# Distributed NetBeans for OpenVMS Version 7.0.1
# Quick Start Guide

**December 2012**

## Contents

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.

# 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:

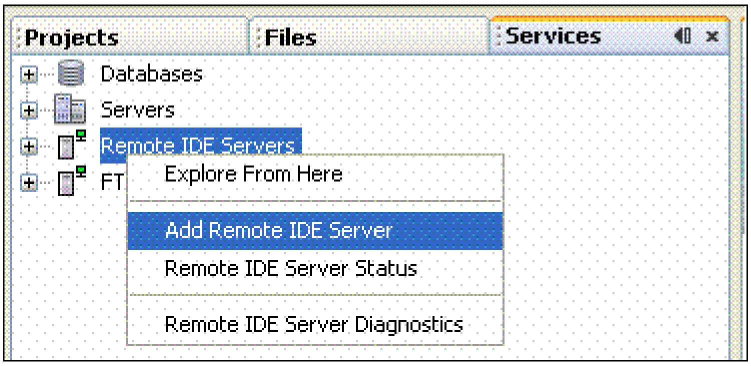
- Adding and Connecting to a Remote IDE Server
- Creating an OpenVMS 3GL Project
- Creating a Java Project
- Creating a WSIT Project

## 2.1 Adding and Connecting to a Remote IDE Server

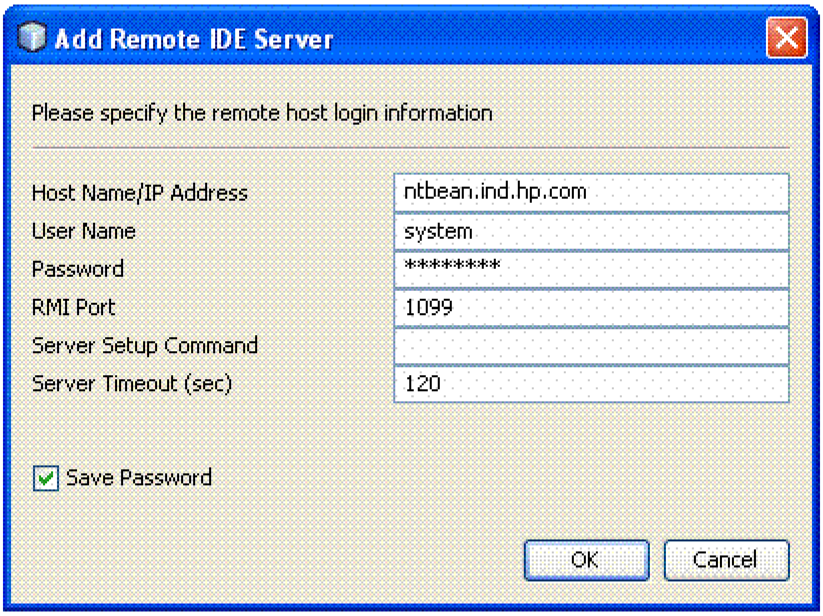
Before creating a remote project, we need a process on the remote OpenVMS machine which will act on our behalf to execute DCL commands, etc. This process is called as “Remote IDE Server”. You create this Remote IDE Server from within the IDE. The Remote IDE Server you create will be a detached process 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.

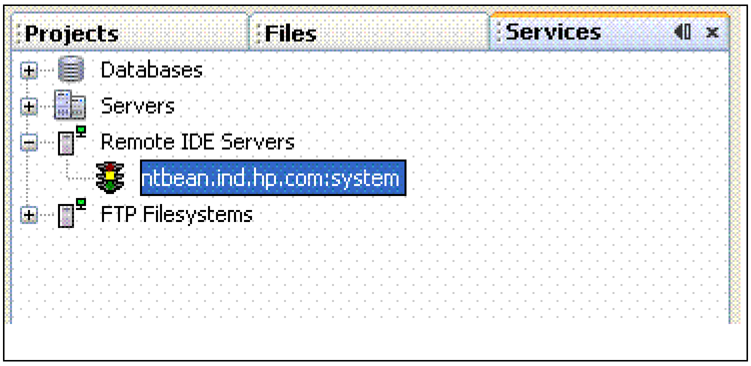


Please specify the remote host login information

|  |  |
|---|---|
| Host Name/IP Address | ntbean.ind.hp.com |
| User Name | system |
| Password | ******** |
| RMI Port | 1099 |
| Server Setup Command |  |
| Server Timeout (sec) | 120 |

☑ Save Password

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

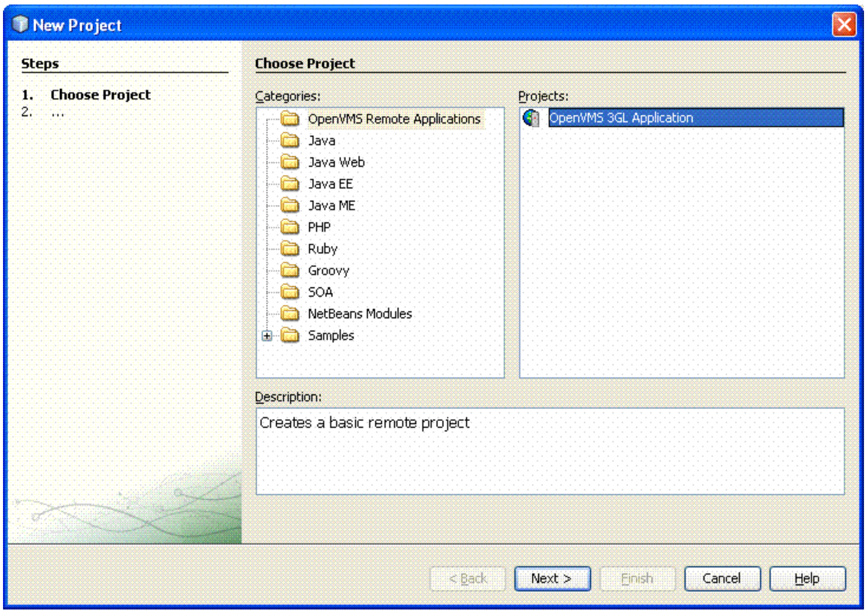
In NetBeans 6.5, you must create a project to work with files on the remote OpenVMS system. Your project can be a Java project or an OpenVMS 3GL application project.

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

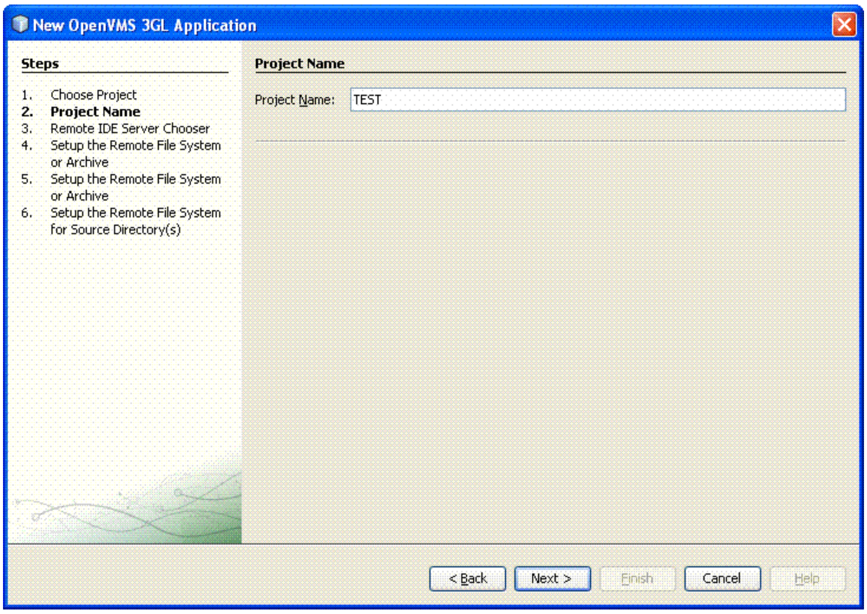
To create an OpenVMS 3GL project from the Distributed NetBeans client, follow these steps:

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**.



