

Oracle® Java CAPS File Binding Component Tutorial

Copyright © 2009, 2011, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group in the United States and other countries.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Using the File Binding Component	5
Tutorial Requirements	5
Prerequisites	5
Starting the GlassFish Server	6
▼ To Check the Status of the GlassFish V2 Application Server in the NetBeans IDE	6
▼ To Register the GlassFish V2 Application Server with the NetBeans IDE	6
▼ To Start the GlassFish V2 Application Server in the NetBeans IDE	7
Creating a BPEL Module Project	7
▼ To Create a BPEL Module Project	7
▼ To Create the WSDL Document for Type Poll	10
▼ To Create the WSDL Document for Type Write	12
▼ To Create a BPEL Process	14
▼ To Add Partner Links to the BPEL Process	16
▼ To Add a Receive Activity to the BPEL Process	18
▼ To Add an Invoke Activity to the BPEL Process	21
▼ To Add an Assign Activity to the BPEL Process	24
▼ To Clean and Build the BPEL Module Project	25
Creating a Composite Application Project	26
▼ To Create a Composite Application Project	26
Building and Deploying the Composite Application Project	28
▼ To Build and Deploy the Composite Application	28
Testing the Composite Application	30
▼ To Test the PollWriteCompApp Composite Application	30
Summary	30

Using the File Binding Component

The File BC provides a comprehensive solution as a transport to interact with the file system within a JBI environment. This tutorial is designed to show how the File Binding Component polls for a file in a specified directory and writes it into the designated directory.

What You Need to Know

These topics provide information you need to know before you start the tutorial:

- [“Tutorial Requirements” on page 5](#)
- [“Prerequisites” on page 5](#)

What You Need to Do

This tutorial includes instructions on how to perform these tasks:

- [“Starting the GlassFish Server” on page 6](#)
- [“Creating a BPEL Module Project” on page 7](#)
- [“Creating a Composite Application Project” on page 26](#)
- [“Building and Deploying the Composite Application Project” on page 28](#)
- [“Testing the Composite Application” on page 30](#)

Tutorial Requirements

Before you proceed, make sure you review the requirements in this section.

Prerequisites

This tutorial assumes that you have some basic programming experience with the Java language and platform, and knowledge of the NetBeans IDE.

Starting the GlassFish Server

Before you can deploy your composite application, the application server and the appropriate JBI components must be configured correctly and must be running. Before deploying and performing test runs of a Composite Application project in the NetBeans IDE, make sure that the GlassFish Application Server is started. You might need to add the server to the NetBeans IDE.

▼ To Check the Status of the GlassFish V2 Application Server in the NetBeans IDE

- 1 If the Services window is not visible, on the NetBeans IDE menubar choose **Window** → **Services**.

- 2 In the Services window, expand the Servers node.

The Servers node should contain a GlassFish V2 subnode. If the GlassFish V2 node is not visible, see [“To Register the GlassFish V2 Application Server with the NetBeans IDE” on page 6](#).

If a green arrow icon appears on the GlassFish V2 node, the server is running. If you do not see a green arrow icon, see [“To Start the GlassFish V2 Application Server in the NetBeans IDE” on page 7](#).

▼ To Register the GlassFish V2 Application Server with the NetBeans IDE

- 1 If the Services window is not visible, on the NetBeans IDE menubar choose **Window** → **Services**.

- 2 In the Services window, right-click the Servers node and choose **Add Server** from the pop-up menu.

The Add Server Instance dialog box opens.

- 3 In the Choose Server page of the dialog box, select **GlassFish V2** from the Server drop-down list.

- 4 If you want to change the server name that the IDE uses to identify the server, type it in the **Name** field.

- 5 Click **Next**.

The Platform Location Folder page opens.

- 6 In the Platform Location field, click **Browse** and select the installation location of the application server.
- 7 Select the **Register Local Default Domain** option and click **Next**.
- 8 Type the user name and password for the domain's administrator.
If you accepted the default values during the installation, the user name is `admin`; the password is defined during installation.
- 9 Click **Finish**.

▼ To Start the GlassFish V2 Application Server in the NetBeans IDE

- 1 On the NetBeans IDE page, in the Services window, right-click the **GlassFish V2** node and choose **Start**.
- 2 Wait until the following message appears in the Output window:
`"Application server startup complete."`
When the server is running, the IDE displays a green arrow icon on the GlassFish V2 node.

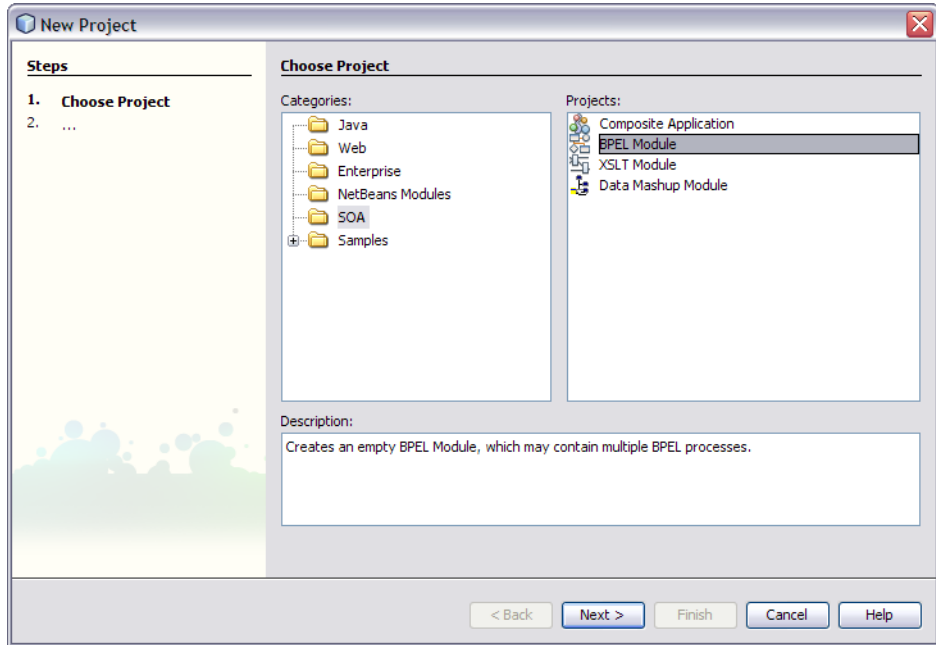
Creating a BPEL Module Project

This section provides step-by-step directions to create a simple BPEL module project called `PollInOut`, and also provides instructions to create a WSDL document, BPEL process, adding partner links and to clean and build the BPEL module project.

