# ECLIPSE IDE Handbook

# Hot Recipes for the Eclipse IDE

# eclipse

# JAVA CODE GEEKS



Code Geeks

# Eclipse IDE Handbook

# Contents

1	Ecli	pse Tutorial for Beginners	1
	1.1	Introduction	1
	1.2	Requirements	1
	1.3	Download	2
	1.4	Installation	4
	1.5	Tool Overview	5
	1.6	Tool Configuration	13
	1.7	Hello World Example	18
		1.7.1 Writing Your First Program	18
		1.7.2 Executing Your First Program	26
		1.7.3 Debugging Your First Program	28
	1.8	Useful Features	34
		1.8.1 Code Formatting	34
		1.8.2 Refactoring	37
		1.8.3 Call Hierarchy	40
		1.8.4 Local History for Files	41
	1.9	Download the Source Code	44
2	Ecli	pse Environment Variable Setup Example	45
	2.1	Introduction	45
	2.2	Create Simple Project	45
		2.2.1 Create New Package and Class	48
	2.3	Set Environment variable	50
	2.4	Use environment Variable	53
	2.5	Conclusion	54
3	Ecli	pse Web Development Tutorial	55
	3.1	Web resources	55
	3.2	Web page design	55
	3.3	Web projects	55
	3.4	Web archive (WAR)	56

	3.5	Creating a dynamic Web project	56
		3.5.1 Project Facets	64
		3.5.2 Context Root	64
	3.6	Dynamic Web applications	64
		3.6.1 WebContent folder	64
		3.6.1.1 6.1.1. META-INF	64
		3.6.1.2 6.1.2. WEB-INF	64
		3.6.1.3 6.1.3. Classes	65
		3.6.1.4 6.1.4. Lib	65
	3.7	Testing and publishing on your server	65
		3.7.1 Server definitions	65
	3.8	Conclusion	67
			(0)
4	How	7 to update Eclipse	68
	4.1		68
	4.2	Add New Repository	68
	4.3	Check for Updates	70
	4.4	Update Manager	70
	4.5	Conclusion	71
5	How	to install plugin in Eclipse	72
	5.1	Introduction	72
	5.2	Plug it in	72
	5.3	Eclipse Marketplace	72
	5.4	Install New Software	77
	5.5	Installed Plugins	78
	5.6	Update Plugin	81
	5.7	Remove Plugin	82
	5.8	Conclusion	83
6	Feli	nse Window Ruilder Tutorial for CIU Creation	81
U	6 1		04 84
	6.2	Simple Java Window Application	04 94
	0.2	6.2.1 2.1 System requirements	04 04
		6.2.1 2.1 System requirements	04 04
		6.2.1.2 2.1.2 Laure	04 05
	6.2	0.2.1.2 2.1.2 Java	0J 0 <i>5</i>
	0.5	New SWT Application	60 80
	0.4	Components in the editor	69 02
	0.5		93
	6.6	Editor Features	93

	6.7	Layouts in SWT
	6.8	New UI Page         95
	6.9	Source View
	6.10	Button Listener
	6.11	Conclusion
	6.12	Download the Code Project
7	Eclip	se Rich Client Platform (RCP) Tutorial 103
	7.1	Introduction
		7.1.1 What is Rich Client Platform?
		7.1.2 Why Eclipse RCP?
	7.2	Open New Project
	7.3	Eclipse 4 Application Project
		7.3.1 Enter Project Name
	7.4	Project Properties
	7.5	Project Configuration
	7.6	RCP Application
	7.7	Structure of Eclipse 4 RCP application
	7.8	Run RCP Application
	7.9	Stand-alone application
	7.10	Create New Part
	7.11	Add New Part
	7.12	Add Controls on Part
	7.13	Export application
	7.14	Completed Application
	7.15	Conclusion
8	How	to Install and Use the Eclipse Marketplace Plugin 125
	8.1	Introduction
	8.2	Install Eclipse MarketPlace Client
	8.3	Eclipse MarketPlace Wizard
	8.4	Search Eclipse Solutions
	8.5	Search by Market
	8.6	Conclusion

Copyright (c) Exelixis Media P.C., 2017

All rights reserved. Without limiting the rights under copyright reserved above, no part of this publication may be reproduced, stored or introduced into a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), without the prior written permission of the copyright owner.

### Preface

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages through the use of plugins (Source: https://en.wikipedia.org/wiki/Eclipse\_(software)).

Eclipse provides IDEs and platforms for nearly every language and architecture. They are famous for their Java IDE, C/C++, JavaScript and PHP IDEs built on extensible platforms for creating desktop, Web and cloud IDEs. These platforms deliver the most extensive collection of add-on tools available for software developers (Source: https://www.eclipse.org/).

In this ebook, we provide a compilation of Eclipse tutorials that will help you kick-start your own programming projects. We cover a wide range of topics, from setup and configuration, to plugins installation and UI creation. With our straightforward tutorials, you will be able to get your own projects up and running in minimum time.

# About the Author

JCGs (Java Code Geeks) is an independent online community focused on creating the ultimate Java to Java developers resource center; targeted at the technical architect, technical team lead (senior developer), project manager and junior developers alike.

JCGs serve the Java, SOA, Agile and Telecom communities with daily news written by domain experts, articles, tutorials, reviews, announcements, code snippets and open source projects.

You can find them online at https://www.javacodegeeks.com/

### **Chapter 1**

# **Eclipse Tutorial for Beginners**

This is an article about the Eclipse Integrated Development Environment (IDE) for Java Developers, more specifically, the Mars release of the Eclipse IDE for Java Developers.

You will get a brief introduction about how to download, install, and use the software.

The following table shows an overview of the entire article:

#### 1.1 Introduction

Eclipse is a well-known and respected Integrated Development Environment (IDE) developed by the Eclipse Foundation. Eclipse is beneficial to programmers because it aids in the development process by providing the following key features:

- An easy to use graphical user interface that navigates through your code hierarchy.
- Syntax highlighting that displays source code in a color code format to improve readability.
- Code completion that makes recommendations on methods and parameters as you type.
- Recommendations on how to fix errors and automatic error correction.
- A graphical debugger that allows for line-by-line code inspection.
- Single key compilation and execution of a program.
- Automatic code generation for commonly used patterns.
- Integration with source code version control repositories.

There are several benefits to the experienced programmer; however, novice programmers should use IDEs cautiously. Oftentimes, novice programmers become dependent on IDEs without really understanding what is going on behind the scenes, especially as it relates to code generation. Once a programmer understands the basics of writing code from scratch, an IDE is a powerful tool to speed up application development.

#### 1.2 Requirements

In order to use the Mars release of Eclipse IDE for Java Developers, at a minimum, Java Development Kit (JDK) 7 is needed. The JDK includes the Java Runtime Environment (JRE), Java Virtual Machine (JVM), and all other tools needed to write, compile, and execute Java programs. If you already have JDK 7 or higher, you do not need to re-install it. However, if you do not have a JDK or an outdated version, go to Oracle and download the JDK.

#### 1.3 Download

There are several versions of Eclipse that cover several different programming languages. Specifically, for this tutorial, we will cover the Mars release of Eclipse for Java Developers. To download Eclipse, go to the Eclipse IDE for Java Developers website. When you go to the website, you should see a page similar to what's shown below.

	*	eclipse.org Ć	• • • +
	MEMBERS PROJECTS	MORE + E FOR JAVA DEVELOPERS	Corgle" Cutim Sease
RELEASES Mars Packages Neon Packages Luna Packages Juno Packages Indigo Packages Helios Packages Gallico Packages Ganymede Package Europa Packages All Releases	s e E A A A A A A A A A A A A A A A A A A	Eclipse IDE for Java Developers age Description mential tools for any java developer, including a Java IDE, a Git client, XML Editor, Maven integration and WindowBuilder ackage includes: dipse Git Team Provider dipse Java Development Tools daven Integration for Eclipse MyIN Task List toode Recommenders Tools for Java Developers WindowSulider Core dipse XML Editors and Tools alled features list Maintained by: Eclipse Packaging	Download Links Windows 32-bit Windows 64-bit Linux 32-bit Linux 64-bit Downloaded 148,019 Times . Checksums Bugzilla . Open Bugs: 22 Resolved Bugs: 93

Figure 1.1: Eclipse IDE for Java Developers website

Click on "Mars Packages" in the upper right hand corner, which will bring you to a screen similar to what's shown below.



Figure 1.2: Eclipse Mars

As mentioned, there are various versions of Eclipse IDE for several languages. This tutorial covers the Eclipse IDE for Java Developers. The version to download is operating system and JDK version dependent, so make sure you download the version appropriate for your operating system and Java installation. For example, I am using a Mac with a 64-bit installation of Java, so I selected the "64-bit" link next to "Mac Cocoa" from the right-hand side of the screen.



Figure 1.3: Eclipse Download

After clicking the link, you should see a screen that is similar to what's shown below:



Figure 1.4: Mirror Sites

You should scroll down to view the mirror sites that are available in the "Choose a mirror close to you" section. Select the mirror that is closest to you in the list in order to speed up your download of the tool. Columbia University is the mirror closest to my location; therefore, I clicked on the Columbia University link, which displayed the screen below and started the download process.

eclipse	Log in € Log in
GETTING STARTED MEMBERS PROJECTS MORE +	Congle" Switch Search
HOME / DOWNLOADS / THANK YOU FOR DOWNLOADING ECLIPSE	
Thank you for down Participate in our success and together le If the download doesn't start in a few seconds, plea	loading Eclipse et's make Eclipse even better. se dick here to start the download.
Your donation will fund Eci	ipse IDE development.
\$5 Donor	First Name (Optional)
\$10 Supporter	Last Name (Optional)
S35     Friend of Eclipse     Mirror site + Friend of Eclipse logo + O'Relly discount     Best Friend     Indet + Eripse in direct + Eriest hereafte	Email Address (Optional) Comments (Optional) LOG Java Code Geeks
Webmaster Idol	JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

Figure 1.5: Download Success

#### 1.4 Installation

Once your download completes, double click the file you downloaded to extract it to your chosen directory. After the compressed file is extracted, no further work is required to install Eclipse apart from making sure you have installed a JDK, which we covered in a previous step. If you are on a Mac, click on the "eclipse" icon file located in the folder where you extracted the file or if you are on Windows, double-click "eclipse.exe" to launch the application.

You may see startup errors if you have downloaded a version that doesn't coincide with your operating system and/or Java installation version.

The Workspace Launcher dialog displays:

	Workspace Launcher	
Select a wor Eclipse store Choose a wo	rkspace es your projects in a folder called a workspace. orkspace folder to use for this session.	
Workspace:	/Users/username/Documents/eclispeWorkspace	Browse
Use this a	as the default and do not ask again	Java Code Geeks
		Cancel

Figure 1.6: Launcher Dialog

An Eclipse workspace stores your Eclipse configuration and workspace data. Determine a directory where you would like the workspace to reside, enter a name for the workspace, and click "OK". For your information, a "workspace" just represents the physical location on your computer where your files will be stored. After clicking "OK", the "Welcome to the Eclipse IDE for Java Developers" screen displays. In the upper right hand corner, click on "Workbench".

@ Welcome 23	10 · · · · · · · · · · · · · · · · · · ·
eclipse	Workbench
Welcome to the Eclipse IDE for Java Developers	Click on the Workbench loon
Overview         Tutorials           Get an overview of the features         Go through tutorial	IS
Samples Try out the samples What's New Find out what is ne	zw.
	Java Code Geeks

Figure 1.7: Workbench

#### 1.5 Tool Overview

After clicking on Workbench, the following screen displays:

● ● ● □• ₩• □ □ □ □ • • • • • • • • ● ⊙• 25 Ø Ø• >	Java - Eclipse - A	Users/ksmith/Documents/eclispeWo	kspace	Guick Access
🐮 Puckage Explorer 🔉 💽 🔁		В		Teak List 23 Teak
	Problems 18 @ Javadoo 10, Declaration			5 × • E
	Description	Resource Path	Location Type	VA CODE GEEKS

Figure 1.8: Tool Overview

When Eclipse initially launches, it defaults to a view of the "Java Perspective", which appears in the upper right-hand corner.



Figure 1.9: Java Perspective

In Eclipse, a Perspective is a grouping of related windows and features that allows for a developer to perform a specific set of tasks. The Java Perspective offers views and editors for creating and executing Java applications.

By default, the main window of the Eclipse IDE includes the following sections:

- Section A Package Explorer
- Section B Editor
- Section C Task List
- Section D Tabbed Views Pane

The Package Explorer allows you to navigate all of the files associated within a project.



Figure 1.10: Package Explorer

You may open a file by double-clicking on it; the opened file appears in the Editor window.

The Editor window allows you to modify Java source code or text-based files. You may have more than one Editor window opened at once, each displaying a different file. The example below shows one file called, "HelloWorld.java".



Figure 1.11: Editor Window

The Task List links to external bug tracking systems and displays assigned tasks. To learn more about the Task List, read about Eclipse Mylyn.



Figure 1.12: Task List

The Task List view is not to be confused with the "Tasks" view. The "Tasks" view is discussed in section the Tabbed Views Pane below.

The Outline window displays the structure of the file currently selected in the Editor window.



Figure 1.13: Outline View

The Tabbed Views Pane is located at the bottom of the screen and houses various views that can be hidden or shown based on developer preference. The default views that display within the tabbed pane are Problems, JavaDoc, and Declarations.

The Problems view shows any error messages or warnings associated with source code found in your project.

🛐 Problems 🕱 🐵 Javadoc 😥 Declaration						
1 error, 0 warnings, 0 others						
Description	^	Resource	Path	Location	Туре	
V 😣 Errors (1 item)						
O Syntax error, insert ";" to complete BlockStatements		HelloWorld.java	/JavaCodeGeeks/src	line 6	Java Problem	
					5	1
					10	S Lovo Codo Cook
					10	
						(WA 2 LWA DEVELOPERS RESOURCE CENTER



The JavaDoc window shows the documentation for an item selected in the Editor window.



Figure 1.15: JavaDoc View

The Declarations window tells you about the declaration of the Java object currently selected in the Editor.



Figure 1.16: Declaration View

You may add additional views to the bottom tabbed pane by clicking on "Window $\rightarrow$ Show View" and selecting what you want to see.

Window Help		۵	🚱 🏠 🌻
Minimize Zoom Toggle Full Screen	^ዤF	ipse - /Users/ksmith/Docu	iments/eclispe
New Window Editor Hide Toolbar	+		
Show View	►	条 Ant	
Perspective	•	Console	C#QC
Navigation	•	Declaration	₹₩Q D
Bring All to Front		Error Log	₹₩QL
		@ Javadoc	₹₩QJ
		₽avigator	
		E Outline	C#QO
		Package Explorer	<b>₩</b> QP
		Problems	₹¥Q X
		Progress	
		Comparison Project Explorer	
		🔗 Search	THQ S
		Structure	
		Task List	₹¥Q K
		Tasks	
		E Templates	
		S Type Hierarchy	TQX
		Other ICG Java C	<b>Haggers</b>

Figure 1.17: Show View

At a minimum, add the "Console" and "Task" tabs as they are very useful during development.

