Brooklyn College

NetBeans

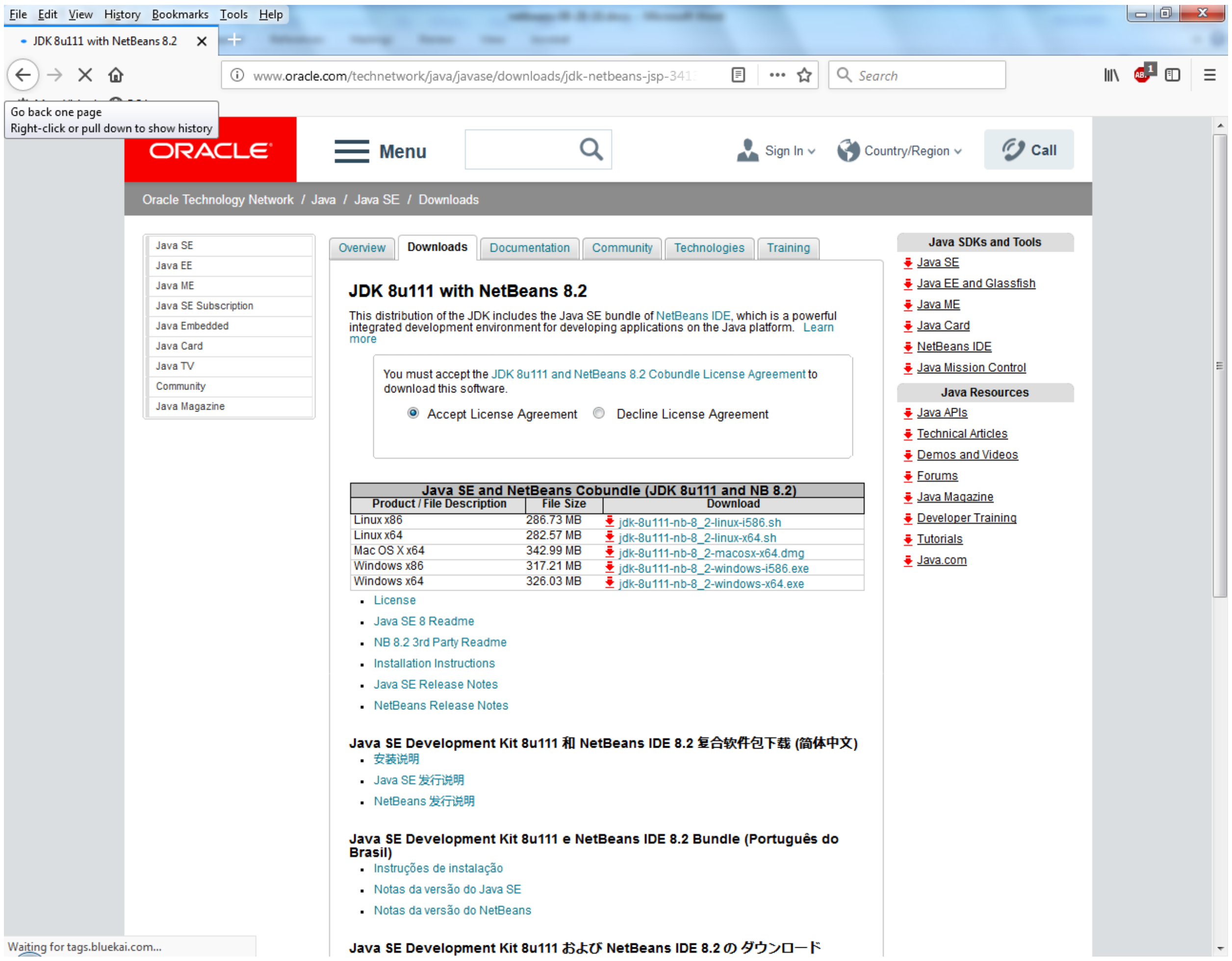
Beginner's Guide

Lawrence Goetz
8/28/2018

Contents

Installation

Download the [JDK with NetBeans IDE Java SE bundle]().



JDK 8u111 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. Learn more

You must accept the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement to download this software.

[✓ Accept License Agreement] Decline License Agreement

Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	jdk-8u111-nb-8_2-linux-i586.sh
Linux x64	282.57 MB	jdk-8u111-nb-8_2-linux-x64.sh
Mac OS X x64	342.99 MB	jdk-8u111-nb-8_2-macosx-x64.dmg
Windows x86	317.21 MB	jdk-8u111-nb-8_2-windows-i586.exe
Windows x64	326.03 MB	jdk-8u111-nb-8_2-windows-x64.exe

- License
- Java SE 8 Readme
- NB 8.2 3rd Party Readme
- Installation Instructions
- Java SE Release Notes
- NetBeans Release Notes

Java SE Development Kit 8u111 和 NetBeans IDE 8.2 复合软件包下载 (简体中文)

- 安装说明
- Java SE 发行说明
- NetBeans 发行说明

Java SE Development Kit 8u111 e NetBeans IDE 8.2 Bundle (Português do Brasil)

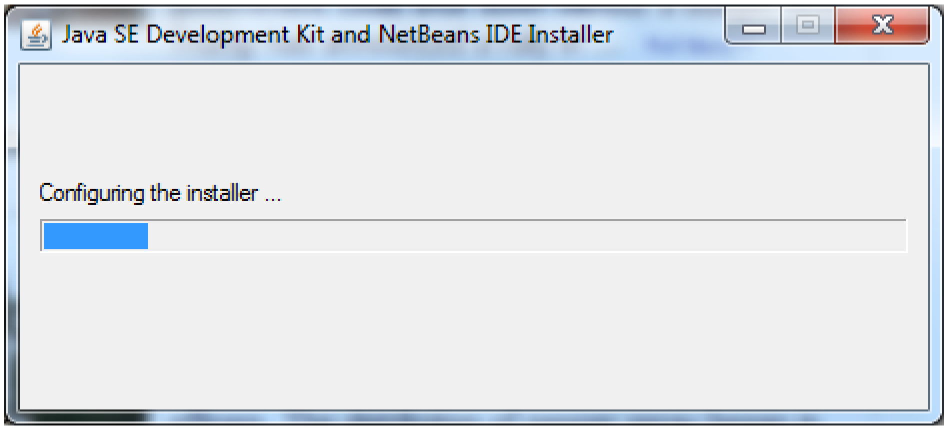
- Instruções de instalação
- Notas da versão do Java SE
- Notas da versão do NetBeans

Java SE Development Kit 8u111 および NetBeans IDE 8.2 の ダウンロード

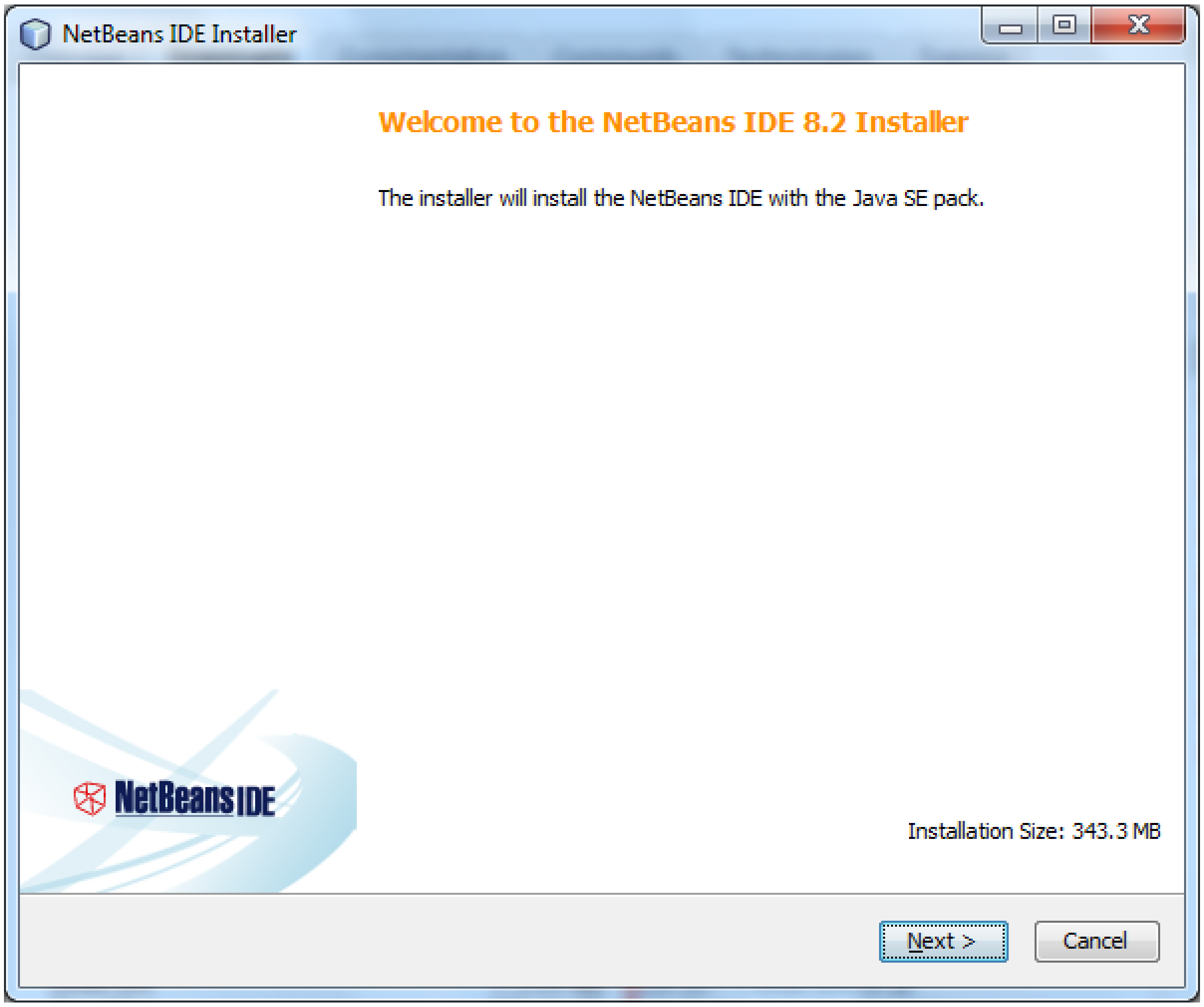
Select the **Accept License Agreement** radio button.

Select the Download link next to the version that matches your Operating System (Such as Windows or Mac x64). Save and then execute the installer.

Windows


Java SE Development Kit and NetBeans IDE Installer

Configuring the installer ...

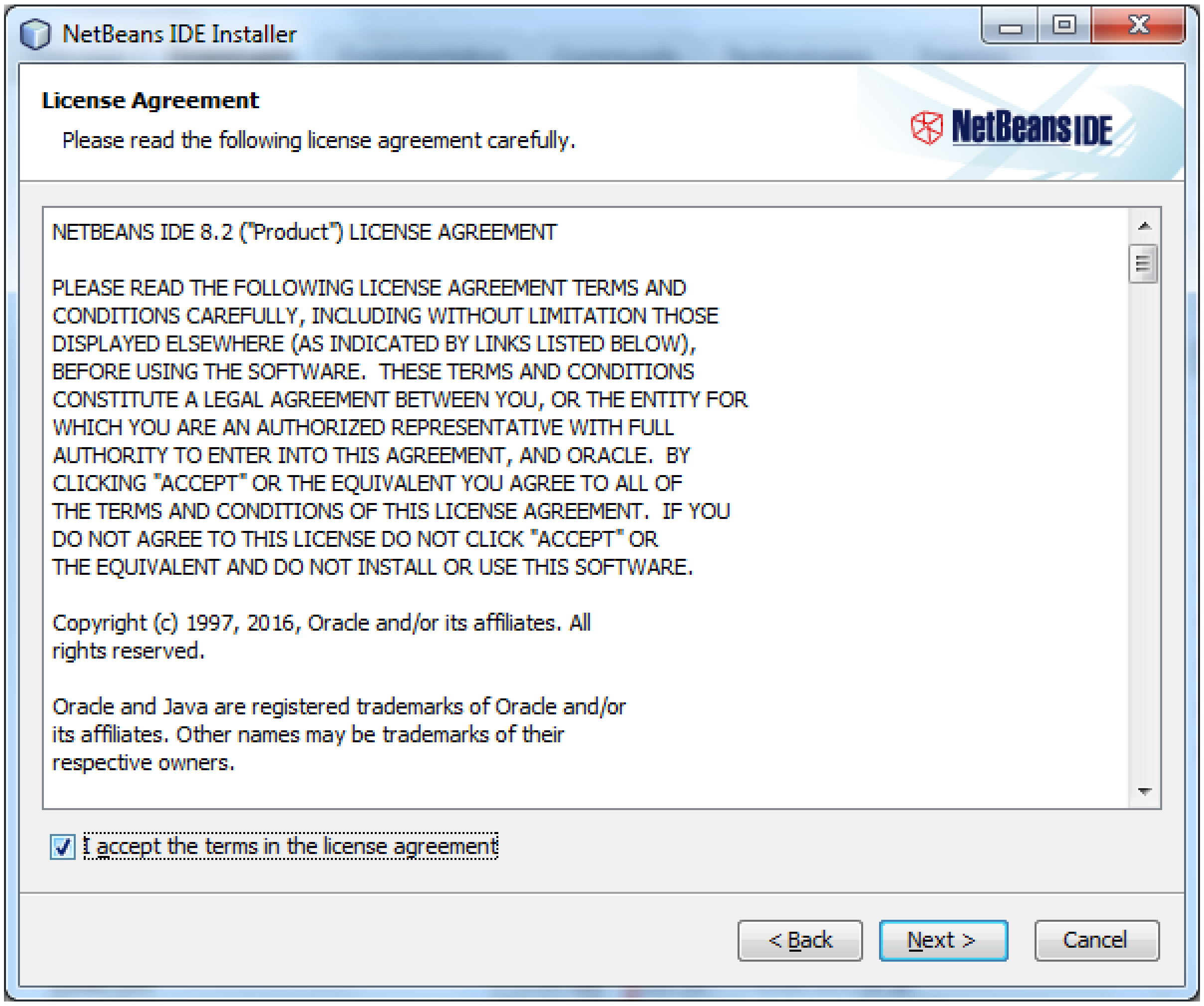

NetBeans IDE Installer

Welcome to the NetBeans IDE 8.2 Installer

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

Installation Size: 343.3 MB

Select **Next**.



NETBEANS IDE 8.2 ("Product") LICENSE AGREEMENT

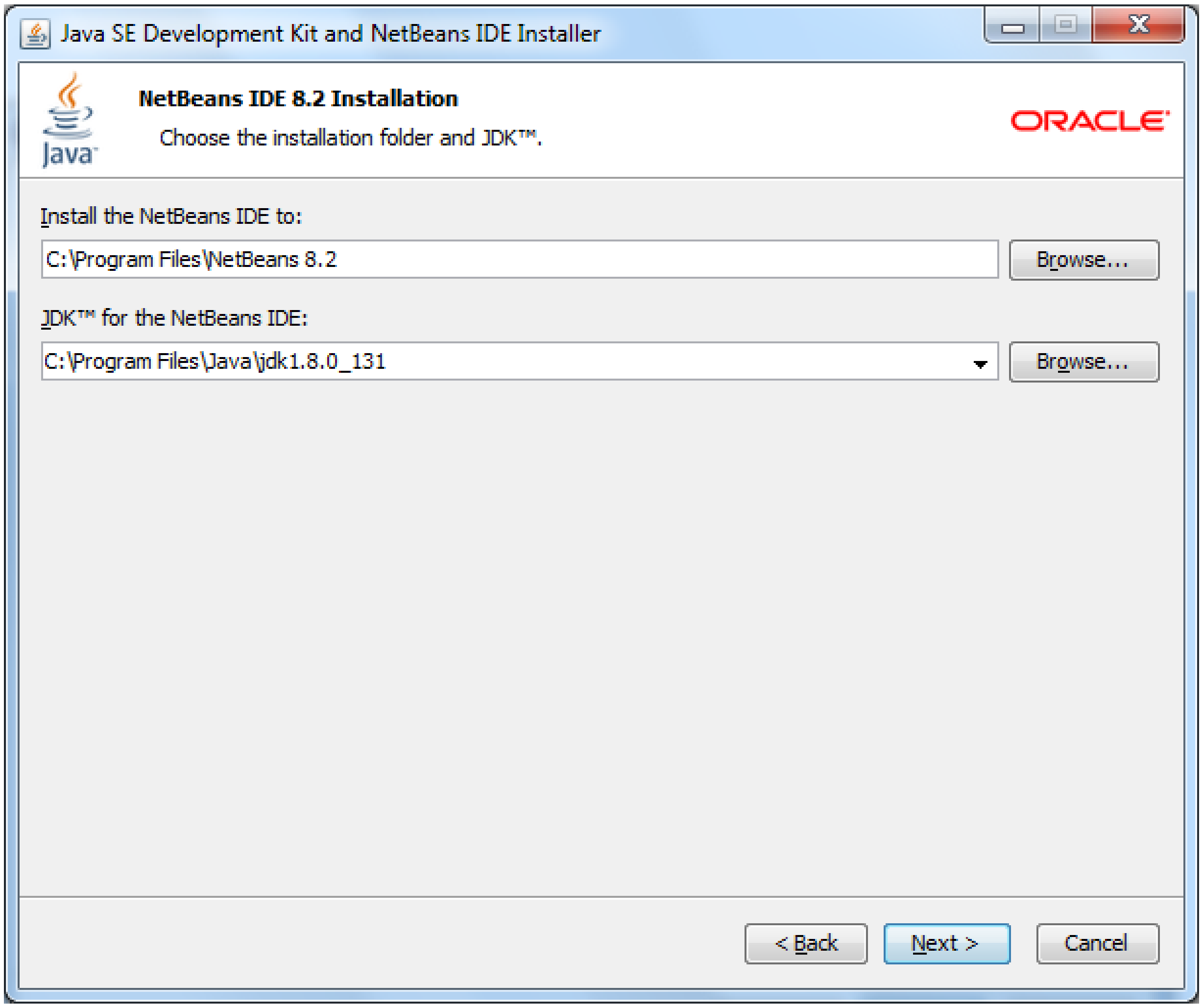
PLEASE READ THE FOLLOWING LICENSE AGREEMENT TERMS AND CONDITIONS CAREFULLY, INCLUDING WITHOUT LIMITATION THOSE DISPLAYED ELSEWHERE (AS INDICATED BY LINKS LISTED BELOW), BEFORE USING THE SOFTWARE. THESE TERMS AND CONDITIONS CONSTITUTE A LEGAL AGREEMENT BETWEEN YOU, OR THE ENTITY FOR WHICH YOU ARE AN AUTHORIZED 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, 2016, 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

Accept the License Agreement and then select **Next**.


Java SE Development Kit and NetBeans IDE Installer

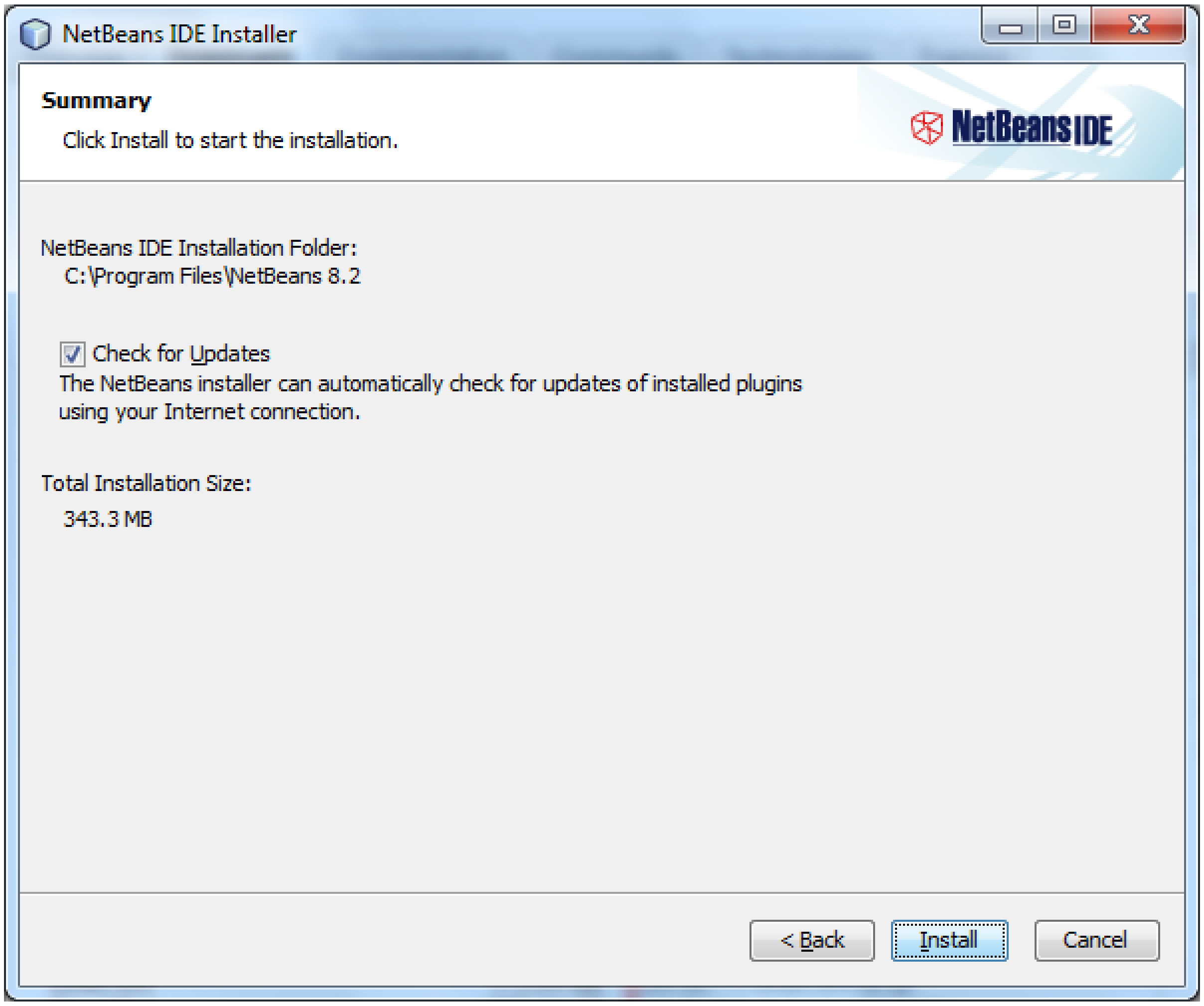
NetBeans IDE 8.2 Installation

Choose the installation folder and JDK™.