Creates a basic remote project

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


New OpenVMS 3GL Application

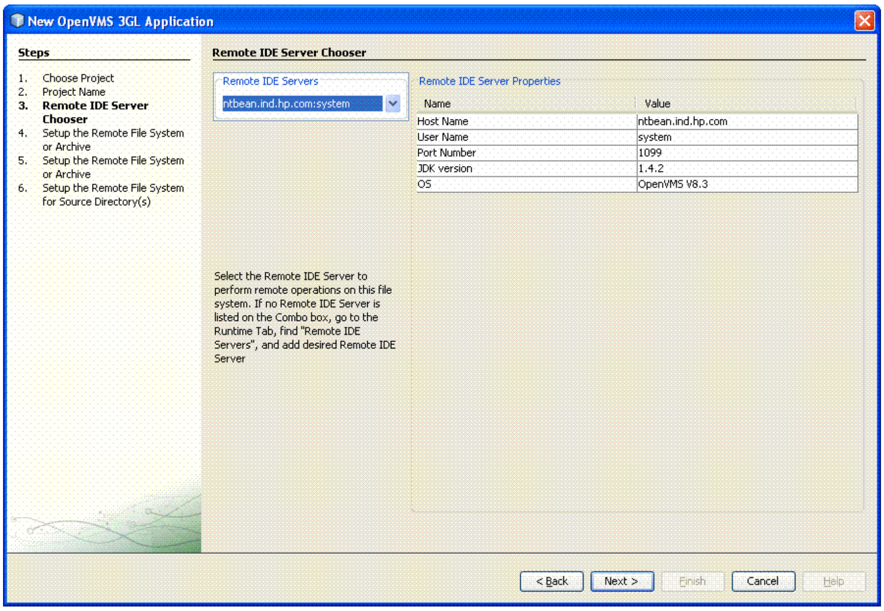
Steps

1. Choose Project
2. **Project Name**
3. Remote IDE Server Chooser
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

Project Name

Project Name: TEST

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.


New OpenVMS 3GL Application

Steps

1. Choose Project
2. Project Name
3. **Remote IDE Server Chooser**
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

Remote IDE Server Chooser

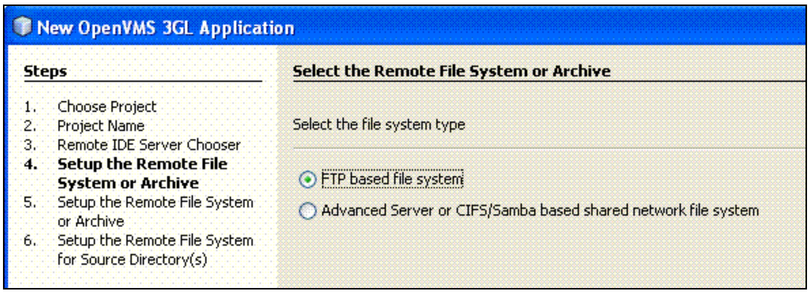
Remote IDE Servers: ntbean.ind.hp.com:system

Remote IDE Server Properties

| Name | Value |
|---|---|
| Host Name | ntbean.ind.hp.com |
| User Name | system |
| Port Number | 1099 |
| JDK version | 1.4.2 |
| OS | OpenVMS V8.3 |

Select the Remote IDE Server to perform remote operations on this file system. If no Remote IDE Server is listed on the Combo box, go to the Runtime Tab, find "Remote IDE Servers", and add desired Remote IDE Server

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



New OpenVMS 3GL Application

Steps

1. Choose Project
2. Project Name
3. Remote IDE Server Chooser
4. **Setup the Remote File System or Archive**
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

**Select the Remote File System or Archive**

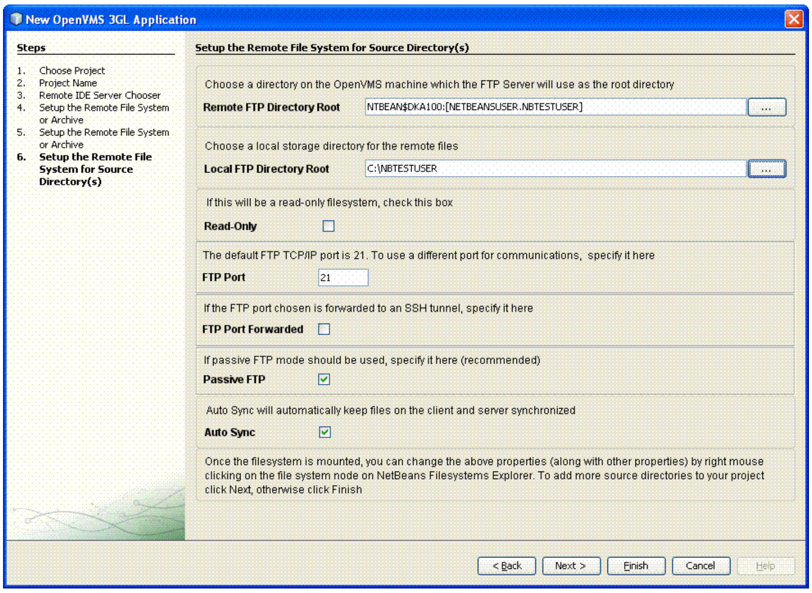
Select the file system type

● FTP based file system

○ Advanced Server or CIFS/Samba based shared network file system

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.



New OpenVMS 3GL Application

Steps

1. Choose Project
2. Project Name
3. Remote IDE Server Chooser
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. **Setup the Remote File System for Source Directory(s)**

**Setup the Remote File System for Source Directory(s)**

Choose a directory on the OpenVMS machine which the FTP Server will use as the root directory

**Remote FTP Directory Root** NTBEAN$DKA100:[NETBEANSUSER.NBTESTUSER]

Choose a local storage directory for the remote files

**Local FTP Directory Root** C:\NBTESTUSER

If this will be a read-only filesystem, check this box

**Read-Only** □

The default FTP TCP/IP port is 21. To use a different port for communications, specify it here

**FTP Port** 21

If the FTP port chosen is forwarded to an SSH tunnel, specify it here

**FTP Port Forwarded** □

If passive FTP mode should be used, specify it here (recommended)

**Passive FTP** ☑

Auto Sync will automatically keep files on the client and server synchronized

**Auto Sync** ☑

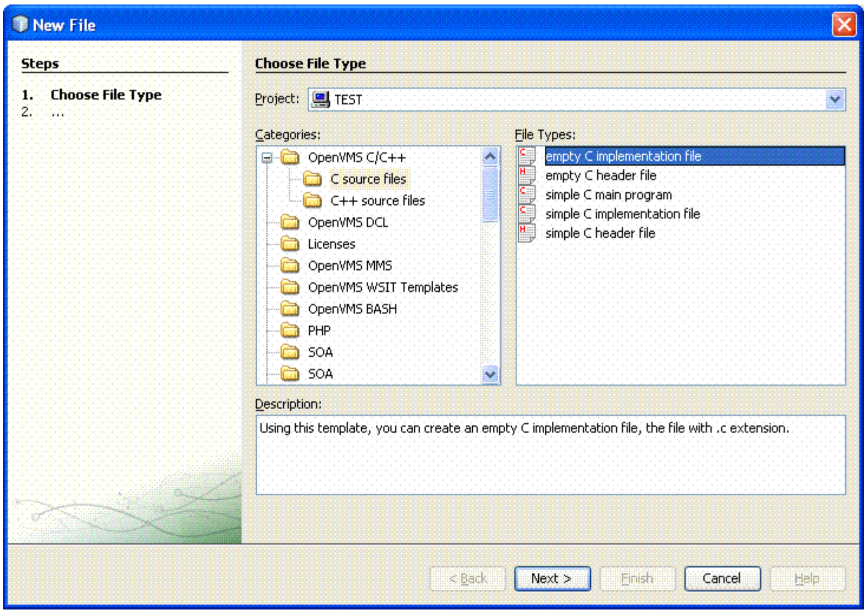
Once the filesystem is mounted, you can change the above properties (along with other properties) by right mouse clicking on the file system node on NetBeans Filesystems Explorer. To add more source directories to your project click Next, otherwise click Finish

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**.

