Distributed NetBeans for OpenVMS Version 6.5.2
Quick Start Guide

September 2010

Contents

1 Introduction

2 Using the Distributed NetBeans Client

- 2.1 Adding and Connecting a Remote IDE Server
- 2.2 Creating an OpenVMS 3GL Project
  - 2.2.1 Creating a New C Source File
  - 2.2.2 Creating a New DCL Command Procedure File
- 2.3 Creating a Java Project
- 2.4 Creating a WSIT Project

3 Conclusion

© Copyright 2010 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

NetBeans, Java, and all Java-based marks are trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation.

X Window System is a trademark of The Open Group.

All other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

1 Introduction

Distributed NetBeans for OpenVMS is a plug-in to the standard NetBeans that allows you to edit, build, execute, and debug programs on an OpenVMS system on the network.

This guide describes how to set up and configure the Distributed NetBeans client on Windows, and how to create an OpenVMS 3GL project and a Java project using the Distributed NetBeans client.

To set up the Distributed NetBeans client for OpenVMS on Windows, the following installation must be performed:

- Install the latest Java JDK Version 5.0 Update 14 or later.
  You can download and install Java JDK from the following web address:
  http://java.sun.com/javase/downloads/index_jdk5.jsp

  Note: JSE v 1.6-04 is not supported)

- Install NetBeans Integrated Development Environment (IDE) Version 6.5 for Windows, which is an open-source on the Windows client). You can download and install NetBeans IDE Version 6.5 from the following web address:
  http://netbeans.org/downloads/6.5/index.html
- Install the Distributed NetBeans Client for OpenVMS, which is a plug-in for NetBeans running on your desktop. You can download and install the Distributed NetBeans Client Module (.nbm file) from the following web address:
  http://h71000.www7.hp.com/openvms/products/ips/netbeans/distnb_files.html

Prerequisites

For information about hardware and software prerequisites, see the *Distributed NetBeans for OpenVMS Installation Guide and Release Notes*:

http://h71000.www7.hp.com/openvms/products/ips/netbeans/docs/DISTNETBEANS_IGUIDE_RELNOTES.HTML#before

2 Using the Distributed NetBeans Client

This section addresses the following topics:

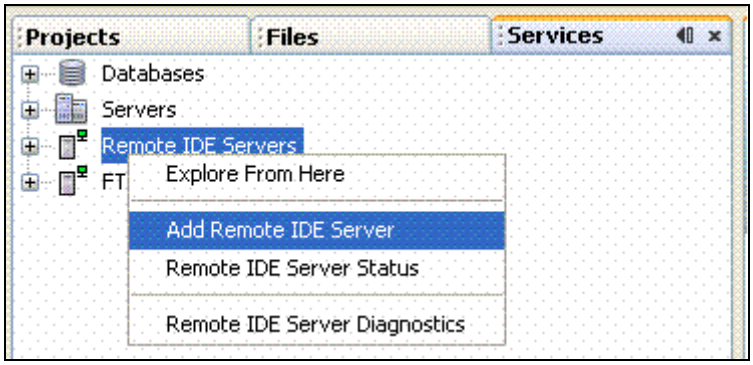
- Adding and Connecting to a Remote IDE Server
- Creating an OpenVMS 3GL Project
- Creating a Java Project
- Creating a WSIT Project

2.1 Adding and Connecting to a Remote IDE Server

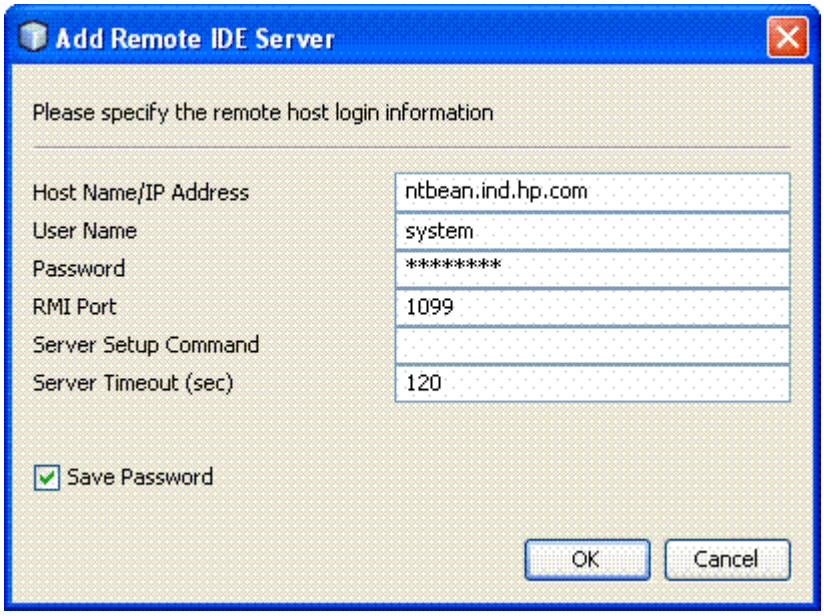
Before creating a remote project, we need a process on the remote OpenVMS machine which will act on our behalf to execute DCL commands, etc. This process is called as “Remote IDE Server”. You create this Remote IDE Server from within the IDE. The Remote IDE Server you create will be a detached process running in your user account on OpenVMS. The Remote IDE Server will make a connection back to the Distributed NetBeans client in the IDE using Java’s RMI (Remote Method Invocation) services.

To add or connect the remote IDE server on an OpenVMS system, follow these steps:

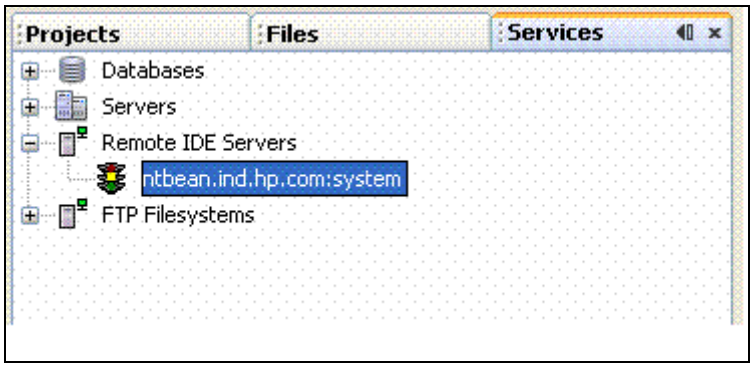
1. In the NetBeans IDE window, select the **Services** tab.
2. Right-click **Remote IDE Servers** and select **Add Remote IDE Server**. The **Add Remote IDE Server** dialog box is displayed.



3. Enter the Host Name/IP Address, User Name, and Password information of the OpenVMS system you are adding and click **OK** to close the dialog box and return to the NetBeans IDE window.



4. The **Remote IDE Server** is automatically connected. The connection icon turns yellow when the connection is in progress, then green to show that you are connected to the remote server.



Section 2.2 explains how to create an OpenVMS 3GL application project on the remote OpenVMS system.