Install the NetBeans IDE to:
C:\Program Files\NetBeans 8.2

JDK™ for the NetBeans IDE:
C:\Program Files\Java\jdk1.8.0_131

Take the defaults. Make sure JDK is listed, and then select **Next**.


NetBeans IDE Installer

Summary

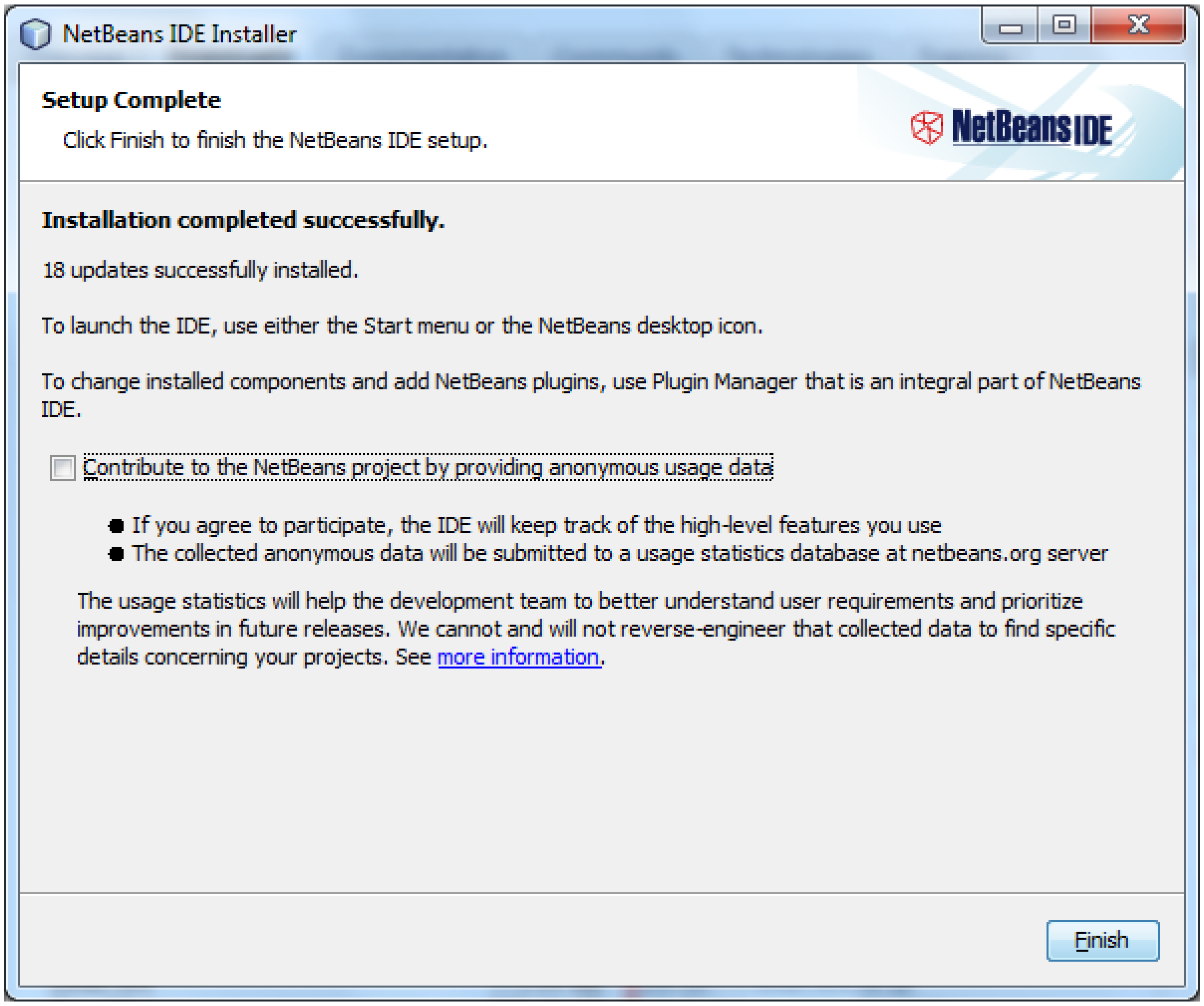
Click Install to start the installation.

NetBeans IDE Installation Folder:
C:\Program Files\NetBeans 8.2

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

Total Installation Size:
343.3 MB

Leave checked to **Check for Updates**. Then click **Install** to proceed with the install.


NetBeans IDE Installer

Setup Complete

Click Finish to finish the NetBeans IDE setup.

Installation completed successfully.

18 updates successfully installed.

To launch the IDE, use either the Start menu or the NetBeans desktop icon.

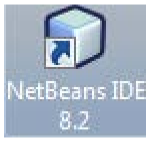
To change installed components and add NetBeans plugins, use Plugin Manager that is an integral part of NetBeans IDE.

☐ Contribute to the NetBeans project by providing anonymous usage data

- If you agree to participate, the IDE will keep track of the high-level features you use
- The collected anonymous data will be submitted to a usage statistics database at netbeans.org server

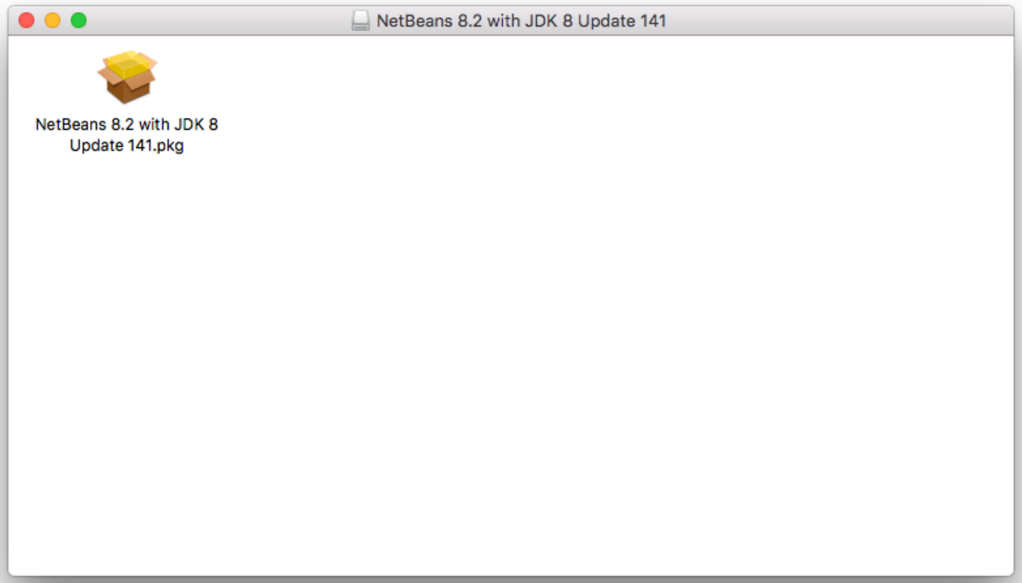
The usage statistics will help the development team to better understand user requirements and prioritize improvements in future releases. We cannot and will not reverse-engineer that collected data to find specific details concerning your projects. See more information.

Uncheck the box to **Contribute** to the Netbeans project. Then click on **Finish**.

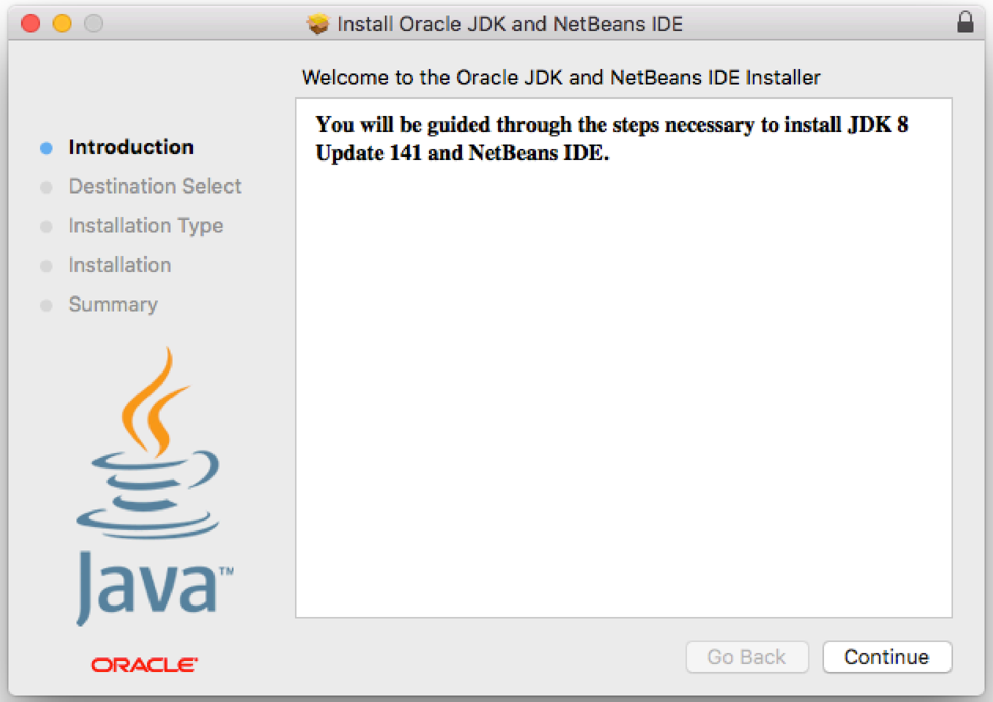


Click on the NetBeans IDE icon to begin.

Mac


NetBeans 8.2 with JDK 8 Update 141
NetBeans 8.2 with JDK 8 Update 141.pkg

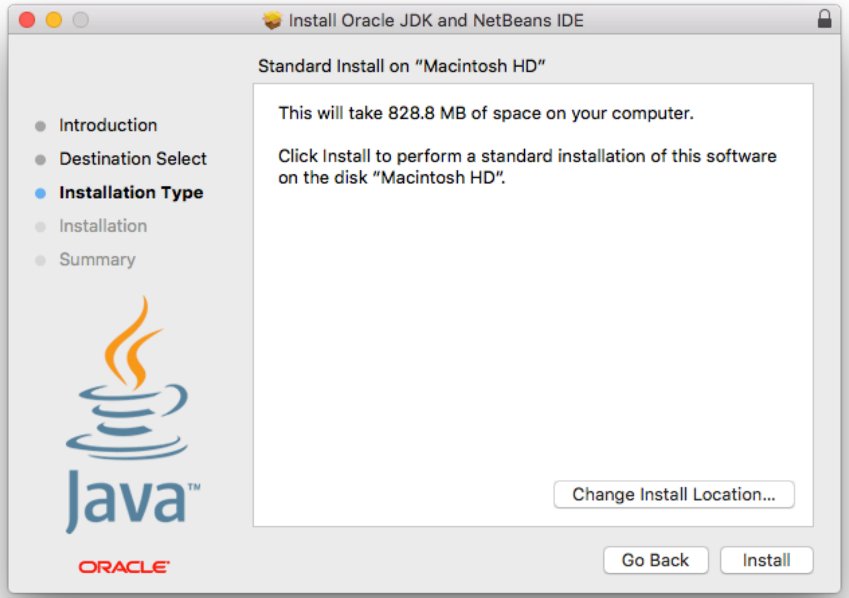
Open the installer package. Select **Continue** from the menu below.



Welcome to the Oracle JDK and NetBeans IDE Installer

You will be guided through the steps necessary to install JDK 8 Update 141 and NetBeans IDE.

Select **Install**

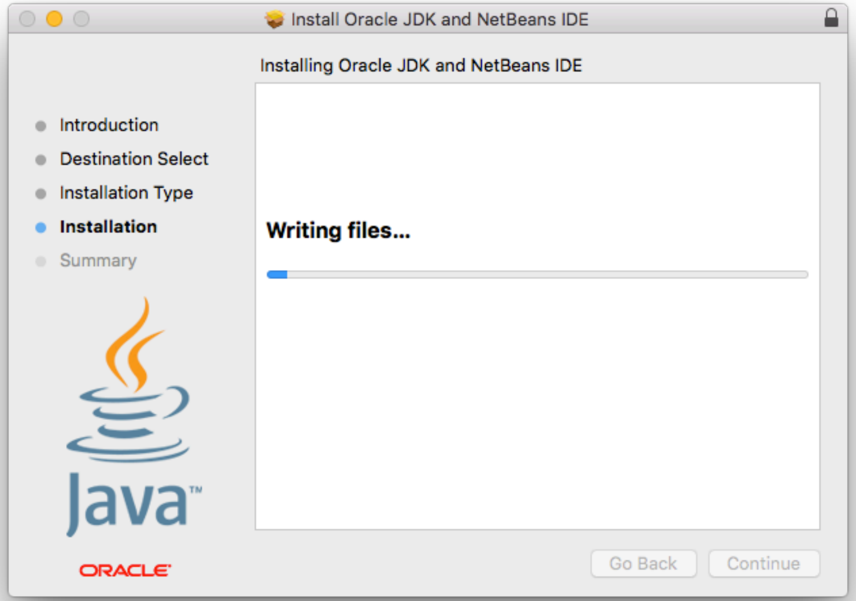


Standard Install on "Macintosh HD"

This will take 828.8 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

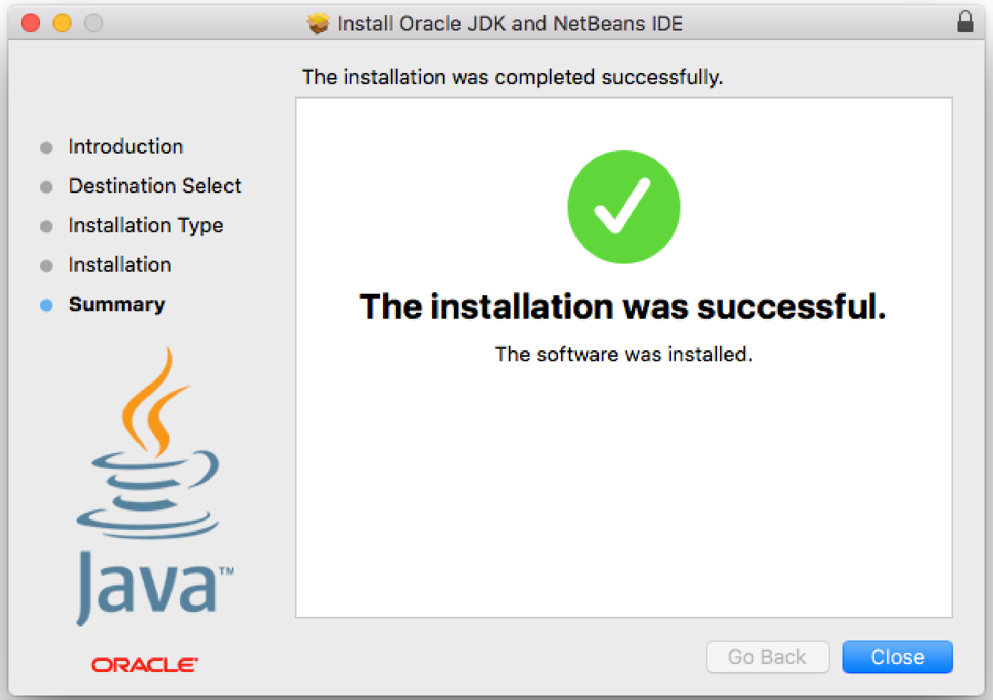
Wait while the files are installed. It will take 5 or more minutes.



Installing Oracle JDK and NetBeans IDE

Writing files...

When done, you can press **Close**.

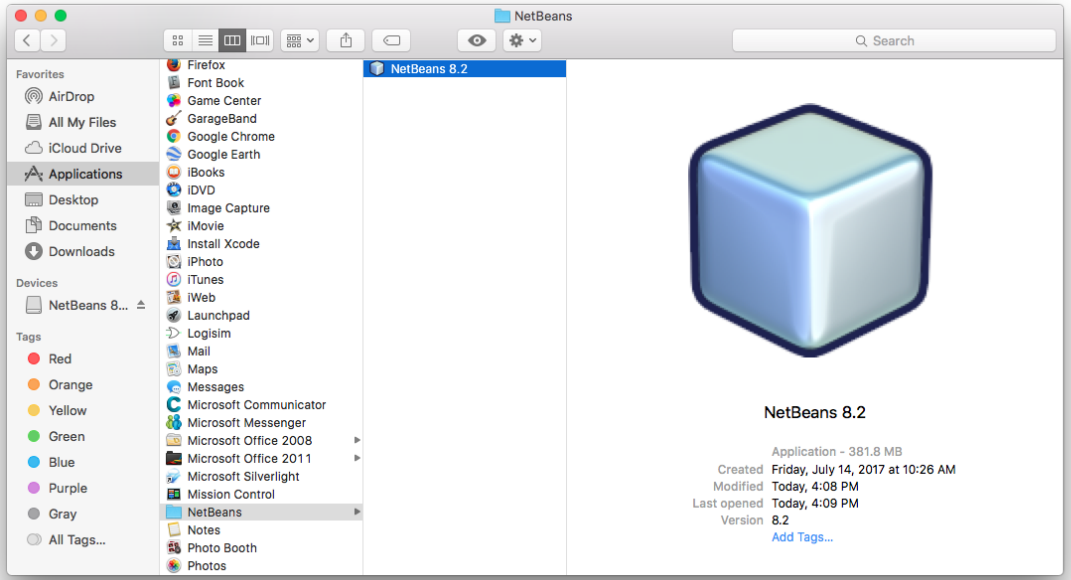

Install Oracle JDK and NetBeans IDE

The installation was completed successfully.

The installation was successful.

The software was installed.

Using the Finder, you will see the icon for **NetBeans** in your **Applications**.

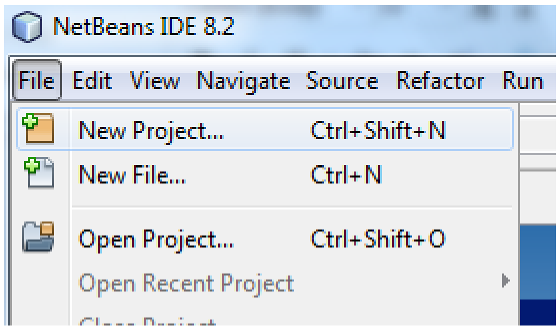

NetBeans 8.2

Application - 381.8 MB

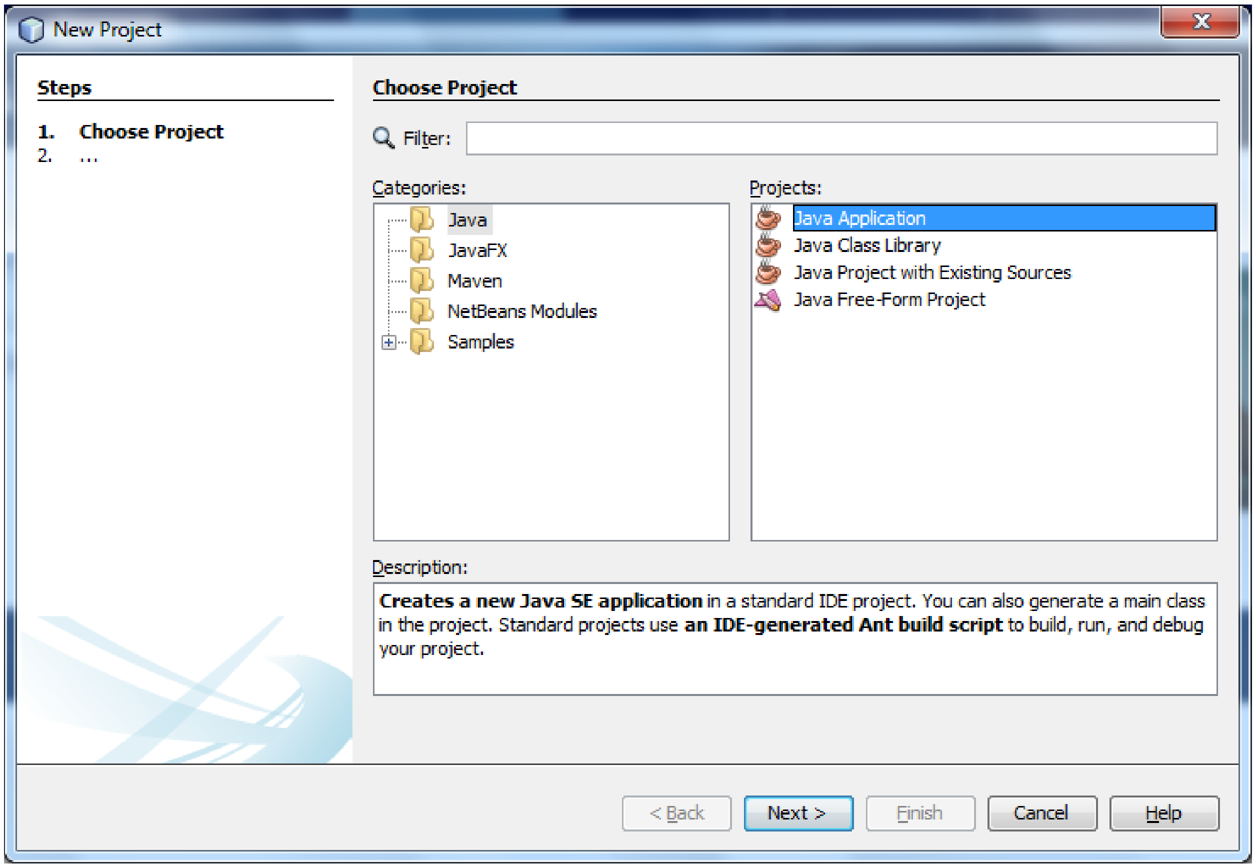
Created	Friday, July 14, 2017 at 10:26 AM
Modified	Today, 4:08 PM
Last opened	Today, 4:09 PM
Version	8.2