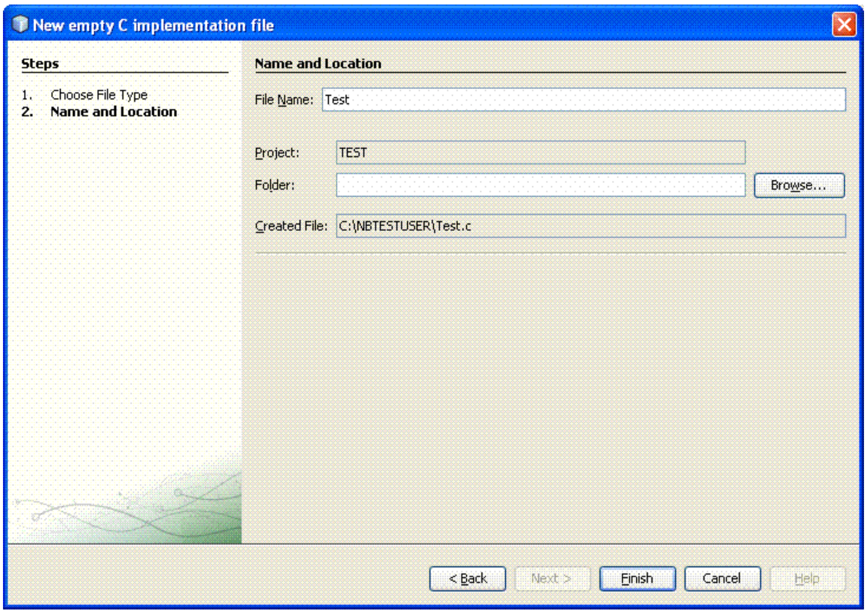


Project: TEST

Description:

Using this template, you can create an empty C implementation file, the file with .c extension.

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**.



File Name: Test

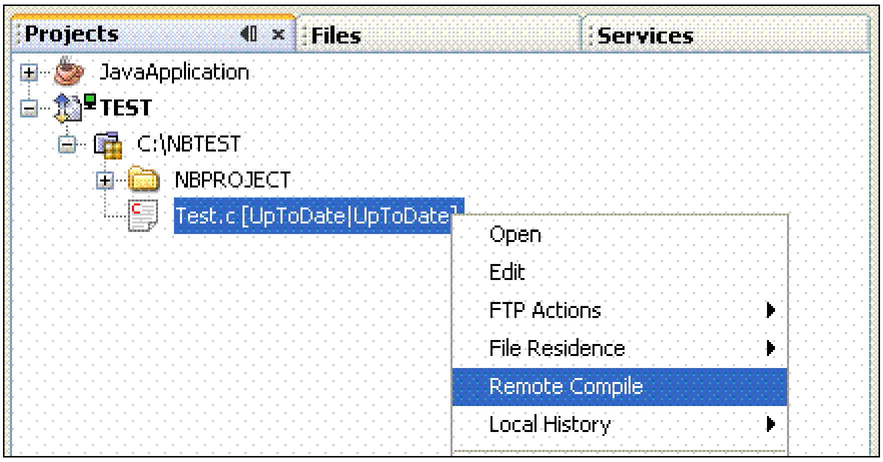
Project: TEST

Folder:

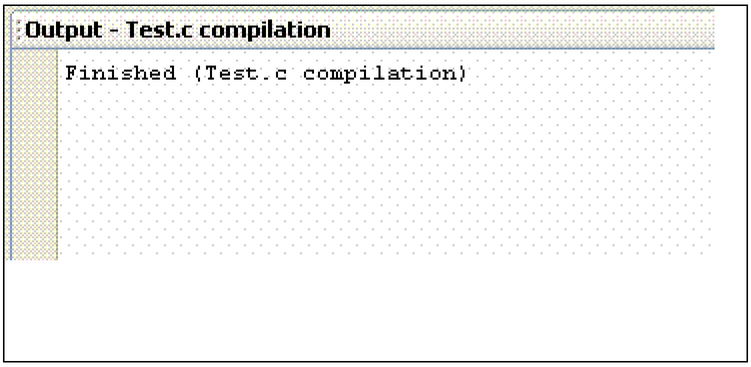
Created File: C:\NBTESTUSER\Test.c

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.*

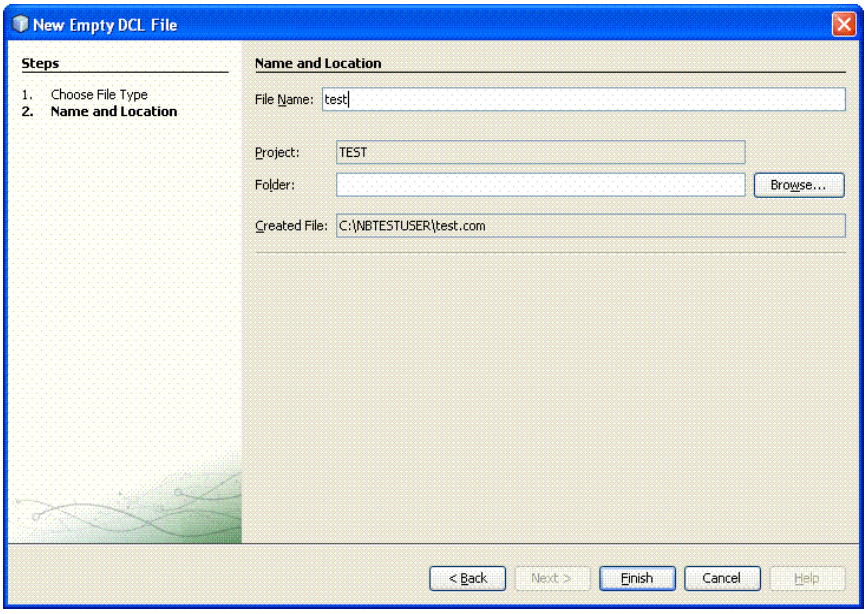


```
Finished (Test.c compilation)
```

## 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**.



File Name: test

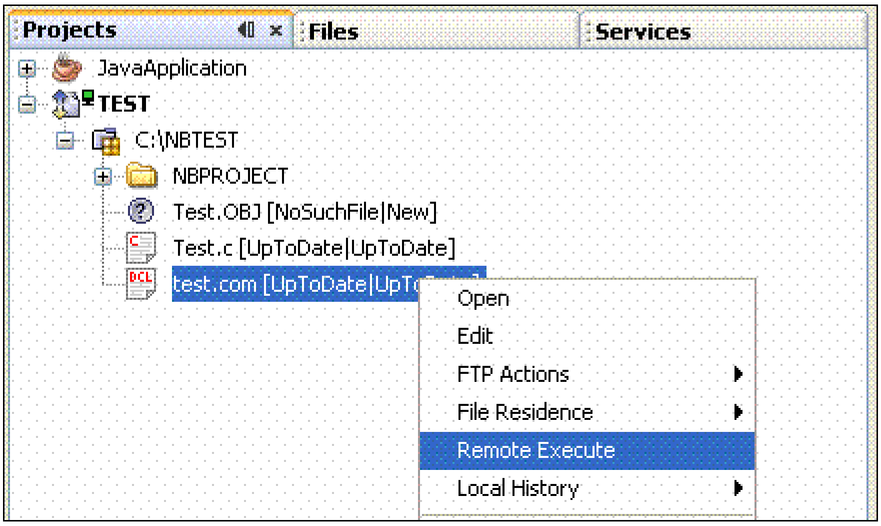
Project: TEST

Folder:

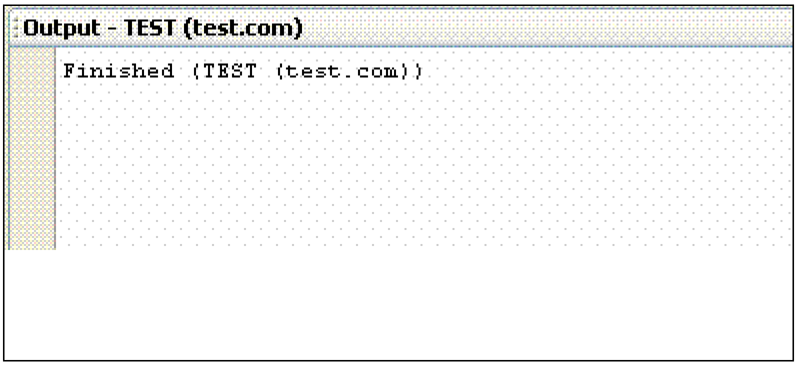
Created File: C:\NBTESTUSER\test.com

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**.