2.2 Creating an OpenVMS 3GL Project

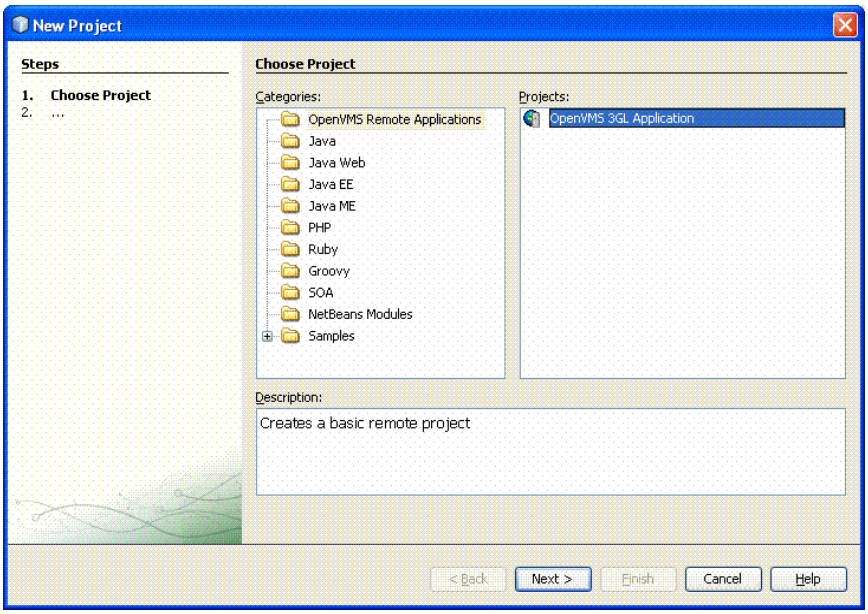
In NetBeans 6.5, you must create a project to work with files on the remote OpenVMS system. Your project can be a Java project or an OpenVMS 3GL application project.

The OpenVMS 3GL project type is useful for OpenVMS applications that are written in 3GL languages instead of Java. This project type allows you to map an OpenVMS source directory to a directory on your desktop. It also allows you to edit and compile files written in the following 3GL languages:

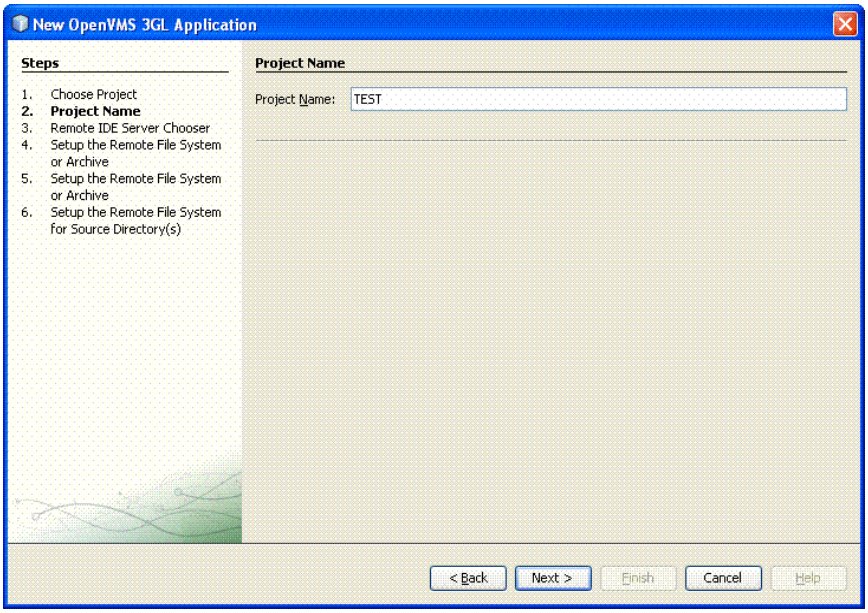
- C/C++
- Cobol
- Fortran
- Basic
- Pascal

To create an OpenVMS 3GL project from the Distributed NetBeans client, follow these steps:

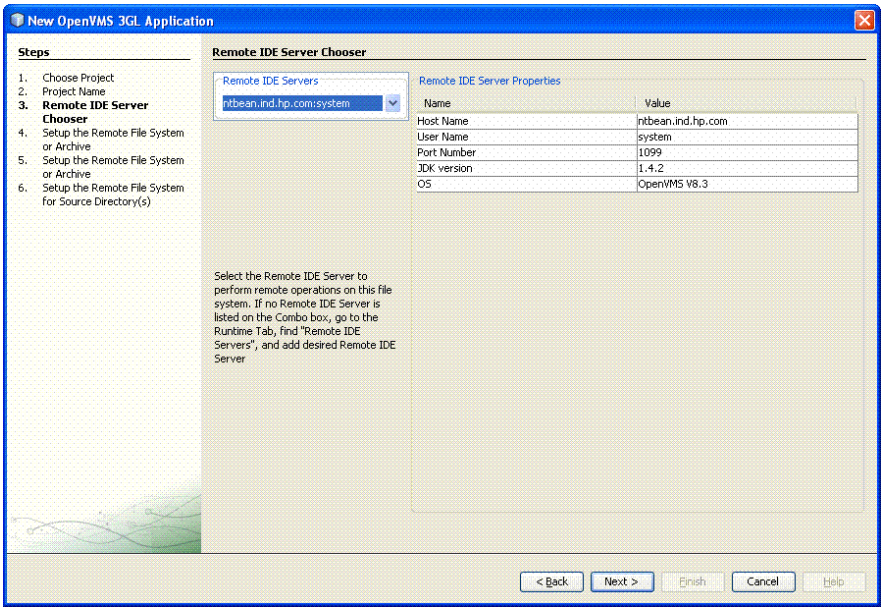
1. In the NetBeans IDE window, select the **Projects** tab.
2. Select **File > New Project**. The **New Project** wizard is displayed.
3. Select **OpenVMS Remote Applications**, and click **Next**.



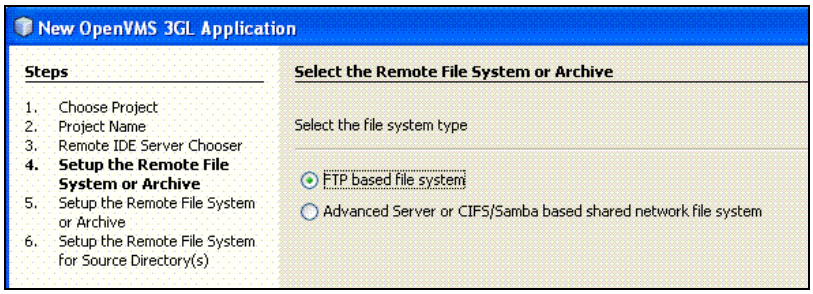
The **New OpenVMS 3GL Application** dialog box is displayed. Enter the “Project Name” and click **Next.**



4. Select the name of the **Remote IDE Server** from the drop-down menu and click **Next**. The **New OpenVMS 3GL Application** dialog box is displayed.

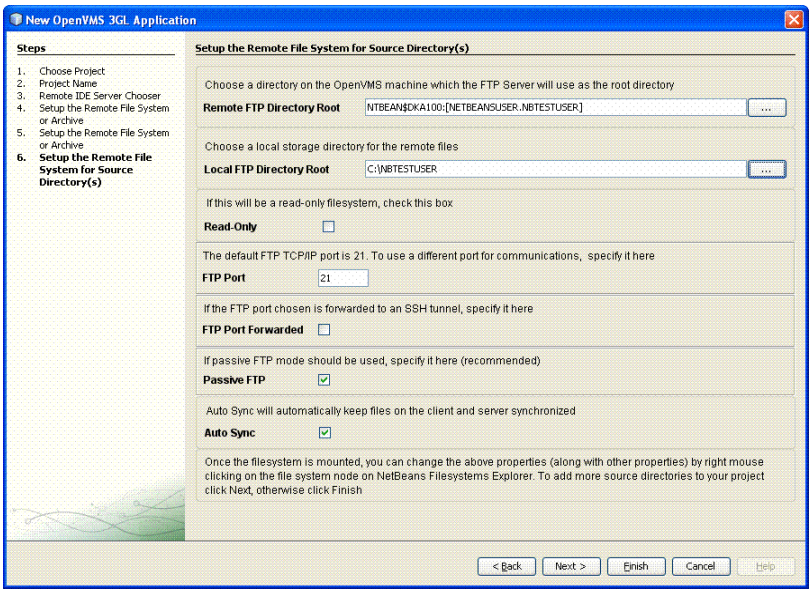


5. Select the remote file system type as **FTP based file system**. Click **Next** to view the **New OpenVMS 3GL Application** dialog box.