Sample NetBeans Project

Making a New Project


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project

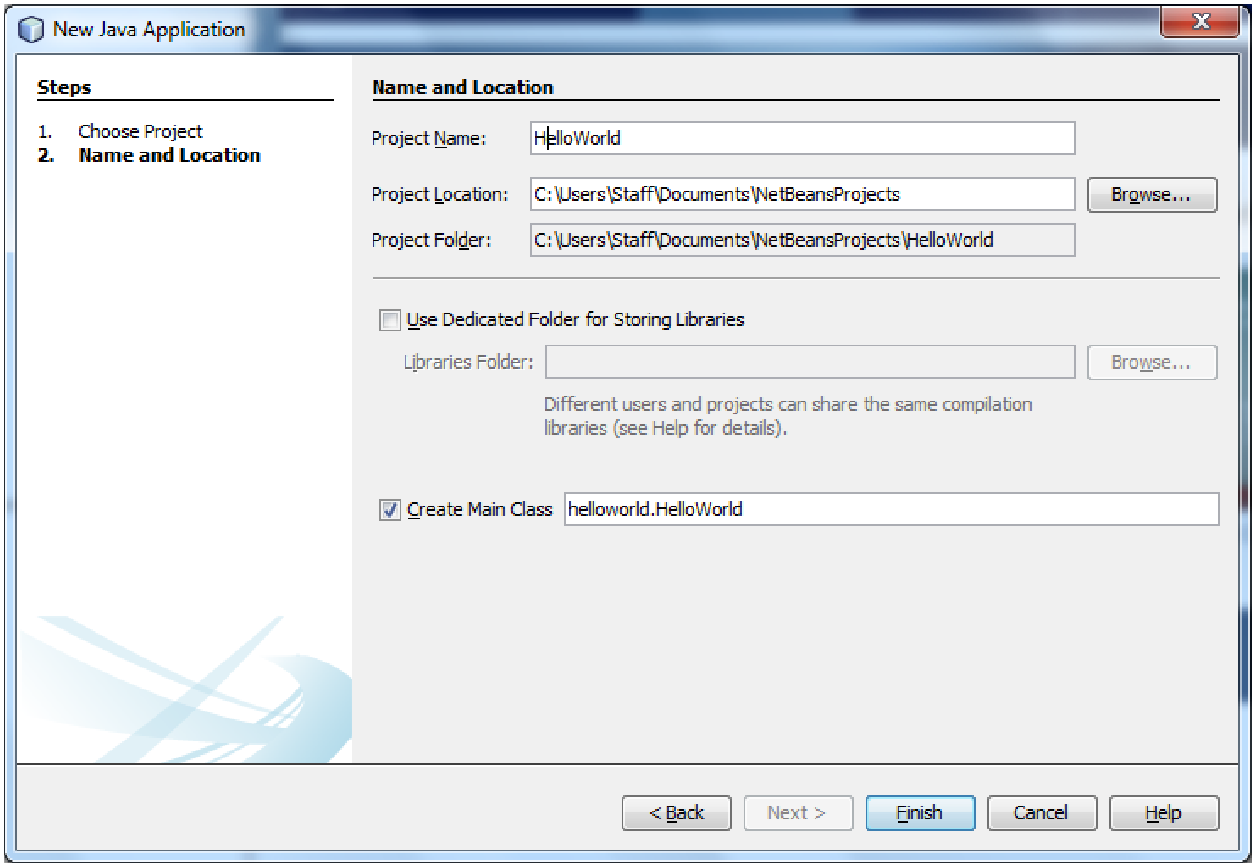
From the **File** menu, select make a **New Project**.



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.

From the Categories menu select **Java**. From the Projects menu select **Java Application**. Then press **Next**.

New Java Application



Project Name: HelloWorld

Project Location: C:\Users\Staff\Documents\NetBeansProjects

Project Folder: C:\Users\Staff\Documents\NetBeansProjects\HelloWorld

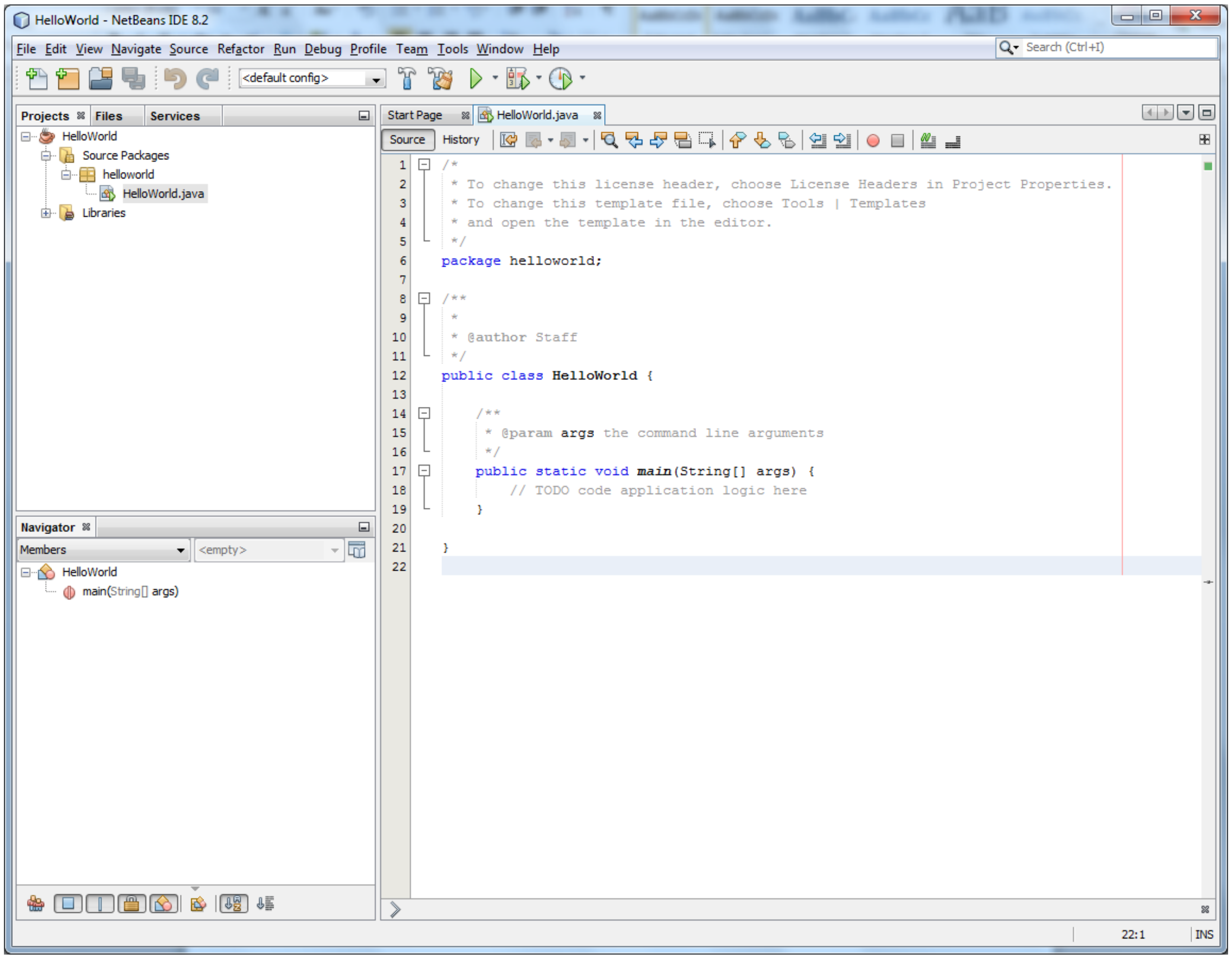
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

☑ Create Main Class helloworld.HelloWorld

The **Project Name** typed was HelloWorld. The NetBeans generated the rest of the values (including the Create Main Class). Click on **Finish** when done. It will take a few seconds as the project is created.

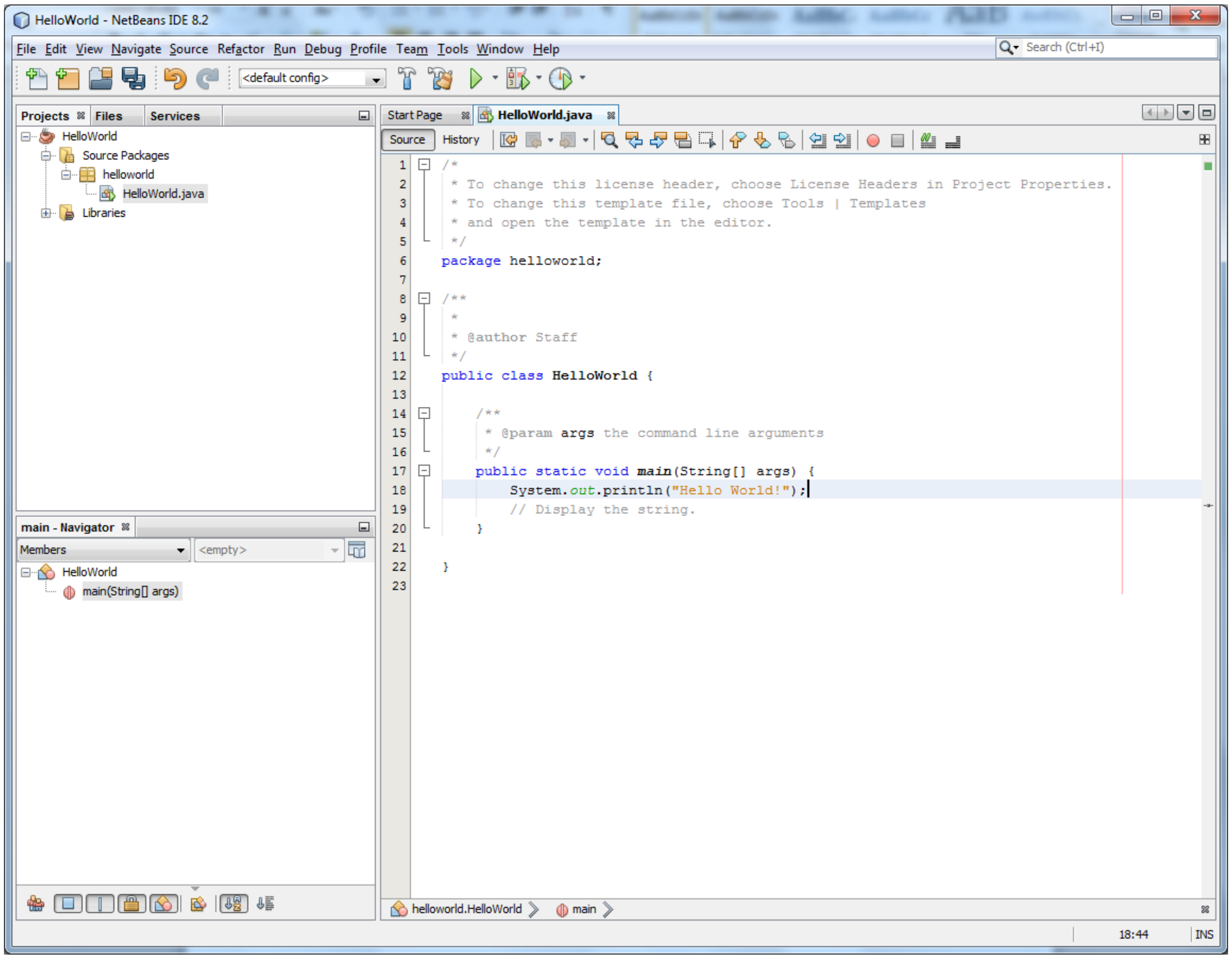


```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package helloworld;
7
8  /**
9   *
10  * @author Staff
11  */
12 public class HelloWorld {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20
21 }
22
```

Here is a blank project. We will now fill in some code to make the program do something. Replace line 18 with the following:

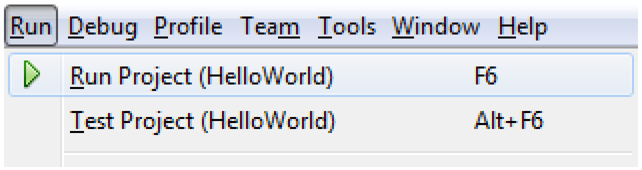
```
System.out.println("Hello World!");
```

Now we have a new program:



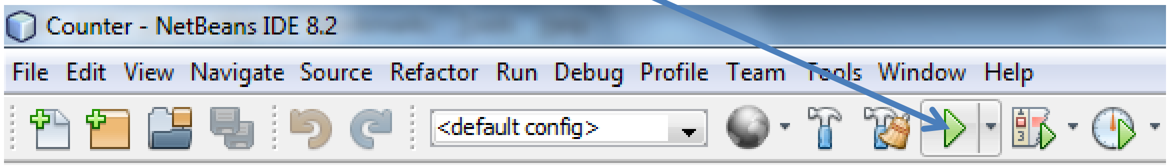
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package helloworld;

/**
 *
 * @author Staff
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        // Display the string.
    }

}
```

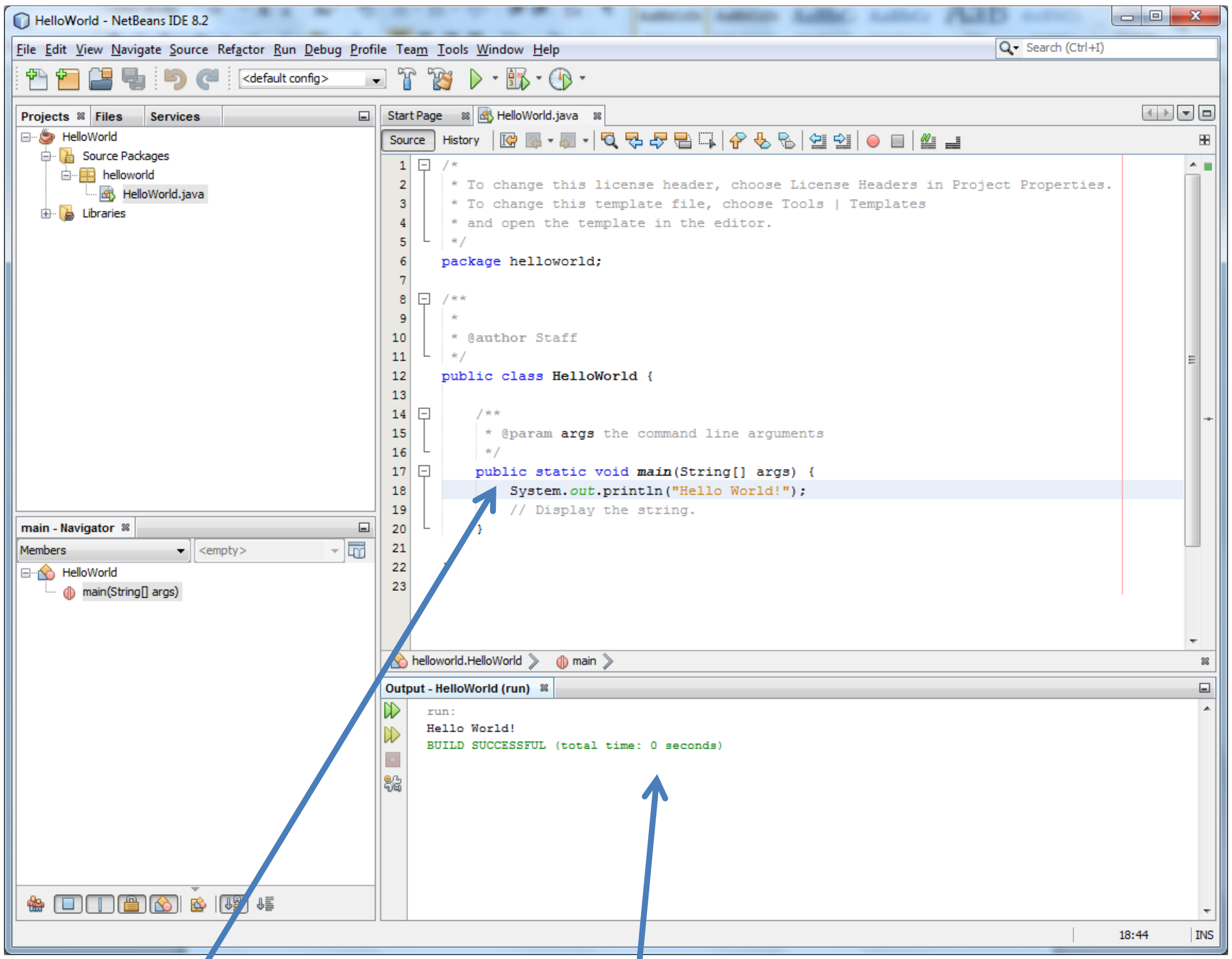
Running a NetBeans Project


Run Debug Profile Team Tools Window Help
Run Project (HelloWorld) F6
Test Project (HelloWorld) Alt+F6

From the **Run** menu, select **Run Project**. You could press the keyboard shortcut F6, if you prefer.

You are also able to press on the green arrow to Run your project.

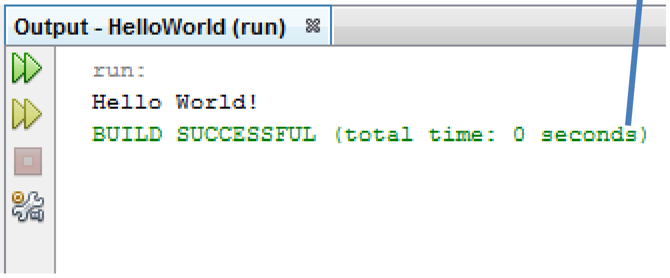

Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package helloworld;

/**
 *
 * @author Staff
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        // Display the string.
    }

}
```

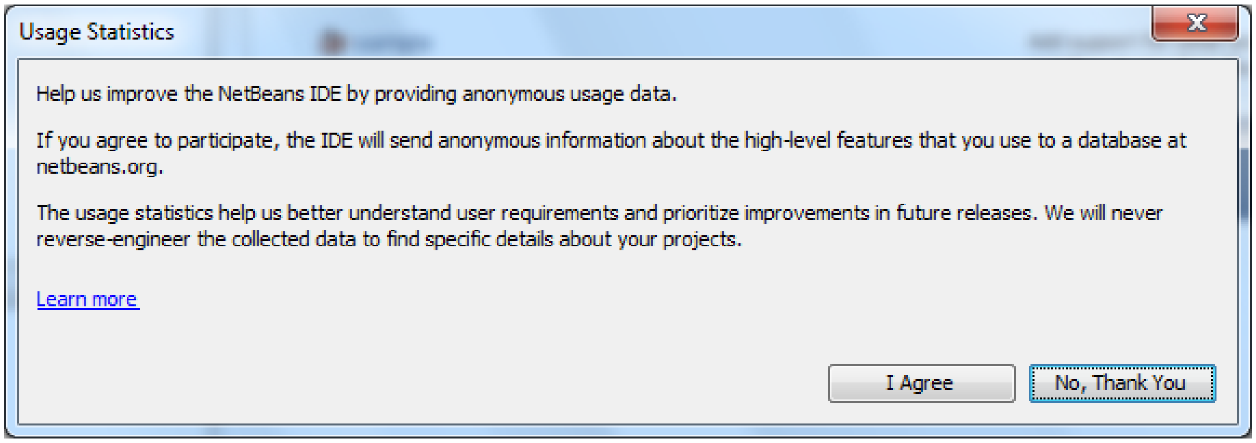
```
run:
Hello World!
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
System.out.println("Hello World!");
```



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

As shown above, the output from the Output panel in NetBeans.



Usage Statistics

Help us improve the NetBeans IDE by providing anonymous usage data.

If you agree to participate, the IDE will send anonymous information about the high-level features that you use to a database at netbeans.org.

