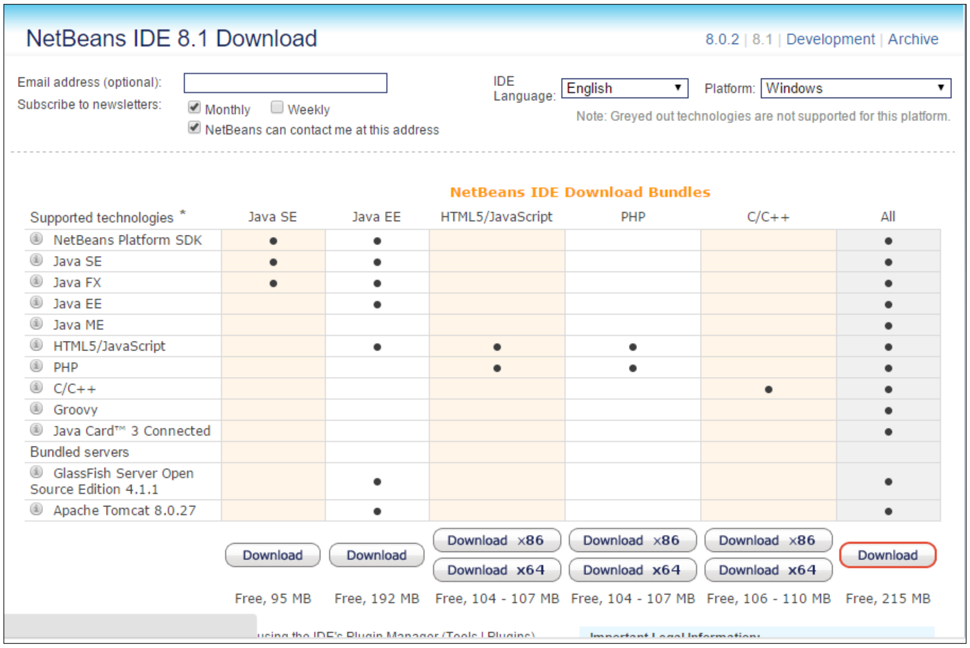
A
Setting up Netbeans Environment

This appendix shows a step-by-step procedure for the reader to set up the development environment for the Netbeans IDE.

Downloading and installing Netbeans

> Before downloading and installing Netbeans, make sure you have installed **Java Development Kit** (**JDK**), which can be downloaded from `https://www.oracle.com/java/index.html`.

Netbeans can be freely downloaded at the project's site `https://netbeans.org/downloads/index.html` (as shown in the following screenshot); choose the installer to download according your operation system, as shown by the dropdown list **Platform**:



NetBeans IDE 8.1 Download

8.0.2 | 8.1 | Development | Archive

Email address (optional):

Subscribe to newsletters: ☑ Monthly ☐ Weekly

☑ NetBeans can contact me at this address

IDE Language: English

Platform: Windows

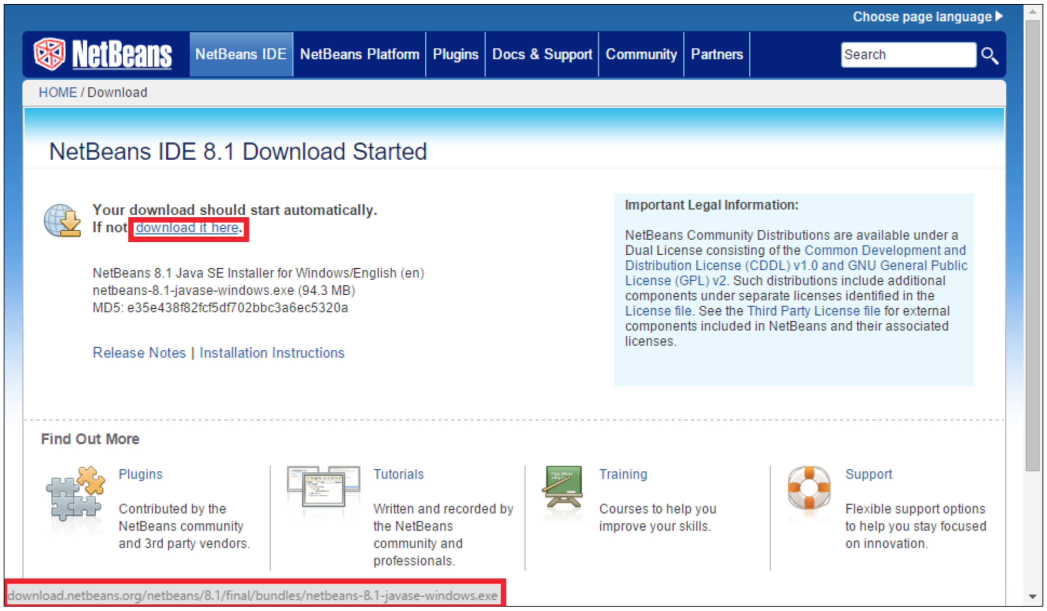
Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	●	●				●
Java SE	●	●				●
Java FX	●	●				●
Java EE		●				●
Java ME						●
HTML5/JavaScript		●	●	●		●
PHP			●	●		●
C/C++					●	●
Groovy						●
Java Card™ 3 Connected						●
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		●				●
Apache Tomcat 8.0.27		●				●
	Download	Download	Download x86 Download x64	Download x86 Download x64	Download x86 Download x64	Download
	Free, 95 MB	Free, 192 MB	Free, 104 - 107 MB	Free, 104 - 107 MB	Free, 106 - 110 MB	Free, 215 MB

using the IDE's Plugin Manager (Tools | Plugins) Important Legal Information:

You can select from the versions displayed in the web page. For the projects in this book, the Java SE version fits very well, in addition to be the smallest and lightest. The download should start shortly, in every case you can click the button **download it here**, as shown in the following screenshot:



NetBeans IDE 8.1 Download Started

Your download should start automatically.
If not download it here.

NetBeans 8.1 Java SE Installer for Windows/English (en)
netbeans-8.1-javase-windows.exe (94.3 MB)
MD5: e35e438f82fcf5df702bbc3a6ec5320a

Release Notes | Installation Instructions

Important Legal Information:

NetBeans Community Distributions are available under a Dual License consisting of the Common Development and Distribution License (CDDL) v1.0 and GNU General Public License (GPL) v2. Such distributions include additional components under separate licenses identified in the License file. See the Third Party License file for external components included in NetBeans and their associated licenses.

Find Out More

Plugins
Contributed by the NetBeans community and 3rd party vendors.

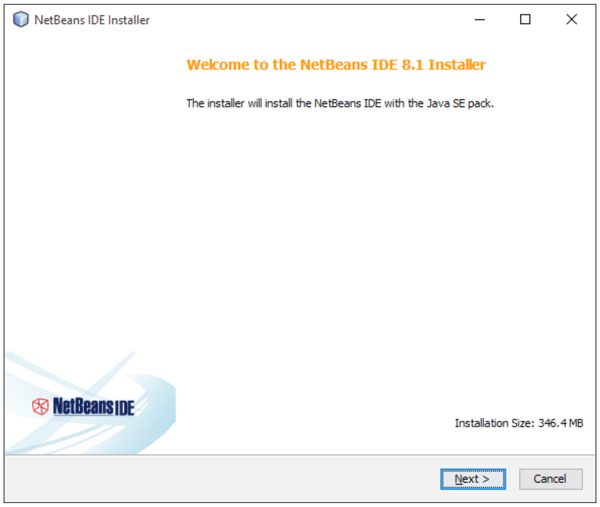
Tutorials
Written and recorded by the NetBeans community and professionals.

Training
Courses to help you improve your skills.

Support
Flexible support options to help you stay focused on innovation.

download.netbeans.org/netbeans/8.1/final/bundles/netbeans-8.1-javase-windows.exe

After download, you should run the executable (`netbeans-<version>-javase-<your_os>.exe`). The following screen appears. You can perform the standard installation. When executing the installer, it tells you the size of the installation and the version. You can click the **Next** button.

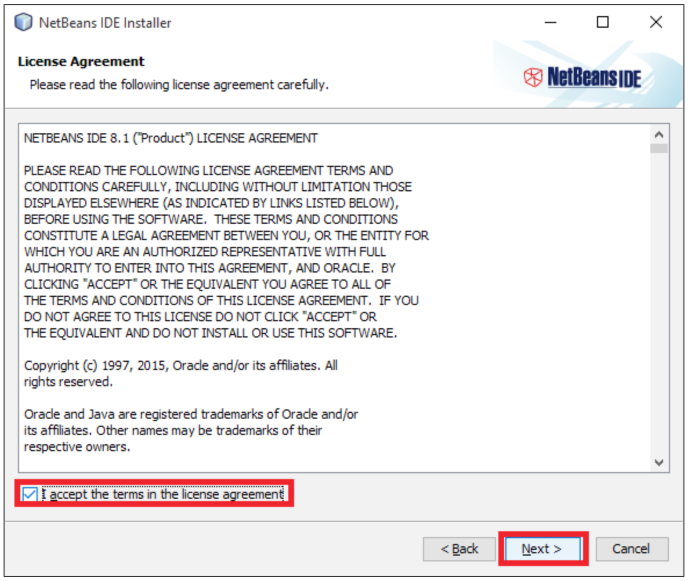


Welcome to the NetBeans IDE 8.1 Installer

The installer will install the NetBeans IDE with the Java SE pack.

Installation Size: 346.4 MB

You must accept the licence agreement by marking the check box, as shown in the following screenshot, then clicking on the **Next** button:



License Agreement

Please read the following license agreement carefully.

NETBEANS IDE 8.1 ("Product") LICENSE AGREEMENT

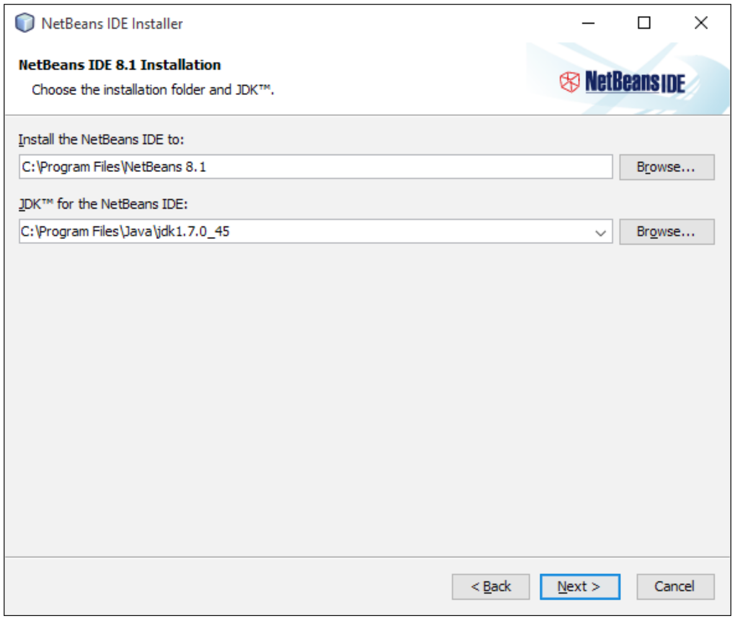
PLEASE READ THE FOLLOWING LICENSE AGREEMENT TERMS AND CONDITIONS CAREFULLY, INCLUDING WITHOUT LIMITATION THOSE DISPLAYED ELSEWHERE (AS INDICATED BY LINKS LISTED BELOW), BEFORE USING THE SOFTWARE. THESE TERMS AND CONDITIONS CONSTITUTE A LEGAL AGREEMENT BETWEEN YOU, OR THE ENTITY FOR WHICH YOU ARE AN AUTHORIZED REPRESENTATIVE WITH FULL AUTHORITY TO ENTER INTO THIS AGREEMENT, AND ORACLE. BY CLICKING "ACCEPT" OR THE EQUIVALENT YOU AGREE TO ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THIS LICENSE DO NOT CLICK "ACCEPT" OR THE EQUIVALENT AND DO NOT INSTALL OR USE THIS SOFTWARE.