6. In the **New OpenVMS 3GL Application** dialog box, enter the following:

- **Remote FTP Directory Root** – Browse to select a remote default directory for the FTP filesystem root on the OpenVMS system. The Remote FTP Directory Root is the root directory that allows you to access and see your source files from the OpenVMS system. Usually, this will be your project directory. For example, USER$:[NBUSER.NBTESTUSER].
- **Local FTP Directory** Root – Browse to select a Local FTP Directory Root to store copies of the files from the OpenVMS system. This directory will contain some or all of the files found in the remote FTP default directory.

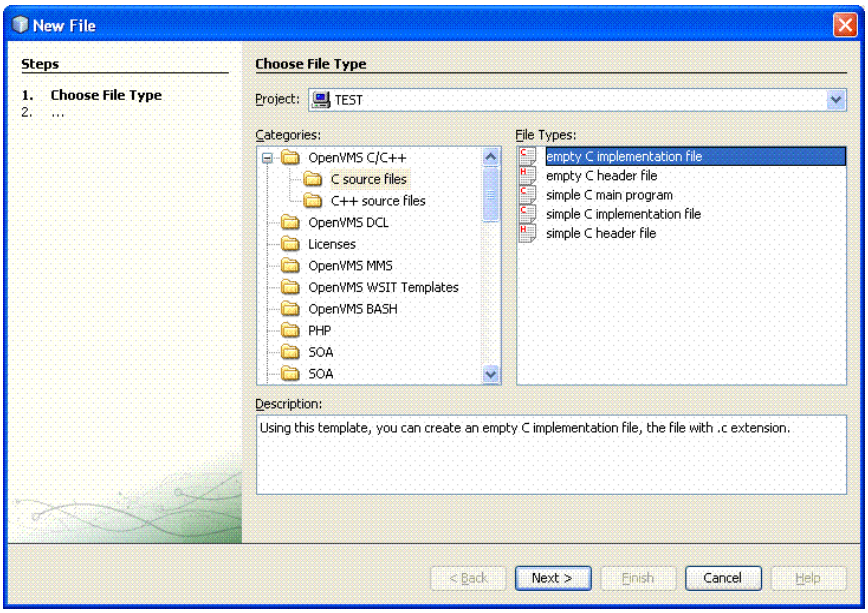


The following sections explain how to create a new C source file and a new DCL command procedure (.com) file on the remote OpenVMS system.

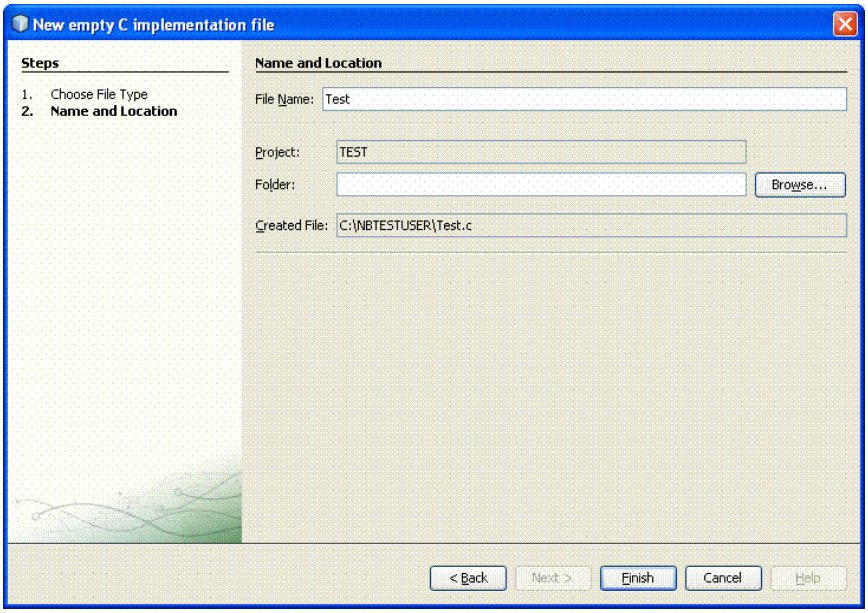
2.2.1 Creating a New C Source File

To create a new C source file in the *Test* project on the remote OpenVMS system, follow these steps:

1. From the main menu in NetBeans IDE, select **File > New File**. The **New File** dialog box is displayed. Alternately, you can right-click your project and select **New File.**
2. Click the plus sign (+) next to OpenVMS C/C++ listed under **Categories** and select C source files.
3. In the **File Types**, select the required template file and click **Next**.

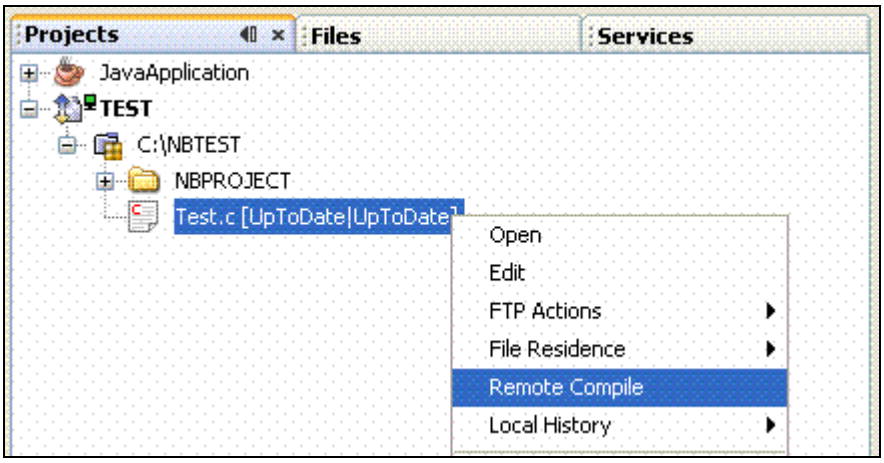


The **New empty C implementation file** dialog box is displayed. Enter the “File Name” and click **Browse** button and create a C file in a different folder and click **Finish**.

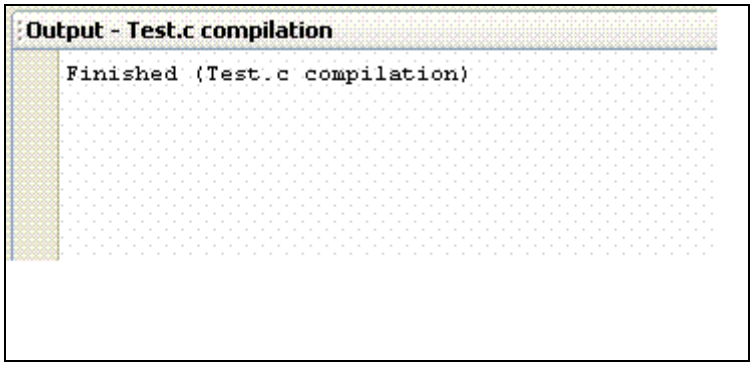


A C source file with **.**c extension is created in the NetBeans IDE. Your C file will appear in your **Projects** tab as remote filesystem.

4. In the **source editor** window, edit *test.c* source file and save it.
5. In the **Projects** tab, right-click on the *test.c* file and select **Remote Compile**.



