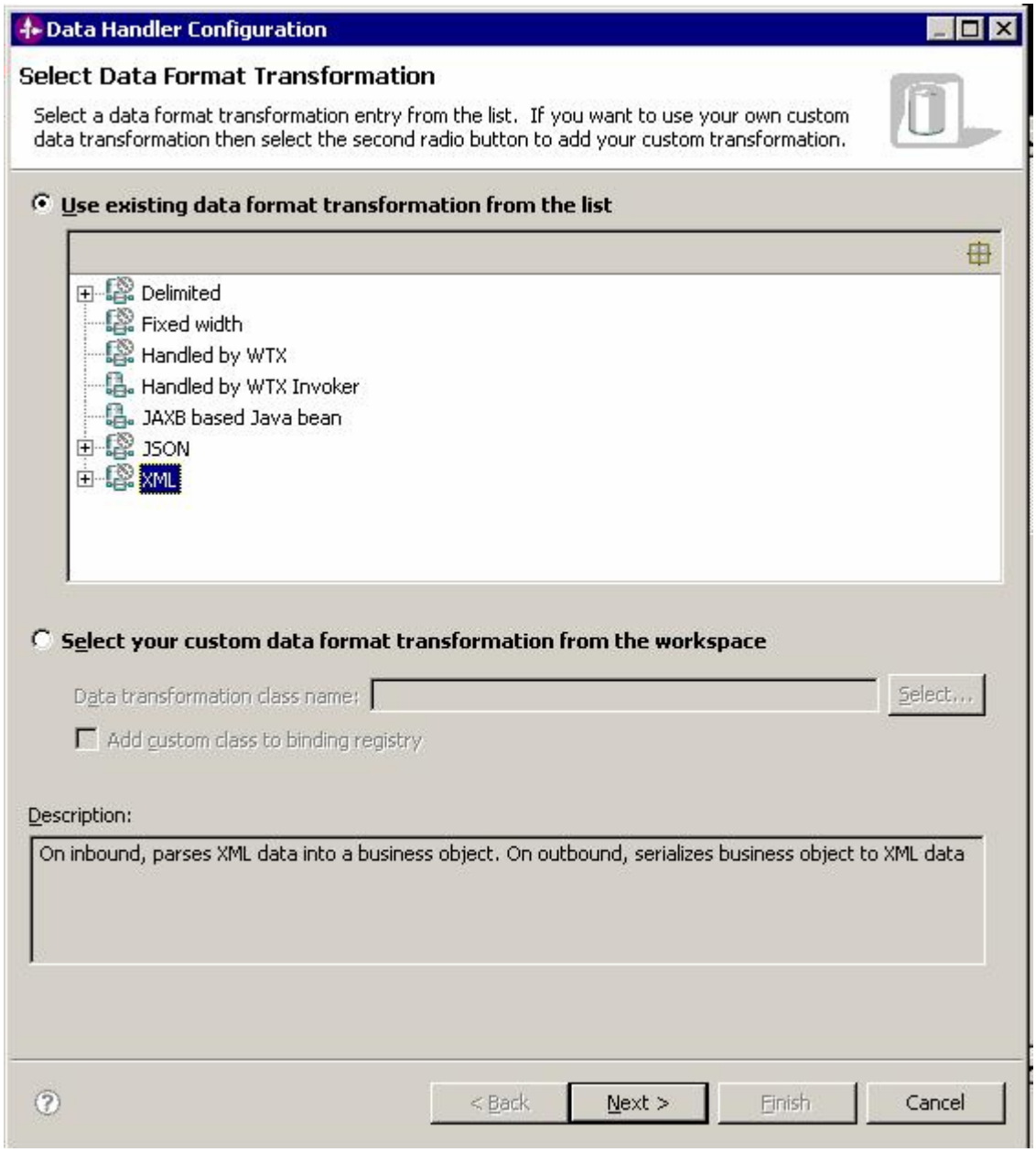
- In the Select function selector window, select **EmbeddedNameFunctionSelector** from the list and click **Next**.



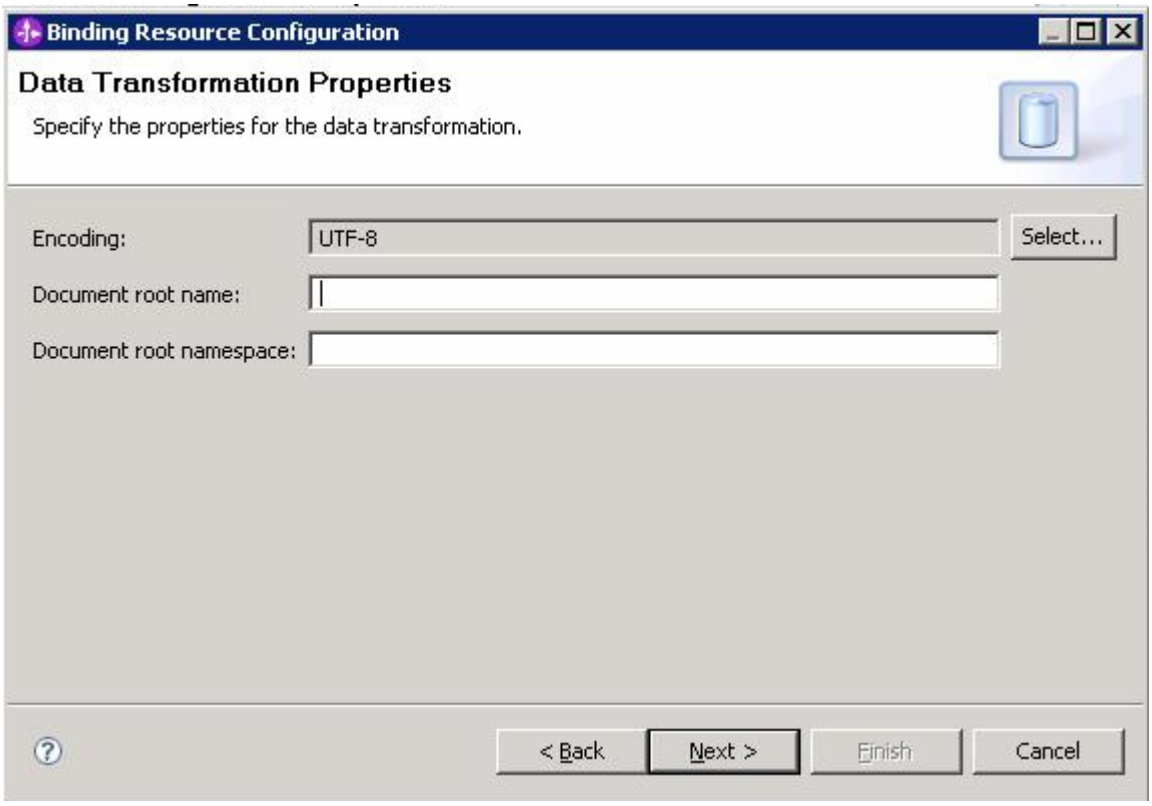
- Since this is a scenario with XML Data handler, click **Browse** alongside the label **Function selector class name**. In the pop-up dialog, choose **EmbeddedNameFunctionSelector.** Click **OK** and click **Next**.



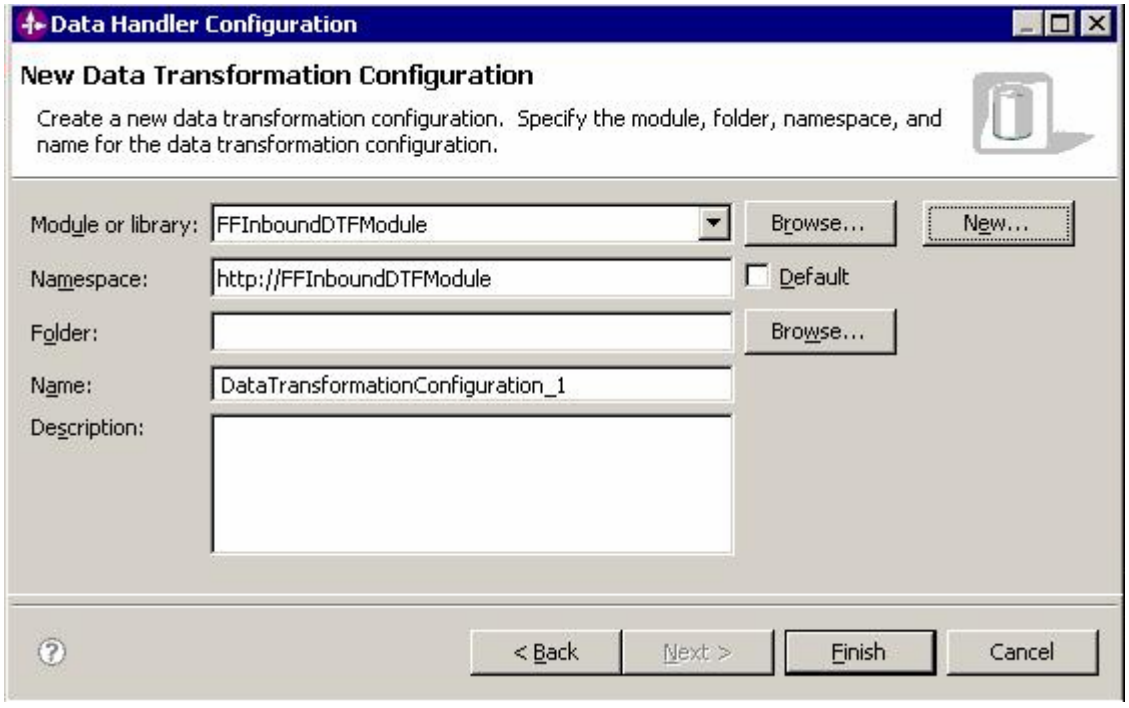
- Also, configure a Data handler capable of handling XML data. Click **Select** button alongside **Data handler configuration name**.



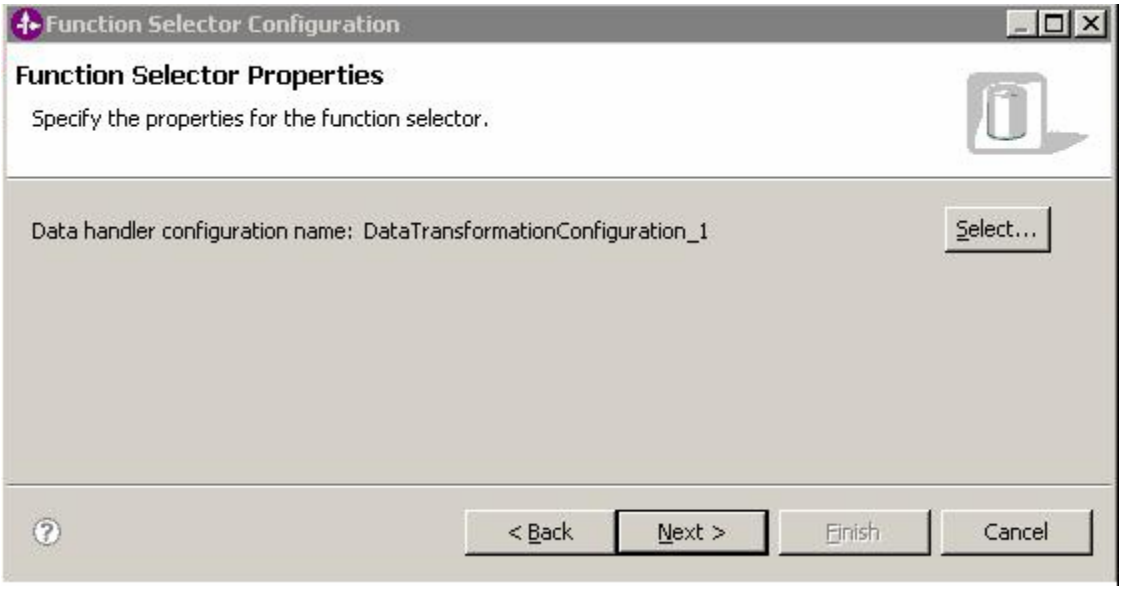
- Browse and choose **XML Data handler** in the next window. Retain the default name for **Data handler configuration** or if you choose you can specify a different name.



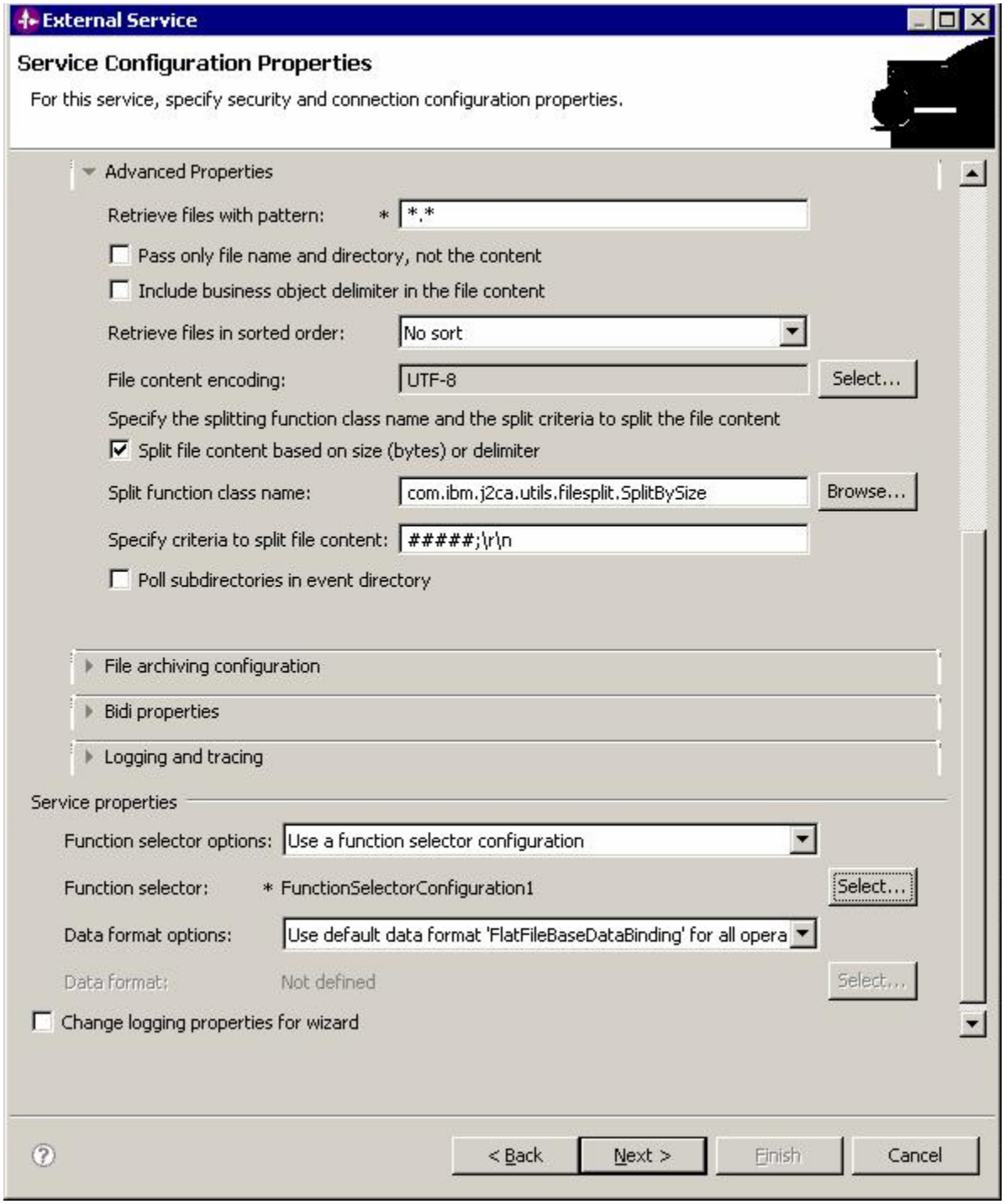
- Select the encoding, Document root name and Document root namespace and then click Next.
- Specify a name for the **Data handler configuration** or optionally default name can also be used. Click **Next**.



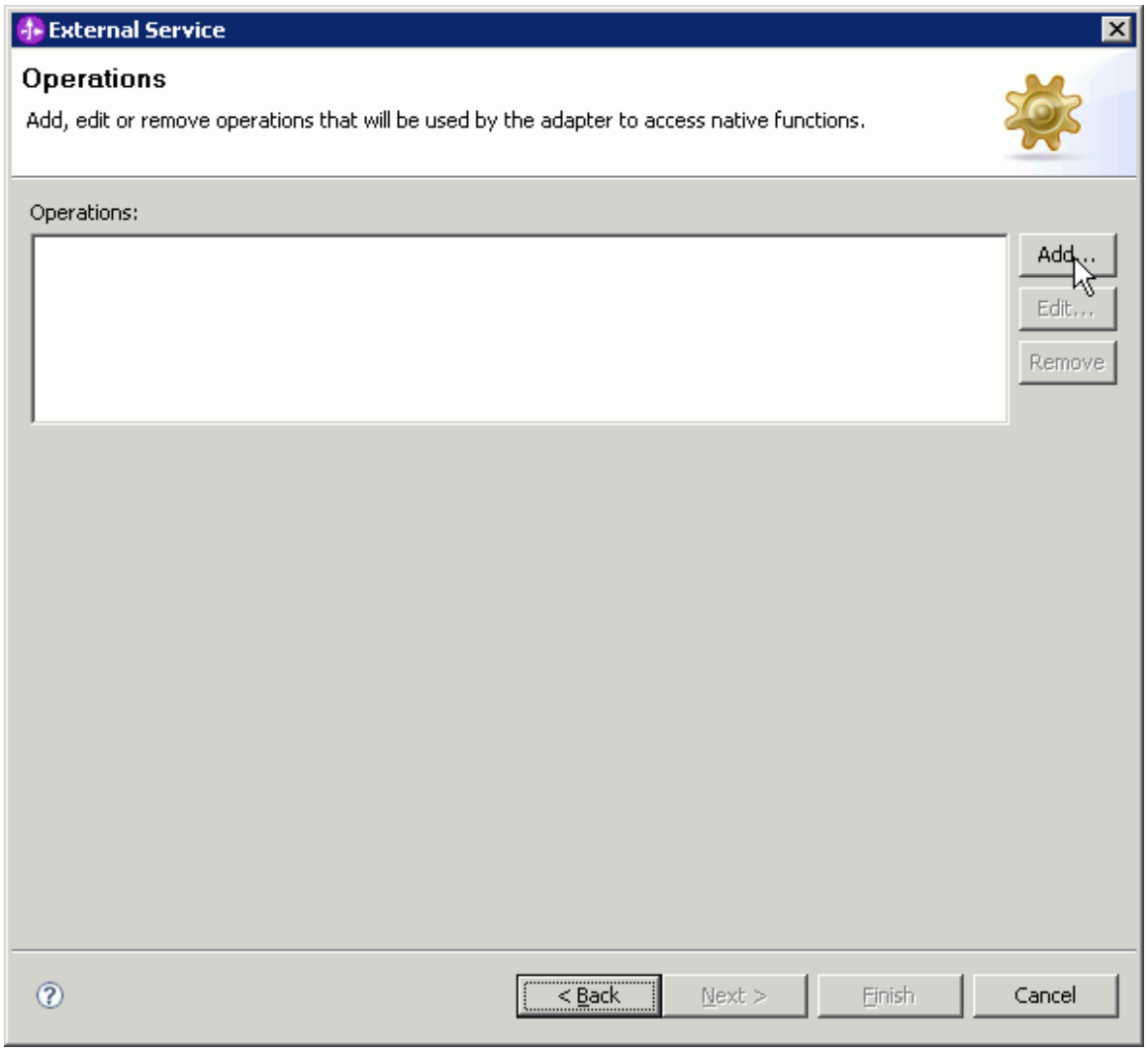
- Click Finish to exit Data handler configuration. Click Next.



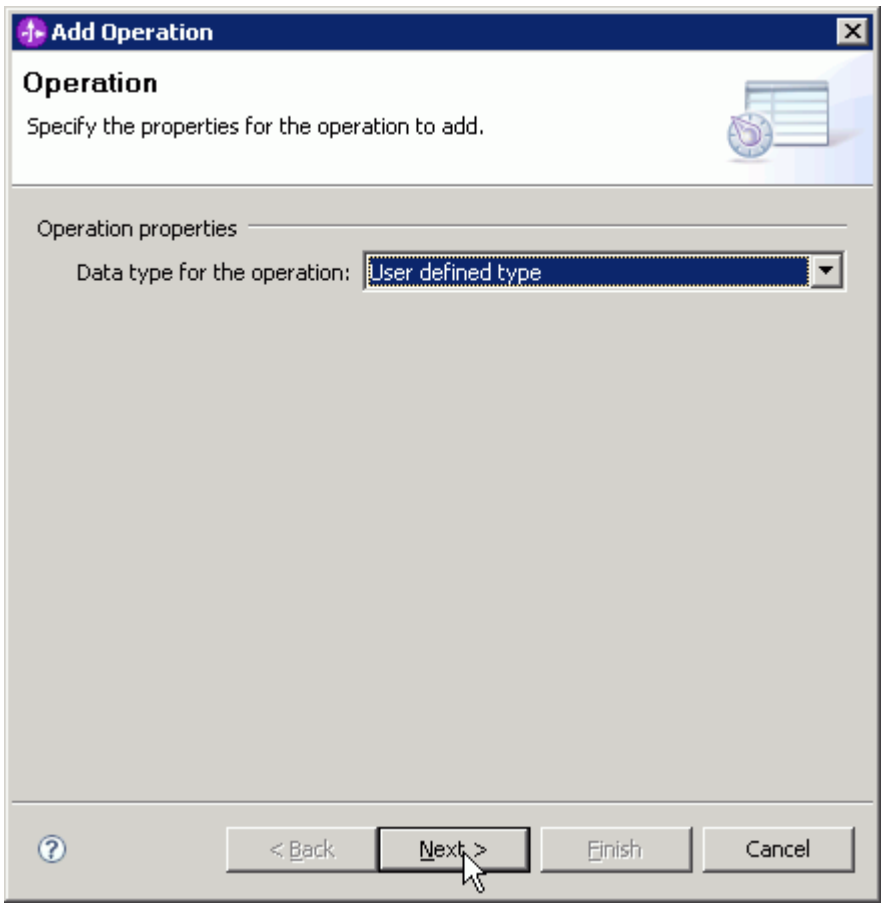
- After having configured the Function selector with XML Data handler, click **Next**.

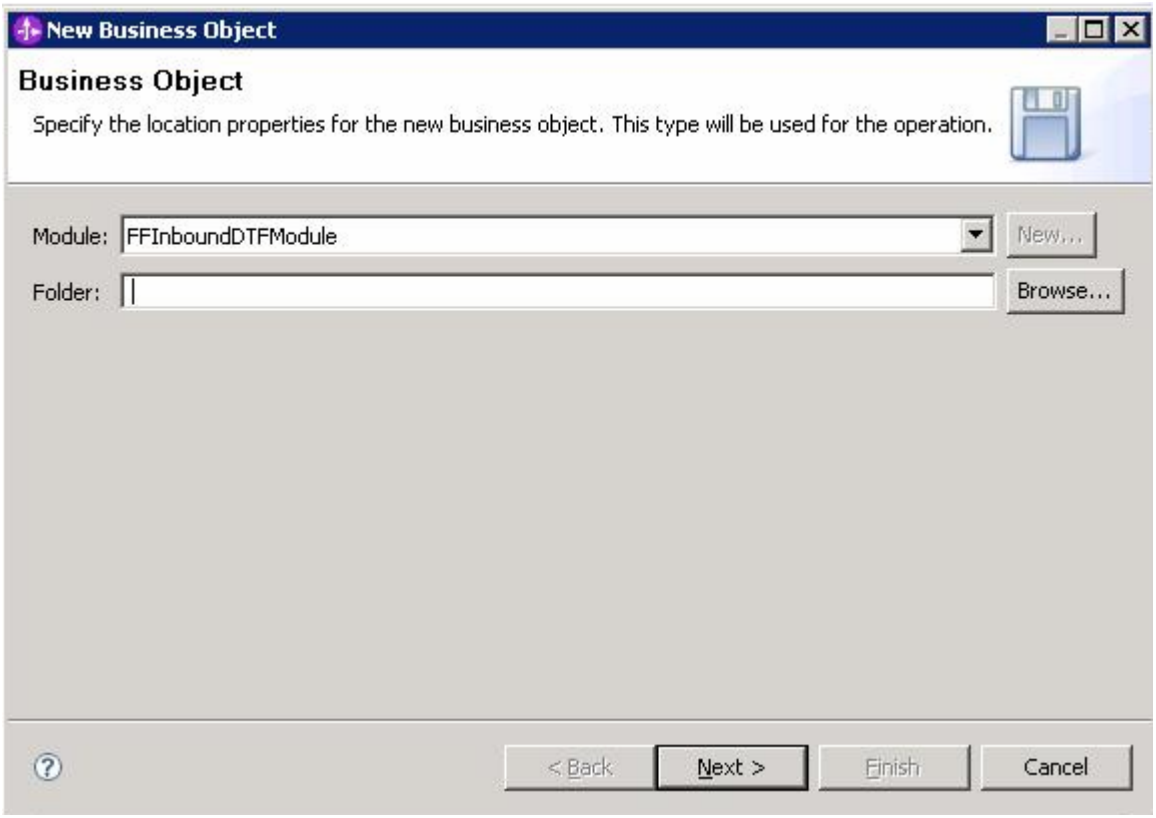


- Next, in the Operations window, click **Add** to add the operations that you want to perform.

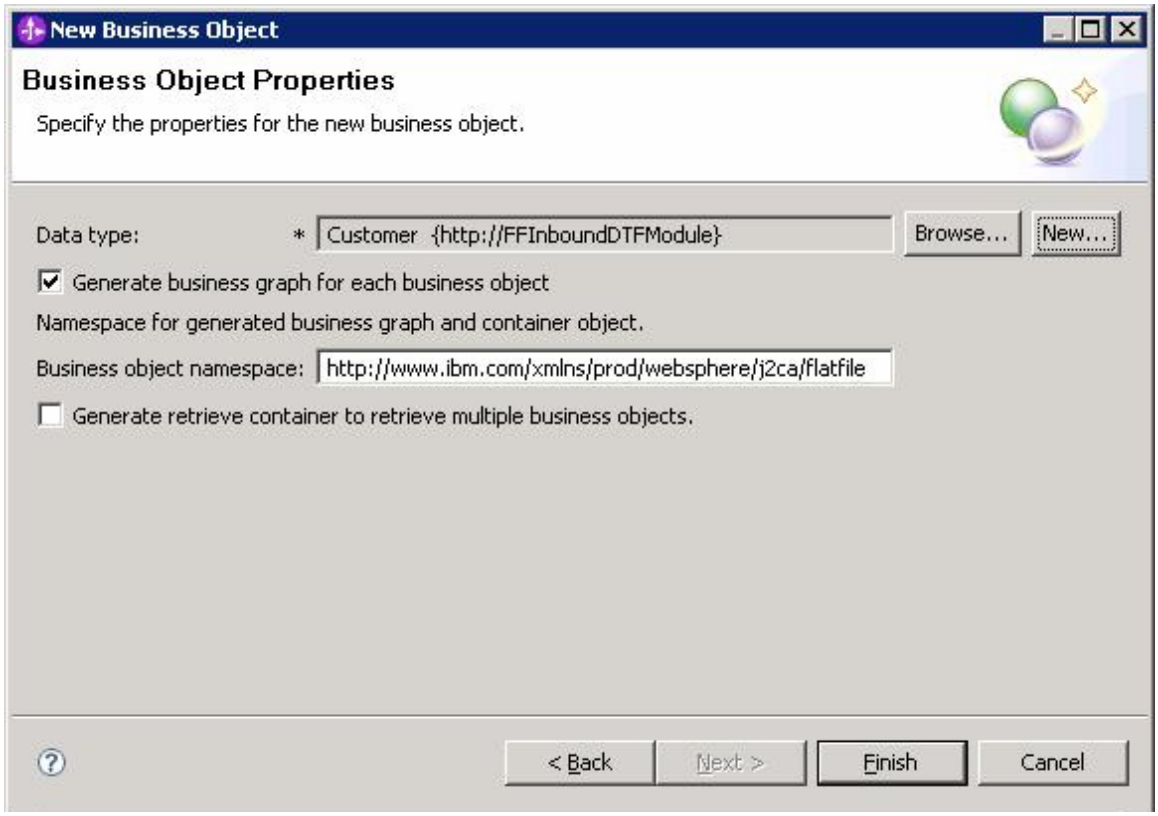


- Click **User defined type** in **Data type for the operation** as this scenario uses a custom business object. Click **Next**.

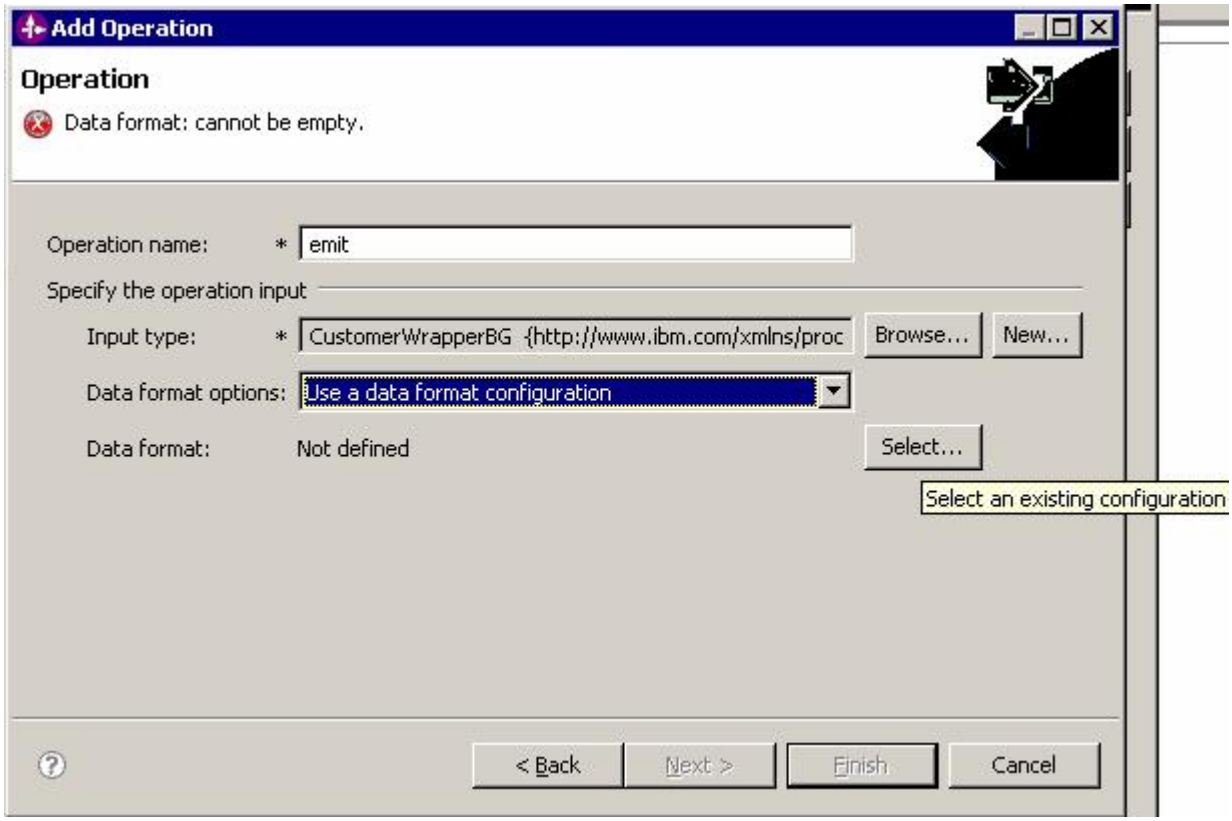

Add Operation
Operation
Specify the properties for the operation to add.
Operation properties
Data type for the operation:
User defined type
< Back
Next >
Finish
Cancel


New Business Object
Business Object
Specify the location properties for the new business object. This type will be used for the operation.
Module:
FFInboundDTFModule
New...
Folder:
Browse...
< Back
Next >
Finish
Cancel

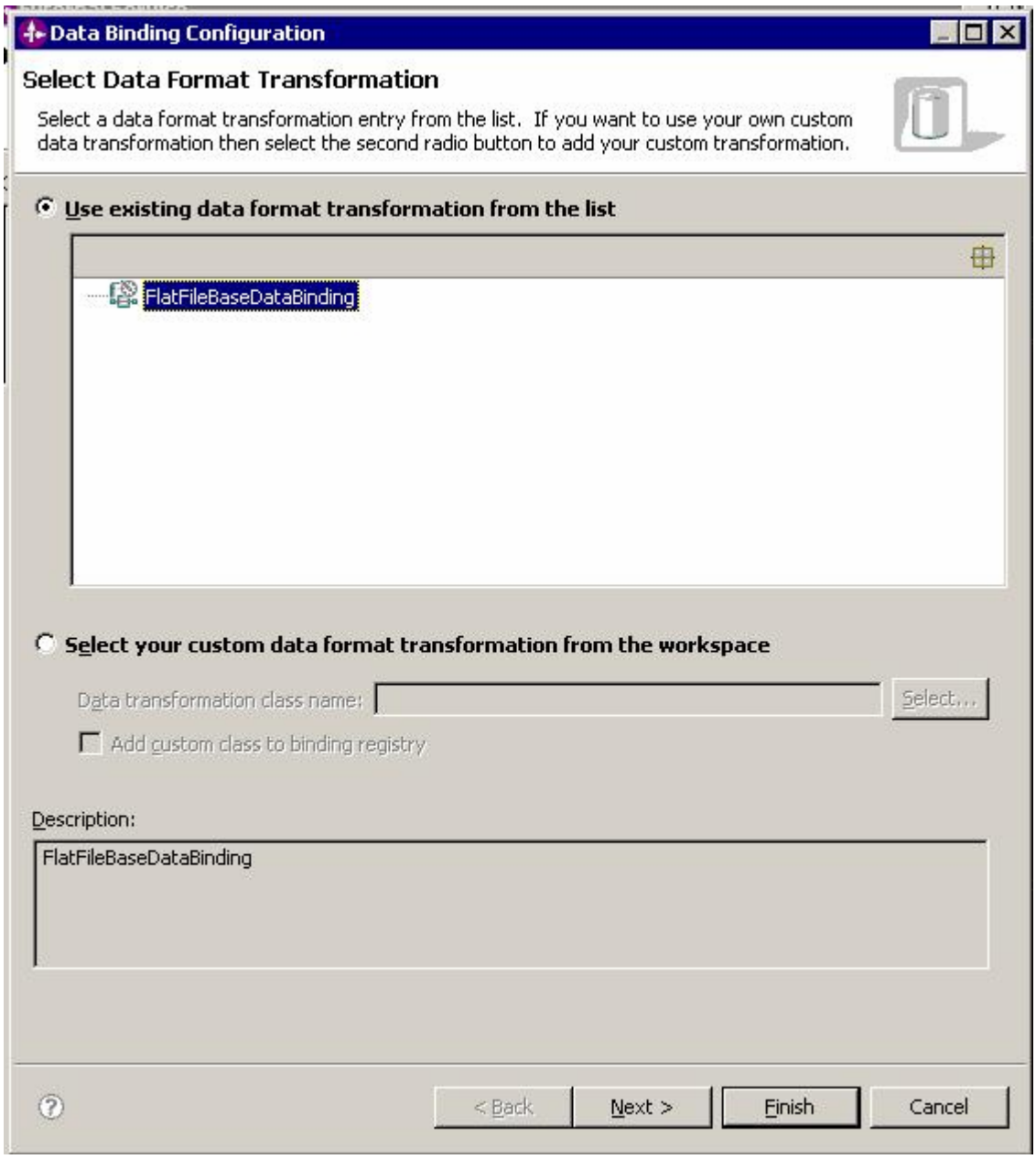
- Next, generate a wrapper business graph for the input type. Click **New** button alongside **Input type** label.
- Click **Next** in the window that follows.



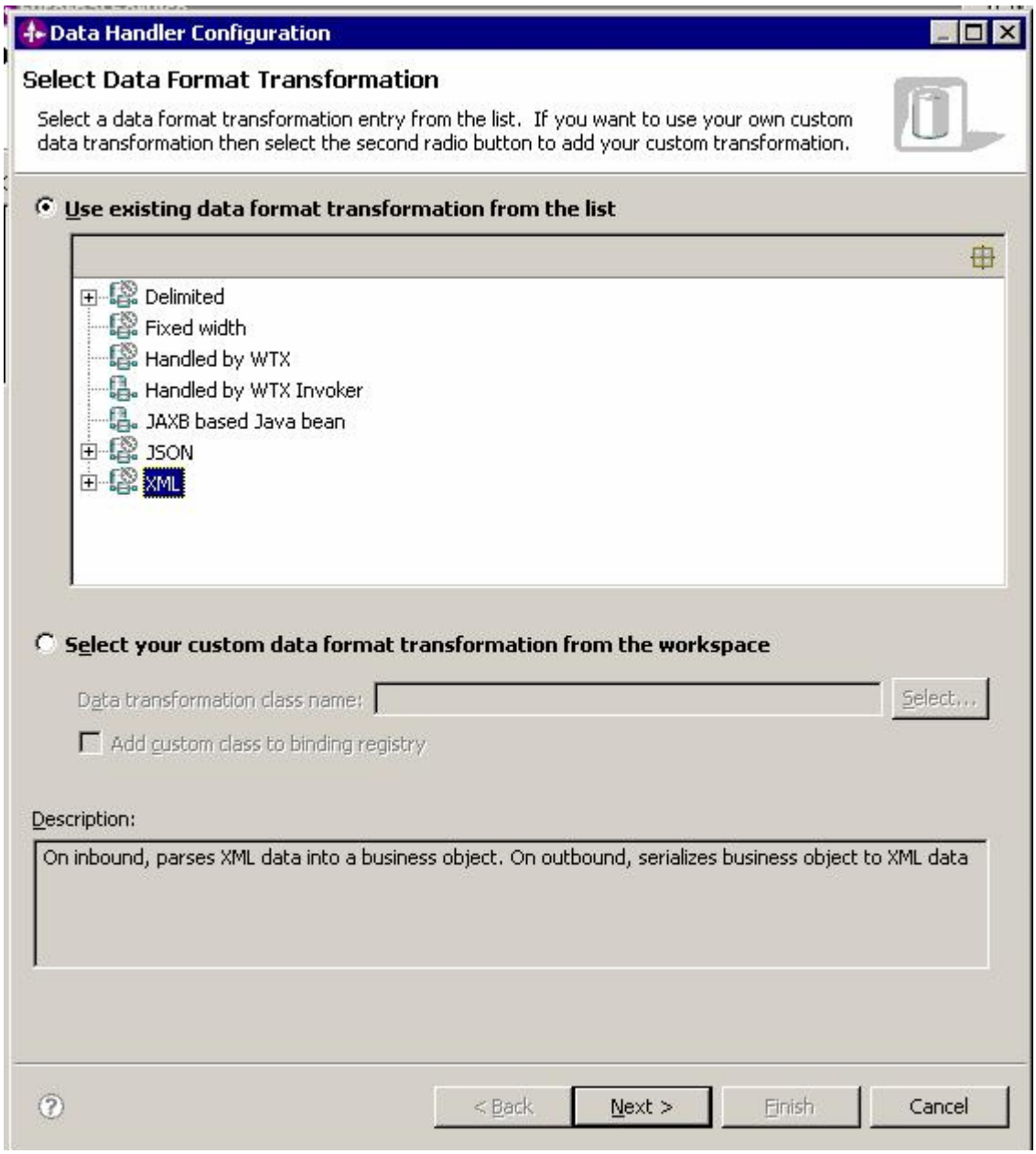
- Click **Finish**.



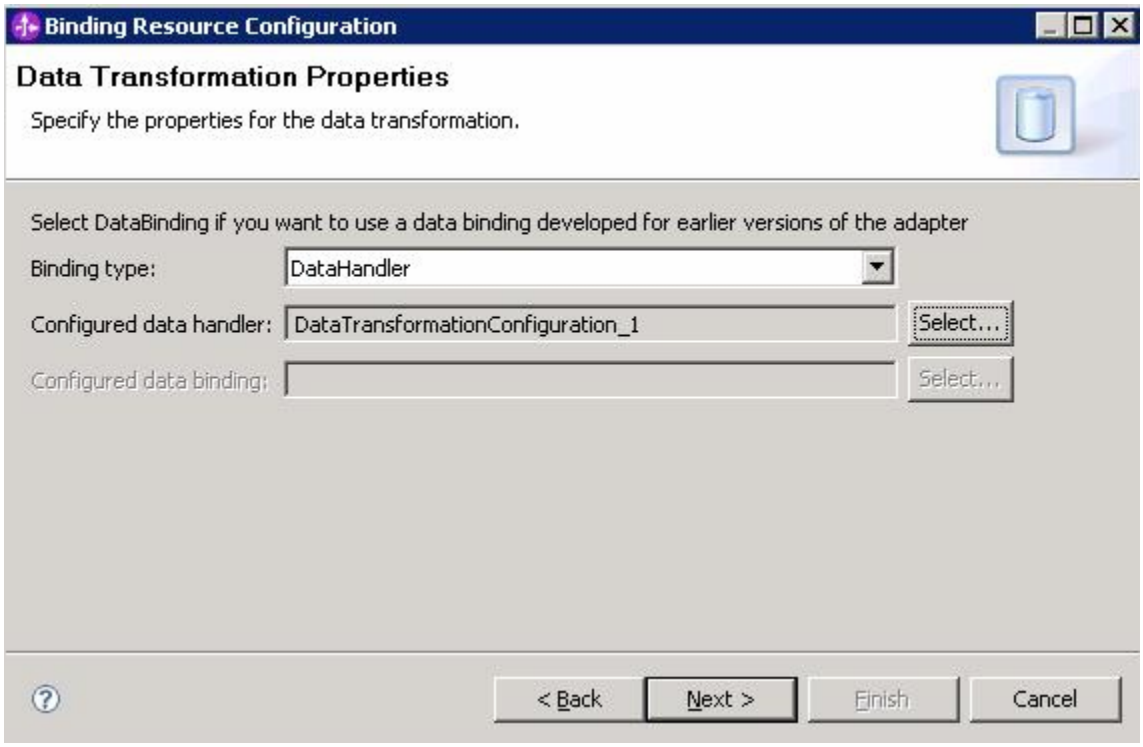
- Browse and choose **Customer** business object imported earlier on into the module (FFOutboundModule). Check on **Generate business graph for each business object** to generate wrapper (CustomerWrapperBG). Click **Finish.**
- Next configure a **Data binding type** by clicking the **Selecting Use a data format configuration** button alongside **Data format options** label, and then clicking select button alongside the **Data format** lable.
- Click **Next** in the window that follows.