Copyright (c) 1997, 2015, Oracle and/or its affiliates. All rights reserved.

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

☑ I accept the terms in the license agreement

Then, you can select the folders where the program will be installed on, or you can just leave the default folders, as shown in the following screenshot:



NetBeans IDE 8.1 Installation

Choose the installation folder and JDK™.

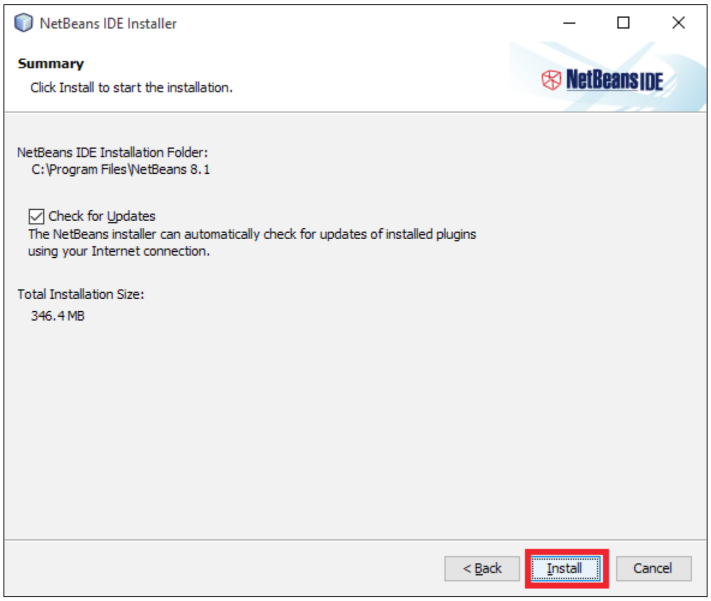
Install the NetBeans IDE to:

C:\Program Files\NetBeans 8.1

JDK™ for the NetBeans IDE:

C:\Program Files\Java\jdk1.7.0_45

Now we're ready to install. You may leave the check box **Check for Updates** marked and just click the **Install** button:



NetBeans IDE Installer

Summary

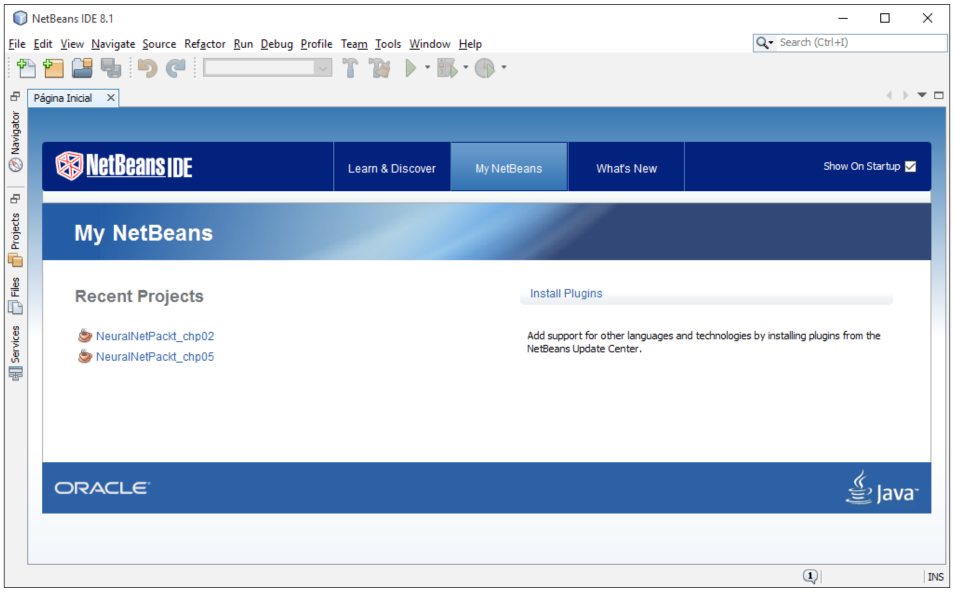
Click Install to start the installation.

NetBeans IDE Installation Folder:
C:\Program Files\NetBeans 8.1

☑ Check for Updates
The NetBeans installer can automatically check for updates of installed plugins using your Internet connection.

Total Installation Size:
346.4 MB

After the installation, you may click on the **Netbeans** icon at the desktop to run Netbeans. An initial page will open:



NetBeans IDE 8.1

My NetBeans

Recent Projects

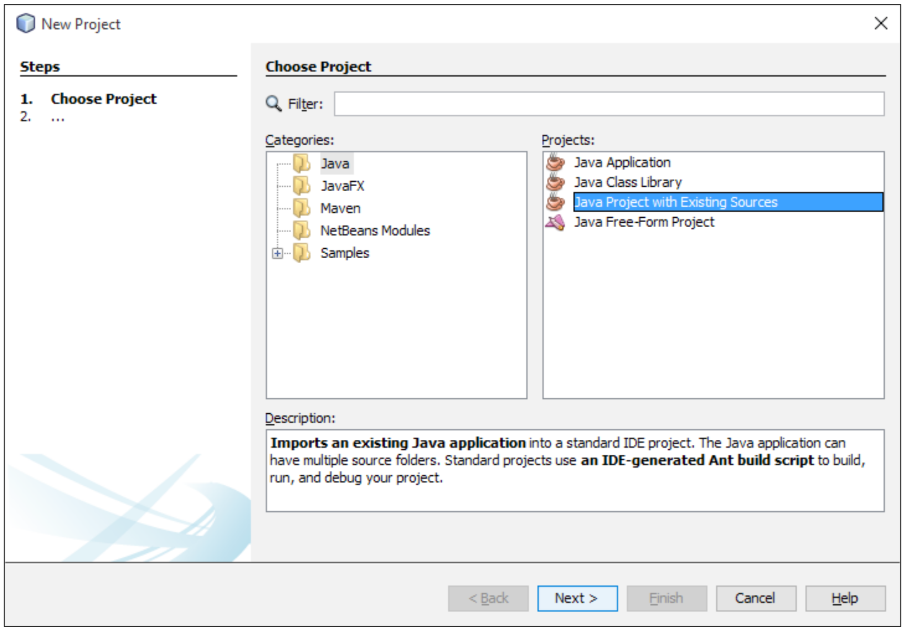
NeuralNetPackt_chp02
NeuralNetPackt_chp05

Install Plugins

Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

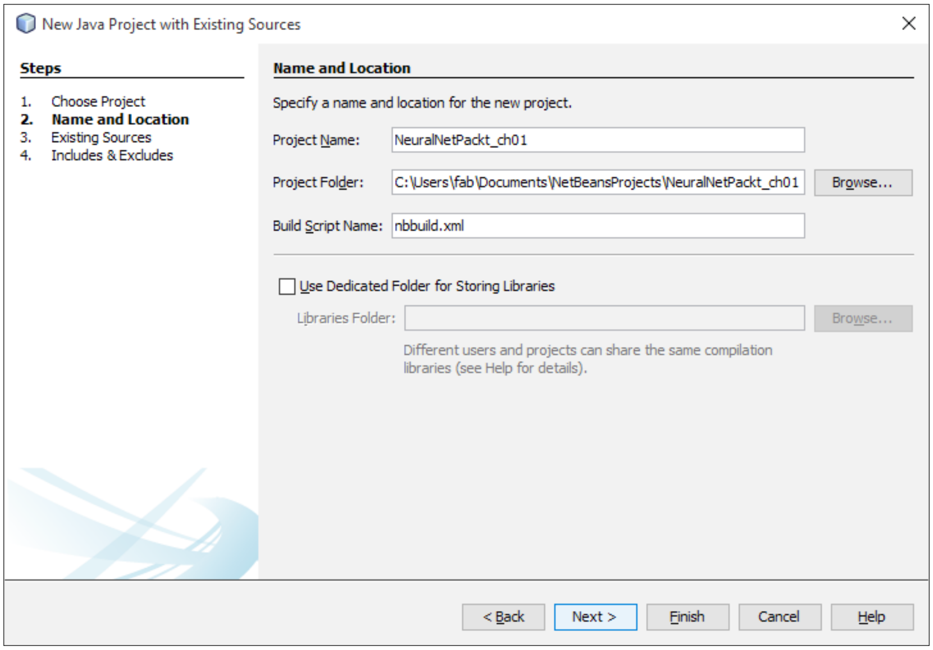
Setting up Netbeans environment

The Netbeans environment already provides options for creating and opening new projects. Now let's create a new project by selecting the menu `File |New Project`. In the dialog window that opens, make sure you selected the project template **Java Project with Existing Sources**, and then click **Next**:



Imports an existing Java application into a standard IDE project. The Java application can have multiple source folders. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Then, you can choose a name for the project, the name `NeuralNetPackt_ch01` is just a mere suggestion, you are free to choose the name you want.



New Java Project with Existing Sources

Steps

1. Choose Project
2. **Name and Location**
3. Existing Sources
4. Includes & Excludes

Name and Location

Specify a name and location for the new project.