▼ To Create a BPEL Module Project

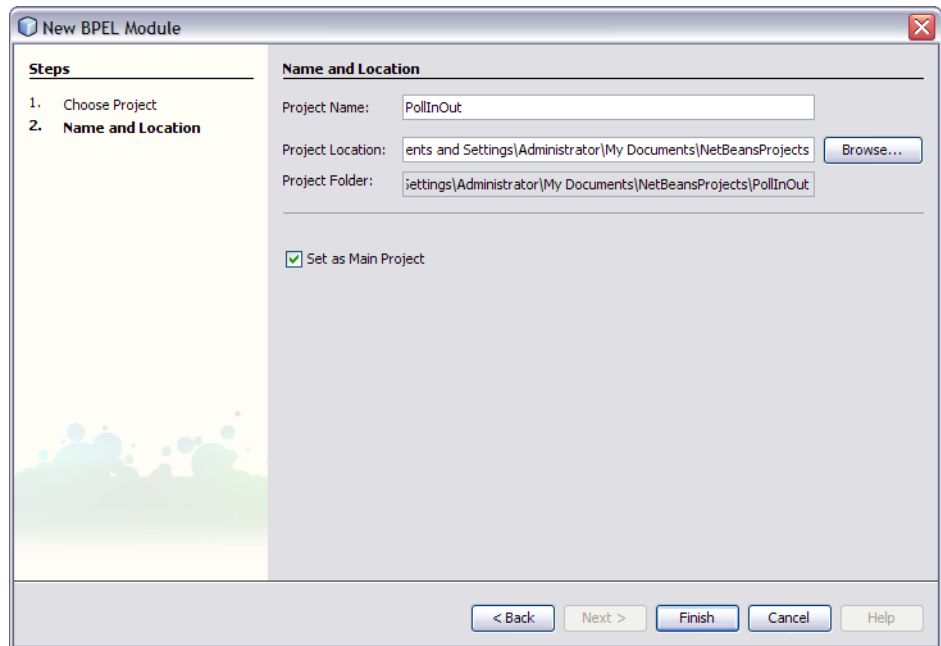
- 1 From the NetBeans IDE's main menu, choose **File** → **New Project**.
The New Project wizard opens.

-
- 2 In the Categories list, select the SOA node and in the Projects list select BPEL Module and click Next.**



-
-
- 3 In the Project Name field, type PoLLInOut.**
You can choose to change the Project Location or just use the default location.

- 4 Leave the **Set as Main Project** option selected and click **Finish**.

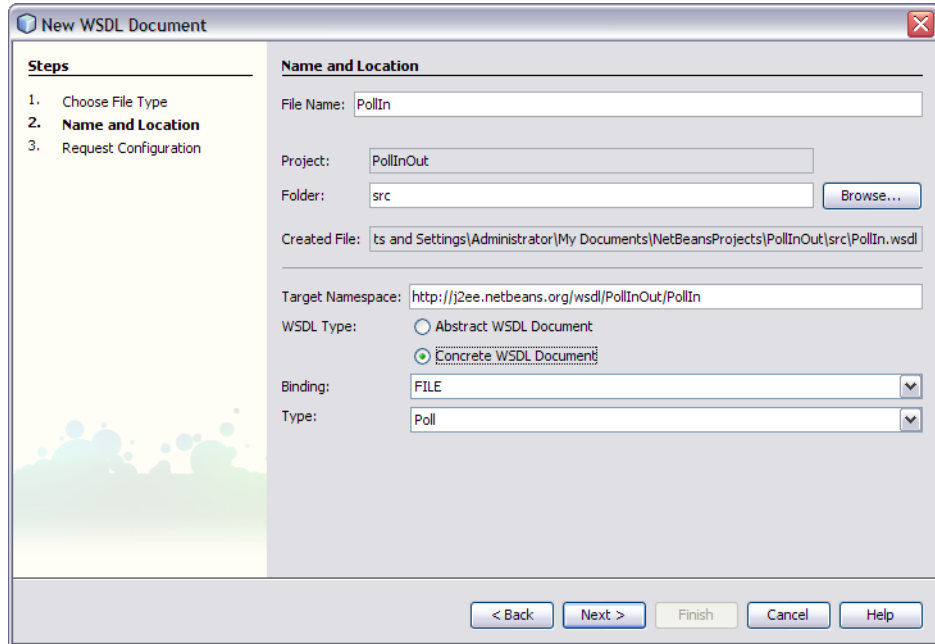


The Projects window now contains a node for the BPEL Module project called `PollInOut`.

▼ To Create the WSDL Document for Type Poll

- 1 In the Projects window, expand the PollInOut Module, right-click the Process Files node, and choose New → WSDL Document from the pop-up menu.

The New WSDL Document dialog box appears.



- 2 In the File Name field type PollIn.
- 3 In the WSDL Type, select the Concrete WSDL Document option.

4 In the Binding field, select FILE and in the Type field, select PoLL and click Next.

The Request Configuration page appears.

Steps

1. Choose File Type
2. Name and Location
3. **Request Configuration**

Request Configuration

File Polling

File Name* (pattern): input.txt

Polling Directory*: c:/temp Browse...

Polling Directory Relative To: <Not Set>

Polling Interval: 1000 (ms)

Enable Archive Details...

Record Processing

Multiple Record Delimited By: LINE FEED

Maximum (Bytes) Per Record 1

Payload Processing

Message Type: text

XSD Element/Type: ...

encoded type: ...

Forward as Attachment

Remove trailing EOL

File Name

Defines the file name relative to the specified directory to read from or write to.
If fileNameIsPattern is not true, this attribute specifies an actual file name.
Otherwise, this attribute specifies a pattern marker used for filtering

< Back Next > Finish Cancel Help

5 In the File Name* (pattern) field, change input.xml to input.text.

You can specify the file name as .txt or else use .xml, but it must be same as the input file type
Note that the Polling Directory is c:/temp by default. You can change the directory if required.

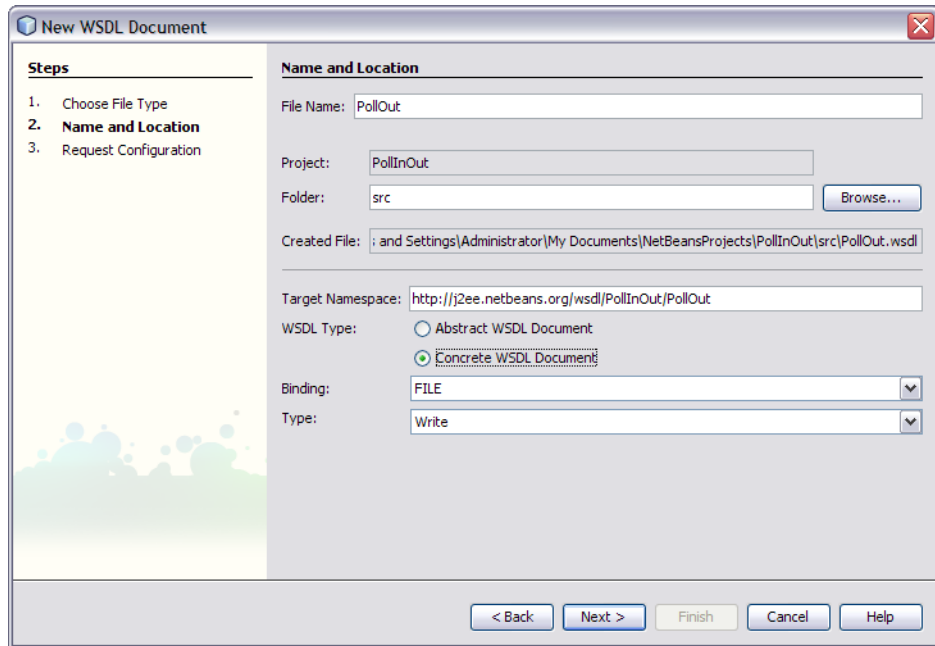
6 Click Finish.

The Projects window now has a node PoLLIn.WSDL under Process Files node.