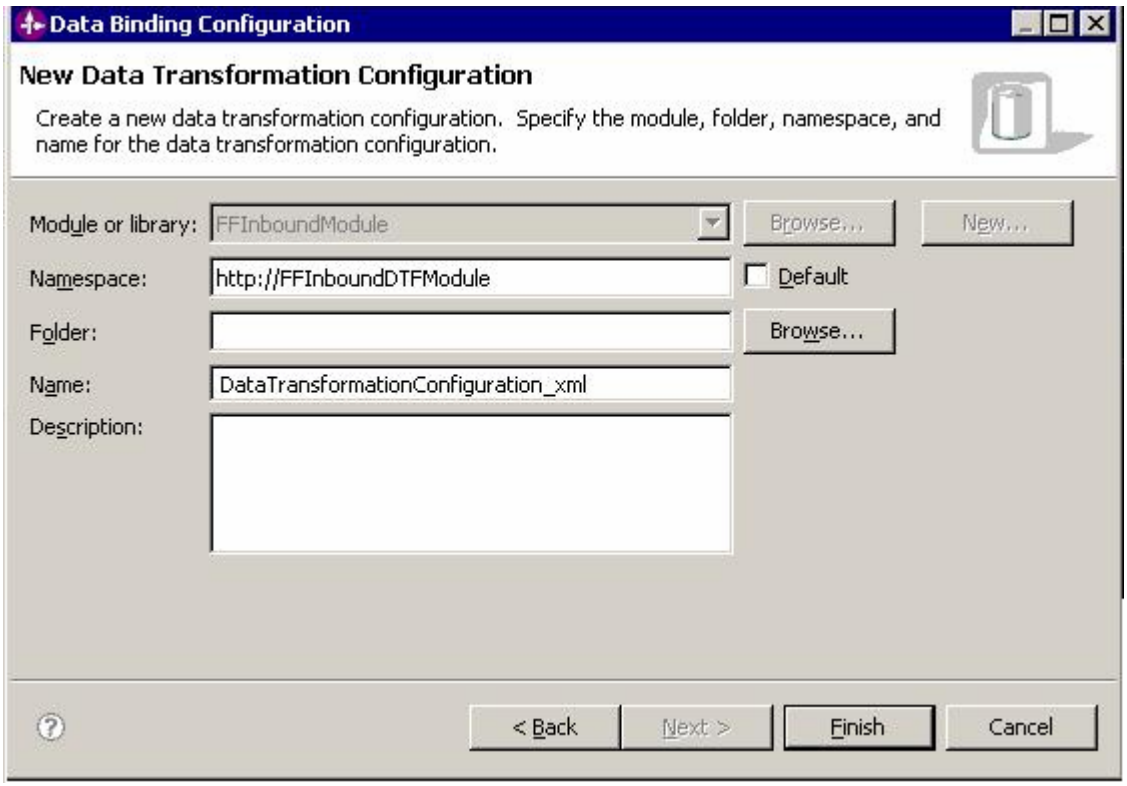
- Choose the **FlatFileBaseDataBinding** available and click **Next**.
- The same Data handler configuration created earlier on can be used. **Browse** and choose **XML-> DataHandlerConfiguration_1**, by clicking the + sign against XML.



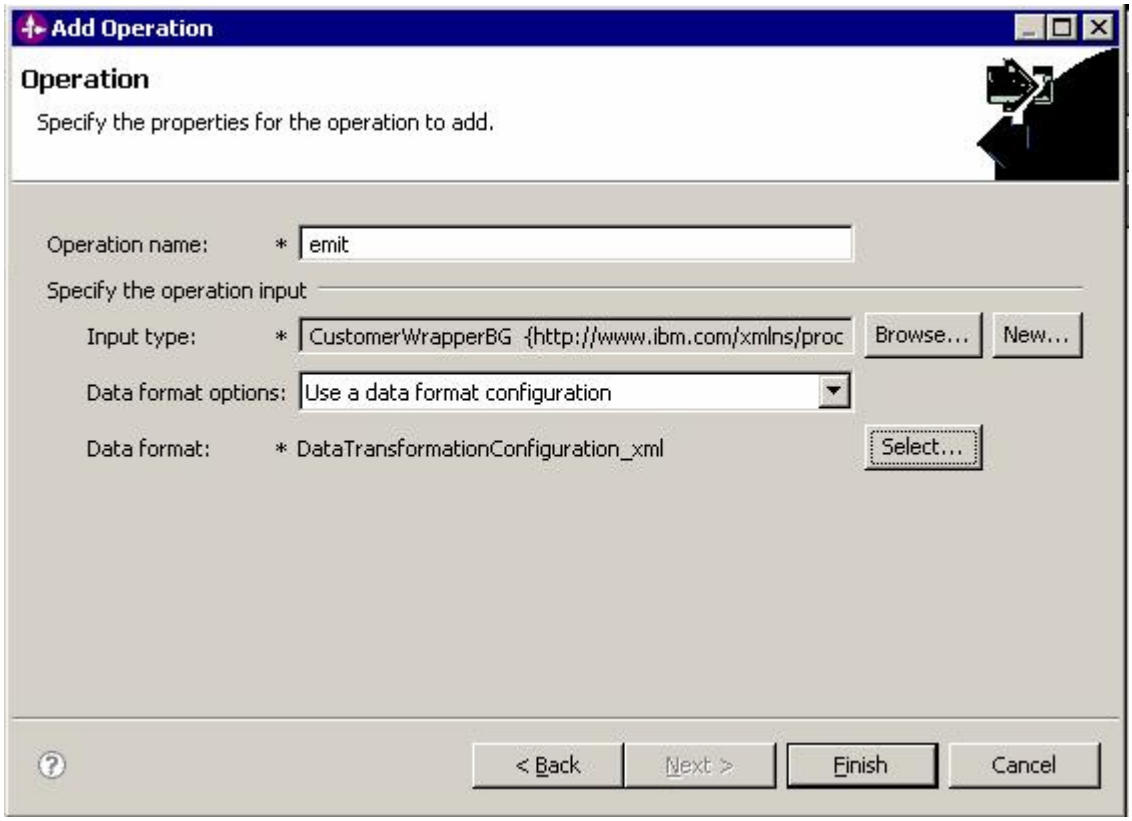
- Click **Finish** to exit Operation window.

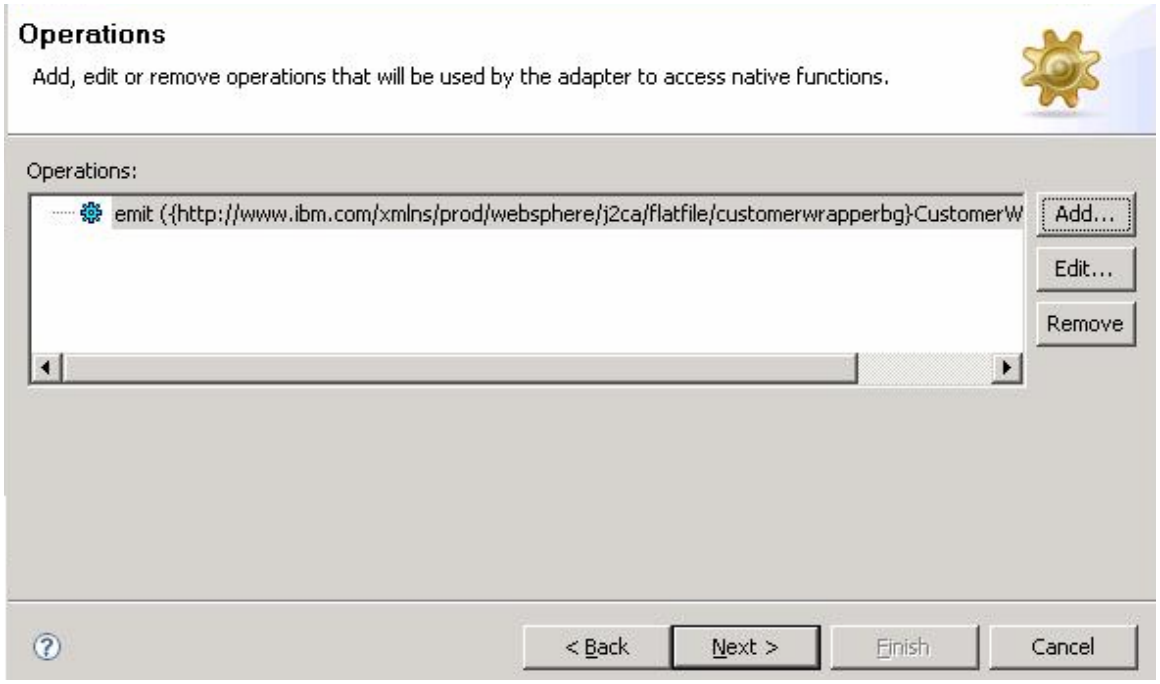


- Click **Next** in the window that follows.

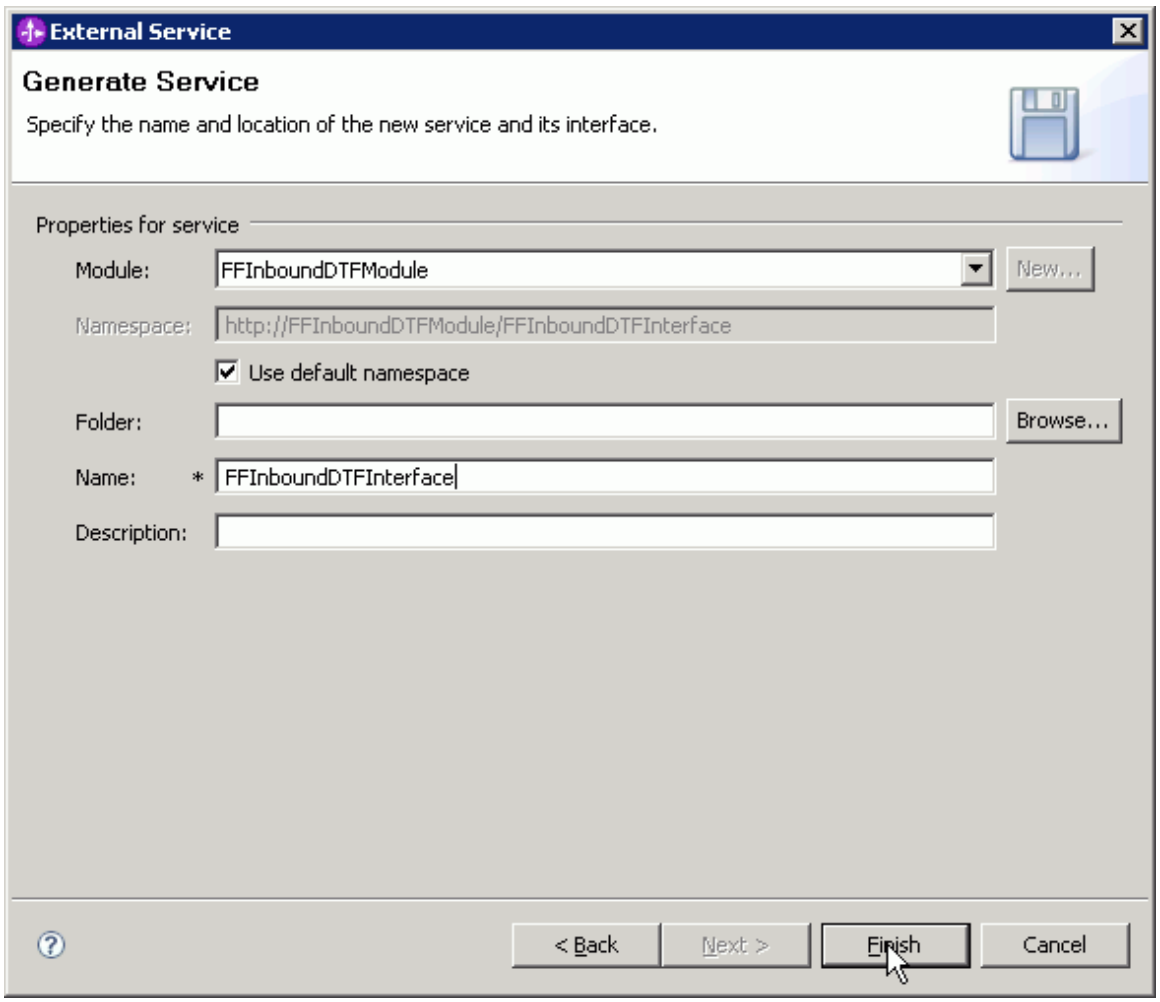


- Click **Finish**.



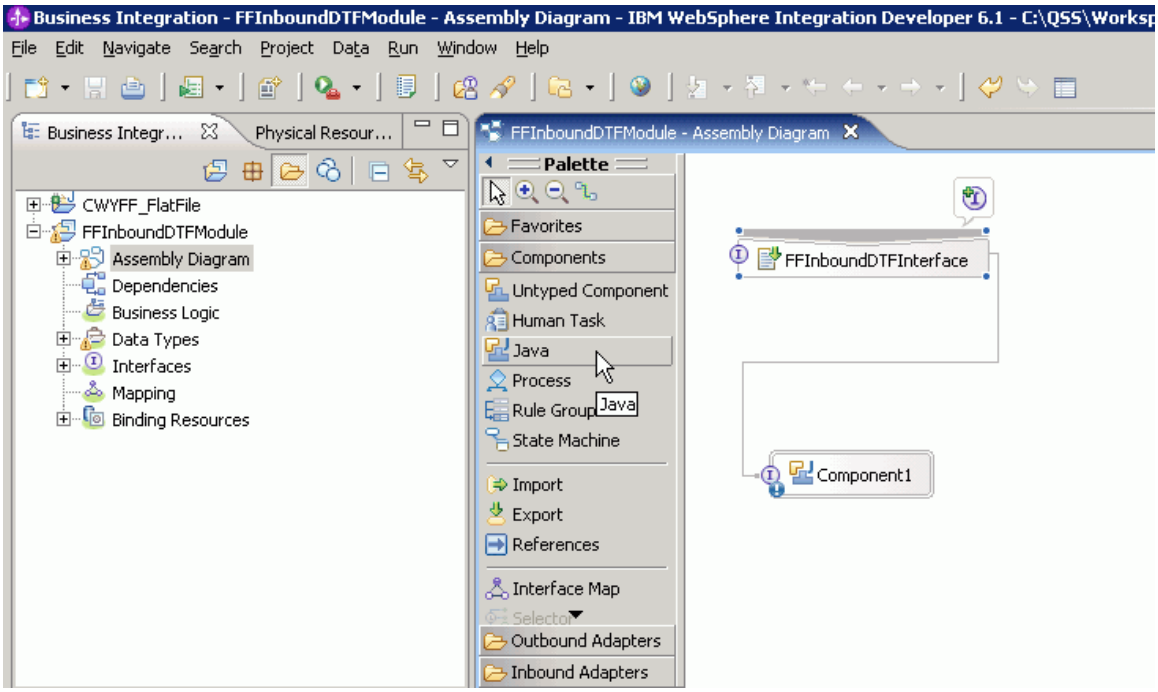


- In the last step of running External service wizard, specify a suitable name for your adapter interface. In this tutorial, FFInboundDFTInterface has been set as the value for the **Name** field. Click **Finish**.

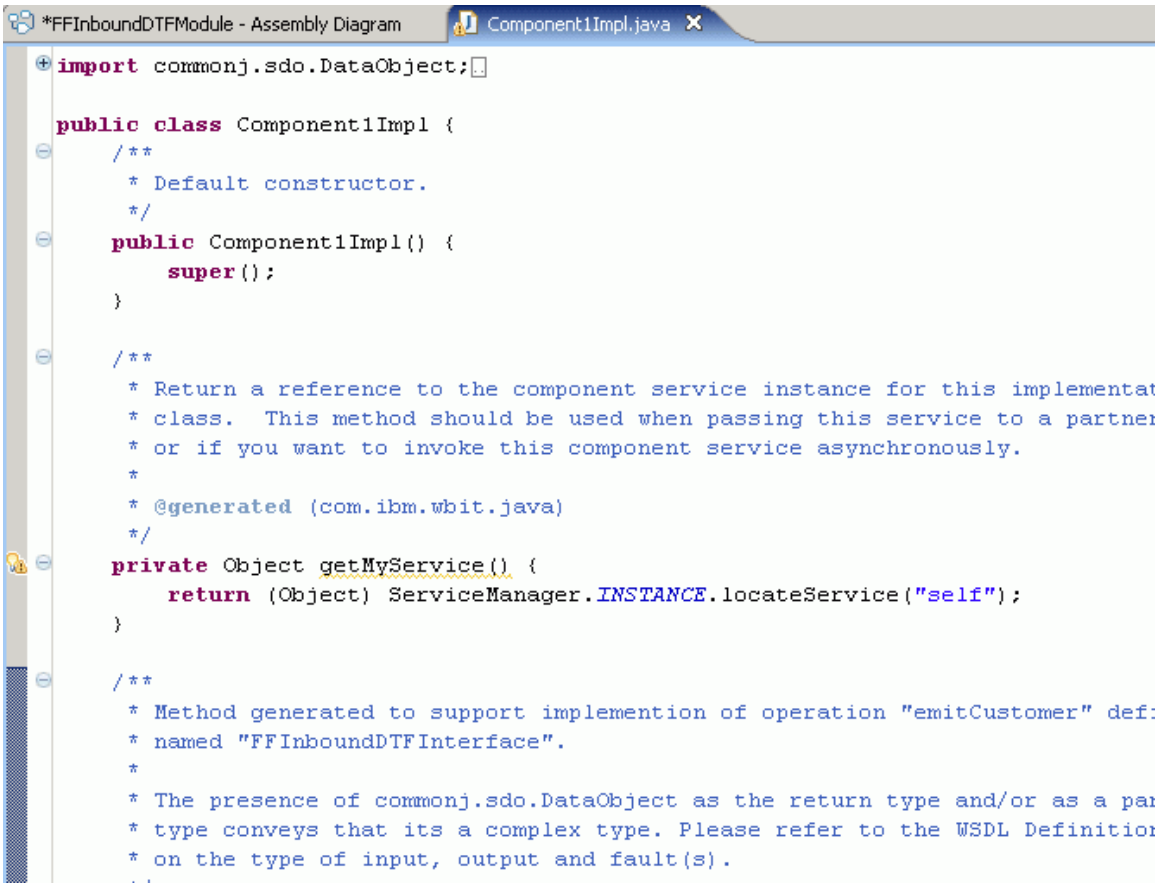


Generating business object definitions and related artifacts

- Generate a component of Java type and draw a wire from **FFInboundDTFInterface** to **Component1**
- Implement the Java component by right clicking Component1 and clicking on **Generate Implementation**. You can choose to have the implementation belong to default package or any other package.



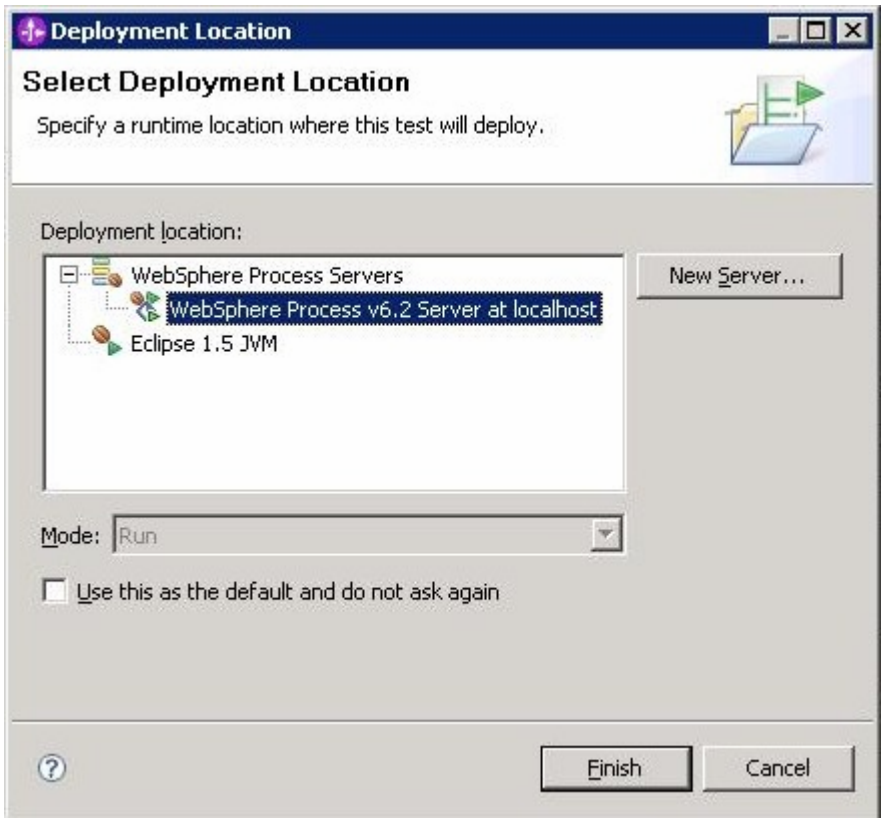
- A window containing a code snippet of the generated implementation of Component1 is shown below.



Deploying the module to the test environment

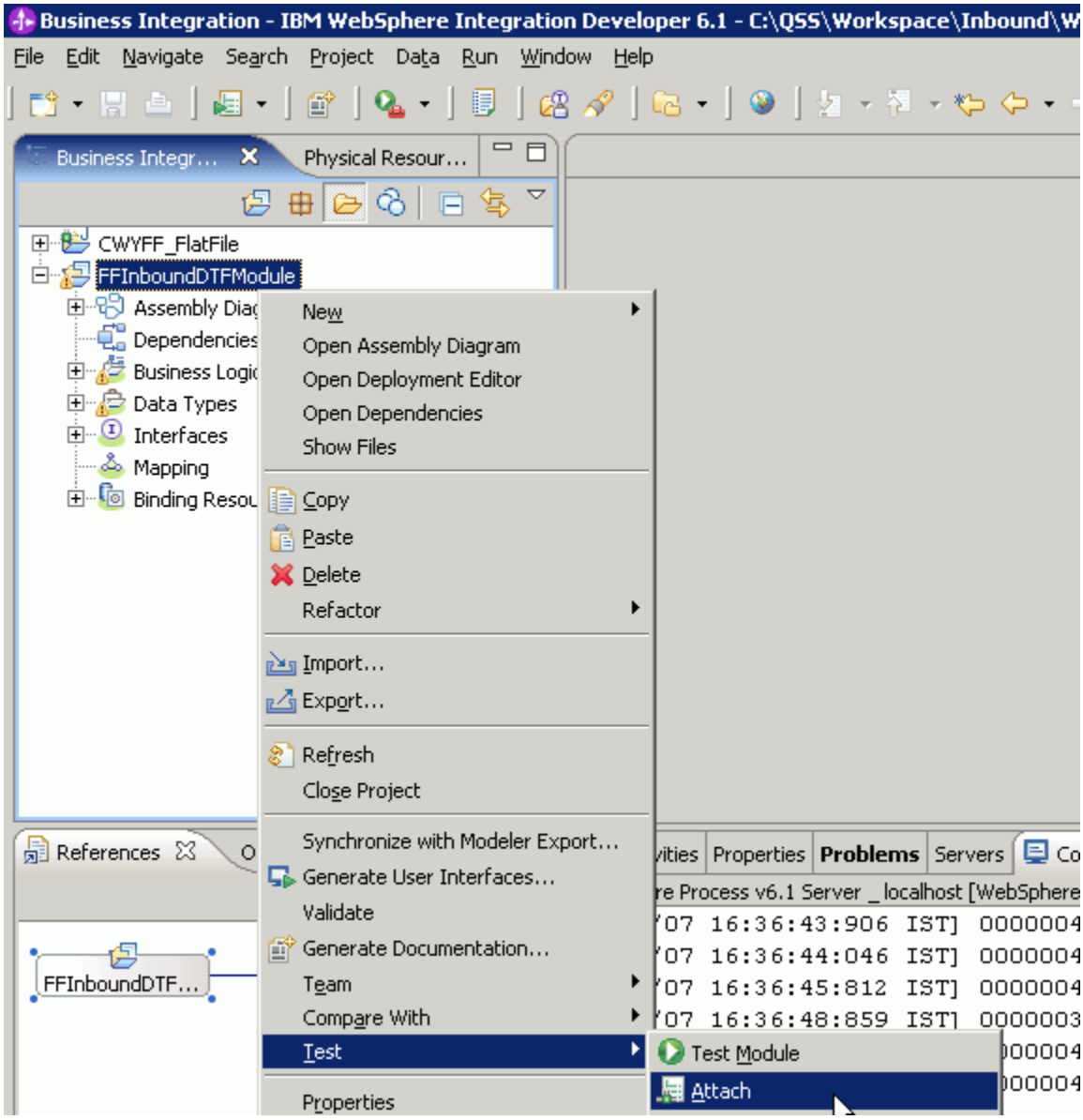
After running the external service wizard, you will have an SCA module that contains an EIS import or export. You must install this SCA module in the WebSphere Integration Developer integration test client.

- Add the SCA module to the server using the server panel in WebSphere Integration Developer. Right-click on the **server**, and select **Add and remove projects**.
- Add the SCA module to the server.



Testing the assembled adapter application

- Right click on the adapter module, **FFInboundDTFModule** and then **Test** -> **Attach**
- Copy a sample event file which complies with XML format to the specified Event directory
- The relevant business object (after chunking the content) will be delivered to the end-point
- This can be verified by checking for the end-point messages in System.Out file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer


Business Integration - IBM WebSphere Integration Developer 6.1 - C:\QSS\Workspace\Inbound\W
File Edit Navigate Search Project Data Run Window Help
Business Integr...
Physical Resour...
CWYFF_FlatFile
FFInboundDTFModule
Assembly Diag
Dependencies
Business Logic
Data Types
Interfaces
Mapping
Binding Resou
New
Open Assembly Diagram
Open Deployment Editor
Open Dependencies
Show Files
Copy
Paste
Delete
Refactor
Import...
Export...
Refresh
Close Project
Synchronize with Modeler Export...
Generate User Interfaces...
Validate
Generate Documentation...
Team
Compare With
Test
Properties
Test Module
Attach
References
FFInboundDTF...
Properties
Problems
Servers
re Process v6.1 Server _ localhost [WebSphere
'07 16:36:43:906 IST] 0000004
'07 16:36:44:046 IST] 0000004
'07 16:36:45:812 IST] 0000004
'07 16:36:48:859 IST] 0000003

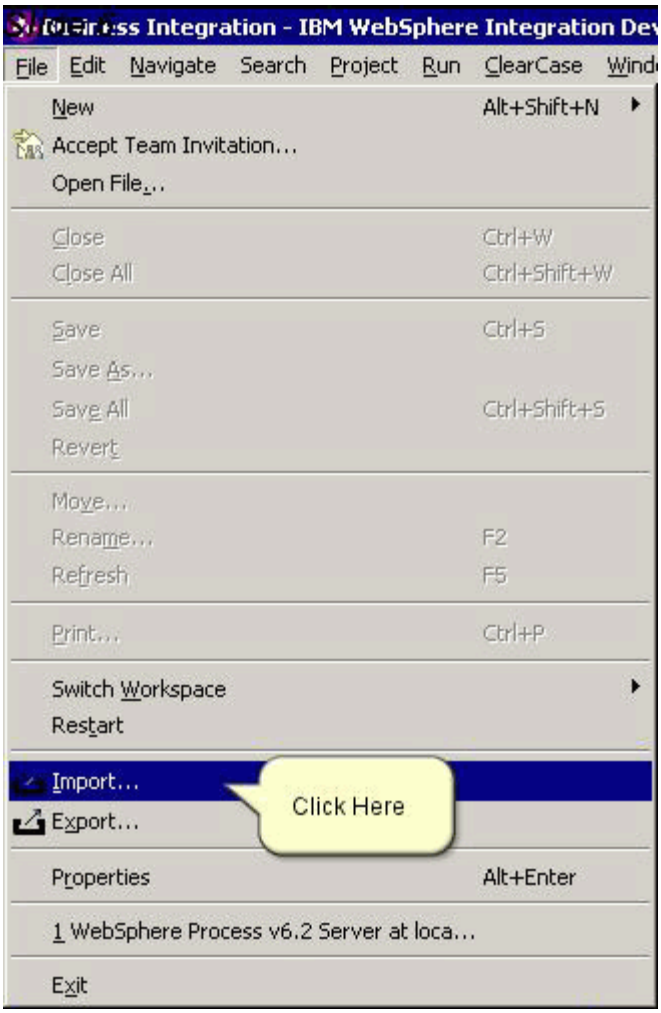
Chapter 8. Tutorial 6: Outbound processing – Creating and retrieving content using Cobol CopyBook records

This tutorial demonstrates how WebSphere Adapter for FlatFiles 6.1.0 can be used to create a file on a file system as well as retrieve content from a file located on local file system. In addition, it demonstrates as to how XML Data handler can be used while retrieving XML content from a file.

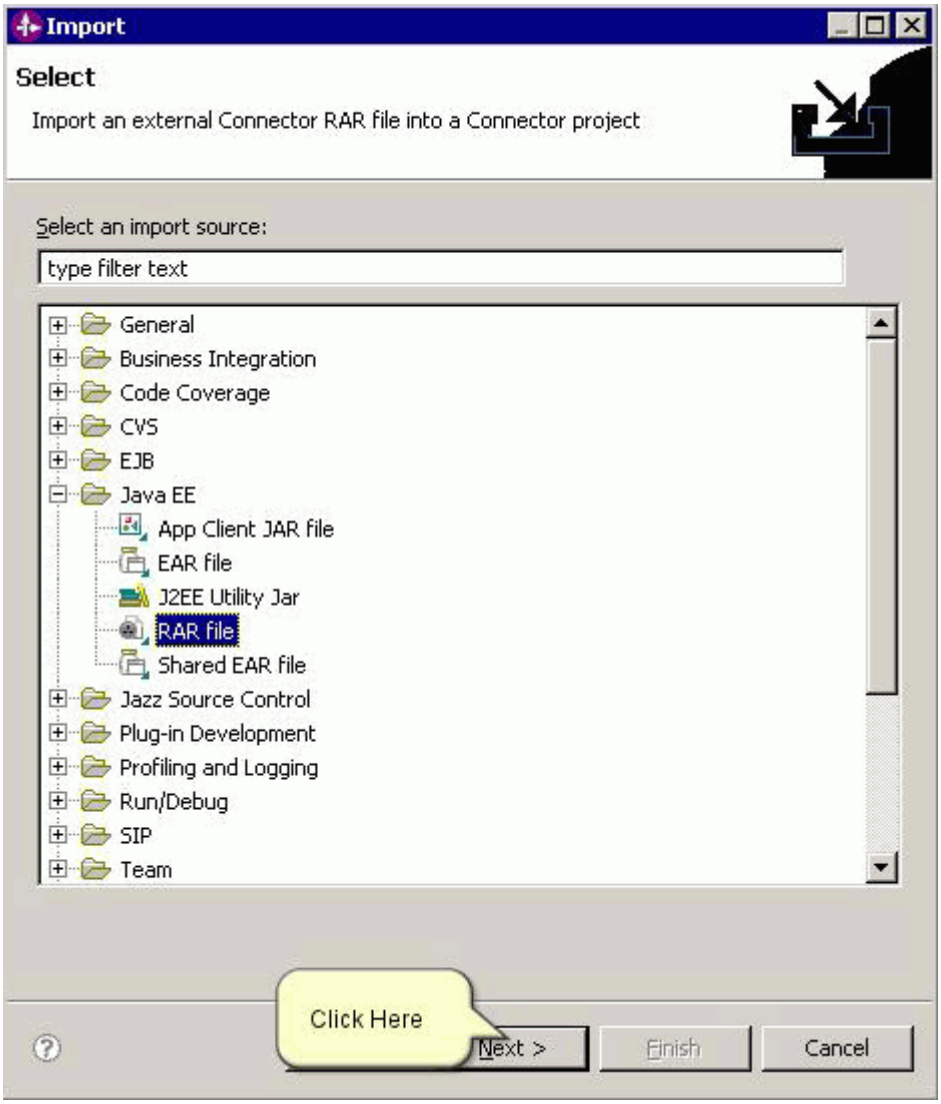
Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

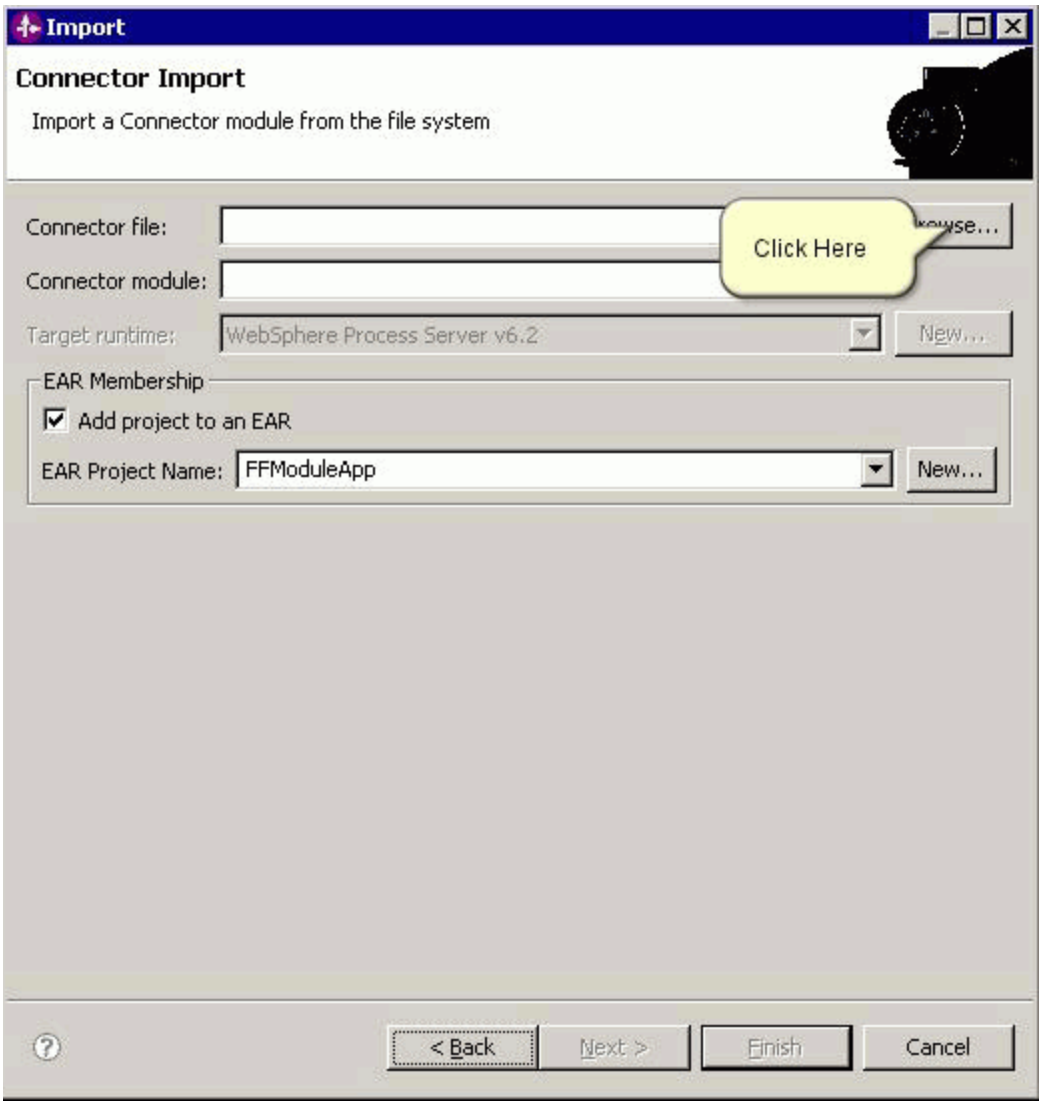
- Import the Flat File RAR by clicking **File → Import**



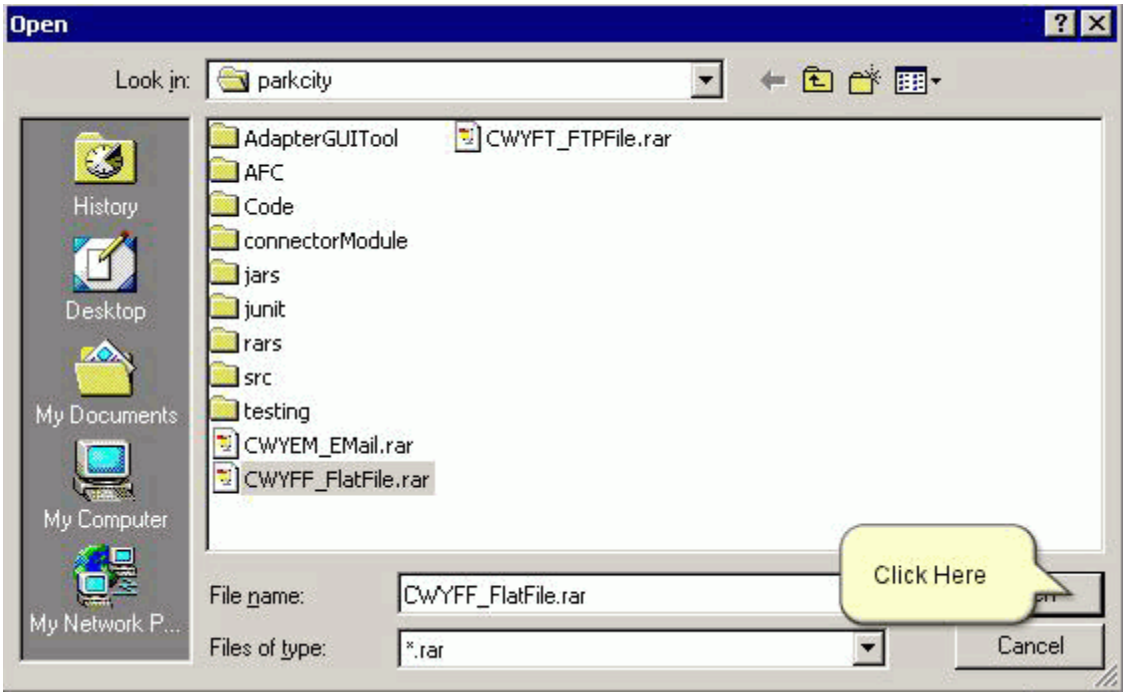
- On the Import window, expand **Java EE**, choose **RAR** and click **Next**.



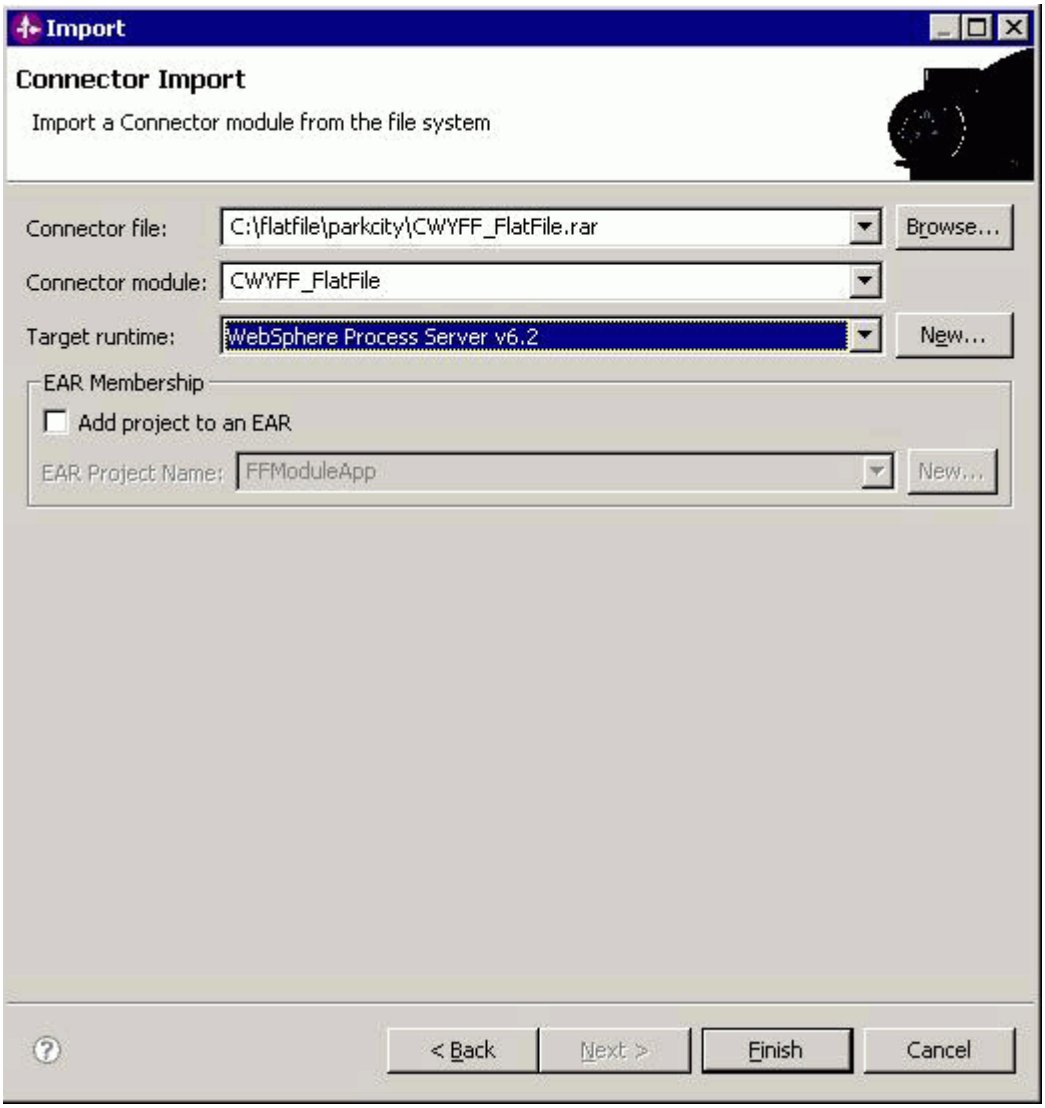
- In the next window, to import the RAR file, click the **Browse** button.



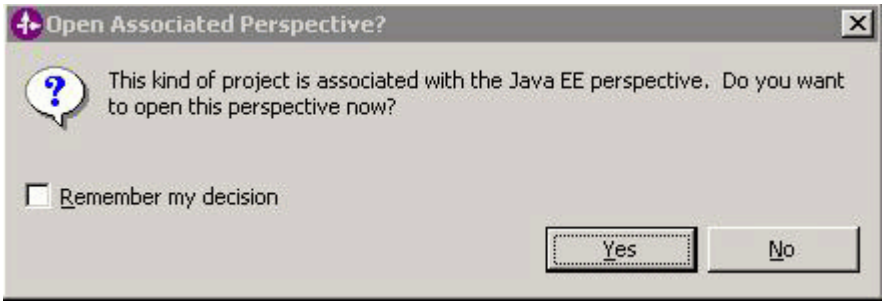
- Browse to the path where the Flat File RAR is kept, select it and click **Open**.