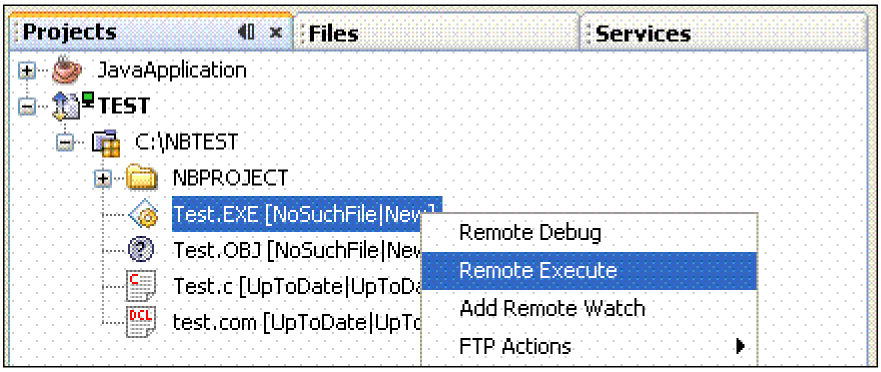


After DCL execution, an executable image (EXE) is created and the following message is displayed in the NetBeans IDE **Output** window.

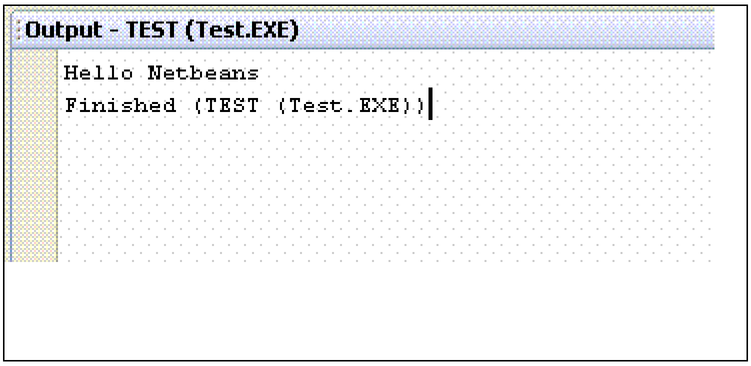


```
Finished (TEST (test.com))
```

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



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



```
Hello Netbeans
Finished (TEST (Test.EXE))
```

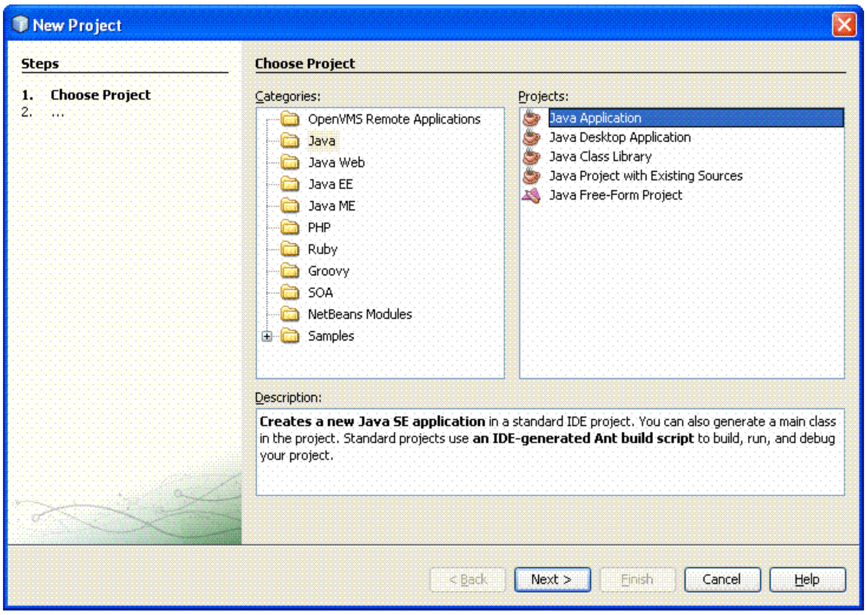
## 2.3 Creating a Java Project

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**.



New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- OpenVMS Remote Applications
- Java
- Java Web
- Java EE
- Java ME
- PHP
- Ruby
- Groovy
- SOA
- NetBeans Modules
- Samples

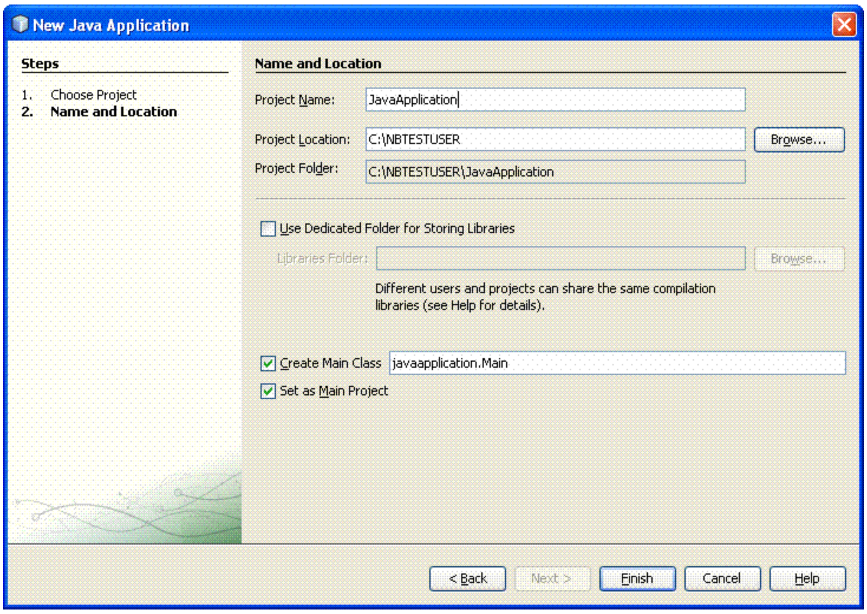
Projects:

- Java Application
- Java Desktop Application
- Java Class Library
- Java Project with Existing Sources
- Java Free-Form Project

Description:

**Creates a new Java SE application** in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

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.



New Java Application

Steps

1. Choose Project
2. Name and Location

Name and Location

Project Name: JavaApplication

Project Location: C:\NBTESTUSER

Project Folder: C:\NBTESTUSER\JavaApplication

□ Use Dedicated Folder for Storing Libraries

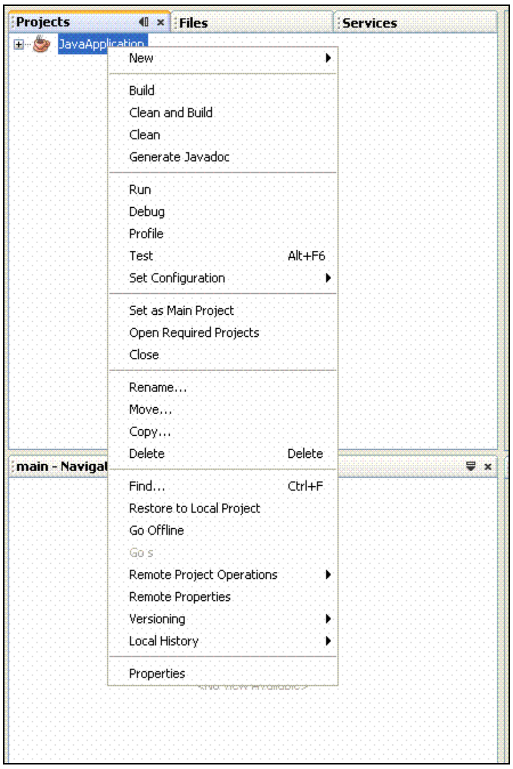
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