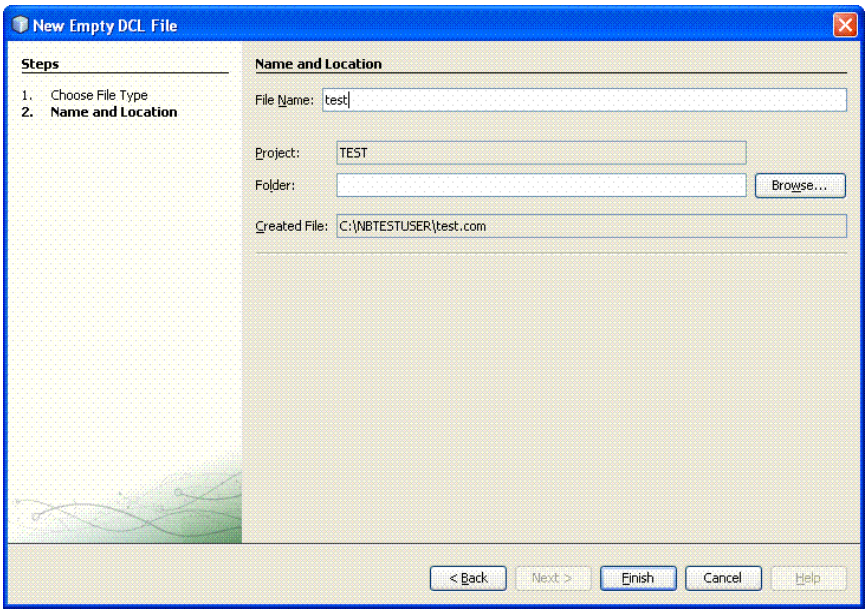
After compilation is done, an OBJ file is created and the following message is displayed in the NetBeans IDE **Output** Window. Then, follow the procedure outlined in Section 2.2.2, *Creating a DCL Command Procedure File.*



2.2.2 Creating a New DCL Command Procedure File

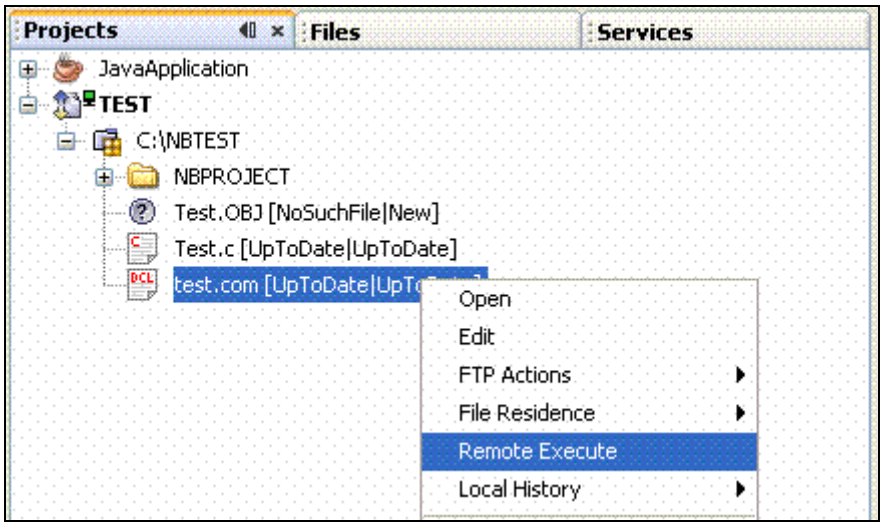
To create a new DCL command procedure (.com) file in the *Test* project on the remote OpenVMS system, follow these steps:

1. From the main menu in NetBeans IDE, select **File > New File**. The **New File** dialog box is displayed.
2. Select the OpenVMS DCL option listed under **Categories**.
3. In the **File Types**, select the required template file and click **Next**. The **New Empty DCL File** dialog box is displayed.
4. Enter the “File Name” . Click the **Browse** button to create the com file in a different folder.and click **Finish**.



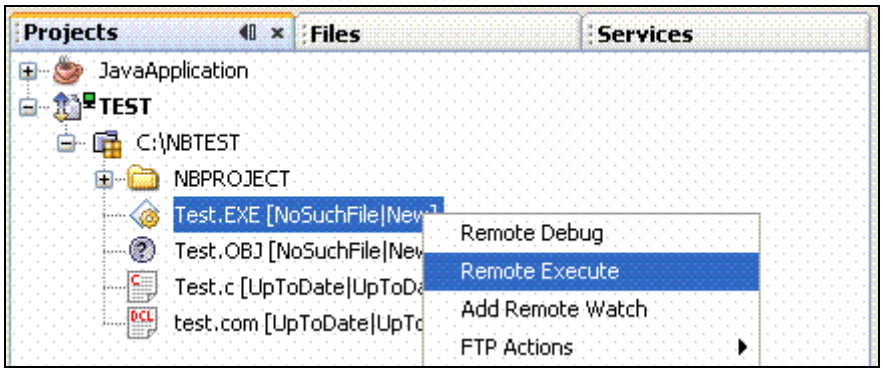
A DCL command procedure file with .com extension is created in the NetBeans IDE. Your DCL command procedure file will appear in the **Projects** tab.

5. In the **source editor** window, edit *test.com* file to create the required image (EXE) and save it.
6. In the **Projects** tab, right-click on the *test.com* file and select **Remote Execute**.

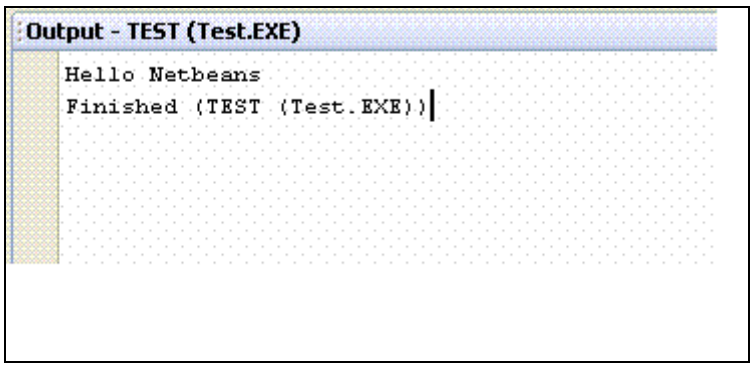


After DCL execution, an executable image (EXE) is created and the following message is displayed in the NetBeans IDE **Output** window.

7. Right-click the EXE file and select **Remote Execute**.



The NetBeans IDE **Output** window is displayed.



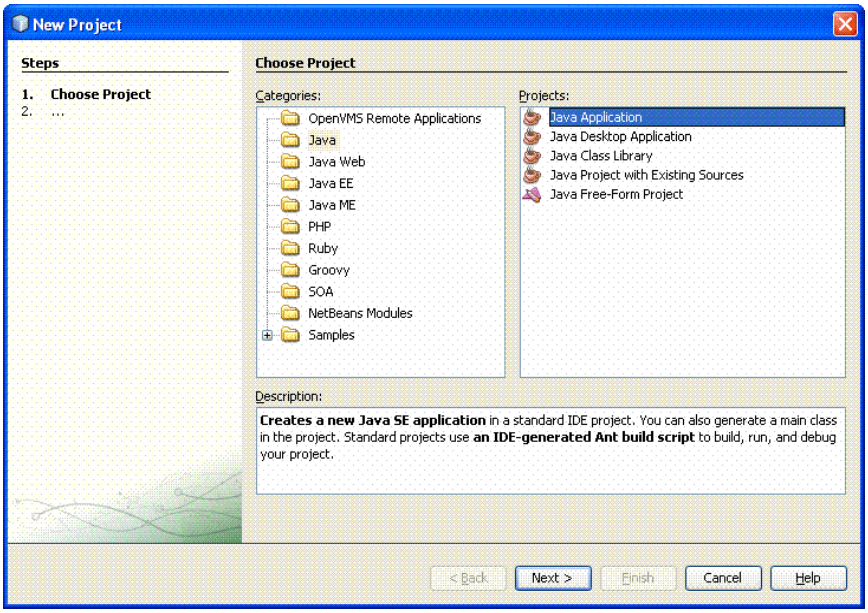
2.3 Creating a Java Project

For creating a Java project, you have to create a standard Java or web-based project and then convert it to a remote project. The conversion to remote project can occur immediately after you create the project or sometime later.