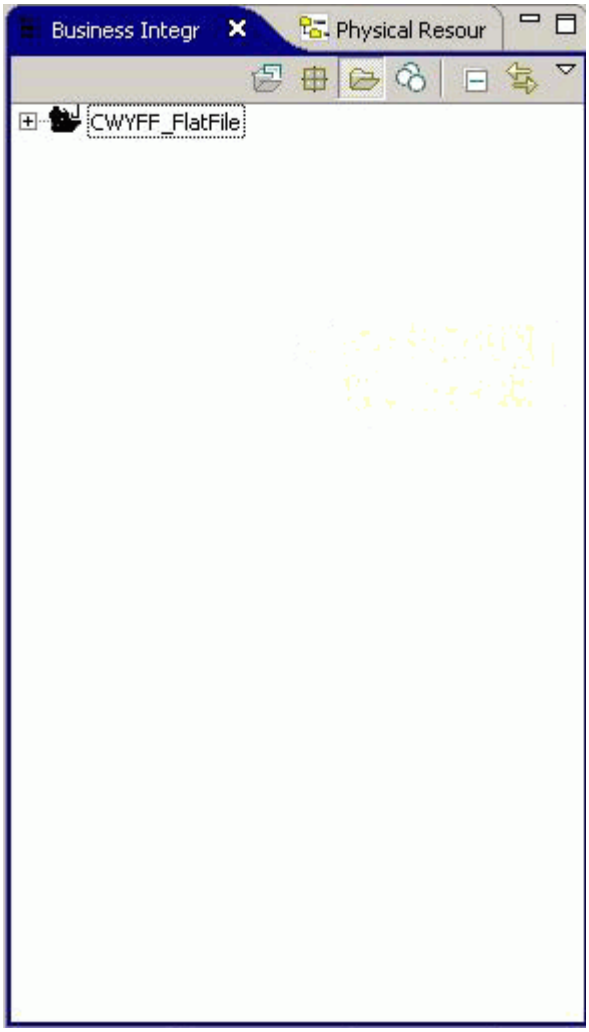
- The RAR path is now entered in the **Connector file** property. Uncheck the **Add project to an EAR** and click **Finish**. Ensure that the **Target runtime** is set to WebSphere Process Server v6.2.



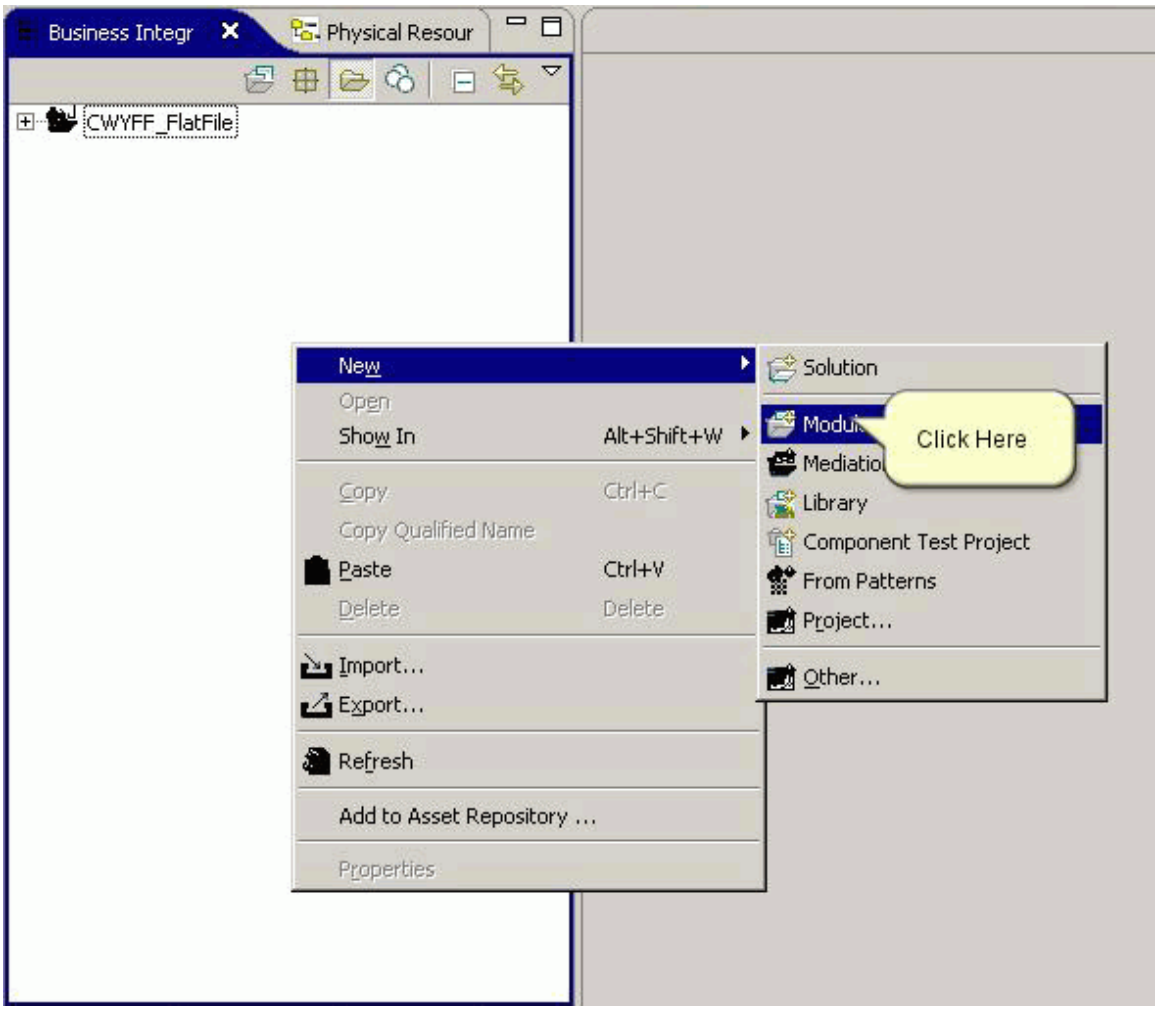
- The following dialog box appears. Click **No**.



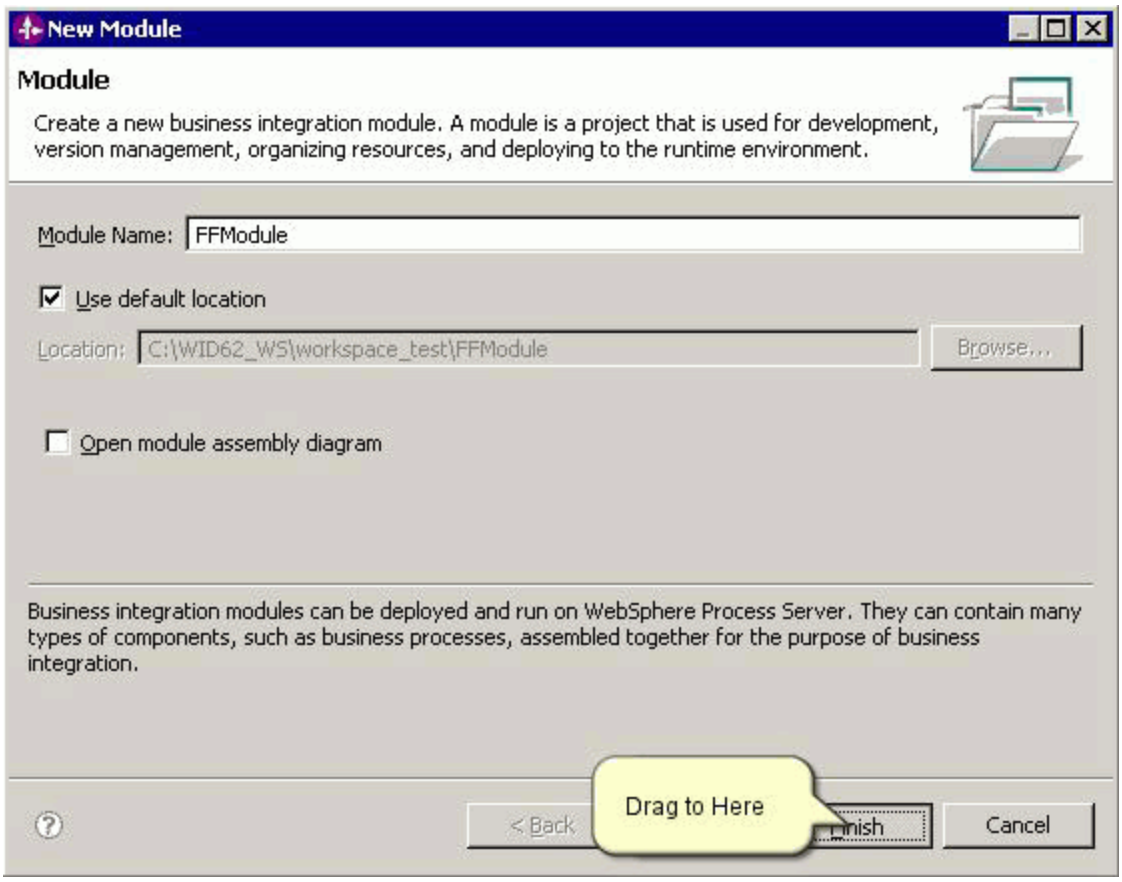
- The Business Integration (BI) perspective shows the imported RAR.



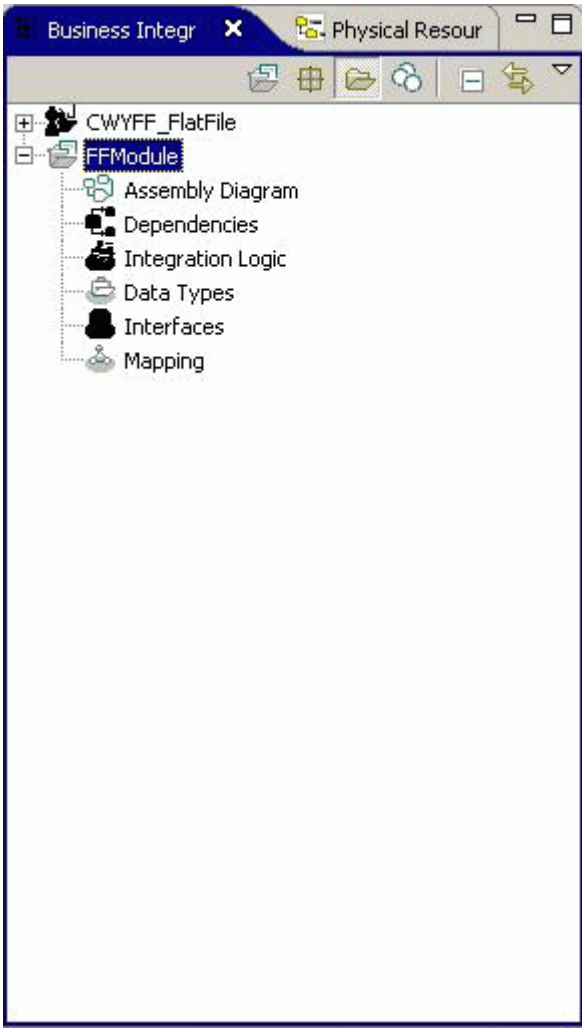
- Next, create a new **Module** to generate adapter outbound artifacts. Go to **File → New → Module** or right click in the BI perspective and click **New → Module**.



- Enter a desired name for your module. Uncheck the **Open assembly diagram** checkbox and click **Finish**.



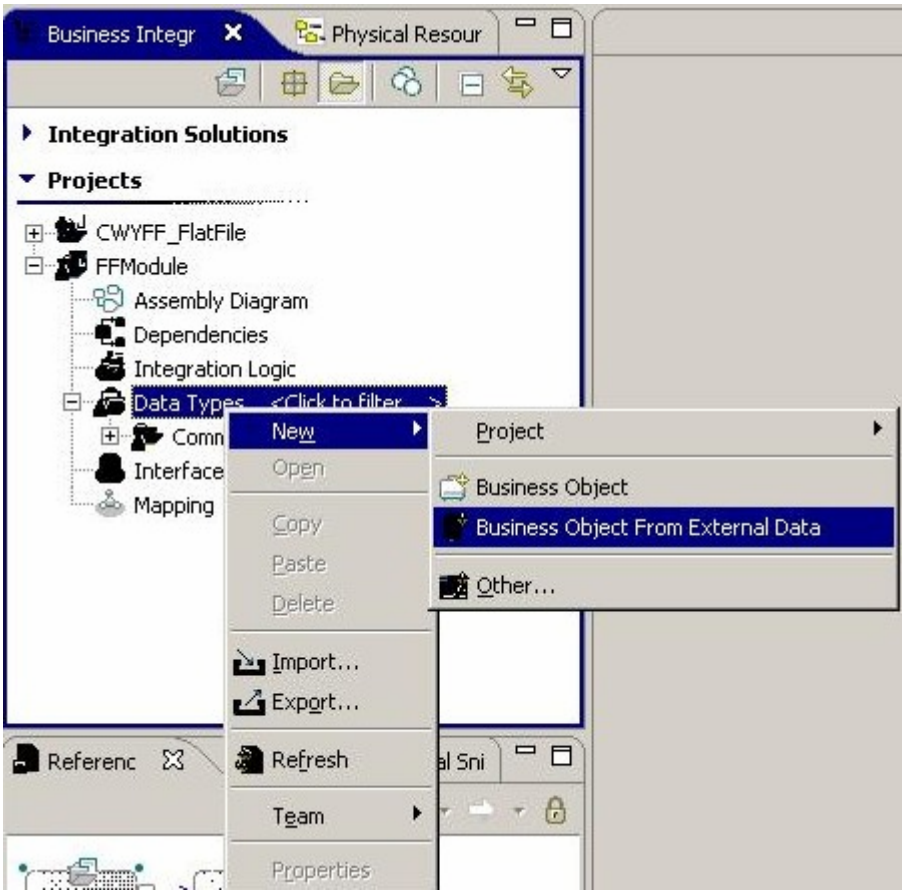
- The newly created module can be seen in the BI perspective.



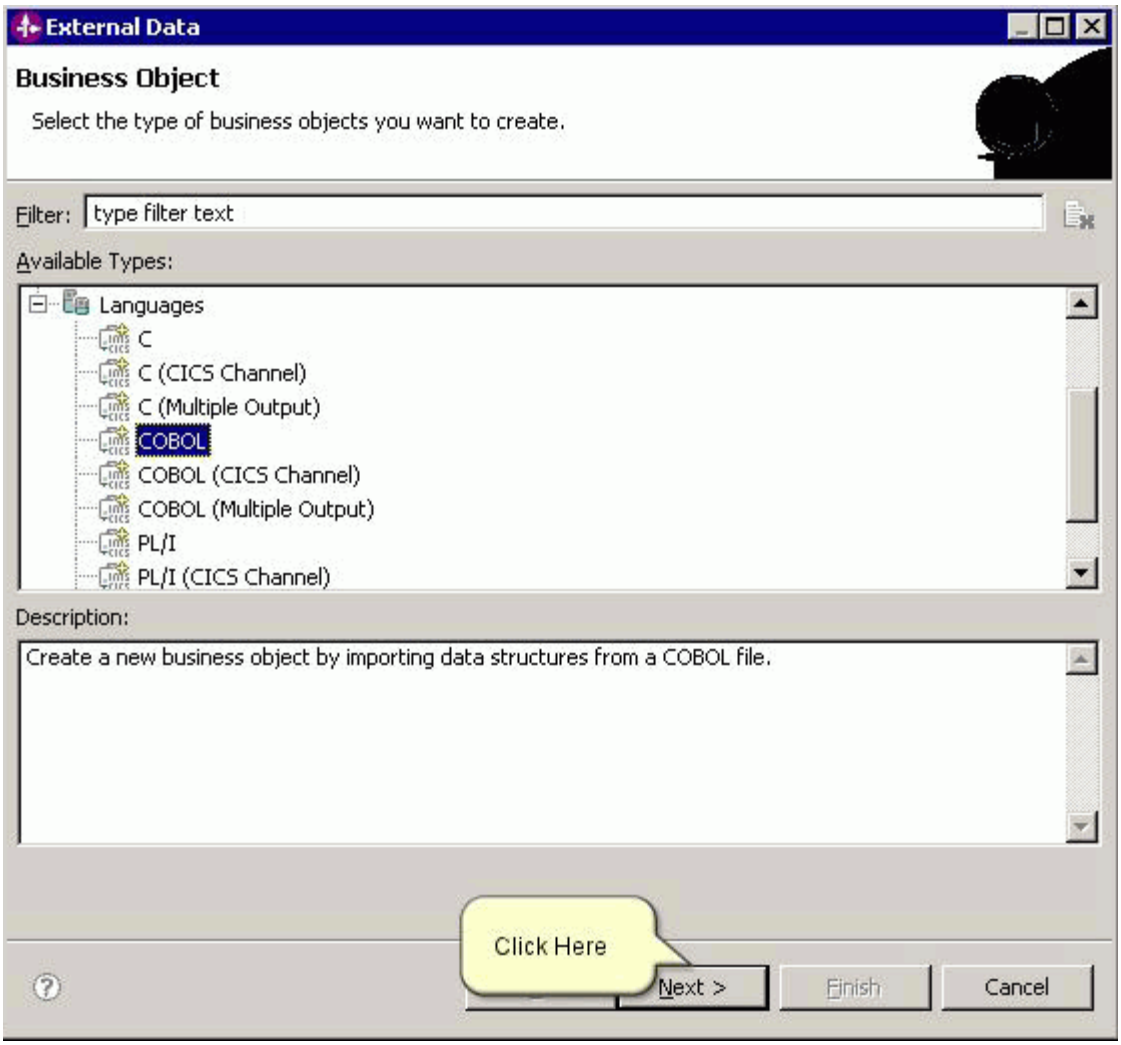
Generating Business Objects from Cobol CopyBook record

The External Data wizard is used to generate business objects from Cobol CopyBook records.

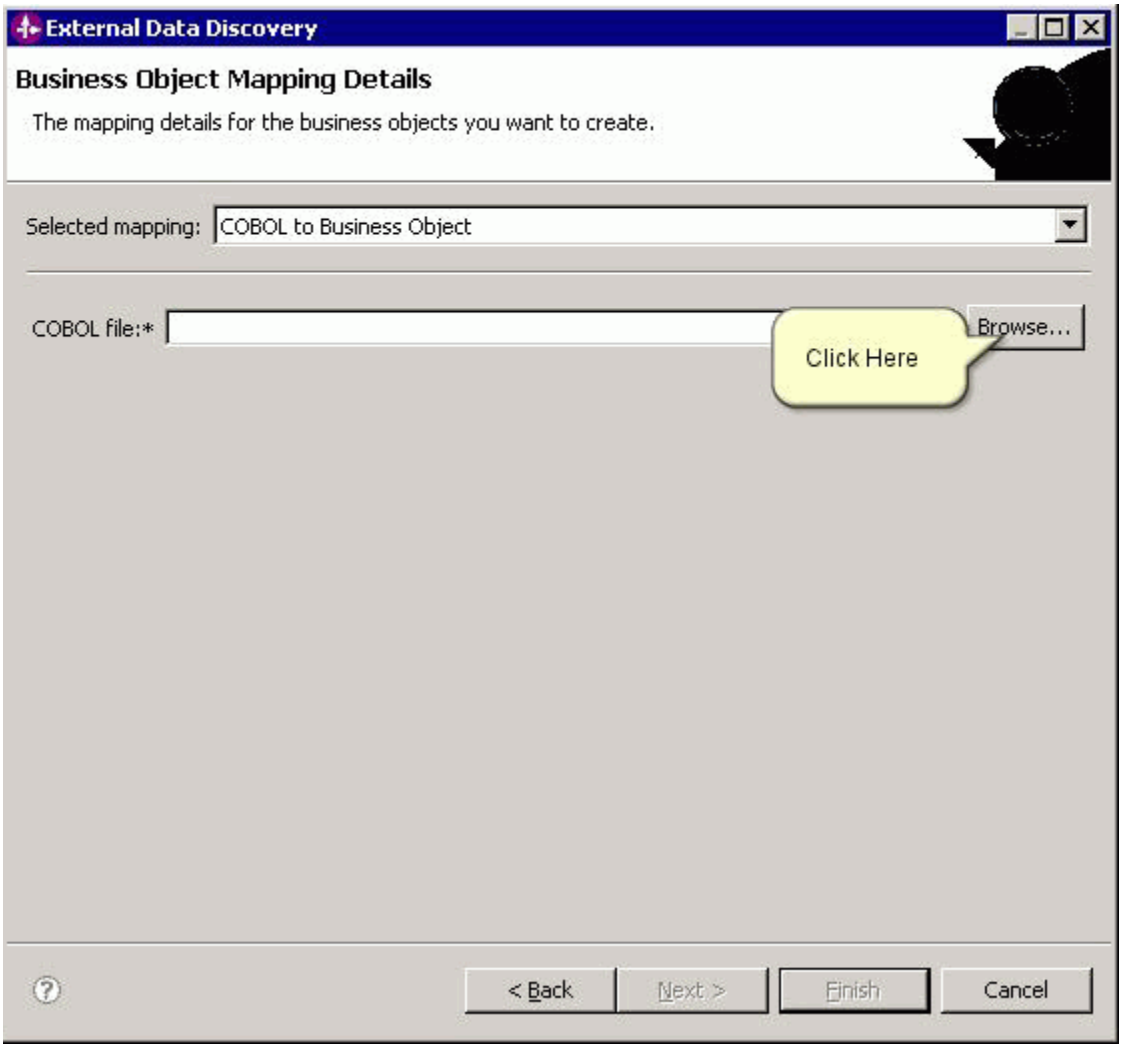
- In the screen capture below, right click on **DataTypes** in the module created and click **New → Business Object from External Data**



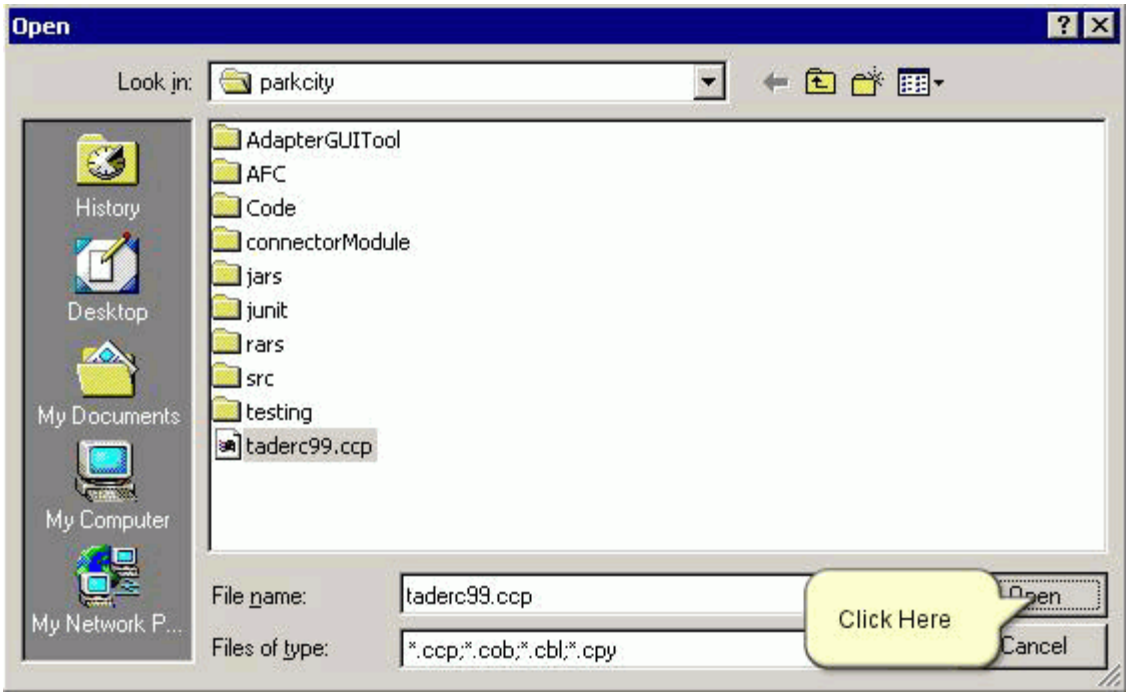
- Expand **Languages**, select **COBOL** and click **Next**.



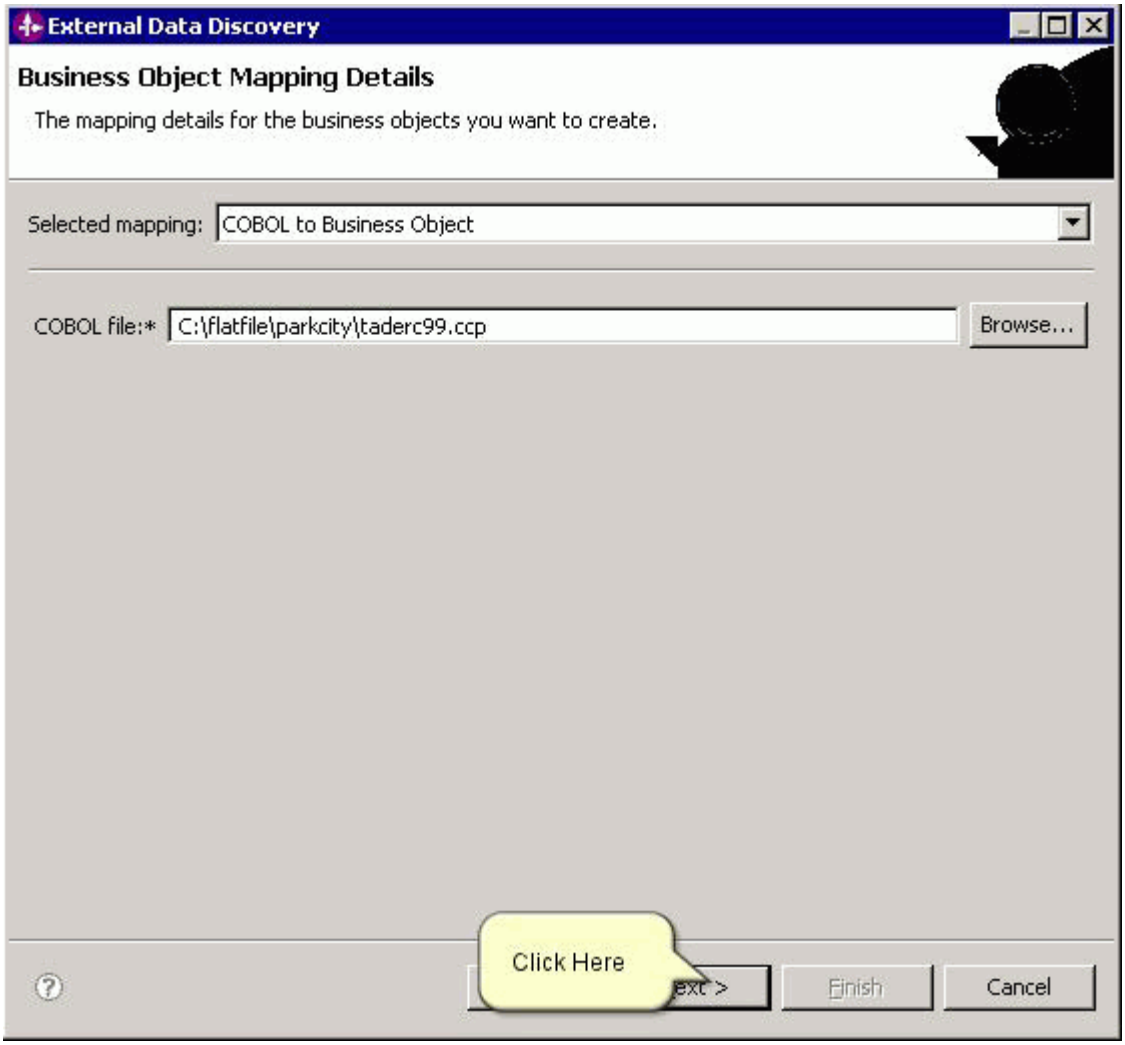
- In the External Data Discovery window, click **Browse** to choose the Cobol CopyBook record.



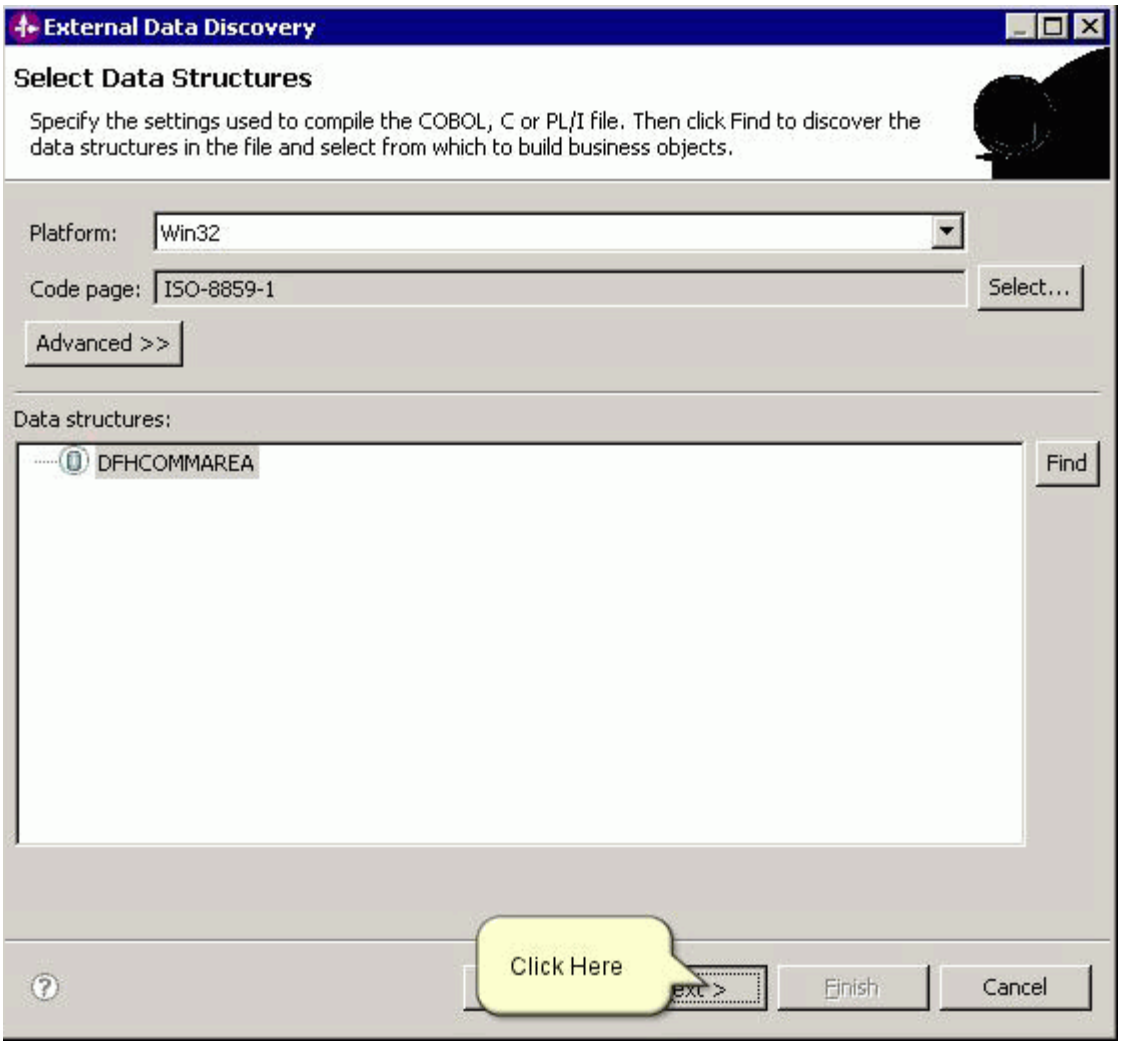
- Browse to the Cobol CopyBook record, select it and click **Open**.



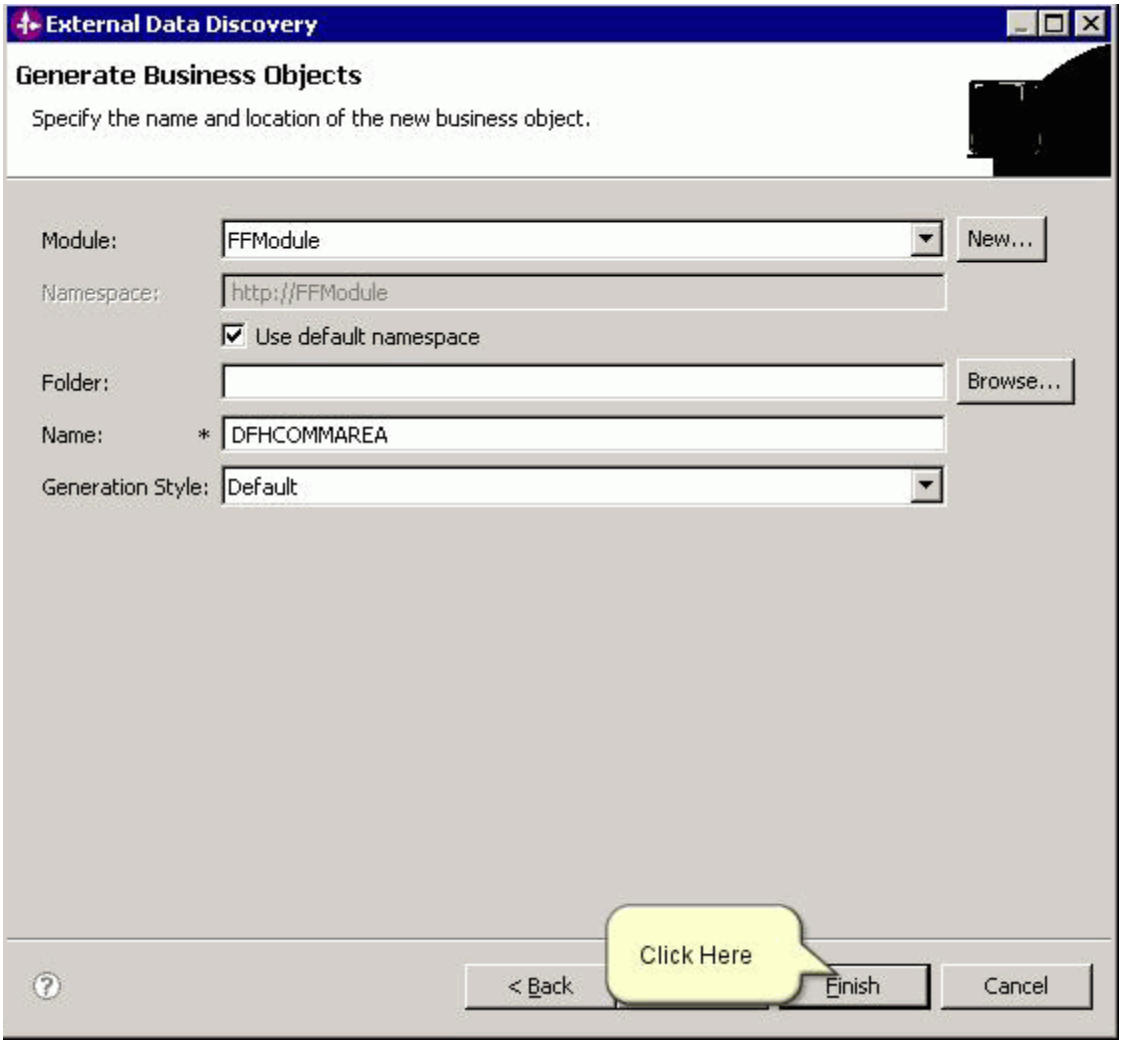
- After selecting the Cobol CopyBook record, click **Next** on the below screen capture.



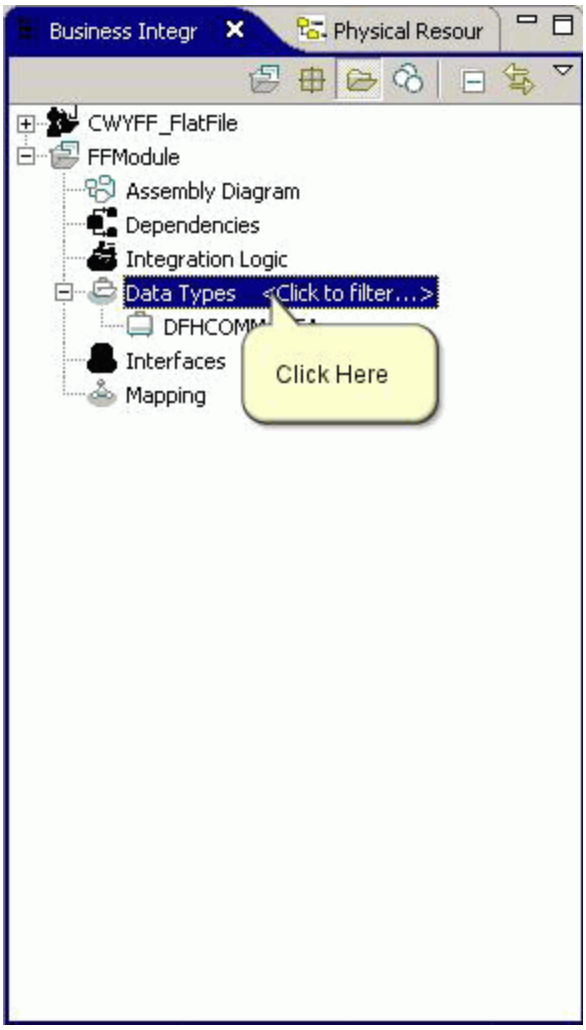
- On the Select Data Structures window, click on **Find**. It will display the Cobol CopyBook data structure. Select it and click **Next**.



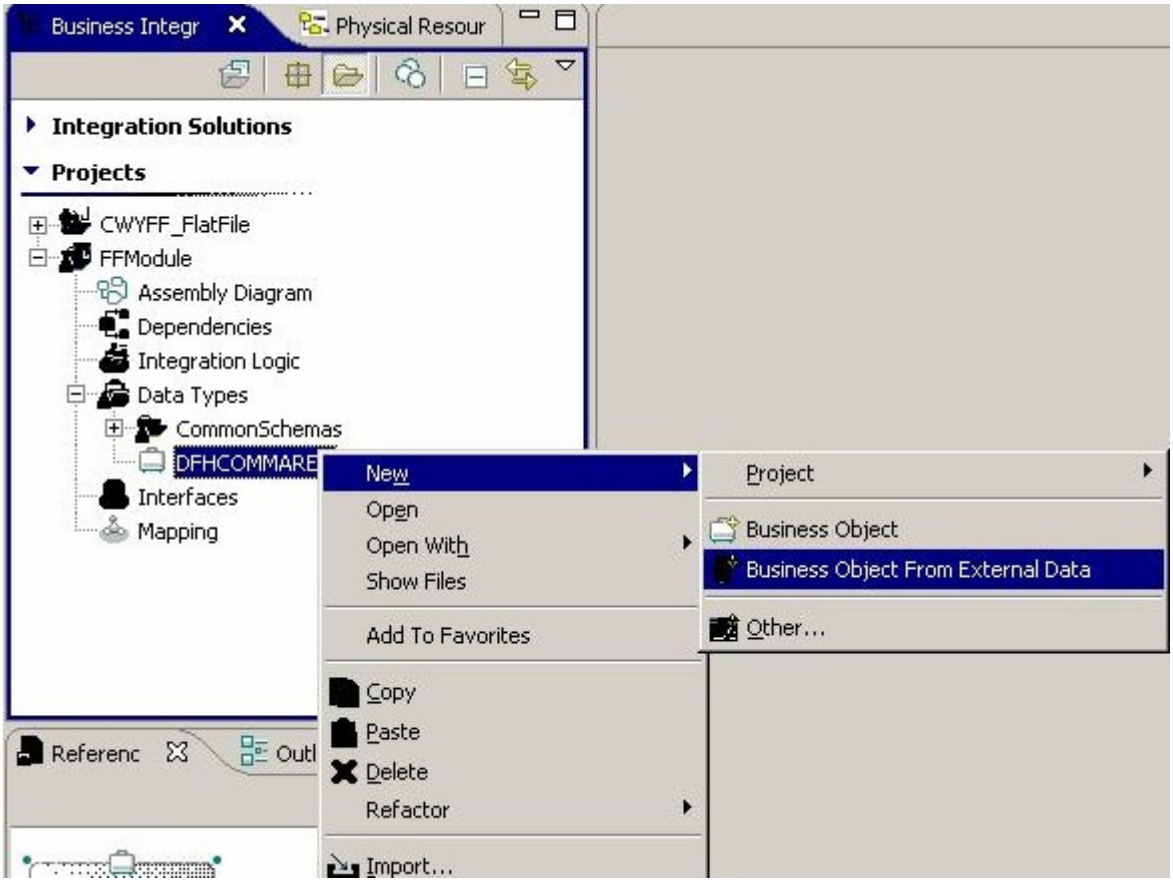
- After the data structure has been chosen, click **Finish** on the following window.