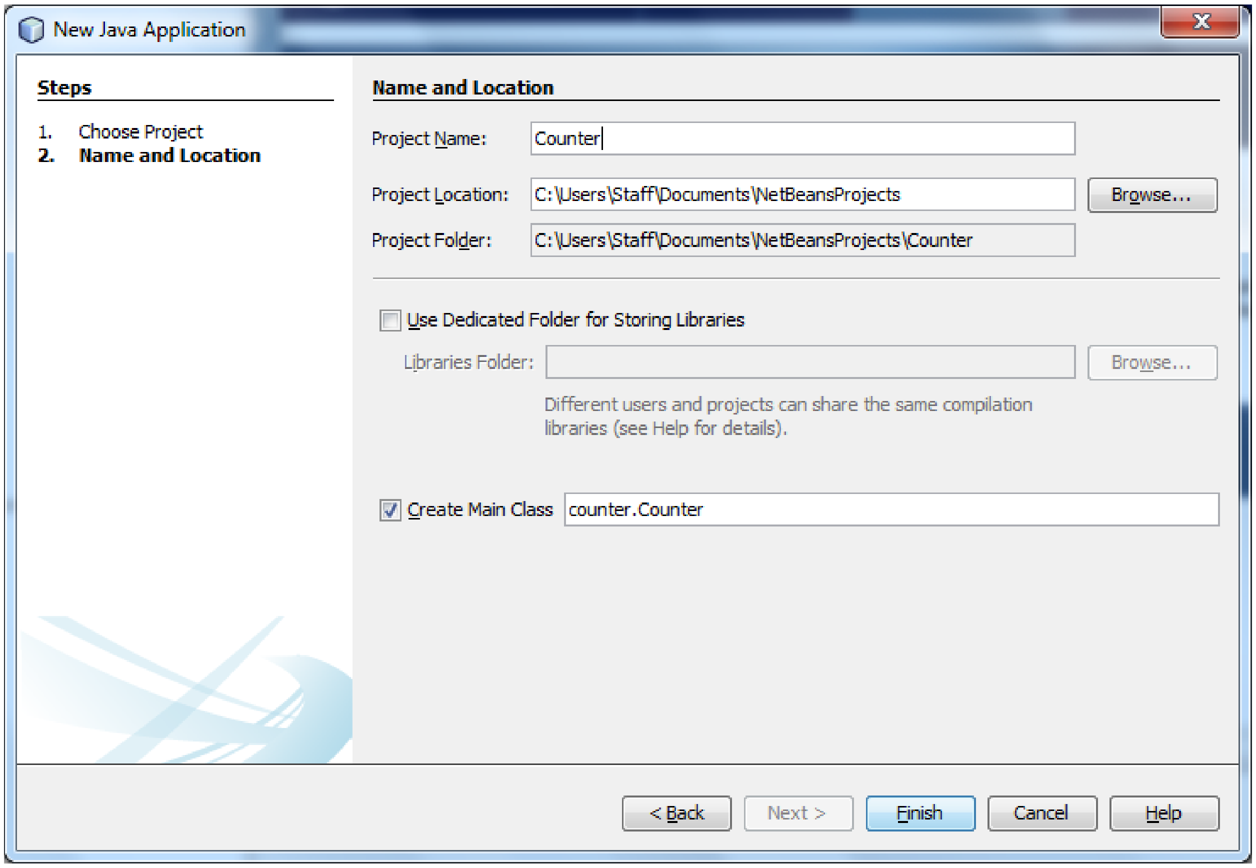
The usage statistics help us better understand user requirements and prioritize improvements in future releases. We will never reverse-engineer the collected data to find specific details about your projects.

Learn more

A popup window about Usage Statistics may appear when you start NetBeans. If you would like to participate in anonymous information being sent to NetBeans to help them improve the program, click on I Agree. Otherwise, you may wish to say **No, Thank You**.

Debugging Example

Let’s make a more complicated program. We will make a **New Java Application**. The Project Name will be called ***Counter***. Check on **Finish** when done.



Project Name: Counter

Project Location: C:\Users\Staff\Documents\NetBeansProjects

Project Folder: C:\Users\Staff\Documents\NetBeansProjects\Counter

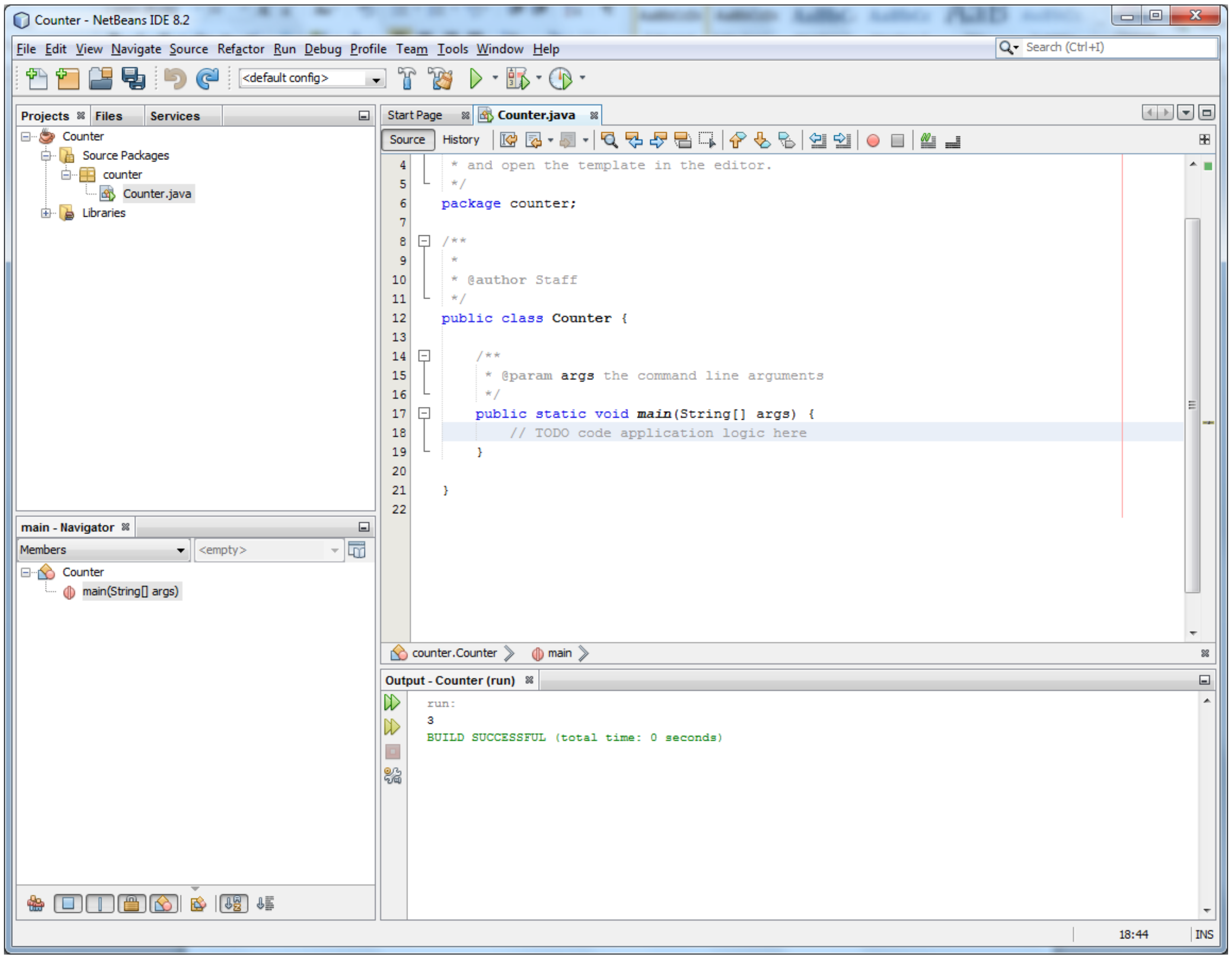
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

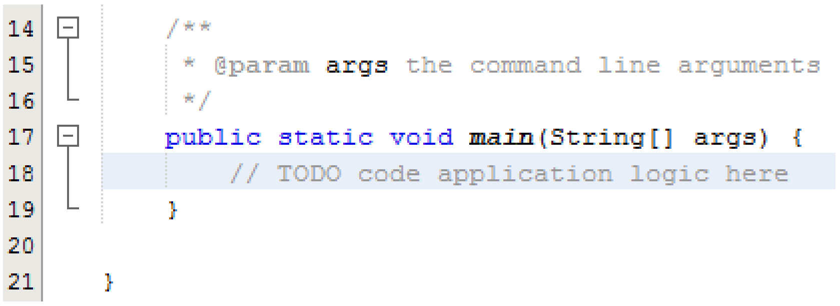
☑ Create Main Class counter.Counter

Now you have a Java Program created with a skeleton code for you to fill in:



```
 4   * and open the template in the editor.
 5   */
 6  package counter;
 7
 8  /**
 9   *
10   * @author Staff
11   */
12  public class Counter {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
22
```

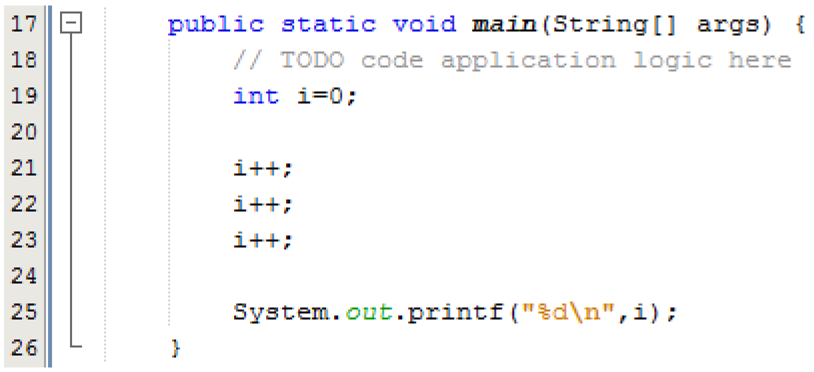
```
run:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
```

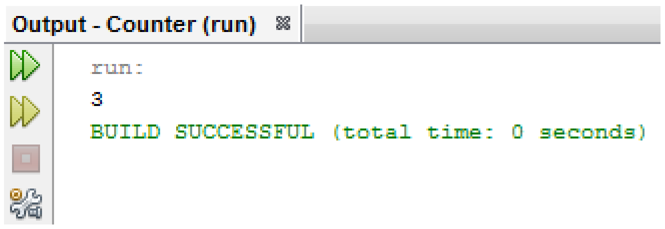
We will begin our program at line 19:

```
int i=0;

i++;
i++;
i++;

System.out.printf("%d\n",i);
```



```
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int i=0;
20
21          i++;
22          i++;
23          i++;
24
25          System.out.printf("%d\n",i);
26      }
```

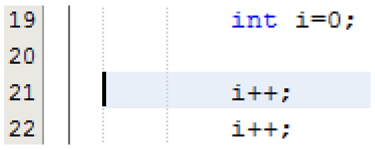
If we Run the program (Press F6), we see the following output:



```
run:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

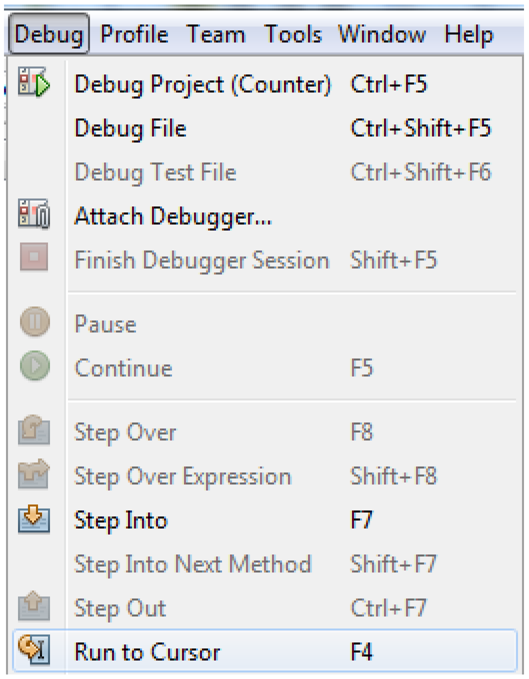
The value of i is displayed on the screen. But how did it become the value of 3. Let us debug the program to see what happens to the variables as the program runs.

Select the location the source code that you would like the program to stop at.

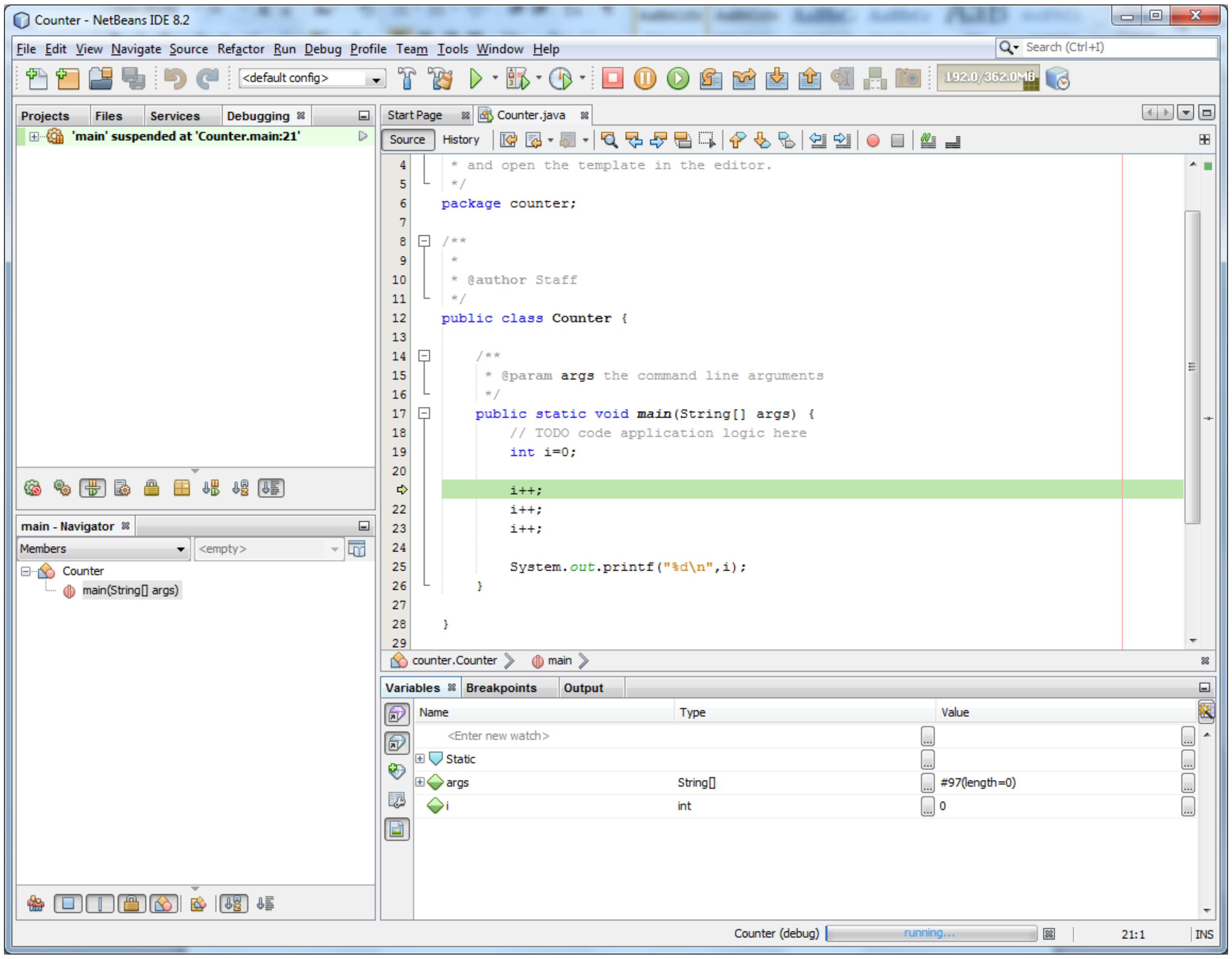


```
19          int i=0;
20
21          i++;
22          i++;
```

In this case we selected in the source code at line 21.


Debug Profile Team Tools Window Help
Debug Project (Counter) Ctrl+F5
Debug File Ctrl+Shift+F5
Debug Test File Ctrl+Shift+F6
Attach Debugger...
Finish Debugger Session Shift+F5
Pause
Continue F5
Step Over F8
Step Over Expression Shift+F8
Step Into F7
Step Into Next Method Shift+F7
Step Out Ctrl+F7
Run to Cursor F4

From the **Debug** menu, select **Run to Cursor** (or Press the keyboard shortcut F4).

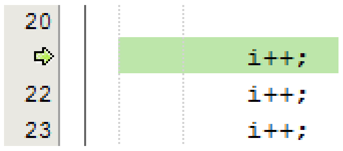


'main' suspended at 'Counter.main:21'

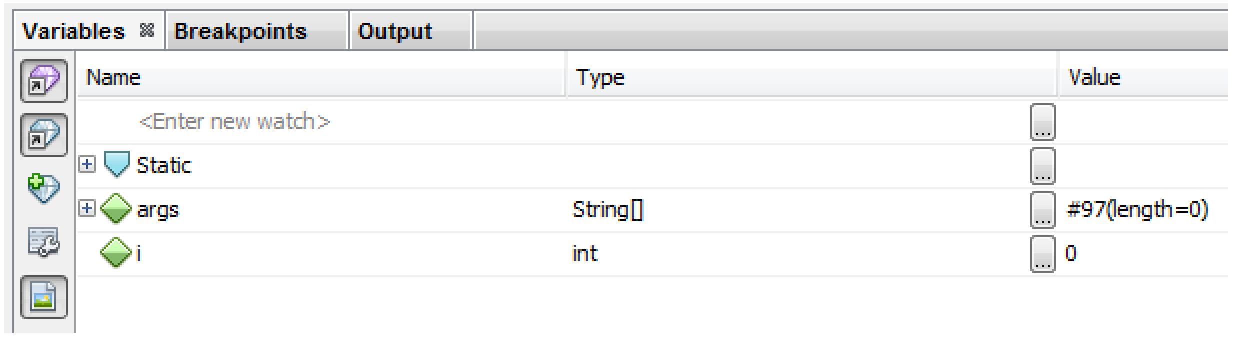
```
 * and open the template in the editor.
 */
package counter;

/**
 *
 * @author Staff
 */
public class Counter {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int i=0;

        i++;
        i++;
        i++;

        System.out.printf("%d\n",i);
    }

}
```

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	0

The debugger stopped the program. We can inspect the variables. Make sure the **Variables** tab in the lower right in part of the window is selected.

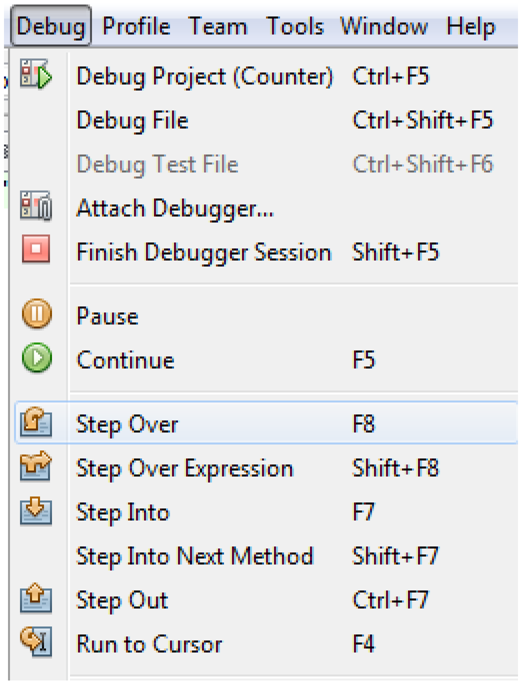


```
20
⇨       i++;
22      i++;
23      i++;
```



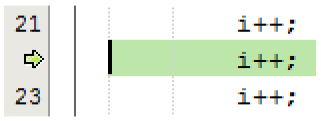
Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	0

Notice at the variable i, has the value of 0.

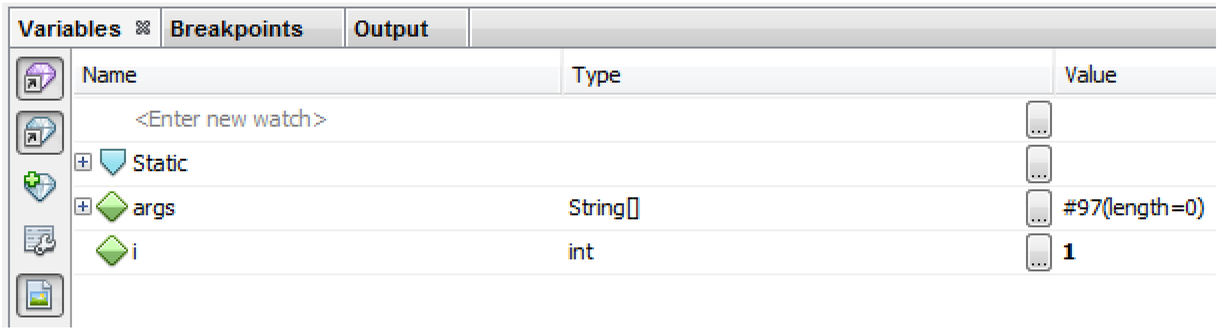

Debug
Profile
Team
Tools
Window
Help
Debug Project (Counter) Ctrl+F5
Debug File Ctrl+Shift+F5
Debug Test File Ctrl+Shift+F6
Attach Debugger...
Finish Debugger Session Shift+F5
Pause
Continue F5
Step Over F8
Step Over Expression Shift+F8
Step Into F7
Step Into Next Method Shift+F7
Step Out Ctrl+F7
Run to Cursor F4

From the **Debug** menu, select **Step Over** (easier to press F8). The debugger will step over this instruction to the next instruction.

As you write more complex programs, you may need to Step Into (to see inside of a method).



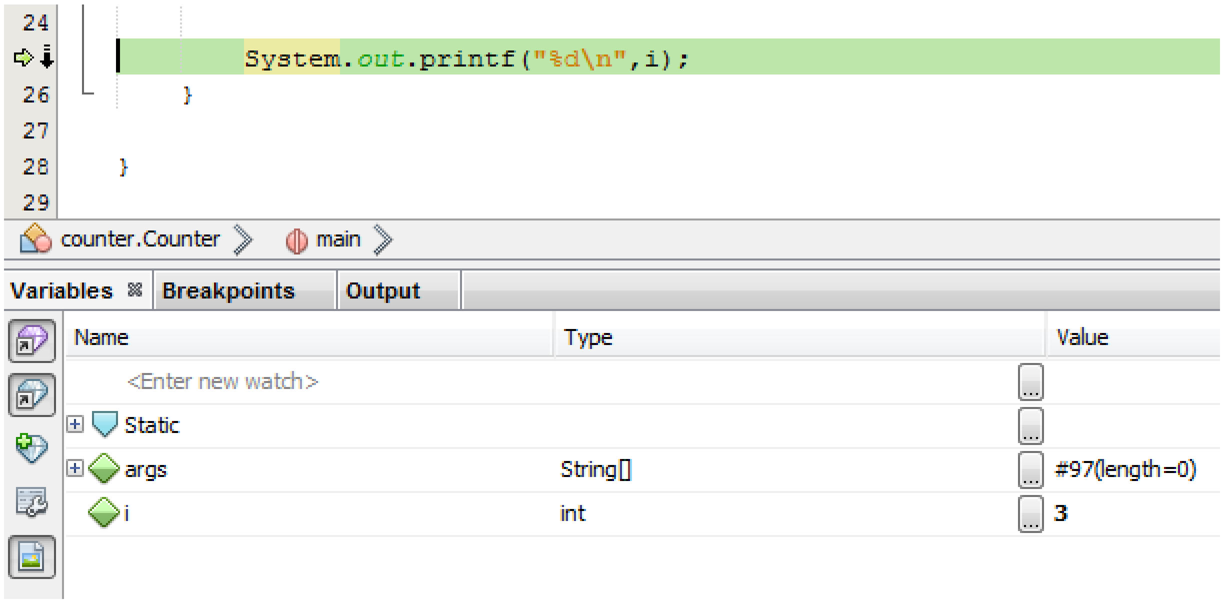
```
            i++;
            i++;
            i++;
```



Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	**1**

The value of i became 1. Press F8 again and i becomes 2.

If we select line 25 of the program, and select from the debug menu to **Run to Cusor** (F4).

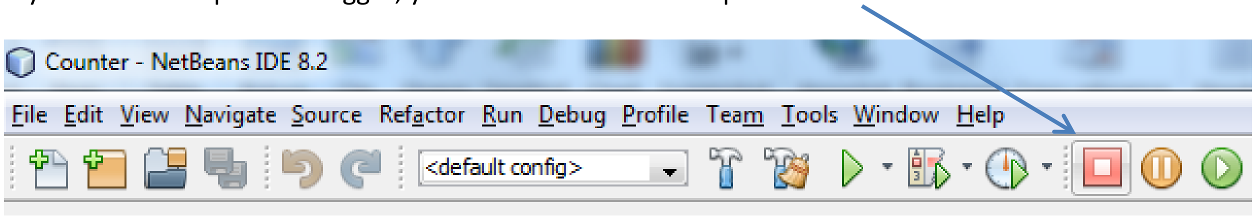