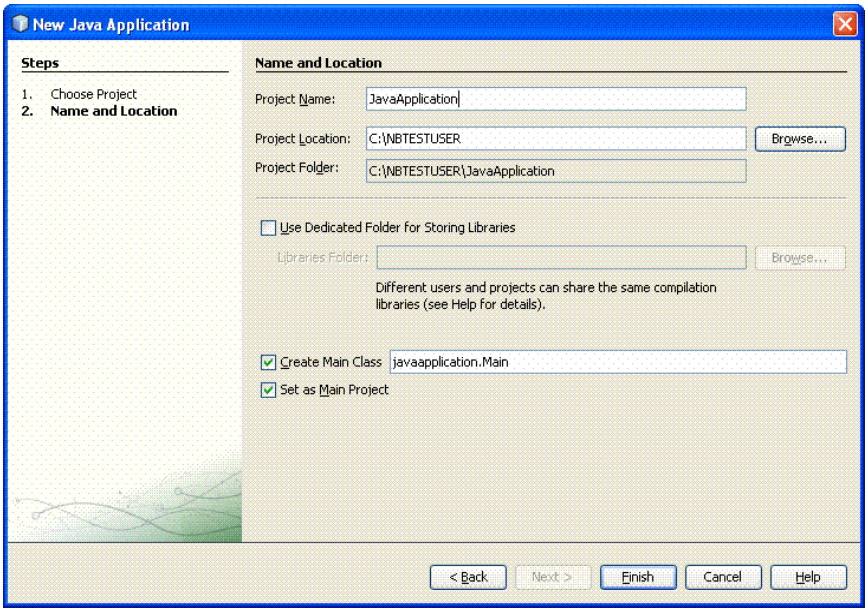
When you convert a project to a remote project, your project files and source files will be copied to the OpenVMS system and modifications will be made to the NetBeans-generated `BUILD.XML` to support remote operations.

To create a Java project using Distributed NetBeans, follow these steps:

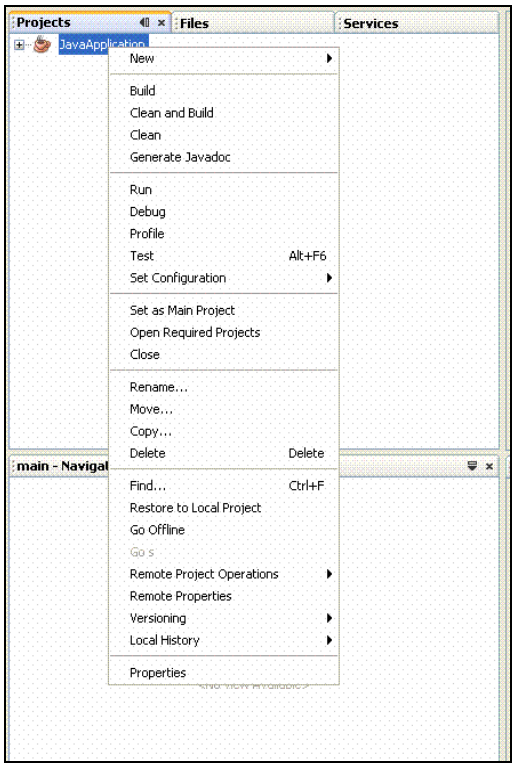
1. From the main menu in NetBeans IDE, select **File > New Project**. The **New Project** dialog box is displayed.
2. Select **Java** listed under **Categories.**
3. In the **Projects** tab, select the required Java Application template file and click **Next**.



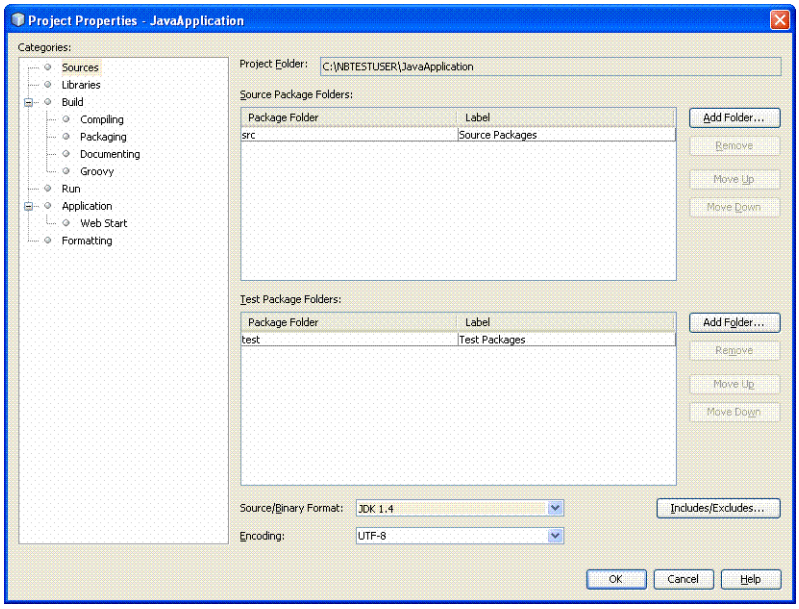
4. The **New Java Application** dialog box is displayed. Enter the “Project Name”, “Project Location” and “Project Folder” and click **Finish** to close the dialog box and return to the NetBeans IDE window.



5. Right-click on your Java project and select **Properties**.

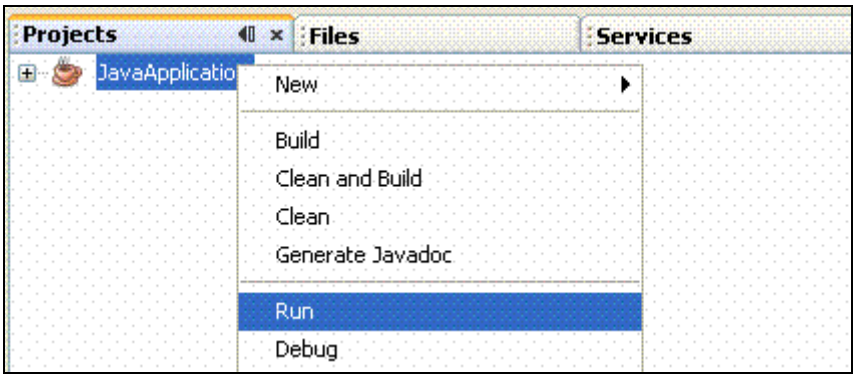


The **Project Properties – JavaApplication** dialog box is displayed.

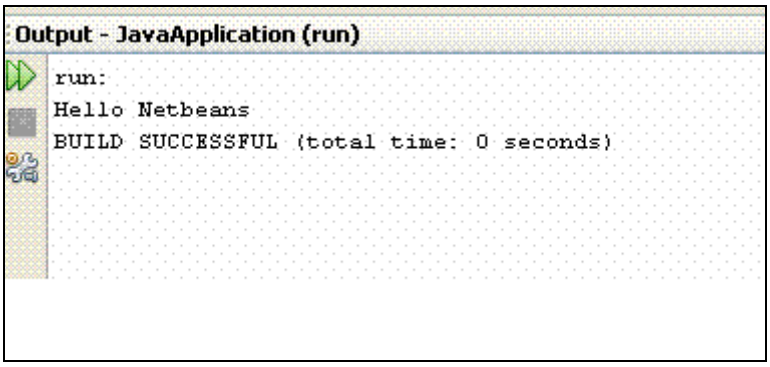


The Java source level should be less than or equal to the version of Java with which you are running the IDE Server on OpenVMS. To change the source level for your project, set the proper source level from the **Source Level** drop-down menu.