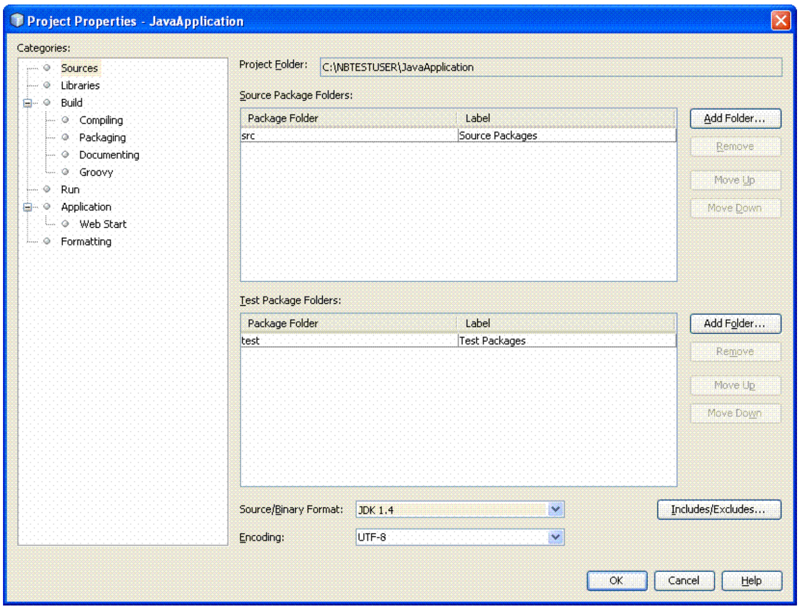
☑ Create Main Class javaapplication.Main

☑ Set as Main Project

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



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



Project Folder: C:\NBTESTUSER\JavaApplication

Source Package Folders:

| Package Folder | Label |
|---|---|
| src | Source Packages |

Test Package Folders:

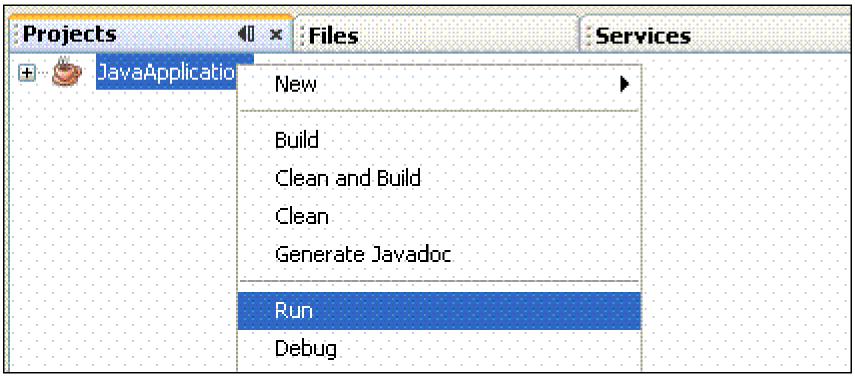
| Package Folder | Label |
|---|---|
| test | Test Packages |

Source/Binary Format: JDK 1.4

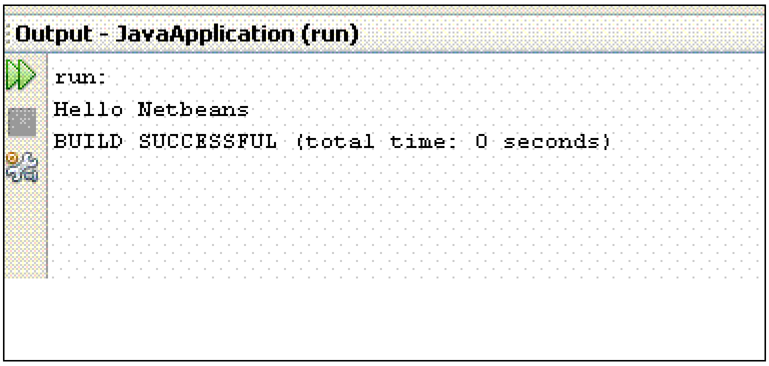
Encoding: UTF-8

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.



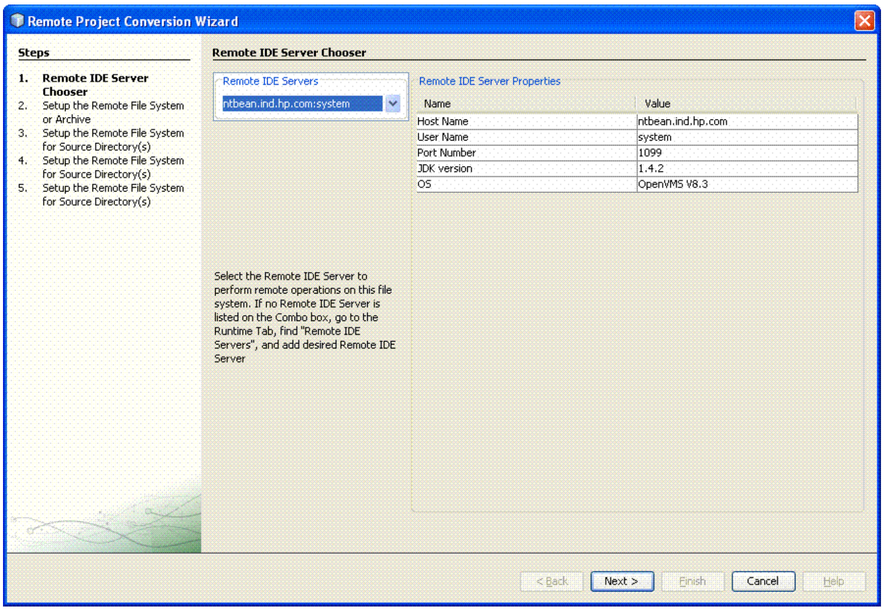
The NetBeans IDE **Output** window is displayed.



Output - JavaApplication (run)

```
run:
Hello Netbeans
BUILD SUCCESSFUL (total time: 0 seconds)
```

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**.



Remote Project Conversion Wizard

Steps

1. **Remote IDE Server Chooser**
2. Setup the Remote File System or Archive
3. Setup the Remote File System for Source Directory(s)
4. Setup the Remote File System for Source Directory(s)
5. Setup the Remote File System for Source Directory(s)

Remote IDE Server Chooser

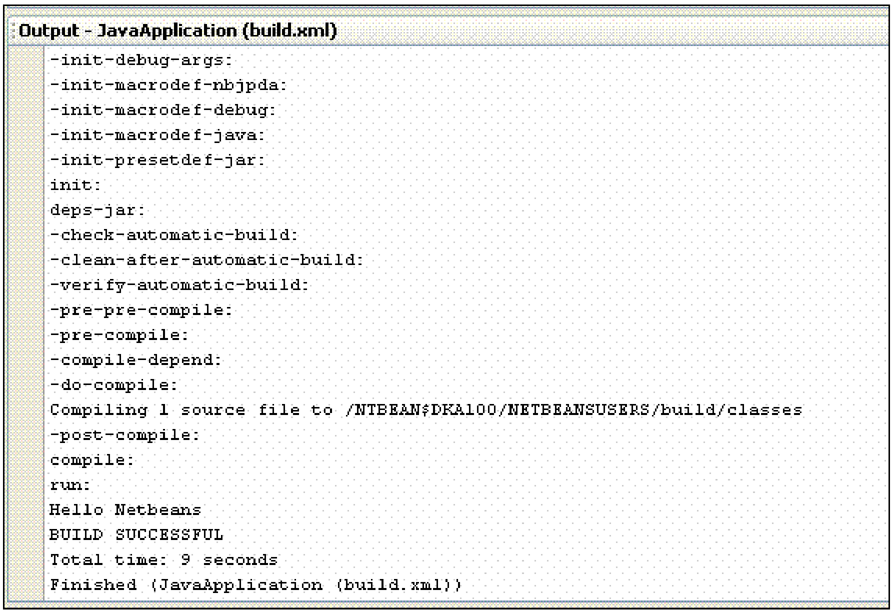
Remote IDE Servers: ntbean.ind.hp.com:system

Remote IDE Server Properties

| Name | Value |
|---|---|
| Host Name | ntbean.ind.hp.com |
| User Name | system |
| Port Number | 1099 |
| JDK version | 1.4.2 |
| OS | OpenVMS V8.3 |

Select the Remote IDE Server to perform remote operations on this file system. If no Remote IDE Server is listed on the Combo box, go to the Runtime Tab, find "Remote IDE Servers", and add desired Remote IDE Server

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.