The "Console" view displays your program's output or any runtime exceptions produced by your code.



Figure 1.18: Console View

The "Tasks" view displays markers, such as, "//TODO" that you have placed in your source code as a reminder to yourself to do something.

	Problem	ns @ Javadoc 🔯 Declaration 📮 Console 🧔 Task	s 23 s				\$ V B
1 it	ama						
*	A 1	Description	Resource	Path	Location	Type	
		TODO: Don't forget to	HelloWorld.java	/JavaCodeGeeks/src	line 5	Java Task	
							17 - 19
							( N lava 0ada 0aala
							JAVA 2 JAVA DEVELOPTIS RESOURCE CENTER

Figure 1.19: Tasks View

#### 1.6 Tool Configuration

Before starting the first example, ensure that Eclipse is properly configured to your development preferences. From the Eclipse menu bar, select Eclipse $\rightarrow$ Preferences to open the Preferences dialog box.

~	Eclipse	File	Edit	Sc
	About E	Eclipse		
Π	Prefere	nces	Ж,	ŧ
ac	Service	s	)	
1	Hide Ed	lipse	жı	1
1	Hide Of	thers	7.81	+
1	Show A	MI .		
	Quit Ec	lipse	Geæ	<b>4S</b> -

Figure 1.20: Preferences



Figure 1.21: Preferences

In the Preferences dialog, you can setup the configuration information for your workspace and set your own development preferences. Expand the "Java" category and select the "Installed JREs" option.

		Preferences					
type filter text	Installed JREs		\$-, ⇒, €				
<ul> <li>General</li> <li>Ant</li> <li>Code Recommenders</li> <li>Help</li> <li>Install/Update</li> <li>Java <ul> <li>Appearance</li> <li>Build Path</li> <li>Code Style</li> <li>Compiler</li> <li>Debug</li> <li>Editor</li> </ul> </li> <li>Installed JRES <ul> <li>Execution Environment</li> <li>JUnit</li> <li>Properties Files Editor</li> </ul> </li> <li>Maven <ul> <li>Mylyn</li> <li>Oomph</li> <li>Run/Debug</li> <li>Team</li> <li>Validation</li> <li>WindowBuilder</li> <li>XML</li> </ul> </li> </ul>	Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.						
	Name	Location /System/Library/Java/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home /Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home	Type MacOS X MacOS X Edit Duplicate Remove Search A Code Geeks				
? .		C	ancel OK				

Figure 1.22: Installed JREs

Eclipse is smart enough to locate the JREs already installed on your computer. If you have more than one, select the appropriate JRE based on your preferences. If your JRE does not appear in the list, you may add it by clicking "Add". Next, under the "Java" category, select the "Compiler" option and set the "Compiler Compliance" Level to the corresponding version of the JDK version you are using.



Figure 1.23: Compiler Options

Next, set your preferences for source code formatting by selecting "Java-Code Style-Formatter".

	Preferences	
type filter text	Formatter	<
▶General ▶Ant ▶Code Recommenders	Active profile:	Configure Project Specific Settings.
<ul> <li>▶ Help</li> <li>▶ Install/Update</li> <li>♥ Java</li> <li>▶ Appearance</li> <li>▶ Build Path</li> <li>♥ Code Style</li> </ul>	Eclipse [built-in] New Import Export All Preview:	C Edit Remove
Clean Up Code Templates Formatter Organize Imports Compiler Debug Editor Installed JREs JUnit Properties Files Editor Maven Mylyn Oomph Run/Debug	<pre>/**  * A sample source file for the code formatter preview  */ package mypackage; import java.util.LinkedList; public class MyIntStack {     private final LinkedList fStack;     public MyIntStack() {         fStack = new LinkedList();     } </pre>	
<ul> <li>▶ Team</li> <li>Validation</li> <li>▶ WindowBuilder</li> <li>▶ XML</li> </ul>	<pre>public int pop() {     return ((Integer) fStack.removeFirst()).intValue(); } public void push(int elem) {     Clark addition(came Tables).</pre>	Restore Defaults Apply
? 🖲		Cancel

Figure 1.24: Code Formatter

The "Formatter" section contains workspace preferences for formatting source code. Under the "Active Profile Option", select the "Java Conventions [built-in]" option for the profile and click "Apply" and then "OK".

000	Preferences	
type filter text	Formatter	
<ul> <li>▶ General</li> <li>▶ Ant</li> <li>▶ Code Recommenders</li> </ul>	Active profile:	Configure Project Specific Settings
<ul> <li>▶ Help</li> <li>▶ Install/Update</li> <li>♥ Java</li> <li>▶ Appearance</li> <li>▶ Build Path</li> <li>♥ Code Style</li> </ul>	Java Conventions [built-in] New Import Export All Preview:	C Edit Remove
Clean Up Code Templates Formatter Organize Imports Debug Editor Installed JREs JUnit Properties Files Editor Maven Mylyn Oomph Run/Debug Team Validation WindowBuilder	<pre>/**  * A sample source file for the code formatter preview  */ package mypackage; import java.util.LinkedList; public class MyIntStack {     private final LinkedList fStack;     public MyIntStack() {         fStack = new LinkedList();     }     public int pop() {         return ((Integer) fStack.removeFirst()).intValue();     } }</pre>	Louis Rada Realiza
PANL	public void push(int elem) {	Restore Defaults Apply
? 🖲		Cancel OK

Figure 1.25: Java Conventions

Now, you are ready to start coding in the Eclipse IDE!

#### 1.7 Hello World Example

#### 1.7.1 Writing Your First Program

To begin developing a Java program using Eclipse, create a new project. A project groups source code, configuration settings, and other files into a deployable unit. From the "File" menu, select "New $\rightarrow$ Java" Project.

e	File Edit Source Refactor	Navigate	Search	Project	Run	Window
	New Open File	₩N Σ		Java Proje Project	ect	va -
olore o <mark>Ge</mark> t	Close Close All	第1 公第1	N N ₿ Ø	Package Class		
fault Helk P H	Save Save As Save All Revert	<del>ዘ</del> ና ሱ ዙና		Interface Enum Annotation Source Fo	n Ider	
/ste burc ur - / b.jar jar -	Move Parame Refresh Convert Line Delimiters To	F	2 <sup>2</sup> 5 <sup>2</sup> 5 <sup>2</sup> 8 <sup>2</sup>	Java Work Folder File Untitled Te	ext File	
ar - /	🕒 Print	жF		Task	Case	
ins.ji cess	Switch Workspace Restart	)		Example		
t.jar aled horr ec.j	<ul><li>➢ Import</li><li>➢ Export</li></ul>			Other	_	ЖN
jce_ pkc s.jai	Properties 1 HelloWorld.java [JavaCode0	第1 Geeks/src]	JGG)	Java C	ode (	leeks
leSo	criptEngine.jar - /System/Library/Java/Ext	tensions		JAVA 2 JAVA DEVEL	OPERS RESOL	RCI CENTER

Figure 1.26: New Java Project

When the "New Java Project" wizard is displayed, enter a name for your new project; accept all defaults when stepping through the rest of the wizard and click 'Finish'.

	New Java Pro	ject				
Create a Java Project Create a Java project in the workspace or in an external location.						
Project name:	HellowWorld					
🗹 Use default	location					
Location: /Us	ers/ksmith/Documents/eclispeWorkspace	/HellowWorld Browse				
JRE						
💽 Use an e	xecution environment JRE:	JavaSE-1.8				
O Use a pro	ject specific JRE:	Java SE 8 [1.8.0_60]				
O Use defa	ult JRE (currently 'Java SE 8 [1.8.0_60]')	Configure JREs				
Project layout						
🔿 Use proje	ect folder as root for sources and class file	s				
<ul> <li>Create se</li> </ul>	parate folders for sources and class files	Configure default				
Working sets						
Add proje	ect to working sets					
Working sets	:	Select				
		Java Code Geeks				
?	< Back N	ext > Cancel Finish				

Figure 1.27: Create A Java Project

An empty project displays, as shown below.

			Java - Eclips	e - Alsers/ksm th/Document	s/eclispeWorks	pace		
11-12-11 · 0-9- 8 0	• 🛎 😅 🌮 🔨 🖓 • 🕅		14 <b>v</b>				0	evel 🖏 😭 🕴 eesotA 20
[1] Package Explorer 전 ♥ @HistanovKhrid (현 rei ▶ ■ JRE System Library (JaveSE-1.8)	86 • • •					~ 0	It Task Ust: 12     □ • • • • • • • • • • • • • • • • • • •	B 🗣 👫 🇐 🔻 🗖 🗖
							Connect Mylyn <u>Connect or your task and ALM tools or greate</u> a local tag	23 5k
							ge Outline is not analysis.	, ·
		Problem	ns 🕫 Javadoc 🕕 Declaration 🥃	Console 🖉 Tasks 🕮				₽ <b>~</b> - <b>-</b>
		×	Description	Навоциси	Puth	Location	Type	
							Java Code	Geeks NOTIRET CENTER

#### Figure 1.28: Empty Project

To add a class to your workspace, right click on your class name and select "New $\rightarrow$ Class". In my example, my project name is "HelloWorld" as shown below:

Package Explorer 🔀	E 😫 👂 🗸 🗆		
HellowWorld	New Go Into	•	/ Java Project
M JRE System Library JavaSE	Open in New Window Open Type Hierarchy Show In て第W	F4	Project  Project  Project  Project  Project  Class  Class  Class
	<ul> <li>Copy</li> <li>Copy Qualified Name</li> <li>Paste</li> <li>Delete</li> </ul>	жс ж∨ ⊗	<ul> <li>Interface</li> <li>Interfac</li></ul>
	<ul> <li>Remove from Context てる</li> <li>Build Path</li> <li>Source て第S</li> <li>Refactor て第T</li> </ul>	↓\$\${ 4 4	Folder File Untitled Text File JUnit Test Case Task
	import ≧ Export		📑 Example
	Refresh Close Project Assign Working Sets	F5	Bright Street S
	Run As Debug As Validate Team Compare With Restore from Local History Configure	• • •	Declaration
	Properties	<b>%</b> 1 //	× .

Figure 1.29: HelloWorld Class

Right click the name of your newly created package in the Package Explorer and click "New $\rightarrow$ Class". The New Java Class dialog displays.

• • •	New Java Class	
Java Class Create a new Java	class.	C
Source folder:	HellowWorld/src	Browse
Package:	(default	t) Browse
Enclosing type:		Browse
Name:	[	]
Modifiers:	public Opackage Oprivate Oprotected     abstract I final I static	1
Superclass:	java.lang.Object	Browse
Interfaces:		Add Remove
Which method stub	s would you like to create?	
	public static void main(String[] args)	
	Inherited abstract methods	
Do you want to add	comments? (Configure templates and default value here)	
	Generate comments	ode Geeks
?	Cancel	Finish

Figure 1.30: Class Dialog

	New Java Class	
Java Class		0
Create a new Java	class.	G
Source folder:	HellowWorld/src	Browse
Package:	com.javacodegeeks.samples	Browse
Enclosing type:		Browse
Name:	HelloWorldClass	
Modifiers:	public Opackage Oprivate Oprotected     abstract Infinal static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stub	s would you like to create?	
	public static void main(String[] args)	
	Constructors from superclass	
-	Inherited abstract methods	
Do you want to add	comments? (Configure templates and default value here)	
	Java Co	<b>de Geeks</b> Re fuscuret center
?	Cancel	Finish



Enter a "Package" and "Name" for your class following the example below.

Under the "Which method stubs would you like to create?", leave the default of "Inherited abstract methods" and also select "public static void main(String[] args)" and click "Finish".



Figure 1.32: Method Stubs

Eclipse generates a class stub that contains several necessary items:

- package line
- class name
- default main method
- default TODO statements

The next step is to add code to your main method.

🚺 HelloWorldClass.java 🔀 1 package com.javacodegeeks.samples; 2 public class HelloWorldClass { 3 4 50 public static void main(String[] args) { // TODO Auto-generated method stub 6 Æ 7 System.out.println("Hello World"); 8 } 9 10 } 11

Figure 1.33: HelloWorldClass.java

```
package com.javacodegeeks.samples;
public class HelloWorldClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World");
    }
}
```

#### 1.7.2 Executing Your First Program

After adding code to the main method, the program can be executed within the Eclipse IDE environment. If you've added the Console view to the bottom tabbed pane of your Eclipse IDE, the execution output of your program will display there. To execute your program, right click on your project name and select "Run As $\rightarrow$ Java" Application.



Figure 1.34: Executing Your Program

The output will display in the Console tab.



Figure 1.35: Console Tab Output

Congratulations! You have written and executed your first Java program in the Eclipse IDE.

#### 1.7.3 Debugging Your First Program

Even before compiling a program in Eclipse, the editor will display problems in your program using Eclipse's auto-correction feature. Notice the red X in the left-most column as it indicates a problem in your code. Upon double-clicking on the red X, a description of the problem and some helpful options to fix the error display.

H H	elloWorldClass.java 🔀							- 0
1	package com.javac	odegeeks.samples;						
3	public class Hell	oWorldClass {						
4 50	public static	void main(String[]	args) { stub					
<b>7</b> 8	Syste.out }	.println("Hello Worl	d");					
10	3							
11	,							
Pr	oblems 🕅 @ Javado	c 🚯 Declaration 🖃 Co	onsole 🧖 Tasks					
1.000	· O warnings O others							
Descr	iption		^	Resource	Path	Location	Type	
V ()	Errors (1 item)			hosourco	- un	Location	iype	
	Syste cannot be res	olved		HelloWorldCI	/HellowWorld/src/c	line 7	Java Problem	
						1.1		
						17-18		
						((JOG)	Java Lode	lieeks
							JAVA 2 JAVA DEVELOPERS I	LISOURCE CENTER

Figure 1.36: Auto-correction


Figure 1.37: Auto-correction

To fix the issue, click "Change to 'System' (java.lang)"; Eclipse automatically corrects the code for you.

This auto-correction feature is useful for correcting compilation errors; however, there are times when errors can only be found during the execution of your program. A debugger is a tool used to capture runtime errors that occur during the execution of your program. Luckily, the Eclipse IDE has a built-in debugger that helps you find the root cause of errors (i.e. bugs) in the code. The Eclipse IDE debugger allows you to examine and step through Java code line by line.

Now, let's look at breakpoint debugging. To fully understand the power of the Eclipse IDE's debugger, add a few more lines of code to your Hello World sample as shown below. The additional lines will be useful to highlight the flow of control during debugging. Ensure that the newly added code builds and executes correctly before continuing on.