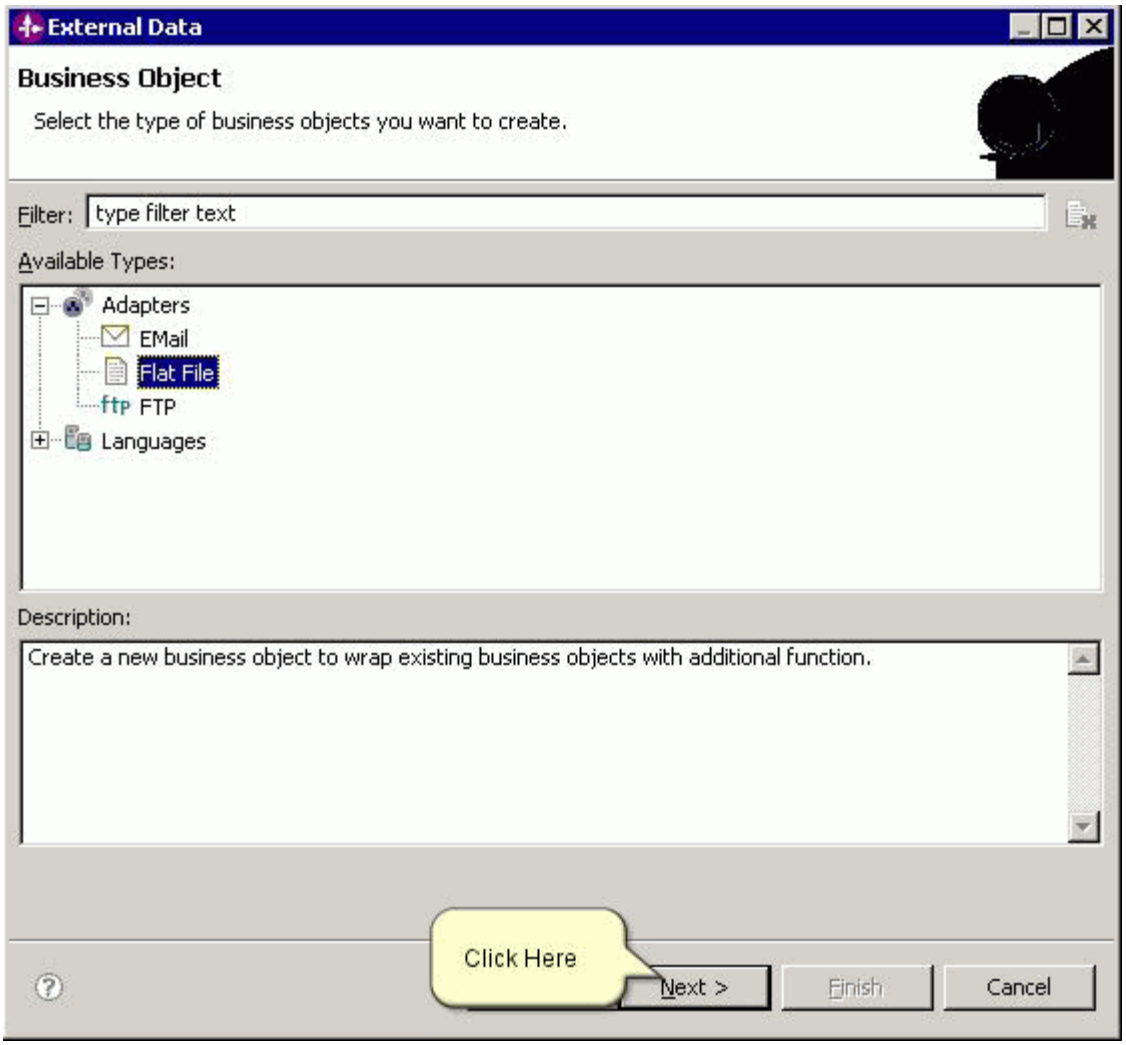
- The business object generated for the Cobol CopyBook record is now listed under **Data Types** in the module in BI perspective.



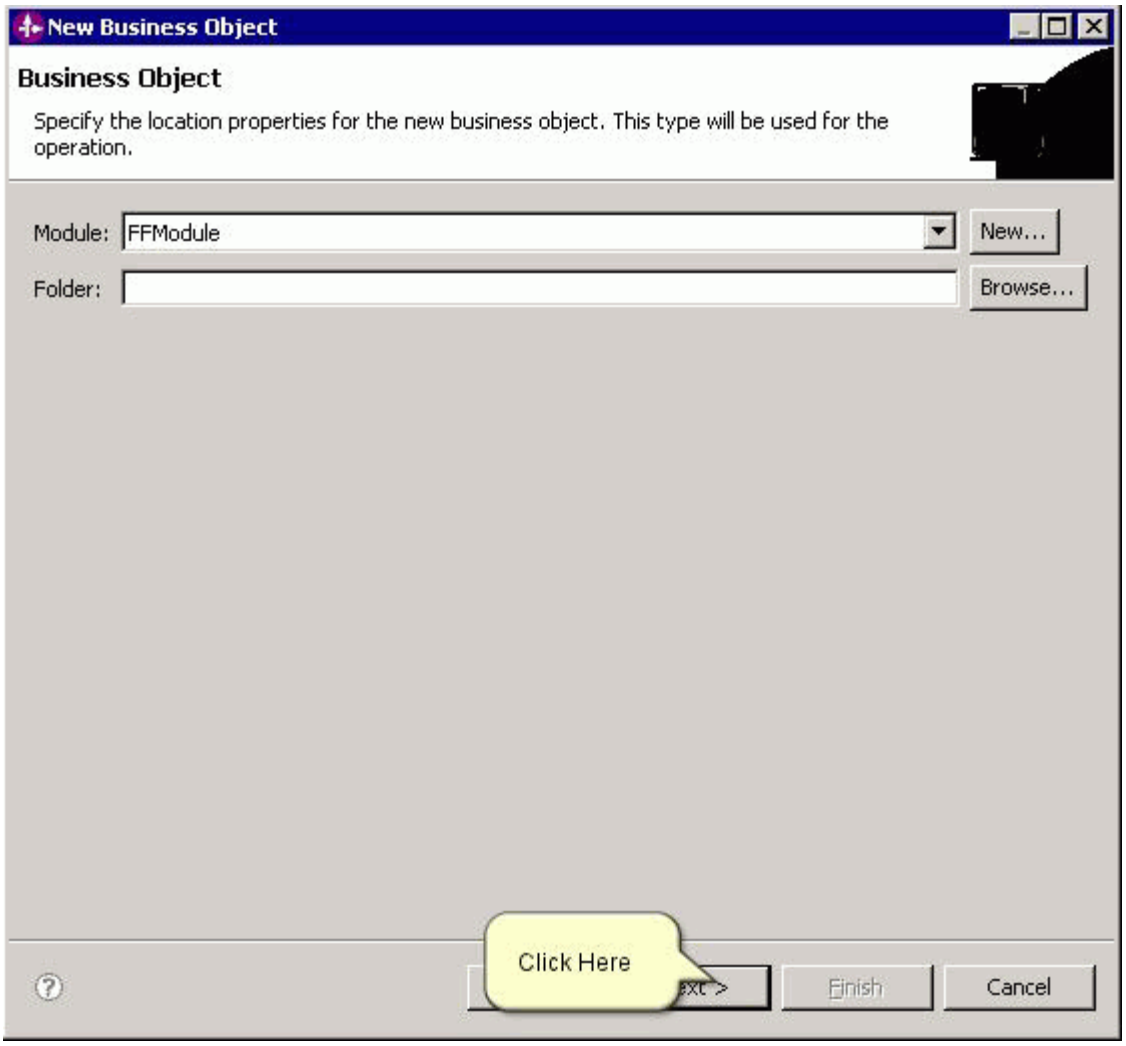
- To generate wrappers for the newly created business object, run the External Data wizard once again. Generate wrappers for Create as well as Retrieve. The wrappers for Retrieve are different from the wrappers generated for any other outbound operation. The wrappers for Create operation will be generated initially.
- Right click on **Data Types**, select **New → Business Object from External Data** as shown in the screen capture below



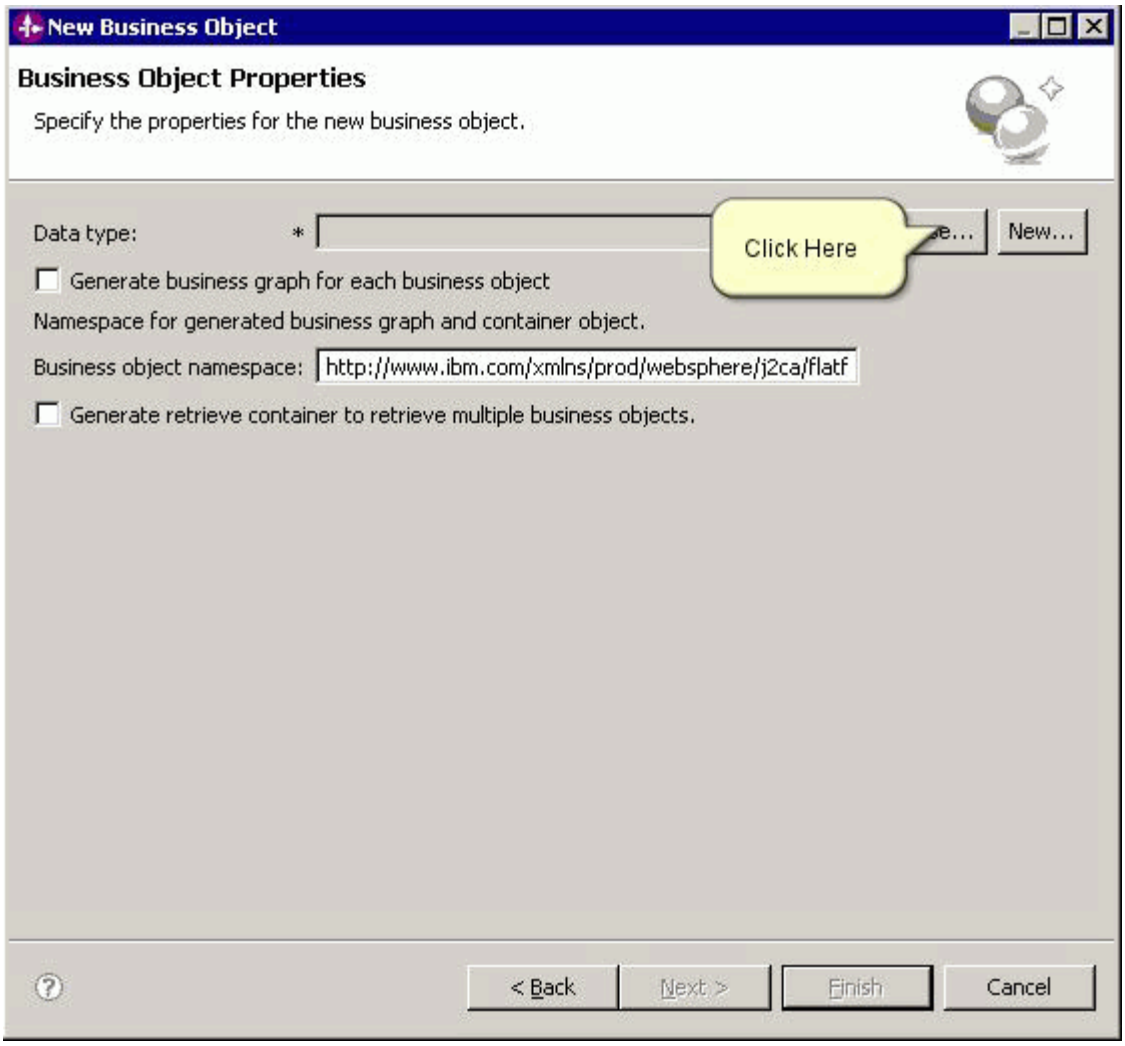
- Expand **Adapters**, select **Flat File** and click **Next**.



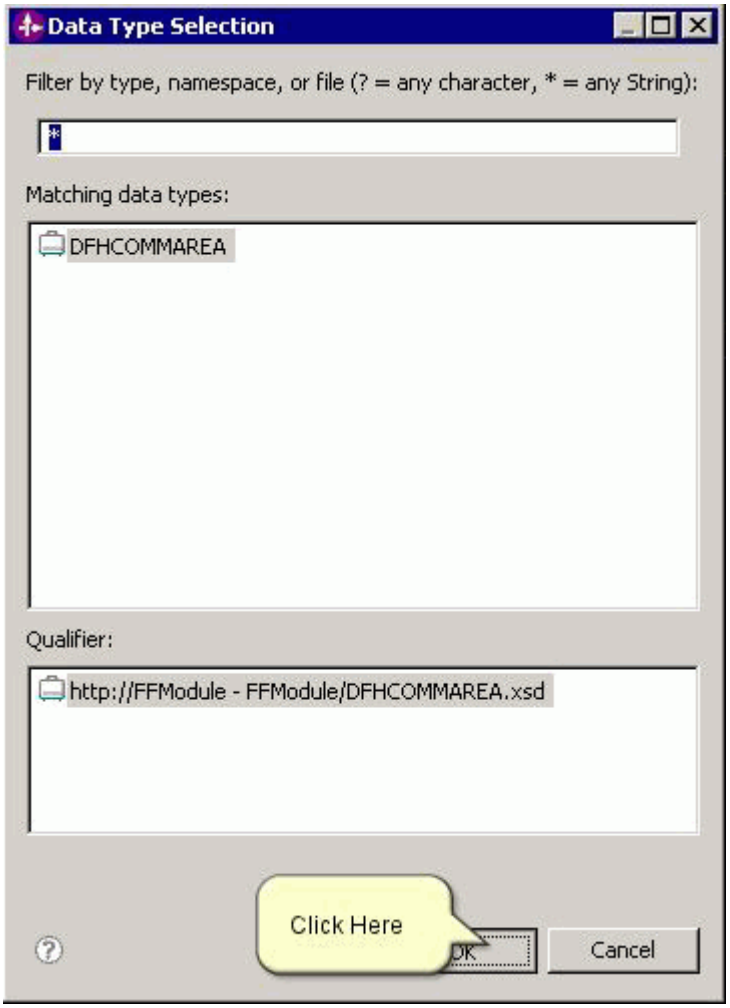
- In the following screen capture, the module name appears by default. Click **Next**.



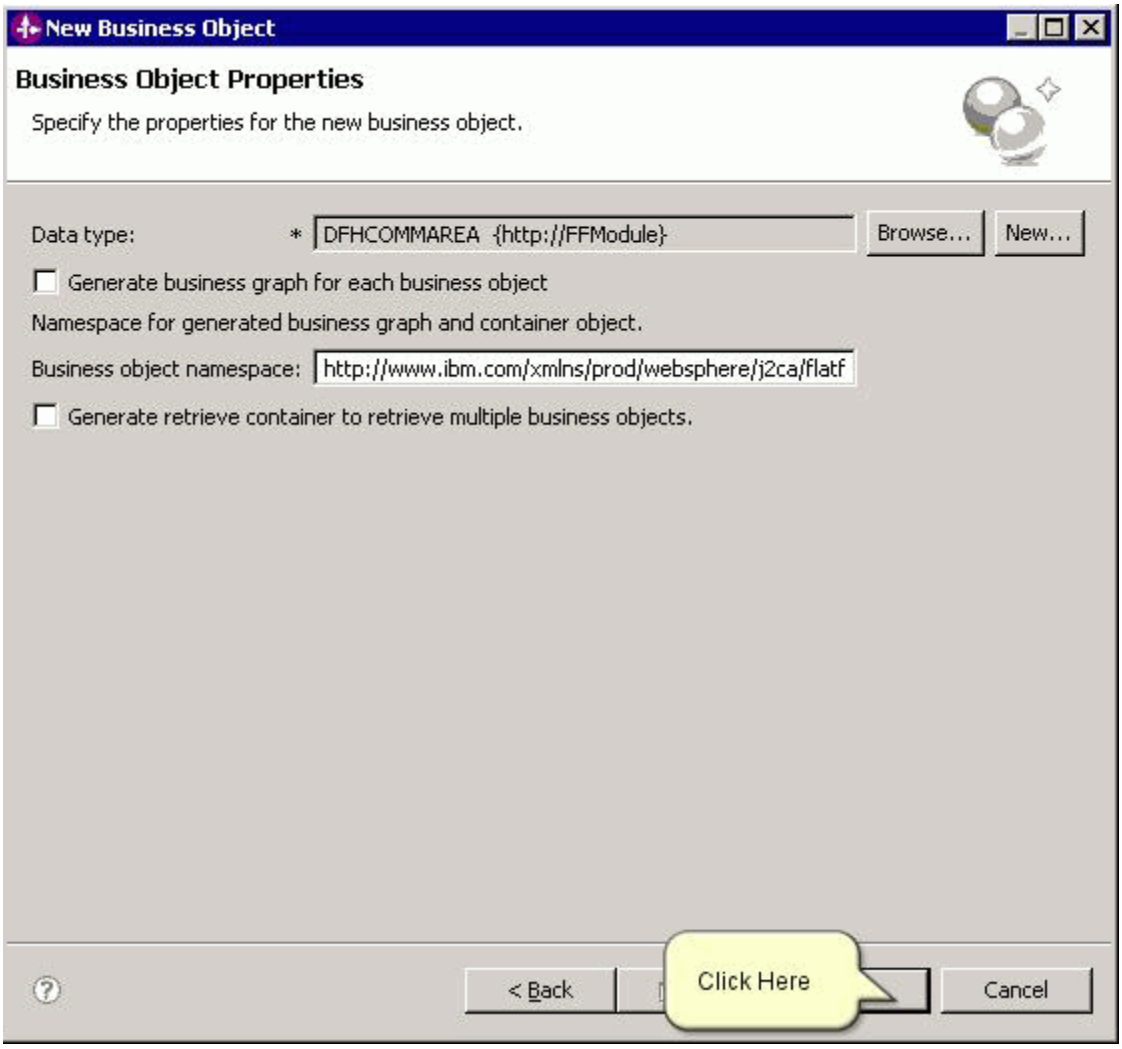
- In the following screen capture, click **Browse** to choose the business object created earlier.



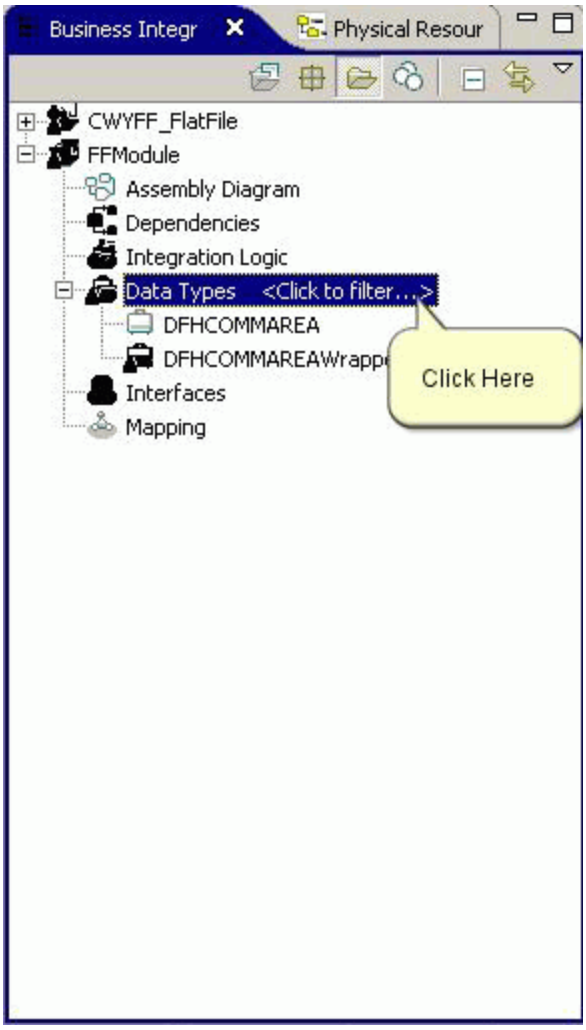
- Choose the business object and click **OK**.



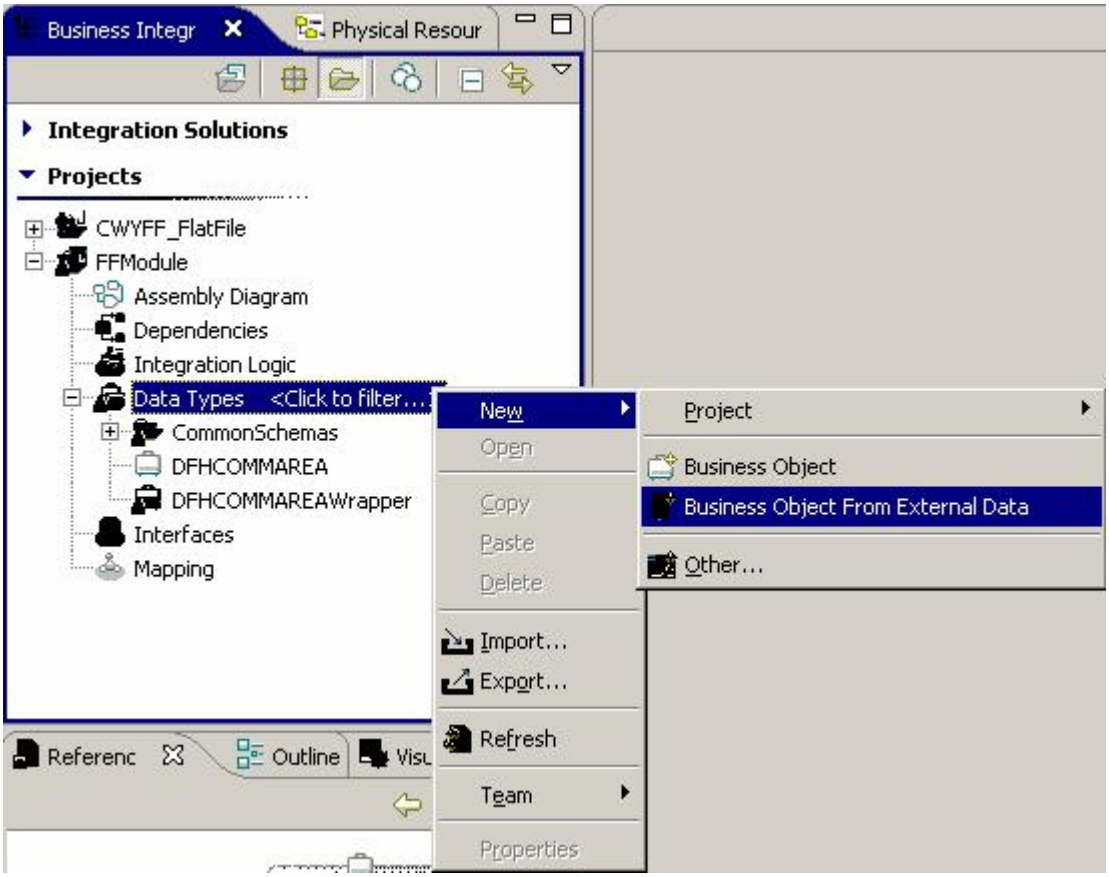
- In the following screen capture, if you want the business graph to be generated alongwith the wrapper, check the **Generate business graph for each business object** checkbox. In the current scenario, you won't be generating the business graph; so, leave it unchecked. Click **Finish**.



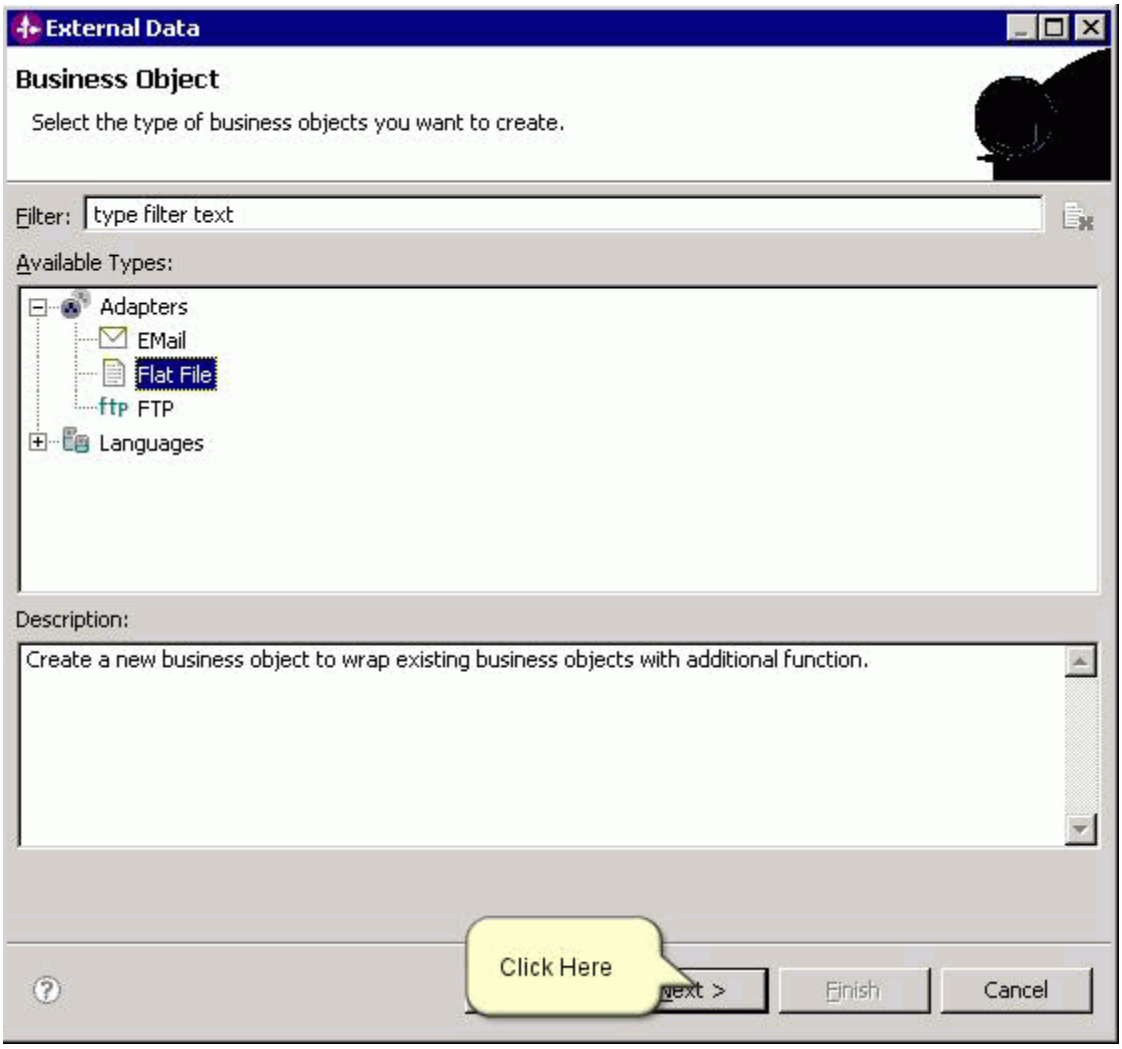
- The Wrapper business object is created as shown in the screen capture below



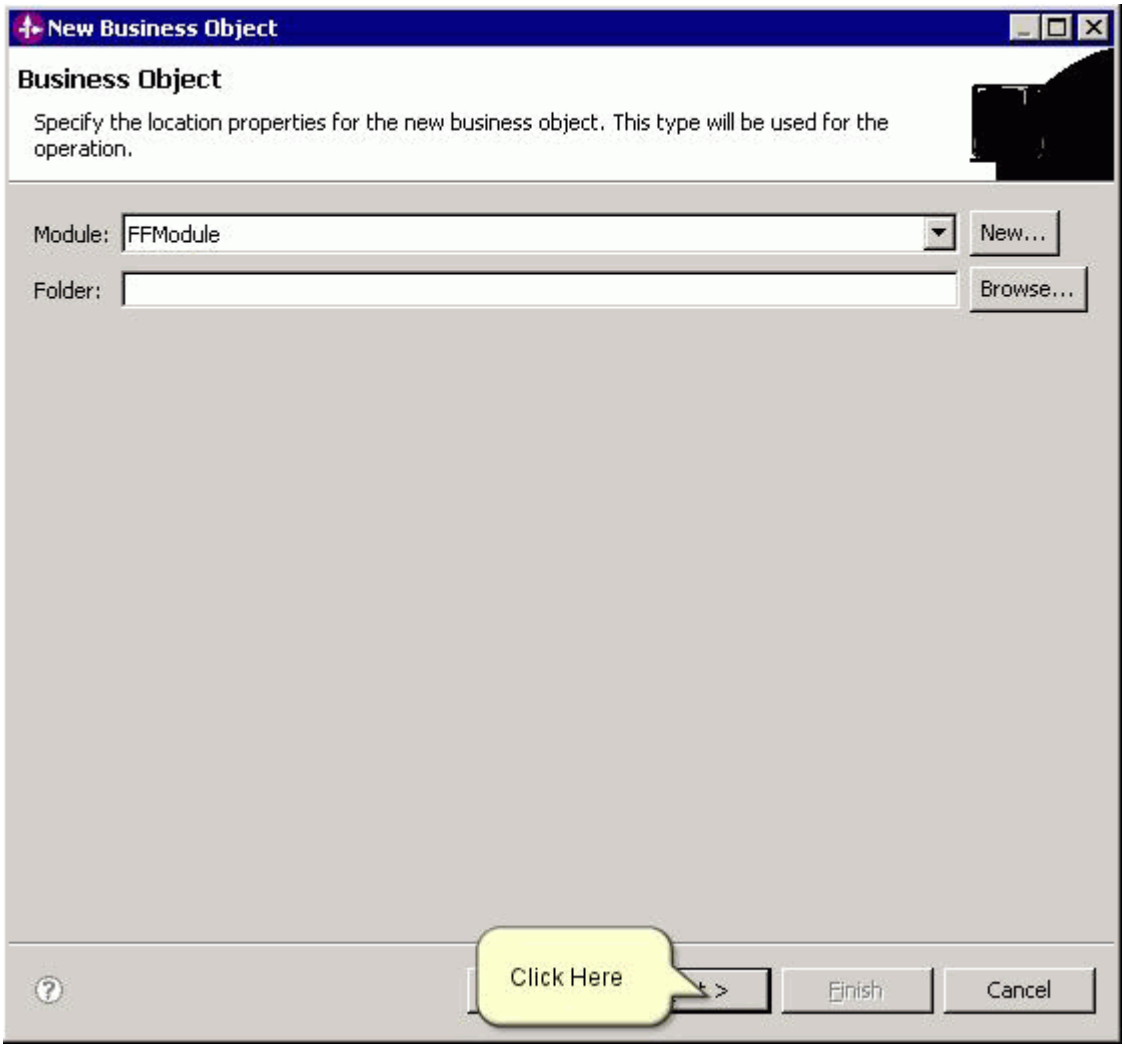
- Next, generate the wrapper for the Retrieve operation. Right click on **Data Types**, click **New → Business Object from External Data**



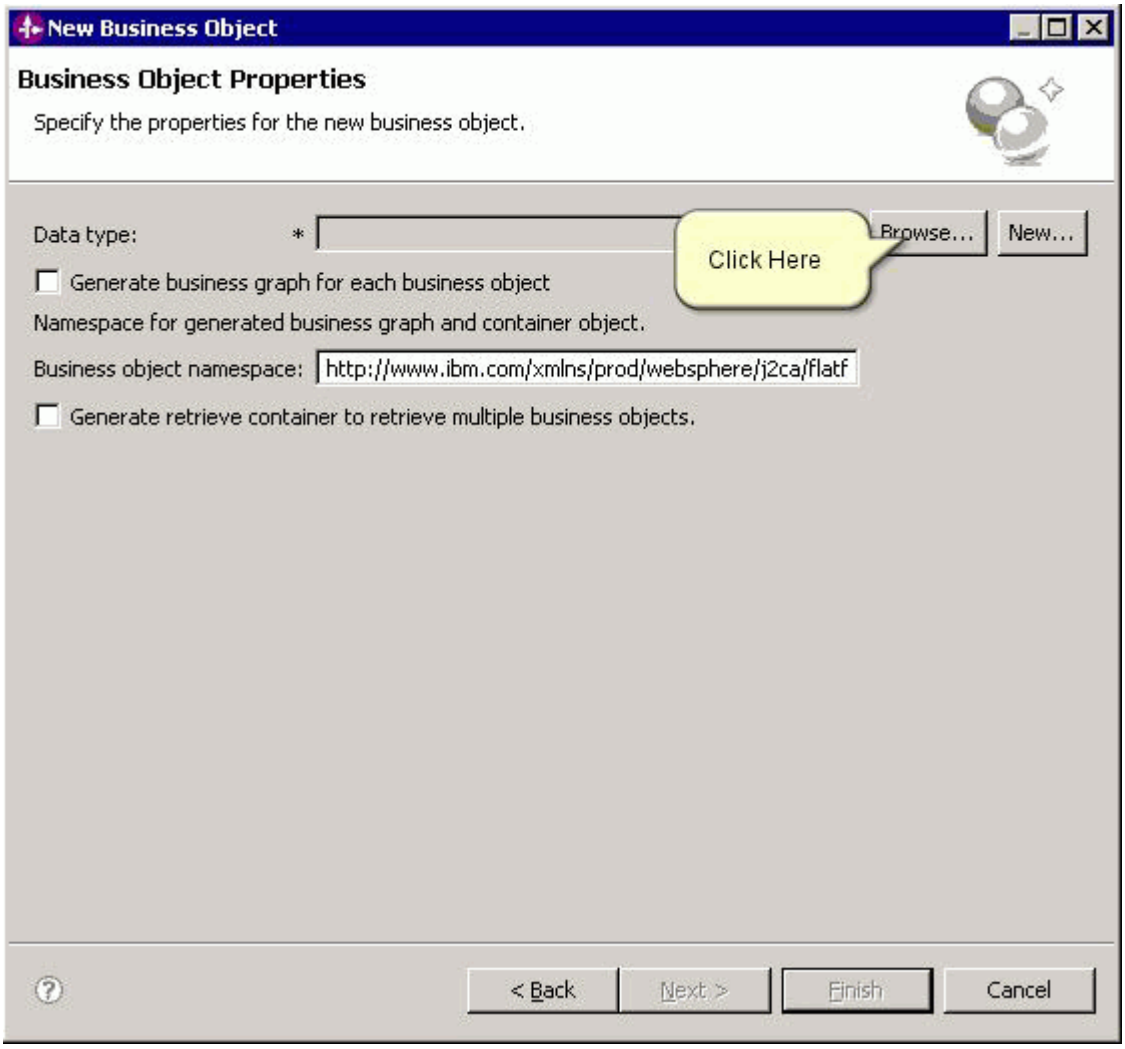
- Expand **Adapters**, select **Flat File** and click **Next**.



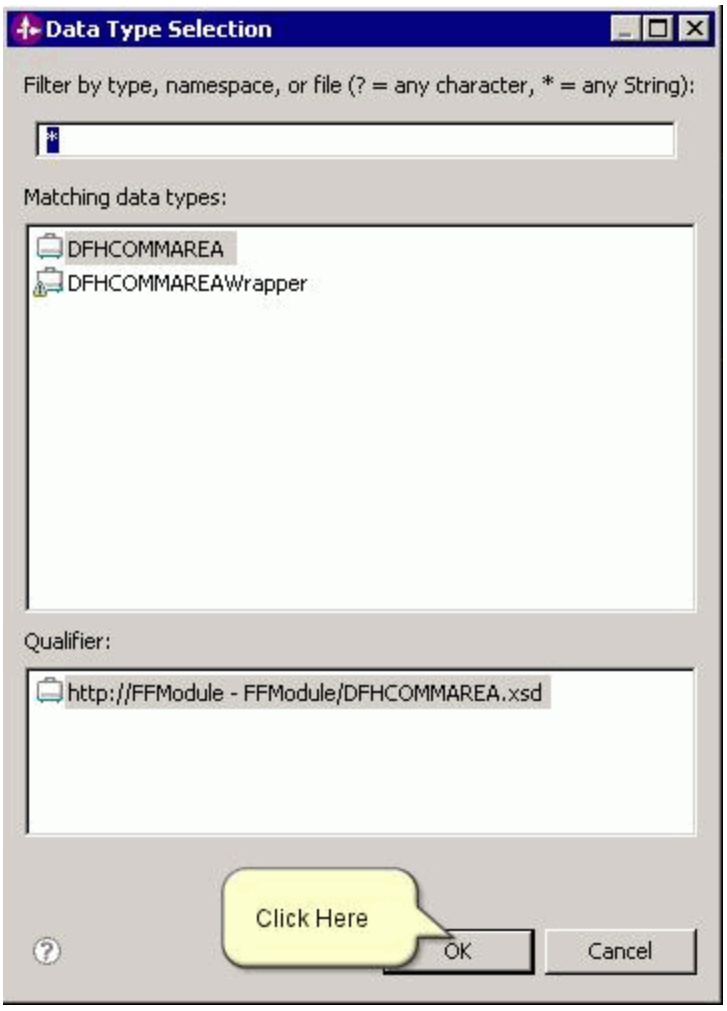
- In the following screen capture, click **Next**.



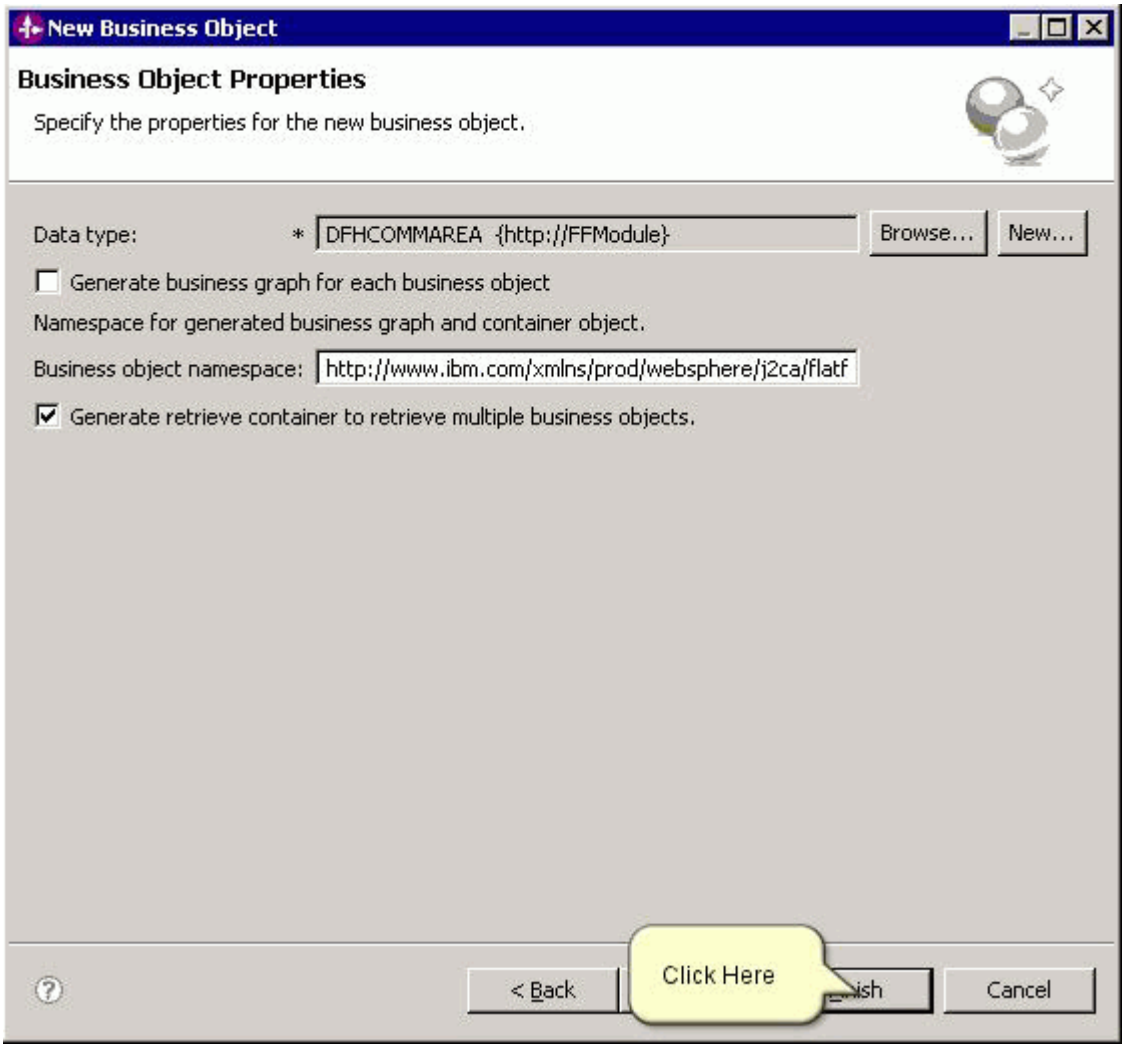
- Click the **Browse** button to choose the business object created earlier.



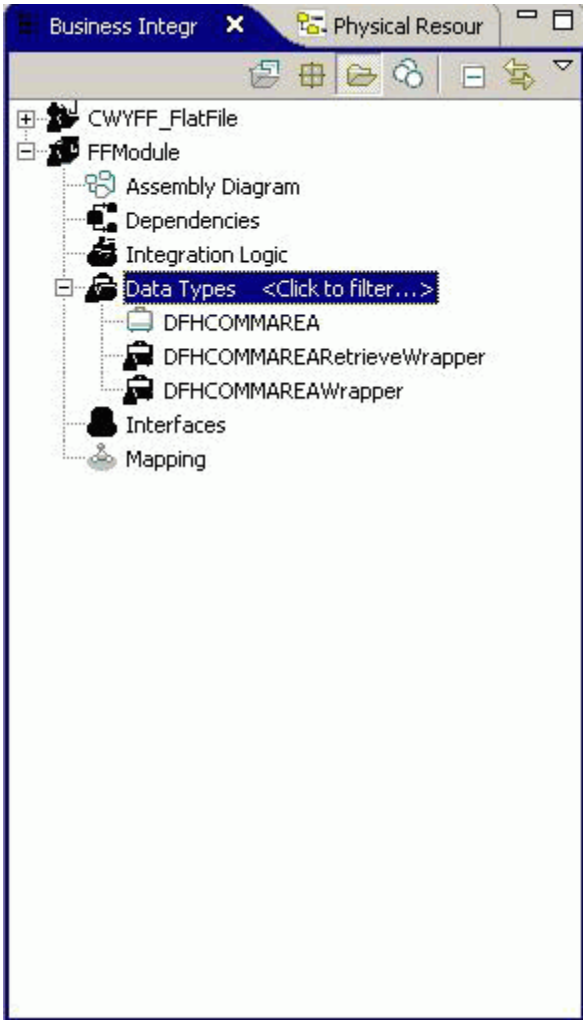
- Choose the business object and click **OK**.



- In the following screen capture, check the **Generate retrieve container to retrieve multiple business objects** checkbox. This will generate the retrieve wrapper business object. Business graph is optional; so, leave the **Generate business graph for each business object** checkbox unchecked. Click **Finish**.



