Install Eclipse for Java and C++ Development

Ronald Mak Department of Computer Engineering Department of Computer Science Department of Applied Data Science January 13, 2020 Revised: February 16, 2020

Introduction

Eclipse is a popular integrated development environment (IDE) for developing Java and C++ applications. You can install plugins that provide special features for specialized development frameworks, such as ANTLR 4.

You can download, configure, build, and install Eclipse on Ubuntu, MacOS X, or Windows 10. If you want to run on Ubuntu, first read the following tutorials:

- "Install Ubuntu on Windows 10 and VirtualBox" <u>http://www.cs.sjsu.edu/~mak/tutorials/InstallUbuntu.pdf</u>
- "Configure Ubuntu for Software Development" <u>http://www.cs.sjsu.edu/~mak/tutorials/ConfigureUbuntu.pdf</u>

The process to install Eclipse on any of the three platforms is similar. The Ubuntu and MacOS X platforms are recommended over Windows 10, as the latter may have compatibility problems when you development programs.

Install the Java Development Kit (JDK)

Eclipse requires the Java Development Kit (JDK) which includes the java application for running Java programs and the javac compiler for compiling java programs. Go to <u>https://www.oracle.com/java/technologies/javase-jdk13-downloads.html</u> to download the installer for the JDK for your platform. This will also install java, javac, javap, and jar programs.

For MacOS X, download the latest .dmg image (currently jdk-13.0.2_osx-x64_bin.dmg) and install from that image.

For Ubuntu, download the latest Linux .tar.gz compressed installer file (currently jdk-13.0.2_linux-x64_bin.tar.gz). First unzip the file with the gunzip utility program, and then unarchive with the tar utility. For example:

gunzip jdk-13.0.2_linux-x64_bin.tar.gz
tar -xvf jdk-13.0.2_linux-x64_bin.tar

which will create a jdk-13.0.2 directory containing Java and its utility programs.

Edit the **.bashrc** file in your home directory (it might instead be **.bash_profile** in MacOS X) to set and export **JAVA HOME** to point to the installed **jdk** directory. For example:

export JAVA_HOME=/home/rmak/jdk-13.0.2

Otherwise, you need to type the above as a command each time you open a terminal window.

If you are using the Ubuntu subsystem of Windows 10, use a browser in Windows (such as Microsoft Edge) to download the .tar.gz compressed installer file. Then in the Ubuntu terminal window, use the mv command to move the installer file out of the Windows downloads directory into your Ubuntu home directory. Use /mnt/c or /mnt/d to access directories and files in the Windows file system. For example, from your Ubuntu home directory, enter a command like:

```
mv /mnt/c/Users/rmak/Downloads/jdk-13.0.2_linux-x64_bin.tar.gz .
```

The final dot represents the current directory (your home directory). Then proceed as described above with gunzip and tar to create the subdirectory jdk-13.0.2 in your home directory.

Download and run the Eclipse installer

Download the Eclipse installer from <u>https://www.eclipse.org/downloads/packages/installer</u>. When you run the installer, click on *Eclipse IDE for Java Developers* to start the installation process (Figure 1).

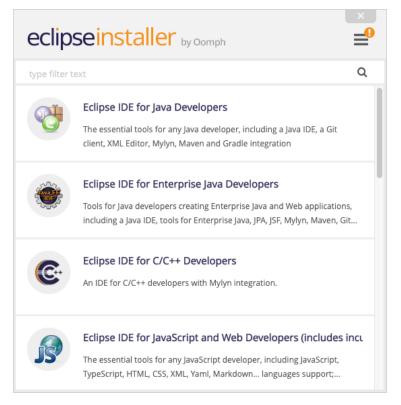


Figure 1. The Eclipse installer.

See the boxed note above about downloading installer files if you're using Ubuntu on Windows 10.

When the installation is complete, launch Eclipse (Figure 2). Eclipse is now ready for you to develop Java projects.

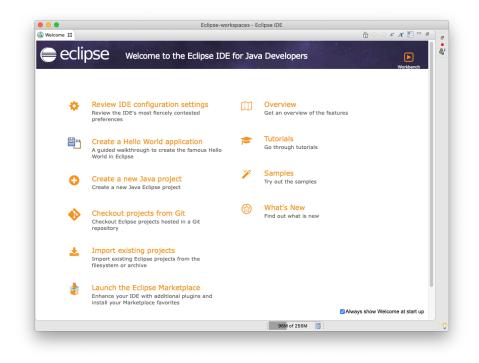


Figure 2. The initial launch of Eclipse.