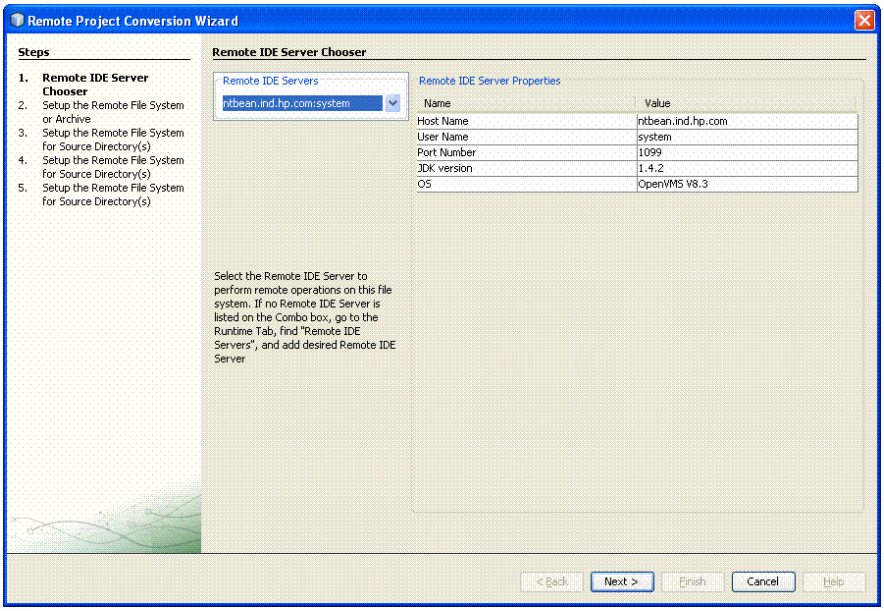
6. In the source editor window, make the required edits to your program in the *JavaApplication.main* file and save it.
7. In the **Projects** tab, right-click on your project and select **Clean and Build**.
8. Right-click your project and select **Run** to run your Java Application.



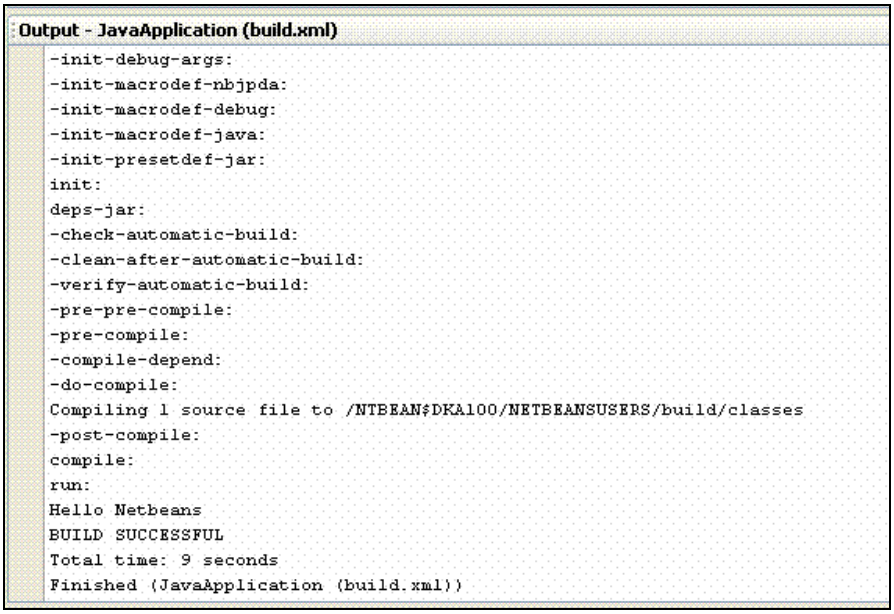
The NetBeans IDE **Output** window is displayed.



9. Convert to remote project by right-clicking on your project and select **Convert to Remote Project**. The **Remote Project Conversion Wizard** dialog box is displayed. Then, follow the steps in the wizard and click **Finish**.



10. After the project has been converted to remote project, right-click and select **Remote Project Operations** and then select **Run Project**. The NetBeans IDE **Output** window is displayed.

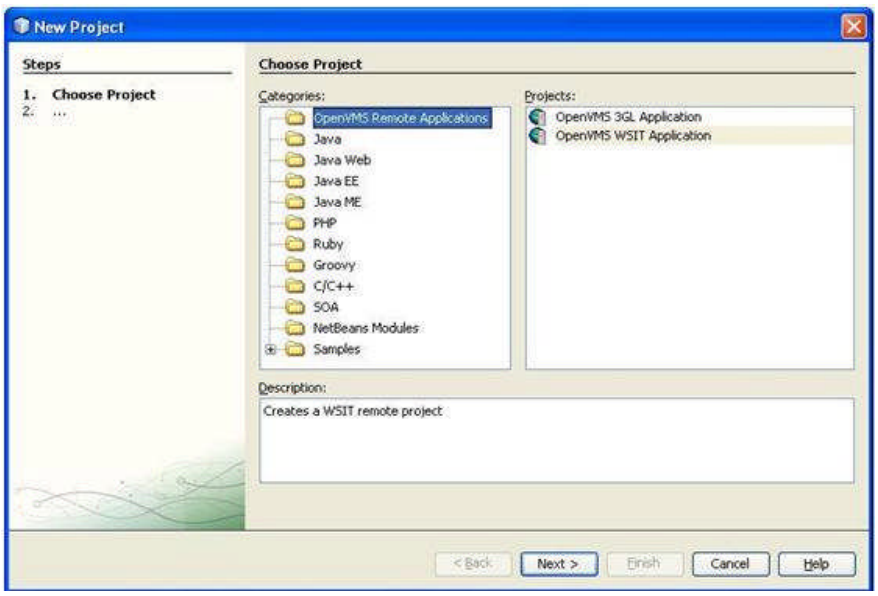


2.4 Creating a WSIT Project

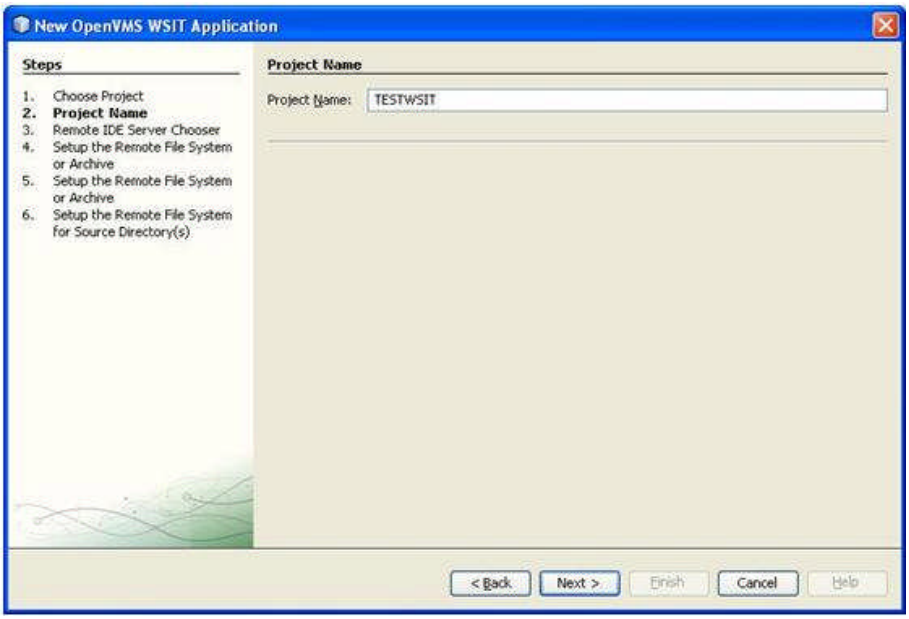
The Web Service Integration Toolkit for OpenVMS (WSIT) contains a collection of integration tools. These tools are easy to use, highly extensible, based on standards and built on open source technology. The toolkit can be used to call OpenVMS applications written in 3GL languages, such as C, BASIC, COBOL, FORTRAN, and ACMS from newer technologies and languages such as Java, Microsoft .NET, Java -RMI, JMS, and web services.