```
            System.out.printf("%d\n",i);
        }

    }
```

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	3

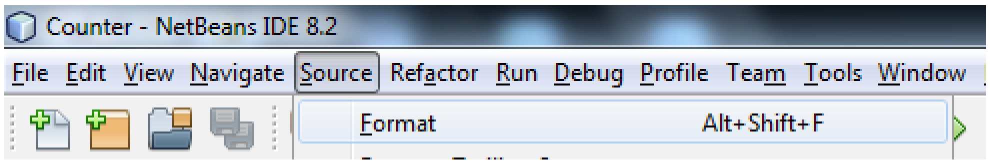
It shows that indeed, i has the value of 3, when it will be displayed to the console.

If you want to stop the debugger, you can select the Red Stop button.


Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>

Source Format

You can have NetBeans automatically format the source code for you. From the Source menu, select Format.


Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window
Format Alt+Shift+F

It will automatically indent, put spaces in between variables and assignments, and fix other formatting issues.

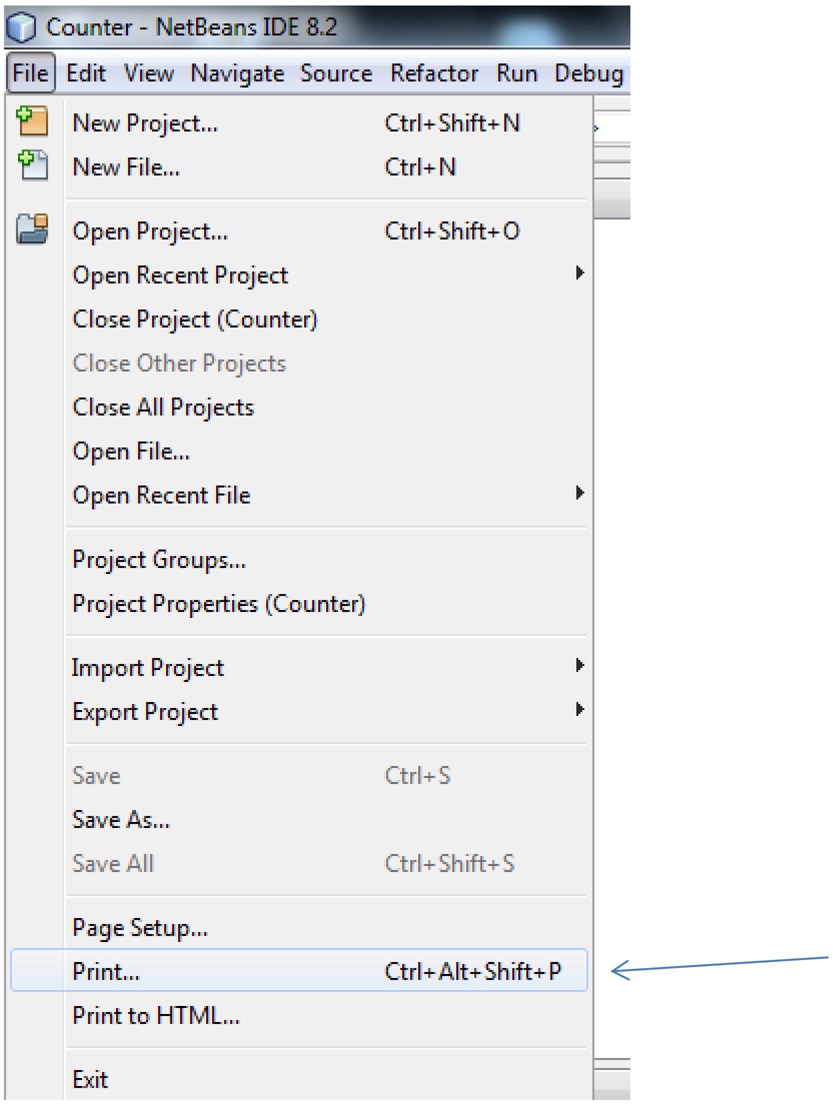
Before formatting is applied:

```
   public static void main(String[] args)
 {
       // TODO code application logic here
      int i=0;

       i++;
       i++;
       i++;

       System.out.printf("%d\n",i);
   }
```

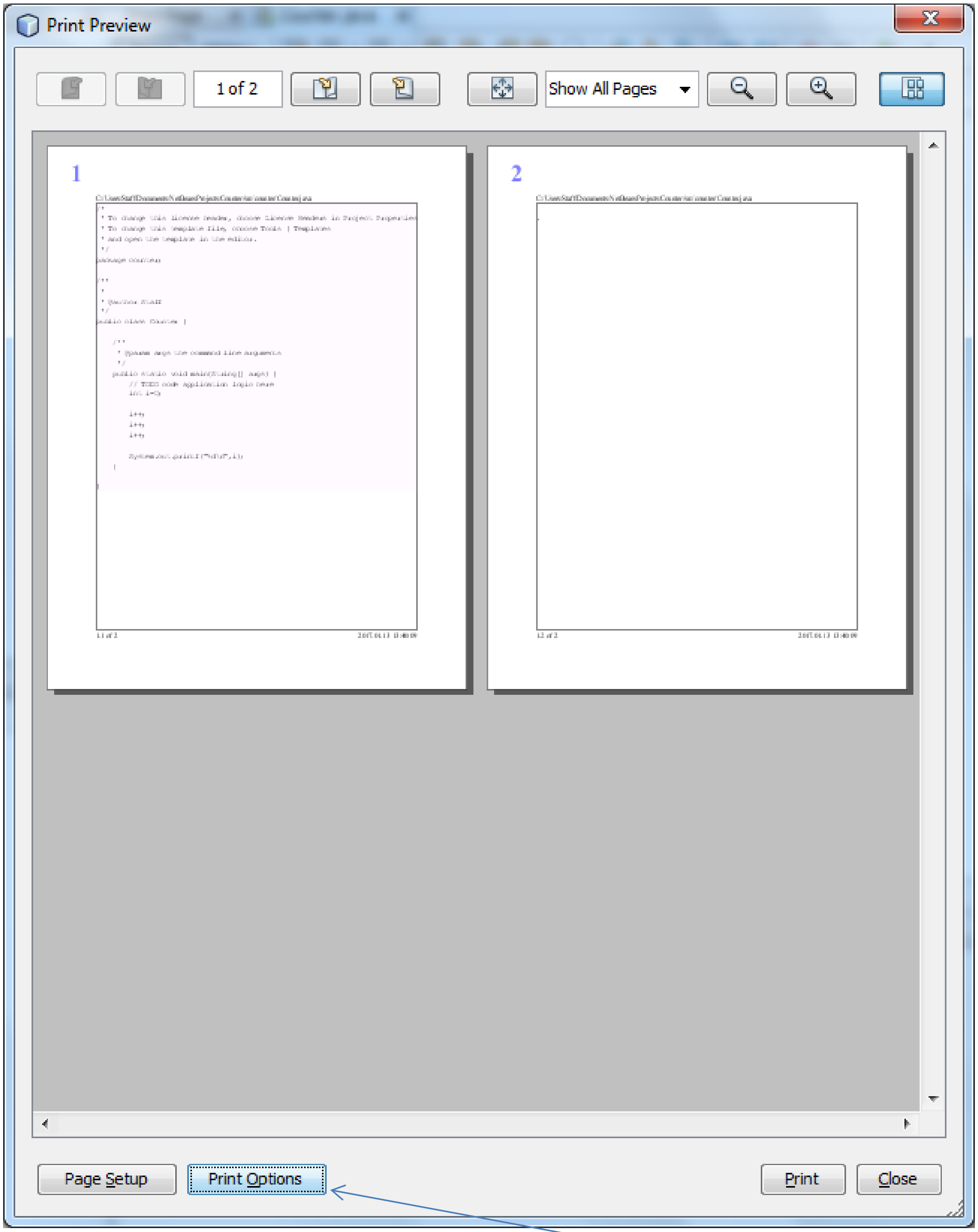
After formatting:

```
public static void main(String[] args) {
    // TODO code application logic here
    int i = 0;

    i++;
    i++;
    i++;

    System.out.printf("%d\n", i);
}
```

Printing



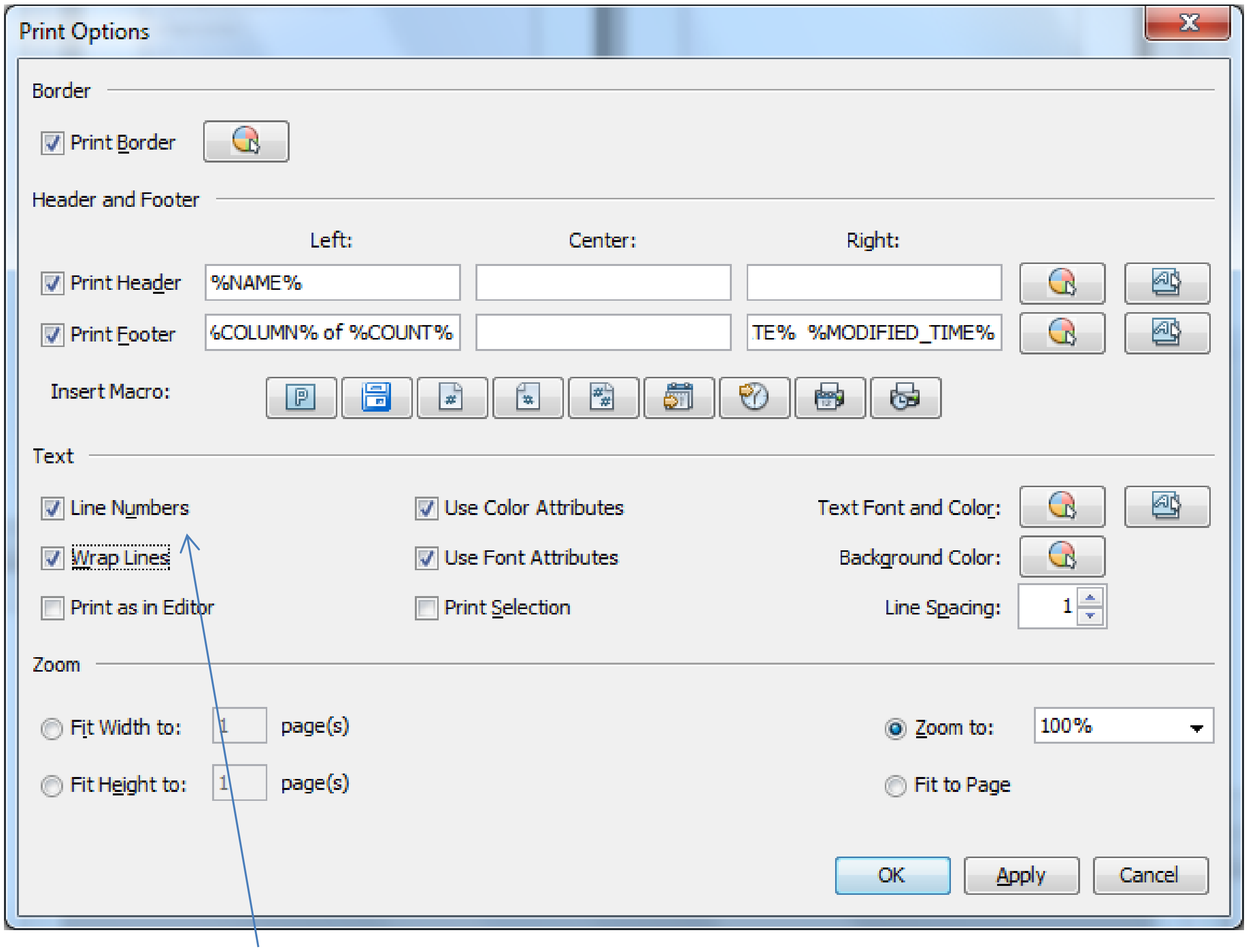
From the **File** menu, select **Print**.



C:\Users\Staff\Documents\NetBeansProjects\Counter\src\counter\Counter.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package counter;

/**
 *
 * @author Staff
 */
public class Counter {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int i=0;

        i++;
        i++;
        i++;

        System.out.printf("%d\n",i);
    }

}
```

The preview is missing the line numbers. Let us add them. Select **Print Options**.



Print Options

Border

[✓] Print Border

Header and Footer

	Left:	Center:	Right:
[✓] Print Header	%NAME%		
[✓] Print Footer	%COLUMN% of %COUNT%		TE% %MODIFIED_TIME%

Insert Macro:

Text

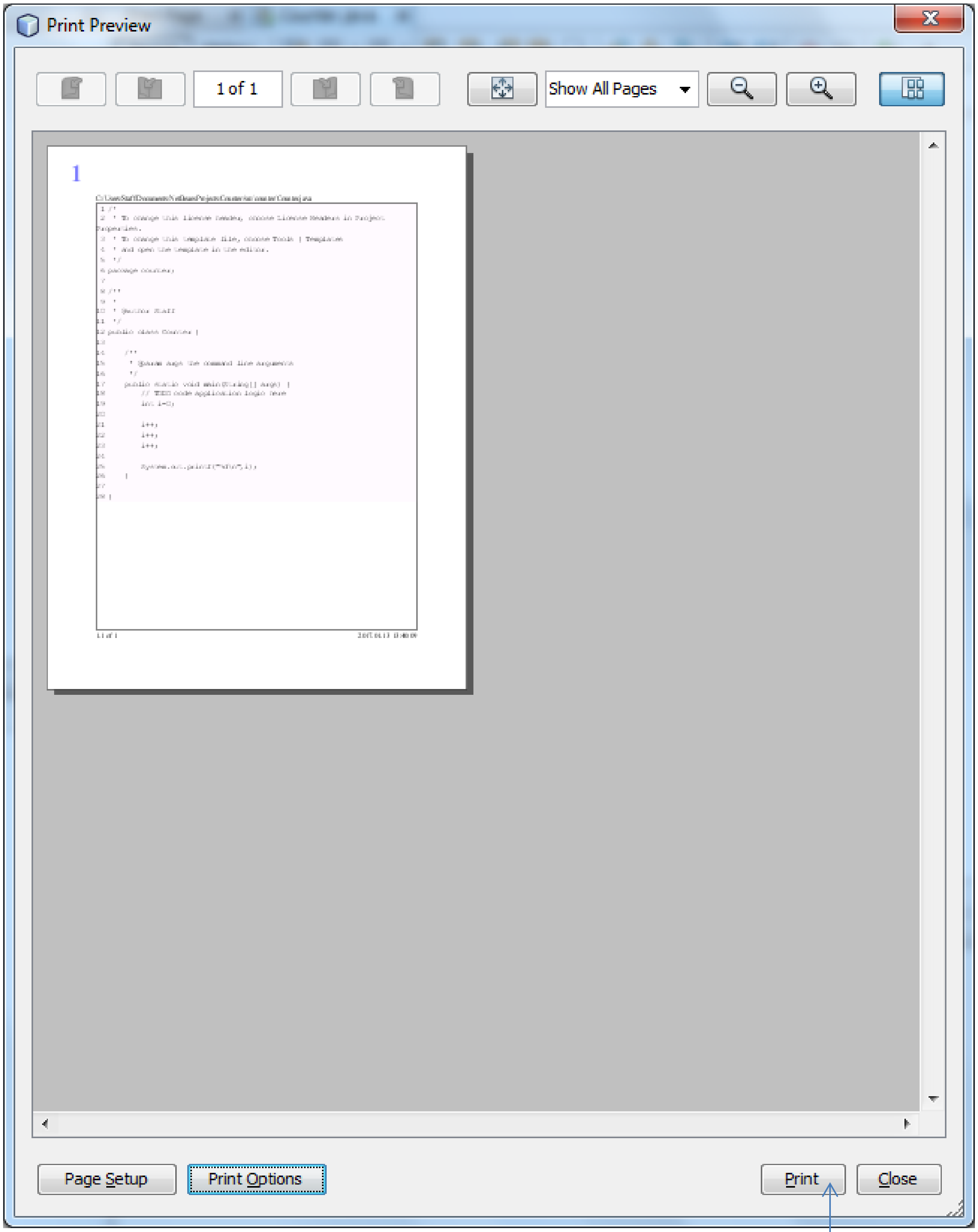
[✓] Line Numbers
[✓] Wrap Lines
[ ] Print as in Editor
[✓] Use Color Attributes
[✓] Use Font Attributes
[ ] Print Selection

Text Font and Color:
Background Color:
Line Spacing: 1

Zoom

( ) Fit Width to: 1 page(s)
( ) Fit Height to: 1 page(s)
(●) Zoom to: 100%
( ) Fit to Page

Select **Line Numbers** and **Wrap Lines**. Then press **OK**.



C:\Users\Staff\Documents\NetBeansProjects\Counter\src\counter\Counter.java

```
1 /*
2  * To change this license header, choose License Headers in Project
Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package counter;
7
8 /**
9  *
10  * @author Staff
11  */
12 public class Counter {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int i=0;
20
21         i++;
22         i++;
23         i++;
24
25         System.out.printf("%d\n",i);
26     }
27
28 }
```

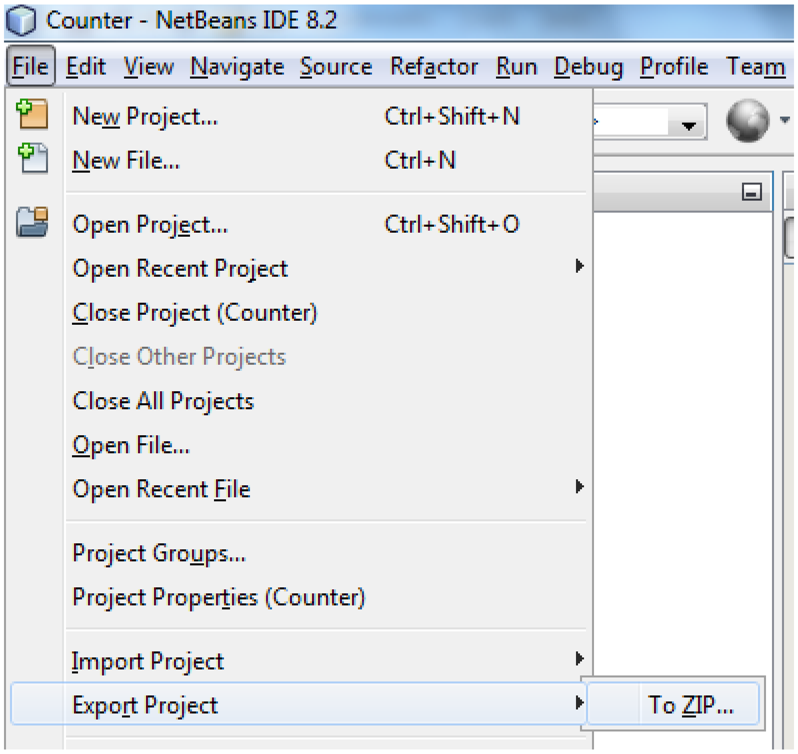
1 of 1 [illegible]

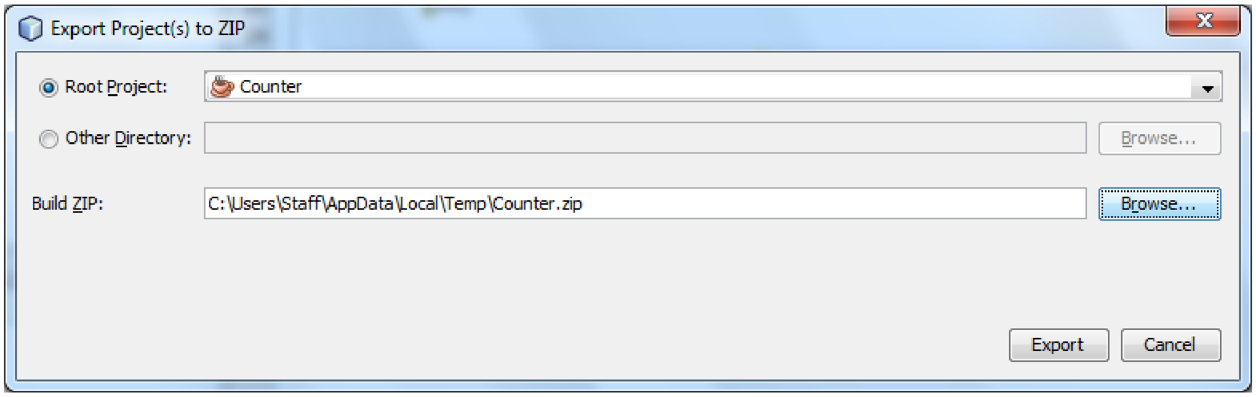
Now the preview shows the line numbers and the text wrapped on the page. Press **Print** to Print your file.