Create a Java project

If you intend to develop Java programs, you're all set. If it's still open, close the Welcome tab.

Right-click in the **Project Explorer** panel at the left. From the context menu, choose New | Java Project.

Enter a project name, such as "MergeSort" (Figure 3). Click the *Finish* button.

R. Mak, Install Eclipse for Java and C++ Development

) • •]• [] [] []	₩ @• \ \ \ \ - Q • Q • Q • } @ - @ - Ø • @ \ - [] - []	ose-workspaces - Eclipse IDE		Q : 😭 C/C++ i
				• • •
Project Exp			- 8	📴 о 🐹 🗐 т 🔳 🖻 б
				There is no active editor that
There are no To add a proj	projects in your workspace. ject:			provides an outline.
/ Create	a Java project			
	a project			
<u>≧</u> <u>Import</u>	projects			
	🔴 🔿 🌑 New Java Pr	-1		
		oject		
	Create a Java Project Create a Java project in the workspace or in an external location.		5	
	Project name: MergeSort		J	
	✓ Use default location			
	Location: /Users/rmak/Eclipse-workspaces/MergeSort	Browse		
	JRE			
	• Use an execution environment JRE:	JavaSE-13		
	 Use a project specific JRE: 	Java SE 13.0.1 [13.0.1]		
	Use default JRE 'Java SE 13.0.1 [13.0.1]' and workspace of			
				7 🗣 8 🗖
	Project layout		Location	Туре
	O Use project folder as root for sources and class files			
	 Create separate folders for sources and class files 	Configure default		
	Working sets			
	working sets			
items selec [.]	Add project to working sets	New		
	Working sets:	Select		
-	?			
	(?) < Back	Next > Cancel Finish		

Figure 3. Create your Java project.

Right-click the project name in the **Project Explorer** panel and select *New* from the context menu to create a class or a package. You can also add files to the project by dragging the files' names from an external file browser onto *src* under the project name. Select the *Copy files* option and click the *OK* button (Figure 4).

File Operation	8		
Select how files should be imported into the project: Copy files Link to files			
Create link locations relative to:	PROJECT_LOC -		
Configure Drag and Drop Settings			
? c	ancel OK		

Figure 4. Copy files into a project.

Now you're ready to edit, compile, and run your Java program (Figure 5).

R. Mak, Install Eclipse for Java and C++ Development

<pre>** A JRE System Library [JavaSE-11] * MergeSorter java * @ MergeSorter.java * @ MergeSorter.java * @ MergeSorter.java * @ MergeSorter.java * The sort method of this class sorts an array, using the merge sort algorithm. * @ MergeSorter.java * * The sort method of this class sorts an array, using the merge sort algorithm. * @ MergeSorter.java * * The sort method of this class sorts an array, using the merge sort algorithm. * # MergeSorter.java * * The sort method of this class sorts an array, using the merge sort algorithm. * # * The sort method of this class sorts an array, using the merge sort algorithm. * # MergeSorter.java * # Office the sort method of this class sorts an array, using the merge sort algorithm. * # * The sort method of this class sorts an array, using the merge sort algorithm. * # * * * * * * * * * * * * * * * * * *</pre>		eclipse-workspace - MergeSort/src/MergeSorter.java - Eclipse IDE 🛛 🖉
<pre>Quick Access Package Explorer 13 Package Explorer 13 MergeSorter.java 32 MergeSort Am JRE System Library [JavaSE-11] MergeSorter.java Me</pre>	- ile Edit <u>S</u> ource Refac <u>t</u> or Navigate Se <u>a</u> rch	Project Run Window Help
<pre>Quick Access Package Explorer 13 Package Explorer 13 MergeSort A JRE System Library [JavaSE-11] MergeSorter.java MergeSorter.java</pre>		
<pre>Package Explorer 12</pre>		
<pre>>> MergeSort >> MargeSystem Library[JavaSE-11] >>> MargeSorterjava >>> MergeSorterjava >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>		Quick Access 🗄 🖻 🖣
<pre>2</pre>	Package Explorer 🛛 📄 😒 🐨 🔽	🕆 🗖 🖉 MergeSorter.java 🕱 🗖
<pre>> A JRE System Library[JavaSE-11] > default package) > default pa</pre>	⊯ MergeSort	
<pre>* @ src * @ (default package) * d1 MergeSorterjava 4 * The sort method of this class sorts an array, using the merge sort algorithm. 5 */ public class MergeSorter 7 { 80 /** 9 * Sort an array, using merge sort. 10 * @param a the array to sort 11 */ 120 public static void sort(int[] a) 13 { 14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length/2]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[1]; 3 3 4 5 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7</pre>	JRE System Library [JavaSE-11]	
<pre>6 public class MergeSorter 7 { 8</pre>	▼ 🥮 src	4 * The sort method of this class sorts an array, using the merge sort algorithm.
<pre>7 { 80 /** 9 * Sort an array, using merge sort. 10 * @param a the array to sort 11 */ 120 public static void sort(int[] a) 13 { 14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[] = a[1]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[] = a[first.length + i]; 26 } </pre>	🔻 册 (default package)	
<pre>9 * Sort an array, using merge sort. * @param a the array to sort 10 */ 120 public static void sort(int[] a) 13 { 14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 27 </pre>	🕨 🛃 MergeSorter.java	
<pre>10 * @param a the array to sort 11 */ 120 public static void sort(int[] a) 13 { 14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } </pre>		
<pre>120 public static void sort(int[] a) 13 { 14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } </pre>		
<pre>13 { 14</pre>		
<pre>14 if (a.length <= 1) return; 15 16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 27</pre>		
<pre>16 int[] first = new int[a.length/2]; 17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 27</pre>		14 if (a.length <= 1) return;
<pre>17 int[] second = new int[a.length - first.length]; 18 19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } </pre>		
<pre>19 // Copy the first half of a into first 20 // and the second half into second 21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 77</pre>		
<pre>20</pre>		
<pre>21 for (int i = 0; i < first.length; i++) { 22 first[i] = a[i]; 23 } 24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 27 </pre>		
<pre>23</pre>		21 for (int i = 0; i < first.length; i++) {
<pre>24 for (int i = 0; i < second.length; i++) { 25 second[i] = a[first.length + i]; 26 } 27</pre>		
25 second[i] = a[first.length + i]; 26 }		
27		<pre>25 second[i] = a[first.length + i];</pre>

Figure 5. Editing a Java program.

If you are going to develop <u>only</u> Java projects, then you are done installing Eclipse, and you do not need to read the rest of this tutorial.

Install the C++ Development Toolkit (CDT)

Install the CDT plugin if you intend to use Eclipse to develop C++ projects.

From the *Help* dropdown menu at the top of the Eclipse window, choose *Install New Software* ... A dialog box opens for you to choose which plugins to install (Figure 7). Enter the URL http://download.eclipse.org/tools/cdt/releases/9.10 into the *Work with* box and press the return key. Select *CDT Main Features* (Figure 6).

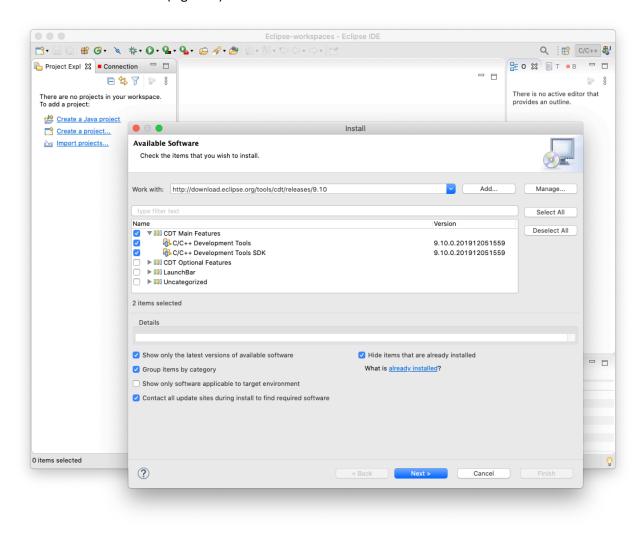


Figure 6. The dialog box for installing plugins.