Figure 1.38: Additional Lines

```
package com.javacodegeeks.samples;
public class HelloWorldClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        printMessage("Hello World");
    }
    public static void printMessage(String message) {
        System.out.println(message);
    }
}
```

The first step to debugging your program is setting a breakpoint, which is a place in the code at which program execution suspends. If you do not set a breakpoint, your program will run to completion without letting you do any debugging. A breakpoint can be set by double-clicking in the gray margin on the left side of the editor, next to the line where the execution should stop. A blue dot appearing in the margin, indicates an active breakpoint.



Figure 1.39: Breakpoint

After setting the breakpoint, select the menu option  $Run \rightarrow Debug As \rightarrow Java Application to start the debugger. Upon starting the debugger, Eclipse switches the display to the debugger perspective. The flow of execution pauses at the breakpoint that was set. Before switching to the Debug Perspective, the Confirm Perspective Switch dialog may appear.$ 



Figure 1.40: Confirm Perspective Switch

If it does appear, click Yes. Notice that in the Debug Perspective, execution suspends at the breakpoint.

🜻 😑 🔹 Debug - HellowWorld/src/com/javacodegeeks/samples/HelloV	VorldClass.java - Eclipse - /Users/ksm	ith/Documents/eclispeWorkspace	
🖆 • 🖆 • 🗇 • 🗛 • 🎘 😂 🛷 • 🍄 🌌 = 🖉 🖷 🚿 🕨 🛛 🖷 🗠 🎘 • 🖉 👘	¢-¢+++		😰 🖑 Java 💠 Dabug 🛛 Quick Access
Desag 23         Implementation           Implementation         Implementation           Implementating         Imple	0-9-Vanablea 22 0 <sub>0</sub> Breakpoints Name © args	Varke Stering[0] (d=1	() ()
<pre>D HeloWordClass.gue 13 1 puckage con.jsvacodegeeks.samples; 2 public class HelloWorldClass { 4</pre>		C Be Outine 12     C Be Ou	8
Concol 23 STasks HeloWardClass Java Application] Albrary/Java/Java/Irtas/Machmas/gk1.8.0_60 jdk/Contents/Home/bin/java (Mar 12, 2016, 8:94	38 AM		AN DEVELOPERS RESOLUCE CENTR
	Writable	Smart Insert 7:1	

Figure 1.41: Debug View

The perspective includes several new views useful for Debugging. One important view at the top left is the Debug view (not to be confused with the Debug Perspective), which shows the call stack, and the status of all current threads.

Stepping through the code is easy. The debug toolbar allows you to control the flow of the program execution.



Figure 1.42: Debug Toolbar

The most useful menu options are:

- Resume
- Suspend
- Terminate
- Step Into
- Step Over

Resume - Starts debugging after suspending debugger.



Figure 1.43: Resume

Suspend - Pauses the debugger.



Figure 1.44: Suspend

Terminate - Stops the debugger.



Figure 1.45: Terminate

Step Into - This takes you into a method that is called.



Figure 1.46: Step Into

Step Over - This allows you to call a method without stepping into it line by line.



Figure 1.47: Step Over

Now, let's evaluate variables and expressions. Click the "Step-Into" button to bring the flow of control to the "printMessage" method. To see the value of a variable, simply hold the mouse pointer over the variable. The content of the variable is displayed in a small window next to the variable name, as shown below.

0 0 0 (2 Deb	ug - HellowWorld/src/com/javacodegeeks/samples/HelloWo	ridClass.java - Eclipse - /Users/ksmith/Documents/	eclispeWorkspace	
📑• 🖆 · 🗉 / · 🕸 • O • 💁 🧶 • 🖗 🌠 -	- 三百 🔍 🕨 🗆 🖴 3. (5. 12) 号 愛 校 • 🖗 •	⇒ ¢)• → •		😰 🖑 Java 🗱 Debug 🛛 Gulck Access
Pbeoug Si           ▼	host 61847	or⊢ Vanables 52 ®g Breakpoints Name ▶ Ø message	Value "Heto Word" ()d	-:9)
HeterWordClass.jevs [2] public class HeterWordClass {     public class HeterWordClass {     public class HeterWordClass {     public static void nen(String] orgs)     public static void nen(String)     public static void printMessage("Hello Norl");     public static void printMessage(String)     medate(Message)     medate(Messagee)     medate(Messagee)     medate(Messageeee	{ massaga) { els Words" jg=18) 23)			

Figure 1.48: Step Into

The other window views are useful to examine the state of the variables in your program. The Variables view shows the state of all the local variables active in the current method. The data can be examined by expanding the variable. This allows us to view recursively down the tree.

(x)= Variables 🔀	Breakpoints	🏠 🐗 🖻 🔻 🗖
Name		Value
▼ <sup>©</sup> message		"Hello World" (id=18)
hash		0
🕨 🧉 value		(id=23)
		Java Code Geeks

Figure 1.49: Other Window Views

This additional information can prove useful when trying to track down errors in your code.

## **1.8 Useful Features**

### 1.8.1 Code Formatting

Sometimes source code may be hard to read if you've copied the code from another source or opened source code written by someone else. The Eclipse IDE provides an easy way to reformat code with just one-click. For example, if we had code similar to what's shown below and wanted to update the format, we would make use of the source code formatting.



Figure 1.50: Poorly formatted code

```
package com.javacodegeeks.samples;
public class HelloWorldClass {
    public static void main(String[] args) {printText("Hello World");}
    public static void printText(String message) {System.out.println(message);}
}
```

To format code, from the toolbar, select "Source $\rightarrow$ Format":

	Source	Refactor	Navigate	Search	Project
•	Toggle Add Bl Remov Genera	Comment ock Commo ve Block Co ate Element	ent mment Comment		、 、 、 、 第、 「 第、 「 第一
ç ır re	Shift R Shift L Correc Format	ight eft t Indentatio t	'n		/ L 分光F
S	Add Im Organi Sort M Clean	aport ze Imports embers			企業M × 企業O
	Overric Genera Genera Genera Genera Genera	de/Impleme ate Getters ate Delegate ate toString ate hashCoo ate Construe ate Construe	nt Methods. and Setters. e Methods () de() and equ ctor using F ctors from S	 ials() ields Superclass	S
	Surrou	nd With		τ₩Z	
	Externa Find B	alize Strings roken Exter	s	java C	D <b>I CONTRA CONTRA CONTRA DE CONTRA D</b>

Figure 1.51: Format Option

The code is automatically changed to what is shown below based on the preferences for source code formatting that were set earlier in the tutorial.



Figure 1.52: Formatted Code

### 1.8.2 Refactoring

After a program is developed, renaming an object, class, or method requires a considerable amount of work. Typically, every place where that object, class, or method is used would have to be changed. Eclipse offers a way to make changes and ripple those changes across the entire application with just one-click using refactoring.

In this example, refactor the code by changing the name of the "printMessage" method to "printText".

Highlight the name of the method:





Select Refactor from the menu

Eclipse File Edit Source	Refactor Navigate Search Project Run	Window Help 🤷 🙆 🎰 💔
● ● ● ➡ ■ ■ ■ ■ ☆ • ● • <b>₽</b> • ₩ ♂ •	Rename て第R Move て第V	ks/samples/HelloWorldClass.java - Eclipse - /Users/ksmit
Package Explorer X Package E	Change Method Signature て第C Extract Method て第M Extract Local Variable て第L Extract Constant Inline て第I	args) {
<ul> <li>HelloworldClass.java</li> <li>JRE System Library [JavaSE-1.8]</li> </ul>	Convert Local Variable to Field Convert Anonymous Class to Nested Move Type to New File	stub
	Extract Superclass Extract Interface Use Supertype Where Possible Push Down Pull Up	tring message) {
	Extract Class Introduce Parameter Object	
	Introduce Indirection Introduce Factory Introduce Parameter Encapsulate Field	
	Generalize Declared Type Infer Generic Type Arguments	
	Migrate JAR File Create Script Apply Script History	le 23 @ Tasks to Debug y/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/bin/jav

Figure 1.54: Refactor Menu Item

Select Rename, which displays a screen like below:



Figure 1.55: Rename

Enter the new name "printText" and press Enter. All occurrences of the method name are updated.



Figure 1.56: Method Rename

### 1.8.3 Call Hierarchy

When maintaining an application with a lot of classes and dependencies, it is sometimes hard to track other methods that call a particular method. The Call Hierarchy functionality lists all methods that call a given method.

In this example, select the "PrintText" method and right click:

(x)= Variables 🔀 💁 Breakpoints	
Name	Value
▼	"Hello World" (id=18)
hash	0
▼ 🚽 value	(id=23)
▲ [0]	н
▲ [1]	e
▲ [2]	I
▲ [3]	L
▲ [4]	0
▲ [5]	
▲ [6]	W
▲ [7]	0
▲ [8]	r
▲ [9]	I and a
▲ [10]	d David David Ocelia
	(UCG)   JAVA CODE GEEKS
	JAVA 2 JAVA DEVELOPERS RESOURCE CENTER



Select "Open Call Hierarchy", which opens the "Call Hierarchy" view in the bottom tabbed pane.



Figure 1.58: Open Call Hierarchy

💽 Problems 🐵 Javadoc 😥 Declaration 📮 Console 🧔 Tasks 🎄 Debug 🍣 Call Hierarchy 🕱			i i i i i i i i i i i i i i i i i i i
Members calling 'printText(String)' - in workspace			
🔻 🧬 printText(String) : void - com.javacodegeeks.samples.HelloWorldClass	Line	Call	
<b>e<sup>9</sup> main(String()) : void</b> - com.javacodegeeks.samples.HelloWorldClass			
			Java Code Geeks



### 1.8.4 Local History for Files

Eclipse IDE keeps a local history of files, which may prove useful when you need to review a previous version of a file that has not been committed to a version control system yet.

The local history for a file can be selected by right clicking on a file name in the Package Explorer and selecting "Compare With $\rightarrow$ Local History".

🚦 Package Explorer 🔀		HelloWorldClass.java	ES	
<ul> <li>HellowWorld</li> <li>Brc</li> <li>HellowWorld</li> </ul>	B 🐌 ▽	<ul> <li>HellowWorld</li> <li>package com.jav</li> <li>public class He</li> </ul>	src ) acodeg	▶ 🖶 com.javacodegeeks.san geeks.samples; rldClass {
HelloWorldClass.jav	New		•	d main(String[] aras)
▶ 🛃 JRE System Library [Java	Open Open Wi Open Typ Show In	th ce Hierarchy て第W	F3 ► F4	<pre>llo World"); d printText(String mes intln(message);</pre>
	ি Copy ि Copy ि Paste ★ Delete	Qualified Name	жс жv ⊗	
	Remo Build Pat Source Refactor	ve from Context てた th て#S て第T	↓#{4 ● ●	
	🚵 Impor 찶 Expor	t t		
	Referenc Declarati	es ons	•	
	🔗 Refres Assign W	sh /orking Sets	F5	- in workspace
	Run As Debug A Validate Team	s	* * *	.javaoouegeeks.sampies.ne
	Compare Replace Restore f	e With With from Local History	•	Each Other Local History
	Propertie	es	REG	Java Code Geeks

Figure 1.60: Compare With Local History

The "History" view displays in the bottom tabbed pane of Eclipse.

Problems @ Javadoc	🗟 Declaration 🛛 🔲 C	onsole 🧟 Tasks 🕸 D	ebug 🍃 Call Hierarchy	🛃 History 🖾		🔶 🕾 🛃 🦾 🖶 🗁 🚍 🗖 🗖
HelloWorldClass.java						
Revision Time						
3/12/16, 9:24 AM						
II 3/12/16, 9:21 AM						
E 3/12/16, 7:59 AM						
III 3/11/16, 2:14 PM						
E 3/11/16, 2:12 PM						
III 3/11/16, 2:02 PM						
E 3/11/16, 1:59 PM					17	
📰 3/11/16, 1:59 PM					(Inc)	adaga ahaga aya
					(free)	
						JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

Figure 1.61: Local History

A previous version may be selected by double-clicking on it. In this example, I will double-click on "3/12/16, 7:59 AM", which displays the following screen.



Figure 1.62: Previous Version Comparison

The current version is displayed on the left-hand side; the previous version is display on the right-hand side. Eclipse also provides the ability to replace the current version with a previous version by right clicking on a file name in the Package Explorer and selecting "Replace With $\rightarrow$ Local History" or "Restore From Local History".

🚦 Package Explorer 🔀		J HelloWorldClass.	java	Compare HelloWorldClass.java Cur
	🖻 😫 👕 🔻	Java Structure	Compare	6
▼ 2 HellowWorld ▼ 3 src ▼ 4 com.javacode	egeeks.samples	Compilation	Unit dClass tring II)	
HelloWorl	New	- maine	▶ ige(S	String)
▶ <table-of-contents> JRE System Lib</table-of-contents>	Open Open With Open Type Hierard Show In	chy ∵≋W	F3 F3 F4 tre ava	•
	<ul> <li>Copy</li> <li>Copy Qualified</li> <li>Paste</li> <li>Delete</li> </ul>	彩 Name 彩	C Llow V Lc v ctC"	<pre>legeeks.samples; /orldClass { roid main(String[] args) { Hello World");</pre>
	Remove from C Build Path Source Refactor	Context て企業 て第S て第T	↓ but.	<pre>oid printText(String message) println(message);</pre>
	≧ Import ≧ Export			
	References Declarations		• oc	🖲 Declaration 🗉 Console 🖉 Tasks
	🔗 Refresh Assign Working Se	ets	F5	
	Run As Debug As Validate			
	Compare With			
	Replace With			Local History
	Restore from Loca	al History	F	Previous from Local History
	Properties	ж		Java Code Geeks

Figure 1.63: Restore From Local History

## 1.9 Download the Source Code

This was a tutorial on the Eclipse IDE for Java Developers.

Download You can download the full source code of this example here: HelloWorld Project

## **Chapter 2**

# **Eclipse Environment Variable Setup Example**

In this article we will see how to set the environment variables in Eclipse. For this example we will make use of Eclipse Luna 4.4.2.

## 2.1 Introduction

Eclipse is a Java-based open source platform that allows a software developer to create a customized development environment from plug-in components built by Eclipse members. Eclipse is managed and directed by the Eclipse.org Consortium. Eclipse is the most common Integrated Development Environment (IDE) used by Java developers. The best thing that eclipse is cabable of is that it uses plugins for adding more features. So if you don't need a feature you can simple ignore the required plugin.

Many operating systems use environment variables to pass configuration information to applications. Like properties in the Java platform, environment variables are key/value pairs, where both the key and the value are strings. The conventions for setting and using environment variables vary between operating systems, and also between command line interpreters

## 2.2 Create Simple Project

In this section we will see how to create a simple project. Then we will set the environment variable for this newly created project.

Click on File⇒New⇒Java Project.

New	3 <u>-</u> 2		×
Select a wizard		_	$\diamond$
Create a Java project			-
<u>W</u> izards:			
type filter text			
🕼 Enum			^
🐨 Interface			
년 Java Project			
🐙 Java Project from Existing Ant Buildfile			
🐇 Java Working Set			
🖶 Package			
Source Folder			
> 🦢 Java Run/Debug			
> > Junit			
> 🔁 Java Ec			
> Dava Emilier Templates			
/ Strascipe			*
		e Gee	<b>KS</b>
(?) < <u>B</u> ack <u>Next</u> > <u>Finish</u>		Cancel	

Figure 2.1: Java Project

Please note that you might need to click 'Other...' link to go to the Java Project section. In the Project name text box give the name of the project. We will use EnvironmentVariableSetup. Leave the other dafault values as it is. Click the Finish button.

🥘 New Java Project		— <b>—</b> X					
Create a Java Project Create a Java project in the workspace or in an external location.							
Project name: EnvironmentVariableSetup							
Use <u>d</u> efault location							
Location: E:\meraj\study\eclipse-workspace	ce\EnvironmentVariable	Setu B <u>r</u> owse					
JRE							
• Use an execution environment JRE:	JavaSE-1.8	~					
○ U <u>s</u> e a project specific JRE:	jre1.8.0_45	$\sim$					
O Use default JRE (currently 'jre1.8.0_45')		Configure JREs					
Project layout							
○ <u>U</u> se project folder as root for sources ar	nd class files						
Oreate separate folders for sources and	class files	Configure default					
Working sets							
Add project to working sets							
Working sets:		<ul> <li>✓ S<u>e</u>lect</li> </ul>					
	- di l						
		A Code Geeks					
? < <u>B</u> ack <u>N</u>	lext > <u>F</u> inish	Cancel					

Figure 2.2: Project Details

Eclipse will create a default src folder and will include some jar files in the classpath (shown as below). Please note that the list of jar files included can be different depending on the java and Eclipse version.



Figure 2.3: Jar files

### 2.2.1 Create New Package and Class

In this section we will see how to create a new java package in Eclipse.

Right click on the src folder and go to New $\Rightarrow$ Package. In the Java Package pop-up give the package name. We will use com.javacodegeeks. Click Finish.



Figure 2.4: New Package

Now to create a new class in this package right click on the package and choose New $\Rightarrow$ Class. Enter the Class name. You can choose some other options as well to configure the newly created class.

New Java Class			×
Java Class		C	
Create a new Java	class.	G	2
Source fol <u>d</u> er:	EnvironmentVariableSetup/src	Br <u>o</u> wse.	
Pac <u>k</u> age:	com.javacodegeeks	Browse.	
Enclosing type:		Bro <u>w</u> se.	
Na <u>m</u> e:	SimpleClass		
Modifiers:	public Opackage Oprivate Oprotected     abstract final static		
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u> .	
Interfaces:		<u>A</u> dd	
		<u>R</u> emov	e
Which method stul	os would you like to create?		
	public static void main(String[] args)		
	Constructors from superclass		
	In <u>h</u> erited abstract methods		
Do you want to add	I comments? (Configure templates and default value <u>here</u> )		
	Generate comments	<b>IE GE</b> Is resource o	<b>eks</b> Jenter
?	<u> </u>	Cancel	



## 2.3 Set Environment variable

In this section we will see how to set an environment variable in eclipse. Right click on the class (SimpleClass), go to Run  $As \Rightarrow Run Configurations...$  Click on the Environment Tab.

Run Configurations		×
Create, manage, and run config Run a Java application	gurations	
Image: Second system         Image: Second system         Jroid Application         Jroid JUnit Test         ache Tomcat         Apache Tomcat v7.0 (2) at localh         Image: Application         pse Application         pse Data Tools         neric Server         neric Server         neric Server(External Launch)         ogle App Engine         (T JUnit Test         TP Preview         E Preview         a Applet         a Applet         a Applet         a Appleclass         nit         iit Plug-in Test         nch Group         ven Build         Gi Framework         k Context Test         b Application	Name:       SimpleClass         Image: Arguments Image: JRE image: Source Im	nmon Browse Search
K > Filter matched 24 of 110 items	Apply	Re <u>v</u> ert
🤊 😡 Java Cor	<u>e Geeks</u>	Close

Figure 2.6: Run Configurations

On the Environment Tab click on the New... button. In the Name textbox enter the name of the environment variable - *TEST\_ME*. In the Value textbox enter *Tested*. Click OK.

<ul> <li>Run Configurations</li> <li>Create, manage, and run config Run a Java application</li> </ul>	gurations			×		
	Name: SimpleClass					
type filter text	🕝 Main 🕬= Arguments 🛋 JRE 🚸 Classpath 🧤 Source 📧 Environment 🛛 Common					
droid Application	Environment variables to set					
droid JUnit Test	Variable	Value		N <u>e</u> w		
Apache Tomcat v7.0 (2) at localh	TEST_ME	Tested		Select		
pse Application				E <u>d</u> it		
pse Data Tools neric Server neric Server(External Launch) ogle App Engine (T JUnit Test TP Preview E Preview a Applet a Application SimpleClass iit				Rem <u>o</u> ve		
int Plug-in Test inch Group ven Build Gi Framework k Context Test b Application	<ul> <li>Append environment to</li> <li>Replace native environment</li> </ul>	native environment ent with specified environment				
Filter matched 24 of 110 items			Apply	Re <u>v</u> ert		
Iava Co	de Geeks		<u>R</u> un	Close		

Figure 2.7: Environment Setup

🔘 Nev	v Environme	nt Variable		×
<u>N</u> ame:	TEST_ME			
<u>V</u> alue:	Tested			Varia <u>b</u> les
			ОК	Cancel
		G	Java C Java 2 Java Deven	ODE GEEKS

Figure 2.8: New Environment Variable

In the window above you can see all the environment variables that have been defined. You can also Edit or Remove these variable from this window.

## 2.4 Use environment Variable

In this section we will see how we can use the newly defined environment variable.

On the Java platform, an application uses System.getenv to retrieve environment variable values. Without an argument, getenv returns a read-only instance of java.util.Map, where the map keys are the environment variable names, and the map values are the environment variable values. With a String argument, getenv returns the value of the specified variable. If the variable is not defined, getenv returns null

In the class previously defined create a main method. In this main method we will use the java.lang.System class to get the value of environment variable.

String envValue = System.getenv("TEST\_ME");

Now we will print the value returned. Below is the full code for this class

#### SimpleClass.java

```
package com.javacodegeeks;
public class SimpleClass {
    public static void main(String... args) {
        String envValue = System.getenv("TEST_ME");
        System.out.print(envValue);
    }
}
```

In the console you will see that the value of the variable will get printed (Tested).

## 2.5 Conclusion

In this article we saw how to setup a simple java project and how to create a new package and class. We then saw how we can setup environment variable using the Run Configuration setting of Eclipse. We have also examined how to use this newly created environment variable.

## **Chapter 3**

# **Eclipse Web Development Tutorial**

The Web development environment provides the tools we need to develop Web applications as defined in the Sun Microsystems Java Servlet 2.3 Specification and the Sun Microsystems JSP 1.2 Specification. Web applications can be simple (consisting of only static Web pages) or they can be more advanced and include JavaServer Pages (JSP) files and Java servlets.

These resources, along with an XML deployment descriptor (and other Web resources, are contained within a Web project during development. We deploy the Web project to the server in the form of a Web archive (WAR) file once it's ready. The end user can then view the Web application as a Web site from a URL.

The integrated Web development environment makes it easy to cooperatively create, assemble, publish, deploy and maintain dynamic, interactive Web applications.

In this tutorial we will see how we can develop a web application using eclipse.

## 3.1 Web resources

In most cases, all of the resources that we need to create for our Web application are developed during Web site or Web page design; However, there are additional resources that we may need to include in our Web project if we are using more advanced Web technologies in your application. These Web resources are not typical Web page files, and are often not the resources that we consider part of the final Web site. For example, tag libraries and Java resources, such as JAR files, are resources we might need to include in our Web project.

In fact, even the WAR file itself could be considered a Web resource, if we consider importing or exporting the resource.

## 3.2 Web page design

Web pages are an integral part of every Web application. Each Web page should serve to help achieve the overall goal of the entire Web site. There are many types of Web pages, ranging from simple HTML pages that contain no dynamic elements, to advanced Java-based pages that make use of servlets, scripts, forms, or data access components. A few of the many items you should consider when designing your pages are markup language, links, images, and style sheets.

## 3.3 Web projects

Web projects hold all of the Web resources that are created and used when developing your Web application. The first step to creating or importing a Web application is to create either a static or a dynamic Web project. **Static Web projects** are meant to contain only simple Web site resources, such as HTML files. **Dynamic Web projects** are used to structure Web applications that will use more complicated, dynamic Web technologies, such as JavaServer Pages files, and possibly data access resources.

Though the Web project is structured on your file system in compliance with the Java EE Web application standard for deployment purposes, the Project Explorer view is designed to show the most convenient display of project resources for use, while you are actually developing the Web application. When you are finished developing your Web application, you use the Web project to deploy the correct resources to the server. These resources will be packaged in a file called a Web archive, or WAR file.

## 3.4 Web archive (WAR)

A Web application is a group of HTML pages, JSP pages, servlets, resources and source file, which can be managed as a single unit. A Web archive (WAR) file is a packaged Web application. WAR files can be used to import a Web application into a Web server. In addition to project resources, the WAR file includes a Web deployment descriptor file. The Web deployment descriptor is an XML file that contains deployment information, MIME types, session configuration details, and other settings for a Web application. The Web deployment descriptor file (web.xml) provides information about the WAR file and is shared with the developers, assemblers, and deployers in a Java EE environment.

## 3.5 Creating a dynamic Web project

You create and maintain the resources for your Web applications in Web projects. Unlike with static Web projects, dynamic Web projects enable you to create resources such as JavaServer Pages and servlets. To create a new dynamic Web project, complete the following steps:

- 1. Open the Java EE perspective
- 2. Go to File  $\Rightarrow$  New  $\Rightarrow$  Dynamic Web Project

	New Alt+Shift+N >	F.	JPA Project		
	Open File	R	Enterprise Application Project		
	Close Ctrl+W	5	Dynamic Web Project		
	Close All Ctrl+Shift+W	59	EJB Project		
	Save Ctrl+S		Connector Project		
1	Save As		Application Client Project		
	Save All Ctrl+Shift+S	M0	Mayon Project		
	Revert		Project		
	Move	6	Servlet		
1	Rename F2	F	Session Bean (EJB 3.x)		
1	Refresh F5		Message-Driven Bean (EJB 3.x)		
	Convert Line Delimiters To		Web Service		
3	Print Ctrl+P	Ċ	Folder		
	Switch Workspace >	Ċ	File		
	Restart	Ľ	Example		
-1	Import		Other Ctrl+N		
73	Export	Γ			
	Properties Alt+Enter		-		
	1 MANIFEST.MF [MyWebProject//META-INF]	RM	Aarkers 🔲 Properti 🙌 Servers 🔀 🙌		
	2 web.xml [Servers/]				
	3 server.xml [Servers/]	No s	ervers are available. Click this link to create a		
	4 context.xml [Servers/]	1	Invo Podo Pooko		
	Fxit		Java Luue Geeks		

Figure 3.1: Dynamic Web Project

If you don't see the Dynamic Web Project option choose Other and in the Wizards text box start writing Dynamic Web.



Figure 3.2: Other Dynamic Web Project

1. In the pop-up enter the Project Name. For our example we will choose MyFirstDynamicProject. Choose the project location and the Target runtime. Use this field to define a new installed runtime environment. Runtimes are used at build time to compile projects. For our example we will use Tomcat 7.0. If you haven't downloaded the Apache Tomcat you can do this from Tomcat 7.0. For the Dynamic web module version we will use 2.5. Leave the other fields as it is and click *Finish* 

New Dynamic Web Project	
Dynamic Web Project	2
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Applie	cation.
Project name: MyFirstDynamicProject	
Project location	
Use default location	
Location: E:\meraj\study\eclipse-workspace\MyFirstDynamicProject	Browse
Target r <u>u</u> ntime	
Apache Tomcat v7.0 (2) V	New <u>R</u> untime
Dynamic web module <u>v</u> ersion	
3.0	~
Configuration	
Default Configuration for Apache Tomcat v7.0 (2)	Mod <u>i</u> fy
A good starting point for working with Apache Tomcat v7.0 (2) runtime. Additional fac installed to add new functionality to the project.	ets can later be
EAR membership	
<u>A</u> dd project to an EAR	
EAR project name: EAR	New <u>P</u> roject
Working sets	
Add project to working sets	
Working sets: 🗸 🗸	S <u>e</u> lect
Java Cod	<b>P Geeks</b> Resource Center
A Back Next > Finish	Cancel

Figure 3.3: New Dynamic Web Project

On the next pop-up click *Next*. In the next pop-up (Web Module) you can define the Context root and content directory. For our example we will leave the default values as it is.

🔘 New Dynamic V	Veb Project			5		x נ
Web Module Configure web mo	dule settings.				-	
Context root:	MyFirstDynan	nicProject				
Content directory:	WebContent					
⊡ <u>G</u> enerate web.xn	nl deployment	descriptor				
			()	Java I	Code	Geeks Surce Center
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	С	ancel

Figure 3.4: Web Module

Click Finish. Eclipse will generate some files.



Figure 3.5: Generated Files

Now lets create a very simple servlet for our example. Right click on the *src* folder and choose New $\Rightarrow$ Package. Give the package name (com.javacodegeeks). Click Finish. Now right click on the package and choose New $\Rightarrow$ Servlet. Give the servlet name (MyServlet) and click *Finish*.

Create Serv	let — [	⊐ ×
Create Servle Specify class f	<b>t</b> ile destination.	S
Project:	MyFirstDynamicProject ~	
Source folder:	/MyFirstDynamicProject/src	Browse
Java pac <u>k</u> age:	com.javacodegeeks	Browse
Class name:	MyServlet	
Superclass:	javax.servlet.http.HttpServlet	Brows <u>e</u>
Use an exist	ing Servlet class or JSP	
Class na <u>m</u> e;	MyServlet	Browse
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel
	Java Code	Geeks Ource Center

Figure 3.6: Create Servlet

Eclipse will create a sample MyServlet class with two methods - *doGet()* and *doPost()*. These are the two most important methods of any servlet. You can read more about these methods here: https://docs.oracle.com/javaee/5/api/javax/servlet/http/-HttpServlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse)[doGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse)[loGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse)[loGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse)[loGet()], https://docs.oracle.com/javaee/5/api/javax/servlet.http.HttpServletResponse.javax.servlet.http.HttpServletResponse.javax.servlet.http.HttpServletResponse.getWr iter () method. This returns a PrintWriter commits the response's character encoding has not been specified as described in getCharacterEncoding (i.e., the method just returns the default value ISO-8859-1), getWriter updates it to ISO-8859-1. Calling flush () on the PrintWriter commits the response. Either this method or getO

The doGet () method will look like below:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ↔
   ServletException, IOException {
   PrintWriter pw = response.getWriter();
   pw.write("My Fist Dynamic Web Project");
}
```

Now lets test the application. Fist we need to start the server. To start the Server right click on the server and select Start.



Figure 3.7: Start Server

Once the server is started successfully, go to your preferred browser and type this URL: https://localhost:8080/MyFirstDynamicProject/-MyServlet and press enter. You will see the text you set in the print writer is displayed. Now lets understand what the composition of URL. URL stands for Uniform Resource Locator. It is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

The first part of the URL is the **scheme**. In out case it's *http*. *http* is referred to as Hyper Text Transfer Protocol. *localhost* refer to the machine where our application is deployed. In our case it's the localhost. After the host, we provide the port where the application is listening to. After the post, we provide the context root. For our application it's the same as the project name. You must be wondering what the last part (MyServlet) is ? When we created the default Servlet using Eclipse, Eclipse update the web.xml with the below entries:

```
<servlet>
  <description></description>
  <display-name>MyServlet</display-name>
  <servlet-name>MyServlet</servlet-name>
  <servlet-class>com.javacodegeeks.MyServlet</servlet-class>
</servlet>
  <servlet-mapping>
   <servlet-name>MyServlet</servlet-name>
   <url-pattern>/MyServlet</url-pattern>
</servlet-mapping>
```

### 3.5.1 Project Facets

A facet represents a unit of functionality in a Web project. For example, the Dynamic Web Module facet enables the project to be deployed as a dynamic Web module. A brief description of a project facet appears in the wizard when you select it. Note that in many instances, you can view the constraints for a project facet by right click on the facet and select project constraints from the pop up menu.

### 3.5.2 Context Root

The context root is the Web application root, which is the top-level directory of your application when it is deployed to the Web server. You can change the context root after you create a project using the project Properties dialog, which you access from the project's pop-up menu. The context root can also be used by the links builder to ensure that your links remain ready to publish as you move and rename files inside your project.

## 3.6 Dynamic Web applications

There are two types of Web projects: dynamic and static. Dynamic web projects can contain dynamic Java EE resources such as servlets, JSP files, filters, and associated metadata, in addition to static resources such as images and HTML files. Static web projects only contain static resources. When you create Web projects, you can include cascading style sheets and JSP tag libraries (for dynamic Web projects), so that you can begin development with a richer set of project resources.

Dynamic Web projects are always embedded in Enterprise Application projects. The wizard that you use to create a dynamic Web project will also create an Enterprise Application (EAR) project if it does not already exist. The wizard will also update the application.xml deployment descriptor of the specified Enterprise Application project to define the Web project as a module element. If you are importing a WAR file rather than creating a dynamic Web project new, the WAR Import wizard requires that you specify a Web project, which already requires an EAR project.

### 3.6.1 WebContent folder

The mandatory location of all Web resources, includes HTML, JSP, graphic files, and so on. If the files are not placed in this directory (or in a subdirectory structure under this directory), the files will not be available when the application is executed on a server. The Web content folder represents the contents of the WAR file that will be deployed to the server. Any files not under the Web content folder are considered development-time resources (for example, .java files, .sql files, and .mif files), and are not deployed when the project is unit tested or published.

Though the default name given to the folder is WebContent, you can change the name in the Project Explorer by right-clicking the folder and selecting RefactorRename or from the Web page of the project's Properties dialog. In a dynamic Web project, changing the folder name will update the Java build output directory.

### 3.6.1.1 6.1.1. META-INF

This directory contains the MANIFEST.MF file, which is used to map class paths for dependent JAR files that exist in other projects in the same Enterprise Application project. An entry in this file will update the run-time project class path and Java build settings to include the referenced JAR files.

### 3.6.1.2 6.1.2. WEB-INF

Based on the Sun Microsystems Java Servlet 2.3 Specification, this directory contains the supporting Web resources for a Web application, including the web.xml file and the classes and lib directories.
#### 3.6.1.3 6.1.3. Classes

This directory is for servlets, utility classes, and the Java compiler output directory. The classes in this directory are used by the application class loader to load the classes. Folders in this directory will map package and class names, as in: /WEB-INF/classes/com/mycorp/servlets/MyServlet.class. Do not place any .class files directly into this directory. The .class files are placed in this directory automatically when the Java compiler compiles Java source files that are in the Java Resources directory. Any files placed directly in this directory will be deleted by the Java compiler when it runs.

#### 3.6.1.4 6.1.4. Lib

The supporting JAR files that your Web application references. Any classes in .jar files placed in this directory will be available for your Web application.

## 3.7 Testing and publishing on your server

The testing and publishing tools provides runtime environments where you can test JSP files, servlets, HTML files, Java classes and many more artifacts. You can use the workbench to test and publish resources from many types of projects. Here are some examples:

- Dynamic Web projects, which typically contain JSP files, HTML files, servlets, and JavaBeans
- Static Web projects, which typically contain HTML files and graphic files
- Enterprise Applications projects, which may contain Java Archive (JAR) files or Web Archive (WAR) files or both, and pointers to other Web or EJB projects
- EJB projects, which contain enterprise beans
- Application client projects

After testing your application, you can use the tools to publish the application.

### 3.7.1 Server definitions

The workbench defines servers to test and publish your projects. Servers are definitions that identify where you want to test your projects. You can either have the development environment create the servers automatically for you, or you can create them using the New Server wizard. To open the Server view go to Window $\Rightarrow$ Show View $\Rightarrow$ Servers. If there is no Server defined you will see a link saying *No servers are available. Click this link to create a new server...* Click on this link



Figure 3.8: Servers View

New Server	×
Define a New Server	
Choose the type of server to create	
Download additional server ada	pters
Select the server type:	
type filter text	
Tomcat v5.0 Server	^
Tomcat v5.5 Server	
Tomcat v6.0 Server	
Tomcat v7.0 Server	
Tomcat v8.0 Server	
> 🗁 Basic	
> 🗁 Google	
Dublishes and suppliers ISEE and laws EE Web assists and ensure and found in the standard	
Tomcat server.	
Server's best names	
Server's nost name: localnost	
Server name: Apache Tomcat v7.0 (2) at localhost	
Server <u>r</u> untime environment: Apache Tomcat v7.0 (2)	dd
Configure runtime environme	nts
1-14	
Java Code Ge	eks
(?) < <u>Back N</u> ext > <u>Finish</u> Cancel	

Figure 3.9: New Server

For our example we will choose *Tomcat v7.0 Server*. Leave the rest of the field values as default. Click *Next*. On the next screen select the project and click *Add* then click *Finish* 



Figure 3.10: Add and Remove

You will see the server in the Servers tab and also in the Project Explorer tab. The Servers view (similar to the one shown below) allows you to manage the servers. This view displays a list of all your servers and projects that are associated with that server. A project displays under a server when a project from the workbench is added to the server. You can use this view to start, start in debug mode, restart, or stop the servers. In addition, you can use the Servers view to determine the current status and state of the server; and the projects added to the server from the workbench.

### 3.8 Conclusion

In this tutorial we saw how we can make use of the in-build features of Eclipse to create a web application. This is a simple example of the features which Eclipse provides. There are a lot other features which can be used for building much more complex applications.

# **Chapter 4**

# How to update Eclipse

In this article we will see how we can update Eclipse. Eclipse is the most popular Integrated Development Environment (IDE) used by Java developers. The Eclipse platform itself is structured as subsystems which are implemented in one or more plug-ins. The subsystems are built on top of a small runtime engine.

The term Workbench refers to the desktop development environment. The Workbench aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workspace resources. Each Workbench window contains one or more perspectives. Perspectives contain views and editors and control what appears in certain menus and tool bars. More than one Workbench window can exist on the desktop at any given time.

## 4.1 Introduction

If you are upgrading to a newer release of Eclipse from an older release, there are simple steps to follow to migrate your workspace to the new release. Your workspace is the directory on disk that contains all of your project files, as well as meta-data such as preferences you may have customized. The steps to follow for upgrading depend on whether or not you used the "-data" command line argument when starting Eclipse. The "-data" argument is recommended because it clearly specifies the location of your workspace.

The workspace chooser dialog allows you to choose the location of your workspace. This dialog appears on first startup in the absence of a -data argument. The default location provided by this dialog will be a "workspace" child of your home directory

Unless you have an existing workspace from a previous Eclipse version, you can keep this default or choose some other location. You should not store your workspace inside the Eclipse install directory, because that will make it more difficult to upgrade to a newer version of Eclipse. You should not copy or move the workspace directory, because it may contain metadata with absolute file system paths, which will be invalid if the workspace is copied elsewhere.

# 4.2 Add New Repository

If upgrading the platform itself to the next full release follow the steps below:

• Go to Window  $\Rightarrow$  Preferences  $\Rightarrow$  Install/Update  $\Rightarrow$  Available Software Sites

pe filter text	Available Software Sites		-
General			
Android	type filter text		
Ant			
C/C++	Name	Location	<u>A</u> dd
Checkstyle	🗖 🖏 CDT	http://download.eclipse.org/tools/cdt/releases/8.6	
Data Management	🗹 🔩 Eclipse Checkstyle Plugin	http://eclipse-cs.sourceforge.net/update/	Edit
Google	Google Update Site for Eclipse 4.3	https://dl.google.com/eclipse/plugin/4.3	Pomovo
Help	G 🖓 GWT	https://dl.google.com/eclipse/plugin/4.4	Wennove
Install/Update		http://download.eclipse.org/releases/luna	Reload
Automatic Updates	M M2E	http://download.eclipse.org/technology/m2e/relea	
Available Software Sites	M2Eclines	http://download.ccnpsclorg/cccnnology/ni2c/relet	E <u>n</u> able
Java	Multim for Eclinical una	http://download.oslinso.org/m/h/n/roloasos/luna	
Java EE	The Foliose Desired Underse	http://download.eclipse.org/mylyh/releases/luna	Import
Java Persistence	The Eclipse Project Updates	nttp://download.eclipse.org/eclipse/updates/4.4	
JavaScript	The Eclipse Web Tools Platform (W	http://download.eclipse.org/webtools/repository/l	E <u>x</u> port
Maven		http://download.eclipse.org/mylyn/releases/3.12	
Mylyn		http://download.eclipse.org/webtools/updates/	
Plug-in Development			
Remote Systems			
Run/Debug			
Server			
leam			
Terminal			
Validation			
Web	4		
Web Services			
VMI		(rg) lava P	Ico2 ohn
AIVIL		Java L	uug uggi
		JAVA Z JAVA DEVEL	OPERS RESOURCE CEN

Figure 4.1: Available Software Sites

• Click *Add*. Enter the URL of the new repository (for example, https://download.eclipse.org/releases/mars/ for Mars (4.5)). Click OK

al
ive
el

Figure 4.2: Add Site

# 4.3 Check for Updates

Eclipse provides the facility to check for any updates for the existing features and install those updates. To check for updates go to Help  $\Rightarrow$  Check For Updates.

Available Updates			8 <u>000</u>		×
Available Updates Check the updates that you wish to install.					
Name         Image: Android DDMS         Image: Android Hierarchy Viewer         Image: Android Native Development Tools         Image: Android Native Development Tools         Image: Android Traceview         <	Version 23.0.7.2120684 23.0.7.2120684 23.0.7.2120684 23.0.7.2120684 1.9.34 3.9.4.v20160706-2 3.9.4.v20160706-2 2.7.0 1.7.0.20160603-1933	ld com.android.ide.eclipse.ddms.feature.fe com.android.ide.eclipse.hierarchyviewer com.android.ide.eclipse.ndk.feature.feat com.android.ide.eclipse.traceview.featur com.google.appengine.eclipse.sdkbundl com.google.appengine.eclipse.wtp.mave com.google.gdt.eclipse.mobile.android.f com.google.gwt.eclipse.sdkbundle.featu orq.eclipse.m2e.feature.feature.group			~
Details		Java Java 2 Java	Code	<b>Gee</b>	÷ KS

Figure 4.3: Available Updates

Here you can choose the items which you want to update. If upgrading the platform itself, when you are prompted to restart it is strongly recommended to do so. A restart may not be required when updating other features, but always select to restart if you are unsure. Check your Available Software Sites for release-specific update sites that may need updating as well, e.g. The Eclipse Project Updates URL changes with each release (typically release-specific sites hold the SDK/Source features and off-cycle hot fixes and are not required)

Occasionally you won't be able to upgrade Eclipse or certain features due to incompatible changes in the update technology. In these cases you will need to download a fresh install. Download a new build from the Eclipse download Web site Eclipse Download and unzip it in a new directory. Complete upgrade instructions are always included in the Eclipse readme\_eclipse.html file included with every build in the readme directory.

Upgrades may require administrator privileges to succeed and may fail with error messages claiming "Only one of the following can be installed:" otherwise. Start Eclipse with "Run as administrator...".

# 4.4 Update Manager

The Update Manager allows you to find new plug-ins on your machine, your network, or the Internet, compare new plug-ins to your configuration, and install only those that are compatible with your current configuration. The Update Manager thinks in terms of features, a logical group of related plug-ins, and also provides support for managing configurations to undo a given installation or to automatically update all the features currently installed in your Eclipse configuration. Before plug-ins can be

installed by the Update Manager, they need to be collected into a feature. The feature itself has to be published using an update site.

The Update Manager is invoked by Help  $\Rightarrow$  Software Updates. Remember to have an *open* connection to the internet when updating. Proxy settings can prevent the update mechanism from reaching the locations you want/need. The proxy-settings can be changed using Windows  $\Rightarrow$  Preferences  $\Rightarrow$  General  $\Rightarrow$  Network Connections.

## 4.5 Conclusion

In this article we saw how we can update Eclipse. It is very important feature of any software as it allows new features to be integrated with existing ones without reinstalling the software again. We also saw how we can add new repositories for downloading new plugins.

# **Chapter 5**

# How to install plugin in Eclipse

Eclipse is a platform that has been designed from the ground up for building integrated web and application development tooling.

## 5.1 Introduction

By design, the platform does not provide a great deal of end user functionality by itself. The value of the platform is what it encourages: rapid development of integrated features based on a **plug-in** model. Eclipse provides a common user interface (UI) model for working with tools. It is designed to run on multiple operating systems while providing robust integration with each underlying OS.

Plug-ins can program to the Eclipse portable APIs and run unchanged on any of the supported operating systems. At the core of Eclipse is an architecture for dynamic discovery, loading, and running of plug-ins. The platform handles the logistics of finding and running the right code. The platform UI provides a standard user navigation model. Each plug-in can then focus on doing a small number of tasks well. What kinds of tasks? Defining, testing, animating, publishing, compiling, debugging, diagramming...the only limit is your imagination. There are various ways to install a plugin in Eclipse.

# 5.2 Plug it in

The Eclipse platform is structured as a core runtime engine and a set of additional features that are installed as platform **plugins**. Plug-ins contribute functionality to the platform by contributing to pre-defined **extension points**. The workbench UI is contributed by one such plug-in. When you start up the workbench, you are not starting up a single Java program. You are activating a platform runtime which can dynamically discover registered plug-ins and start them as needed.

When you want to provide code that extends the platform, you do this by defining system extensions in your plug-in. The platform has a well-defined set of extension points - places where you can hook into the platform and contribute system behavior. From the platform's perspective, your plug-in is no different than basic plug-ins like the resource management system or the workbench itself.

### 5.3 Eclipse Marketplace

In this section we will see how to use Eclipse Marketplace to install a new plugin.

Go to Help⇒Eclipse Marketplace... A pop-up will appear as shown below:

Eclipse Mar	ketplace				- 12 - 12		×
Eclipse Marke	tplace					2	3
Select solution Press the infor	s to install. Press mation button to	Finish to provide the see a detail	roceed with i	nstallation and a li	on. nk to more informatio	on. 🕤	3
Search Recen	t Popular <u>I</u> n	talled 💡	September N	lewslett	er (loT)		-
F <u>i</u> nd:	٩	All Market	ts	~	All Categories	~	Go
Featured							^
	XRebel Per	formance	e Tool 3.1				
Rebel	Promoted - ( you understan initial more by ZeroTurnary J2EE eclipse jav developer tool development x web apps in-development x	ives you re d and resolv info ound, Comr a ee tools p s profiling n rebel 10 pro evelopment 4.66K (839	al time insigh ve performan mercial <u>productivity e</u> <u>inja tool perf</u> <u>ofilers xrebel.c</u> <u>profiling jav</u> last month)	it into aj ce issue <u>clipse id</u> <u>ormanc</u> <u>com XRe</u> a web ap	pplication performan s faster and earlier, du <u>le software tools Deve</u> <u>e monitoring profiler</u> ebel for Eclipse eclipse ops	ce to help uring elopment for e plugins	
				7.5			
vaadin }>	Vaadin Plug Promoted - \ rich web user i create more	<b>jin for Ec</b> /aadin Fram nterfaces. U <u>info</u>	clipse 3.0.( ework is an c sing its comp	) pen sou ponent k	irce Java UI library for based API developers	creating can	*
Marketpla	aces						
	<b>A</b>						
				Coc	Java Co	de Ge	eks
?		< <u>B</u> ack	<u>I</u> nstall N	ow >	<u>F</u> inish	Cancel	

Figure 5.1: Eclipse Marketplace

In the Find text box write the name of the plugin you want to install and click the search icon. Let say we want to install the Subclipse plugin. We will type *Subclipse* in the search box and will click the search icon. Eclipse will display the results matching the criteria:

Eclipse Mark	cetplace				×
Eclipse Market	tplace			~	5
Select solution: Press the inform	s to install. Press Finish to proceed wit mation button to see a detailed overvie	h installati ew and a li	on. ink to more informat	ion. 🏹	3
Search Recen	t Popular Installed 🖓 Septembe	er Newslett	ter (loT)		
Find: Subclipse	e 🔍 🖉 All Markets	~	All Categories	~	Go
5	Subclipse 1.10.13 An Eclipse Team Provider plug-in pro Eclipse IDE. Developed and maintaine Subclipse is <u>more info</u> by Subclipse Project, EPL syn subversion team provider mylyn	oviding su ed by Sub alm Subc	pport for Subversion version core commit iose version control	within the ters,	^
* 559	Installs: 1.51M (20,591 last mo	onth)		Install	
*1	Subclipse AN I task This ECLIPSE plug-in adds an additio which is available in ANT builds which the more info by null, BSD subversion Installs: 0 (0 last month)	nal ANT t. h have th	ask <de.unkrig.subcl e option 'Run in the :</de.unkrig.subcl 	ipse.svn> same JRE as Install	- •
Marketpla	aces	(CG)	Java Coc	E Gee	KS
?	< <u>B</u> ack <u>I</u> nstall	Now >	<u>F</u> inish	Cancel	

Figure 5.2: Search Results

Click on the Install button for the plugin you want to install. In the next window you will be required to Confirm the selected feature. Here you can unselect the Option features as well. Eclipse helps you to decide by putting the (Optional) text in front of the features which are not Required.

Eclipse Marketplace		×
<b>Confirm Selected Features</b> Confirm the features to include in this provisioning operation. Or go back to choose	2	2
more solutions to install.		
Subclipse 1.10.13 https://dl.bintray.com/subclipse/archive/release/1.12.x/         CollabNet Merge Client         Subclipse (Required)         Subclipse Integration for Mylyn 3.x (Optional)         Subversion Client Adapter (Required)         Subversion JavaHL Native Library Adapter         Subversion Revision Graph         SVNKit Client Adapter (Not required)         SVNKit Client Adapter (Not required)	B Gee	INTER INTER
Install More         Confirm >         Einish	Cancel	

Figure 5.3: Confirm Selected Features

On the next window we need to Accept the licenses. Click on the *I accept the terms of the license agreements* radio button and then click Finish.

Eclipse Marketplace	— D X
Review Licenses Licenses must be reviewed and accepted before the	software can be installed.
Licenses:	License <u>t</u> ext:
<ul> <li>Eclipse Public License - v 1.0</li> <li>Subclipse Software User Agreement</li> <li>Subclipse Software User Agreement</li> </ul>	Eclipse Public License - v 1.0 THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT. 1. DEFINITIONS "Contribution" means: a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and b) in the case of each subsequent Contributor: i) changes to the Program, and ii) additions to the Program; where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel

Figure 5.4: Review Licenses

Eclipse will start installing the plugin. Eclipse will display a Warning pop-up, click OK. Once the features are installed Eclipse will ask you to restart the Eclipse for the changes to take effect. Click Yes. Once the Eclipse has restarted you can verify whether the plugin was properly installed or not. Go to Help $\Rightarrow$ Eclipse Marketplace... then click on the *Installed* tab. Here you can see the list of plugins that are been installed.

After the Find text box there are two other drop-downs which you can use to filter the search. The first one lets you choose where

you want to search the plugin and the second one lets you choose the category for the plugin.

# 5.4 Install New Software

The other way to install a plugin is you use the Available Software window. Go to Help $\Rightarrow$ Install New Software... Eclipse will displace a pop-up window like below:

Install	– D X
Available Software Select a site or enter the location of a site.	
Work with: type or select a site	Find more software by working with the <u>"Available Software Sites"</u> preferences.
type filter text	
Name	Version
Select All Deselect All	
Details	
	<u>.</u>
Show only the latest versions of available software	✓ Hide items that are already installed
☑ <u>G</u> roup items by category	What is <u>already installed</u> ?
Show only software applicable to target environment	
<u> <u> </u> <u>         C</u>ontact all update sites during install to find required software     </u>	Java Code Geeks
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish <b>Cancel</b>

Figure 5.5: Available Software

In the Work with search text box type the site name. You can also select from the existing sites which can be found by clicking the drop-down button. Let's say we want to install the Subclipse, enter the URL: https://dl.bintray.com/subclipse/releases/subclipse/-4.2.x/ in the search box and click Add. Eclipse will display a pop-up window as below:

Add Re	epository	×
<u>N</u> ame:		L <u>o</u> cal
Location:	1ttps://dl.bintray.com/subclipse/releases/subclipse/4.2.x/	<u>Archive</u>
?	ОК	Cancel
•		
	Java Con	IE GEEKS

Figure 5.6: Add repository

Give the name of the repository. We will use Subclipse. Click OK. Eclipse will display the search result. Choose the features you want to install and click Next. IN the next window Eclipse will ask you you review the items to be installed. Click Next. In the next window accept the license and click Finish. Eclipse will install the selected plugin.

# 5.5 Installed Plugins

In this section we will see how to see the list of installed plugin. Go to Help $\Rightarrow$ Installation Details. Here you will see the list of installed plugins:

	Installation History Features Flug-ins Conlig	uration	
Na	me	Version	Id
	Android DDMS	23.0.6.1720515	com.android.ide.eclipse.dd
	Android Hierarchy Viewer	23.0.6.1720515	com.android.ide.eclipse.hie
>	Android Native Development Tools	23.0.6.1720515	com.android.ide.eclipse.nd
	Android Traceview	23.0.6.1720515	com.android.ide.eclipse.tra
	CollabNet Merge Client	4.1.0	com.collabnet.subversion.r
	Eclipse Checkstyle Plug-in	6.19.1.201607051943	net.sf.eclipsecs.feature.grou
>	P Eclipse IDE for Java EE Developers	4.4.2.20150219-0708	epp.package.jee
	P Google App Engine Java SDK 1.9.19	1.9.19	com.google.appengine.ecli
	Google App Engine Maven Integration	3.8.0.v20141030215	com.google.appengine.ecli
>	🚯 Google App Engine Tools for Android	3.8.0.v20141030215	com.google.gdt.eclipse.mo
>	Google Plugin for Eclipse 4.3	3.8.0.v20141030215	com.google.gdt.eclipse.suit
	🚯 Google Web Toolkit SDK 2.6.0	2.6.0	com.google.gwt.eclipse.sdl
	🚯 GWT Designer Core	3.1.3.r44x201405021	com.google.gdt.eclipse.des
>	🚯 GWT Designer Editor	3.1.3.r44x201405021	com.google.gdt.eclipse.des
>	🚯 GWT Designer GPE	3.1.3.r44x201405021	com.google.gdt.eclipse.des
>	🖚 m2e - Maven Integration for Eclipse (includes Incubating co	1.6.1.20150625-2338	org.eclipse.m2e.feature.feat
	🚯 Subclipse (Required)	1.10.13	org.tigris.subversion.subclij
	Subclipse Integration for Mylyn 3.x (Optional)	3.0.0	org.tigris.subversion.subcli;
	🚯 Subversion Client Adapter (Required)	1.10.3	org.tigris.subversion.clienta
	🚯 Subversion JavaHL Native Library Adapter	1.9.3	org.tigris.subversion.clienta
	🚯 Subversion Revision Graph	1.1.1	org.tigris.subversion.subcli;
	🚯 SVNKit Client Adapter (Not required)	1.8.9	org.tigris.subversion.clienta
	🚯 Tracer for OpenGL ES	23.0.6.1720515	com.android.ide.eclipse.glc
	🖗 WindowBuilder Core	1.7.0.r44x201405021	org.eclipse.wb.core.feature.
<	R WindowRuilder CSS Sunnort	1 7 0 rA/v201/05021	ora eclince wh are festure f.
			VA CODE GEEKS

Figure 5.7: Installation Details

In this window you can see the Name, Version, Id and the Provider of the Plugin. You can search for a particular plugin by typing the name in the search text box on the top of the window. To see the History of the installation click on the *Installation History* tab:

stalled Software Installati	on History F	eatures Plug-ins	Configuration			
revious configurations:						
Date	Tag					
S Current Installation						
10/09/2016 14:56:21 BST	r i					
🛞 10/09/2016 14:05:20 BST						
🛞 10/09/2016 13:54:59 BST	1					
🛞 11/07/2015 08:56:14 BST	r i					
🛞 26/04/2015 19:06:12 BST	r I					
36/04/2015 18:29:07 BST	ſ					
🛞 26/04/2015 18:11:10 BST						
126/04/2015 18:11:08 BST 38:00						
🛞 19/02/2015 08:31:31 GM	IT					
🛞 19/02/2015 08:31:29 GM	IT					
19/02/2015 08:31:19 GM	IT					
on <u>f</u> iguration contents:						
Name			Version			1
Android DDMS			23.0.6.172	20515		
Android Hierarchy	Viewer		23.0.6.172	20515		
> 🖗 Android Native Dev	elopment Too	ols	23.0.6.172	20515		
🚯 Android Traceview			23.0.6.172	20515		
> 🚯 Eclipse IDE for Java	EE Developers		4.4.2.2015	50219-0708		
🖗 Google App Engine	Java SDK 1.9.1	19	1.9.19			
Google App Engine	Maven Integr	ation	3.8.0.v201	1410302155-rel-r44		
> I Google App Engine	Tools for And	lroid	3.8.0.v201	1410302155-rel-r44		
> C Google Plugin for E	clipse 4.3		3.8.0.v201	1410302155-rel-r43		
Google Web Toolkit	SDK 2.0.0		2.0.0	201405021502		
GWT Designer Core			2.1.2.r44x	201405021502		
GWT Designer GPF	,		3 1 3 r44v	201405021554		
ma Mayon Inter	ntion for Ealin	includes Incub	ating co. 1.6.1.2015	50675 7220		~
D		Co <u>m</u> pare	<u>D</u> elete	Revert	2	lose

Figure 5.8: Installation History

In this window you can see the installation history by dates.

# 5.6 Update Plugin

To update a plugin, go to Help $\Rightarrow$ Installation Details. Click on the plugin you want to update and then click on the *Update* button at the bottom of the window. Eclipse will update the plugin accordingly. If there is no updates Eclipse will display a pop-up saying *No updates were found*.



Figure 5.9: No Updates Found

You can also update the plugin using Eclipse Marketplace. Go to Eclipse Marketplace and in the search text box insert the plugin name which you want to update. Eclipse will display the *Update* button which you can use to update the plugin:

Eclipse Mar	ketplace					×
Eclipse Marke Select solution Press the infor	<b>tplace</b> s to install. Press mation button to	Finish to proceed w	ith installati view and a li	on. nk to more informat	tion.	3
Search Recen	nt Popular Ins	talled 💡 Septemb	er Newslett	ter (IoT)		
Find: Maven	٩R	All Markets	~	All Categories	~	Go
Maven Integration for Eclipse (Luna) 1.5.0         m2e provides comprehensive Maven integration for Eclipse. You can use m2e to manage both simple and multi-module Maven projects, execute Maven builds via the more info         by Eclipse.org, EPL maven java build development         15       Installs: 70.5K (1,889 last month)       Update       Uninstall						
★4	Maven Prof This feature has recent version, team <u>more i</u> by Red Hat, EPI <u>maven m2e pro</u> Installs:	Files Manageme s been contributed t the JBoss Tools versi nfo <u>ofiles</u> 21.7K (309 last mon	ent 1.5.5 o m2e 1.5.0 ion should b	. If you're using it or be uninstalled. The Ji	a more Boss Tools Install	
Marketplaces						
?		< Back Insta	all Now >	Finish	Cance	I

Figure 5.10: Update Plugin

# 5.7 Remove Plugin

In this section we will see how we can uninstall a plugin. To Uninstall a plugin go to Help $\Rightarrow$ Installation Details. Now select the plugin which you want to uninstall and click the *Uninstall*... button. Eclipse will ask you to review and confirm that you want to uninstall the particular plugin. Click Finish. The plugin will be removed from the Eclipse. You will be required to restart the

Eclipse for the changes to take effect.

You can also uninstall a plugin using Eclipse Marketplace. Go to Eclipse Marketplace and in the search text box insert the plugin name which you want to uninstall. Eclipse will display the *Uninstall* button which you can use to uninstall the plugin

# 5.8 Conclusion

In this article we saw what is an Eclipse plugin and how useful it can be. We also saw how we can install/update/uninstall a plugin. We also discussed the various ways of using a plugin and how Eclipse Marketplace makes it easy for us to manage the plugins.

# **Chapter 6**

# **Eclipse Window Builder Tutorial for GUI Creation**

# 6.1 Introduction

In this example, we will show you how to develop Java GUI Application using Eclipse WindowBuilder plug-in.

Eclipse WindowBuilder is a powerful and easy to use bi-directional Java GUI designer that makes it very easy to create Java GUI applications without spending a lot of time writing code to display simple forms.

The bi-directional Java GUI designer means the developer can seamlessly move between a Drag n' Drop designer and the generated code.

Using Eclipse WindowBuilder, the developer will enjoy creating Java GUI based applications. One can create complicated windows in minutes using WindowBuilder.

WYSIWYG (What You See Is What You Get) layout tools in WindowBuilder are used to generate back-end java code by dragand-drop of components to the container.

# 6.2 Simple Java Window Application

Now, we will see how fast a simple Java GUI application can be created uisng Eclipse WindowsBuilder.

### 6.2.1 2.1 System requirements

Tools required to run this example are:

#### 6.2.1.1 2.1.1 Eclipse

WindowBuilder is built as a plug-in to Eclipse. *Eclipse for RCP and RAP Developers* is the default IDE bundled with *Windows Builder* plug-in. This IDE has a complete set of tools for developers who want to create Eclipse plug-ins, Rich Client Applications (RCA).

Download Eclipse for RCP and RAP Developers from here. Please refer the picture given below to identify the correct IDE.



Figure 6.1: Eclipse IDE for RCP and RAD

#### 6.2.1.2 2.1.2 Java

• Download Java SE 7 or above from here

# 6.3 Open New Project

Let us create a new 'SWT/JFace Java Project' to see the usage of WindowBuilder for building GUI components. Open 'File - New - Other' and then click *SWT/JFace Project* as depicted below

Plug-in Development - Eclipse

File	Edit Source Refactor Navigate Search	Project Run	Wind	ow Help	
	New Onen File	Alt+Shift+N >	13	Plug-in Project	
	Clara	C+rL+W		Project	
	Close All	Ctrl+Shift+W	4	Task	
	Save	Ctrl+S	R	Component Definition	
	Save As			Product Configuration	
2	Save All	Ctrl+Shift+S		Target Definition	
	Revert		H C	Package	
	Move		Ø	Interface	
1	Rename	F2	€9	Source Folder	
	Kefresh Convert Line Delimiters To	61 (	Ľ	File	
	Drint	Chilly D		Folder	
2		Ctil+P		Example	
	Switch Workspace Restart	>		Other	Ctrl+N
-1	Import			6.	Oode Oceles
5	Export			<b>UCG</b> Java	<b>GOOB GEEKS</b>
	Properties	Alt+Enter		in the part of	

Figure 6.2: Open Project



Figure 6.3: SWT JFace Java Project

New SWT/JFace Java Project		3		×
Create a Java Project			ſ	E.
Create a Java project in the workspace or in a	n external locatio	on.		S
Project name: MyWinApp				
ocation: ustomers\JavaCodeGeeks\exam	ples\eclipsewinb	uilder\app	Browse	
JRE				
Use an execution environment JRE:	JavaSE-1.8			~
O Use a project specific JRE:	jdk1.8.0_40			~
O Use default JRE (currently 'jdk1.8.0_40')		C	Configure JRE	<u>s</u>
Project layout				
Use project folder as root for sources a	nd class files			
○ <u>C</u> reate separate folders for sources and	class files	Con	<mark>figure defaul</mark>	<u>t</u>
Working sets				
Add project to working sets				
W <u>o</u> rking sets:		~	S <u>e</u> lect	
The wizard will automatically configure existing source.	e the JRE and the	project layou	ut based on th	ne
? < <u>B</u> ack [	<u>v</u> ext >	<u>F</u> inish	Cance	:I

Figure 6.4: Project Name

The reason for creating new project as *SWT/JFace Java Project* is to have all the necessary JARs and native libraries included by the IDE itself. Otherwise, you have to add all these dependent JARs and native libraries on your own.

The Standard Widget Toolkit (SWT) is a graphical widget toolkit to be used with the Java platform. It provides a portable graphics API independent of the OS but that relies on the native widgets.

JFace is a UI toolkit with classes for handling many common UI programming tasks. JFace is window-system-independent in both its API and implementation, and is designed to work with SWT without hiding it.

JFace is a higher-level user interface toolkit that uses the raw SWT widgets to provide model-driven widgets, and to some extent some functionality that isn't available in the Swing libraries, such as advanced editors, dialog boxes, and wizards.

# 6.4 New SWT Application

Let us add widget to the project. As a main window, create Application Window as shown below. Right click on the project and select *New - Other - Window Builder - SWT Designer - SWT - Application Window*. And then click *Next* 



Figure 6.5: Application Window

Enter Class Name and click Finish

lew SWT A	pplication		223		×
Create SWT	Application			T	R
Create a simpl	e SWT application with Shell and event	loop.		S	wт
Source fol <u>d</u> er:	MyWinApp/src			Br <u>o</u> wse	
Pac <u>k</u> age:	com.jcg.rca.main			Bro <u>w</u> se	
Na <u>m</u> e:	MainWindow				
Create content	s in: ed createContents() method open() method static main() method	Java Code Geeks			
?		< <u>B</u> ack <u>N</u> ext > <u>Finish</u>		Cance	I

Figure 6.6: Class Name

A basic window application has been created. Window Builder can be used to get your UI up and running quickly. Click *Design* tab as shown below.



Figure 6.7: Basic Window Application

Now, you will see the graphical representation (Design View) of your code.

Package Explorer 23 35 Plug-ins	🚽 MainWindow.java 💱		- 0 0 * - 0
Package Explorer 33 SPug-ins C C       Package Explorer 33 SP Pug-ins       Package Explorer 34       Package Explorer 35       Package Explorer 35	Structure      Components     H     SwT Application*	Palette     Palette     System     Selection     Marquee     Selection     Marquee     Secomposite     Group     ScrolledCo     SashForm     TabFolder     TabFolder     TabItem     CtabFolder     Composite     Composite     TabFolder     TabFolder     TabFolder     TabFolder	Complexity of the second secon
	Properties       *a       3       3*       1*         Variable       shell       ^         Layout       (absolute)       •         Gass       org-eclipse.swt       *         Bindings       []       alpha       255         backgrou       240,240,240       =         enabled       © true       image         modified       [] false       tab order         tab order       *       *	Layouts     Layout     Second Layout     Gorde Layout     Gorde Layout     Gorde Layout     Gorde Layout     Controls     Controls     Controls     Controls     Controls     DateTime     DateTime     DateTime     DateTime     DateTime	(\$
			445x201506110820

Figure 6.8: Design View

This application can be simply executed like any other java program with main method. Right click on the class name and *Run* As - Java Application



Figure 6.9: Run Application

As we have not yet added any other elements, you will see a simple window popping-up as a result of the execution.

😫 Package Explorer 😫 🛸 Plug-ins 😐 🗖	d MainWindow.java 83	- 0 0x », - 0
> S comjegarepest > MyWinApp > MyWinApp > S src - Comjegareamain	Structure     O     File     Components     File     System     Sell- "SWT Application"     Selling Marquee     Schoose co; Tab Order     SWT Application	
<ul> <li>[2] MainWindow.java</li> <li>mi JRE System Library [jdk1.fl.0_40]</li> <li>mi Referenced Libraries</li> </ul>	SWT Application - X SWT Appli	© createC
	Problems Target Platform State Console 13 / Search     MainWindow (Java Application) Cham/Java)dk1.8.0.40(bin)javaw.exe (Dec 19, 2015, 529:57 PM)	
< >		

Figure 6.10: Executed Application

# 6.5 Components in the editor

As shown above, the editor is composed of the following major components:

- Design View the main visual layout area.
- Source View write code and review the generated code
- Structure View composed of the Component Tree and the Property Pane.
  - Component Tree shows the hierarchical relationship between all of the components.
  - Property Pane displays properties and events of the selected components.
- Palette provides quick access to toolkit-specific components.
- Toolbar provides access to commonly used commands.
- Context Menu provides access to commonly used commands.

### 6.6 Editor Features

The editor supports the following major features;

- Bi-directional Code Generation read and write almost any format and reverse-engineer most hand-written code
- Internationalization (i18n) / Localization externalize component strings, create and manage resource bundles.
- Custom Composites & Panels create custom, reusable components.
- Factories create custom factory classes and methods.
- Visual Inheritance create visual component hierarchies.

- Event Handling add event handlers to your components.
- Menu Editing visually create and edit menubars, menu items and popup menus.
- Morphing convert one component type into another.

# 6.7 Layouts in SWT

Layouts are non-visible widgets used to give GUI windows a specific look and it helps to control the position and size of children in a *Composite*.

To make sure the GUI application developed in one environment works perfect in another platform, Java provides a system of portable layout managers. We use these layout managers to specify rules and constraints for the layout of the UI in a way that will be portable.

Layout managers gives you the advantages as given below,

- Correctly positioned components that are independent of fonts, screen resolutions, and platform differences.
- Intelligent component placement for containers that are dynamically resized at runtime.
- Ease of translation. If a string increases in length after translation, the associated components stay properly aligned.

SWT Designer supports the following layout managers.

AbsoluteLayout	AbsoluteLayout or Null Layout helps to specify the exact
	position, the width and the height of components. In a
	generic environment where the size of the screens may
	vary, this layout manager should be avoided.
FillLayout	FillLayout is the simplest layout class. It lays out controls
	in a single row or column, forcing them to be the same size.
RowLayout	Puts the widgets in rows or columns and allows you to
	control the layout with options, e.g., wrap, spacing, fill and
	so on.
GridLayout	Arranges widgets in a grid.
FormLayout	Arranges the widgets with the help of the associated
	attachments.
StackLayout	A StackLayout object is a layout manager for a container.
	It treats each component in the container as a card. Only
	one card is visible at a time, and the container acts as a
	stack of cards.
BorderLayout	BorderLayout lays out a container, arranging and resizing
	its components to fit in five regions: north, south, east,
	west, and center. Each region may contain no more than
	one component, and is identified by a corresponding
	constant: NORTH, SOUTH, EAST, WEST, and CENTER.
BoxLayout	BoxLayout allows multiple components to be laid out
	either vertically or horizontally. The components will not
	wrap so, for example, a vertical arrangement of
	components will stay vertically arranged when the frame is
	resized. Nesting multiple panels with different
	combinations of horizontal and vertical gives an effect
	similar to GridBagLayout, without the complexity.
FlowLayout	A flow layout arranges components in a left-to-right flow,
	much like lines of text in a paragraph. Flow layouts are
	typically used to arrange buttons in a panel. It will arrange
	buttons left to right until no more buttons fit on the same
	line.

## 6.8 New UI Page

We will now design a new Login UI page using Window Builder. For this normal size screen, we will continue with the default (absolute) layout. We are going to have an image, two labels, one text field, one password field and a button on the screen.

To display image use CLabel widget. CLabel supports aligned text and/or an image and different border styles.

As shown below, click *CLabel* once and keep your cursor on the screen and click. Now, the *CLabel* is placed on the screen.



Figure 6.11: New Login UI

Let us attach an image with *CLabel*. For this, you need to have an image in the folder where your *MainWindow* source file is placed. For this example, I have used eclipse logo.

Click on the *CLabel* and then, in the *Properties* window select *image*.





Figure 6.12: CLabel Image

You will now see the Image chooser window pops up. Select *Classpath resource* option and navigate to the image file, select it and then click *OK*.



Figure 6.13: Select Image

Adjust the field bounds according to the size of the logo so that the image is visible on the screen.



Figure 6.14: Image Attached

Similarly, add Labels, Text Fields and a Button. Finally the screen will be looking like the one shown below.

MainWindow.	java 🛛						-
<ul> <li>Structure</li> </ul>	ure ====	9 🖬 🖬 🥔 😒	* 🗈 🖪	× 🖂	- 🐨		
tomponents 2	ΞΞ	Palette		1			
✓ ShlLogin	- "Login"	Choose co 37 Tab	Order 🔺				
Iabel ·		Composites			Login		
🔁 IblUse	ername - "Username"	Composite 📑 Gro	oup	-	1 Login		
🔁 IblPas	ssword - "Password"	ScrolledCo Sas	hForm				
text 🖞		TabFolder 🔲 Tab	oltem				eclipse
text_1		CTabFolder CTa	abitem				
🗀 btnLo	ogin - "Login"						
			anner				
		Cayouts				Ucornamo	
Properties	10 3 3 K	🖁 🖁 🖁 Absolute Ia 📃 Fill	Layout			Usemanie	I
Variable	shill ogin		mLayout			Password	
Constructor	(Constructor prop	💾 RowLayout 🛛 🗐 Sta	ckLayout				
<b>≣</b> Style	[CLOSE, MIN, TIT	Here FlowLayout Bo	Lavout				Login
Layout	(absolute)						Login
Class	org.eclipse.swt.wi	BorderLayout					
bindings	0	Controls				1 - Alt	
alpha	255	🔁 Label 🏼 💭 Tex	t			(m	Jour Ondo Ocolio
background	240,240,240	Combo But	ton			(Dee)	JUVA LUUG DEGNA
backgroundlm	•••	Check Button @ Rac	tio Button				AVA 2 JAVA DEVELOPERS RESOURCE CENTER
hackamundM		E encer outon e nac	V				1.8.0.r45x201506
🔚 Source 📃 D	esign 📎 Bindings						

Figure 6.15: Login UI Page

To test this design, right click on the window and select *Test/Preview* from the popup menu.



Figure 6.16: Test GUI

MainWindow.ja	ava 🛙		- 8
Structure     Components	re	9     2     3     5     2     3 </th <th></th>	
v 🔄 shlLogin -	"Login"	Schoose co 13 Tab Order	
- label -		Composites	1
CalibiUser	rname - "Username"	Composite Group	
Calible Ible ass	sword - "Password"		
text		🛛 Login — 🗆 🗙 Andrea State	
text_1	gin - "Login"	Username	
	····	Password Login	
Variable	shiLogin		
± Constructor	(Constructor prop		
I Style	(absolute)	Login	
Cher	(absolute)		1
T hindings	n		5
aloha	255	10 million	
background	233		0
backaroundim			S
backgroundM	INHERIT_NONE	TableColumn     TableColumn     TableColumn     TableColumn     TableColumn	I.R.



# 6.9 Source View

Click Source tab to see the code generated by the IDE. Single line of code in this was not written manually.

```
protected void createContents() {
    shlLogin = new Shell(SWT.CLOSE | SWT.TITLE | SWT.MIN);
    shlLogin.setSize(450, 300);
   shlLogin.setText("Login");
    CLabel label = new CLabel(shlLogin, SWT.NONE);
    label.setImage(SWTResourceManager.getImage(MainWindow.class, "/com/jcg/rca/main/eclipse logo.png"));
   label.setBounds(176, 10, 106, 70);
   label.setText("");
   Label lblUsername = new Label(shlLogin, SWT.NONE);
    lblUsername.setBounds(125, 115, 55, 15);
   lblUsername.setText("Username");
   Label lblPassword = new Label(shlLogin, SWT.NONE);
   lblPassword.setBounds(125, 144, 55, 15);
   lblPassword.setText("Password");
    text = new Text(shlLogin, SWT.BORDER);
   text.setBounds(206, 109, 173, 21);
    text_1 = new Text(shlLogin, SWT.BORDER);
                                                                              Java Code Geeks
   text_1.setBounds(206, 144, 173, 21);
   Button btnLogin = new Button(shlLogin, SWT.NONE);
   btnLogin.setBounds(206, 185, 75, 25);
   btnLogin.setText("Login");
```

Figure 6.18: Source View

### 6.10 Button Listener

Attach listener with the button to validate field entries. Refer the source code of the main file given below.

#### MainWindow.java