To create an OpenVMS WSIT project from the Distributed NetBeans client, follow these steps:

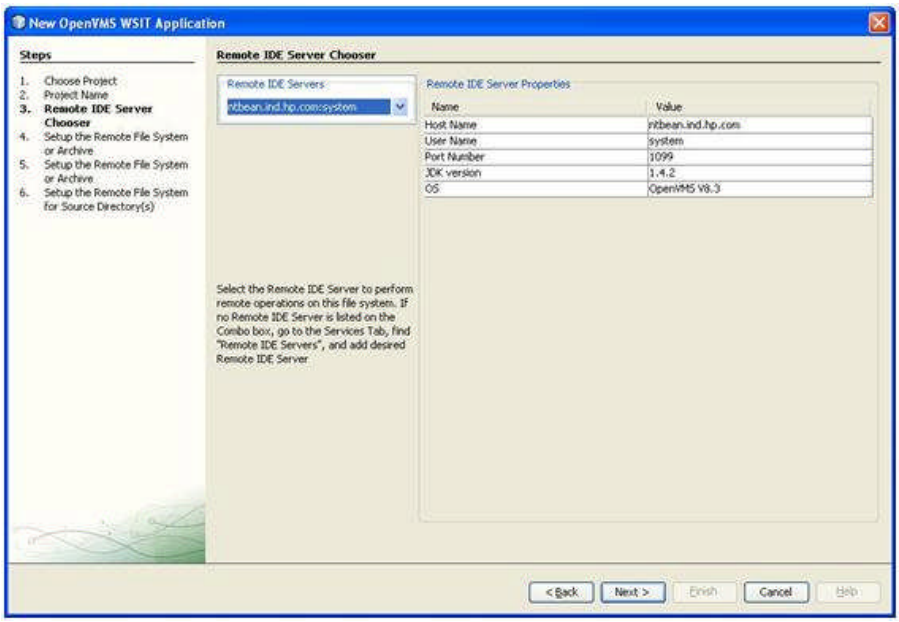
1. In the NetBeans IDE window, select the **Projects** tab.
2. Select **File > New Project**. The **New Project** wizard is displayed.
3. Select **OpenVMS Remote Applications**.
4. Select **OpenVMS WSIT Application,** and click **Next**.



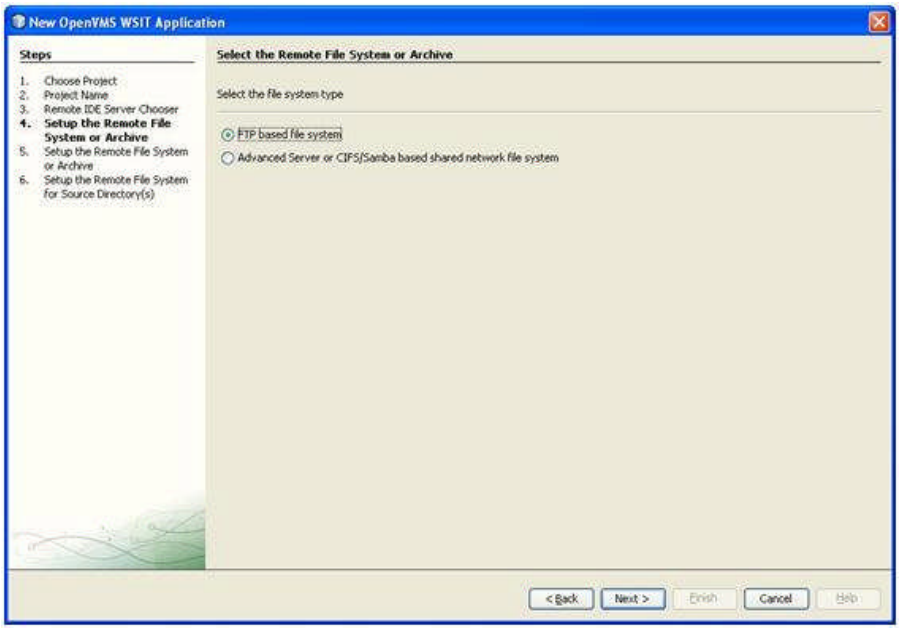
The **New OpenVMS WSIT Application** dialog box is displayed. Enter the “Project Name” and click **Next**.



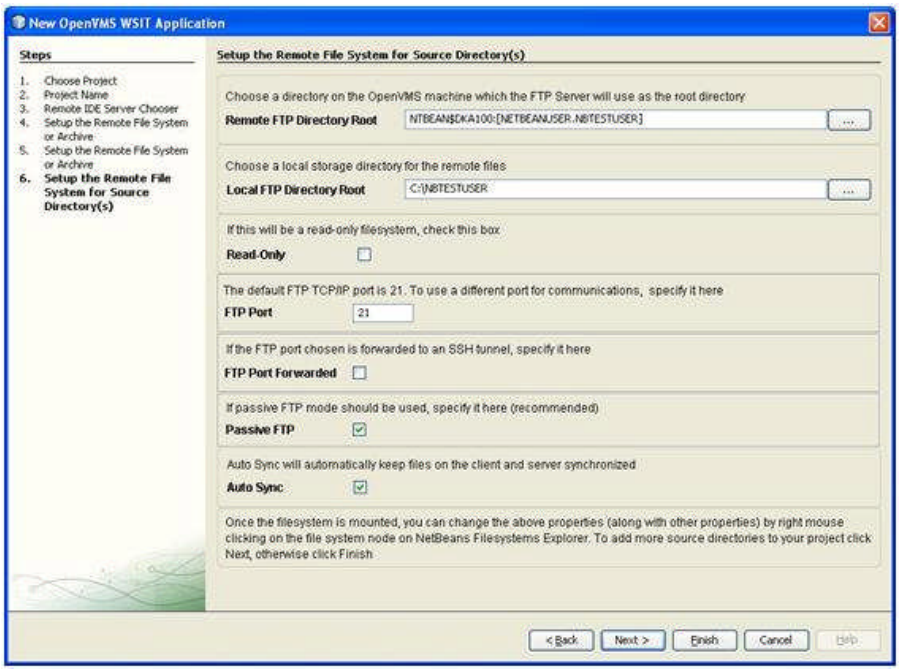
5. Select the name of the **Remote IDE Server** from the drop-down menu and click **Next**. The **New OpenVMS WSIT Application** dialog box is displayed.



6. Select the remote file system type as **FTP based file system**. Click **Next** to view the **New OpenVMS WSIT Application** dialog box.



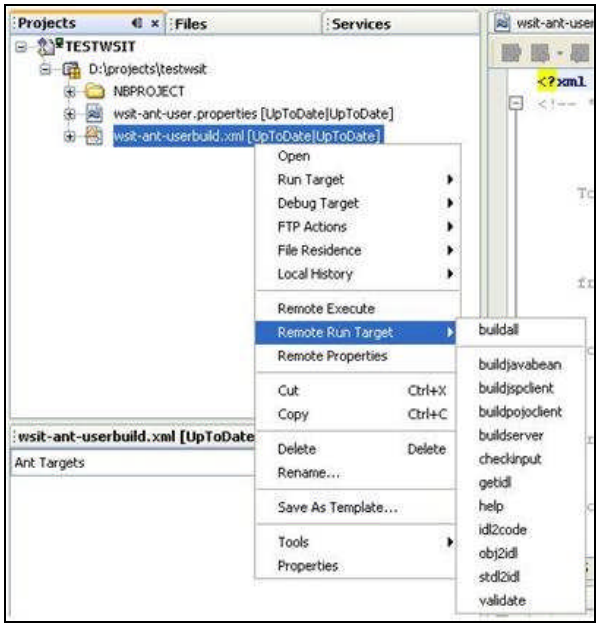
7. In the **New OpenVMS WSIT Application** dialog box enter the following:
   - **Remote FTP Directory Root** – Browse to select a remote default directory for the FTP filesystem root on the OpenVMS system. The Remote FTP Directory Root is the root directory that allows you to access and see your source files from the OpenVMS system. Usually, this will be your project directory. For example, USER$:[NBUSER.NBTESTUSER].
   - **Local FTP Directory** Root – Browse to select a Local FTP Directory Root to store copies of the files from the OpenVMS system. This directory will contain some or all of the files found in the remote FTP default directory.