Click the *Next* button twice, accept the terms of the license, and click the *Finish* button. Wait for the plugin to finish installing. Eclipse will then prompt you to restart it.

Create a C++ project

After Eclipse restarts, close the *Welcome* tab if it's still open.

From the *Window* dropdown menu, choose *Perspective* | *Open Perspective* | *Other* ... | *C/C++* (Figure 7). Click the *Open* button.

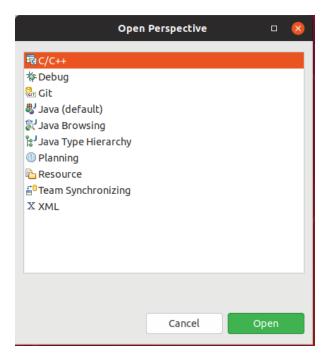


Figure 7. Open the C/C++ perspective.

Right-click in the *Project Explorer* panel at the left. From the context menu, choose *New* | *C/C++ Project*. Choose *C++ Managed Build* and click the *Next* button (Figure 8).

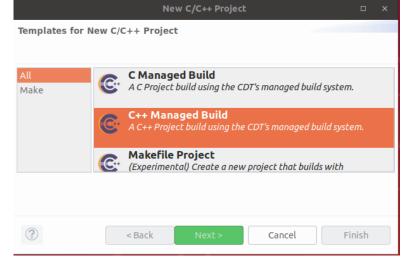


Figure 8. Choose the C++ managed build.

Enter a name for your project. *Select Empty Project* and *Linux GCC*. Do <u>not</u> select a "cross compiler" in the "Toolchains" box. Click the *Finish* button to create your C++ project (Figure 9).

		• 😣		
C++ Project Create C++ project of selected type				
<u>P</u> roject name: BigPi∣ ✔ Use <u>d</u> efault location				
Location: /home/ron/eclipse-workspace/	BigPi	B <u>r</u> owse		
 Project type: Executable Empty Project Hello World C++ Project ⇒ Shared Library Static Library Makefile project 	Toolchains: Linux GCC			
Show project types and toolchains only	if they are supported on th	e platform		
(?) < Back N	ext > Cancel	Finish		

Figure 9. Create your C++ project.

Right-click the project name in the **Project Explorer** panel and select *New* from the context menu to create a class, a header file, or a C++ source file. You can also add files to the project by dragging the files' names from an external file browser onto the project name. Select the *Copy files* option and click the *OK* button (Figure 10).

Figure 10. Copy files into a project.

File Operation 😣		
Select how files should be imported into the project: Copy files Link to files		
Create link locations relative to: PROJECT_LOC -		
Configure Drag and Drop Settings		
Cancel OK		

Now you're ready to edit, compile, and run your C++ program (Figure 11).

R. Mak, Install Eclipse for Java and C++ Development

File Edit <u>S</u> ource Refac <u>t</u> or Navigate Se <u>a</u> rch	eclipse-workspace - BigPi/BigPi.cpp - Eclipse IDE 2
	Quick Access 🖻 🕏 😨
陷 Project Explorer 🛿 📄 🕏 🔍 🗖 🗖	🖻 BigPi.cpp 🛙 🗖
 ✓ SegPi ▶ Binaries ▶ Includes ▶ Debug ▶ BigPi.cpp 	<pre>12 #include <iostream> 13 #include <iostream> 14 #include <iostream> 15 #include <iostream> 16 #include <stdlib.h> 16 #include <strling.h> 17 #include <chrono> 18 19 using namespace std; 20 using namespace std::chrono; 21 22 const int MAX_ITERATIONS = 10; 23 const int PLACES = 1000000; // desired decimal places 24 const int PLACES = 1000000; // the digit 3 before the decimal 25 26 const int BASE = 10; // base 10 numbers 27 const int BIT_COUNT = 8; // bits per machine word 20 20 20 20 20 20 20 20 20 20 20 20 20</chrono></strling.h></stdlib.h></iostream></iostream></iostream></iostream></pre>
	<pre> Problems Tasks Console</pre>
🔓 ВідРі	

Figure 11. Editing and compiling a C++ program.