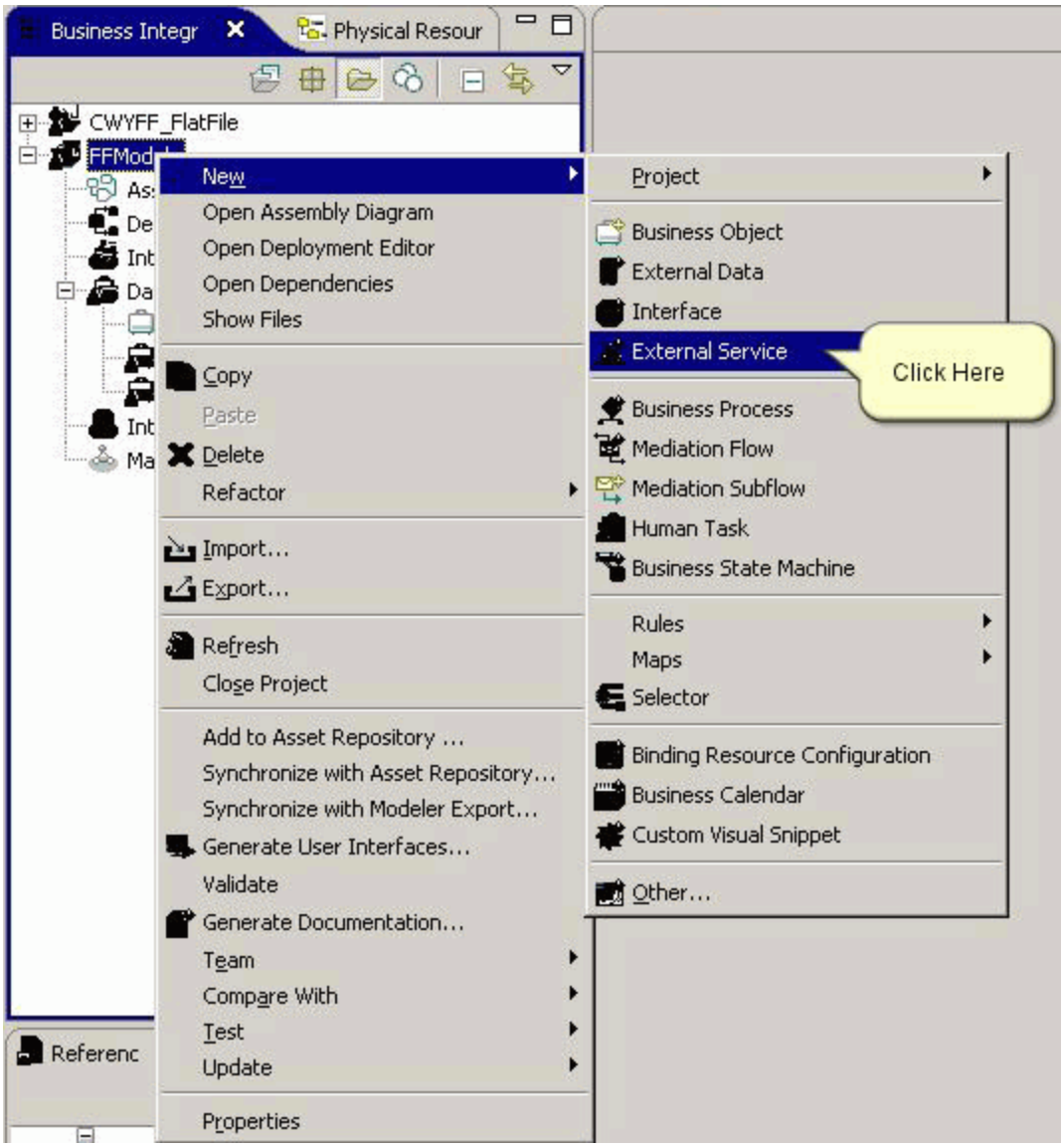
- The following figure shows the retrieve wrapper in the BI perspective



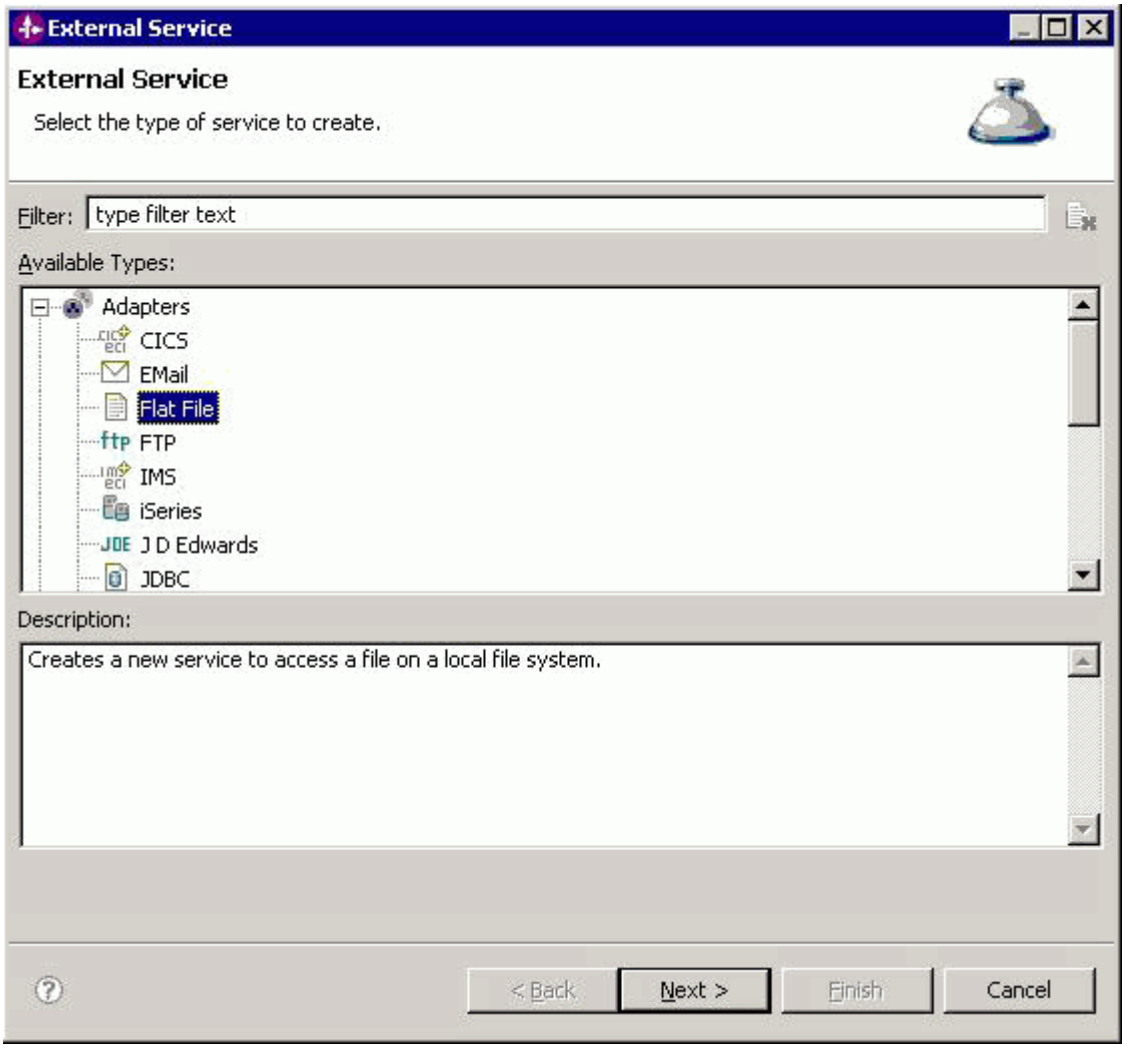
Generating the artifacts by running the External Service Wizard

To configure the adapter and generate the artifacts, run the External Service wizard.

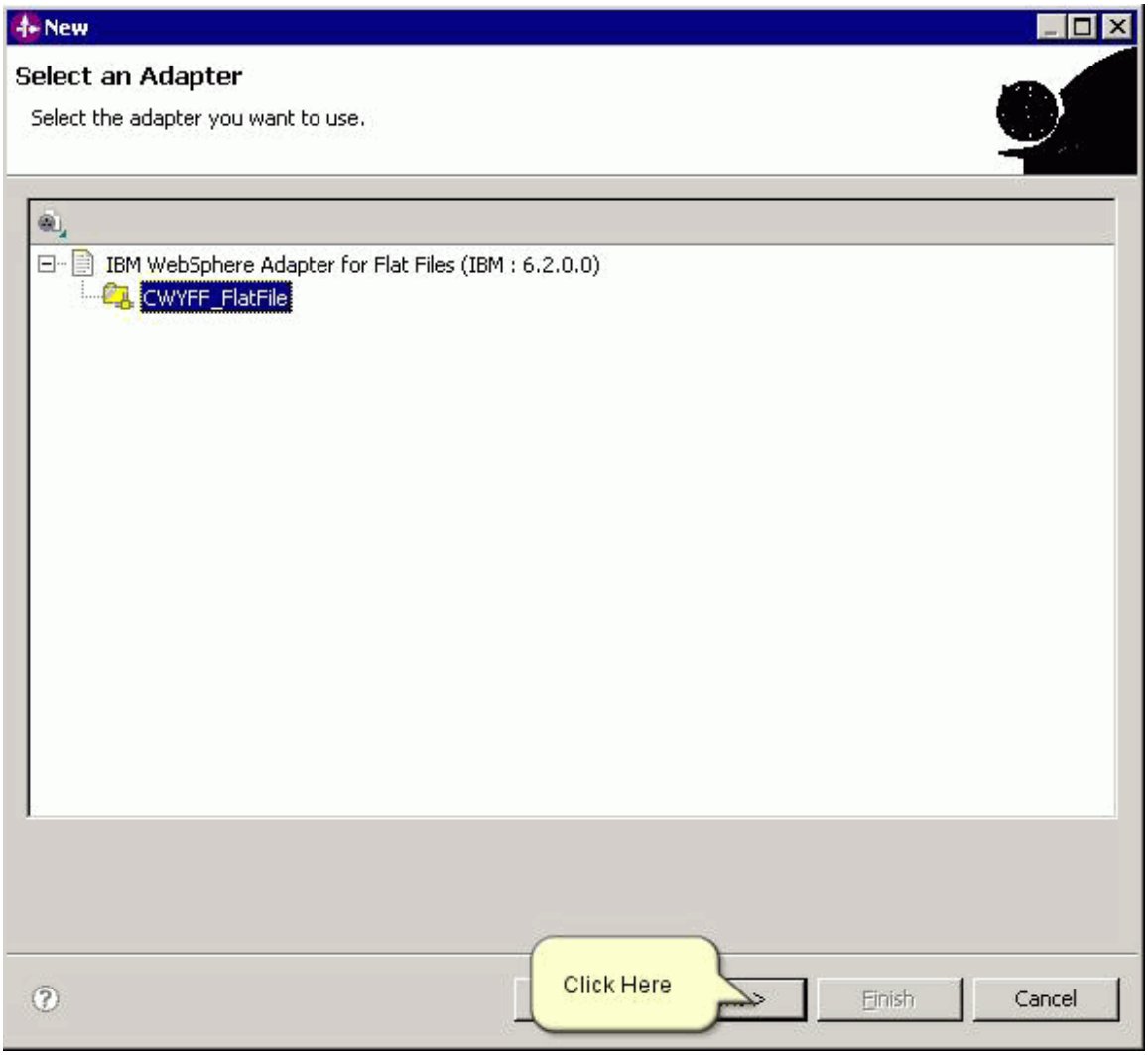
- Right click on the module and select **New → External Service**



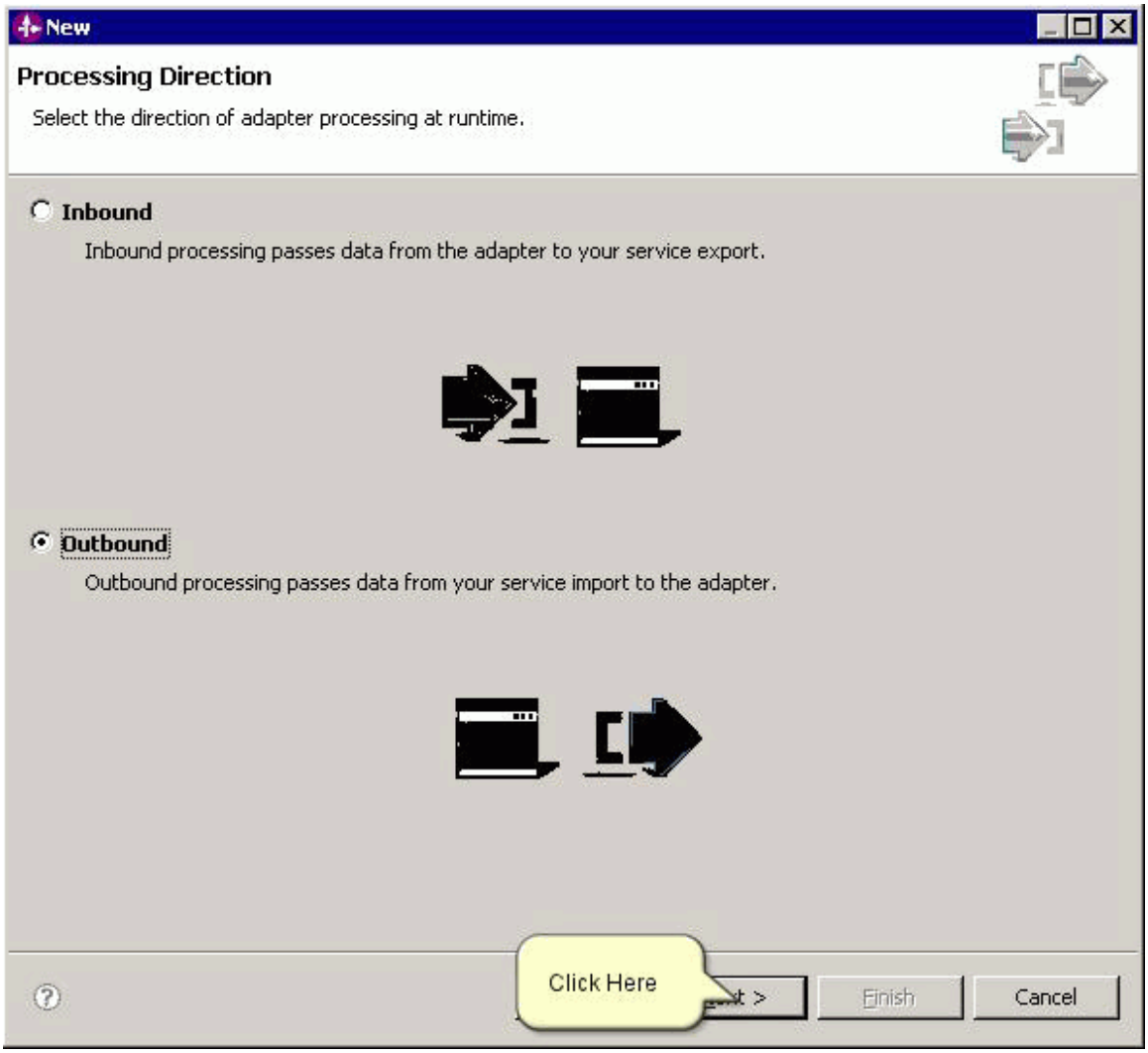
- Expand **Adapters**, choose **Flat File** and click **Next**.



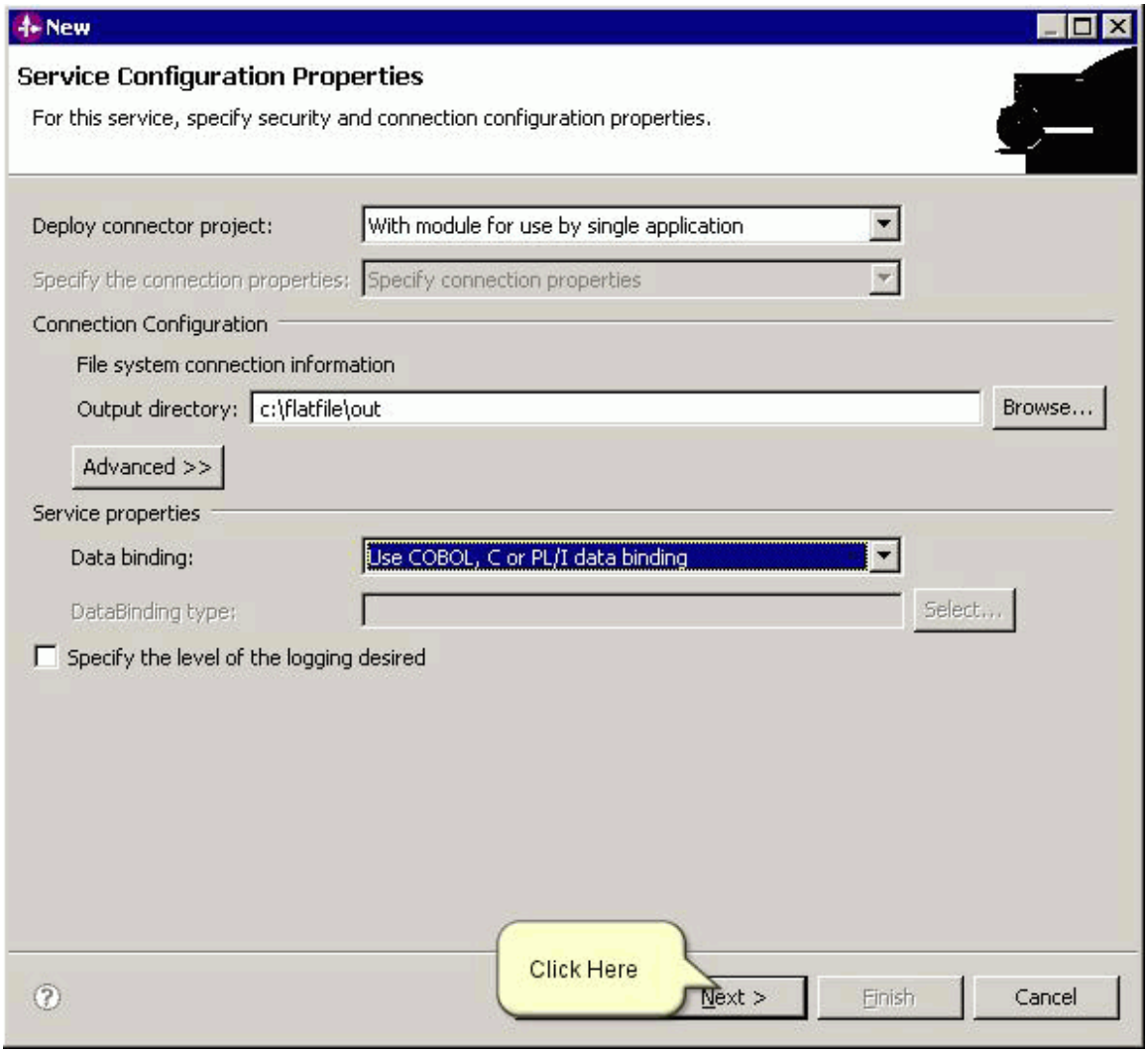
- In the following screen capture, choose the RAR that was imported and click **Next**.



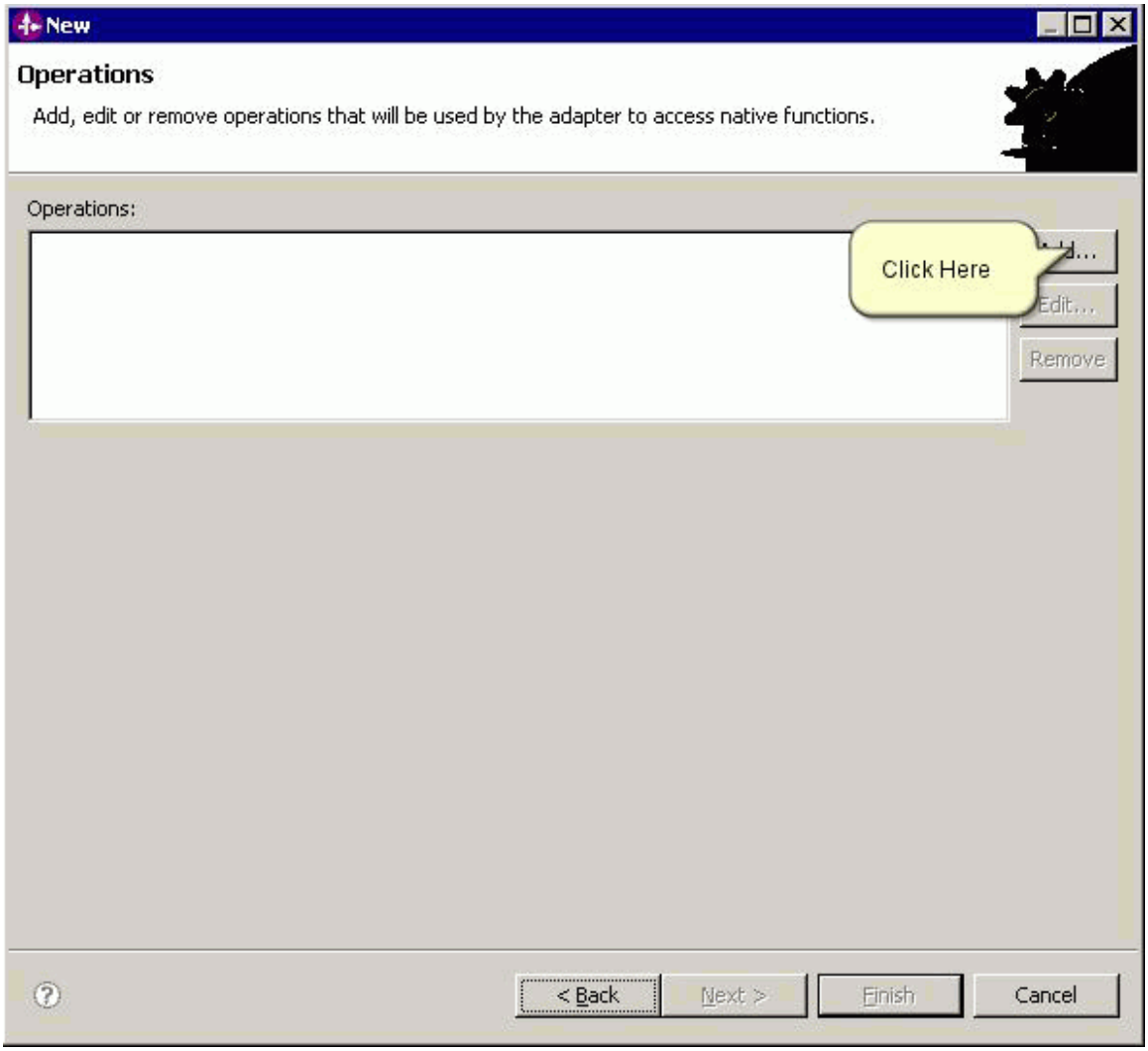
- In the next screen capture, choose the mode of processing. The samples deal with Create and Retrieve outbound scenarios. So, choose the **Outbound** radio button. Click **Next**.



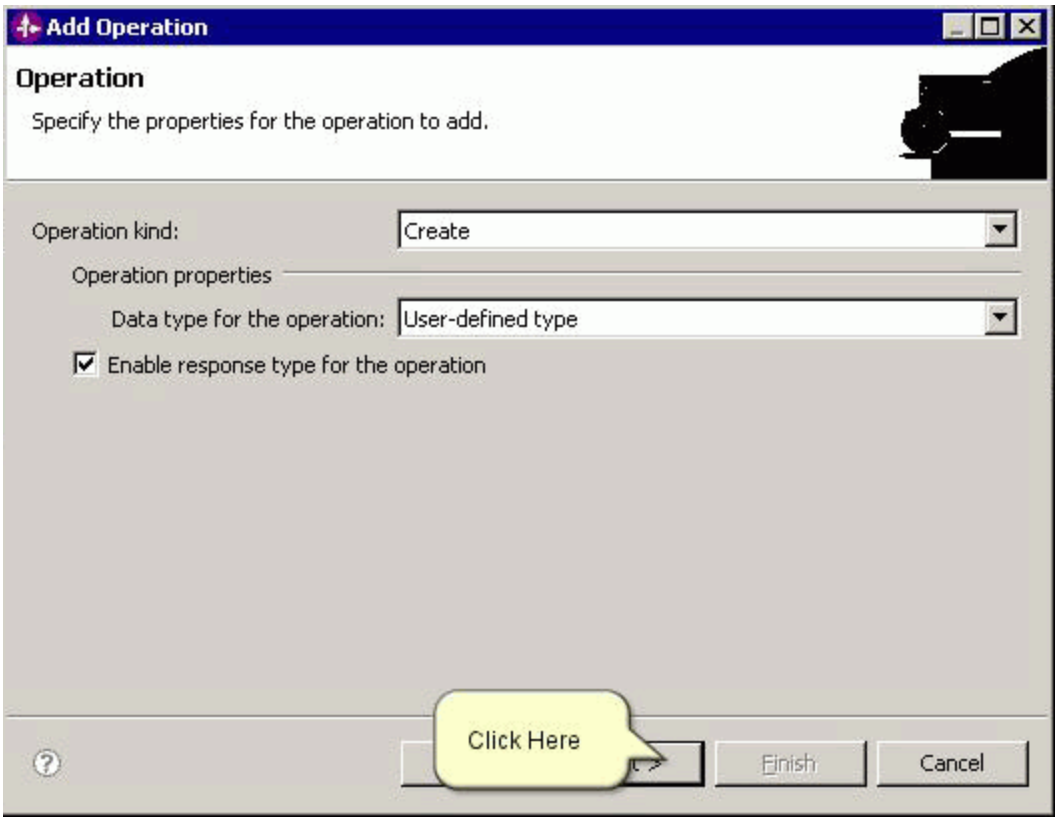
- In the Service Configuration Properties window, specify the **Output directory**. In the **Data binding** dropdown, choose **Use COBOL, C or PL/I data binding**. Click **Next.**



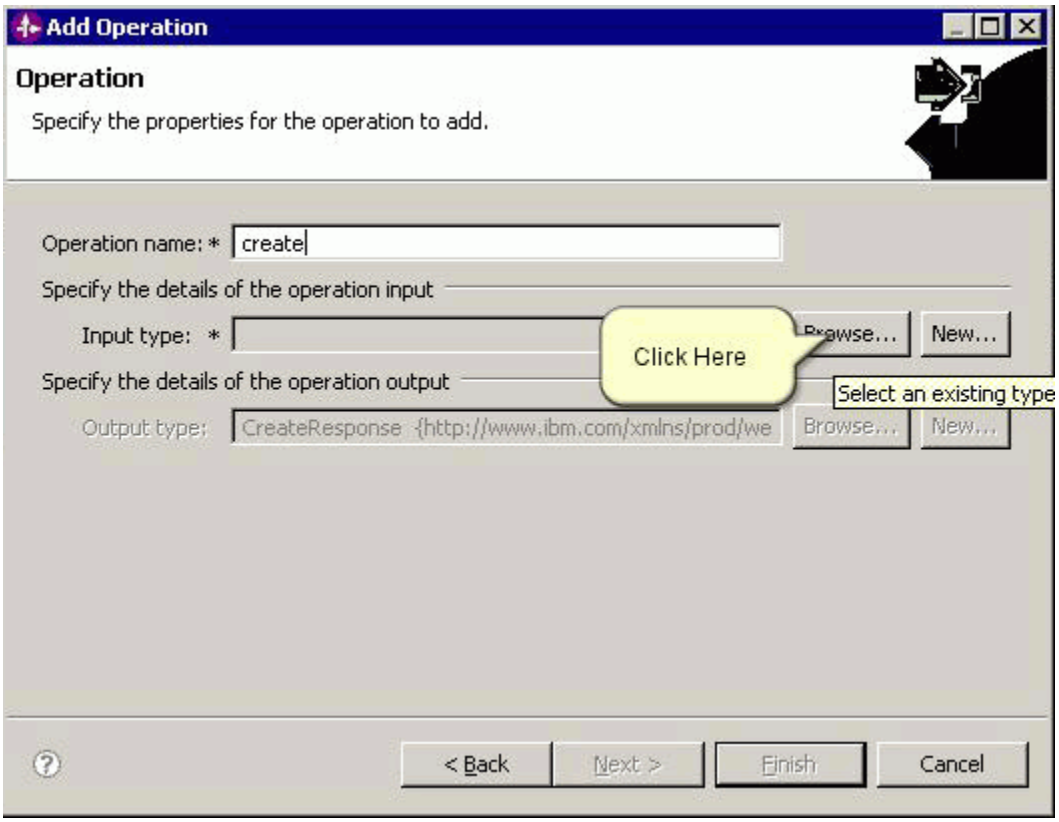
- Click **Add** to add the operations.



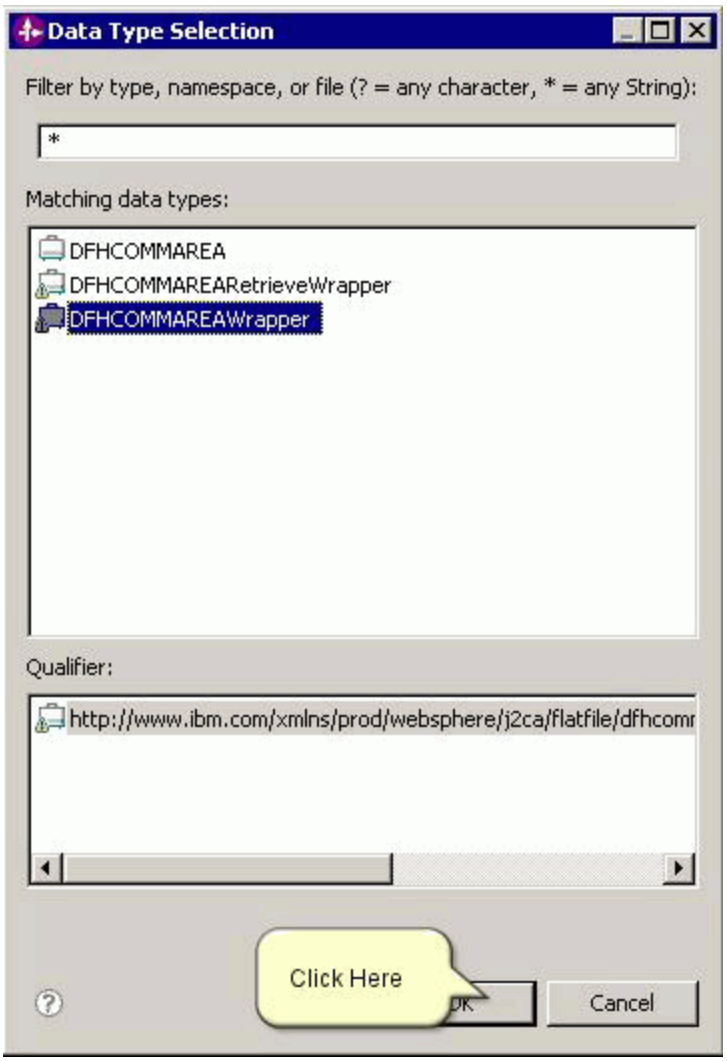
- Add the Create operation first. Select **Create** as the operation and set **User-defined type** as the **Data type for the operation**. Also, select the **Enable response type for the operation** checkbox to return a response object with the filename of the file created.



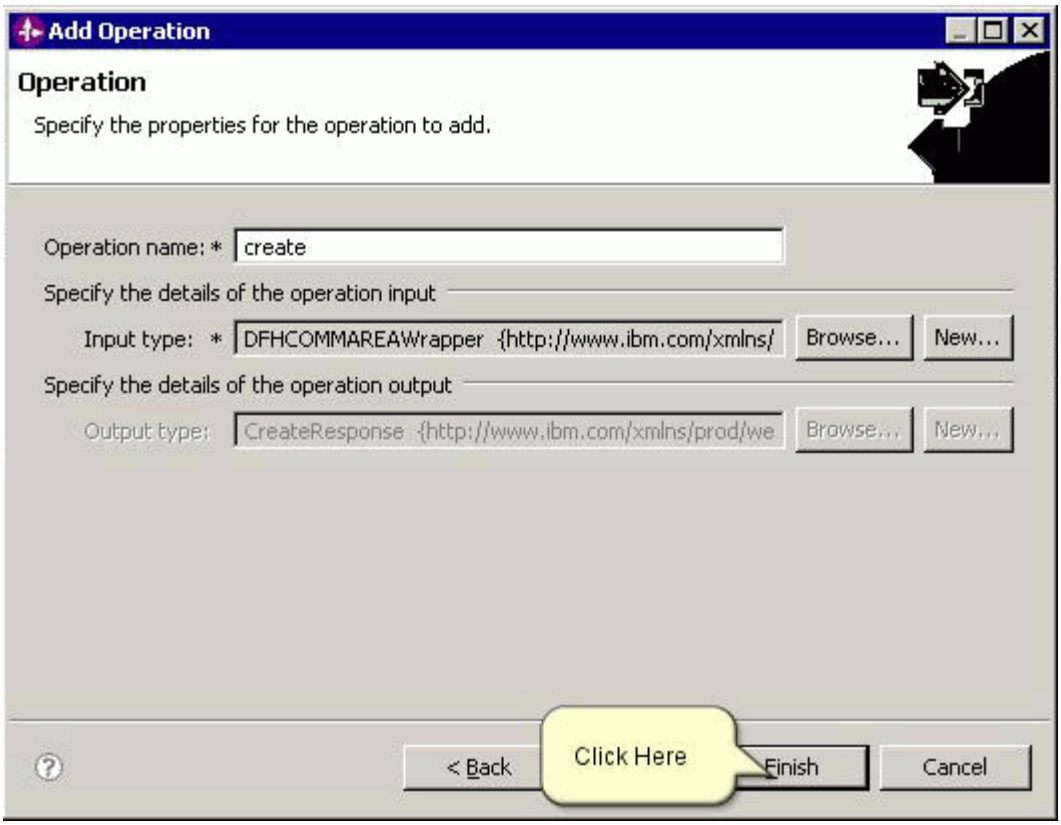
- Click on **Browse** to choose the input type for the Create operation



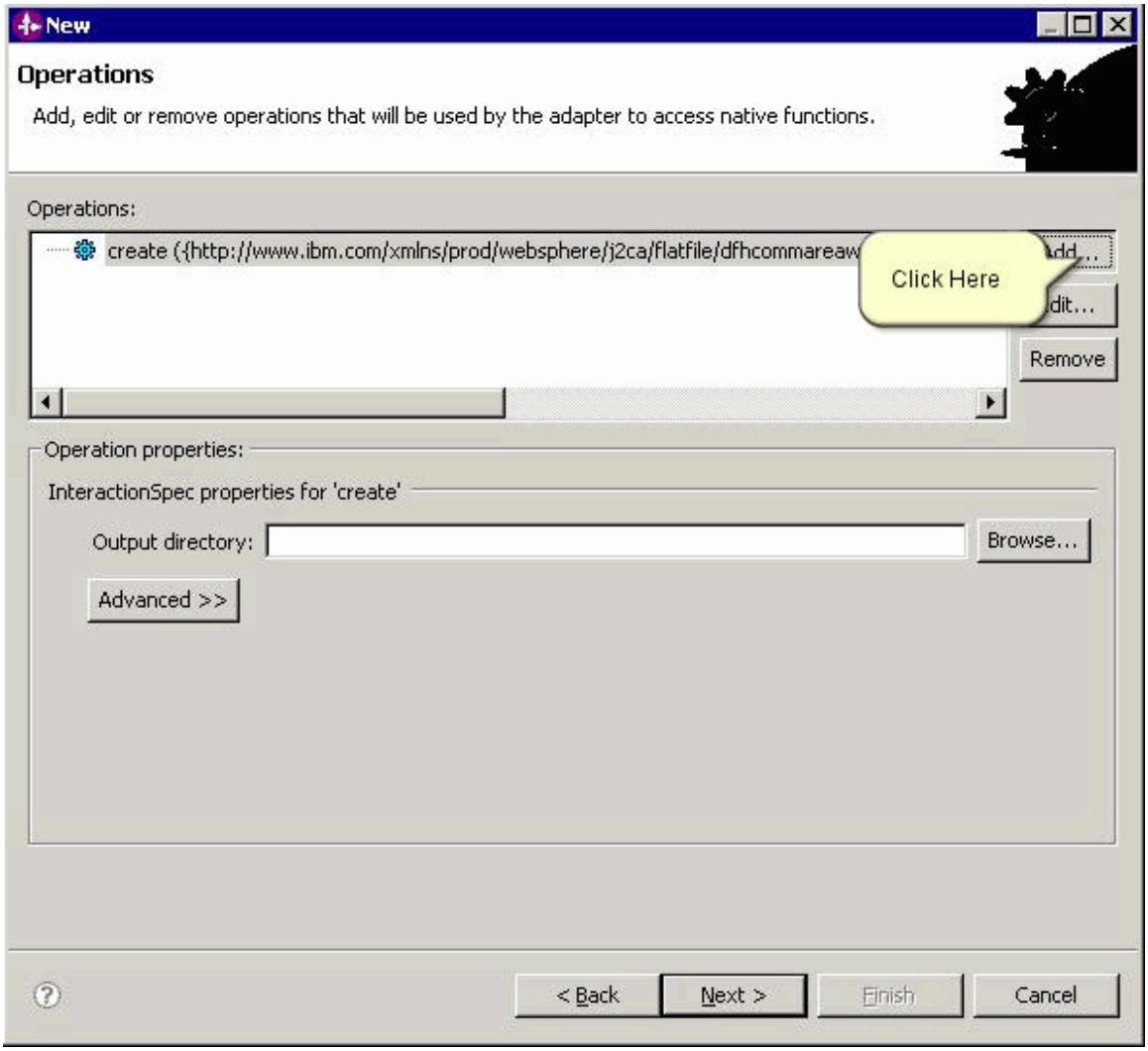
- Choose the wrapper generated previously as the input type and click **OK**.



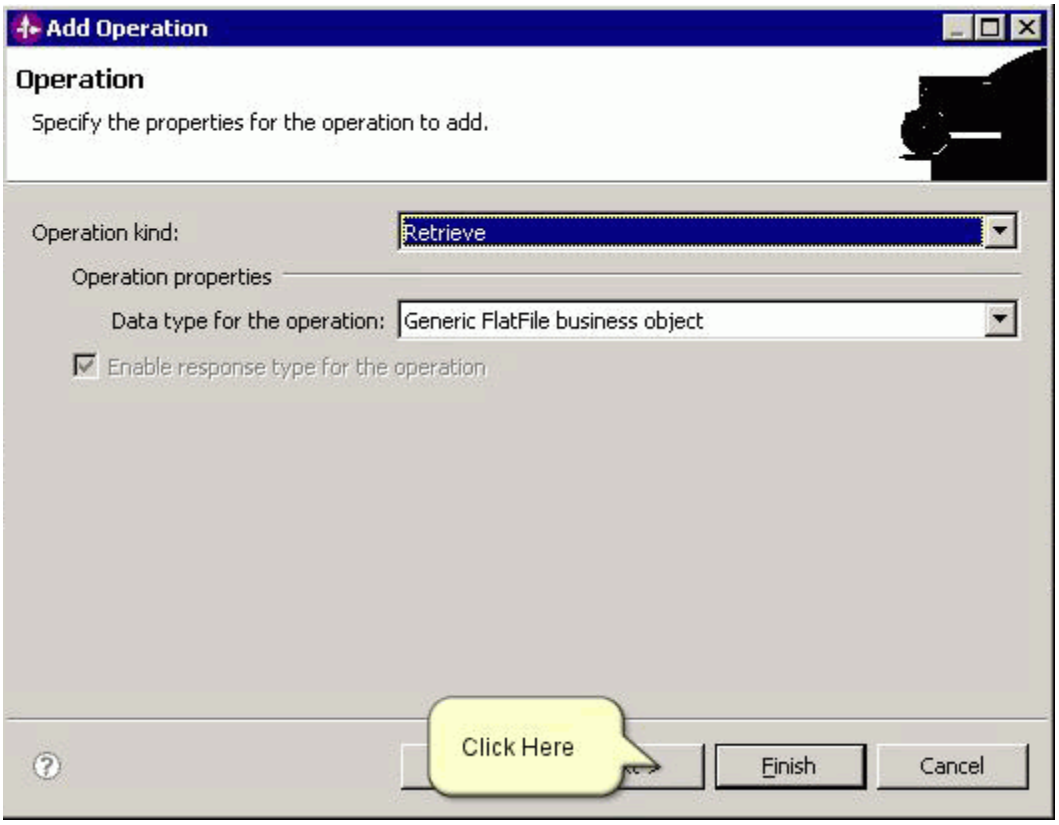
- The output type is already set by default. Click **Finish**.



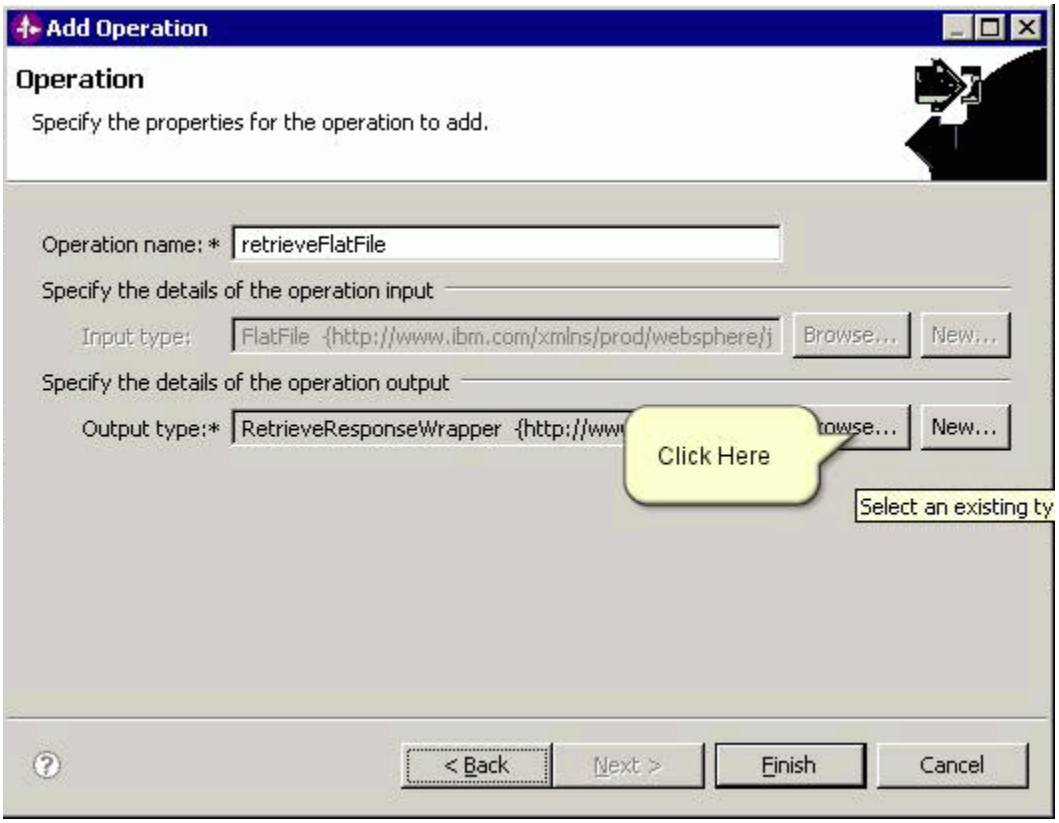
- The following screen capture shows the interaction specification properties for the Create operation. Do not set any values at the interaction specification level for this scenario.
- To add the Retrieve operation, click on the **Add** button.