Exporting / Importing

If you want to move your project to another machine, you can Export it.

From the **File** menu, select **Export Project**, **To ZIP**.



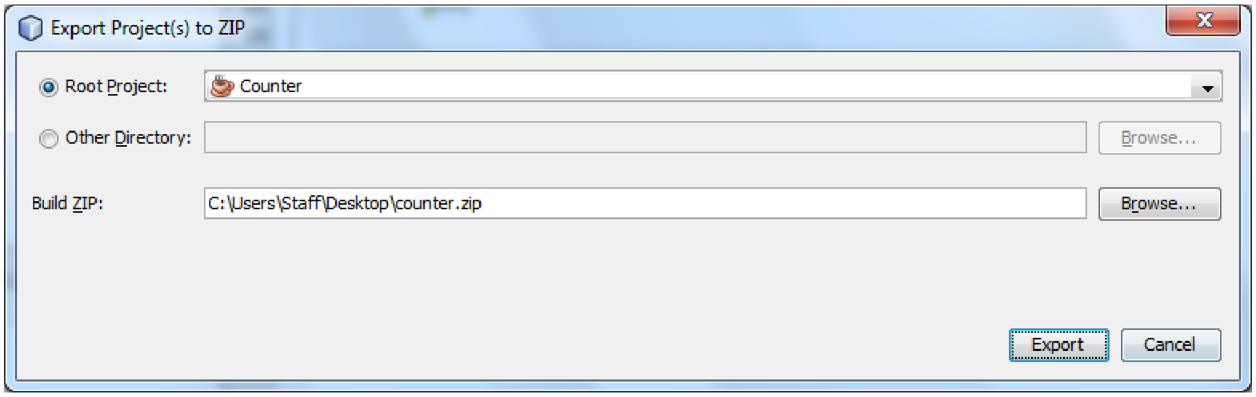


Root Project: Counter

Other Directory:

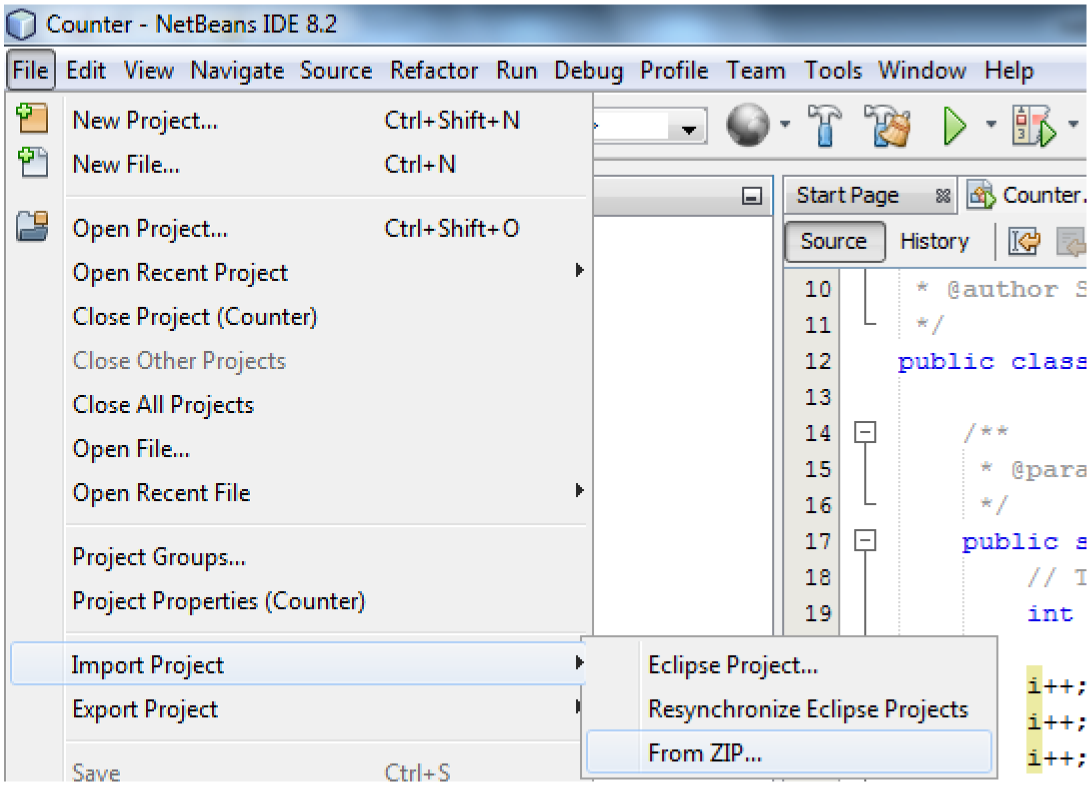
Build ZIP: C:\Users\Staff\AppData\Local\Temp\Counter.zip

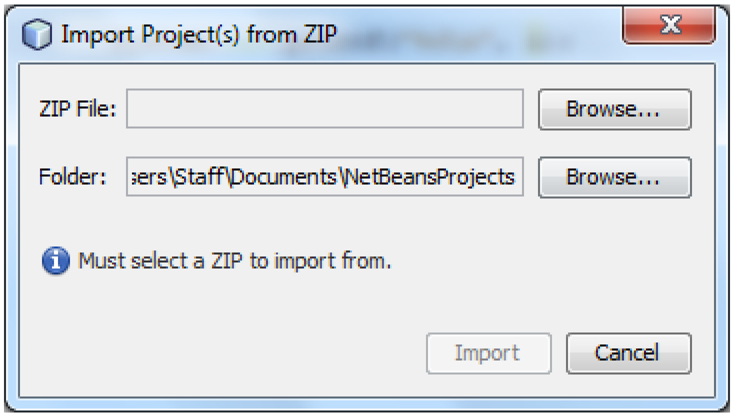
Select **Browse** to specify a location to save the zip file.



Specify the location to save and then press **Export**.

To import the project back on a new machine (or if you used the export as a backup and wish to restore back the original), from the **File** menu, select **Import Project**, **From ZIP**.

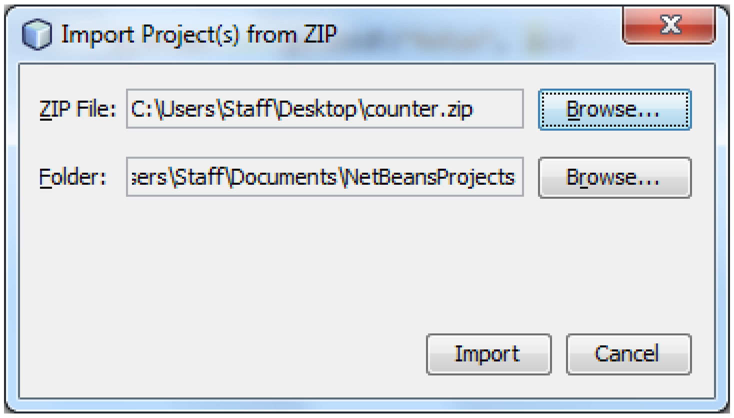




ers\Staff\Documents\NetBeansProjects

Must select a ZIP to import from.

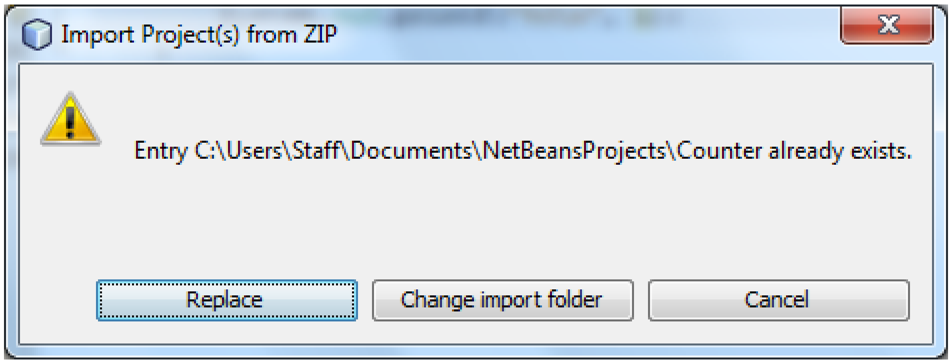
The top Browse button will let you select the ZIP file that contains your project. The bottom Browse button selects the folder that contains where your NetBeans projects are stored. Leave the bottom Folder location alone. Select from the top Browse button where to find the ZIP file.



C:\Users\Staff\Desktop\counter.zip

ers\Staff\Documents\NetBeansProjects

Press Import once things are set properly.

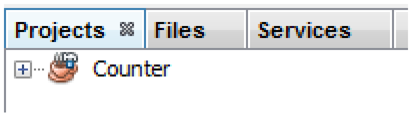


Entry C:\Users\Staff\Documents\NetBeansProjects\Counter already exists.

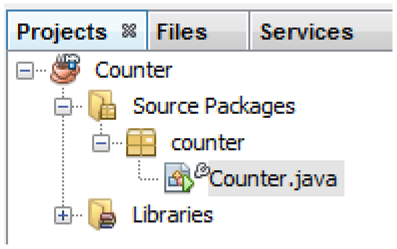
If the project already exists it will replace (overwrite it). Do this only if you are restoring to replace the current project files with the files contained in the ZIP file. Otherwise you can Chance Import Folder to place the files in a new folder. Otherwise press Cancel to not do the import sequence.

Multiple Class Files

The project has an expandable/collapsible tree listing of the files in your project.

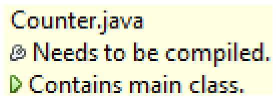


Click on the **+** symbol to expand it.

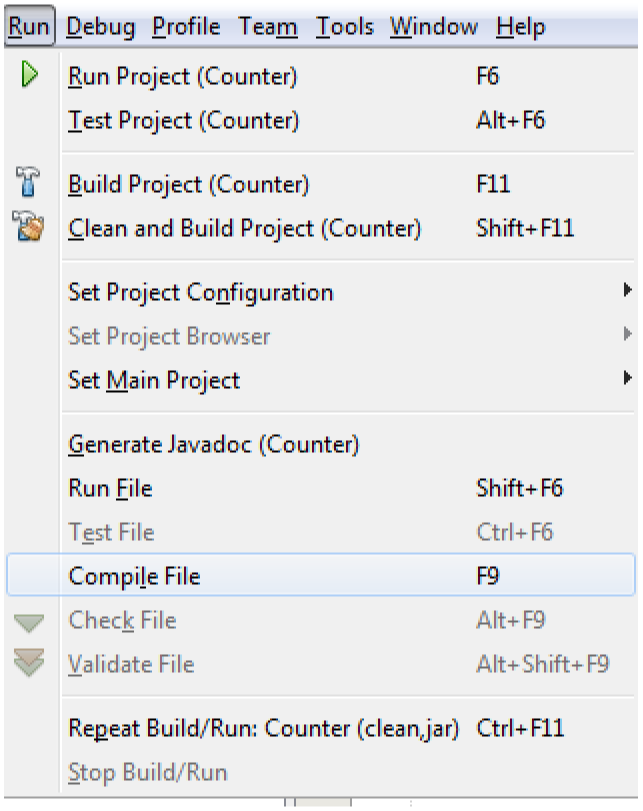


Double click on the file name to open it.

The symbols next to the file name mean:



Counter.java
Needs to be compiled.
Contains main class.

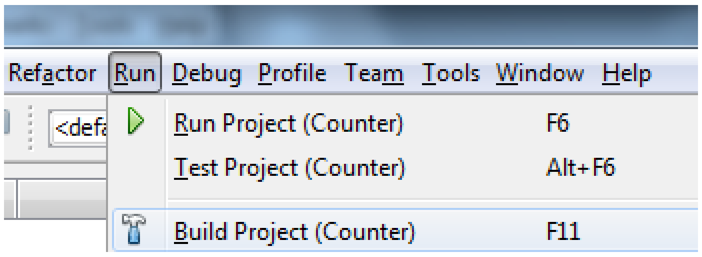


Run Debug Profile Team Tools Window Help
Run Project (Counter) F6
Test Project (Counter) Alt+F6
Build Project (Counter) F11
Clean and Build Project (Counter) Shift+F11
Set Project Configuration
Set Project Browser
Set Main Project
Generate Javadoc (Counter)
Run File Shift+F6
Test File Ctrl+F6
Compile File F9
Check File Alt+F9
Validate File Alt+Shift+F9
Repeat Build/Run: Counter (clean,jar) Ctrl+F11
Stop Build/Run

You can press **F9** to compile the file (or from the **Run** menu, select **Compile File**).

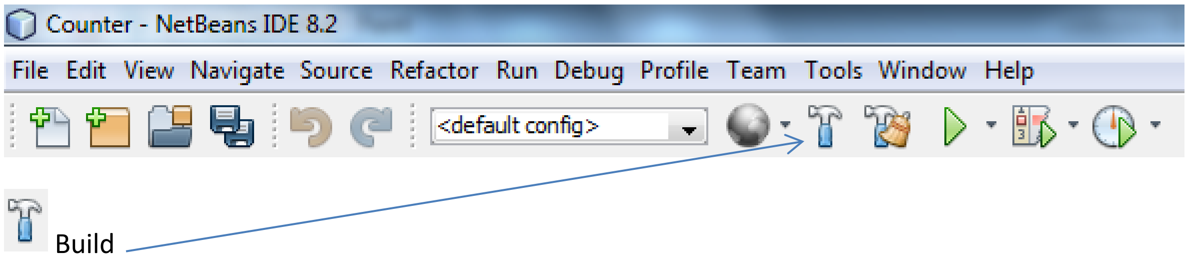
Build/JAR file

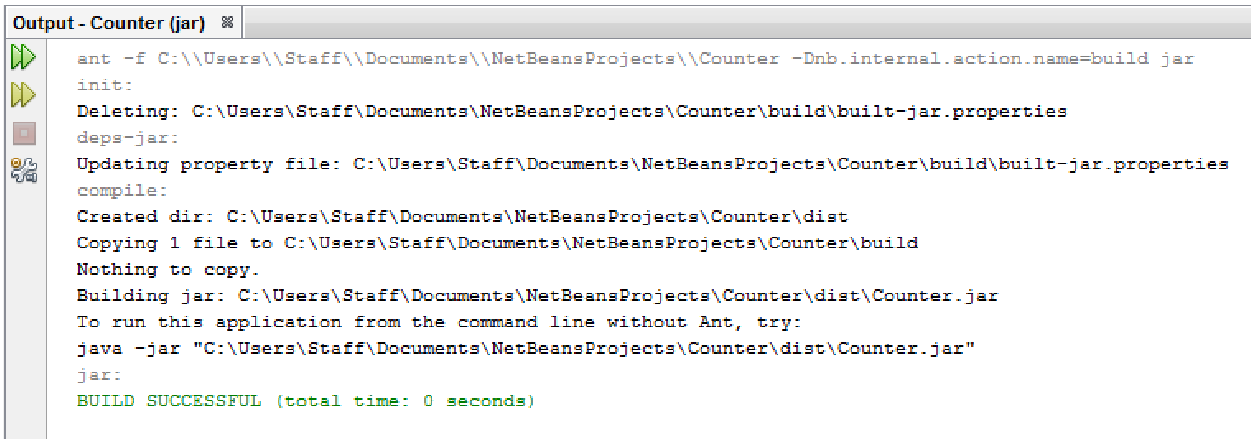
If your project contains multiple files, you can build a **JAR** (**J**ava **AR**chive) file that contains the all your class files and associated resources (such as media or data files) for your profile into a single file for distribution. To do so, you can select from the **Run**, **Build Project.**



Refactor Run Debug Profile Team Tools Window Help
<defa
Run Project (Counter) F6
Test Project (Counter) Alt+F6
Build Project (Counter) F11

From the toolbar you can select the Hammer icon to Build as well.

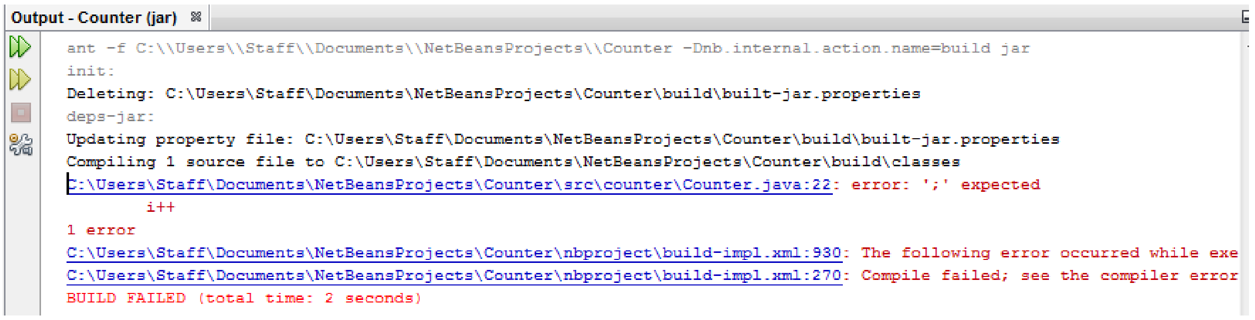

Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Build



```
ant -f C:\\Users\\Staff\\Documents\\NetBeansProjects\\Counter -Dnb.internal.action.name=build jar
init:
Deleting: C:\Users\Staff\Documents\NetBeansProjects\Counter\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Staff\Documents\NetBeansProjects\Counter\build\built-jar.properties
compile:
Created dir: C:\Users\Staff\Documents\NetBeansProjects\Counter\dist
Copying 1 file to C:\Users\Staff\Documents\NetBeansProjects\Counter\build
Nothing to copy.
Building jar: C:\Users\Staff\Documents\NetBeansProjects\Counter\dist\Counter.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\Staff\Documents\NetBeansProjects\Counter\dist\Counter.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

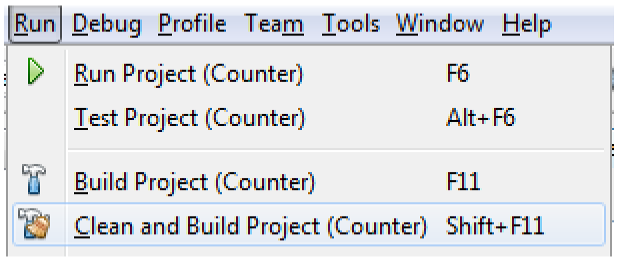
You should get a Build Successful message. If your code contained errors, you would see them below.

Such as for the following:

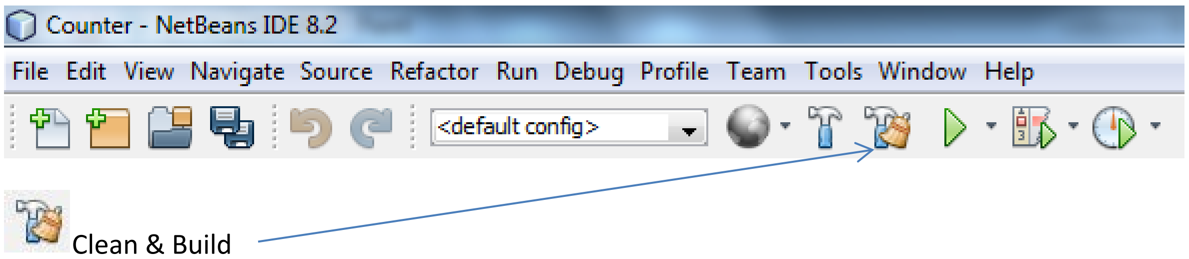


```
ant -f C:\\Users\\Staff\\Documents\\NetBeansProjects\\Counter -Dnb.internal.action.name=build jar
init:
Deleting: C:\Users\Staff\Documents\NetBeansProjects\Counter\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Staff\Documents\NetBeansProjects\Counter\build\built-jar.properties
Compiling 1 source file to C:\Users\Staff\Documents\NetBeansProjects\Counter\build\classes
C:\Users\Staff\Documents\NetBeansProjects\Counter\src\counter\Counter.java:22: error: ';' expected
           i++
1 error
C:\Users\Staff\Documents\NetBeansProjects\Counter\nbproject\build-impl.xml:930: The following error occurred while exe
C:\Users\Staff\Documents\NetBeansProjects\Counter\nbproject\build-impl.xml:270: Compile failed; see the compiler error
BUILD FAILED (total time: 2 seconds)
```

If you want to Clean (erase the compiled class files) and then Build the project completely, you can select **Clean and Build Project**, from the **Run** menu. You should do this, if you previously built a project and want to assure that newly compiled classes are in the project.


Run Debug Profile Team Tools Window Help
Run Project (Counter) F6
Test Project (Counter) Alt+F6
Build Project (Counter) F11
Clean and Build Project (Counter) Shift+F11

From the toolbar you can select the Hammer/Broom icon to Clean & Build as well.


Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Clean & Build

The JAR file is located in the dist folder of your project's directory. From a **Command Prompt** you can run it by doing the following command (in our case):