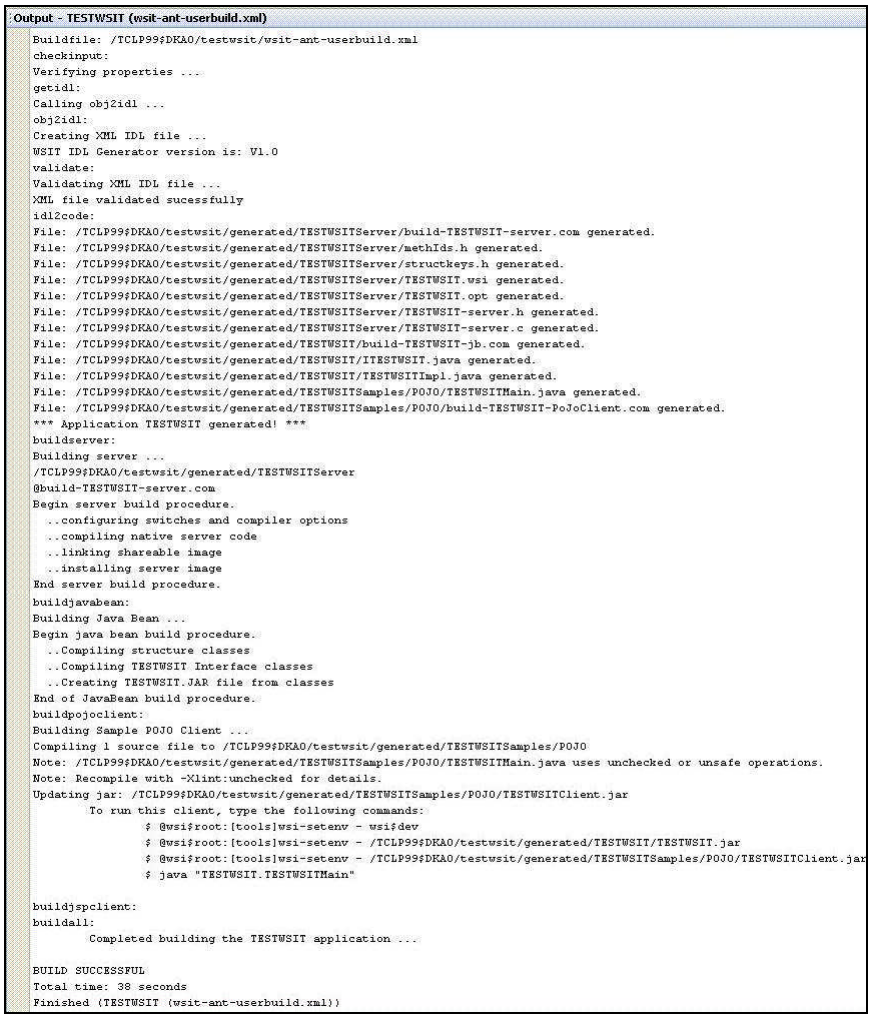
8. The WSIT project window is displayed. To create a new C source file, see Section 2.2.1, *Creating a New C Source File*

9. In the **Projects** tab, right-click on the *test.c* file and select **Remote Compile**. After compiling, .OBJ file is created.

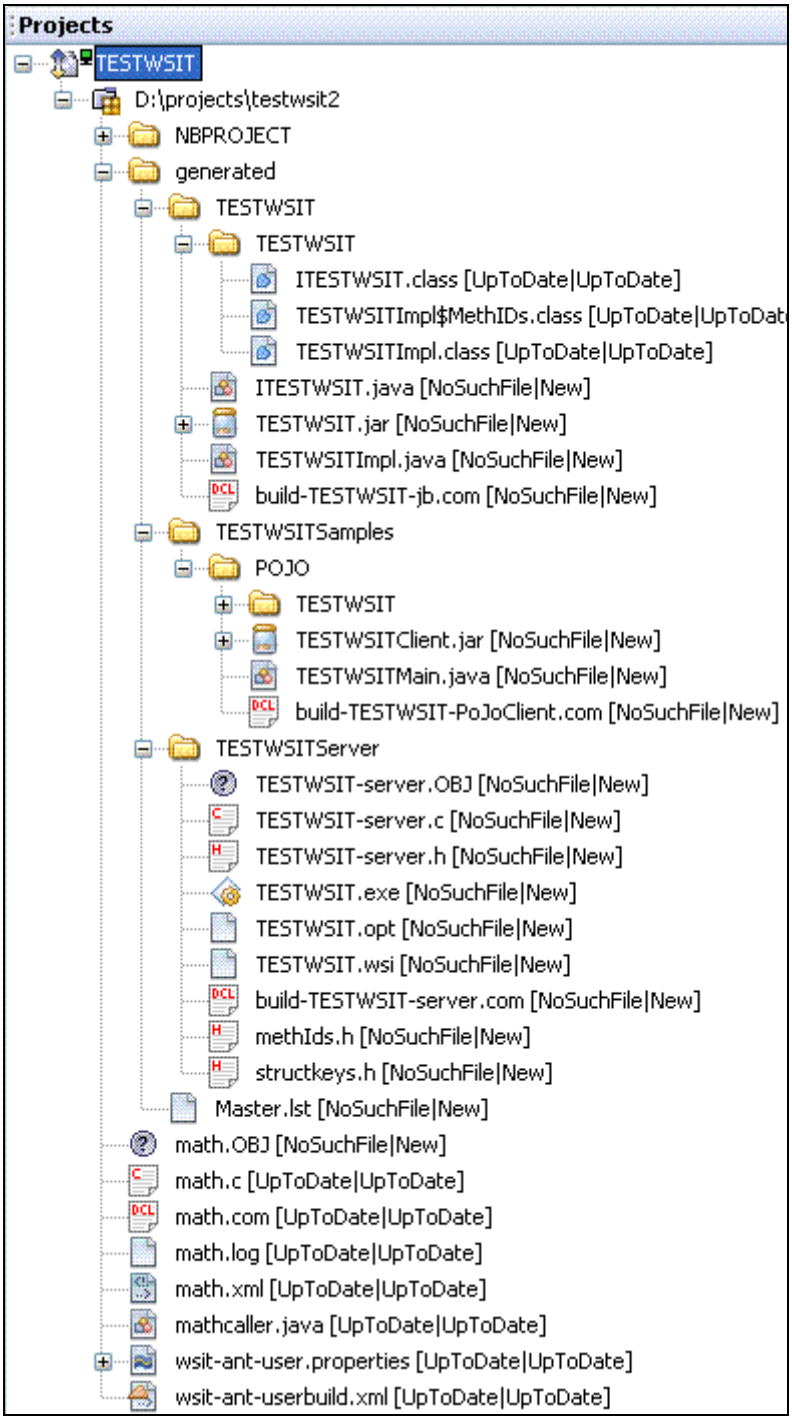
10. Specify the path of the .OBJ file in the *wsit-ant-user.properties* file template created along with the WSIT project.
11. Right-click the *wsit-ant-user.xml* file and select **Remote Run Target**.



The NetBeans IDE **Output** window is displayed.



The generated files are listed in the directory as shown:



3 Conclusion

This document has explained briefly on how to create, build, and execute your OpenVMS 3GL program, Java program, and WSIT applications. For more information about Distributed NetBeans, see:

http://h71000.www7.hp.com/openvms/products/ips/netbeans/distnb.html