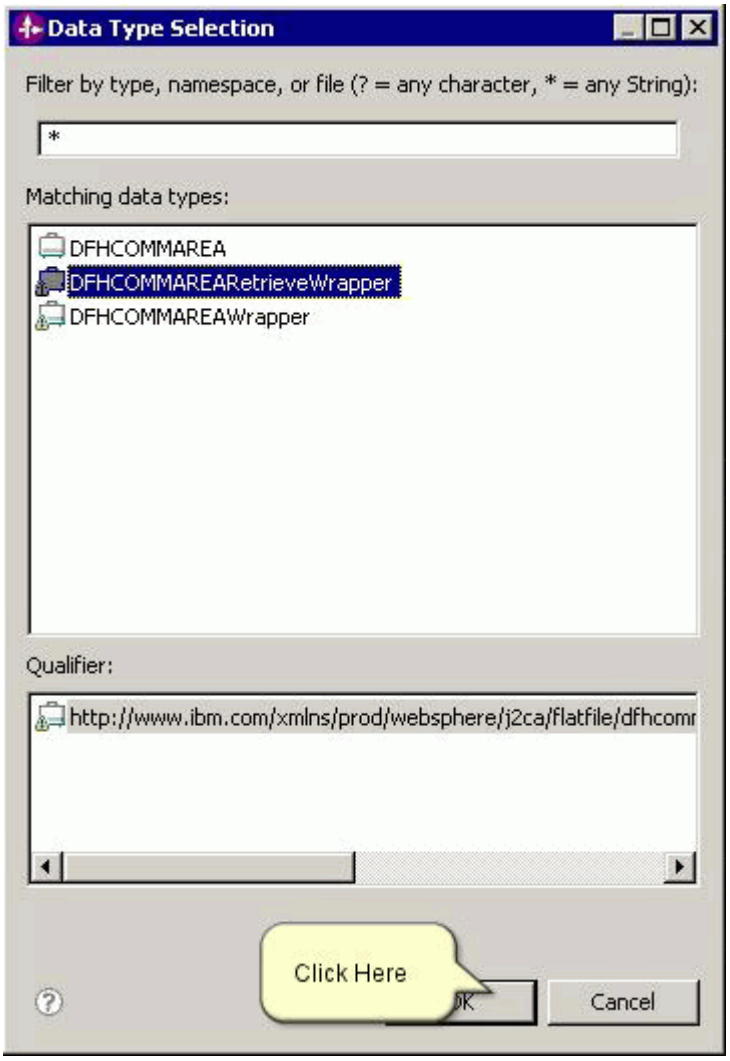
- Choose the **Operation kind** as **Retrieve** and set the **Data type** as **Generic FlatFile business object** or **Generic FlatFile business object with graph**.



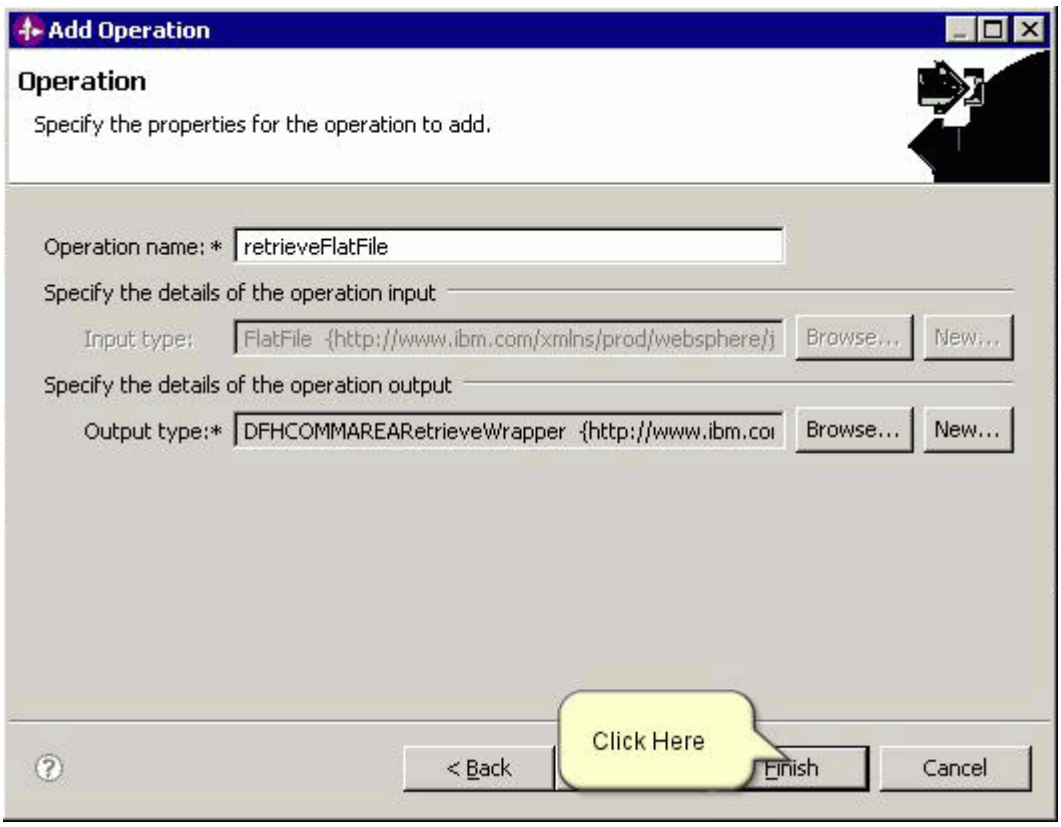
- For Retrieve operation, if the file to be retrieved is of structured type, set the **Output type**. Click on **Browse** to set the output type.



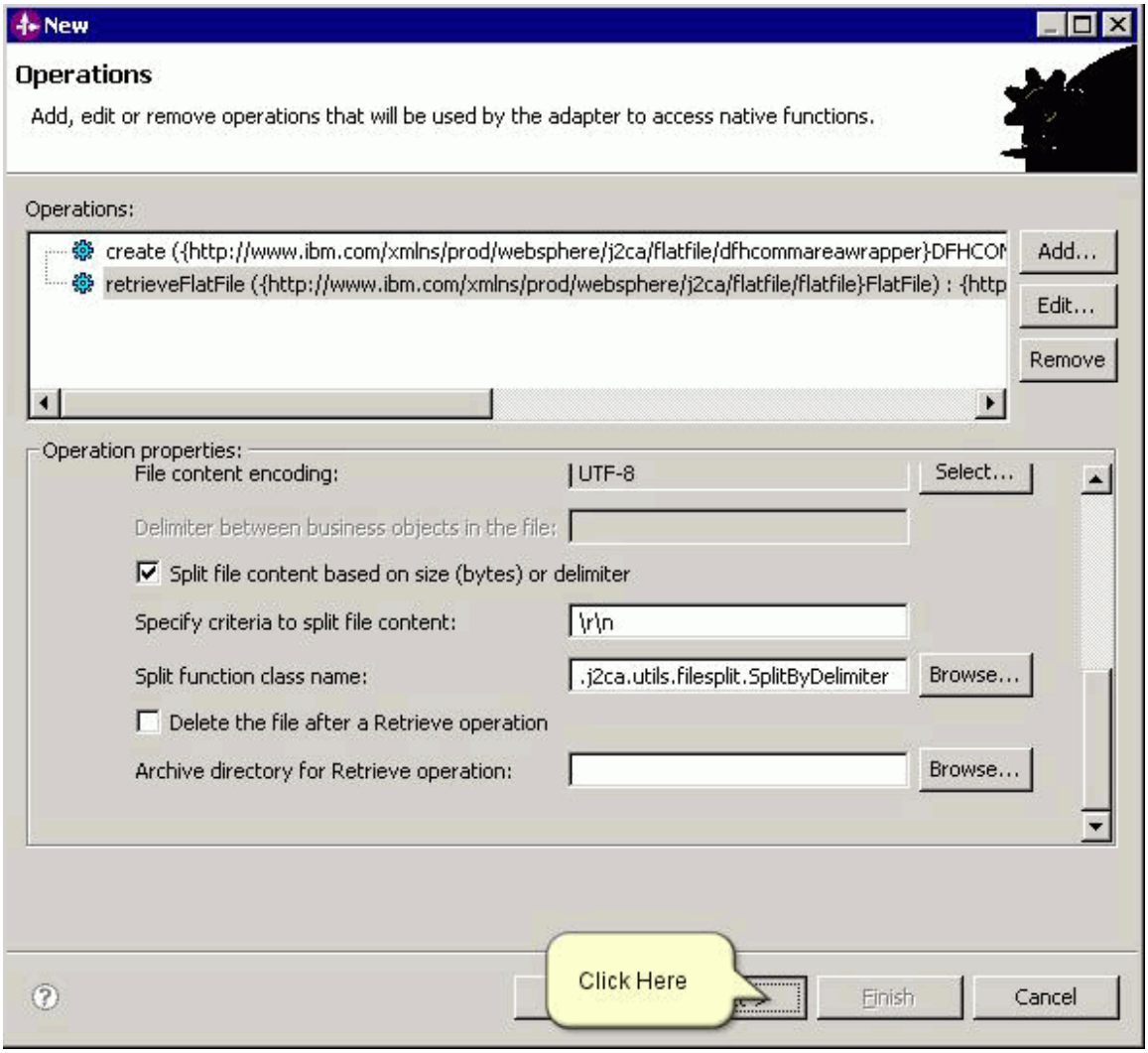
- Select the retrieve wrapper generated earlier as the output type. Click **OK**.



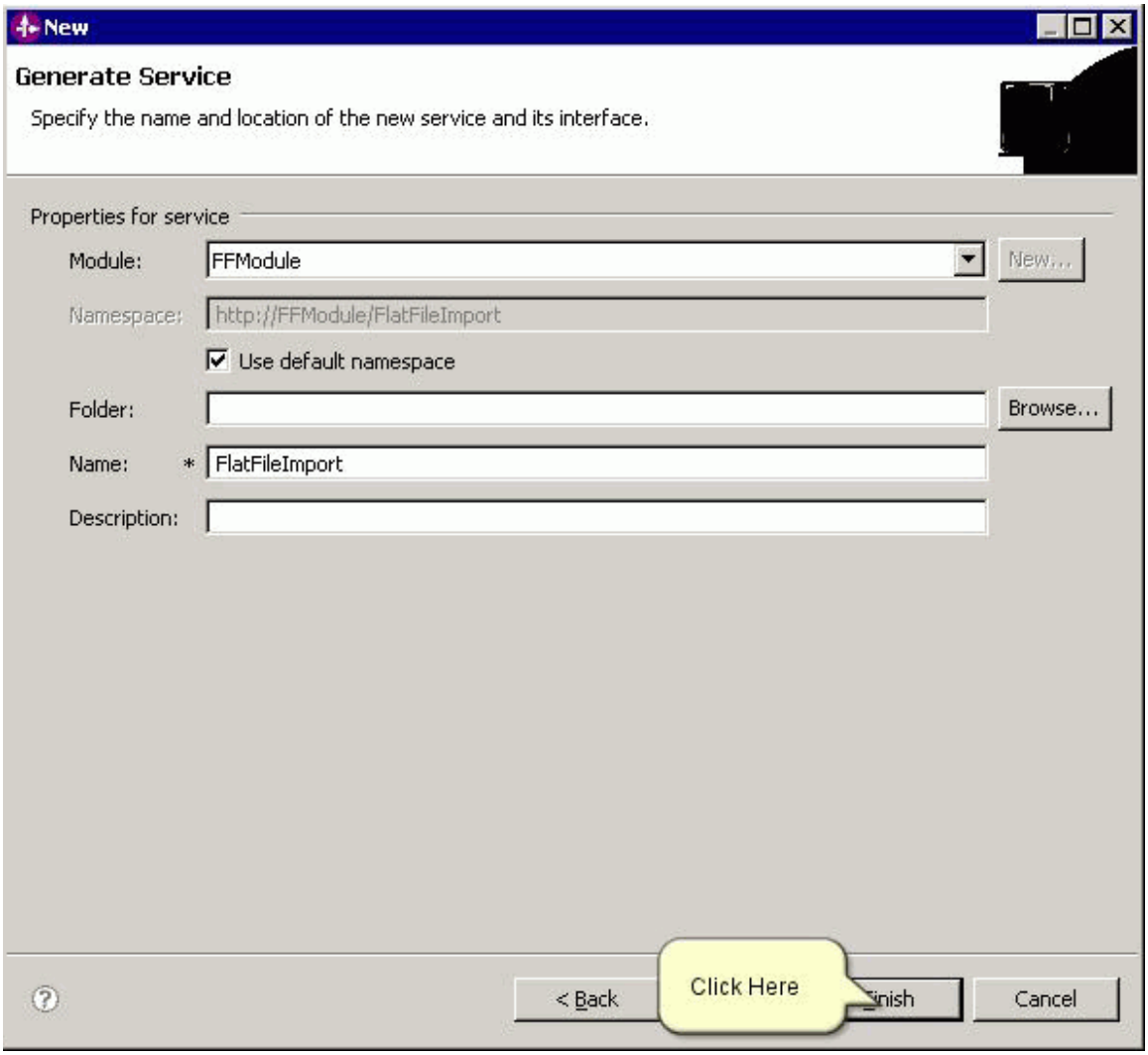
- The output type has been set. Click **Finish**.



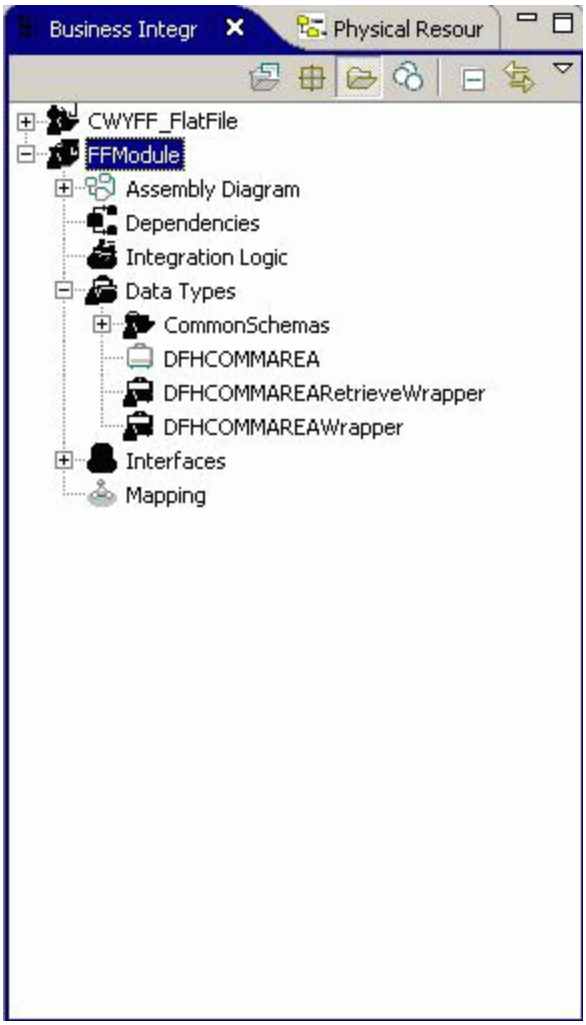
- In the following screen capture, the interaction specification properties for Retrieve operation are shown. Set the **Split criteria** and the **Split function class name**. This will enable splitting of content within the file being retrieved. The split criteria is set to **new line** and the splitting function class name is set to **SplitByDelimiter**, where the delimiter is new line as set in the split criteria. Click **Next**.



- The next screen capture shows the name with which the import file will be created. Click **Finish**.



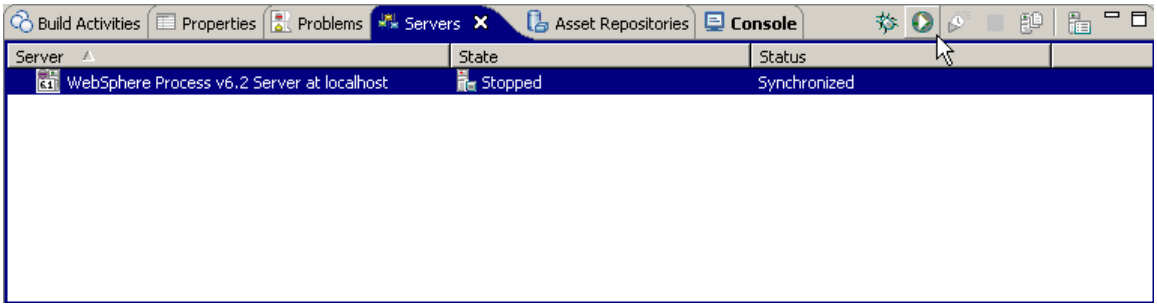
- The following screen capture shows the module in the BI perspective.



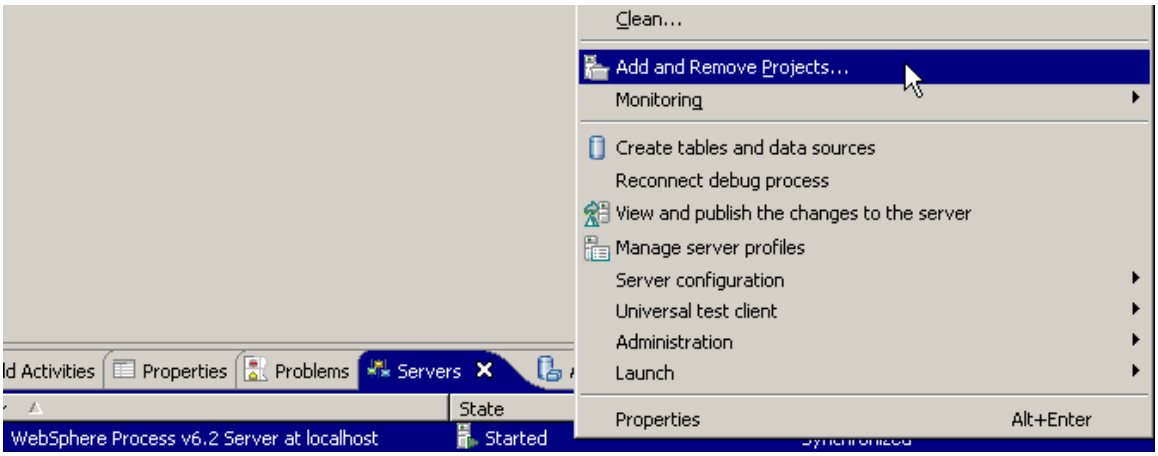
Deploying the module to the test environment

Next, deploy the module to the test environment and test the Create and Retrieve operations.

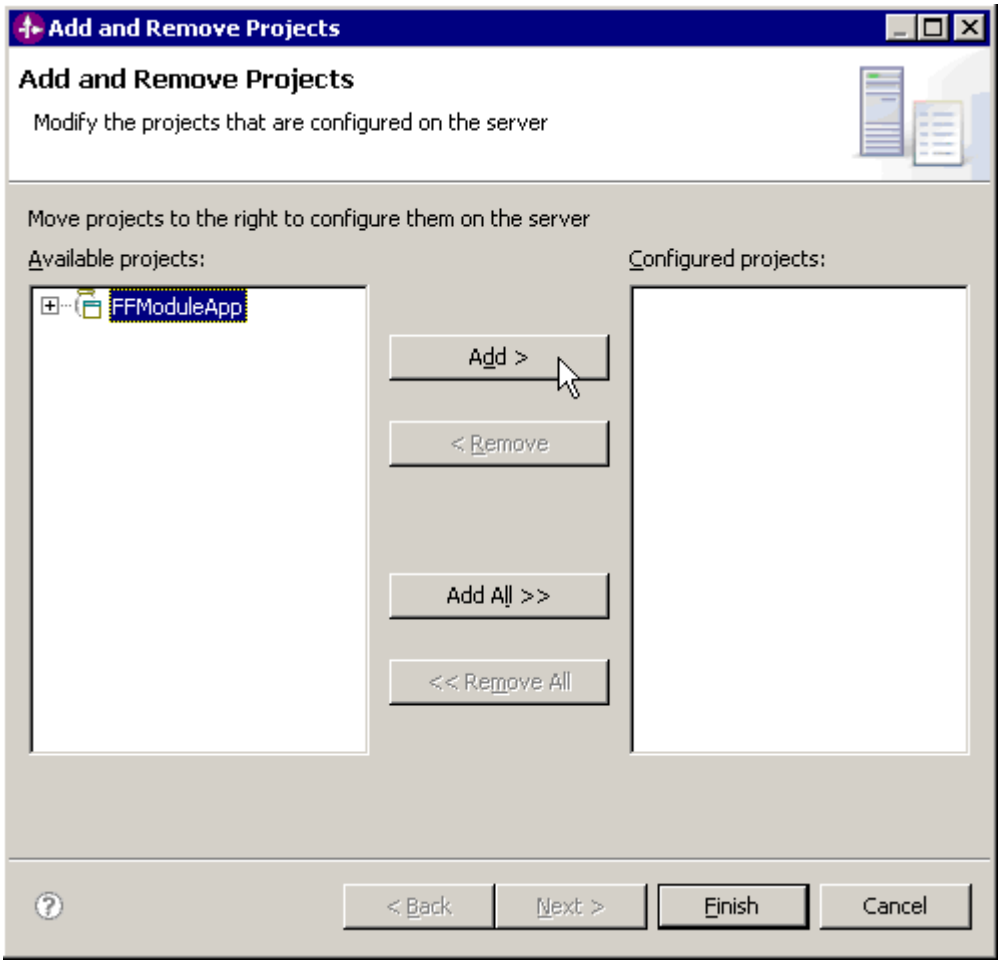
- In the **Servers** tab, if the server is not already started, click on the green start button highlighted in the figure below.



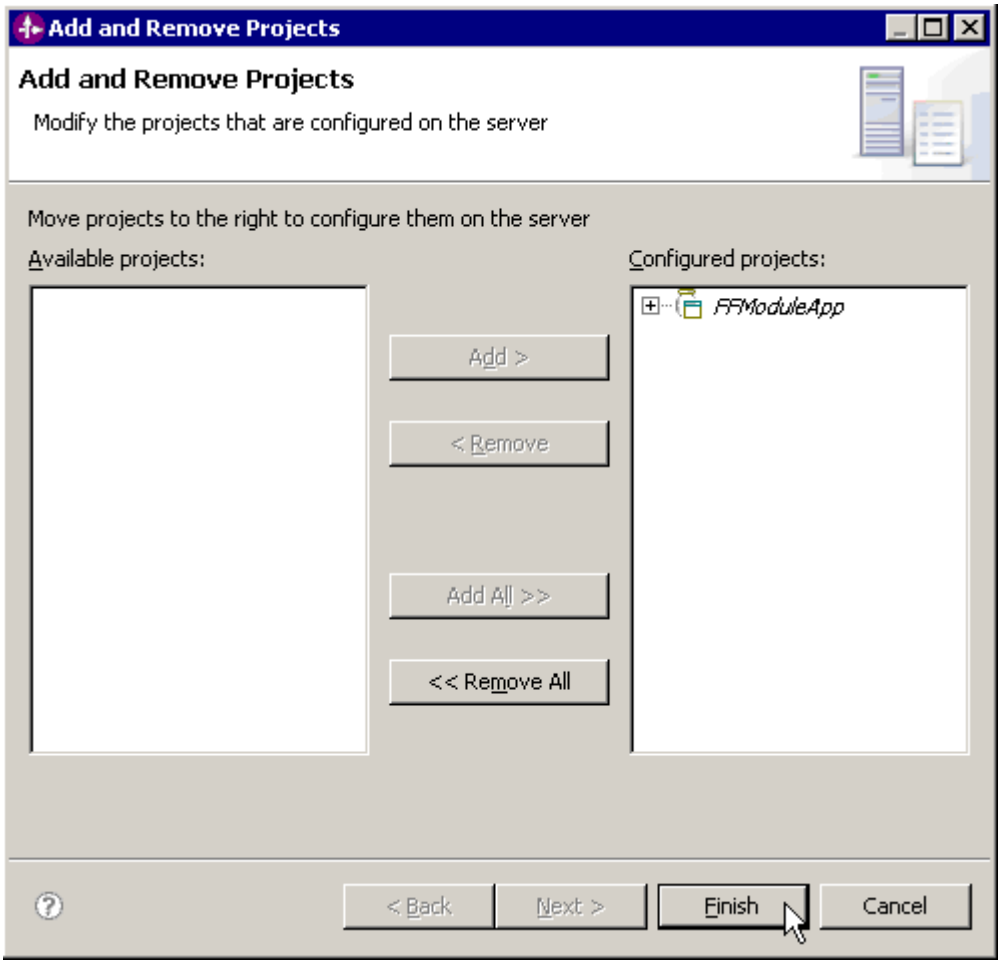
- To deploy the module, right click on the server and click **Add and Remove Projects…**



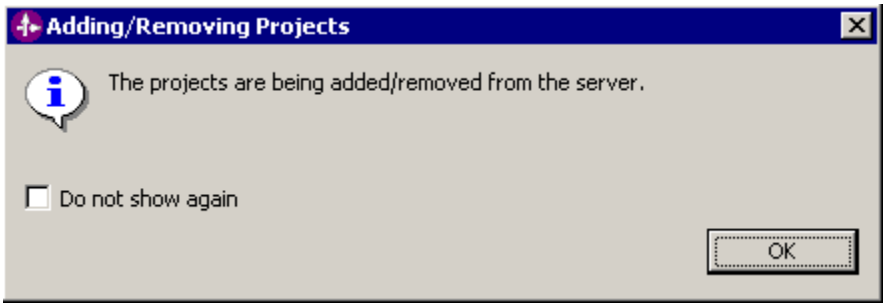
- Choose the module that was created and click the **Add >** button.



- Click **Finish**.

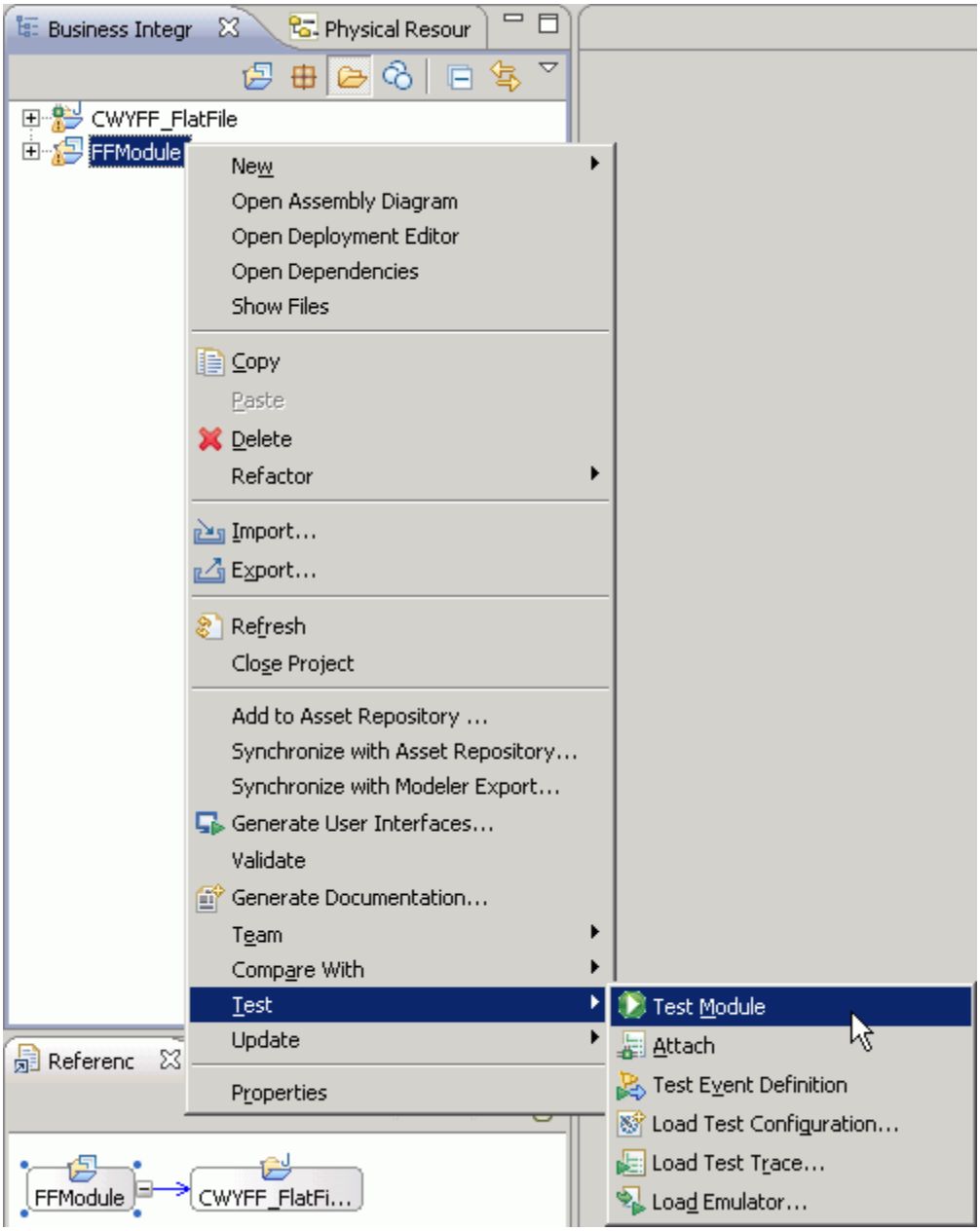


- On the Adding/Removing Projects window, click **OK**.

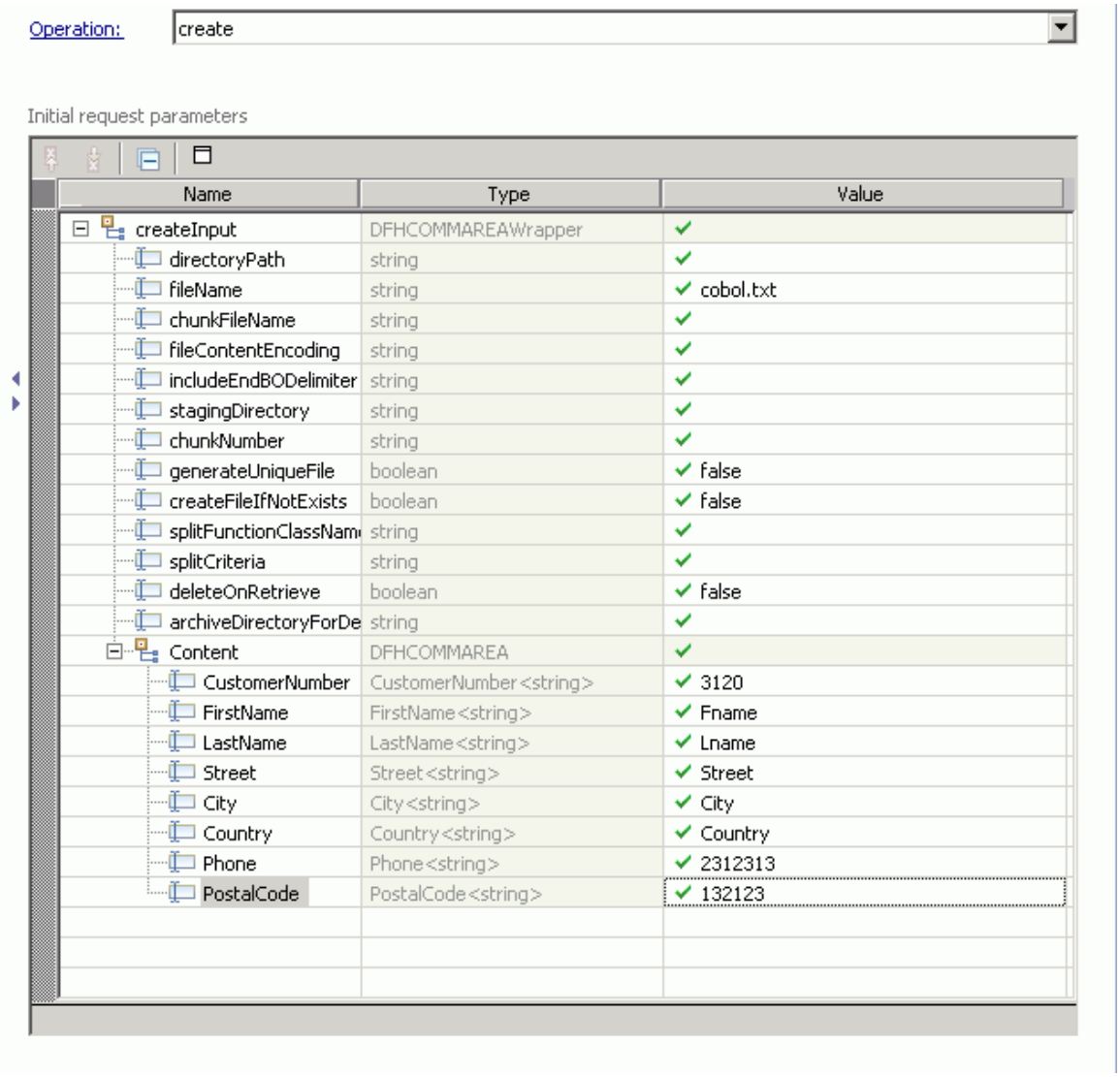


Testing the assembled adapter application

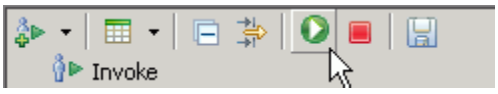
- Right click on the **module** and select **Test → Test Module**



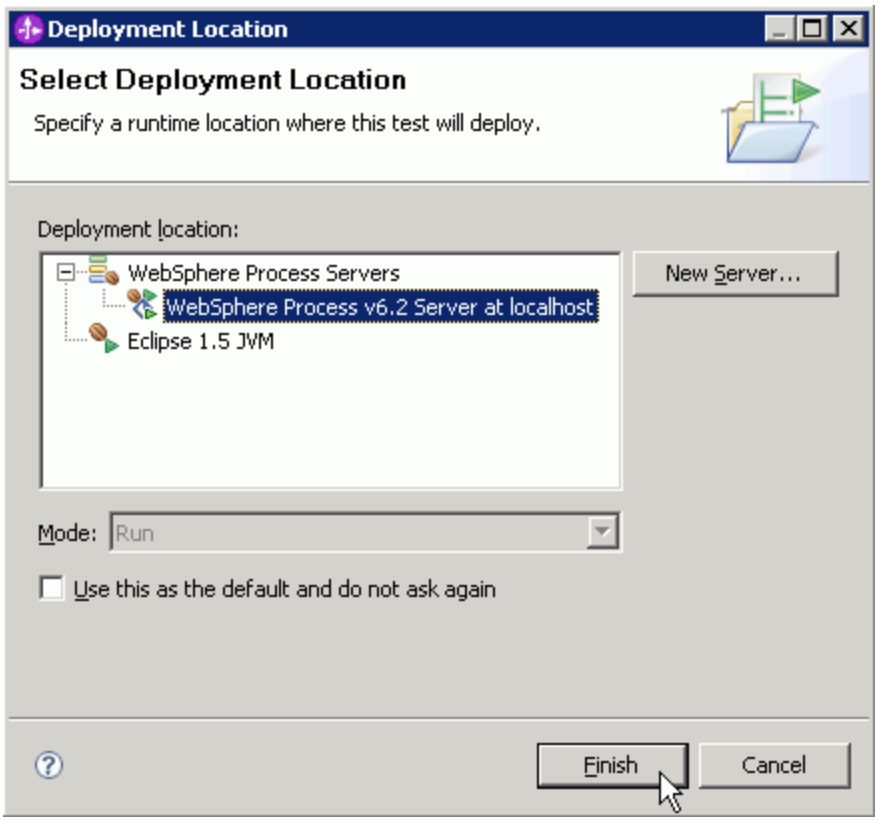
- Select the Create operation. Set the **fileName** property as well as the **Content** attributes as shown in the screen capture below.



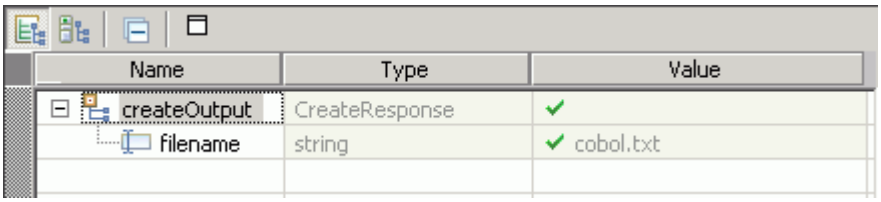
- Click on the **Continue** button highlighted in the screen capture below.



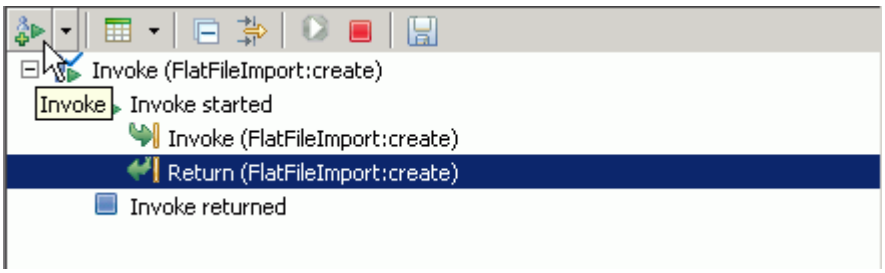
- In the Select Deployment Location window, click **Finish**.



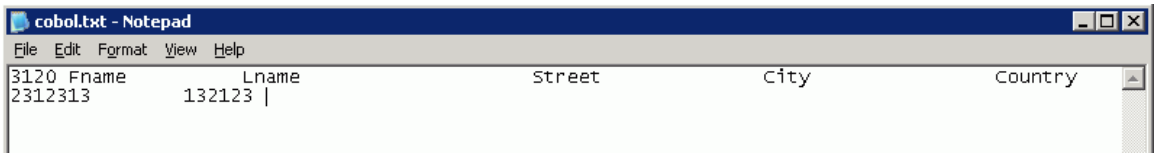
- The response object shows that the file with the name cobol.txt has been created.



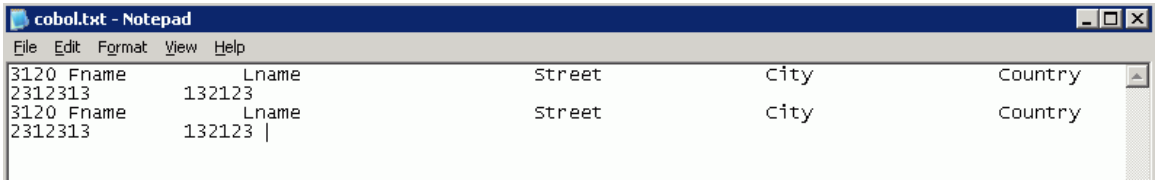
- To test the Retrieve operation, click on the **Invoke** button as shown in the screen capture below



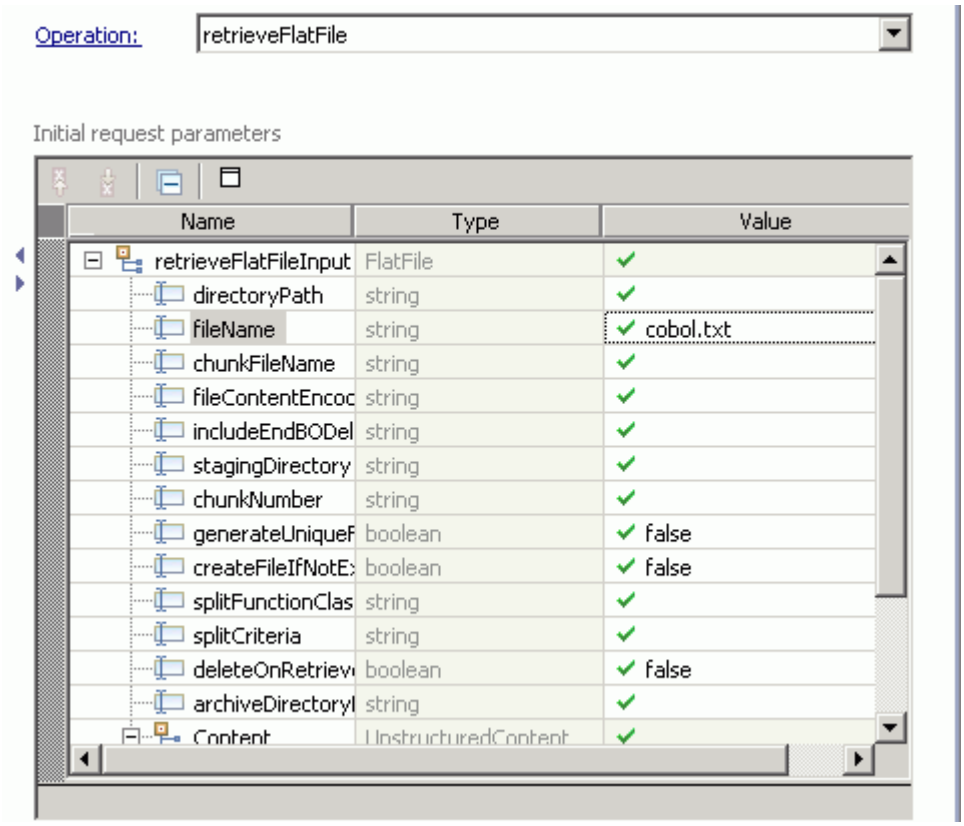
- To test retrieval of multiple data separated by a delimiter, set the delimiter as new line in the interaction specification for Retrieve operation. Open the file that was created by the Create operation.



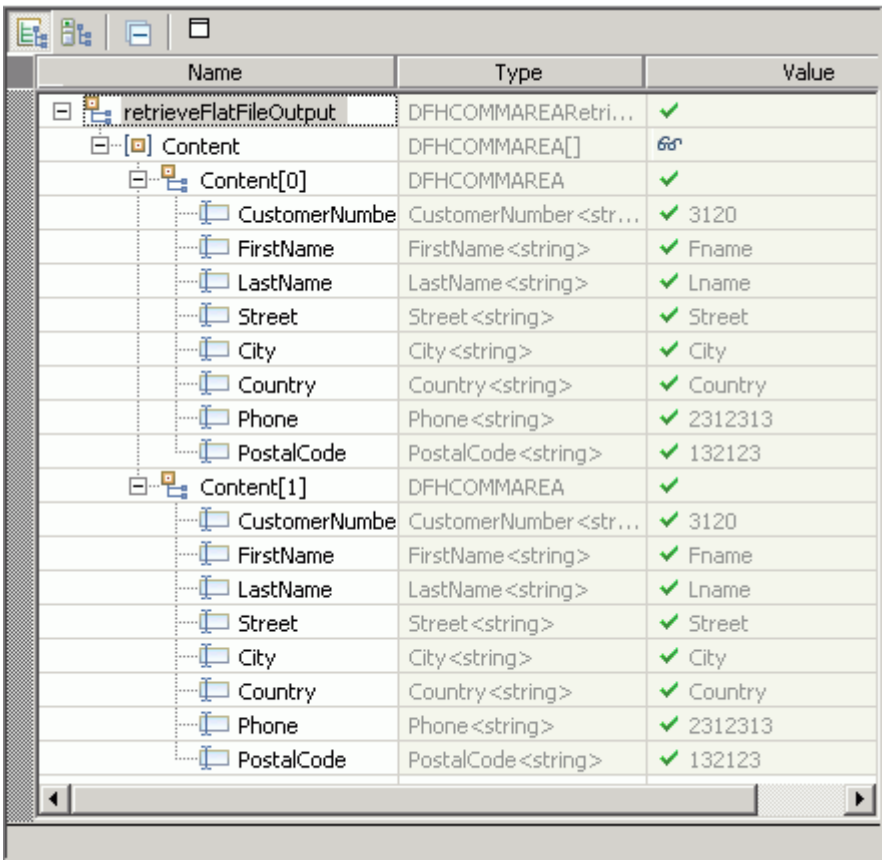
- Copy the content and paste it on a new line. So, now, there are two records that will be returned when this file is retrieved.