```
java -jar "C:\Users\Staff\Documents\NetBeansProjects\Counter\dist\Counter.jar"
```

This jar file can then be used to run your project from any machine that has a JVM (Java Virtual Machine).

NetBeans will create a README.TXT which gives the precise command that can be used to invoke it. The README.TXT file is best opened in Wordpad (on Windows).

Javadoc

You should get into the habit of writing your comments in the Javadoc format:

- https://www.tutorialspoint.com/java/java_documentation.htm
- https://en.wikipedia.org/wiki/Javadoc
- http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

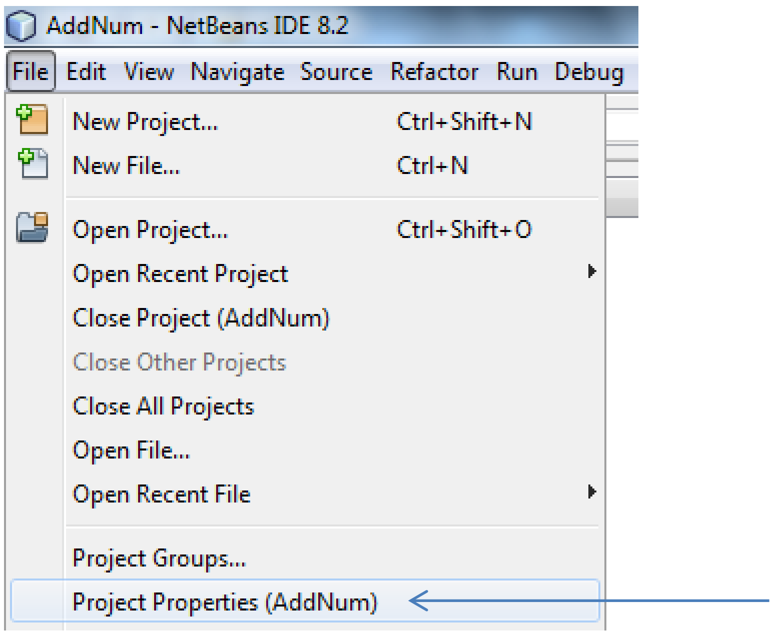
Traditionally /* and */ are used to indicate normal text comments to the compiler. However, Javadoc comments have an opening tag (also known as the begin-comment delimiter) with **two asterisks.**

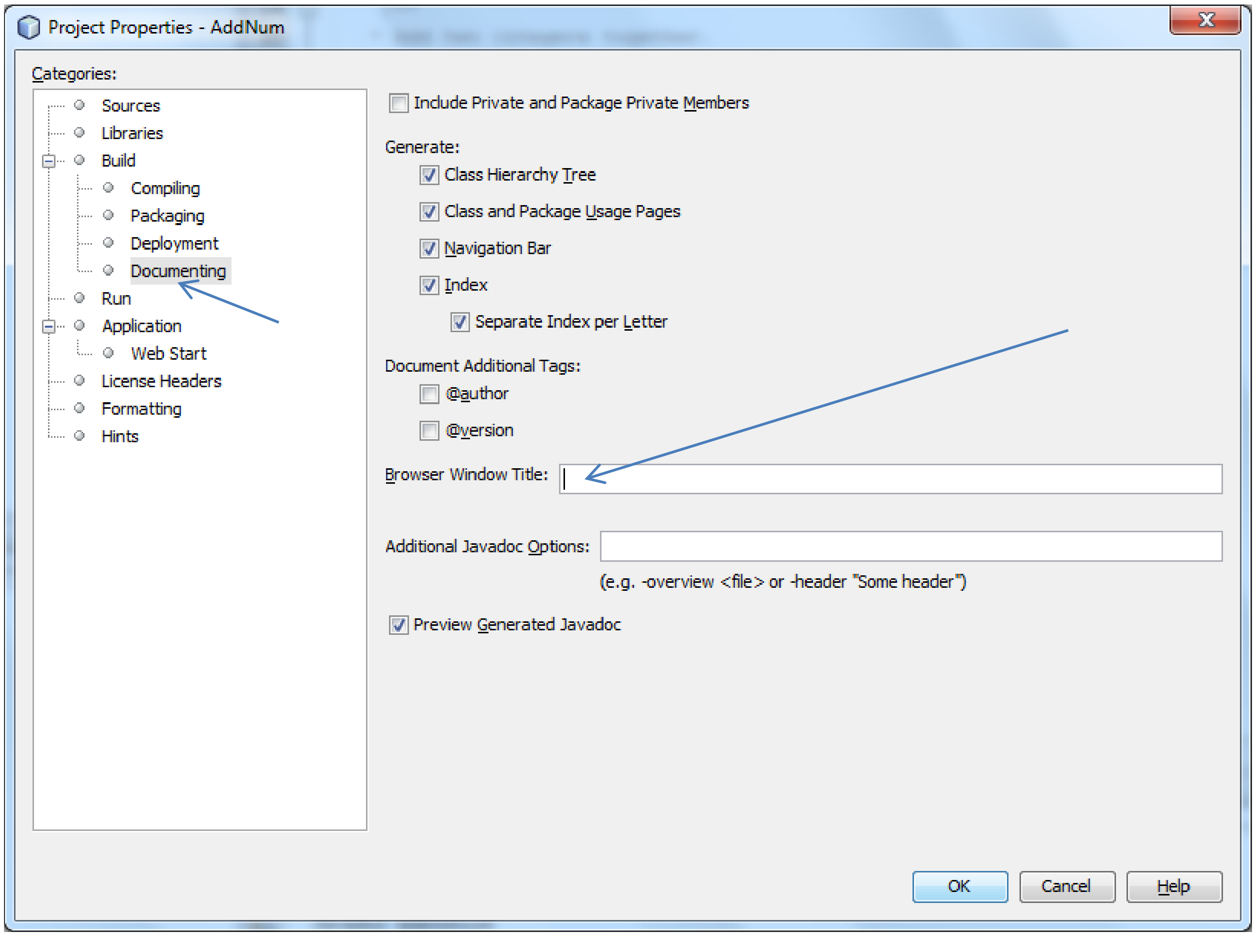
```
For example:
/** I am a Javadoc comment. */
```

Here is an example for a method.

```
/**
* Add two integers together.
* @param numA First number to add
* @param numB  Second number to add
* @return int Sum of numA and numB.
*/
public int addNum(int numA, int numB) {
   return numA + numB;
}
```

When generating Javadoc files, you will get message *Warning: Leaving out empty argument '-windowtitle'*, if you do not set the document title. You can ignore this. However if you want to add more to the title than just the package name, go to the **Properties** of the project. From the **File** menu, select **Project Properties**.


AddNum - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project (AddNum)
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties (AddNum)



☐ Include Private and Package Private Members

Generate:

☑ Class Hierarchy Tree

☑ Class and Package Usage Pages

☑ Navigation Bar

☑ Index

☑ Separate Index per Letter

Document Additional Tags:

☐ @author

☐ @version

Browser Window Title:

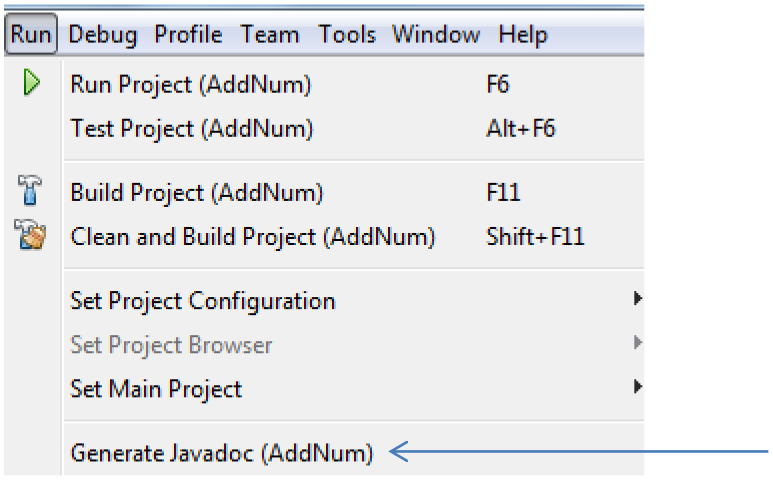
Additional Javadoc Options:

(e.g. -overview <file> or -header "Some header")

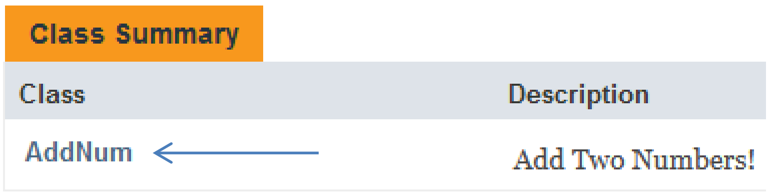
☑ Preview Generated Javadoc

From the Build category, select **Documenting**. Then enter in a **Browser Window Title**.

To generate the Javadoc information for your project, from the **Run** menu select **Generate Javadoc**.

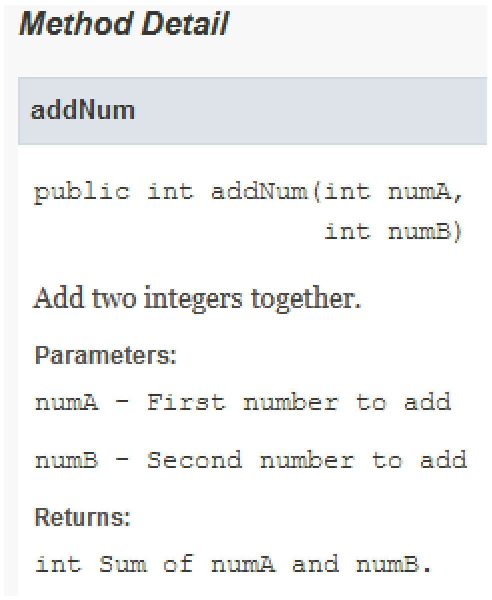

Run
Debug
Profile
Team
Tools
Window
Help
Run Project (AddNum) F6
Test Project (AddNum) Alt+F6
Build Project (AddNum) F11
Clean and Build Project (AddNum) Shift+F11
Set Project Configuration
Set Project Browser
Set Main Project
Generate Javadoc (AddNum)

That will open up the Javadoc files for your project. For example with AddNum, it will have a description of the class.



Class	Description
AddNum	Add Two Numbers!

Click on the class name to get more information on the class.



Method Detail

addNum

```
public int addNum(int numA,
                  int numB)
```

Add two integers together.

Parameters:

`numA - First number to add`

`numB - Second number to add`

Returns:

`int Sum of numA and numB.`

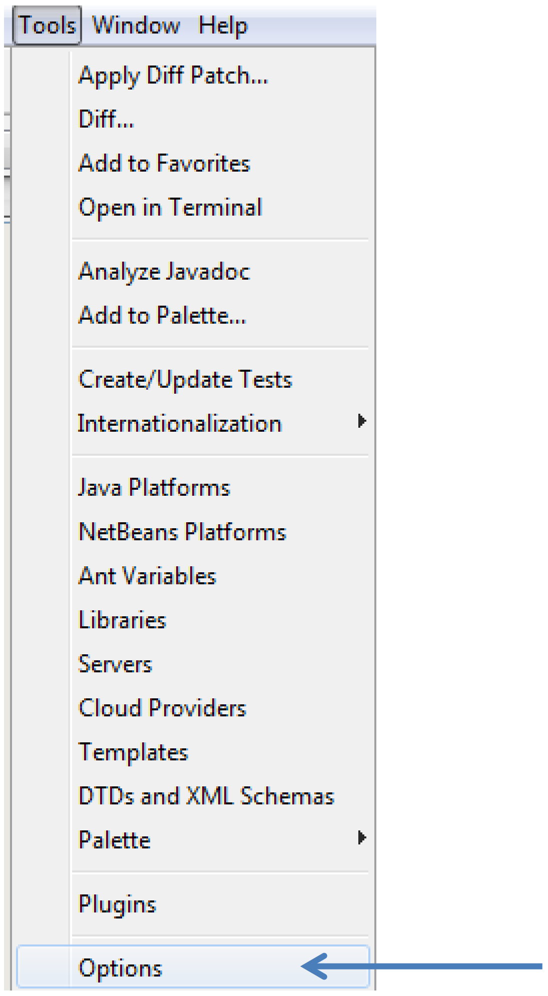
For example, this is the information was generated from the Javadoc comments. The Javadoc comments became HTML code viewable in a web browser. The Javadoc associated files are located within the **dist\Javadoc** directory of your project. You might be encouraged to design a Java library to for use by other developers around the world. The Javadoc comments you add will be invaluable to others. The more descriptive you are, the easier others (and yourself) will be able to read your code.

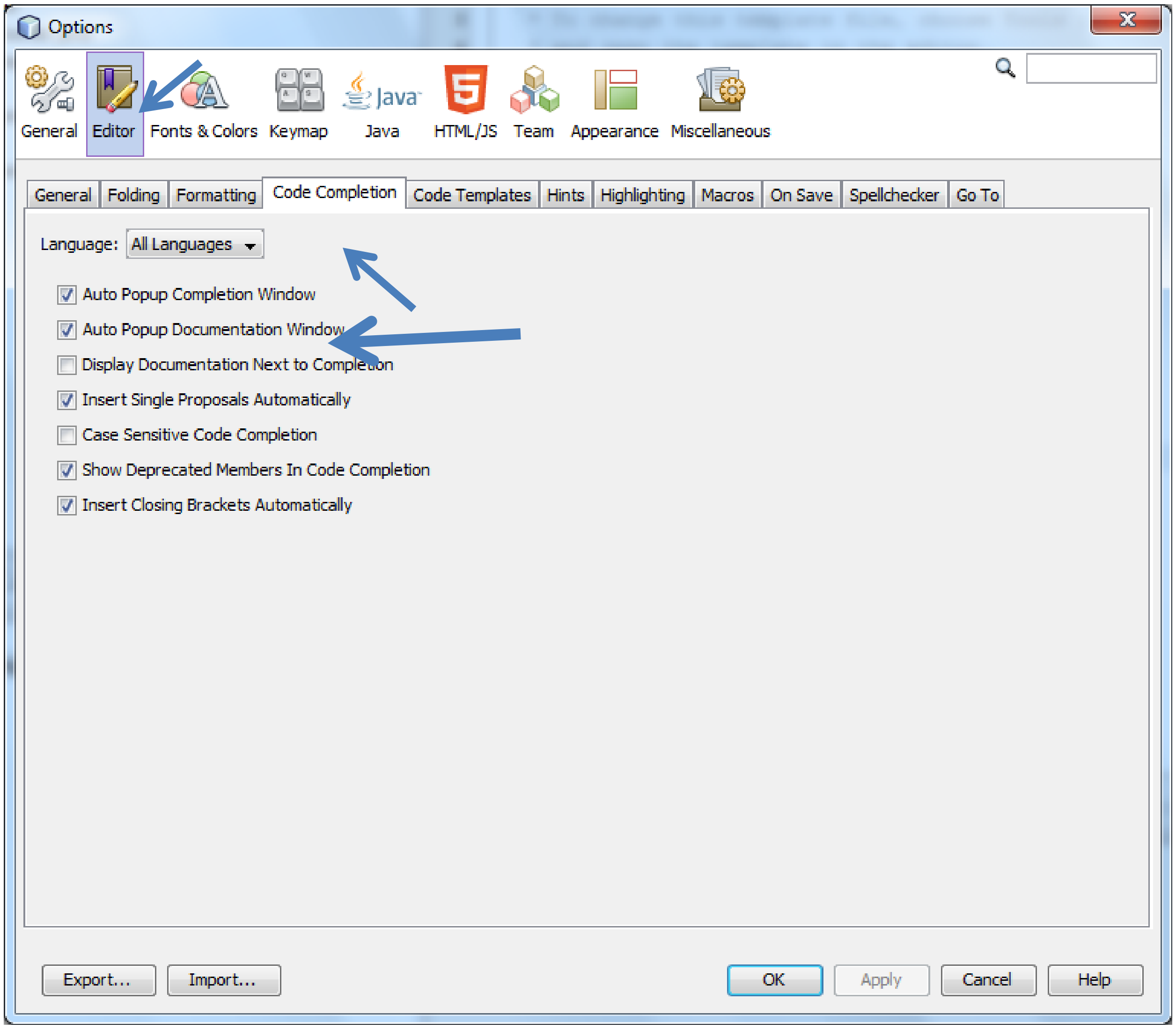
Code Completion

Code Assistance in the NetBeans IDE Java Editor: A Reference Guide
https://netbeans.org/kb/73/java/editor-codereference.html

Code Completion will help assist you when writing your programs. It will offer the various methods and objects associated with the code that you are writing and complete the command you are writing.

Make sure that Code Completion is enabled. From the **Tools** menu, select **Options**.



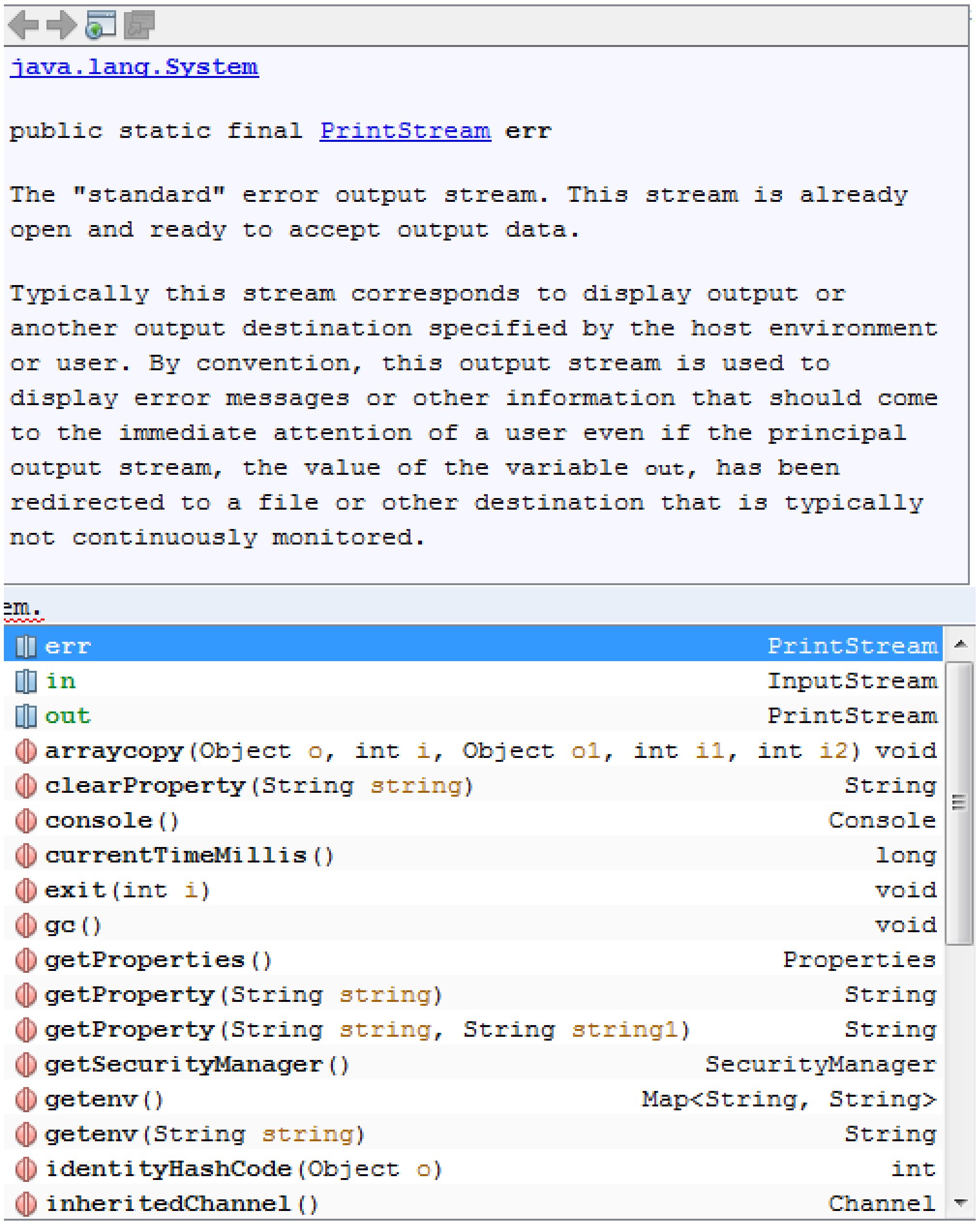


Language: All Languages

- [✓] Auto Popup Completion Window
- [✓] Auto Popup Documentation Window
- [ ] Display Documentation Next to Completion
- [✓] Insert Single Proposals Automatically
- [ ] Case Sensitive Code Completion
- [✓] Show Deprecated Members In Code Completion
- [✓] Insert Closing Brackets Automatically

Select **Editor** options, then select **Code Completion**. You can adjust the **Auto Popup** options as you see fit. But for learning purposes, it's best to keep them set to aid you in coding.

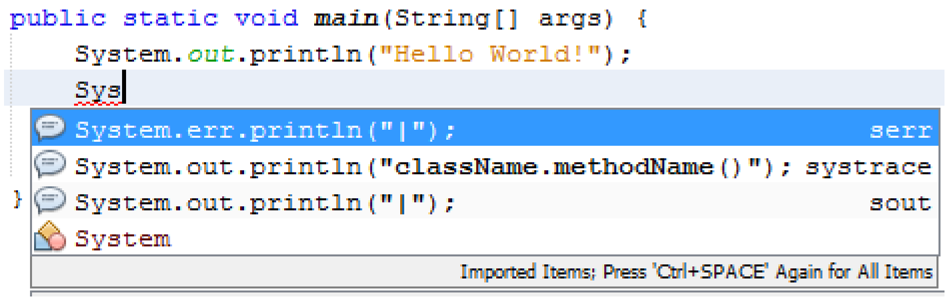
When typing your code and you press a "." a code completion windows popup.



```
java.lang.System

public static final PrintStream err

The "standard" error output stream. This stream is already
open and ready to accept output data.

Typically this stream corresponds to display output or
another output destination specified by the host environment
or user. By convention, this output stream is used to
display error messages or other information that should come
to the immediate attention of a user even if the principal
output stream, the value of the variable out, has been
redirected to a file or other destination that is typically
not continuously monitored.
```

```
em.
err                                                  PrintStream
in                                                   InputStream
out                                                  PrintStream
arraycopy(Object o, int i, Object o1, int i1, int i2) void
clearProperty(String string)                         String
console()                                            Console
currentTimeMillis()                                  long
exit(int i)                                          void
gc()                                                 void
getProperties()                                      Properties
getProperty(String string)                           String
getProperty(String string, String string1)           String
getSecurityManager()                                 SecurityManager
getenv()                                             Map<String, String>
getenv(String string)                                String
identityHashCode(Object o)                           int
inheritedChannel()                                   Channel
```

However, you can press **Ctrl-Space** at any time to pop open the code completion window. Such as if you type *Sys* and then press Ctrl-Space up will come the following.