Project Name: NeuralNetPackt_ch01

Project Folder: C:\Users\fab\Documents\NetBeansProjects\NeuralNetPackt_ch01

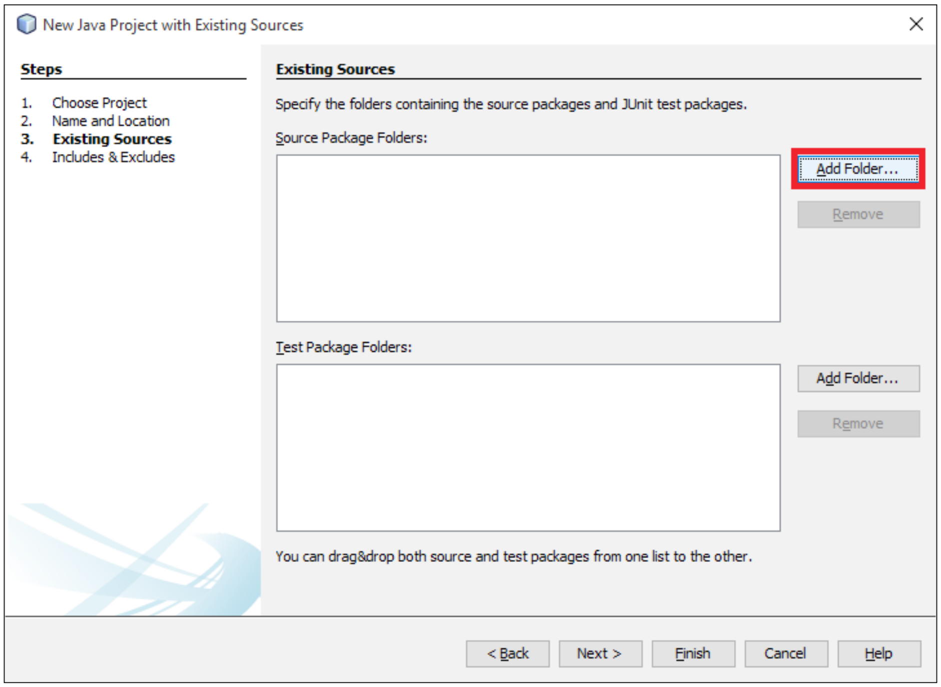
Build Script Name: nbbuild.xml

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

In the next screen you can select the folder where the source codes are stored:



New Java Project with Existing Sources

Steps

1. Choose Project
2. Name and Location
3. **Existing Sources**
4. Includes & Excludes

Existing Sources

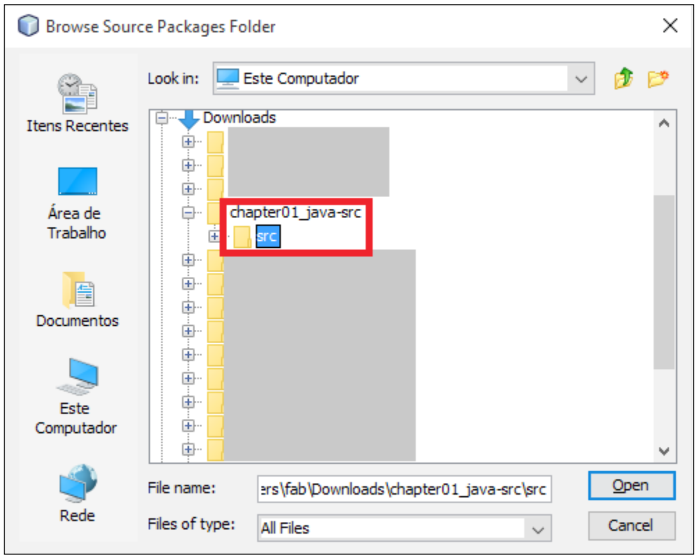
Specify the folders containing the source packages and JUnit test packages.

Source Package Folders:

Test Package Folders:

You can drag&drop both source and test packages from one list to the other.

In the file open dialog that opens, browse to the folder where the files are stored and select it:



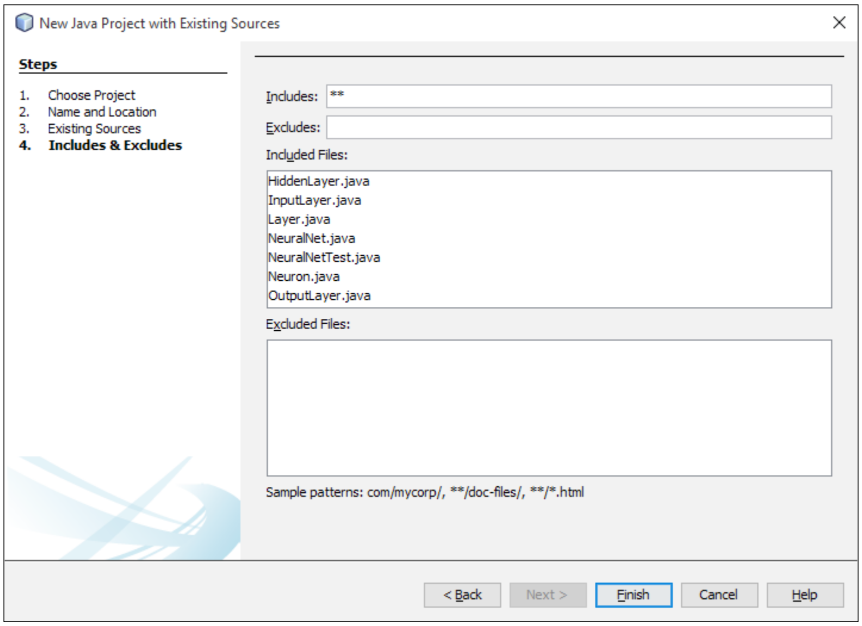
Browse Source Packages Folder

Look in: Este Computador

File name: ers\fab\Downloads\chapter01_java-src\src

Files of type: All Files

Once you've selected the folder, you can click the **Open** button, and then the next button in the parent window. Now a list of includes and excludes is displayed. You can just leave as is and click the **Finish** button:



New Java Project with Existing Sources

Steps

1. Choose Project
2. Name and Location
3. Existing Sources
4. **Includes & Excludes**

Includes: **

Excludes:

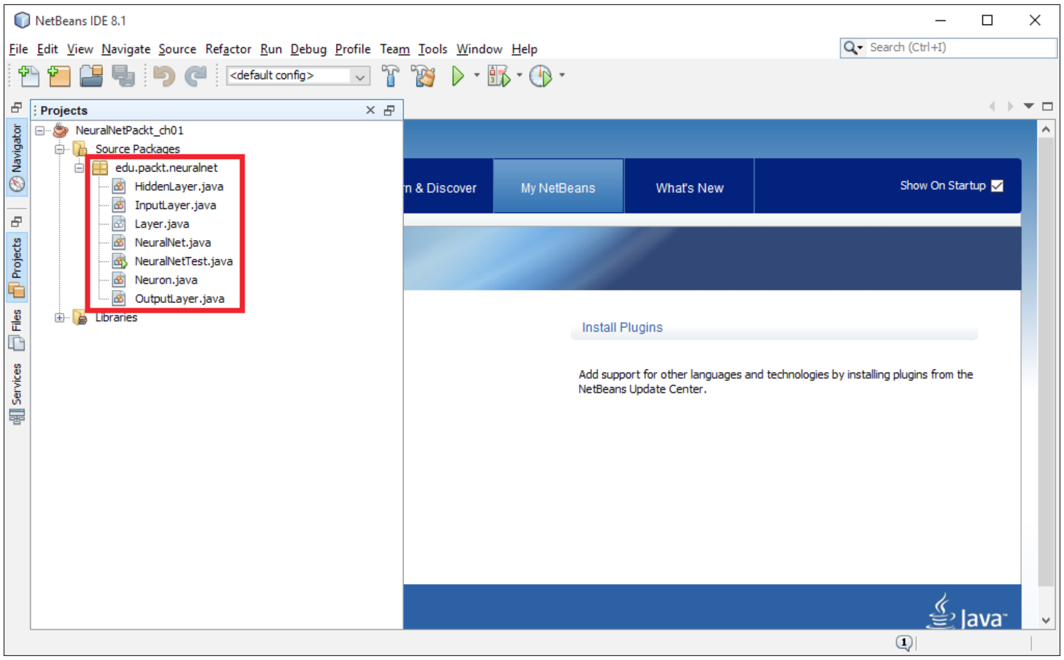
Included Files:

HiddenLayer.java
InputLayer.java
Layer.java
NeuralNet.java
NeuralNetTest.java
Neuron.java
OutputLayer.java

Excluded Files:

Sample patterns: com/mycorp/, **/doc-files/, **/*.html

And we're done! Now you are ready to work on the codes of each chapter in your Netbeans installation:

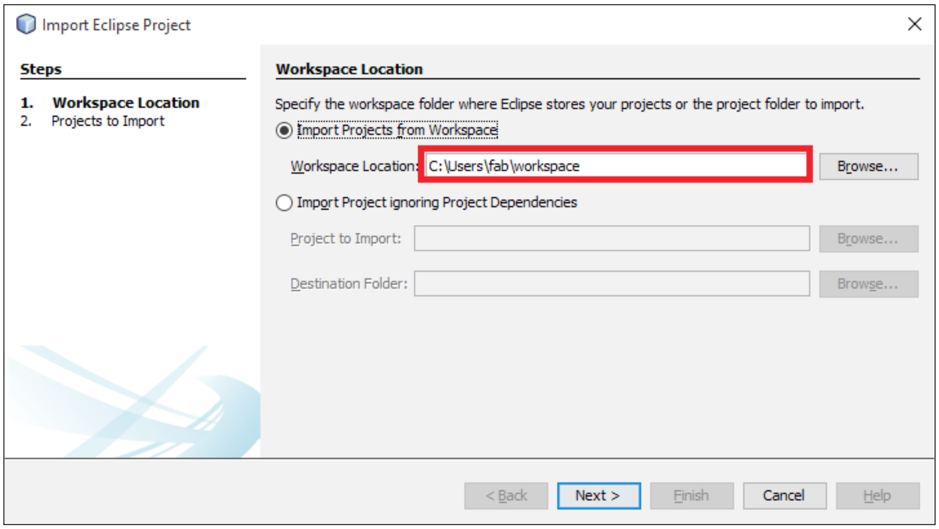


Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

Importing a project

Netbeans offers an option to import an existing project, whether created on Netbeans or Eclipse. You can go to the menu `File | Import Project` and select the appropriate option.

If you already have Eclipse installed and you want to import it into Netbeans, just select the folder for the workspace location, and click **OK**:



Import Eclipse Project

Steps

1. **Workspace Location**
2. Projects to Import

Workspace Location

Specify the workspace folder where Eclipse stores your projects or the project folder to import.

● Import Projects from Workspace

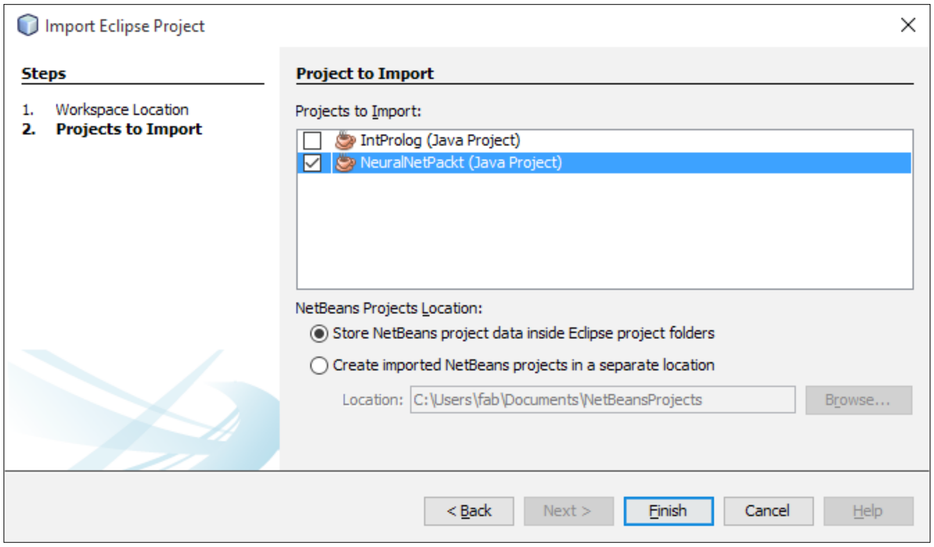
Workspace Location: C:\Users\fab\workspace

○ Import Project ignoring Project Dependencies

Project to Import:

Destination Folder:

Select the project you want to import and click **Finish**:



Import Eclipse Project

Steps

1. Workspace Location
2. **Projects to Import**

Project to Import

Projects to Import:

□ IntProlog (Java Project)
☑ NeuralNetPackt (Java Project)

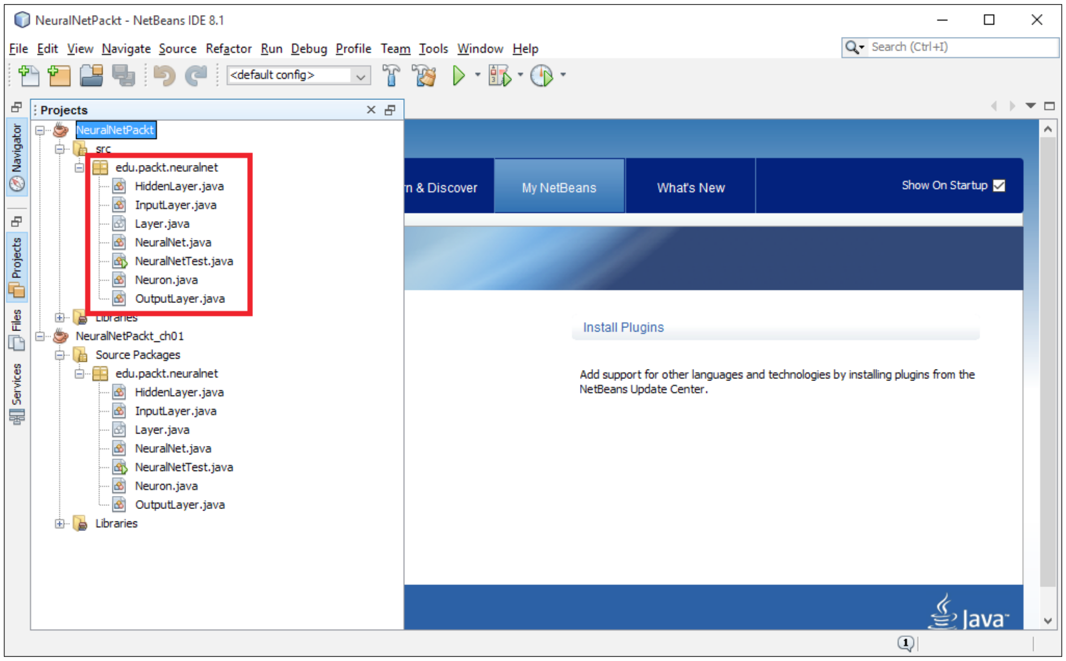
NetBeans Projects Location:

● Store NetBeans project data inside Eclipse project folders

○ Create imported NetBeans projects in a separate location

Location: C:\Users\fab\Documents\NetBeansProjects

And the project is imported successfully:



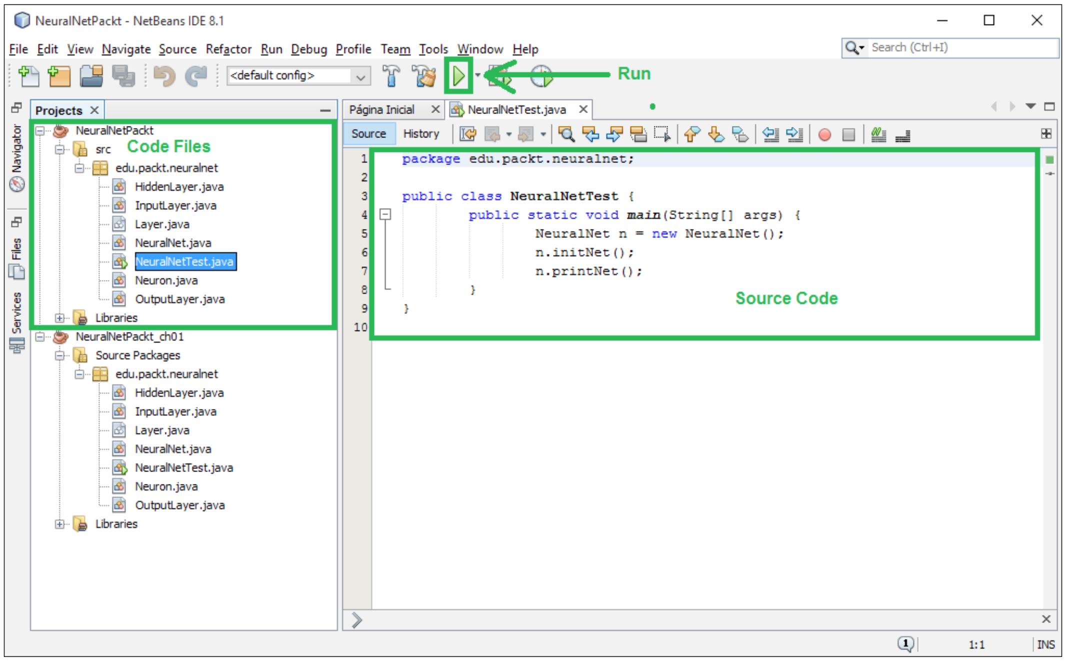
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

If you want to import from a zip file, you can choose the option **From Zip** from the menu `File | Import Project`. Just make sure the zip file was created from the Netbeans **Projects**.

Programming and running code with Netbeans

We recommend to run the IDE as an administrator, but it is not necessary.

After all previous steps, you are able to start Java programming. The next screenshot shows the structure of the Netbeans environment:

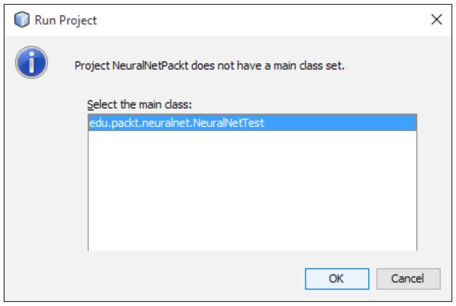


```
package edu.packt.neuralnet;

public class NeuralNetTest {
        public static void main(String[] args) {
                NeuralNet n = new NeuralNet();
                n.initNet();
                n.printNet();
        }
}
```

- **Projects**: This section is displayed in the left of the packages and classes that compose the Java project.
- **Code File**: It is shown in the middle of the screen and brings the code you should interact.
- **Run**: To run the code, as displayed in the button indicated in the screen.
- **Debug**: To debug the code, select the menu **Debug** and then choose a file to debug (or by clicking *Ctrl+Shift+F5*).

Debugging with Netbeans

To debug a Java program in Netbeans, you just select a project to debug or the class file itself:

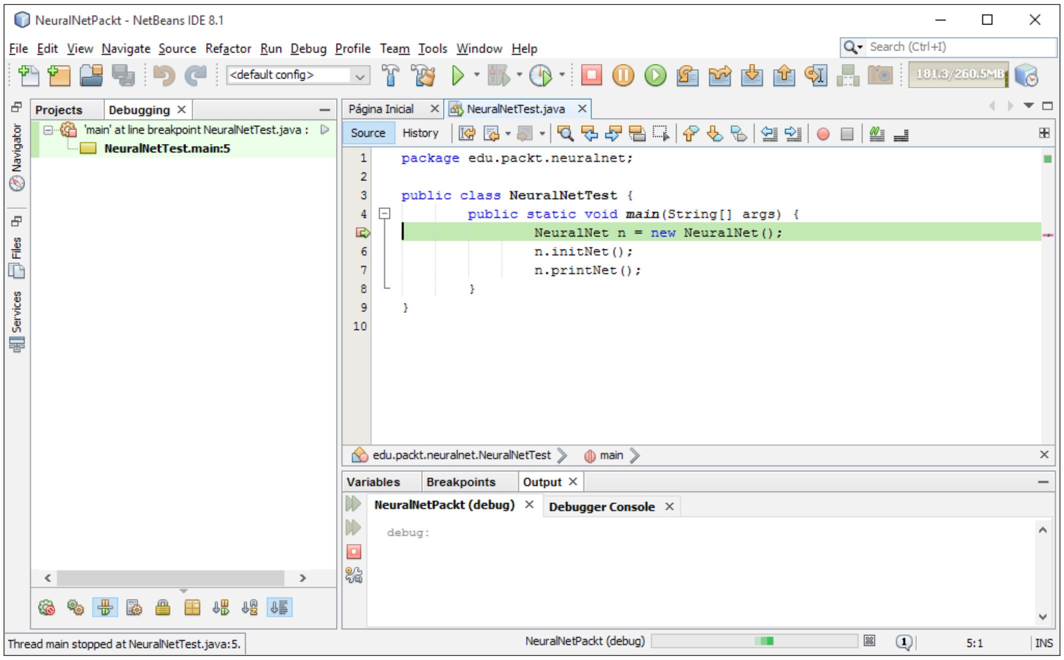

Run Project

Project NeuralNetPackt does not have a main class set.

Select the main class:

edu.packt.neuralnet.NeuralNetTest

To debug line-to-line, you should add a breakpoint. So you can place a breakpoint by clicking on the corresponding line number. Let's add a breakpoint in the beginning of the main method.


NeuralNetPackt - NetBeans IDE 8.1

'main' at line breakpoint NeuralNetTest.java :
NeuralNetTest.main:5

```
package edu.packt.neuralnet;

public class NeuralNetTest {
        public static void main(String[] args) {
                NeuralNet n = new NeuralNet();
                n.initNet();
                n.printNet();
        }
}
```

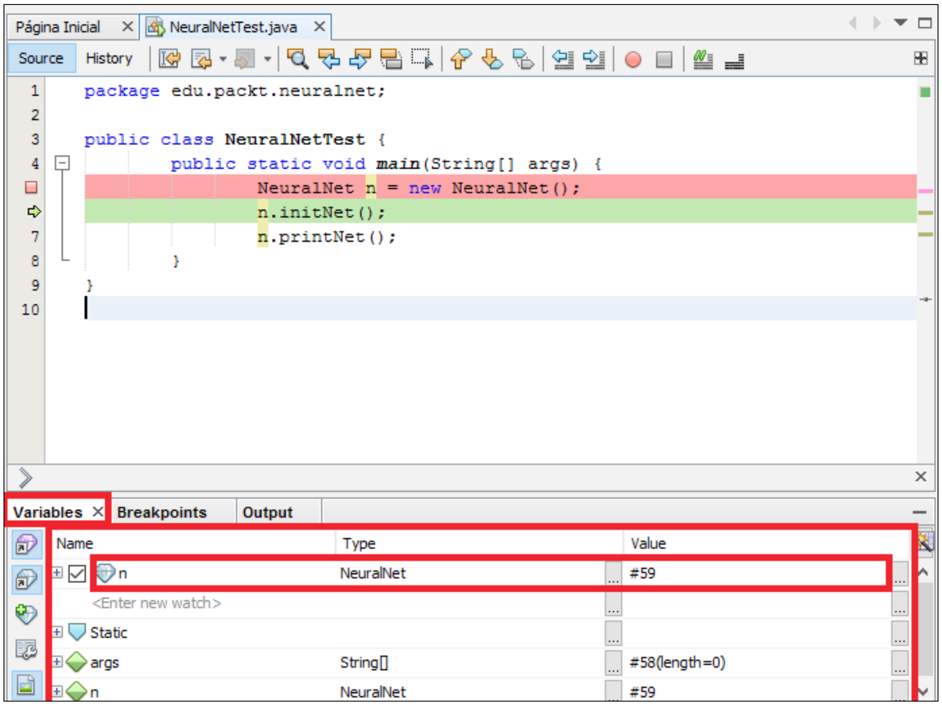
```
debug:
```

Thread main stopped at NeuralNetTest.java:5.

Use the following commands for debugging line-to-line on the source code:

- *F5*: To step into method;
- *F6*: To step over method;
- *F7*: To step return;
- *F8*: To resume debug;
- *Ctrl+F2*: To terminate the debug:

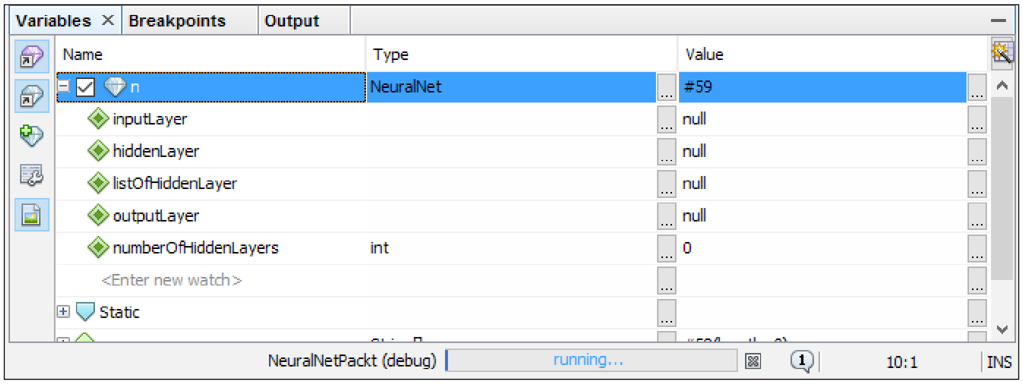
To inspect the value of a variable, just right click on the code screen and select the option **New Watch** (or just *Ctrl+Shift+F7*). Insert the name of the variable or expression you want to watch and click **ok**. You can see at the bottom of the screen a section called variables, where all the user custom expressions and relevant variables are displayed with their current values:



```
package edu.packt.neuralnet;

public class NeuralNetTest {
        public static void main(String[] args) {
                NeuralNet n = new NeuralNet();
                n.initNet();
                n.printNet();
        }
}
```

Name	Type	Value
n	NeuralNet	#59
<Enter new watch>		
Static		
args	String[]	#58(length=0)
n	NeuralNet	#59

In the preceding case, `n` is an object, so you can expand all its attributes by clicking on the **+** sign at its left. And there you are, all the following attributes are shown:



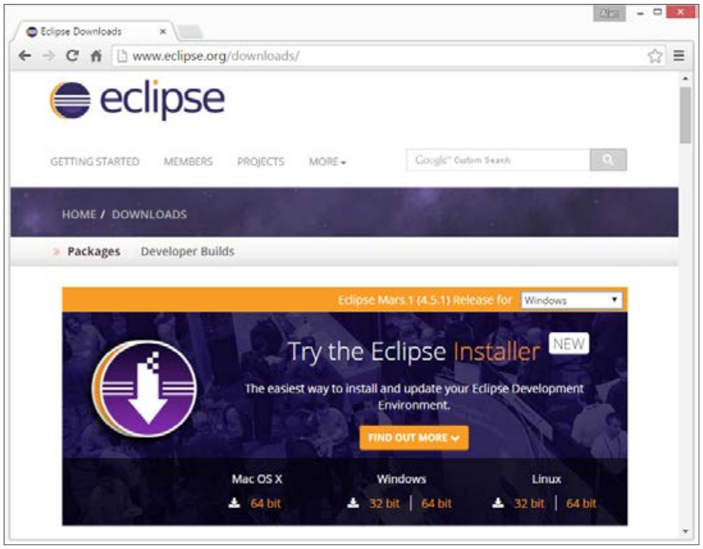
Name	Type	Value
n	NeuralNet	#59
inputLayer		null
hiddenLayer		null
listOfHiddenLayer		null
outputLayer		null
numberOfHiddenLayers	int	0
<Enter new watch>		
Static		

B
Setting up Eclipse Environment

This appendix shows a step-by-step procedure for the reader to set up your development environment if you want to use Eclipse IDE.

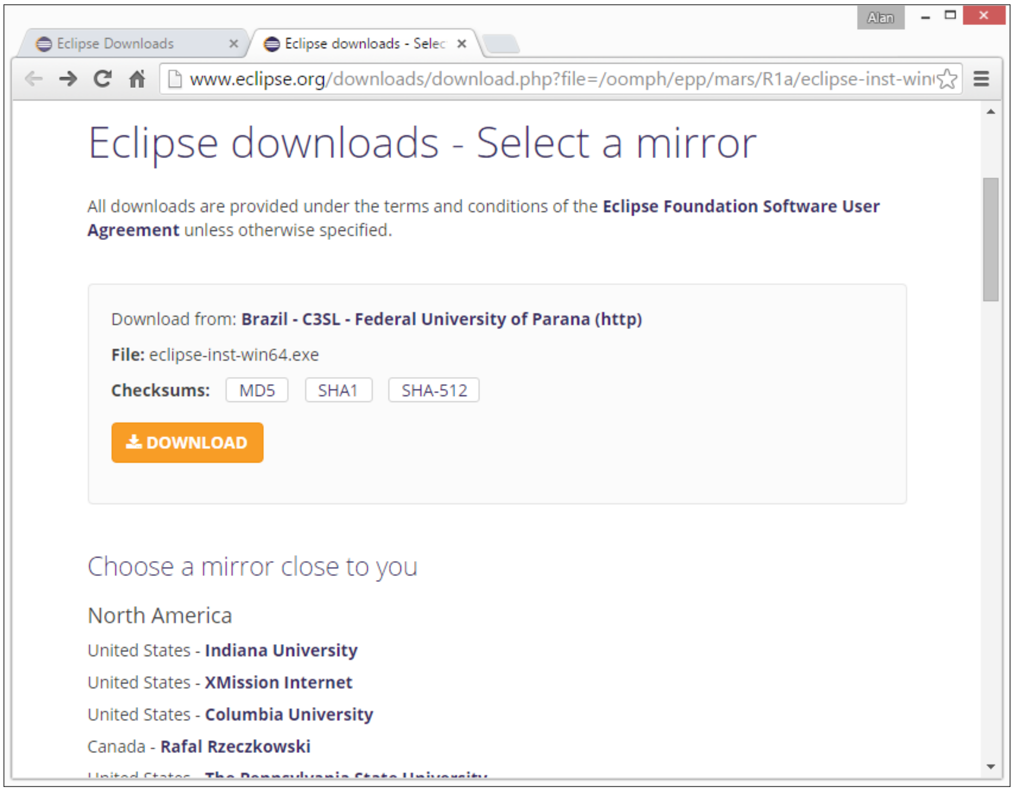
Downloading and installing Eclipse

Before downloading and installing Eclipse, ensure you have installed **Java Development Kit** (**JDK**), accessing `https://www.oracle.com/java/index.html`. Access `http://www.eclipse.org/downloads` (as shown in the following screenshot); choose the Eclipse installer to download according your operation system. Recently, Eclipse team made the installation process easier trough Eclipse Installer:



The easiest way to install and update your Eclipse Development Environment.

The web page represented by following screenshot appears. You should click on the button **Download**. The best mirror to download is selected automatically, but if you want to choose another one you may do in the bottom:


Alan
Eclipse Downloads
Eclipse downloads - Selec
www.eclipse.org/downloads/download.php?file=/oomph/epp/mars/R1a/eclipse-inst-win
MD5
SHA1
SHA-512
DOWNLOAD

Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

Download from: **Brazil - C3SL - Federal University of Parana (http)**

File: eclipse-inst-win64.exe

Checksums:

Choose a mirror close to you

North America

United States - **Indiana University**

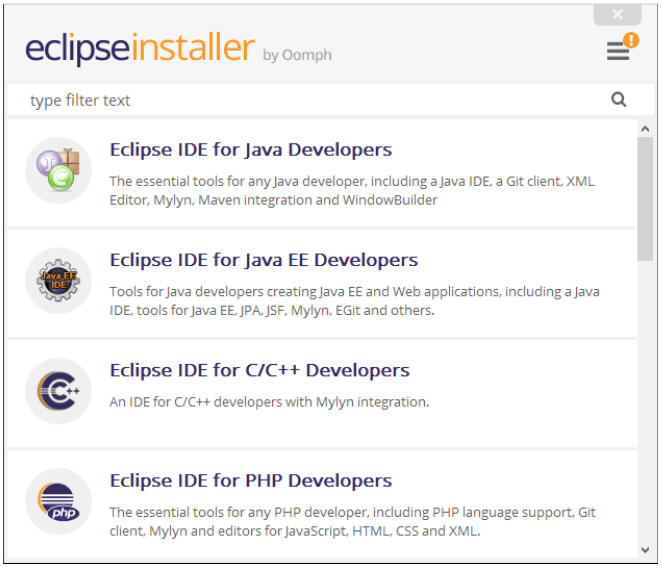
United States - **XMission Internet**

United States - **Columbia University**

Canada - **Rafal Rzeczkowski**

United States - **The Pennsylvania State University**

After download, you should run the executable (`eclipse-inst-<your_os>.exe`). The following screen appears. As we will not develop web applications, you should click on the first option **Eclipse IDE for Java Developers**:



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder

Eclipse IDE for Java EE Developers

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.

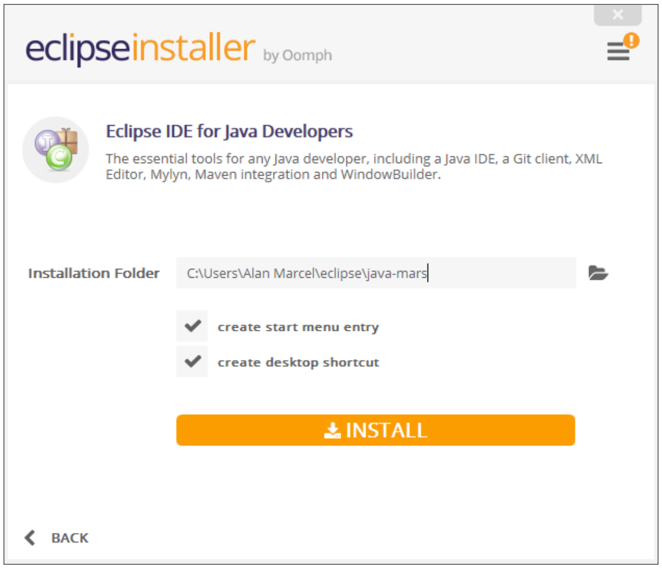
Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers with Mylyn integration.

Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Now you should select the installation folder and decide if you want create start menu entry and desktop shortcut. Then, click on the **INSTALL** button:



Eclipse IDE for Java Developers

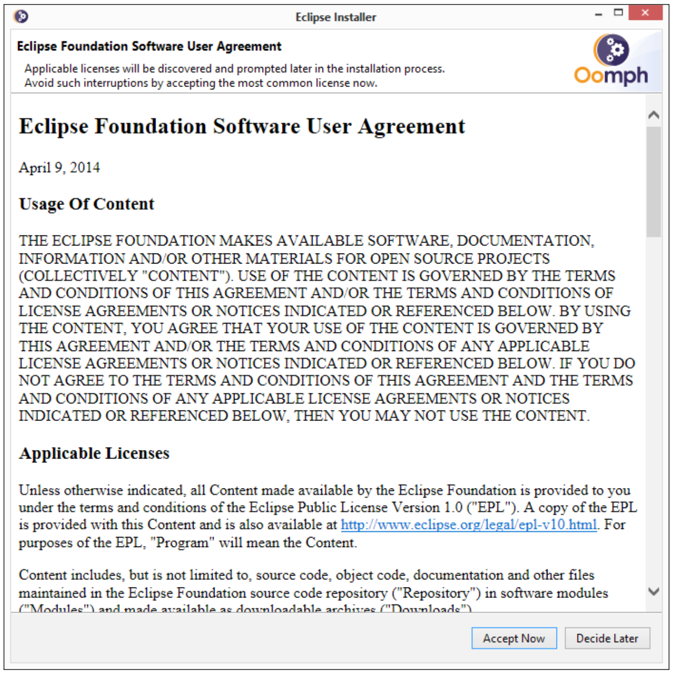
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder.

Installation Folder: C:\Users\Alan Marcel\eclipse\java-mars

☑ create start menu entry

☑ create desktop shortcut

You must accept the licence, clicking on the **Accept Now** button:



Eclipse Foundation Software User Agreement

Applicable licenses will be discovered and prompted later in the installation process.
Avoid such interruptions by accepting the most common license now.

Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

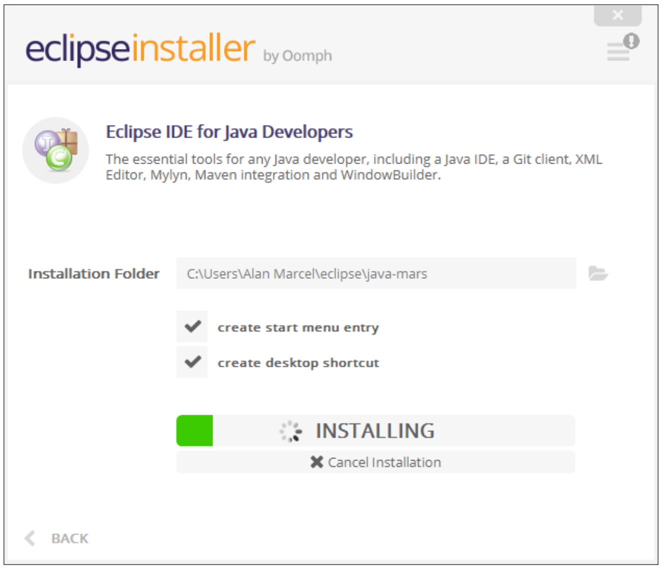
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at http://www.eclipse.org/legal/epl-v10.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as downloadable archives ("Downloads").

Finally, the installation process begins:



Eclipse IDE for Java Developers

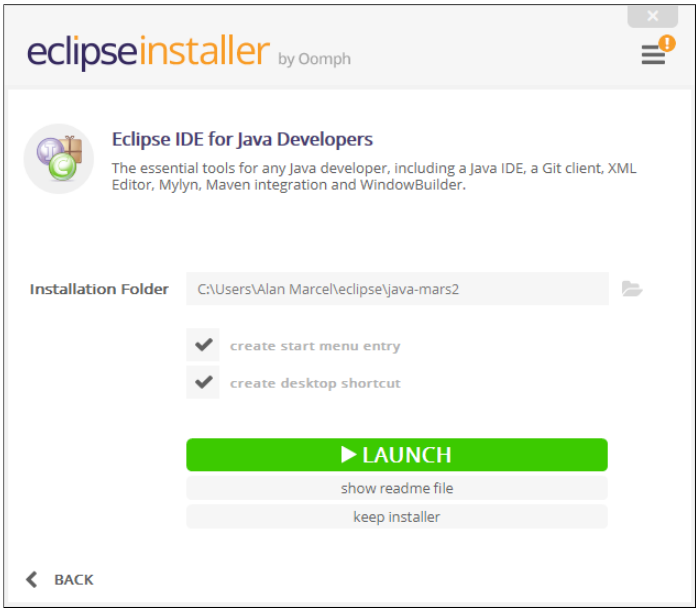
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder.

Installation Folder C:\Users\Alan Marcel\eclipse\java-mars

✓ create start menu entry

✓ create desktop shortcut

After the installation, you may click on the **LAUNCH** button to run Eclipse:



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder.

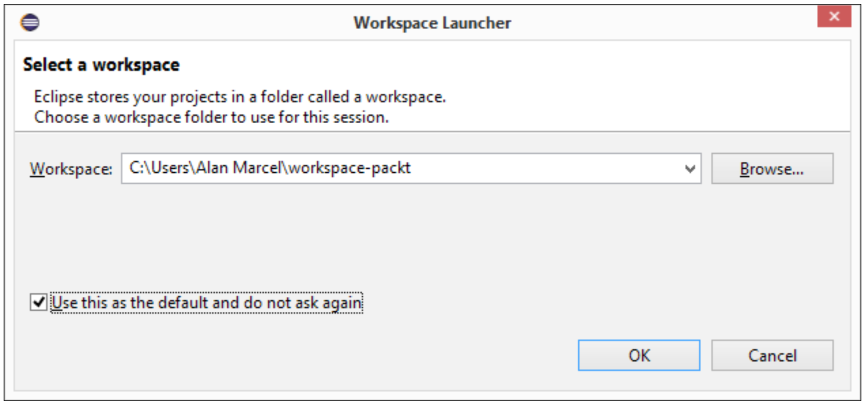
Installation Folder: C:\Users\Alan Marcel\eclipse\java-mars2

☑ create start menu entry

☑ create desktop shortcut

Setting up Eclipse environment

Now, the next step consists in choose the workspace folder where your projects will be placed. If you mark the option **Use this as the default and do not ask again**, next time you run Eclipse, it will not be necessary inform the workspace folder again. Click on the **OK** button:



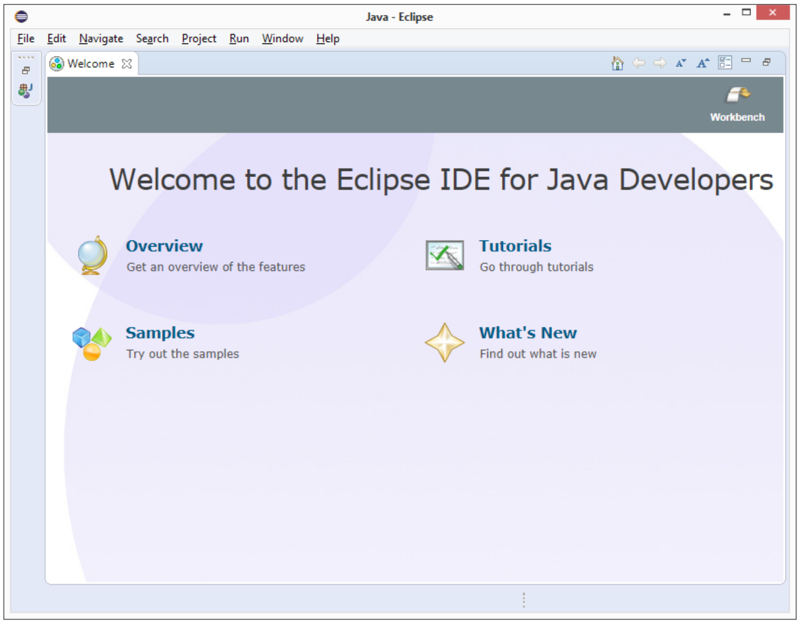
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\Alan Marcel\workspace-packt

☑ Use this as the default and do not ask again

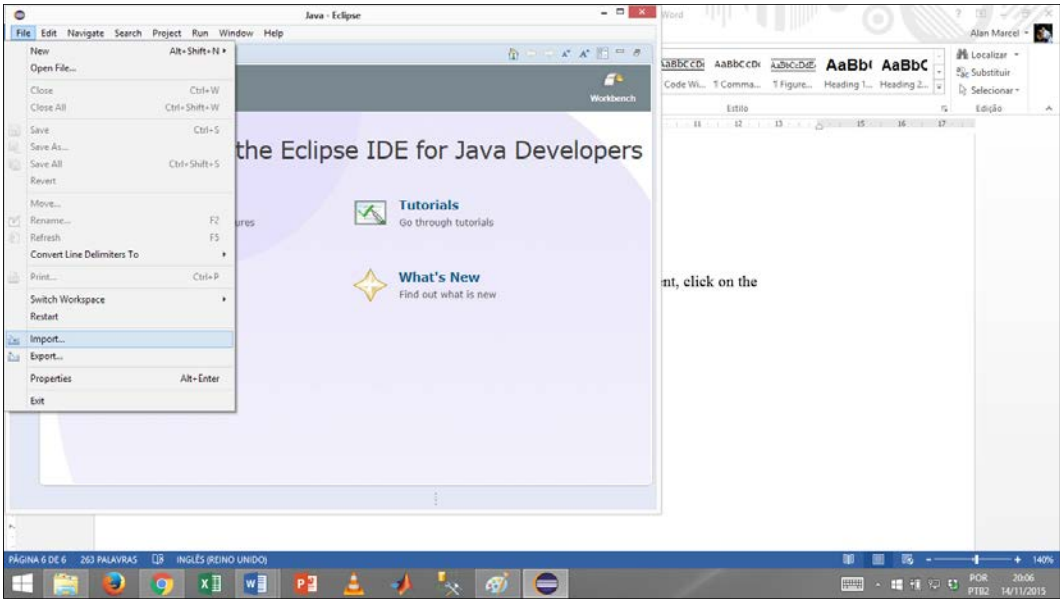
The welcome screen shows and you are ready to start the Java programming:


Java - Eclipse
File Edit Navigate Search Project Run Window Help
Welcome
Workbench
Welcome to the Eclipse IDE for Java Developers
Overview
Get an overview of the features
Tutorials
Go through tutorials
Samples
Try out the samples
What's New
Find out what is new

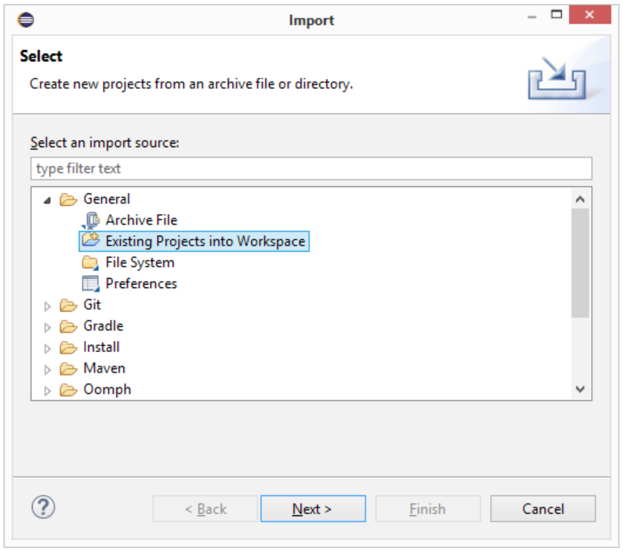
Importing a project

> Before importing the project, make sure you unzip it in a folder you know.

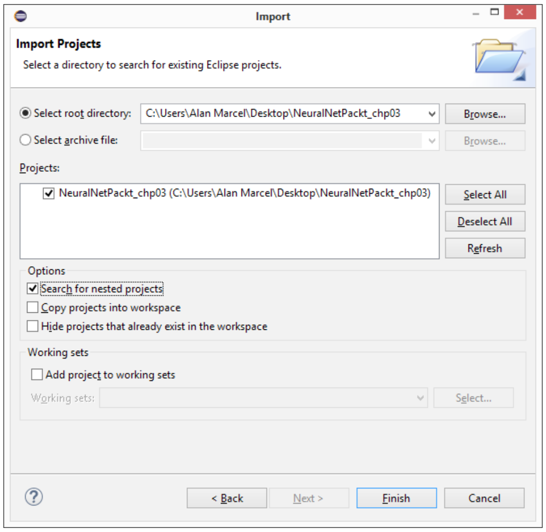
To import a project already developed to your Eclipse IDE, follow the menu **File** | **Import...**:



The import prompt displays, you should expand the **General** option, choose **Existing Projects into Workspace** and click on the **Next** button:



Now, click on the **Browse...** button to search for the project unzip folder. After that, do not forget to mark **Search for nested projects** option and click on the **Finish** button:



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Users\Alan Marcel\Desktop\NeuralNetPackt_chp03

○ Select archive file:

Projects:

☑ NeuralNetPackt_chp03 (C:\Users\Alan Marcel\Desktop\NeuralNetPackt_chp03)

Options

☑ Search for nested projects

□ Copy projects into workspace

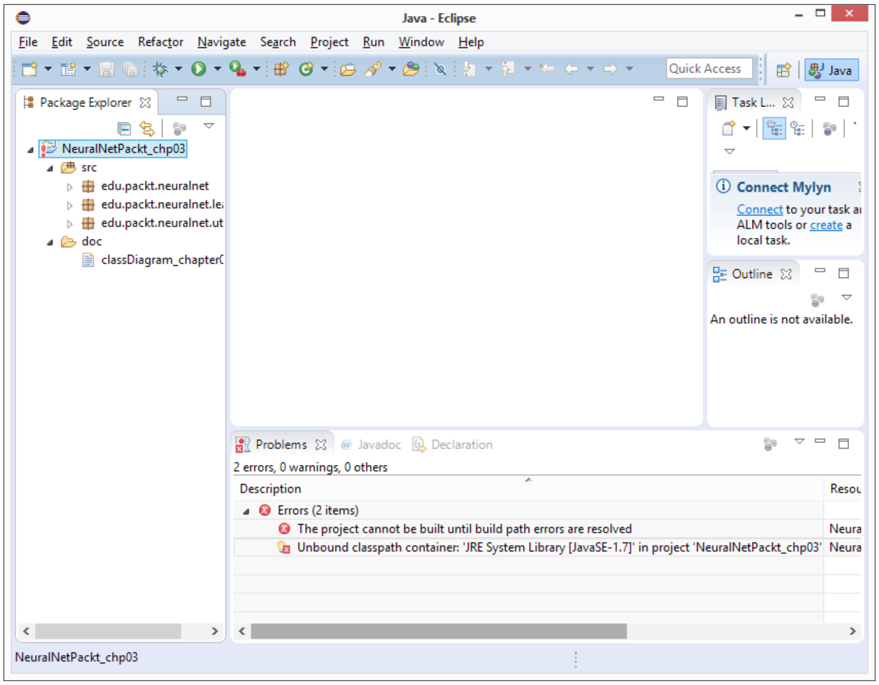
□ Hide projects that already exist in the workspace

Working sets

□ Add project to working sets

Working sets:

In this step, you should close the Welcome screen and you will see in the **Package Explorer** the project you imported to Eclipse:



NeuralNetPackt_chp03
src
edu.packt.neuralnet
edu.packt.neuralnet.le
edu.packt.neuralnet.ut
doc
classDiagram_chapter(

Connect to your task ar ALM tools or create a local task.

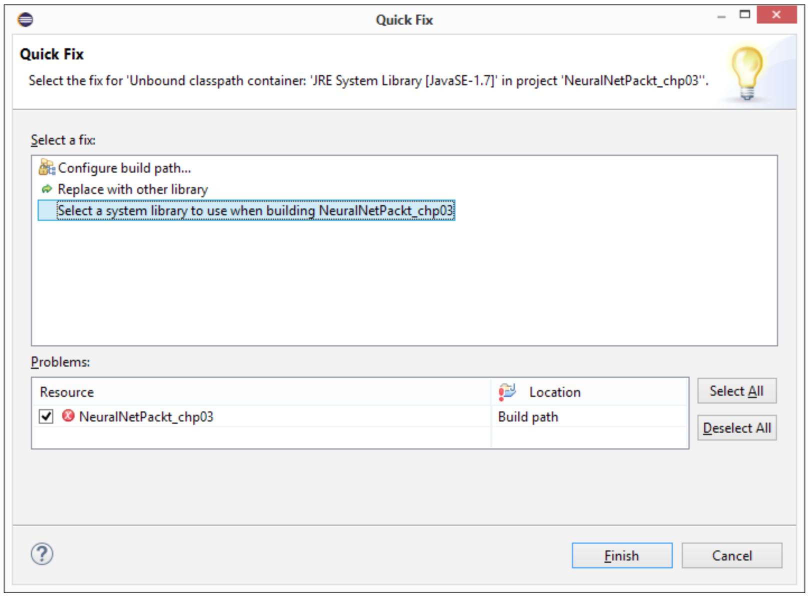
An outline is not available.

2 errors, 0 warnings, 0 others

Description	Resou
Errors (2 items)	
The project cannot be built until build path errors are resolved	Neura
Unbound classpath container: 'JRE System Library [JavaSE-1.7]' in project 'NeuralNetPackt_chp03'	Neura

NeuralNetPackt_chp03

If you see a JRE version error, you should open **Problems** tab, click with right button in the **Unbound classpath container…**, and choose the **Quick Fix**. The next screen will be exhibit:



Quick Fix

Select the fix for 'Unbound classpath container: 'JRE System Library [JavaSE-1.7]' in project 'NeuralNetPackt_chp03''.

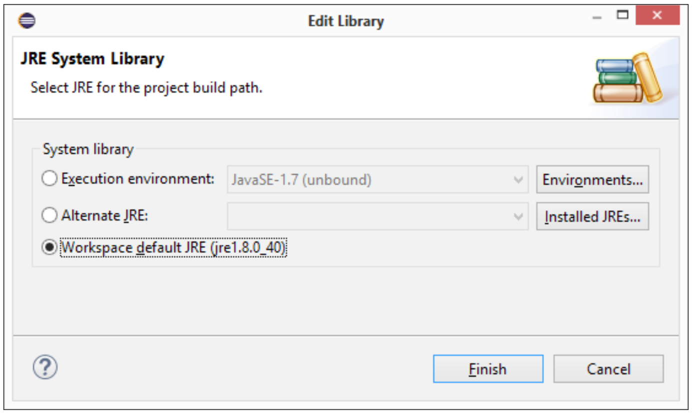
Select a fix:

- Configure build path...
- Replace with other library
- Select a system library to use when building NeuralNetPackt_chp03

Problems:

Resource	Location
☑ NeuralNetPackt_chp03	Build path

Now, click on the **Select a system library to use…** option and **Finish** button. The **Edit Library** window appears and you should the **Workspace default JRE (jre1.8.0_40)** and click on the **Finish** button:



JRE System Library

Select JRE for the project build path.

System library

○ Execution environment: JavaSE-1.7 (unbound)

○ Alternate JRE:

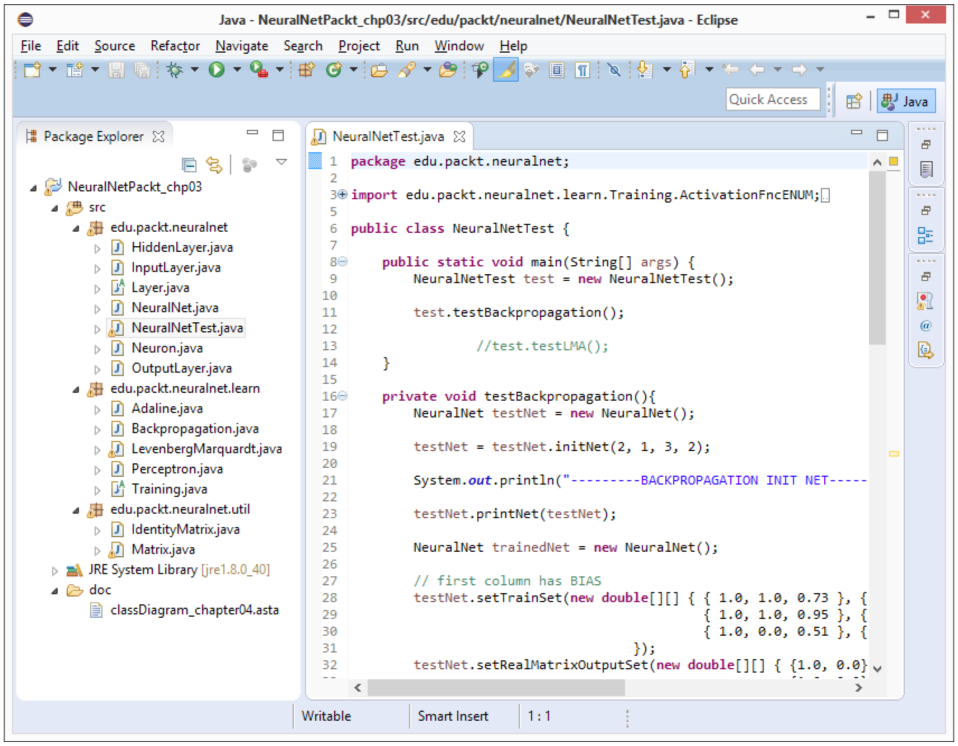
● Workspace default JRE (jre1.8.0_40)

Programming and running code with Eclipse IDE

Make sure you run Eclipse IDE as administrator.

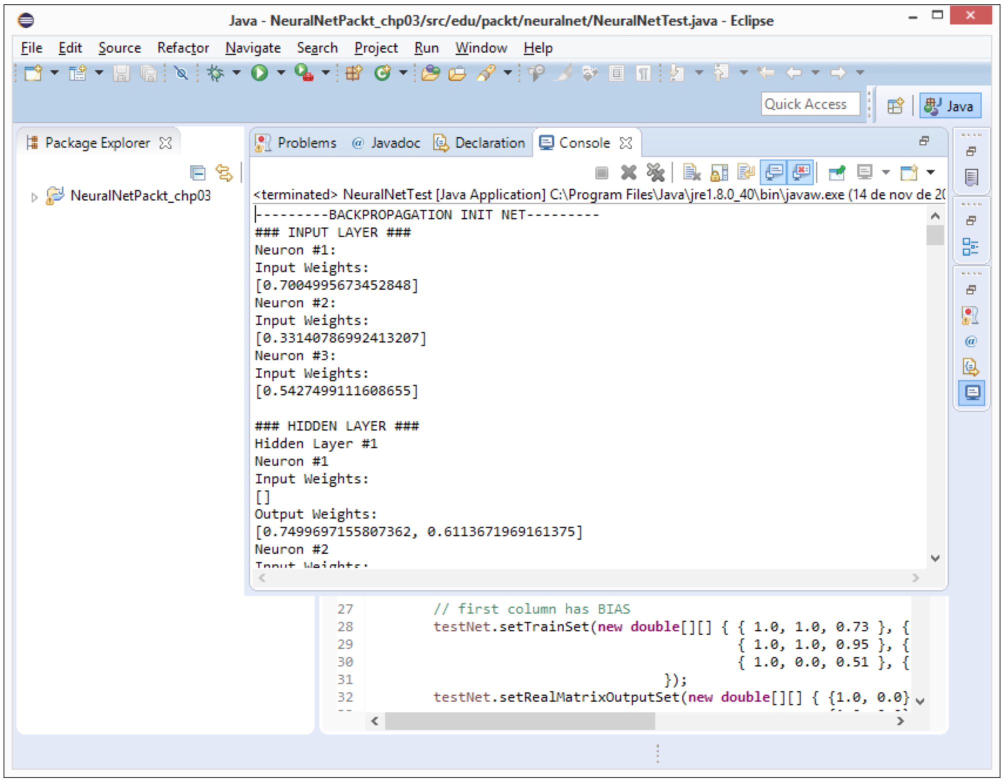
After all previous steps, you are able to start Java programming. The next screenshot shows the structure of the Eclipse:

- **Package Explorer**: This section is displayed in the left of the packages and classes that compose the Java project.
- **Code**: It is shown in the middle of the screen and brings the code you should interact.
- **Run the code**: There are many ways to run the code. Perhaps, the easiest one is the **Play Button** indicated by arrow **A**.
- **Debug the code**: There are also many ways to debug the code. The easiest one is the **Bug Button** appointed by arrow **B**.



```
package edu.packt.neuralnet;

import edu.packt.neuralnet.learn.Training.ActivationFncENUM;

public class NeuralNetTest {

    public static void main(String[] args) {
        NeuralNetTest test = new NeuralNetTest();

        test.testBackpropagation();

                //test.testLMA();
    }

    private void testBackpropagation(){
        NeuralNet testNet = new NeuralNet();

        testNet = testNet.initNet(2, 1, 3, 2);

        System.out.println("---------BACKPROPAGATION INIT NET-----

        testNet.printNet(testNet);

        NeuralNet trainedNet = new NeuralNet();

        // first column has BIAS
        testNet.setTrainSet(new double[][] { { 1.0, 1.0, 0.73 }, {
                                             { 1.0, 1.0, 0.95 }, {
                                             { 1.0, 0.0, 0.51 }, {
                                        });
        testNet.setRealMatrixOutputSet(new double[][] { {1.0, 0.0}
```

When you click on the button to run the code, it runs into **Console** tab as shown in the following screenshot:



```
<terminated> NeuralNetTest [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (14 de nov de 2(
---------BACKPROPAGATION INIT NET---------
### INPUT LAYER ###
Neuron #1:
Input Weights:
[0.7004995673452848]
Neuron #2:
Input Weights:
[0.33140786992413207]
Neuron #3:
Input Weights:
[0.5427499111608655]

### HIDDEN LAYER ###
Hidden Layer #1
Neuron #1
Input Weights:
[]
Output Weights:
[0.7499697155807362, 0.6113671969161375]
Neuron #2
Input Weights:
```

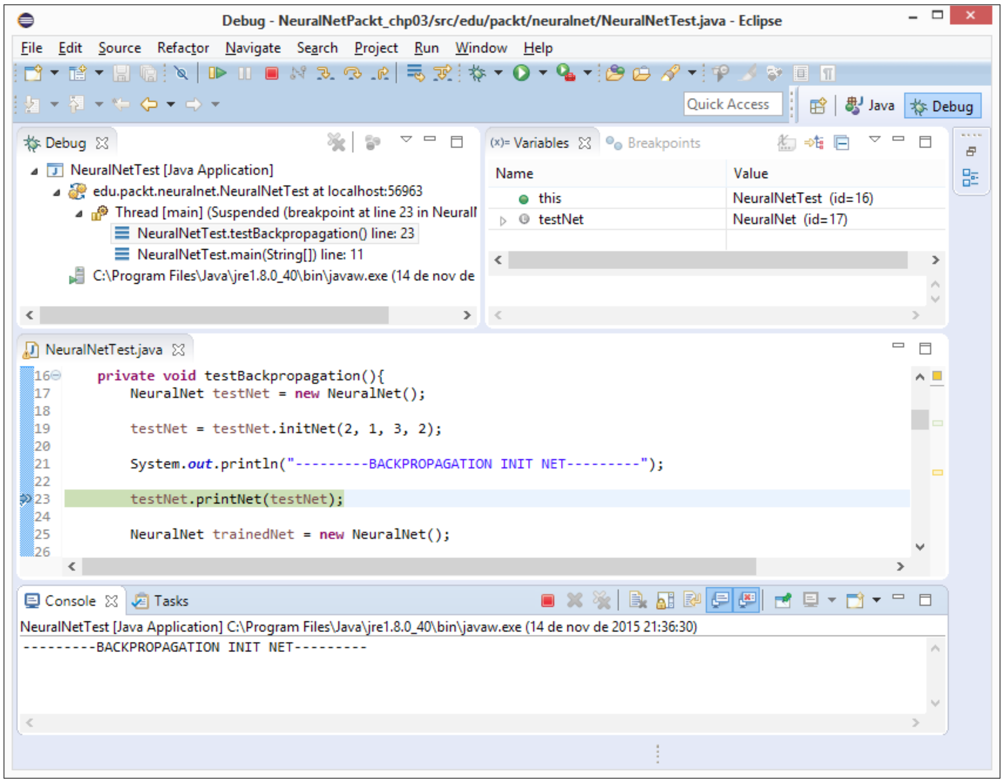
```
        // first column has BIAS
        testNet.setTrainSet(new double[][] { { 1.0, 1.0, 0.73 }, {
                                             { 1.0, 1.0, 0.95 }, {
                                             { 1.0, 0.0, 0.51 }, {
                                    });
        testNet.setRealMatrixOutputSet(new double[][] { {1.0, 0.0}
```

Debugging with Eclipse IDE

To debug a Java class using Eclipse IDE, you must create a breakpoint. It can be make simply double-clicking near the line number (a blue round will be displayed). Then, when you click on the **Debug** Button to run the debugging process, the execution of the class will stop right on the line marked with breakpoint and you may type on the keyboard:

- *F5*: To step into method;
- *F6*: To step over method;
- *F7*: To step return;
- *F8*: To resume debug;
- *Ctrl+F2*: To terminate the debug.

The next screenshotshows the Eclipse debug screen. There is an important section in this screen, on the right-top corner, named **Variables**, that shows the variables and it respective current values:



NeuralNetTest [Java Application]
edu.packt.neuralnet.NeuralNetTest at localhost:56963
Thread [main] (Suspended (breakpoint at line 23 in NeuralI
NeuralNetTest.testBackpropagation() line: 23
NeuralNetTest.main(String[]) line: 11
C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (14 de nov de

Name	Value
this	NeuralNetTest (id=16)
testNet	NeuralNet (id=17)

```
    private void testBackpropagation(){
        NeuralNet testNet = new NeuralNet();

        testNet = testNet.initNet(2, 1, 3, 2);

        System.out.println("---------BACKPROPAGATION INIT NET---------");

        testNet.printNet(testNet);

        NeuralNet trainedNet = new NeuralNet();

```

NeuralNetTest [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (14 de nov de 2015 21:36:30)

```
---------BACKPROPAGATION INIT NET---------
```