- Set the operation as retrieve and set the **fileName** of the file to be retrieved as cobol.txt and click the **Continue** button.



- The output will be as shown in the screen capture below.


Name
Type
Value
retrieveFlatFileOutput
DFHCOMMAREARetri...
Content
DFHCOMMAREA[]
Content[0]
DFHCOMMAREA
CustomerNumbe
CustomerNumber<str...
3120
FirstName
FirstName<string>
Fname
LastName
LastName<string>
Lname
Street
Street<string>
Street
City
City<string>
City
Country
Country<string>
Country
Phone
Phone<string>
2312313
PostalCode
PostalCode<string>
132123
Content[1]
DFHCOMMAREA
CustomerNumbe
CustomerNumber<str...
3120
FirstName
FirstName<string>
Fname
LastName
LastName<string>
Lname
Street
Street<string>
Street
City
City<string>
City
Country
Country<string>
Country
Phone
Phone<string>
2312313
PostalCode
PostalCode<string>
132123

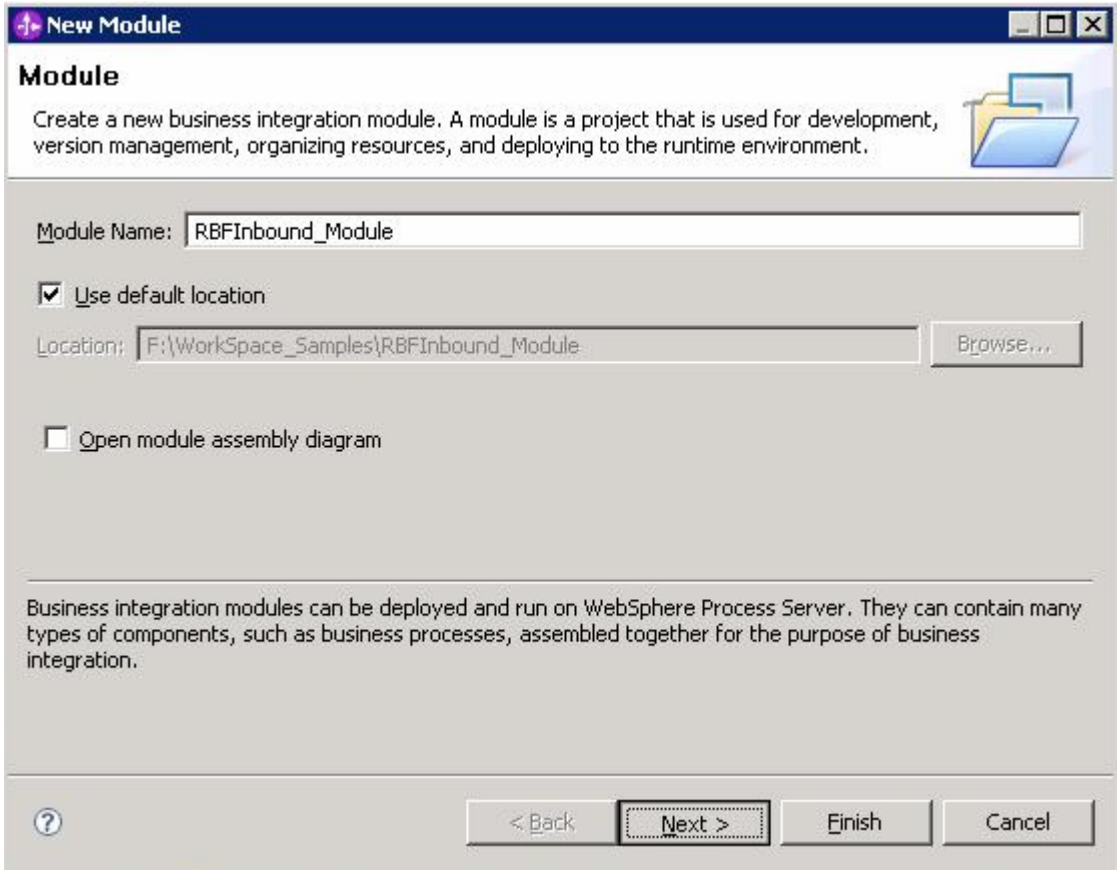
Chapter 9. Tutorial 7: Rule based filtering of inbound events (picking up event files based on rules)

This tutorial demonstrates how WebSphere® Adapter for FlatFiles 6.2.0.0 can be used to selectively filter and pick the polled files in an inbound scenario, through the use of configured rules and then send the events to the end-point. The adapter provides support during generation of artifacts using the External Service wizard to enter rules.

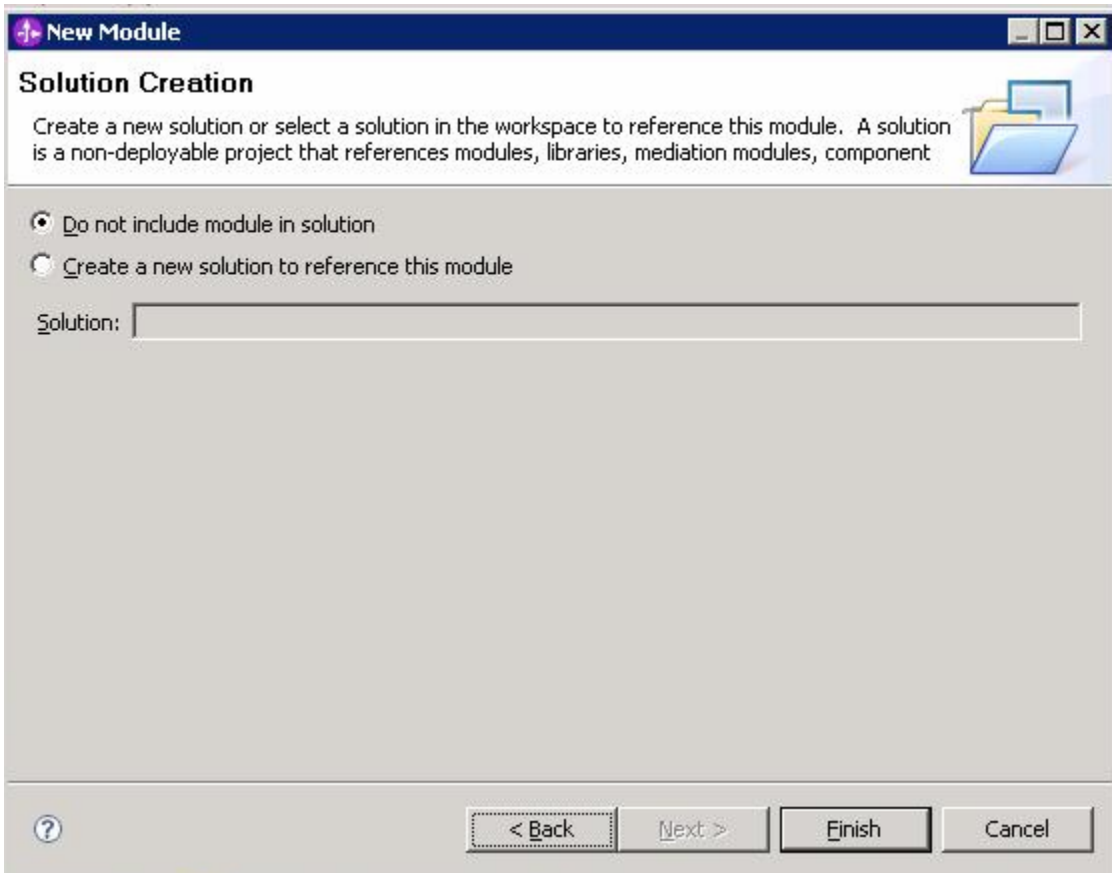
Configuring the adapter for inbound processing

Run the external service wizard to specify business objects, services, and artifacts to be used to run the scenario in this tutorial.

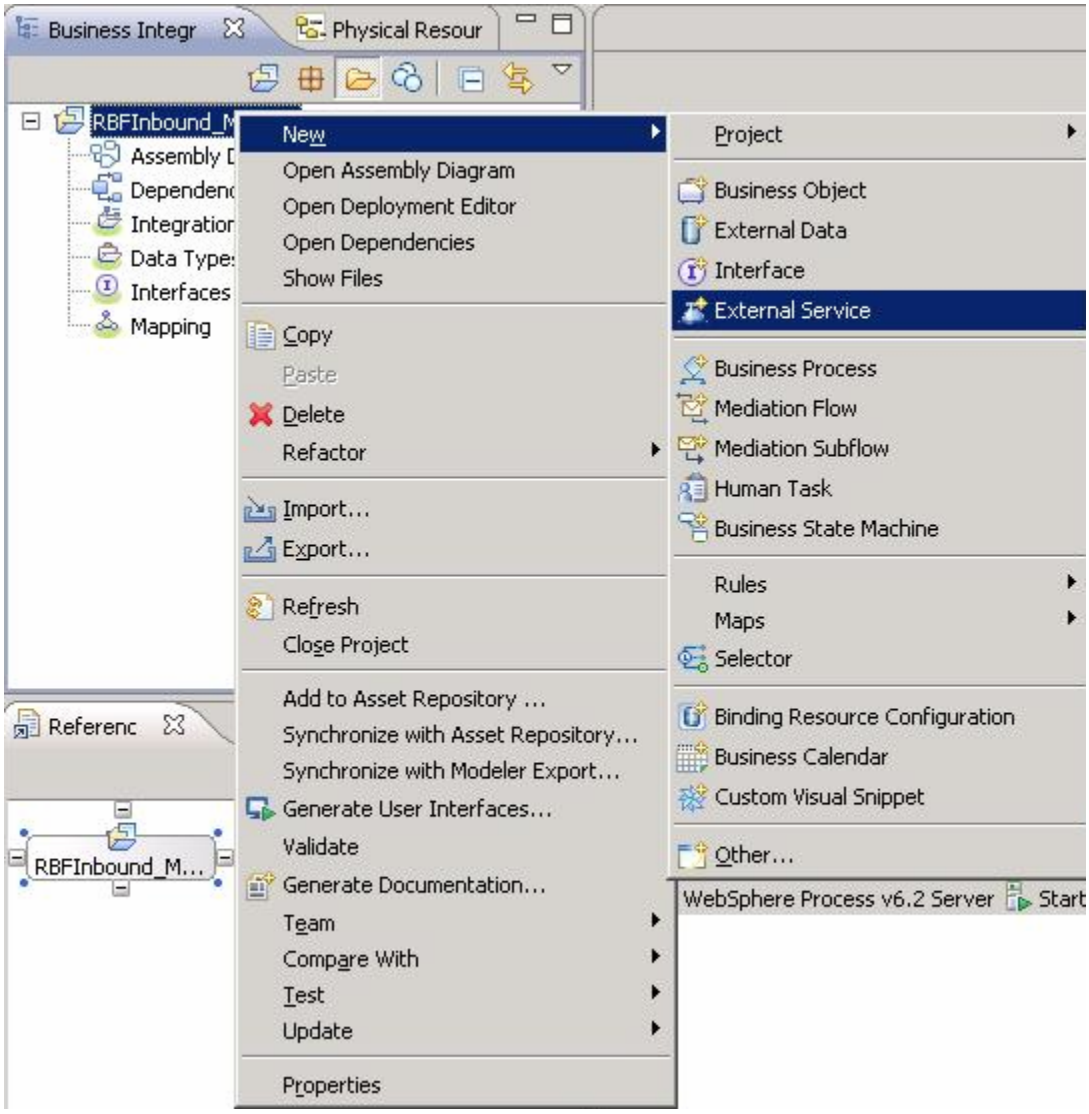
- Create a new module. Go to **File** -> **New** -> **Module**
- Enter a desired name for your module and click **Next**.



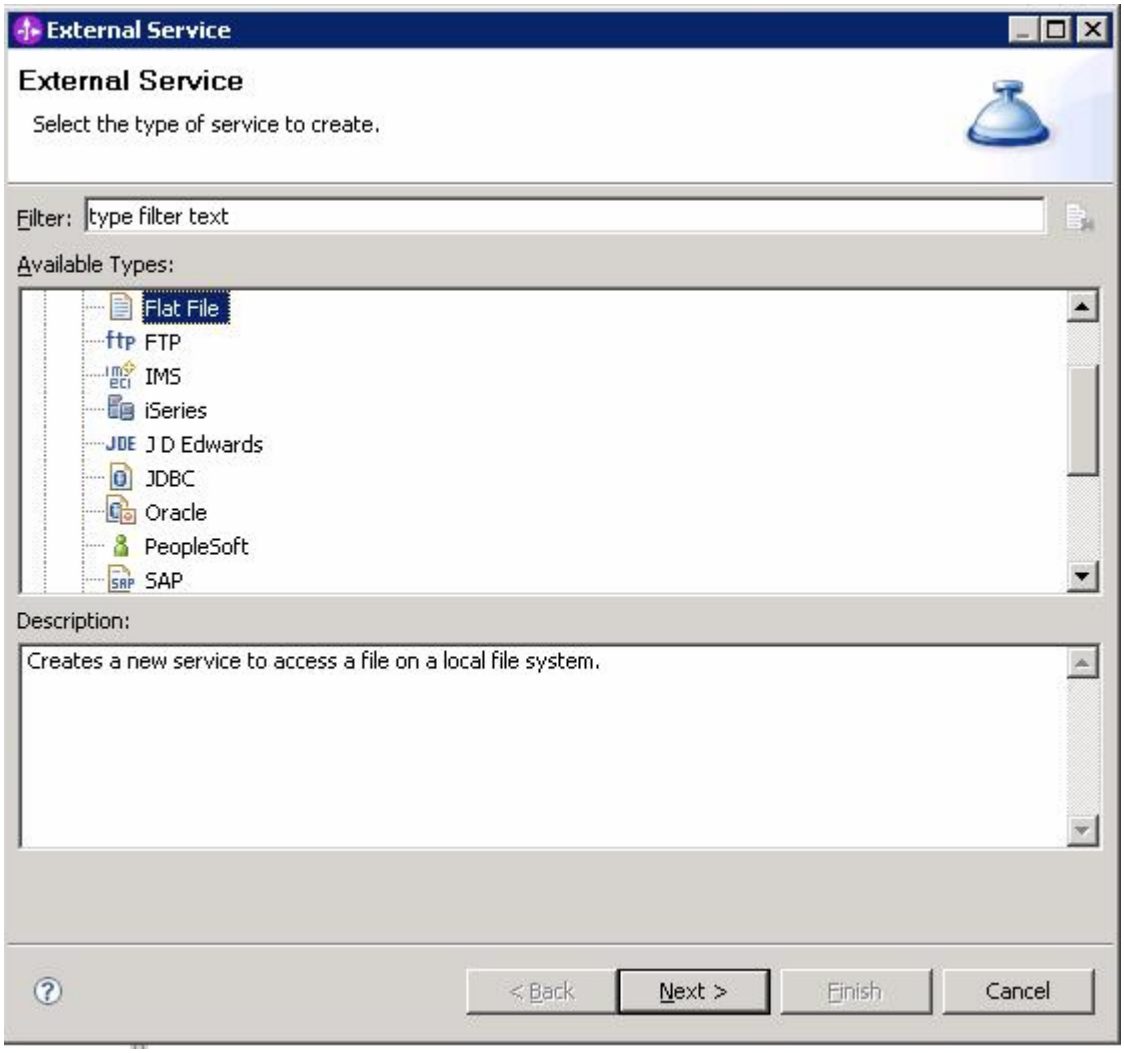
-
- Leave Do not include module in solution selected and click Finish.



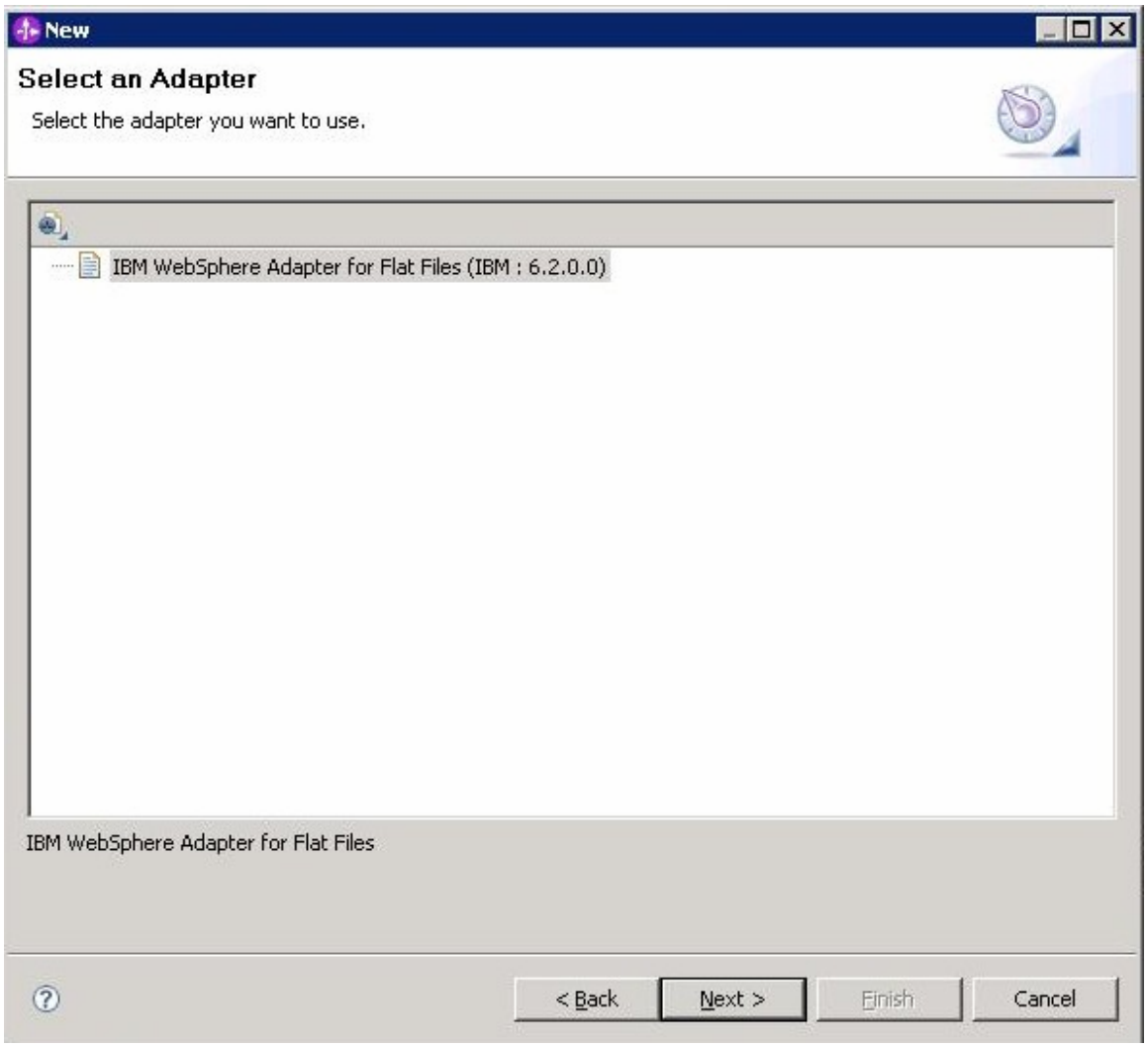
- Start the external service wizard by selecting File -> New -> External Service.



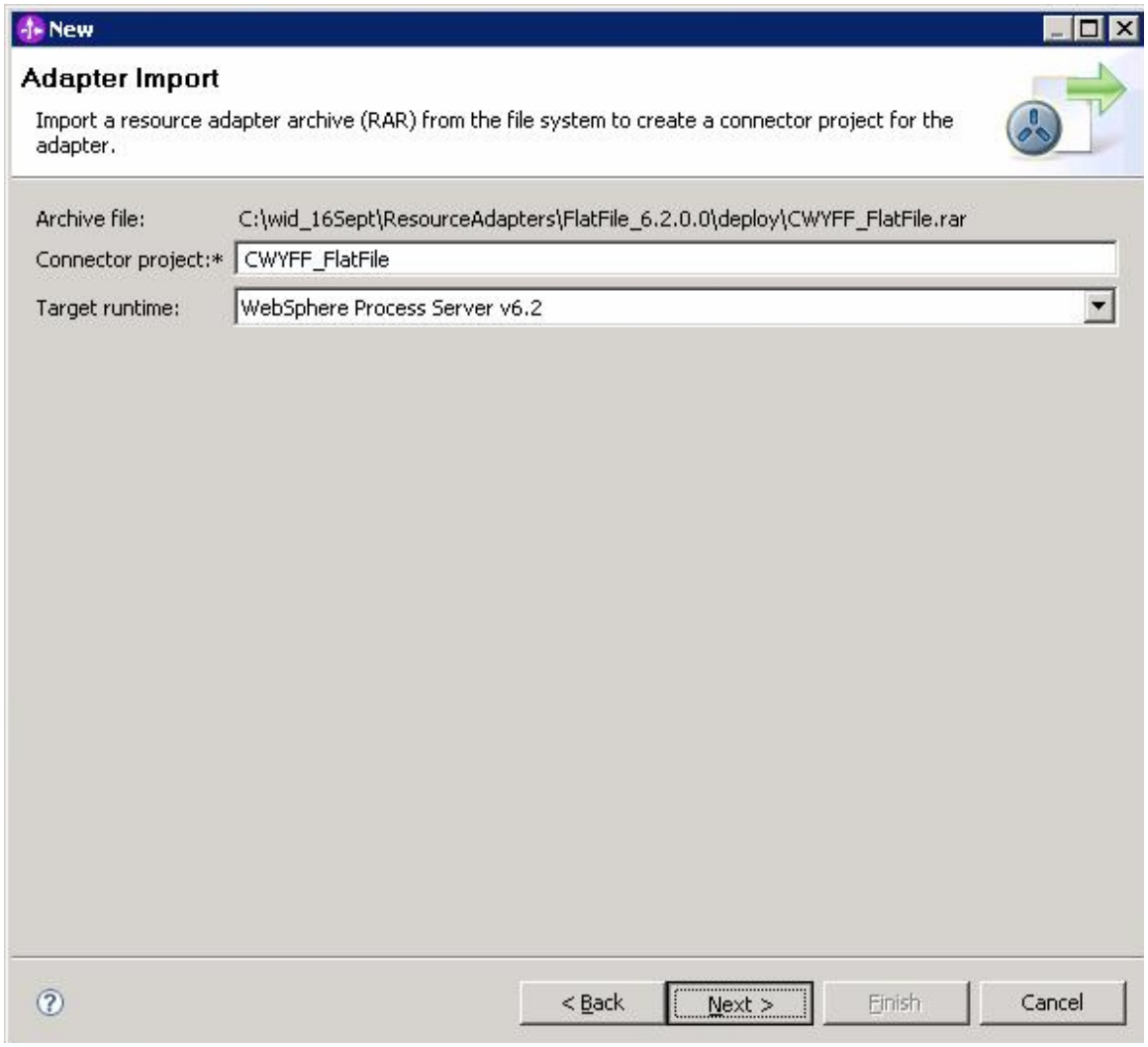
- Select **Adapters** and **FlatFile** and then click **Next**.



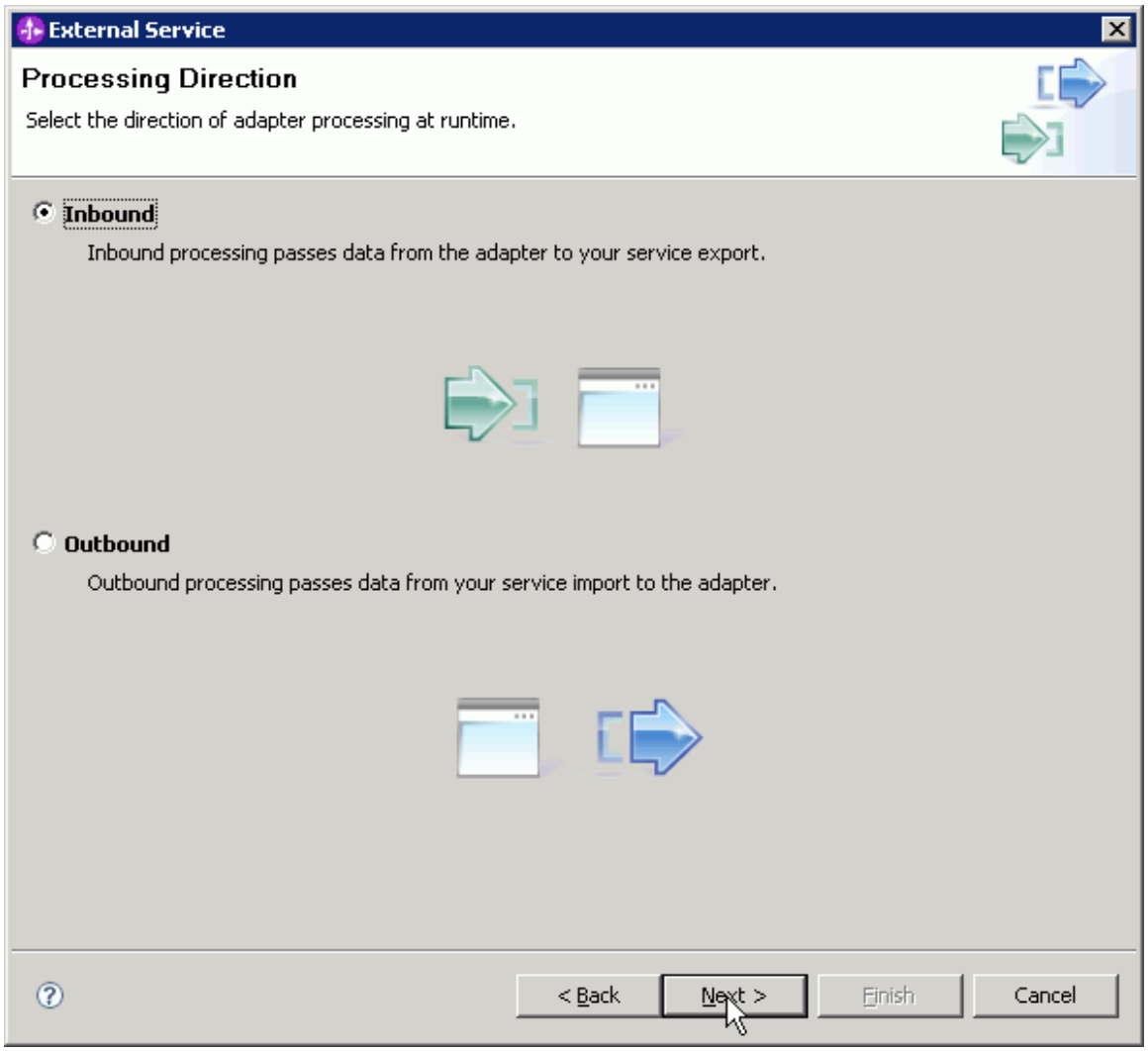
- In the Select an Adapter window, select IBM WebSphere Adapter for FlatFiles (IBM : 6.2.0.0). Click Next.



- In the Adapter Import window, adapter RAR file will be imported into the module. Leave the default value for **Connector project**. Ensure that the value set for **Target runtime** is **WebSphere Process Server v6.2** if it's not set by default. Click **Next**.

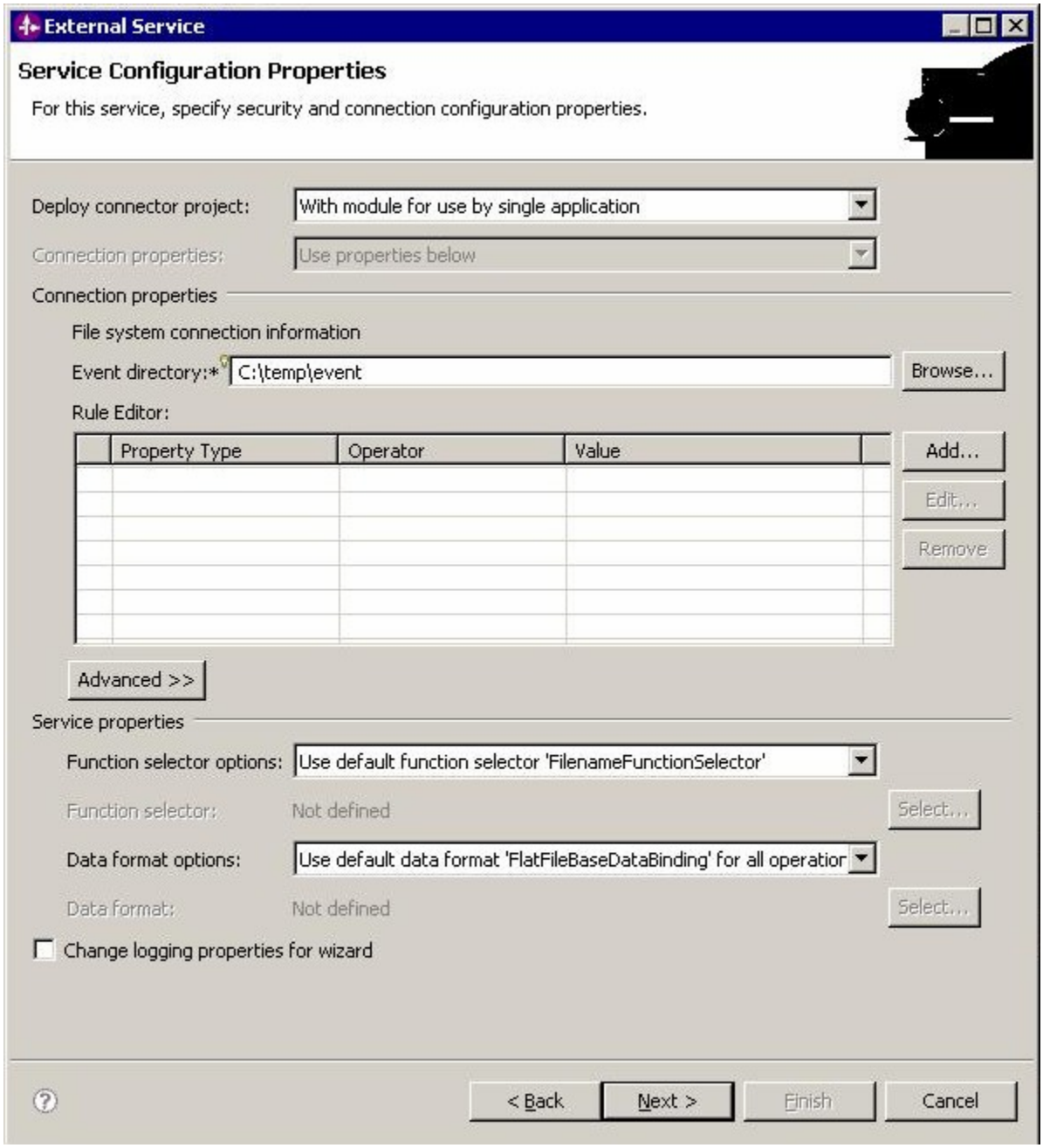


- To use rules to filter inbound events, in the Processing Direction window, select **Inbound** and click **Next**.

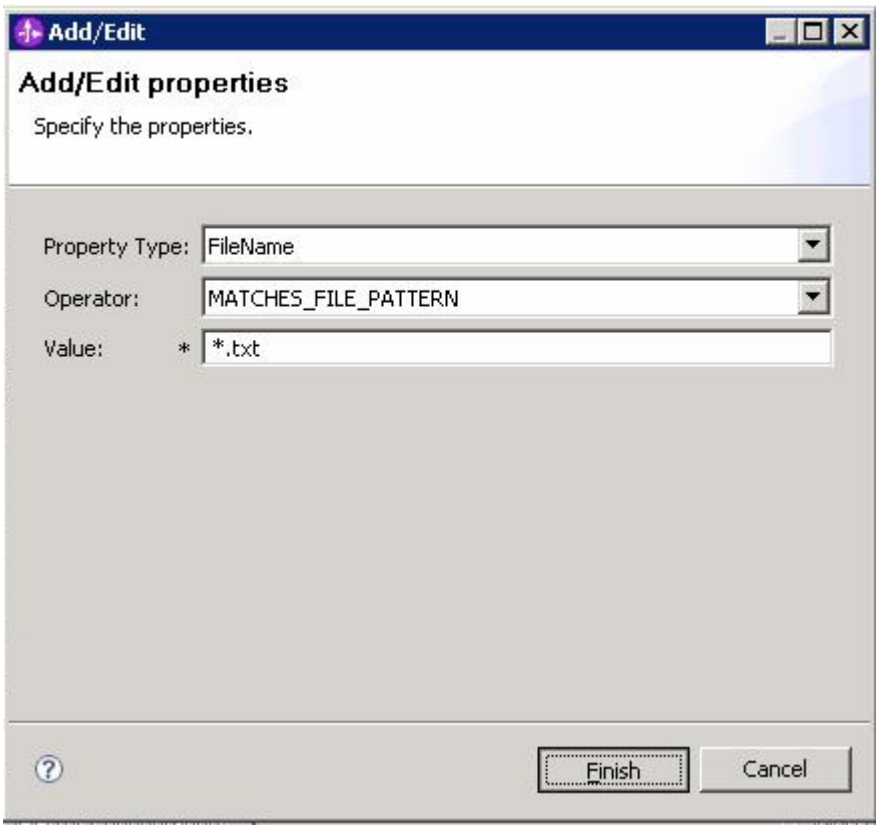


Setting properties for the external service wizard

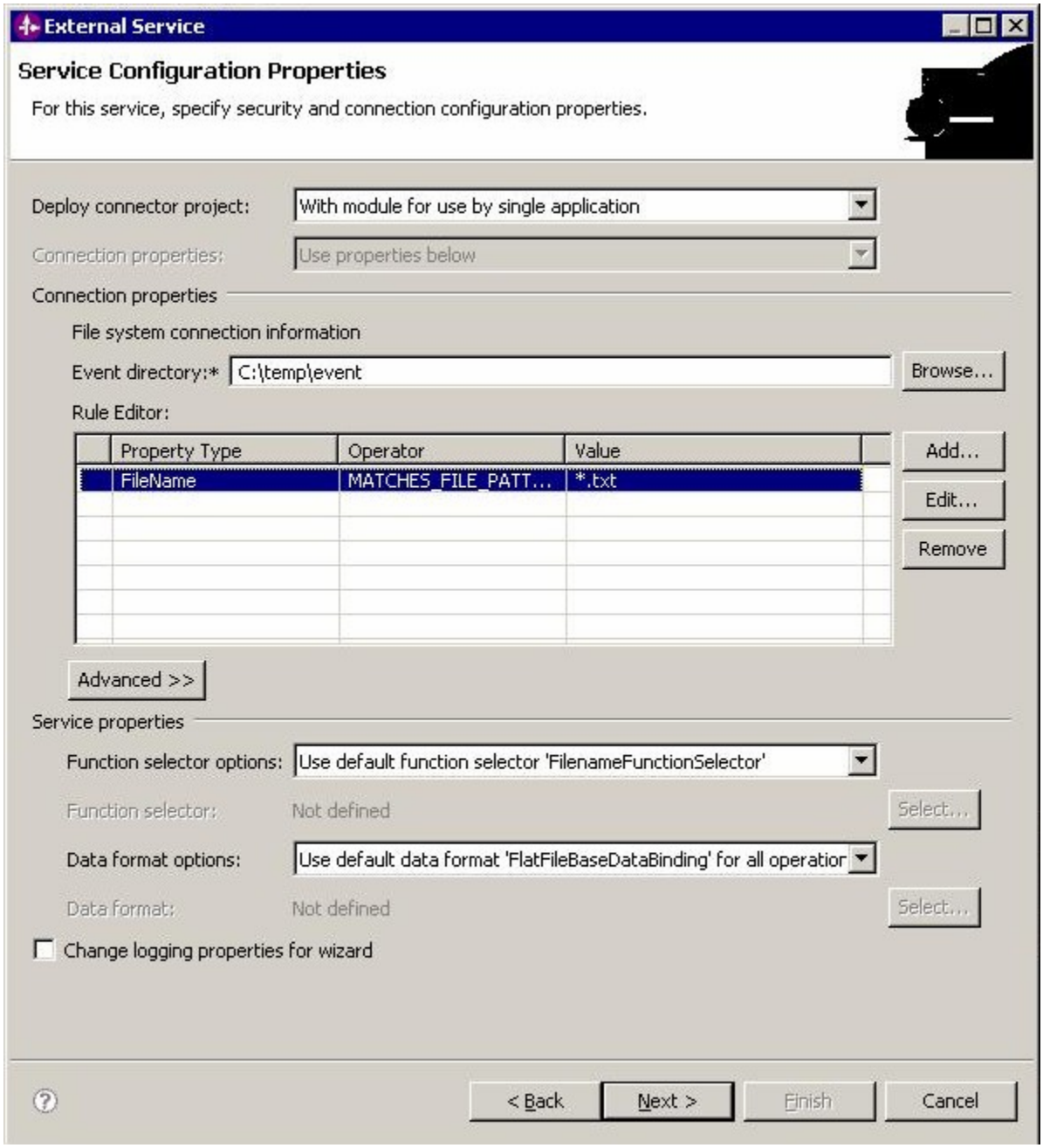
- In the Service Configuration Properties window shown below, specify values for **Event directory**, **Archive directory** and **Rule Editor**.
- **Event recovery table name** and **Event recovery data source (JNDI) name** can be left blank in which case in-memory processing of file is carried out.



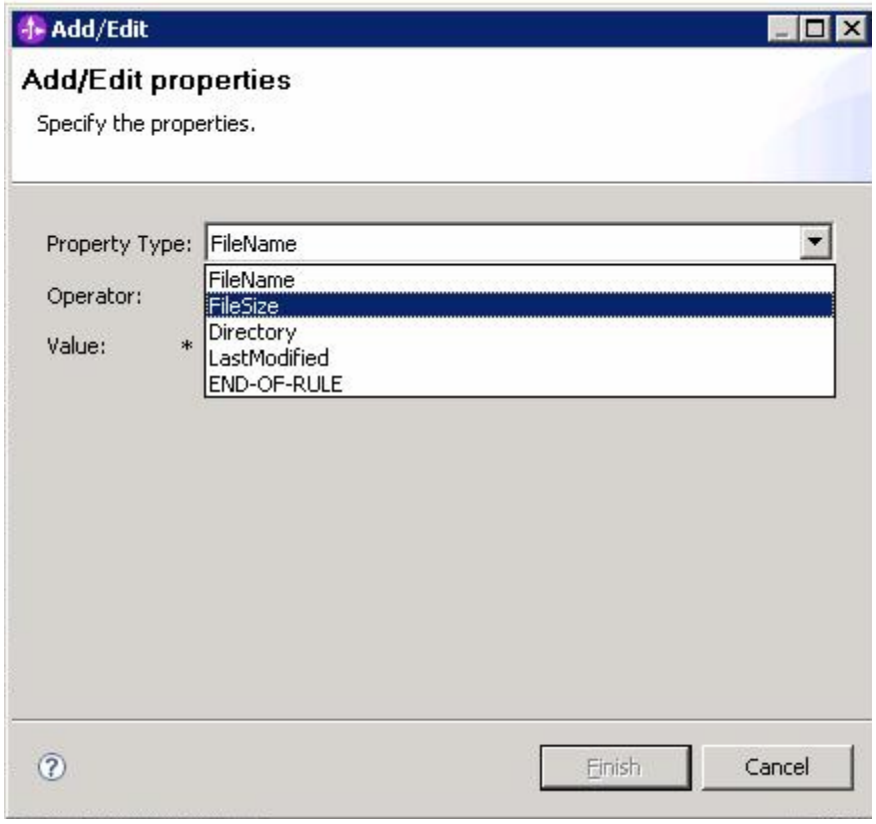
- Configure the rules to filter out polled flat files. This tutorial will illustrate how to configure rules to pick files that either have a file name with a *.txt extension or a file size greater than or equal to 1000 KB. To configure one rule click the **Add** button near the **Rule Editor** table in the previous window.



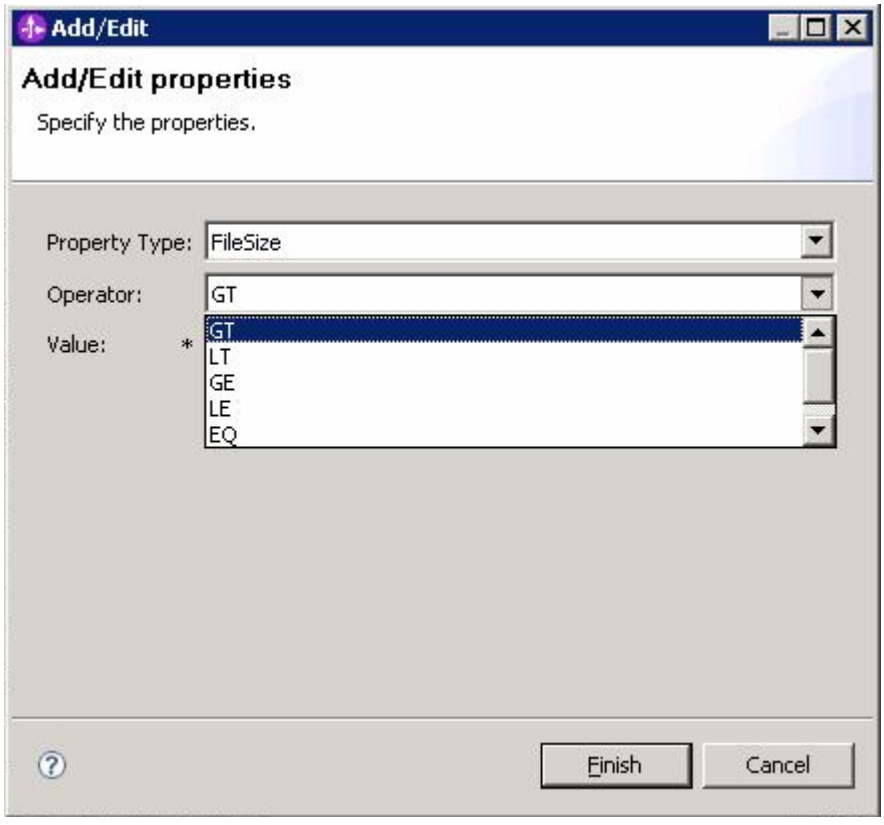
- In the Add/Edit properties window, select **FileName** for the **Property Type** list, then select MATCHES_FILE_PATTERN from the **Operator** list. Type ***.txt** in the **Value** field and click **Finish**. The result of the above will be as shown in the screen capture below.