```
public static void main(String[] args) {
    System.out.println("Hello World!");
    Sys
}
```

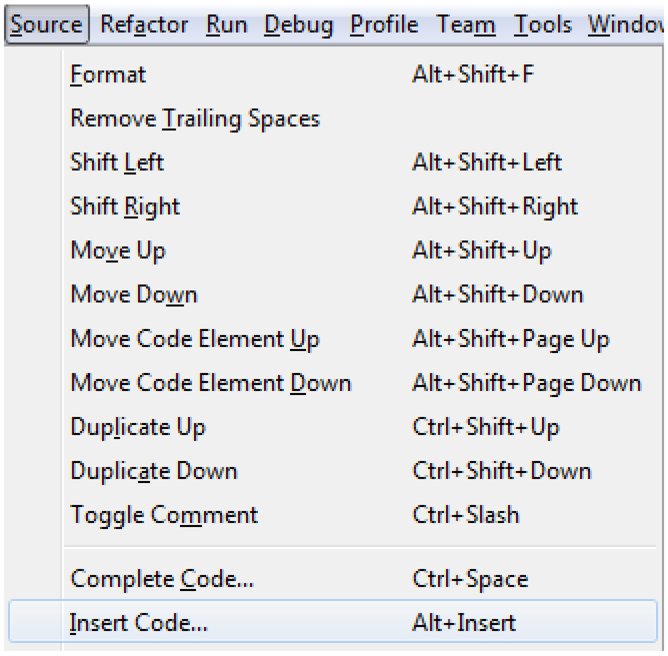
```
System.err.println("|");                            serr
System.out.println("className.methodName()"); systrace
System.out.println("|");                            sout
System
                 Imported Items; Press 'Ctrl+SPACE' Again for All Items
```

You can then select which line to insert into your code by using the arrow keys to select it and pressing enter (or by double clicking on it).

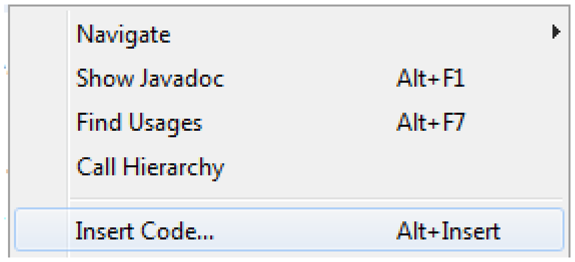
Insert Code

You can also have NetBeans generate `get` and `set` methods for access to the variables in your class.

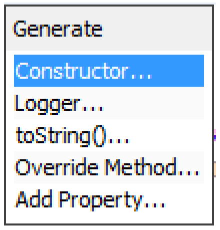
Place the cursor at the location in the file you want to insert the code and then press Alt + Insert (or from the **Source** menu, select **Insert Code**).


Source
Refactor
Run
Debug
Profile
Team
Tools
Windo
Format Alt+Shift+F
Remove Trailing Spaces
Shift Left Alt+Shift+Left
Shift Right Alt+Shift+Right
Move Up Alt+Shift+Up
Move Down Alt+Shift+Down
Move Code Element Up Alt+Shift+Page Up
Move Code Element Down Alt+Shift+Page Down
Duplicate Up Ctrl+Shift+Up
Duplicate Down Ctrl+Shift+Down
Toggle Comment Ctrl+Slash
Complete Code... Ctrl+Space
Insert Code... Alt+Insert

It is also accessible by right clicking at the code location and selecting **Insert Code**.

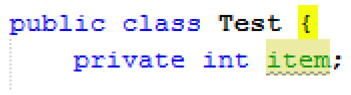


A menu will appear that allows you to generate a piece of code.



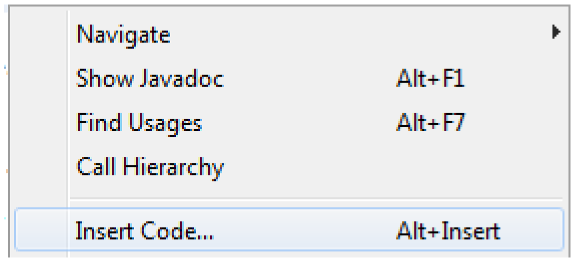
Example – Getter and Setter

Generate get and set methods for the private variable `item`.

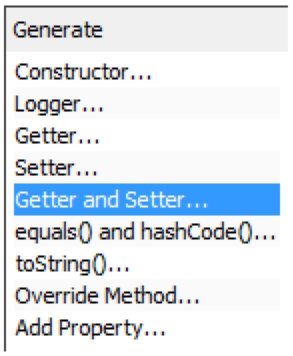


```
public class Test {
    private int item;
```

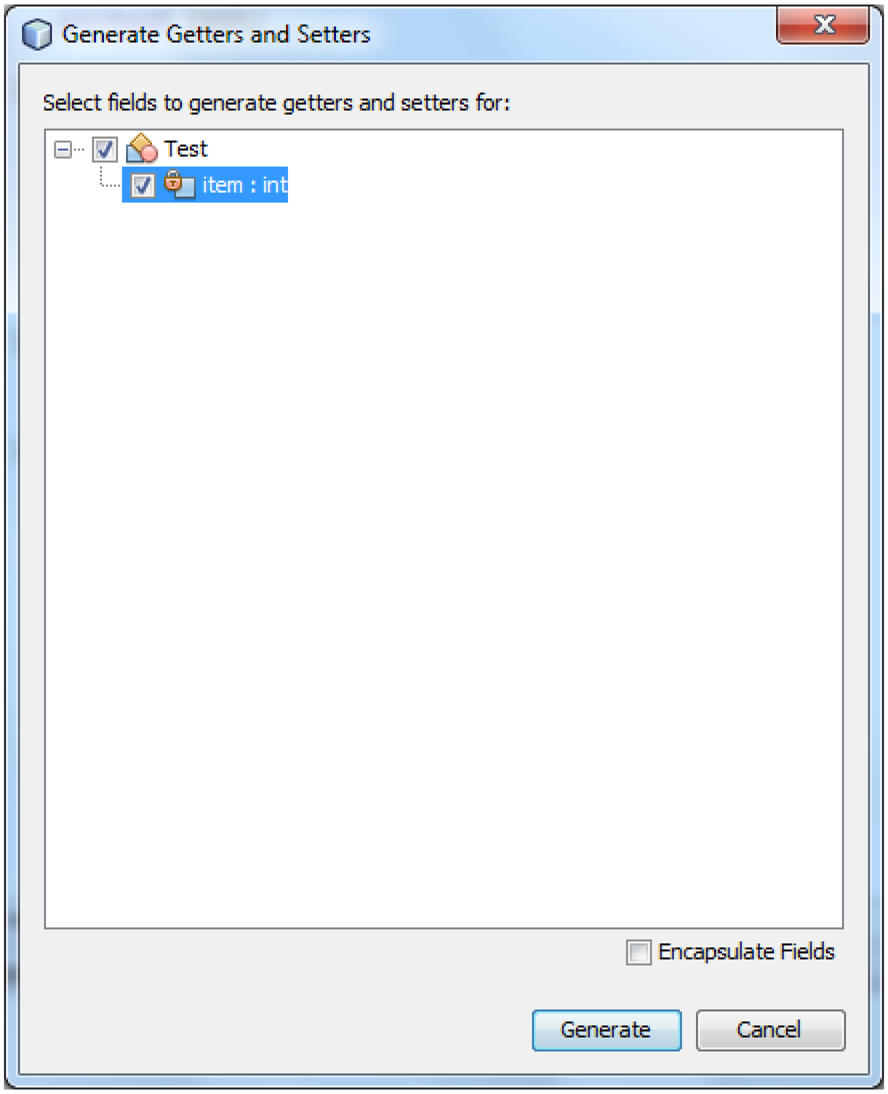
Right click in the class that contains the variable(s) you want to generate code for.



Select **Insert Code**.

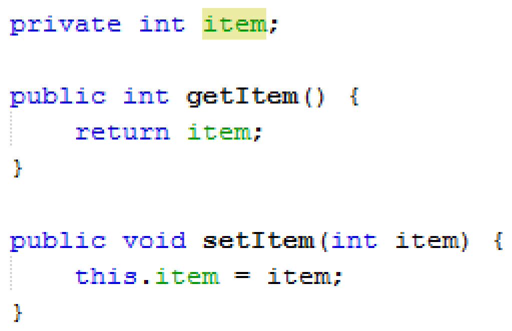


Select **Getter and Setter**.


Generate Getters and Setters
Select fields to generate getters and setters for:
Test
item : int
Encapsulate Fields
Generate
Cancel

Check the variables that you want getters and setters generated. Then press the **Generate** button. If you have multiple variables, you can select them all by clicking the check box next to the class name.

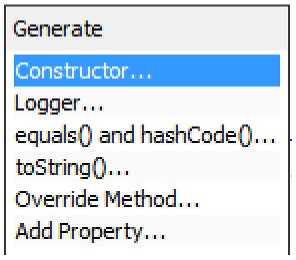
For the variable `item`, NetBeans generated the functions `getItem` and `setItem` for your program to gain access to the variable.

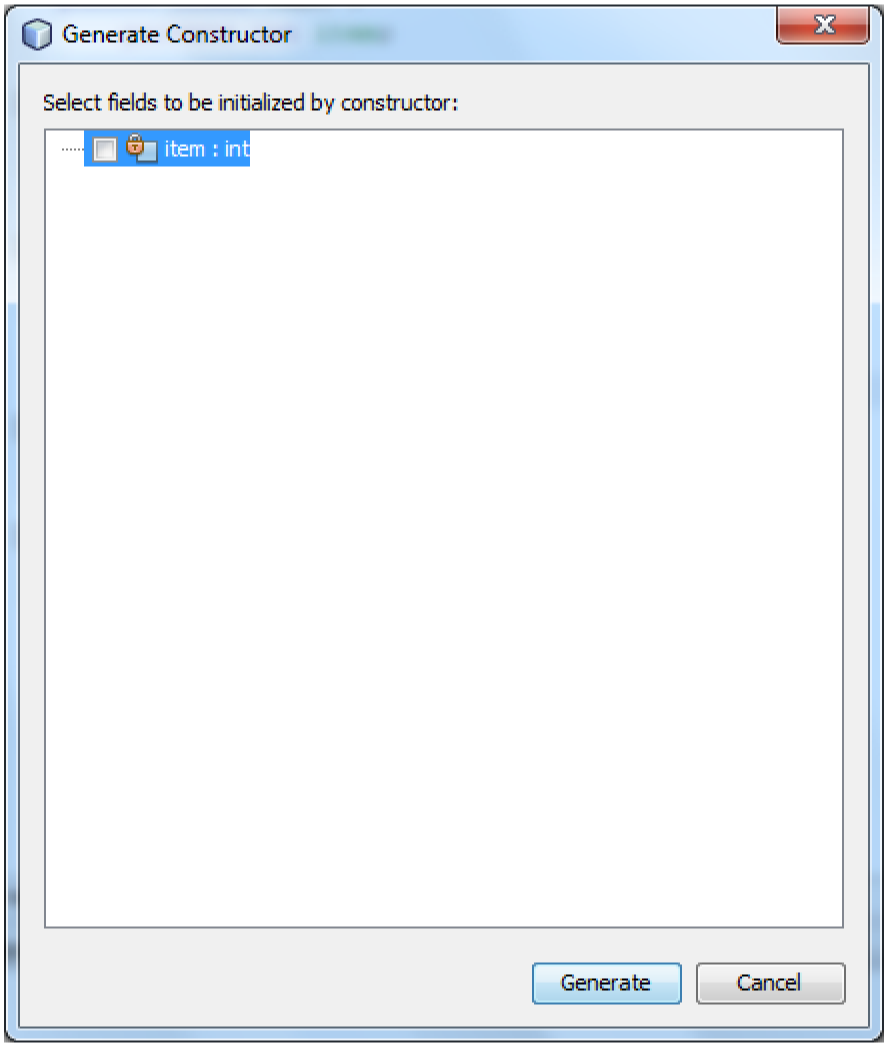


```
private int item;

public int getItem() {
    return item;
}

public void setItem(int item) {
    this.item = item;
}
```

Example - Constructor

Another useful code to insert is a Constructor.





Select the fields you want to be initialized by the constructor and then press **Generate**.

```
public Test(int item) {
    this.item = item;
}
```

A Tutorial is available here:

https://platform.netbeans.org/tutorials/nbm-code-generator.html

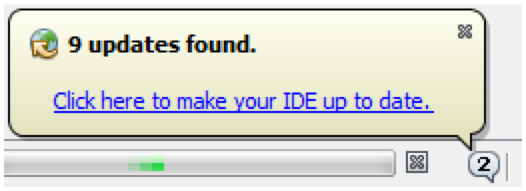
Icons in NetBeans

If you are on a computer with a very high-resolution monitor, the icons in NetBean may appear small. To fix this, you can change the configuration settings by doing the following:

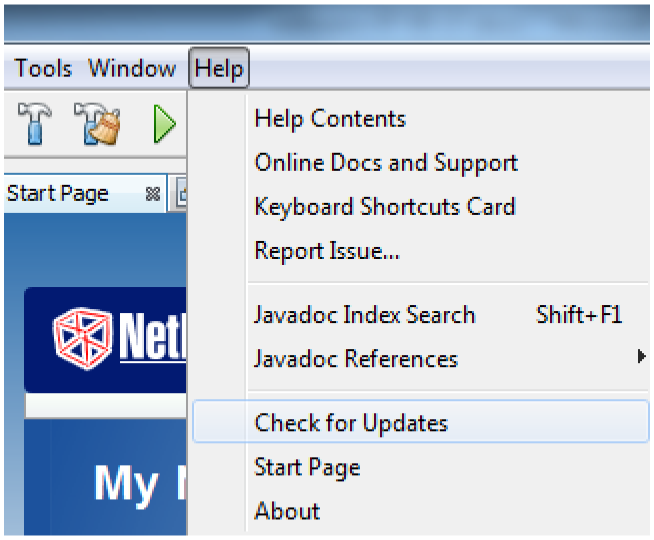
- Make sure NetBeans is not running.
- Open WordPad as **administrator** by right clicking on the WordPad icon and select **Run As Administrator**.
- Open the configuration file:
  - **C:\Program Files\NetBeans 8.2\etc\netbeans.conf**
- Change:
  - `-J-Dsun.java2d.dpiaware=true`

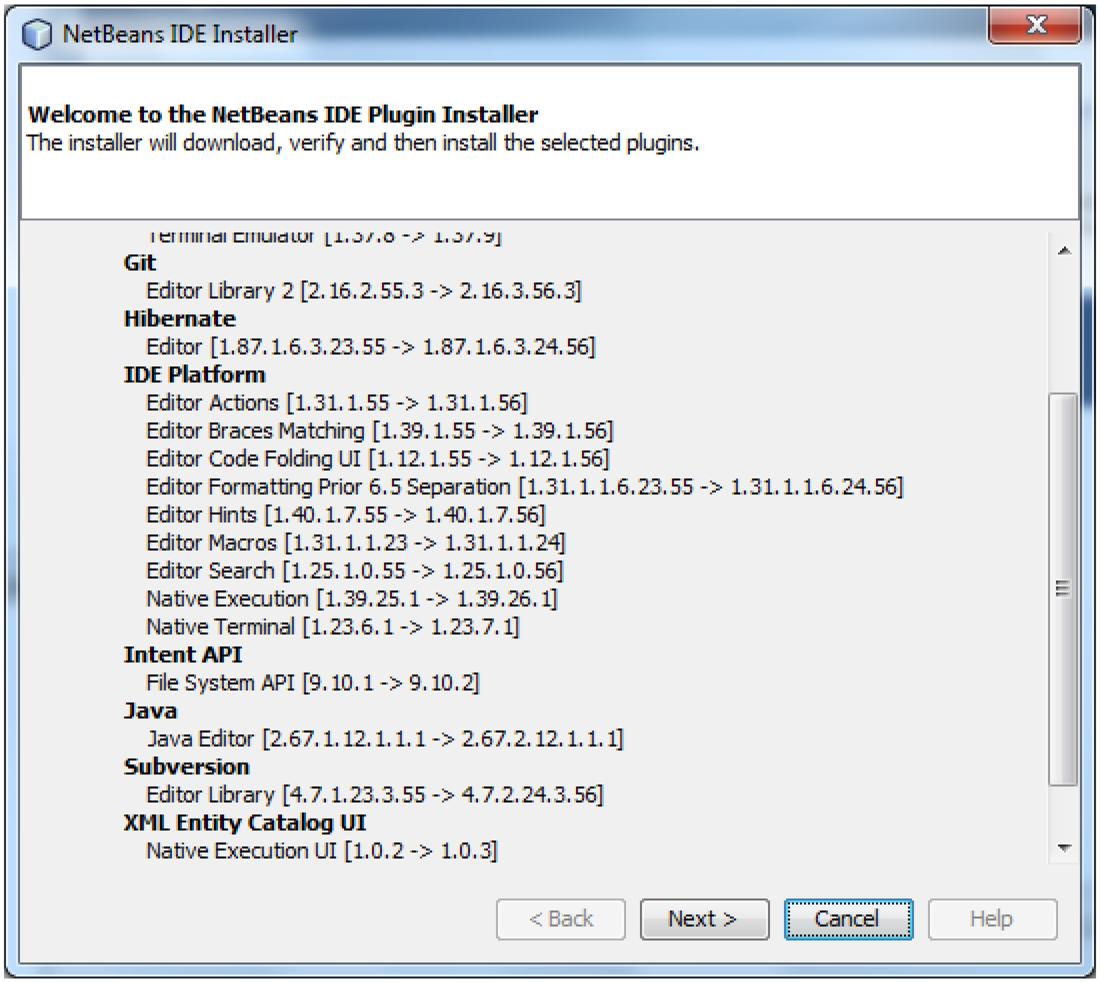
    to
  - `-J-Dsun.java2d.dpiaware=false`
- Save the file, close WordPad and then restart NetBeans.

Updates



If you see a popup bubble that indicating “updates found”, you should update. If this bubble disappears, or you want to check for updates go to the **Help** menu and select **Check for Updates**.

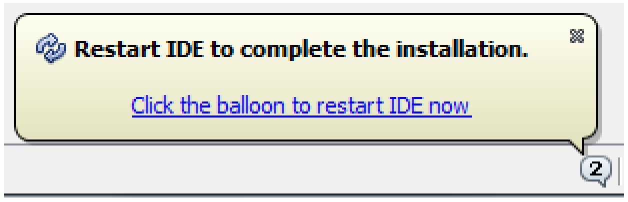



NetBeans IDE Installer

Welcome to the NetBeans IDE Plugin Installer
The installer will download, verify and then install the selected plugins.

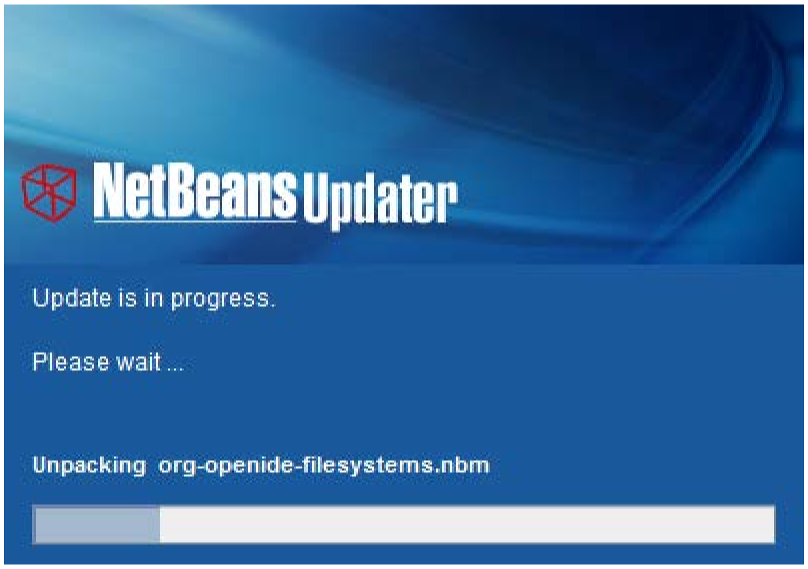
Terminal Emulator [1.37.8 -> 1.37.9]
Git
Editor Library 2 [2.16.2.55.3 -> 2.16.3.56.3]
Hibernate
Editor [1.87.1.6.3.23.55 -> 1.87.1.6.3.24.56]
IDE Platform
Editor Actions [1.31.1.55 -> 1.31.1.56]
Editor Braces Matching [1.39.1.55 -> 1.39.1.56]
Editor Code Folding UI [1.12.1.55 -> 1.12.1.56]
Editor Formatting Prior 6.5 Separation [1.31.1.1.6.23.55 -> 1.31.1.1.6.24.56]
Editor Hints [1.40.1.7.55 -> 1.40.1.7.56]
Editor Macros [1.31.1.1.23 -> 1.31.1.1.24]
Editor Search [1.25.1.0.55 -> 1.25.1.0.56]
Native Execution [1.39.25.1 -> 1.39.26.1]
Native Terminal [1.23.6.1 -> 1.23.7.1]
Intent API
File System API [9.10.1 -> 9.10.2]
Java
Java Editor [2.67.1.12.1.1.1 -> 2.67.2.12.1.1.1]
Subversion
Editor Library [4.7.1.23.3.55 -> 4.7.2.24.3.56]
XML Entity Catalog UI
Native Execution UI [1.0.2 -> 1.0.3]

You will see which updates are available. Select **Next**.


Restart IDE to complete the installation.

Click the balloon to restart IDE now

Restart NetBeans to complete the installation of the updates.



Update is in progress.

Please wait ...

Unpacking org-openide-filesystems.nbm

You will see the NetBeans Updater window while the updates are in progress.

When the updates are completed, NetBeans will open.

Enjoy using NetBeans…