Output - JavaApplication (build.xml)

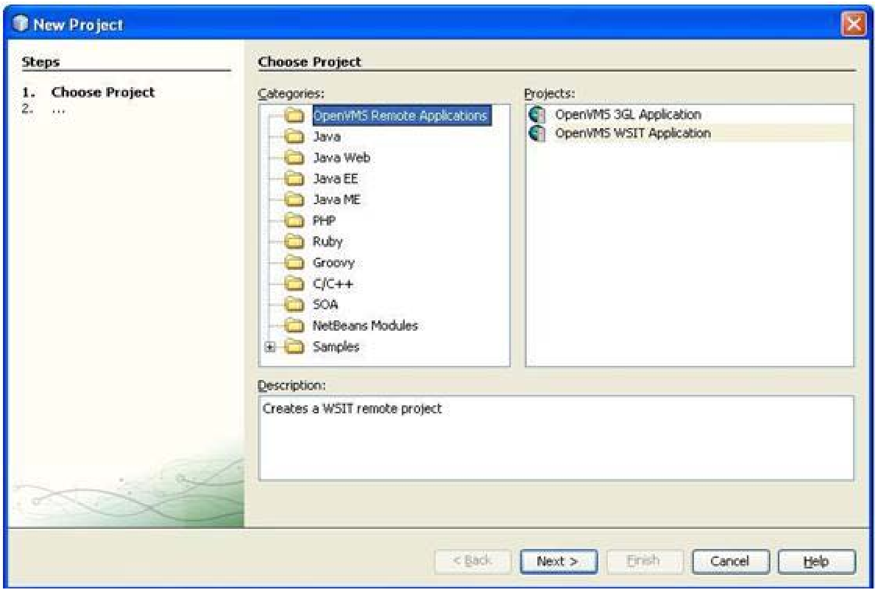
```
-init-debug-args:
-init-macrodef-nbjpda:
-init-macrodef-debug:
-init-macrodef-java:
-init-presetdef-jar:
init:
deps-jar:
-check-automatic-build:
-clean-after-automatic-build:
-verify-automatic-build:
-pre-pre-compile:
-pre-compile:
-compile-depend:
-do-compile:
Compiling 1 source file to /NTBEAN$DKA100/NETBEANSUSERS/build/classes
-post-compile:
compile:
run:
Hello Netbeans
BUILD SUCCESSFUL
Total time: 9 seconds
Finished (JavaApplication (build.xml))
```

## 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.

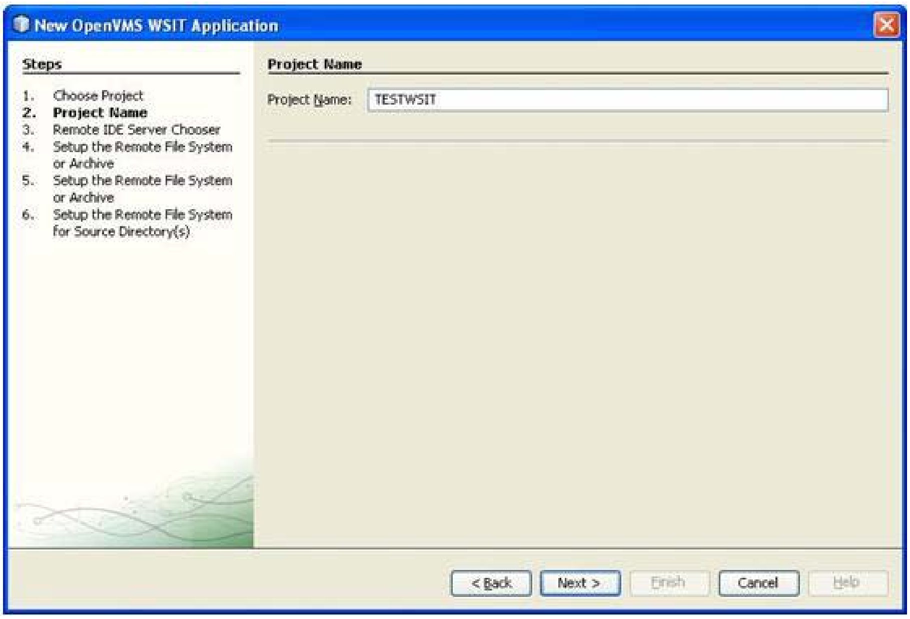
To create an OpenVMS WSIT project from the Distributed NetBeans client, follow these steps:

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**.



Creates a WSIT remote project

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



New OpenVMS WSIT Application

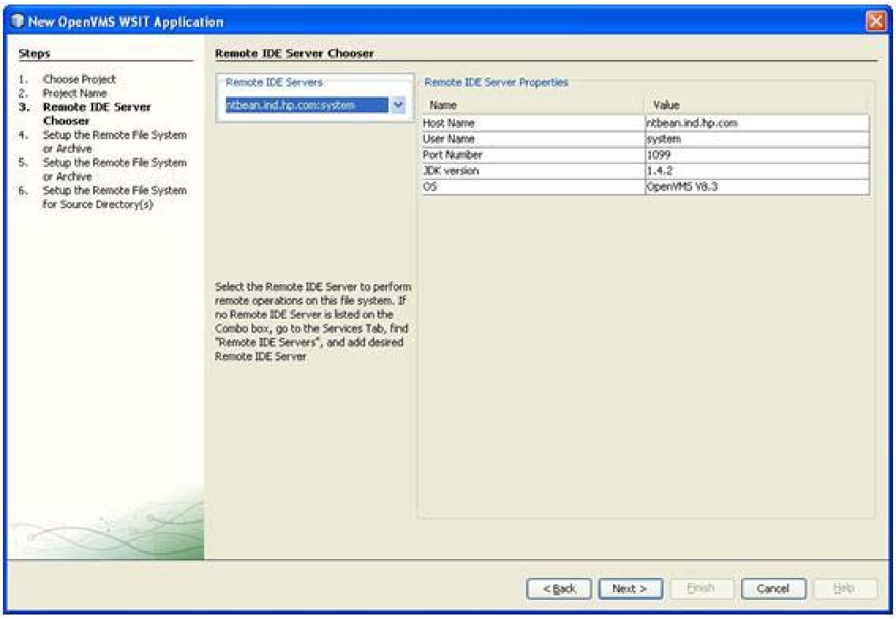
Steps

1. Choose Project
2. **Project Name**
3. Remote IDE Server Chooser
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

**Project Name**

Project Name: TESTWSIT

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.



New OpenVMS WSIT Application

Steps

1. Choose Project
2. Project Name
3. **Remote IDE Server Chooser**
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

**Remote IDE Server Chooser**

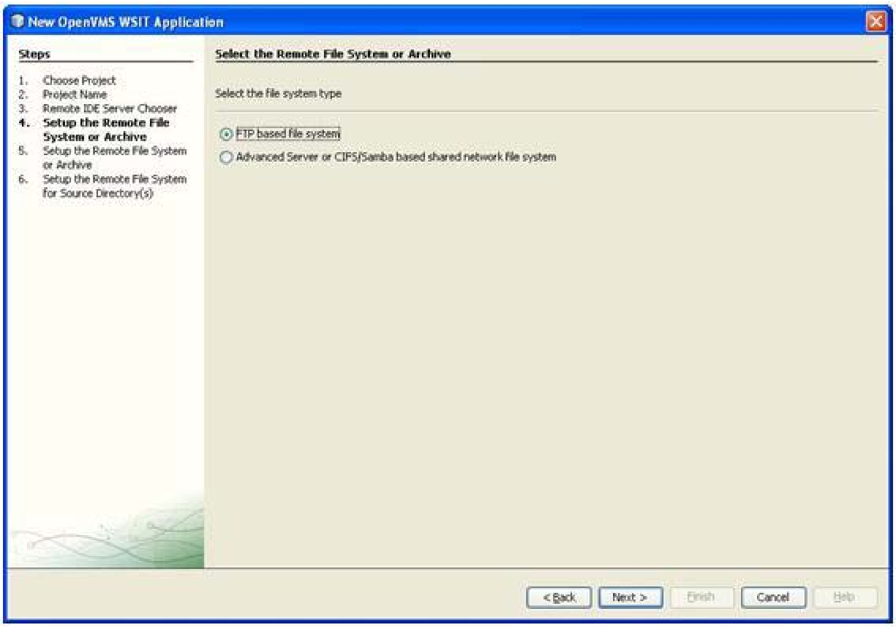
Remote IDE Servers: ntbean.ind.hp.com:system

Remote IDE Server Properties

| Name | Value |
|---|---|
| Host Name | ntbean.ind.hp.com |
| User Name | system |
| Port Number | 1099 |
| JDK version | 1.4.2 |
| OS | OpenVMS V8.3 |

Select the Remote IDE Server to perform remote operations on this file system. If no Remote IDE Server is listed on the Combo box, go to the Services Tab, find "Remote IDE Servers", and add desired Remote IDE Server

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


New OpenVMS WSIT Application

Steps

1. Choose Project
2. Project Name
3. Remote IDE Server Chooser
4. **Setup the Remote File System or Archive**
5. Setup the Remote File System or Archive
6. Setup the Remote File System for Source Directory(s)

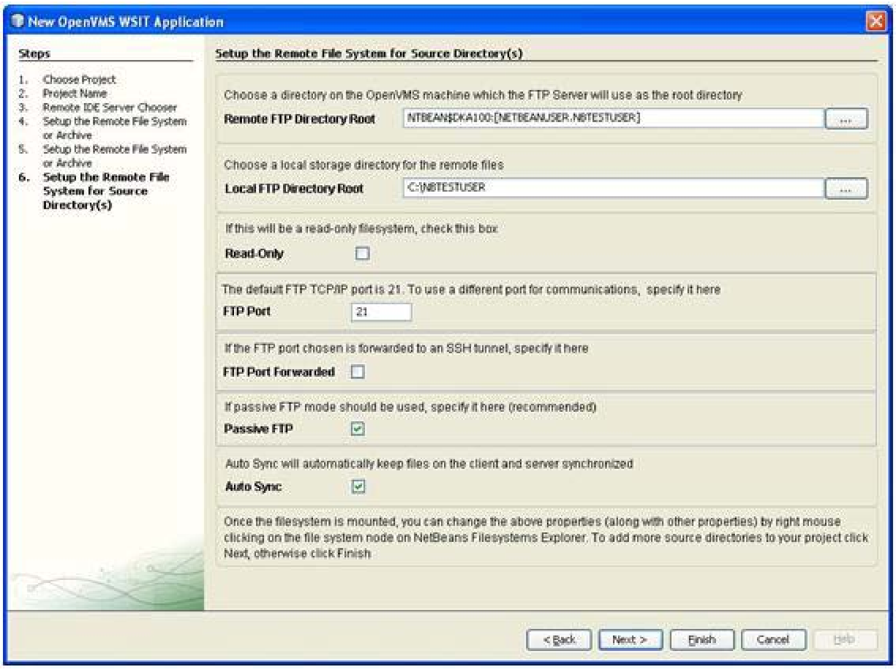
**Select the Remote File System or Archive**

Select the file system type

● FTP based file system

○ Advanced Server or CIFS/Samba based shared network file system

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.


New OpenVMS WSIT Application

Steps

1. Choose Project
2. Project Name
3. Remote IDE Server Chooser
4. Setup the Remote File System or Archive
5. Setup the Remote File System or Archive
6. **Setup the Remote File System for Source Directory(s)**

**Setup the Remote File System for Source Directory(s)**

Choose a directory on the OpenVMS machine which the FTP Server will use as the root directory

**Remote FTP Directory Root** NTBEAN$DKA100:[NETBEANUSER.NBTESTUSER]

Choose a local storage directory for the remote files

**Local FTP Directory Root** C:\NBTESTUSER

If this will be a read-only filesystem, check this box

**Read-Only** □

The default FTP TCP/IP port is 21. To use a different port for communications, specify it here

**FTP Port** 21

If the FTP port chosen is forwarded to an SSH tunnel, specify it here

**FTP Port Forwarded** □

If passive FTP mode should be used, specify it here (recommended)

**Passive FTP** ☑

Auto Sync will automatically keep files on the client and server synchronized

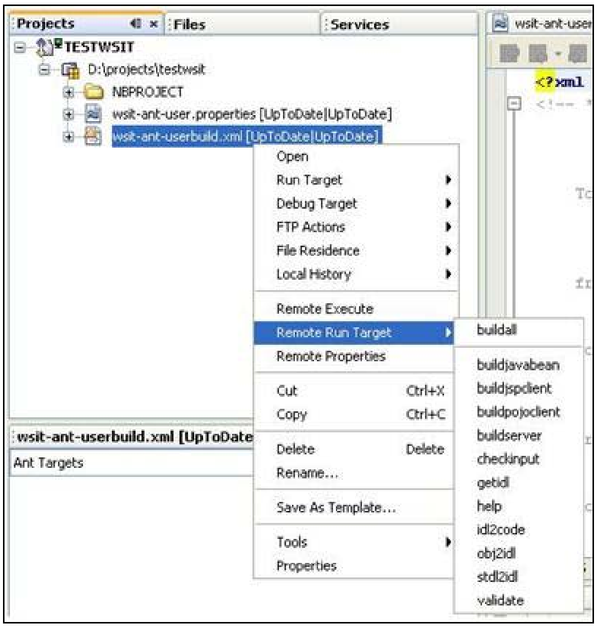
**Auto Sync** ☑

Once the filesystem is mounted, you can change the above properties (along with other properties) by right mouse clicking on the file system node on NetBeans Filesystems Explorer. To add more source directories to your project click Next, otherwise click Finish

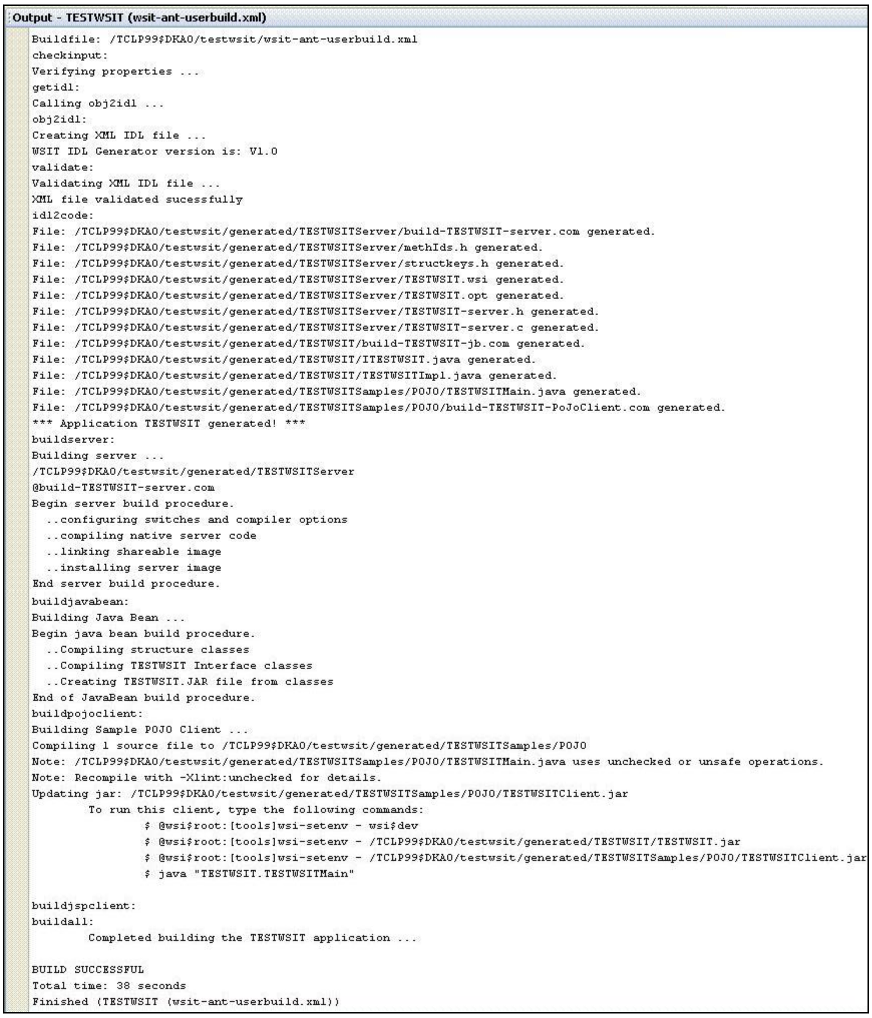
*8.* The WSIT project window is displayed. To create a new C source file, see Section 2.2.1, *Creating a New C Source File*

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.

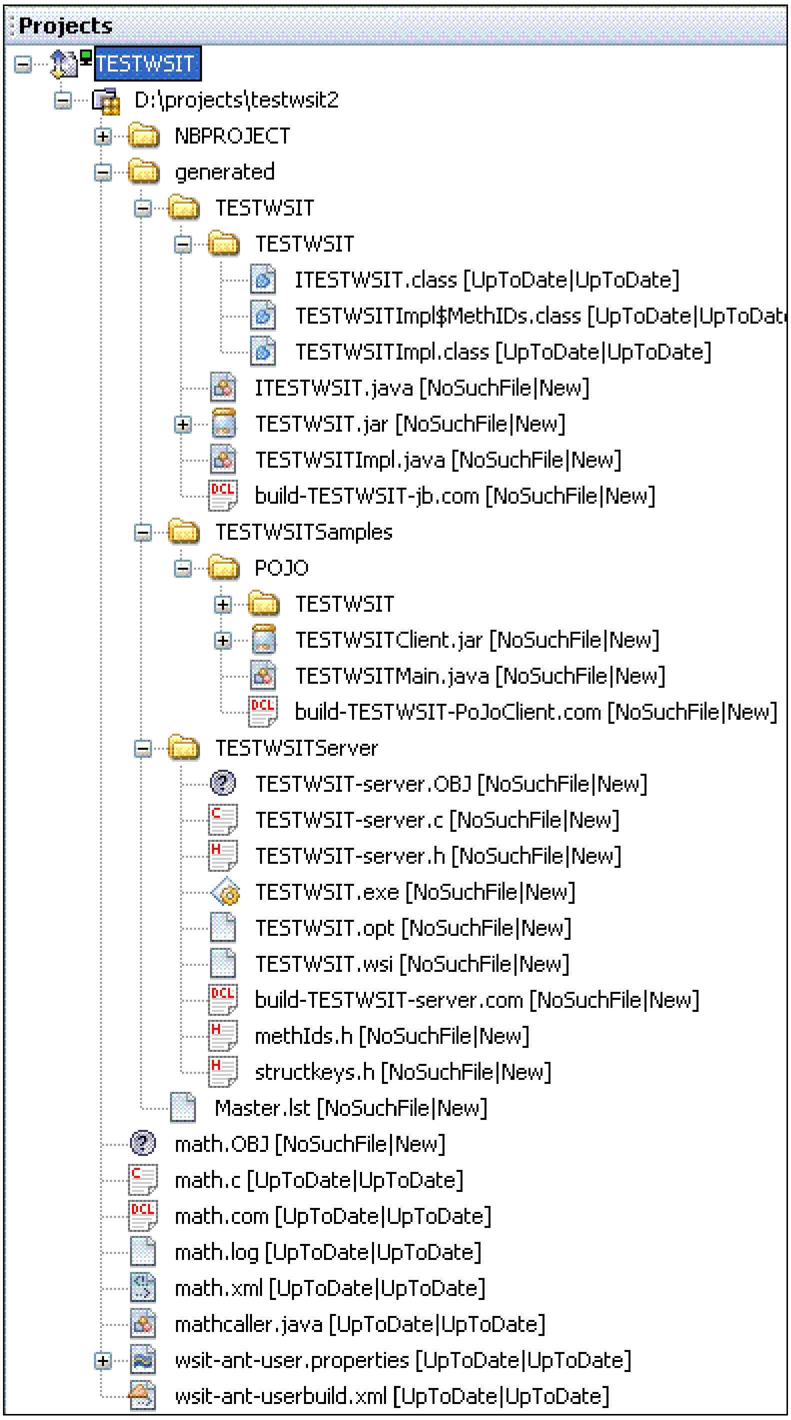


```
Buildfile: /TCLP99$DKA0/testwsit/wsit-ant-userbuild.xml
checkinput:
Verifying properties ...
getidl:
Calling obj2idl ...
obj2idl:
Creating XML IDL file ...
WSIT IDL Generator version is: V1.0
validate:
Validating XML IDL file ...
XML file validated sucessfully
idl2code:
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/build-TESTWSIT-server.com generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/methIds.h generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/structkeys.h generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/TESTWSIT.wsi generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/TESTWSIT.opt generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/TESTWSIT-server.h generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITServer/TESTWSIT-server.c generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSIT/build-TESTWSIT-jb.com generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSIT/ITESTWSIT.java generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSIT/TESTWSITImpl.java generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO/TESTWSITMain.java generated.
File: /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO/build-TESTWSIT-PoJoClient.com generated.
*** Application TESTWSIT generated! ***
buildserver:
Building server ...
/TCLP99$DKA0/testwsit/generated/TESTWSITServer
@build-TESTWSIT-server.com
Begin server build procedure.
  ..configuring switches and compiler options
  ..compiling native server code
  ..linking shareable image
  ..installing server image
End server build procedure.
buildjavabean:
Building Java Bean ...
Begin java bean build procedure.
  ..Compiling structure classes
  ..Compiling TESTWSIT Interface classes
  ..Creating TESTWSIT.JAR file from classes
End of JavaBean build procedure.
buildpojoclient:
Building Sample POJO Client ...
Compiling 1 source file to /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO
Note: /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO/TESTWSITMain.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Updating jar: /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO/TESTWSITClient.jar
        To run this client, type the following commands:
                $ @wsi$root:[tools]wsi-setenv - wsi$dev
                $ @wsi$root:[tools]wsi-setenv - /TCLP99$DKA0/testwsit/generated/TESTWSIT/TESTWSIT.jar
                $ @wsi$root:[tools]wsi-setenv - /TCLP99$DKA0/testwsit/generated/TESTWSITSamples/POJO/TESTWSITClient.jar
                $ java "TESTWSIT.TESTWSITMain"

buildjspclient:
buildall:
        Completed building the TESTWSIT application ...

BUILD SUCCESSFUL
Total time: 38 seconds
Finished (TESTWSIT (wsit-ant-userbuild.xml))
```

The generated files are listed in the directory as shown:



```
Projects
TESTWSIT
  D:\projects\testwsit2
    NBPROJECT
    generated
      TESTWSIT
        TESTWSIT
          ITESTWSIT.class [UpToDate|UpToDate]
          TESTWSITImpl$MethIDs.class [UpToDate|UpToDat
          TESTWSITImpl.class [UpToDate|UpToDate]
        ITESTWSIT.java [NoSuchFile|New]
        TESTWSIT.jar [NoSuchFile|New]
        TESTWSITImpl.java [NoSuchFile|New]
        build-TESTWSIT-jb.com [NoSuchFile|New]
      TESTWSITSamples
        POJO
          TESTWSIT
          TESTWSITClient.jar [NoSuchFile|New]
          TESTWSITMain.java [NoSuchFile|New]
          build-TESTWSIT-PoJoClient.com [NoSuchFile|New]
      TESTWSITServer
        TESTWSIT-server.OBJ [NoSuchFile|New]
        TESTWSIT-server.c [NoSuchFile|New]
        TESTWSIT-server.h [NoSuchFile|New]
        TESTWSIT.exe [NoSuchFile|New]
        TESTWSIT.opt [NoSuchFile|New]
        TESTWSIT.wsi [NoSuchFile|New]
        build-TESTWSIT-server.com [NoSuchFile|New]
        methIds.h [NoSuchFile|New]
        structkeys.h [NoSuchFile|New]
      Master.lst [NoSuchFile|New]
    math.OBJ [NoSuchFile|New]
    math.c [UpToDate|UpToDate]
    math.com [UpToDate|UpToDate]
    math.log [UpToDate|UpToDate]
    math.xml [UpToDate|UpToDate]
    mathcaller.java [UpToDate|UpToDate]
    wsit-ant-user.properties [UpToDate|UpToDate]
    wsit-ant-userbuild.xml [UpToDate|UpToDate]
```

## 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

To access the sourceforge website and download the open source code for Distributed NetBeans 7.0.1, click here.