▼ To Create the WSDL Document for Type Write

- 1 In the Projects window, expand the PollInOut Module, right-click the Process Files node, and choose New → WSDL Document from the pop-up menu.

The New WSDL Document dialog box appears.



- 2 In the File Name field type PollOut.
- 3 In the WSDL Type, select the Concrete WSDL Document option.

4 In the Binding field, select FILE and in the Type field, select Write and click Next.

The Request Configuration page appears.

New WSDL Document

Steps

1. Choose File Type
2. Name and Location
3. **Request Configuration**

Request Configuration

File Write

File Name* (pattern):

Path*:

Is Relative To:

File Exists:

Rename Existing File

Overwrite Existing File

Append to Existing File, Delimited By:

Payload Processing

Message Type: Add trailing EOL

XSD Element/Type: ...

encoded type:

File Name

Defines the file name relative to the specified directory to read from or write to. If fileNameIsPattern is not true, this attribute specifies an actual file name. Otherwise, this attribute specifies a pattern marker used for filtering

< Back Next > Finish Cancel Help

5 In the File Name* (pattern) field, change output.xml to output.txt.

You can specify the file name as .txt or else use .xml, but it must be the same as the output file type.

Note that the Path is c:/temp by default. You can change it if required.

6 In File Exists, select the Overwrite Existing File option.

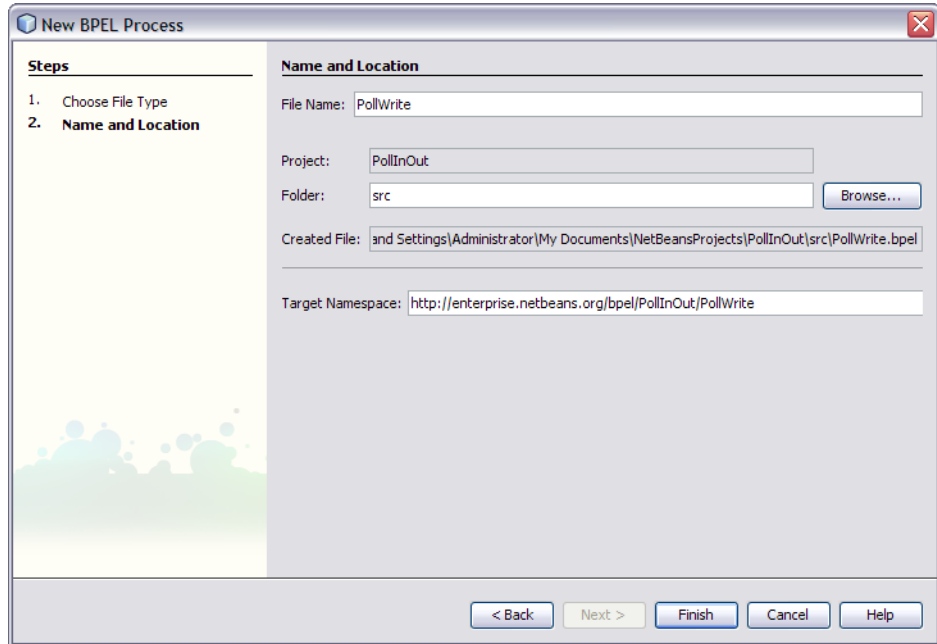
7 Click Finish.

The Projects window now has a node `Pol10ut.WSDL` under Process Files node.

▼ To Create a BPEL Process

- 1 In the Projects window, expand the PollInOut Module, right-click the Process Files node, and choose New → BPEL Process from the pop-up menu.

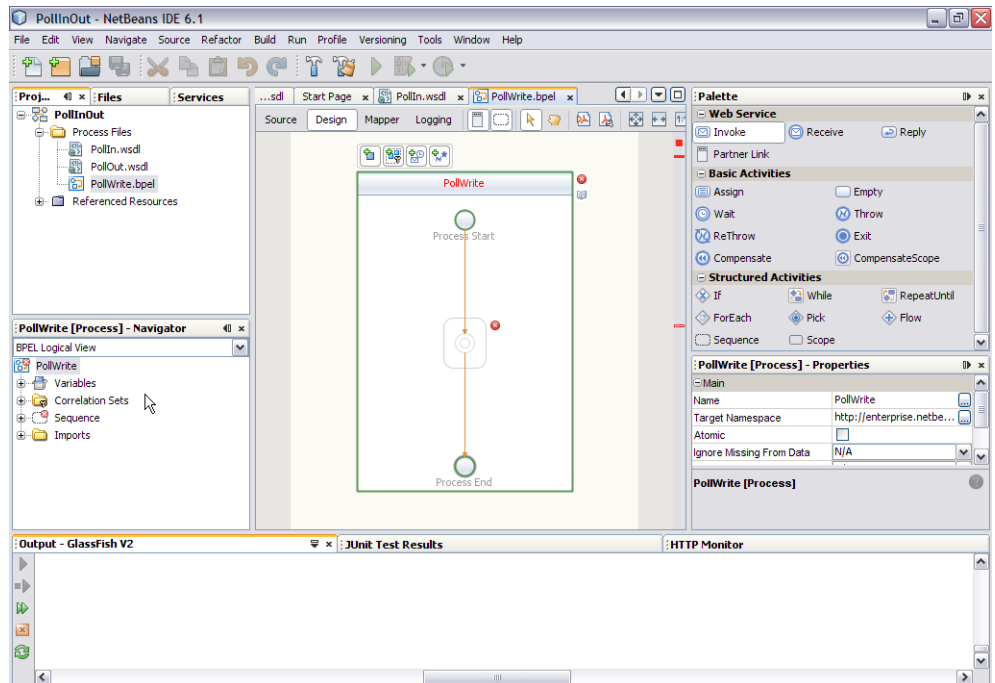
The New BPEL Process dialog box appears.



- 2 In the File Name field type PollWrite and click Finish.

The Projects window now has a node PollWrite . BPEL under Process Files node.

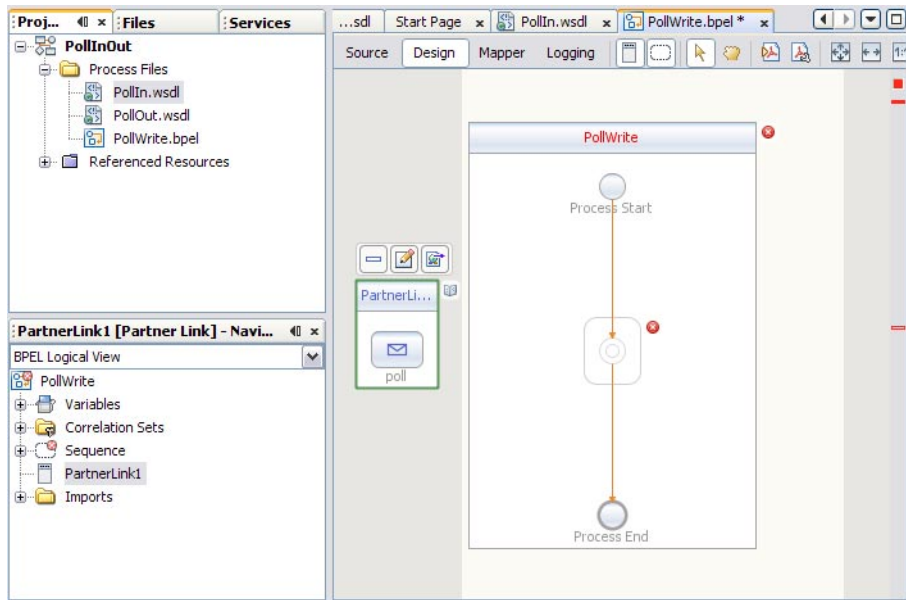
You can now see the new BPEL file, `PollWrite` in the Design view.



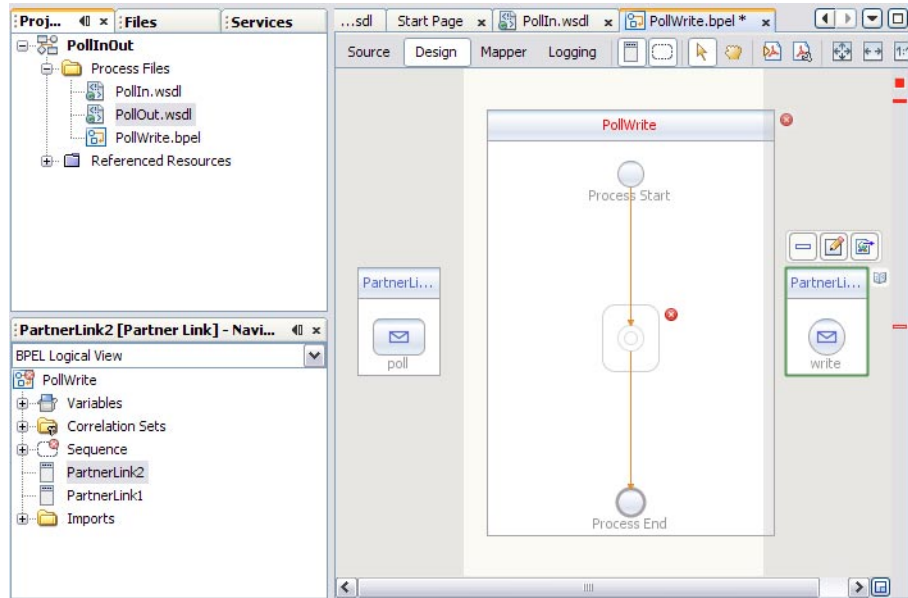
If the Palette and Properties windows are not displayed in your current view, click **Windows** → **Palette** on the NetBeans menu.

▼ To Add Partner Links to the BPEL Process

- 1 In the Projects window, expand PollInOut and Process Files nodes and select the PollIn.wsdl file.
- 2 Drag the PollIn.wsdl file from the Projects window to the left side of the Design view canvas. The IDE provides visual prompts to show you where you can drop the selection. The BPEL Editor adds a partner link to the canvas.



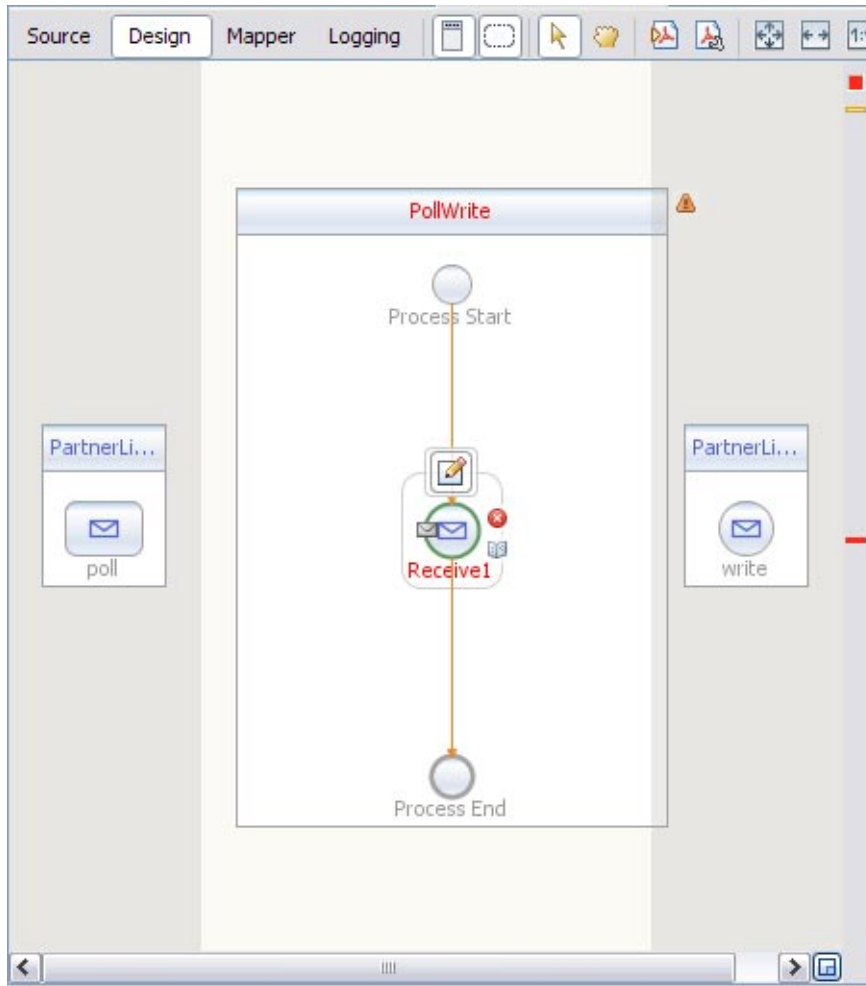
- 3 Now, select the PollOut.wsdl file from the Projects window and drag it to the right side of the Design view canvas.



▼ To Add a Receive Activity to the BPEL Process

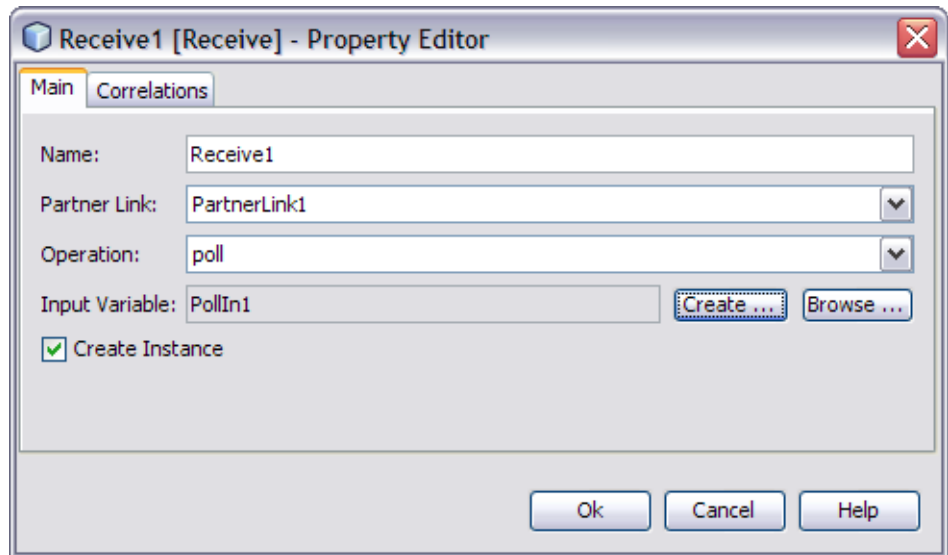
- 1 From the Palette window, in the Web Service section , select the Receive activity and drag it to the PollWrite process box in the Design view canvas, between the Process Start and the Process End activities.

The Receive1 activity is added to the process box.



- 2 Click the Receive1 activity's Edit icon.

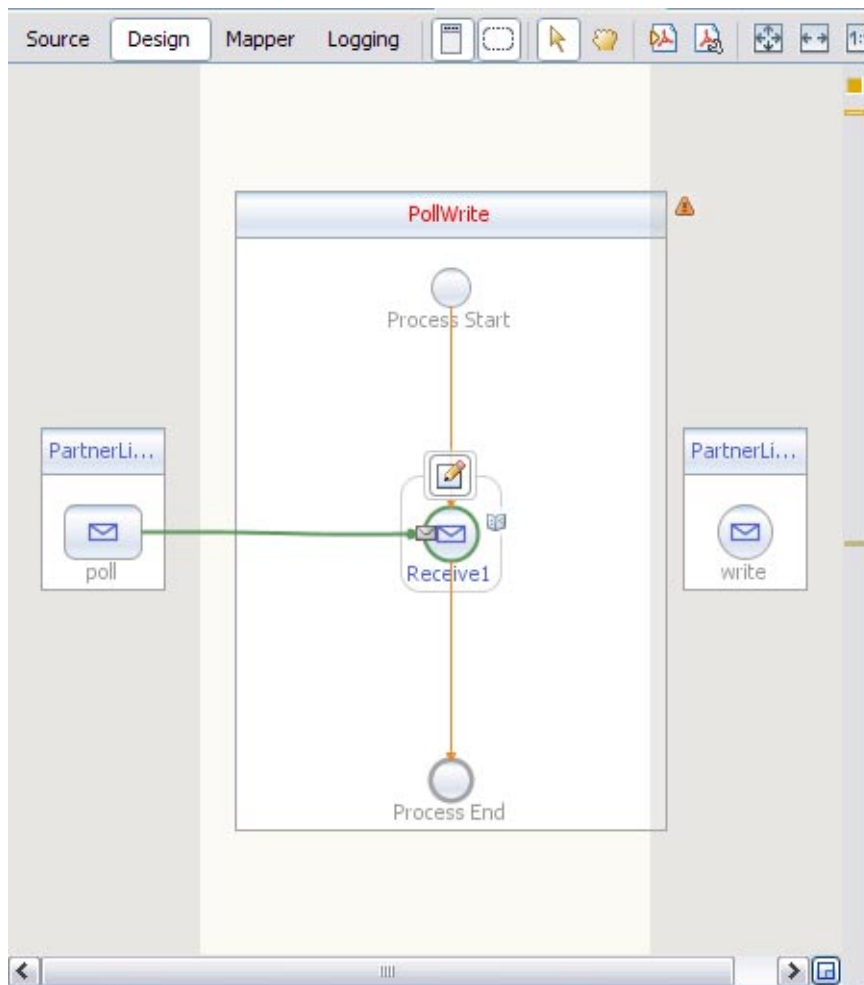
The Receive1 Property Editor appears.



- 3 On the Main tab, accept the value in the Name field. From the Partner Link drop-down list, choose PartnerLink1. The IDE populates the Operation field with PoLL.
- 4 Click the Create button next to the Input Variable Field. The New Input Variable dialog box appears. Click OK to accept the default values.

5 Click OK to close the Receive1 Property Editor.

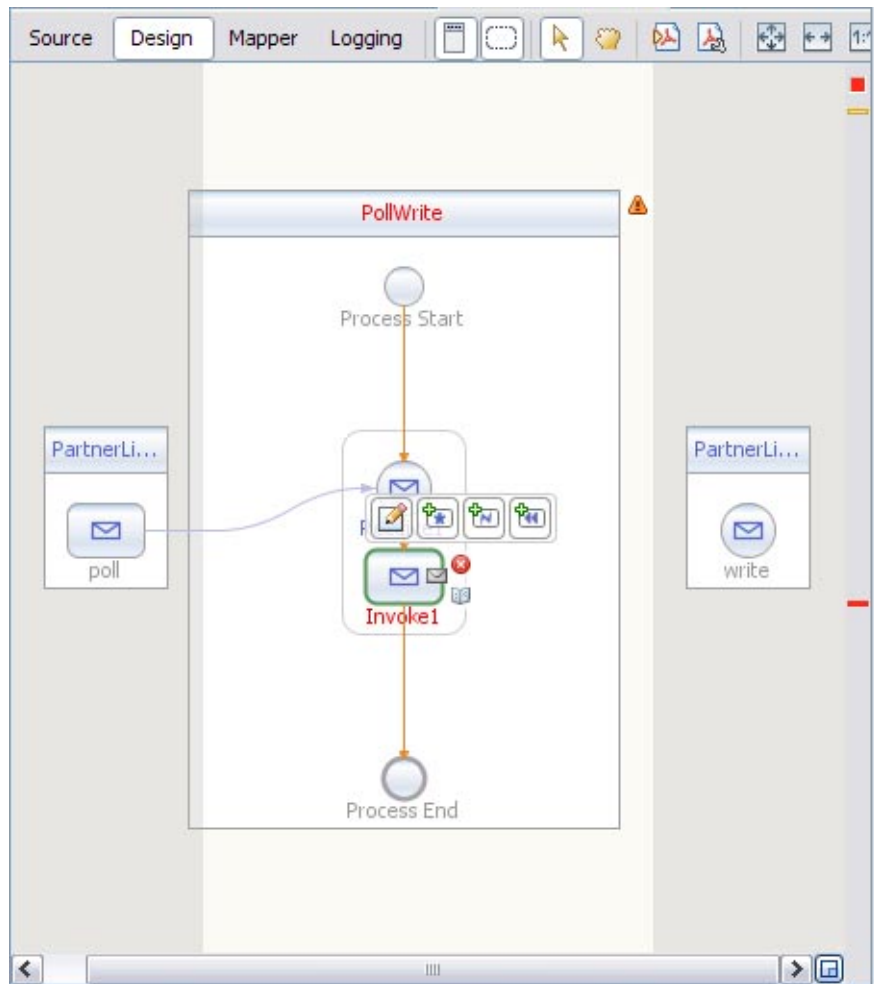
The Design view displays the new connection between PartnerLink1 and the Receive1 in the process box.



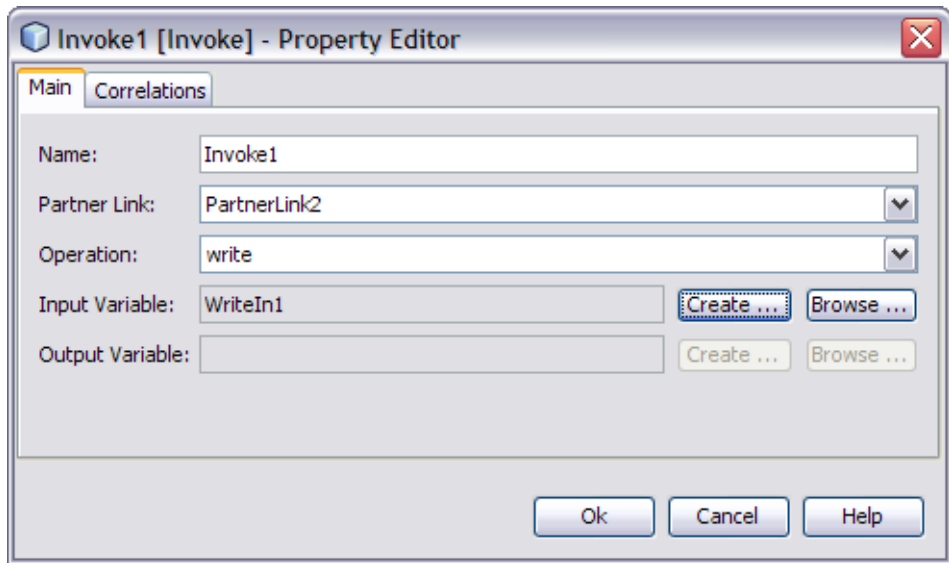
▼ To Add an Invoke Activity to the BPEL Process

- 1 From the Palette window, in the Web Service section , select the Invoke activity and drag and drop it to the prompt between the Receive1 activity and the Process End activity in the process box on the design view canvas.

A Invoke1 activity is added to the design view canvas.



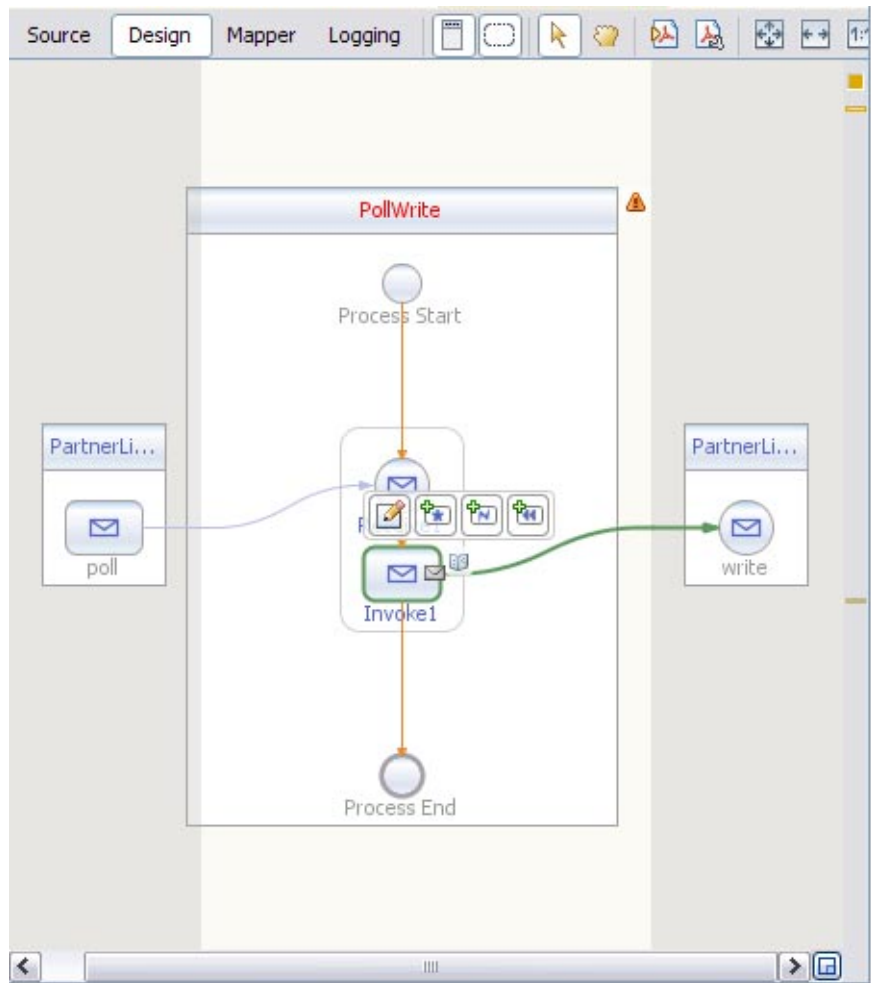
- 2 Click the Invoke1 activity's Edit icon.
The Invoke1 Property Editor appears.



- 3 On the Main tab, accept the value in the Name field. From the Partner Link drop-down list, choose PartnerLink2. The IDE populates the Operation field with write.
- 4 Click the Create button next to the Input Variable Field. The New Input Variable dialog box appears. Click OK to accept the default values.

5 Click **OK** to close the **Invoke1 Property Editor**.

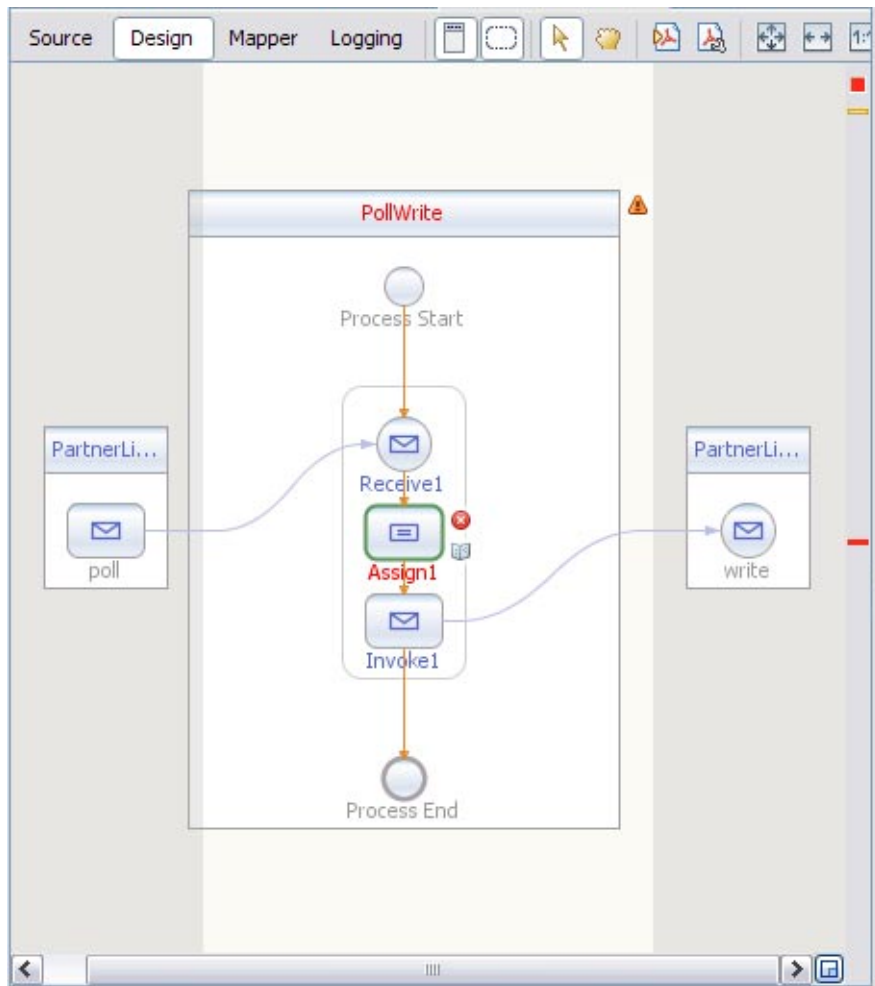
The Design view displays the new connection between **PartnerLink2** and the **Invoke1** in the process box.



▼ To Add an Assign Activity to the BPEL Process

- 1 From the Palette window, in the Basic Activities section, select **Assign** and drag and drop it the prompt between the **Receive1** activity and the **Invoke1** activity in the process box on the design view canvas.

The **Assign1** activity is added to the design view canvas.

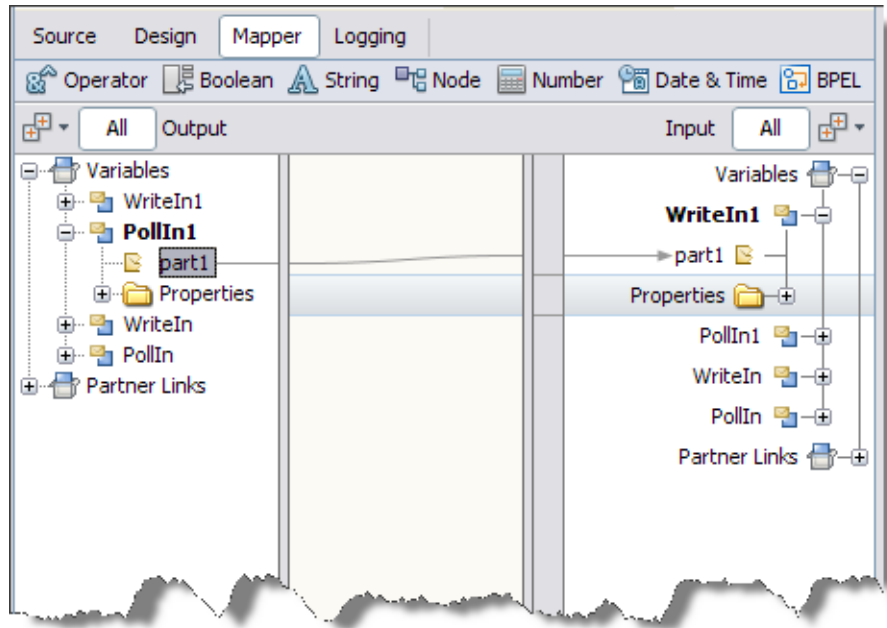


- 2 Select the **Assign1** activity and click the **Mapper** button on the editors toolbar.

The BPEL Mapper appears.

- 3 Map the `part1` node under `Variables` → `PollIn1` in the Output pane of the BPEL Mapper, to the `part1` node under `Variables` → `WriteIn1` in the Input pane of the Mapper.

To do this, select the `part1` node in the Output pane, and drag your cursor to the `part1` node in the Input pane.



This assignment copies the input statement into the output.

- 4 Click the **Save All** icon on the IDE toolbar.

▼ To Clean and Build the BPEL Module Project

- In the **Projects** window right-click the `PollInOut` node and choose **Clean and Build** from the pop-up menu.

When the build is complete the **Output** window reports **BUILD SUCCESSFUL**.

If the **Output** window is not visible, from the NetBeans IDE's menubar choose **Window** → **Output** → **Output**.

Creating a Composite Application Project

A BPEL Module project is not directly deployable. You must first add a BPEL Module project, as a JBI module, to a Composite Application project. You can then deploy the Composite Application project.

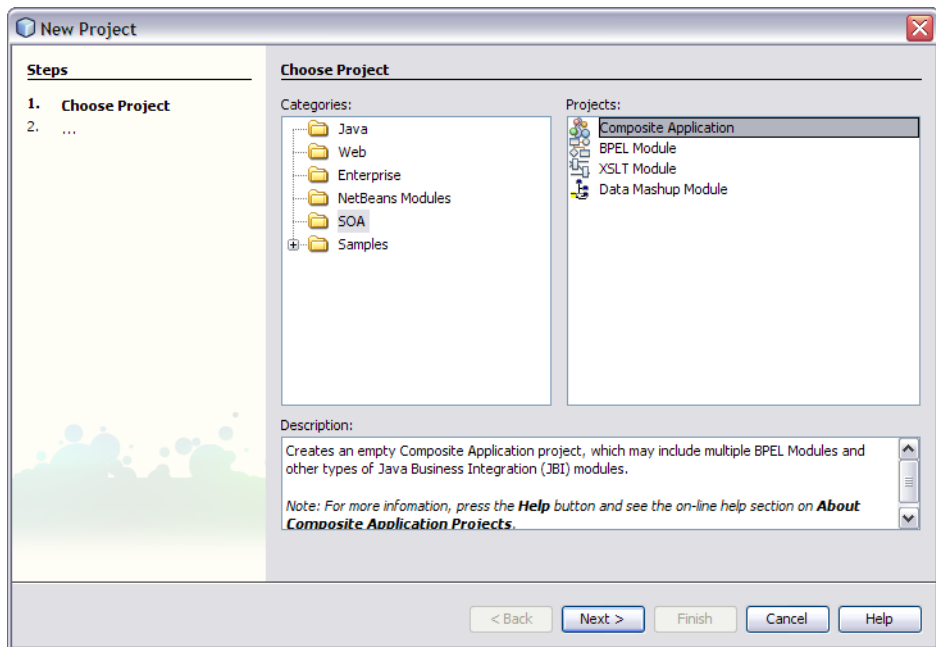
Deploying the project makes the service assembly available to the application server and enables its service units to run.

▼ To Create a Composite Application Project

- 1 From the NetBeans IDE's main menu, choose **File** → **New Project**.

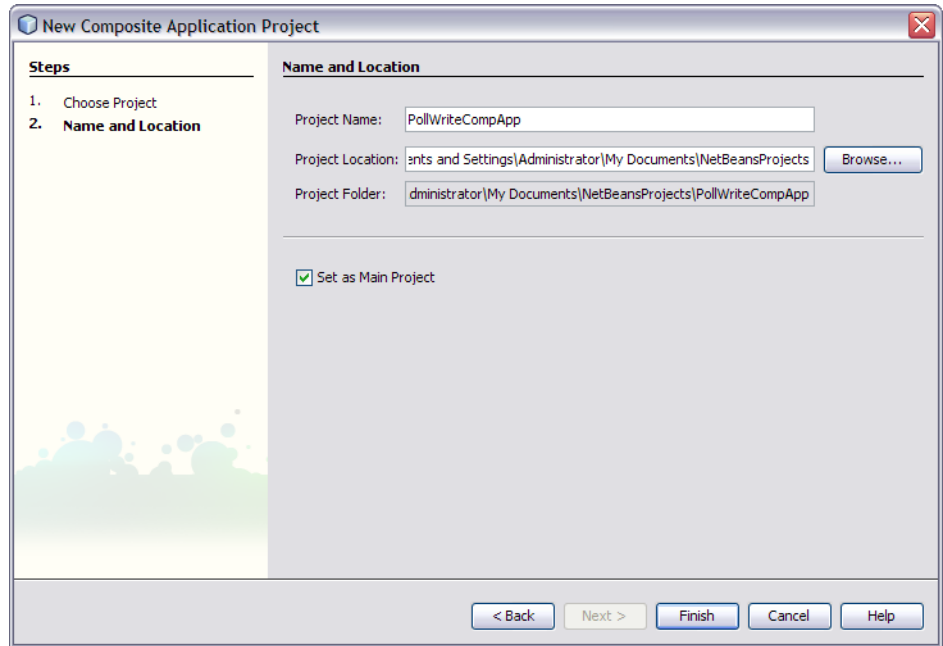
The New Project wizard opens.