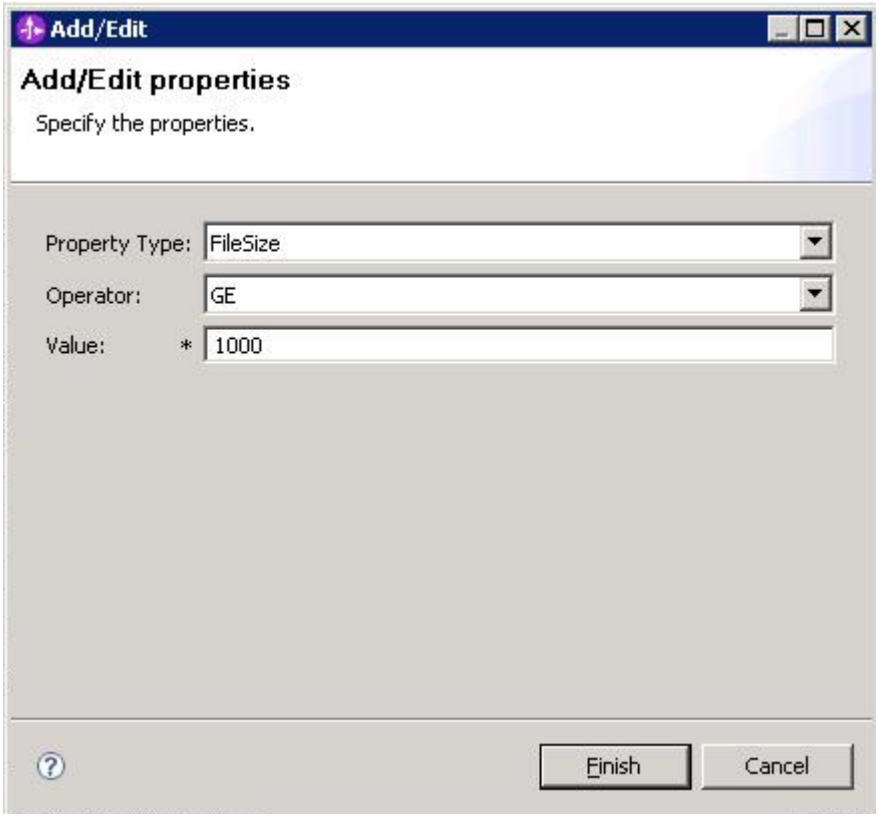


- Click **Add** to specify another rule. This rule will be logically ORed with the previous rule.

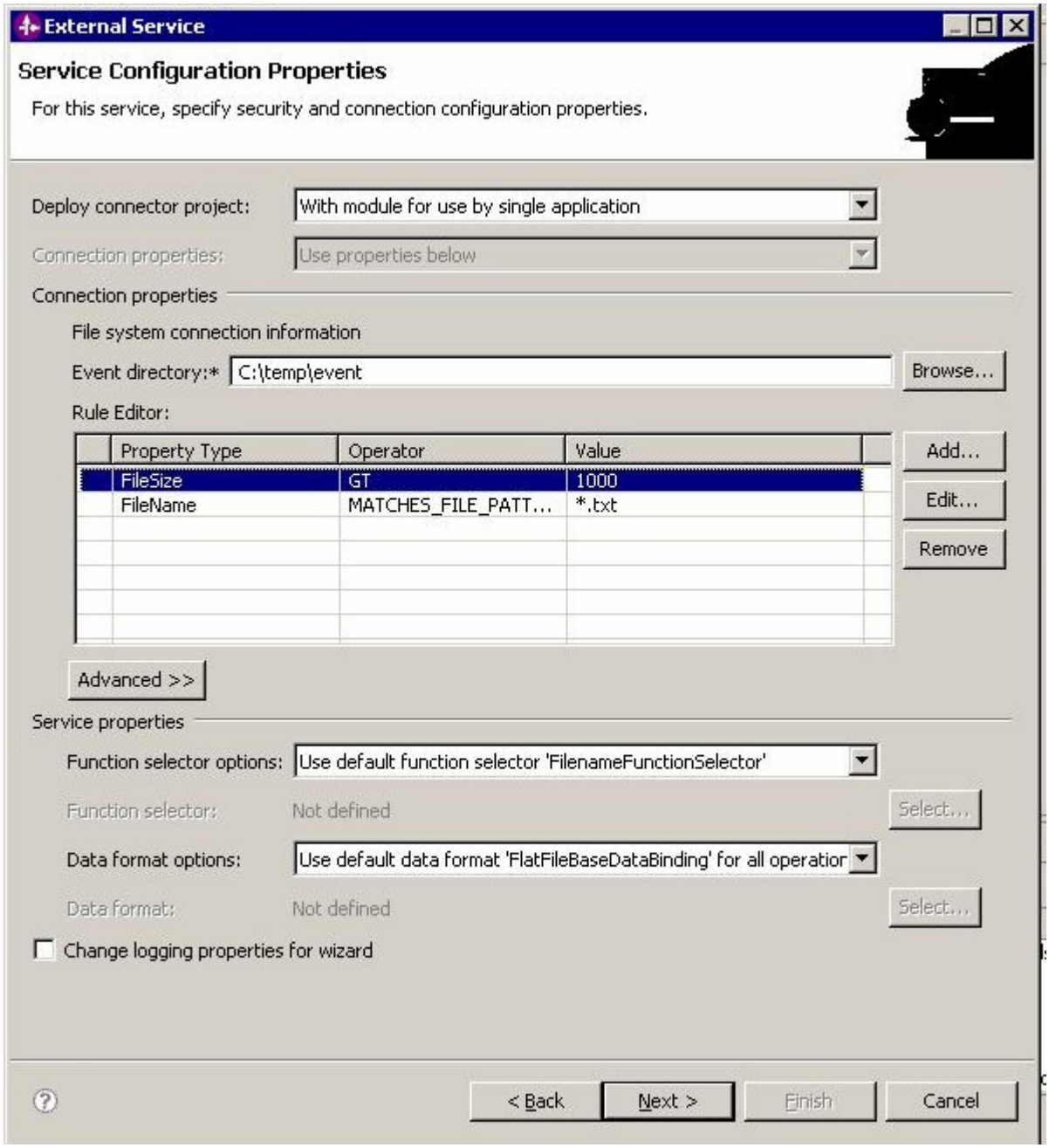


- In the Add/Edit properties window, select **FileSize** from the **Property Type** list.



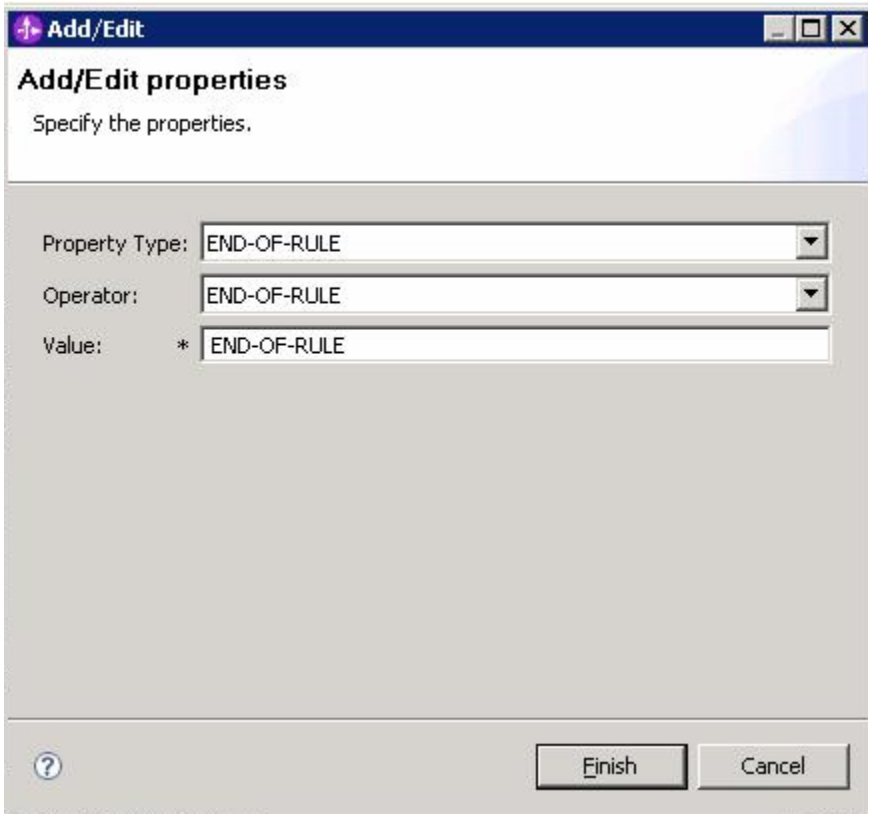
- Select **GE** as the corresponding **Operator**. (GE means Greater Than).
- 

- Specify **1000** in the **Value** field, and then click **Finish**.



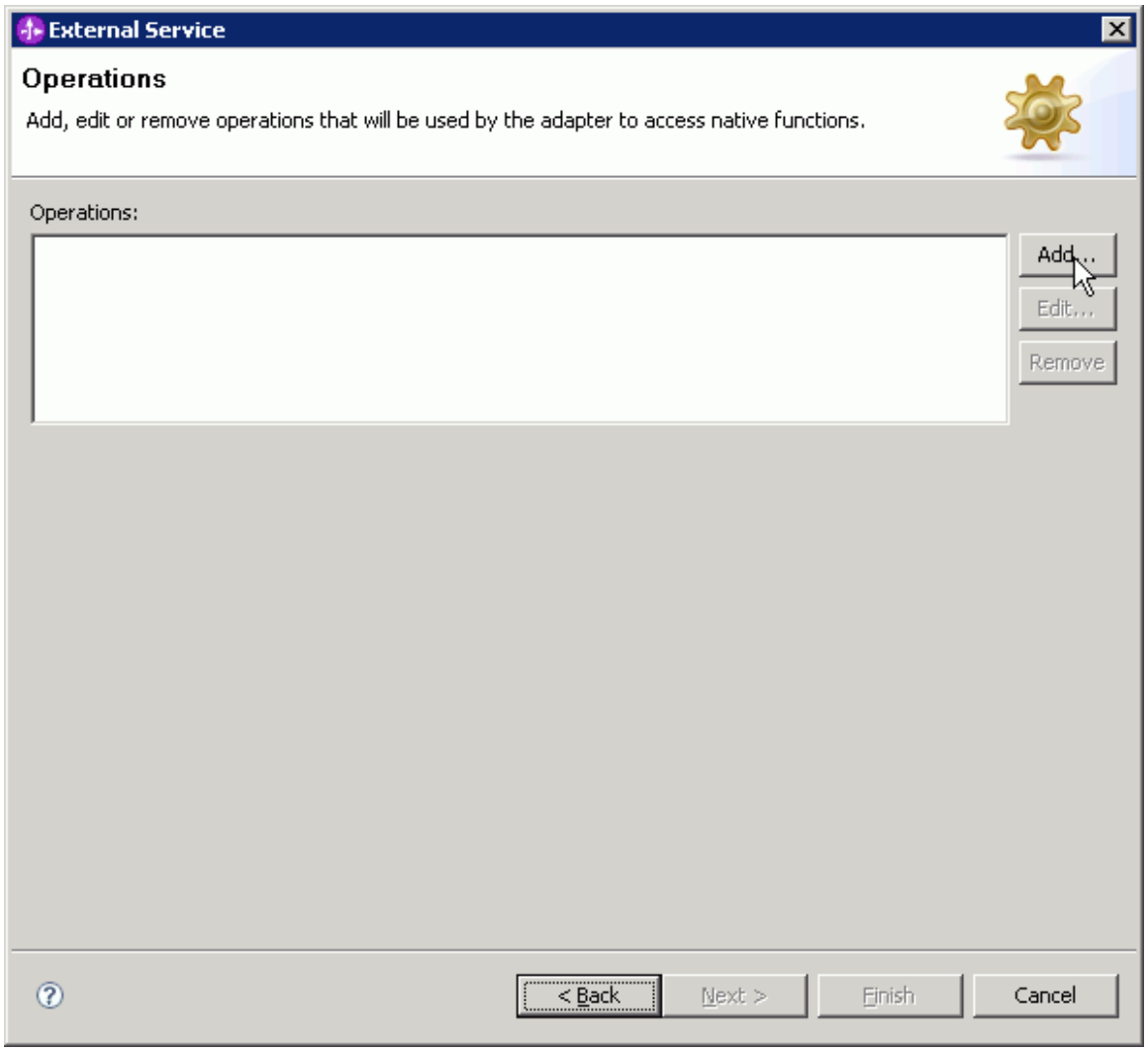
- In order to configure second rule, click **Add** , in the Add/Edit properties window, select **END-OF-RULE** from the **Property Type** list, which will populate both **Operator** as well as **Value** fields with **END-OF-RULE**. Else if you don't want to configure second rule, do not select **END-OF-RULE.**
- Please NOTE: If user needs to configure only one rule then user should not select the **END-OF-RULE,** or the rule will be invalid at runtime.

- Click **Finish** and click **Next**.

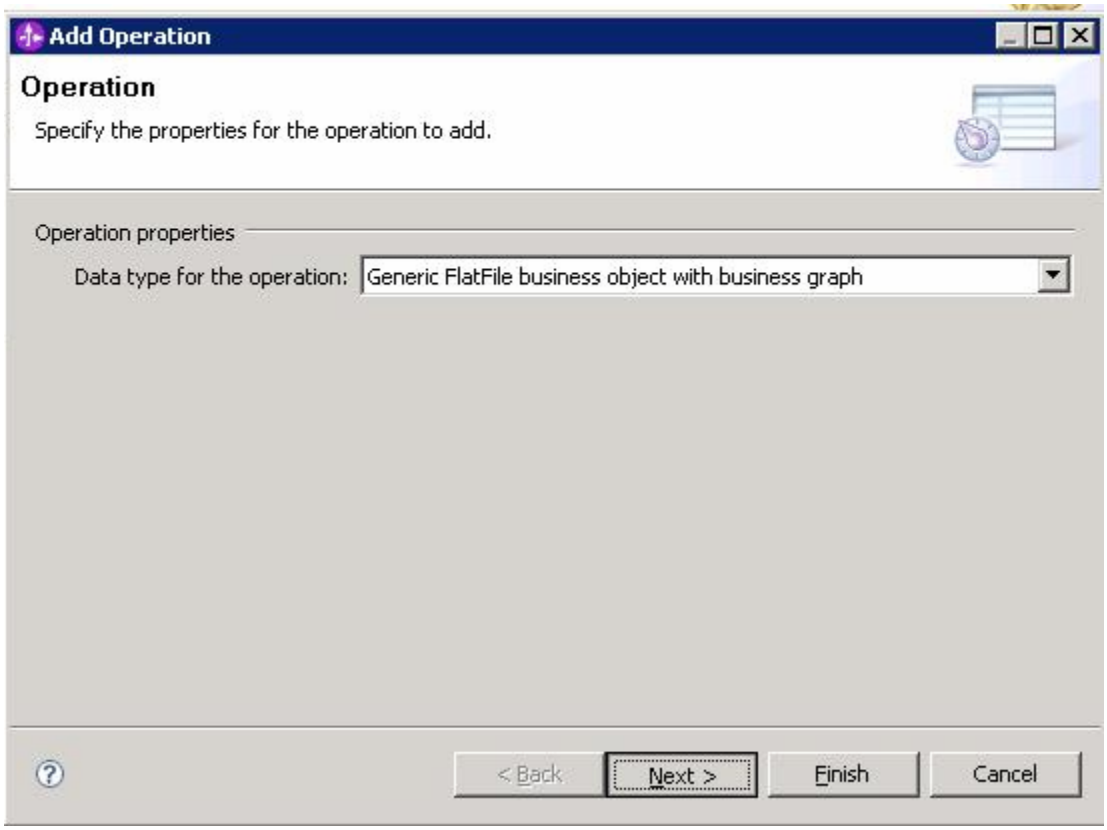


- Since we want to configure only one rule, we will not select END-OF-RULE this time.

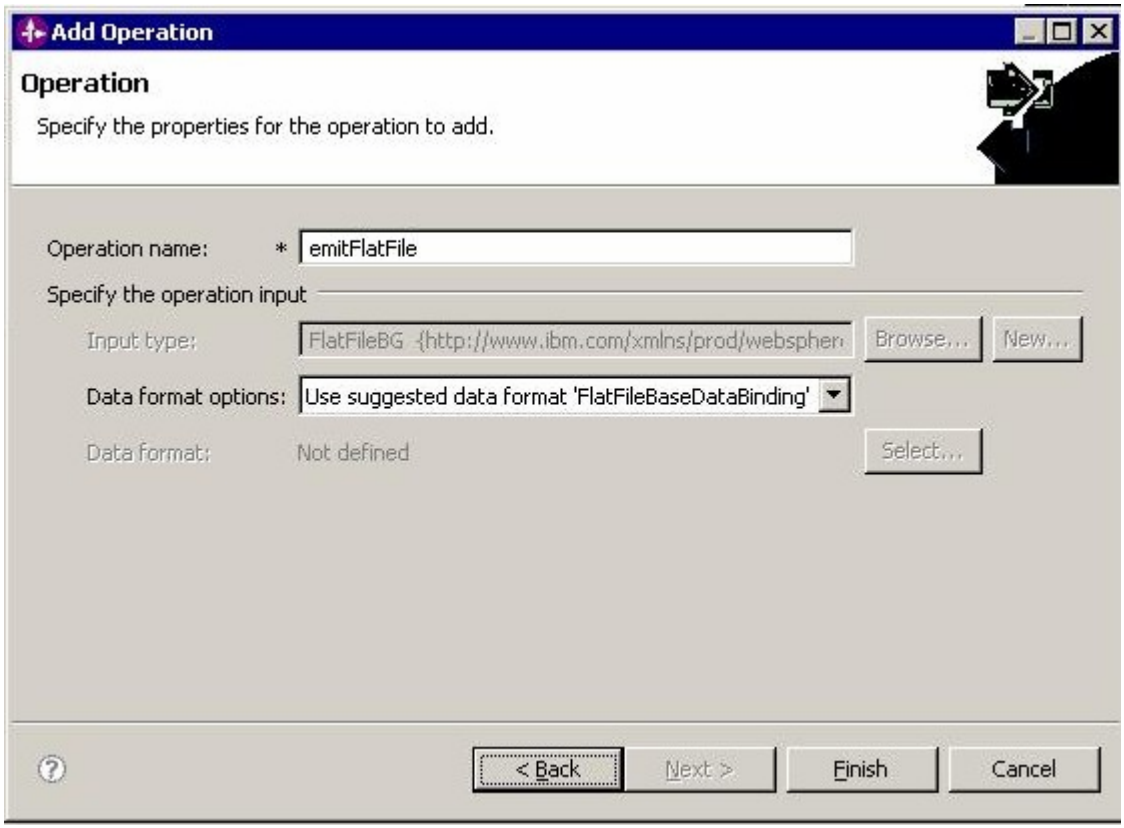
- In the Operations window, click **Add** to add the operations that you want to perform.



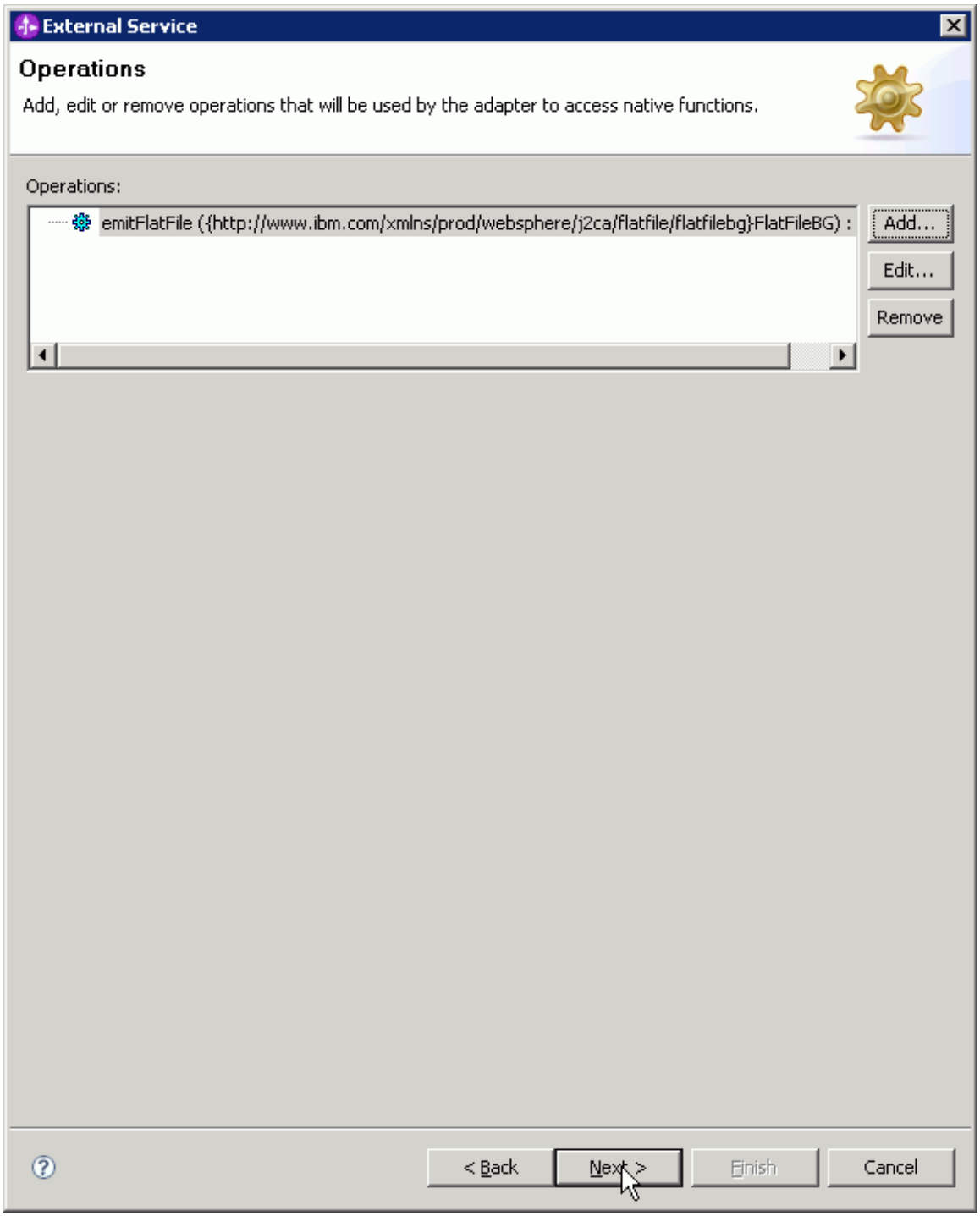
- Because this tutorial is using a pass through scenario without any data transformation, select the **Data type for the operation** as **Generic Flat file Business Object with Business Graph** in the drop down menu. Click **Next**.



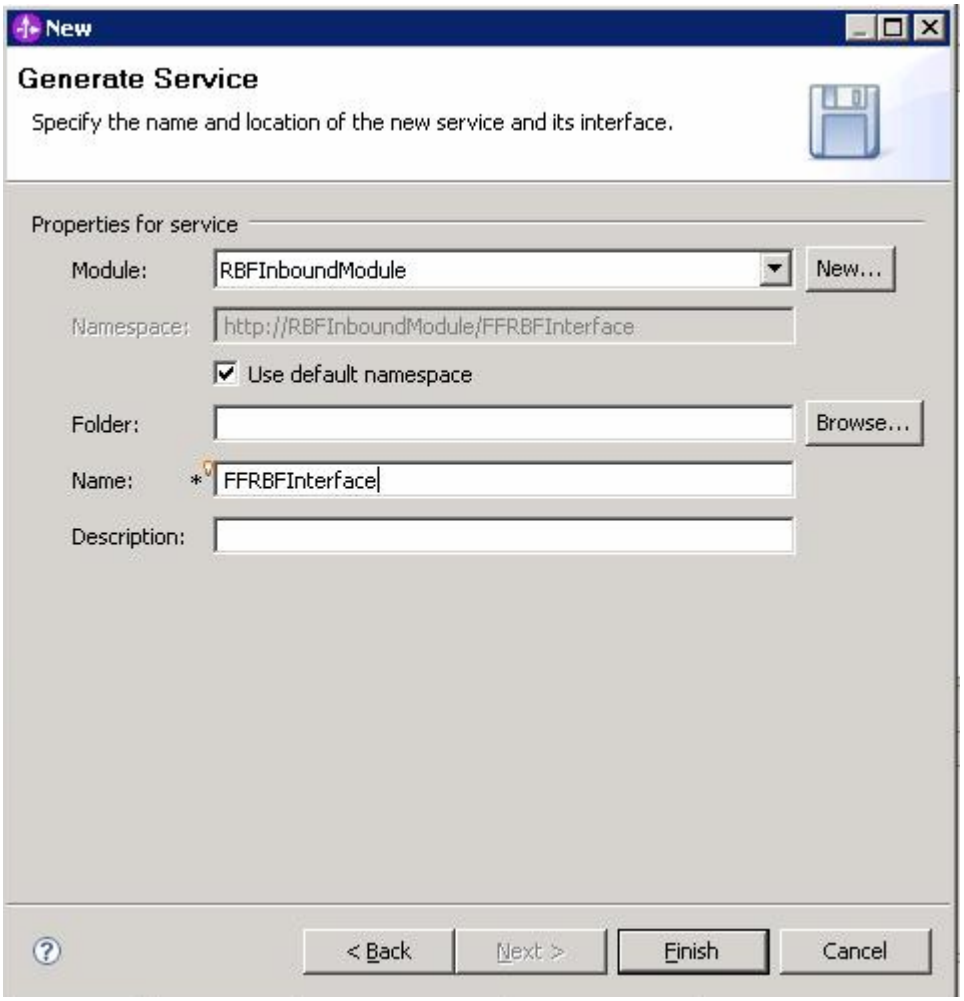
- In the next window, click **Finish**.



- Click **Next** in the window that follows.



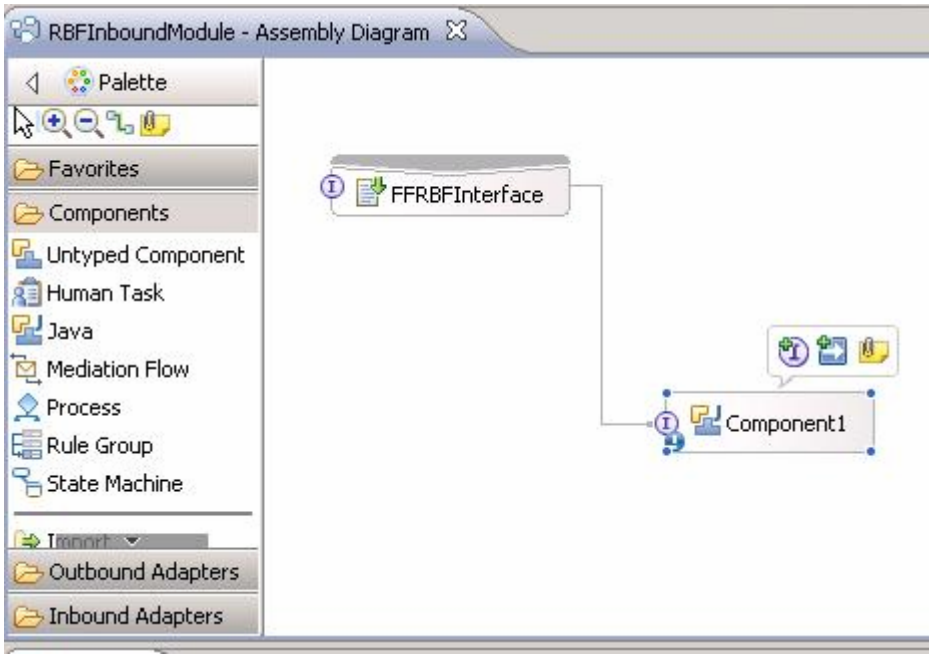
- In the Generate Service window, either specify a name for the interface of the module or use the default name FlatFileExport. For this tutorial, the name RBFInboundModule is used.



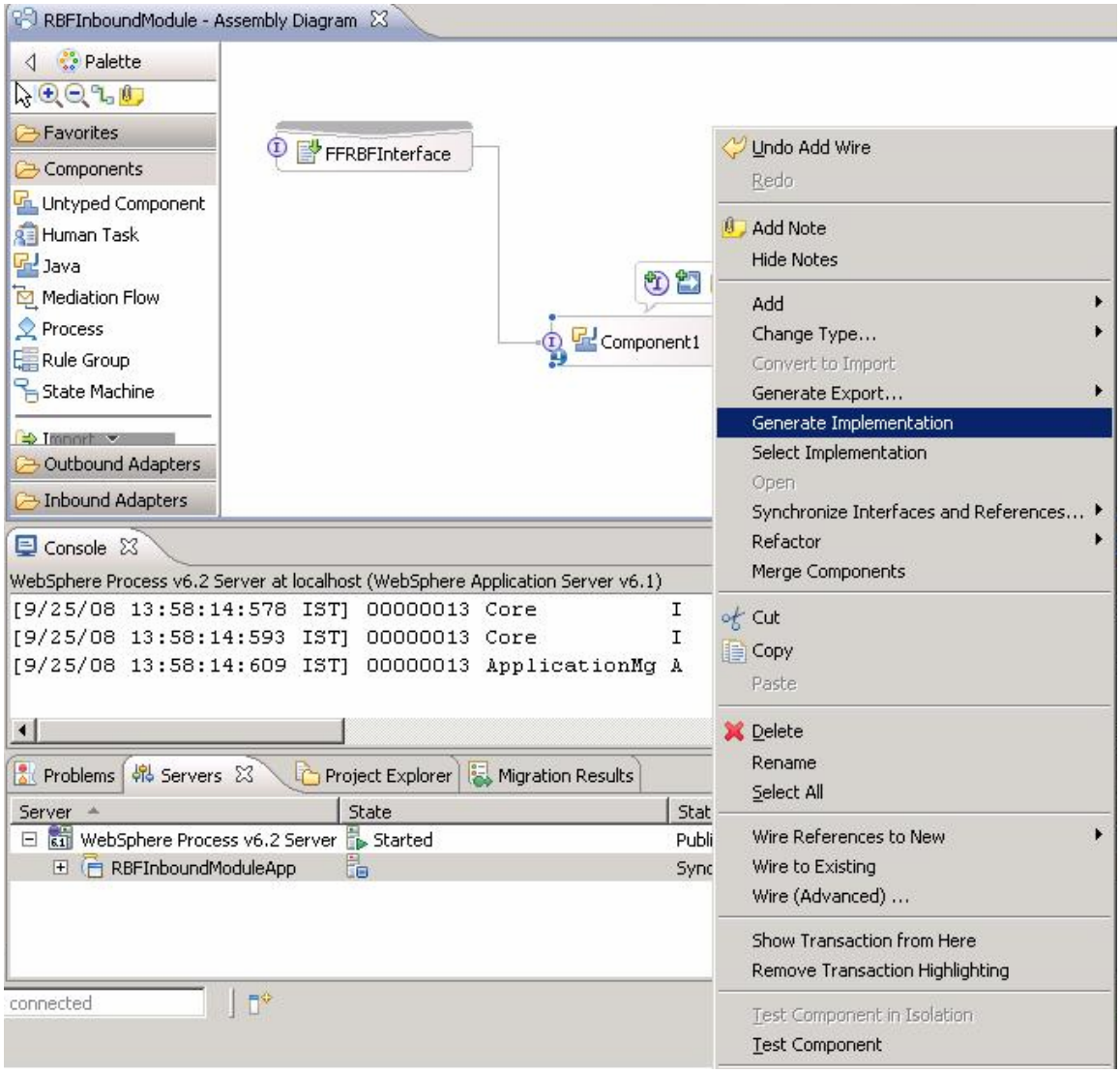
- This is last step of running the external service wizard; specify a suitable name for your adapter interface in the Name field. Click **Finish.**

Generating business object definitions and related artifacts

- Generate a component of Java™ type and draw a wire from **FFRBFInterface** to **Component1.**



- Implement the java component by right clicking Component1 and then clicking on **Generate Implementation**. You can choose to have the implementation belong to default package or any other package.



- A window (shown in the following screen capture) containing a code snippet of the generated implementation of Component1 will be displayed. Customize the implementation of the method emitFlatFile as shown in the screen capture below to test the module.

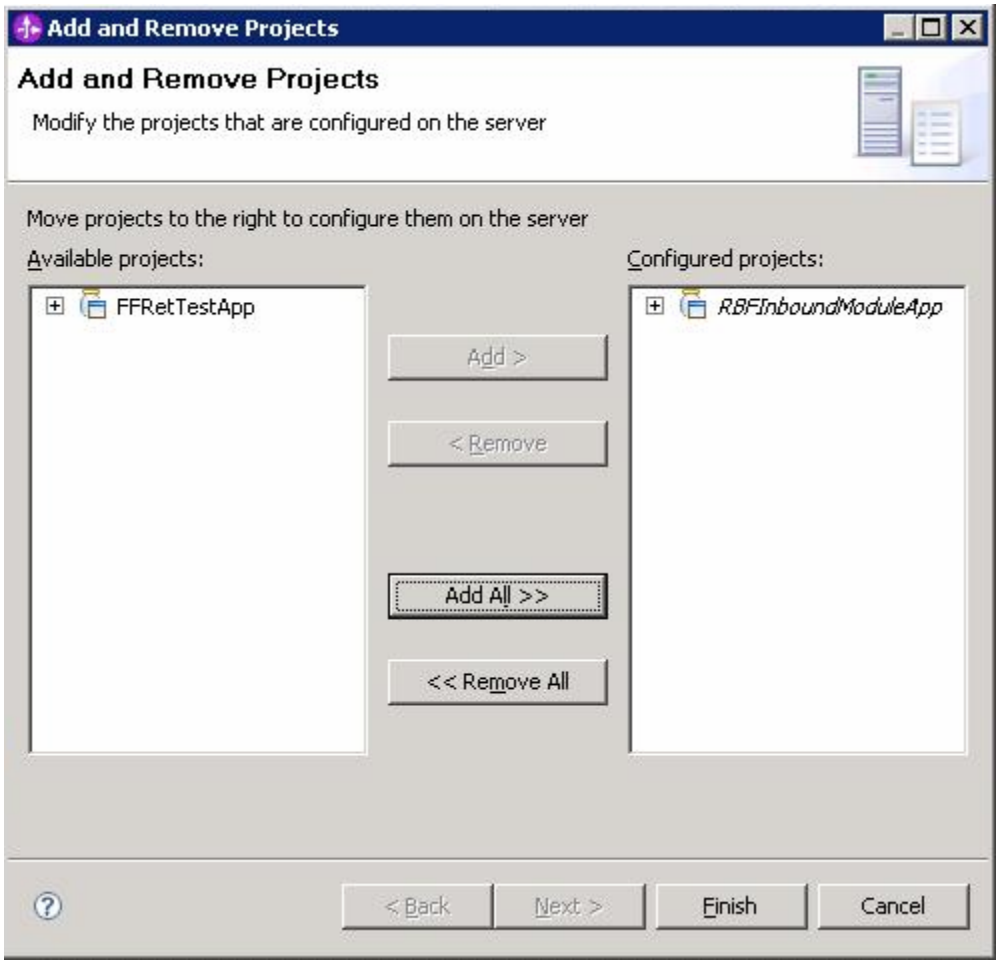
```
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a para
     * type conveys that its a complex type. Please refer to the WSDL Definition
     * on the type of input, output and fault(s).
     */
    public void emitFlatFile(DataObject emitFlatFileInput) {

        DataObject exampleDO = emitFlatFileInput.getDataObject("FlatFile");

        String chunkInfo = exampleDO.getString("chunkFileName");
        System.out.println("ENDPOINT ::: " + chunkInfo);

        DataObject unstructuredContent = exampleDO.getDataObject("content");
        System.out.println("ENDPOINT ::: After getting unstructured content");

        /*byte[] content = unstructuredContent.getBytes("AsBinary");
        if(content != null)
            System.out.println("ENDPOINT content::: "+content.length);
        else
            System.out.println("CONTENT IS NULL");*/

        String content = unstructuredContent.getString("AsText");
        if((!content.equalsIgnoreCase("")) || (content != null))
```

-

Deploying the module to the test environment

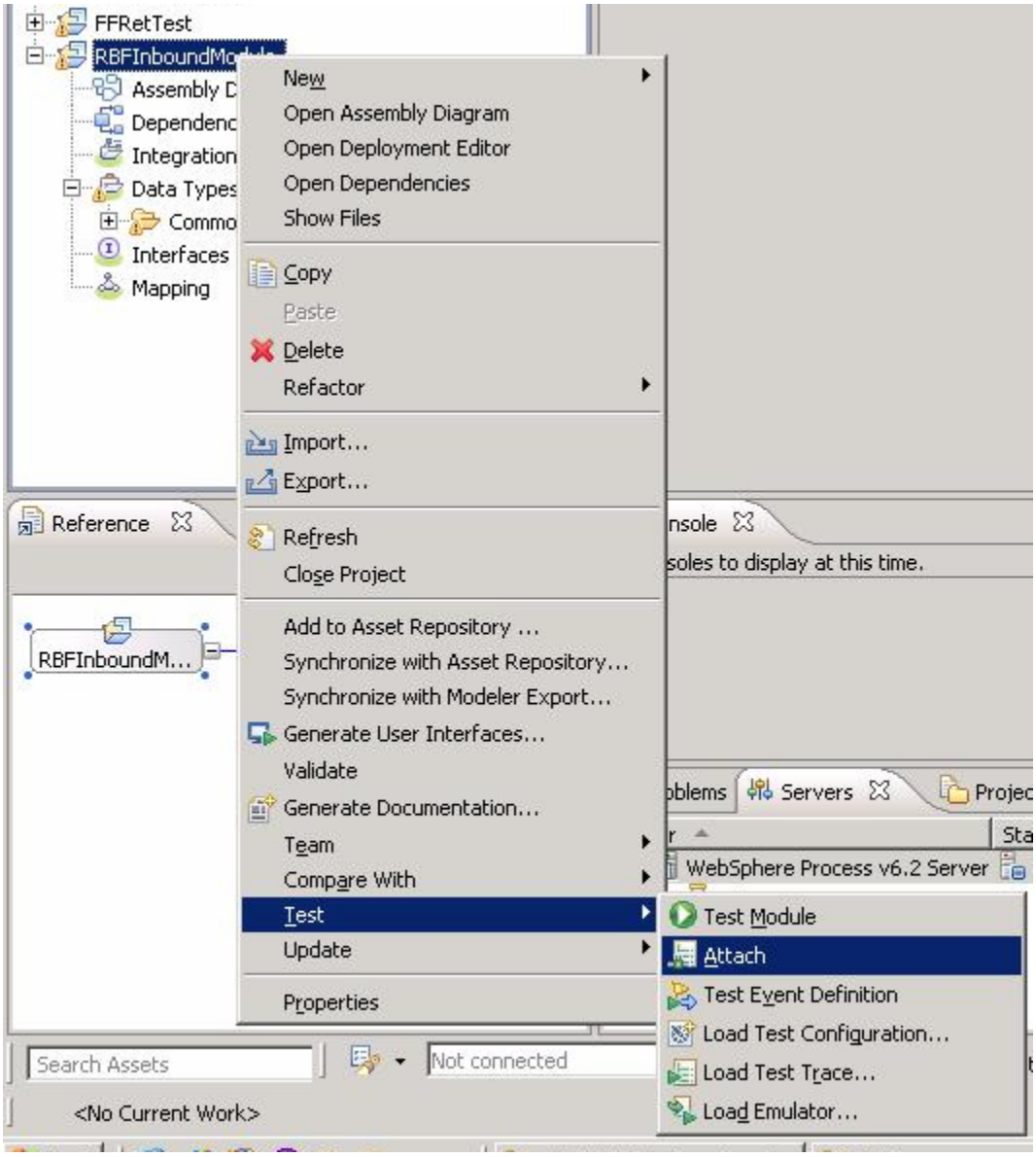
The following steps show how to perform a Retrieve operation with the module you just created.

- Start WebSphere Process Server.
- Add the adapter module to the server. In the **Server** tab, right click on **WebSphere Process Server** then select **Add and Remove Projects.**
- From **Available projects** pane, select your adapter module, click **Add >** and click **Finish**.



Testing the assembled adapter application

- Right click the adapter module, **RBFInboundModule** then select **Test** -> **Attach.**
- Copy a sample event file to the specified Event directory
- The relevant business object will be delivered to the end-point
- Verify that the business object has been delivered by either checking for the end-point messages in the System.Out file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer


FFRetTest
RBFInboundMo
Assembly D
Dependenc
Integration
Data Types
Commo
Interfaces
Mapping
New
Open Assembly Diagram
Open Deployment Editor
Open Dependencies
Show Files
Copy
Paste
Delete
Refactor
Import...
Export...
Refresh
Close Project
Add to Asset Repository ...
Synchronize with Asset Repository...
Synchronize with Modeler Export...
Generate User Interfaces...
Validate
Generate Documentation...
Team
Compare With
Test
Update
Properties
Reference
RBFInboundM...
nsole
soles to display at this time.
blems
Servers
Projec
WebSphere Process v6.2 Server
Test Module
Attach
Test Event Definition
Load Test Configuration...
Load Test Trace...
Load Emulator...
Search Assets
Not connected
<No Current Work>

Chapter 10. Troubleshooting

- **Enabling traces:** Open the server admin console. Go to Troubleshooting -> Logging and tracing -> server1 -> change log Detail Levels. Under the components -> groups include *=info: **com.ibm.j2ca.*=finest**
- **Transaction related error:** While working with the Adapter for FlatFiles on WebSphere Process Server, it arbitrarily generates an exception related to "Transaction Rollback Recovery."
    - Stop the instance of WebSphere Process Server
    - Navigate to the folder where transaction logs are written, <WID_installation_directory>\runtimes\bi_v61\profiles\ProcSrv01\tranlog in WebSphere Process Server v 6.2
    - Delete the entire of the contents of this folder
    - Restart WebSphere Process Server again

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in

writing, to:

IBM World Trade Asia Corporation Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR

IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or

implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.

IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.

The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department

2Z4A/SOM1 294 Route 100

Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include

the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided

to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks