

Quick Start Guide to the CASA Editor

This tutorial provides an introduction to the CASA Editor provided with NetBeans 6.1.

For more information about working with NetBeans IDE, see the *Support and Docs* page on the NetBeans web site.

- “Tutorial Requirements” on page 2
- “Overview” on page 3
- “Create the Sample Application Project” on page 3
- “Open the CASA Editor and Build the Application” on page 4
- “Modify the WSDL Bindings of the Composite Application” on page 5
- “Run a Test Case for the Composite Application” on page 7

Tutorial Requirements

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

Prerequisites to Creating the Project

This tutorial assumes that you have basic knowledge of the following technologies:

- Composite Applications and Service-oriented Architecture (SOA)
- Web Services

Software Needed for This Tutorial

To follow this tutorial, you need the following software and resources.

Software or Resources	Version Required
NetBeans IDE	"All" installation, version 6.0 or 6.1
Java Developer Kit (JDK)	Version 6 or version 5
Java EE-compliant web or application server	GlassFish V2 UR2 for NetBeans 6.1 GlassFish V2 UR1 for NetBeans 6.0

Note – The Download “All” option includes SOA Pack and the GlassFish V2 Application Server, which are required for this tutorial.

Window Layout of the NetBeans IDE

This tutorial assumes you have a standard window layout of the NetBeans IDE which includes the following visible windows:

- Projects
- Files
- Services
- Properties
- Palette (for editors using the Palette window)
- Output

Use the Window menu of the NetBeans IDE to change the layout if necessary.

Overview

This tutorial shows how to use the CASA Editor to edit the configuration and build a composite application.

The composite application you configure and build is the Synchronous Sample composite application provided with the NetBeans 6.1 distribution.

The tutorial illustrates the following:

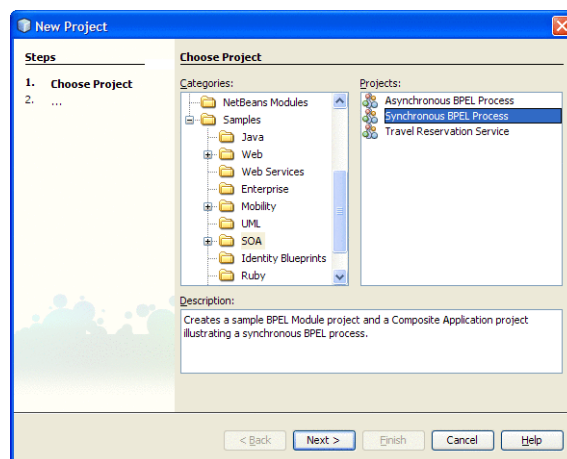
- Building a composite application from within the CASA Editor
- Deleting a connection to an existing WSDL endpoint in the existing composite application
- Adding a connection to a new WSDL endpoint to the composite application
- Modifying the properties of the new WSDL endpoint
- Deploying the composite application
- Running a test case for the composite application

Create the Sample Application Project

In this section you create a project for the sample Synchronous Composite Application, which is provided with NetBeans 6.1.

▼ To create the Synchronous Sample Application project

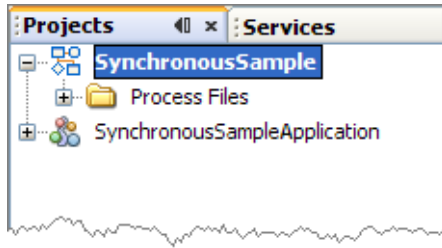
- 1 In the NetBeans IDE, select the Projects tab to display the Projects window.
- 2 From the File menu, select New Project.
- 3 In the Categories list of the New Project window, select Samples > SOA > Synchronous BPEL Process and click Next.



- 4 Accept the default project name and location (or specify an alternate location using the Browse button).
- 5 Click Finish to create the project.

The following projects appear in the Projects window:

- SynchronousSample, the BPEL project
- SynchronousSampleApplication, the composite application project

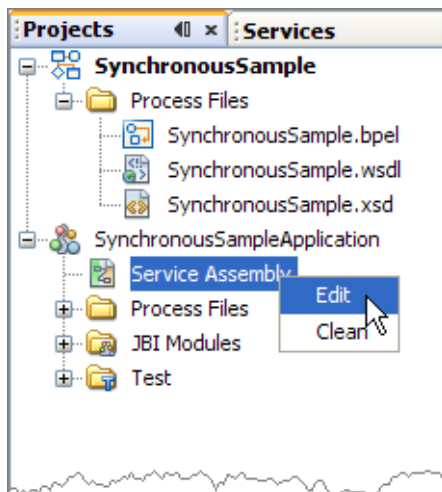


Open the CASA Editor and Build the Application

This section shows how you can use the CASA Editor to build a composite application.

▼ To open the CASA Editor and build the composite application

- 1 In the NetBeans IDE, expand the SynchronousSampleApplication node in the Projects window, right-click Service Assembly and select Edit.

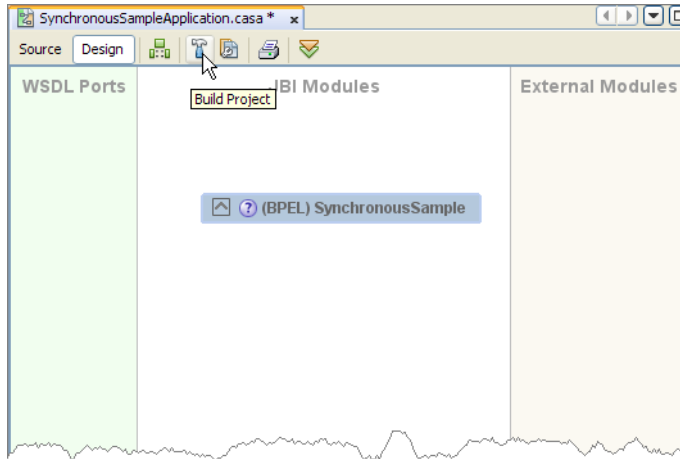


Note – To invoke the CASA Editor in NetBeans 6.0, right-click the composite application in the Projects window and select Edit Application Configuration.

The CASA Editor opens in the NetBeans IDE, as shown in the next step, displaying the Design View of the Synchronous Sample application.

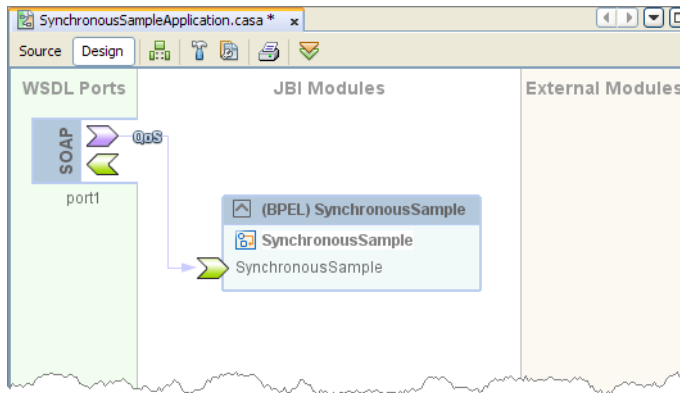
The CASA editor creates and modifies a .casa file, which contains the configuration information for the composite application. For this sample, the CASA Editor created the SynchronousSampleApplication.casa file.

- 2 In the CASA Editor, select the Build Project icon to build the composite application.



The Output window displays messages about the build.

When the build successfully completes, the Design View displays a WSDL port endpoint, a JBI module, and the connection between the endpoint and the JBI module.

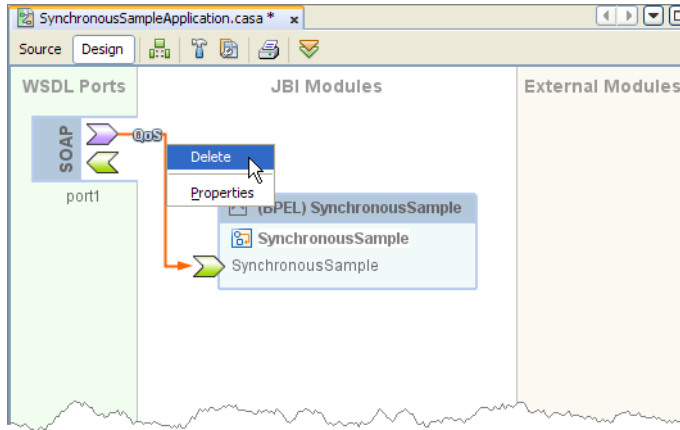


Modify the WSDL Bindings of the Composite Application

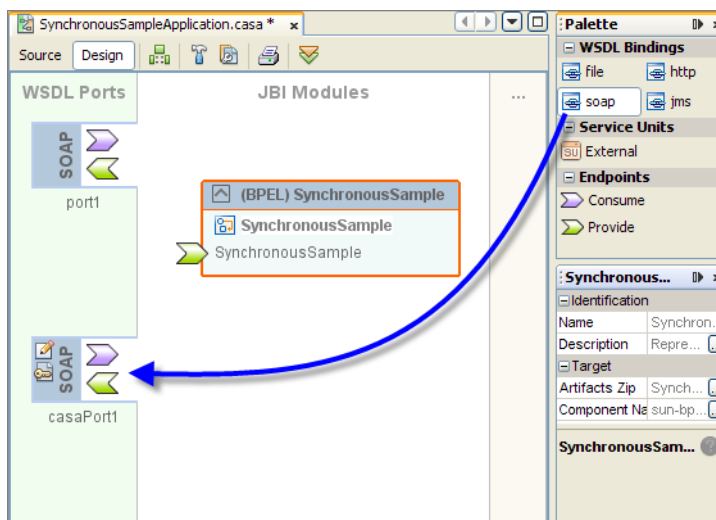
This section shows how to modify connections between a WSDL port and a JBI module, and also how to add and configure new WSDL ports.

▼ To modify the endpoints of the composite application

- 1 In the Design View of the CASA Editor, select the connection between the WSDL port and the JBI module.
- 2 Delete the selected connection using the Delete key from your keyboard.
You can also right-click the connection and select Delete to delete the connection.

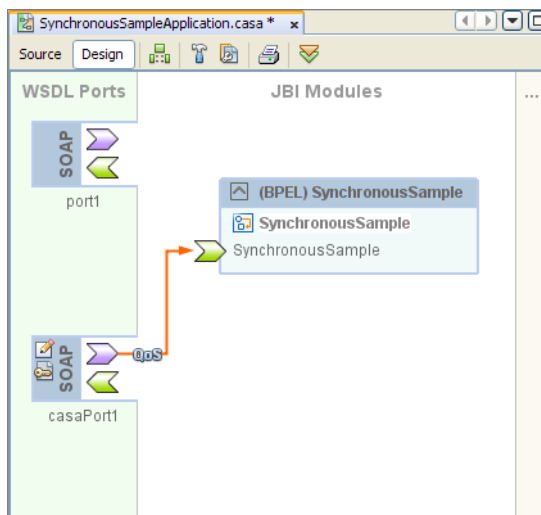


- 3 In the Palette Window of the NetBeans IDE, select the Soap WSDL Binding and drag it to the WSDL Ports area of the CASA Editor.



- 4 Select the Consume endpoint (outbound endpoint) of the newly added SOAP Binding, and drag it to the Provide endpoint (inbound endpoint) of the JBI module.

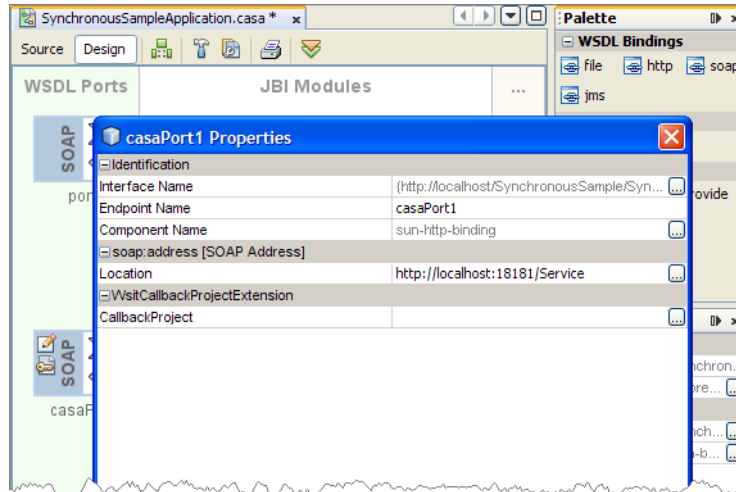
This creates a connection between the SOAP Binding and BPEL JBI module.



- 5 Click the **Edit** icon for the newly added SOAP Binding to view its properties.

This feature allows you to view details of the underlying WSDL without having to navigate through the WSDL file using the WSDL editor interface.

- 6 In the **Properties** window, click the **Edit** button for the **Location** property and enter the property value: `http://localhost:18181/Service`.



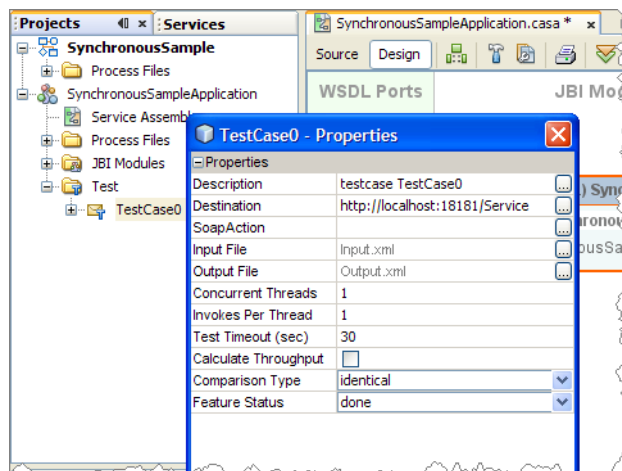
For the next procedure, you will specify this same location, `http://localhost:18181/Service`, in a test case for the composite application.

Run a Test Case for the Composite Application

This section shows how to test the newly configured composite application.

▼ To run a test case for the composite application

- 1 In the **Projects Window** of the NetBeans IDE, expand the `SynchronousSampleApplication > Test` node, right-click the default test case, `TestCase0`, and select **Properties**.



- 2 In the **Properties** window for `TestCase0`, select the **Edit** button for the **Destination** property.

- 3 **Replace the default Destination property with the destination from the previous procedure:**
http://localhost:18181/service.
- 4 **Click OK, and then close the Properties window.**
- 5 **Right-click the SynchronousSampleApplication node, and select Deploy. In the Select Server window, specify the GlassFish Application Server.**

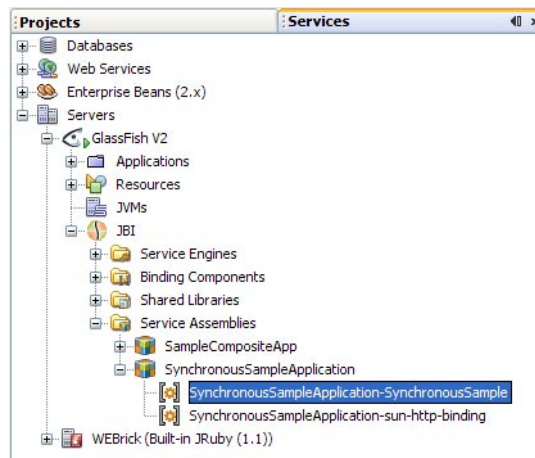
The Output window informs you when the composite application has been successfully deployed.

Notice that the SynchronousSampleApplication Service Assembly has been deployed. You can also view the properties of the Service Units within the Service Assembly to see details of the composite application.

- 6 **In the Services window of the NetBeans IDE, expand the servers > GlassFish V2 > JBI > Service Assemblies node.**

The Output window informs you when the composite application has been successfully deployed.

Notice that the SynchronousSampleApplication Service Assembly has been deployed. You can also view the properties of the Service Units within the Service Assembly to see details of the composite application.

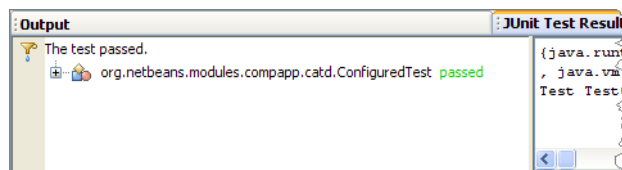


- 7 **In the Projects window, right-click the TestCase0 and select Run.**

Because this is the first time the test case is run, the output is an empty test case, and the Overwrite Empty Output dialog box appears. Select yes to save your most recent output as the test case's expected output.

- 8 **Run the test for TestCase0 again.**

The Output window indicates that the test case was successful.



Copyright 2008 Sun Microsystems, Inc. All rights reserved. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

820-4721-10

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A.