- 2 In the **Categories** list, select the **SOA** node and in the **Projects** list select **Composite Application** and click **Next**.



- 3 In the **Name and Location** page, change the project name to `PollWriteCompApp`, and specify the location of project files or just use the default location.

- 4 Leave the **Set as Main Project** option selected and click **Finish**.

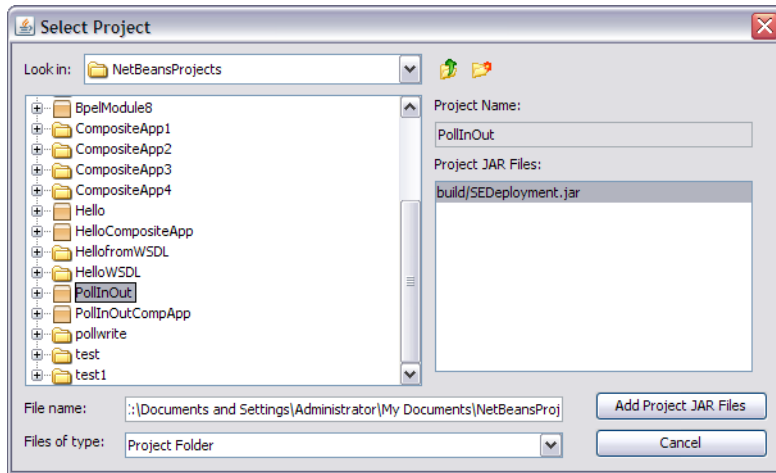


In the Projects window the new Composite Application `PollWriteCompApp` node is visible.

- 5 To add the **BPEL Module** project as a **JBI module** to the Composite Application project, right-click `PollWriteCompApp` and choose **Add JBI Module**.

The Select Project dialog box opens.

- 6 Select the `PollInOut` BPEL project you created earlier. Under **Project JAR Files**, select the `.jar` file and click the **Add Project JAR Files** button.



The **Select Project** dialog box closes and the BPEL module project is added to the **JBI Modules** node of the `PollWriteCompApp` Composite Application.

Building and Deploying the Composite Application Project

Building a project compiles the BPEL source file and packages the BPEL file and web service artifacts, including WSDL.

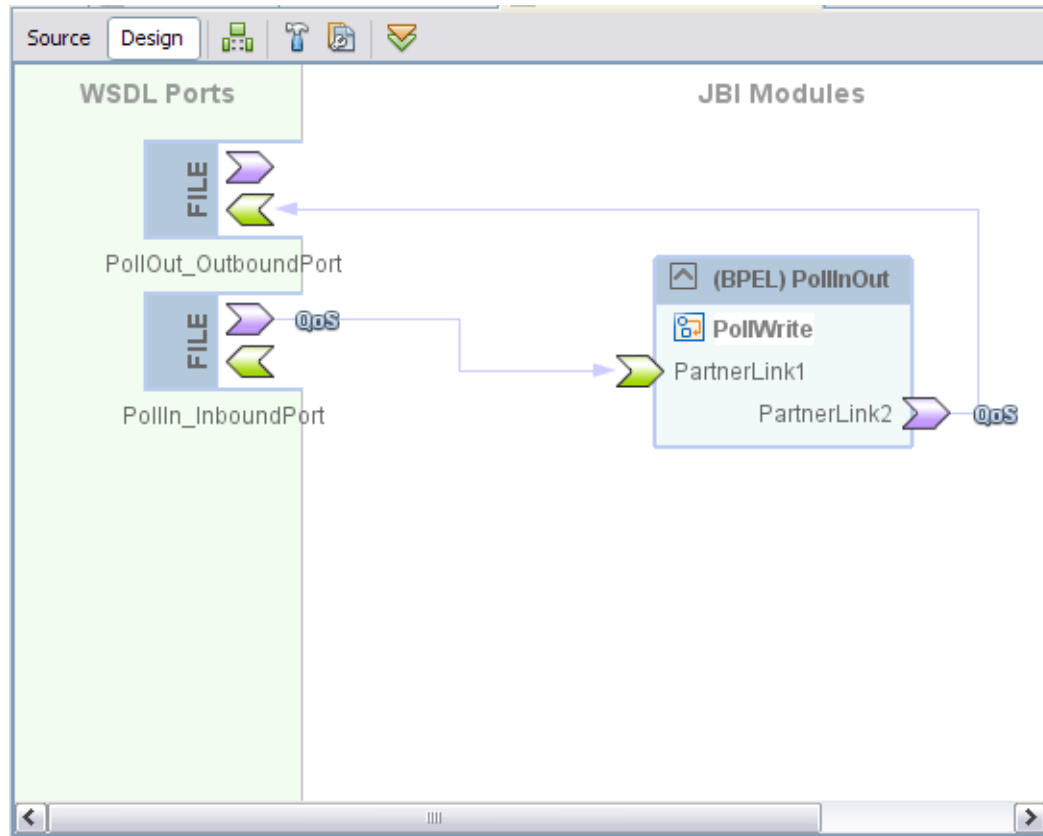
Deploying the project compiles the files in the Composite Application project, packages the compiled BPEL and related web service artifacts (including WSDL) into an archive, and deploys them to the Application Server.

▼ To Build and Deploy the Composite Application

- 1 In the **Projects** window, right-click the `PollWriteCompApp` project's node, and choose **Build** from the pop-up menu.

When the build is complete the **Output** window reports **BUILD SUCCESSFUL**.

- 2 **Right-click the `PollWriteCompApp` node, and choose `Clean and Build` from the pop-up menu.**
The design view displays the build of the project.



- 3 **Right-click the `PollWriteCompApp` node, and choose `Deploy` from the pop-up menu.**
Deployment has succeeded when you see a `BUILD SUCCESSFUL` message in the GlassFish tab of the Output window.
- 4 **Open the Services window and expand `Servers` → `GlassFish V2` → `JBI` → `Service Assemblies` to see your new deployed Service Assembly.**
If you do not see the deployed project, right-click the Service Assemblies node and choose Refresh.

Testing the Composite Application

You can test your Composite Application project by navigating to the `c:\temp` folder and creating a `input.txt` file.

▼ To Test the PollWriteCompApp Composite Application

1 Navigate to the `c:\temp` folder of your system.

Note that `c:\temp` was the designated polling directory for both `Poll` and `Write` operations when WSDL documents were created.

2 Create a `.txt` file and add some content to the file and rename it as `input.txt`.

3 Instantly the `input.txt` file is replaced by `output.txt` file.

The `output.txt` file retains the content of the `input.txt` file. That is, the content of `input.txt` is written to `output.txt`.

Summary

In this tutorial, you learned how to use the File Binding Component to implement a simple `Poll` and `Write` operation. You created and tested a composite application.

This tutorial demonstrates how to:

- Create a BPEL Module project
- Create the WSDL Documents for Types `Poll` and `Write`
- To add activities to the BPEL process
- Build and deploy a Composite Application project to GlassFish
- Test the composite application