```
package com.jcg.rca.main;
import org.eclipse.swt.SWT;
import org.eclipse.swt.custom.CLabel;
import org.eclipse.swt.widgets.Button;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Event;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.widgets.Listener;
import org.eclipse.swt.widgets.MessageBox;
import org.eclipse.swt.widgets.Shell;
import org.eclipse.swt.widgets.Text;
import org.eclipse.wb.swt.SWTResourceManager;
public class MainWindow {
        protected Shell shlLogin;
        private Text userNameTxt;
        private Text passwordTxt;
        private String userName = null;
        private String password = null;
        /**
        * Launch the application.
```
```
* @param args
 */
public static void main(String[] args) {
        try {
                MainWindow window = new MainWindow();
                window.open();
        } catch (Exception e) {
                e.printStackTrace();
        }
}
/**
 * Open the window.
*/
public void open() {
        Display display = Display.getDefault();
        createContents();
        shlLogin.open();
        shlLogin.layout();
        while (!shlLogin.isDisposed()) {
                if (!display.readAndDispatch()) {
                        display.sleep();
                }
        }
}
/**
* Create contents of the window.
*/
protected void createContents() {
        shlLogin = new Shell(SWT.CLOSE | SWT.TITLE | SWT.MIN);
        shlLogin.setSize(450, 300);
        shlLogin.setText("Login");
        CLabel label = new CLabel(shlLogin, SWT.NONE);
        label.setImage(SWTResourceManager.getImage(MainWindow.class, "/com/jcg/rca/ ↔
           main/eclipse_logo.png"));
        label.setBounds(176, 10, 106, 70);
        label.setText("");
        Label lblUsername = new Label(shlLogin, SWT.NONE);
        lblUsername.setBounds(125, 115, 55, 15);
        lblUsername.setText("Username");
        Label lblPassword = new Label(shlLogin, SWT.NONE);
        lblPassword.setBounds(125, 144, 55, 15);
        lblPassword.setText("Password");
        userNameTxt = new Text(shlLogin, SWT.BORDER);
        userNameTxt.setBounds(206, 109, 173, 21);
        passwordTxt = new Text(shlLogin, SWT.BORDER | SWT.PASSWORD);
        passwordTxt.setBounds(206, 144, 173, 21);
        Button btnLogin = new Button(shlLogin, SWT.NONE);
        btnLogin.setBounds(206, 185, 75, 25);
        btnLogin.setText("Login");
        btnLogin.addListener(SWT.Selection, new Listener() {
                public void handleEvent(Event event) {
```



#### 6.11 Conclusion

From this example, we learned how quickly a Java GUI application can be developed using Eclipse Window Builder. Window-Builder Engine provides a rich API for creating UI designers. It supports Java-based UI frameworks such as Swing, SWT/RCP, eRCP, GWT etc. It also supports XML-based UI frameworks like XWT, GWT UiBinder, Android etc.

#### 6.12 Download the Code Project

This was a Tutorial about Eclipse Window Builder for GUI Creation.

Download You can download the full source code of this example here: WindowBuilderExample

# **Chapter 7**

# **Eclipse Rich Client Platform (RCP) Tutorial**

#### 7.1 Introduction

In this example, we will learn how to use Eclipse Rich Client Platform (RCP) to develop stand alone applications written in Java and built on top of Eclipse platform technologies. Using RCP, programmers can develop customized window applications, menus, tool bars, palettes, wizards and other specialized features.

#### 7.1.1 What is Rich Client Platform?

The minimal set of plug-ins needed to build a rich client application is collectively known as the Rich Client Platform. This is a platform for building client applications with rich functionality.

#### 7.1.2 Why Eclipse RCP?

RCP is a collection of lower-level frameworks. It is a well-suited platform for Java based desktop applications. The basic advantage of Eclipse RCP is module re-usability. Not just class re-usability but full component reuse. Eclipse architecture make this very much possible than ever before.

Applications written with RCP are completely portable and will run equally well on Windows, Mac or Linux. RCP development has been made simple with Eclipse 4.x API compared to Eclipse 3.

This example is tested with *Eclipse (Mars) for RCP and RAP Developers* IDE. Before we start, please make sure you have the tools mentioned below are installed in your system. This example assumes basic programming knowledge in Java programming language using Eclipse IDE is a plus.

#### System requirements

Tools required to run this example are:

Eclipse

Download Eclipse for RCP and RAP Developers from here. Please refer the picture given below to identify the correct IDE.



Figure 7.1: Eclipse IDE for RCP and RAD

#### Java

Download Java SE 7 or above from here Let's start:

#### 7.2 Open New Project

This needs to be created as new Eclipse 4 Application Project. For that open File - New - Other

File	Edit Source Refactor N	Vavigate Search	Projec	t Run	Window	w Help	р		
	New Open File	Alt+Shift+N >		Plug-in	Project Project			$( \Leftrightarrow \bullet \Rightarrow \bullet \bullet$	
	Close	Ctrl+W		Project.					
	Close All	Ctrl+Shift+W	Ċ	Task					
	Save Save As Save All Revert	Ctrl+S Ctrl+Shift+S		Compo Product Target I Packag	nent Defi t Configu Definition e				
12 8	Move Rename Refresh Convert Line Delimiters To	F2 F5 >		Class Interfac Source File Folder	e Folder				
•	Print	Ctrl+P	-9	Exampl	e				
	Switch Workspace Restart	>		Other			Ctrl+N		
è	Import								
2	Export							. Oooko	
	Properties	Alt+Enter	JAVA COUE GEEKS						
_	Exit					1			

#### Plug-in Development - Eclipse



## 7.3 Eclipse 4 Application Project

Select Eclipse 4 Application Project and click Next

New				813			×
Select a wizard						_	$\diamond$
Create an Eclipse 4 application project							-
<u>W</u> izards:							
type filter text							
> 🗁 General							^
<ul> <li>Eclipse 4</li> <li>Eclipse 4 Application Project</li> </ul>							
> 🦢 Model							
> 🗁 Git							
> 🦳 Java	1	1					
> 🗁 Maven	(JOG)		ava (	:ode	G	eks	
<ul> <li>&gt; Plug-in Development</li> <li>&gt; &gt; SWTBot</li> </ul>		JAM	a 2 Java Devi	LOPERS RE	SOURC	I CENTER	<b>_</b>
? < <u>B</u> ack	<u>N</u> ext >		<u>F</u> inis	h		Cance	I

Figure 7.3: Eclipse 4 Application Project

#### 7.3.1 Enter Project Name

Leave other default values and click Next

New Plug-in Project	t						0	19		Х
Project name: com.jc	:g.rcp.ex	:1								
Use <u>d</u> efault locatio	n									
Location: F:\gmr\java	a\IDE\ec	lipse-ro	p-m	nars-work	spac	e\ws1\co	om.jcg.r		B <u>r</u> owse.	
Project Settings Create a <u>J</u> ava projec	ct									
Source folder:	src									
O <u>u</u> tput folder:	bin									
Target Platform This plug-in is targete <u>E</u> clipse versio	Target Platform This plug-in is targeted to run with:									
. ● <u>a</u> n OSGi fram	ework:	Equin	ох	~						
Working sets										
W <u>o</u> rking sets:							~		S <u>e</u> lect	
				Gog		Java	Cod	B	Geek	<b>S</b>
?	< <u>B</u> ac	k		<u>N</u> ext >		<u>F</u> in	ish		Cance	el

Figure 7.4: Project Name

# 7.4 Project Properties

Leave other default values and click Next

New Plug-in Project			
Content Enter the data required to	generate the plug-in.		1
Properties			1
<u>I</u> D:	com.jcg.rcp.ex1		
Version:	1.0.0.qualifier		
N <u>a</u> me:	Ex1		
Ven <u>d</u> or:	JCG	~	
Execution Environment:	JavaSE-1.8 ~	Envi <u>r</u> onments	
Options <u>G</u> enerate an activator, Activator: com.jcg. <u>Th</u> is plug-in will make Enable A <u>P</u> I analysis	, a Java class that controls the plug-in's life cycl rcp.ex1.Activator e contributions to the UI	e Geeks Durce Center	
?	< Back Next > Finish	Cancel	]

Figure 7.5: Project Properties

## 7.5 Project Configuration

Make sure Create sample content(parts, menu etc.) check box is checked and click Finish

New Plug-in Project		×
Eclipse 4 Application	-	
Configure application with special values.	1	
Product		
Name:* com.jcg.rcp.ex1		
Properties		
CSS Style: css/default.css		
Preference Customization:		
Enable development mode for application model		
Template option		
Create sample content (parts, menu etc.)		
Add a lifecycle class		
LifeCycle class name: E4LifeCycle		
Java Code	Geeks	<b>S</b>
( <u>Back</u> <u>Next</u> > <u>Finish</u>	Canc	el

Figure 7.6: Application Configuration

## 7.6 RCP Application

Yes, a basic RCP application using built-in template has been created.

<ul> <li>Package Exp X Plug-ins</li> <li>Plug-ins</li> </ul>	<ul> <li>com.jcg.rcp.ex1.product</li> <li>Application</li> <li>Add-ons</li> <li>Binding Contexts</li> <li>Binding Tables</li> <li>Handlers</li> <li>Commands</li> <li>Commands</li> <li>Command Categories</li> <li>Windows and Dialogs</li> <li>Part Descriptors</li> <li>Menu Contributions</li> <li>Toolbar Contributions</li> <li>Trim Contributions</li> <li>Snippets</li> </ul>	Application      ID org.eclipse.e4.ide.application      Binding Contexts      Add Remove      Binding Context - In Dialog and Windows
	Java Code Geeks	To Be Rendered Visible Default Supplementary

Figure 7.7: Basic Application

#### 7.7 Structure of Eclipse 4 RCP application

Eclipse Version 4 has introduced many new concepts and APIs, such as the application model, dependency injection (DI), and CSS styling. The structure of the application is described via the application model in the Application.e4xmi file. Views, menus, and toolbars of your application can be defined in this file.

Open Application.e4xmi file and navigate to *Part - Sample Part* as shown in the picture below. This is the default class gets executed and shown on the window as view part.

As per Eclipse 4 application model, Parts are the UI components which can be used to navigate and modify data. All Parts can be stacked next to each other or it can be positioned.



Figure 7.8: Structure of the Application

## 7.8 Run RCP Application

We will see how to run this application before we add our own components in this application. To run, double click on the product file as shown in the picture below (1). Launch your Eclipse application by pressing on *Launch an Eclipse application* hyperlink (2) from the *Overview* tab.



Figure 7.9: Run Application

It can also be executed by selecting  $Run As \rightarrow Eclipse Application$  as depicted below.

😫 Package Explorer 🛿 💲	Plug-	ins 🗆 🗖	com.jcg.rcp.ex	1 📰 Ap	pplication.e4xmi	eom.jcg.rcp	ex1.product 🖾
		🖻 😫 🚏 ▽	Version: 1.0.0.c	ualifier			
✓			Name: com.i	ra.rcn.ex1			
> 🛋 JRE System Library		New		>	e launcher artifact	s	
> 🚔 Plug-In Dependent		Open		F3			
> 🗁 css		Open With		>			
> 🗁 icons		Show In	Alt	+Shift+W >	nching product ex	tension identifier a	nd application.
META-INF MANIFEST.MF		Copy Ctrl+C			product		
Application.e4xmi	B	Copy Qualified Name			i.workbench.swt.E4	4Application	
build.properties	B	Paste		Ctrl+V	based on:	plug-ins O	features
com.jcg.rcp.ex1.pr	ĸ	Delete		Delete			
- pignixin	Ð	Remove from Context	Ctrl+Alt+Sh	ift+Down	-		Exporting
U	d	Mark as Landmark	Ctrl+Alt+	Shift+Up	ration with the pro	duct's defining	Use the Eclipse Pro
		Build Path		>			the product define
		Refactor	Al	t+ <mark>Shift+T</mark> >	thing a runtime ins	stance of it:	<b>T</b>
	~	Import	1 0. 1. 0.	Ocalia	ipplication		Platform Wiki Pag
	20	Export	Iava code Gi	BEKS	Ication upplication in Dobu	a mode	
	~			E CENTER	ication in Debug n	node	
	Se	Refresh		F5	ication in Debug in	noue	-
		Assign Working Sets			tion Launching S	Splash Branding	Customization Licens
		Validate 2			m State E Con		
L		Run As		>	1 Eclipse App	olication Alt-	+Shift+X, E
		Debug As		>	Run Configu	rations	
		Team					

Figure 7.10: Run As Eclipse Application

# 7.9 Stand-alone application

Separate desktop application is up and running.

🔝 com.jcg.rcp.ex1	-		Х	-		
File Help				$ \bigcirc \bullet \bullet \bigcirc$	*	
• 🗟				lication.e4xm	ni 📄 com.	jcg.rcp.ex
Sample Part						
Enter text to mark part as dirty						
Sample item 1				🛯 launcher art	ifacts	
Sample item 2						
Sample item 3						
Sample item 4				r ching produc	ct extension ider	ntifier and
Sample item 5				.product		
				i workbench.s	wt.E4Application	n
				based on:	Plug-ins	⊖ fea
				ration with the	e product's defir	ning
				ning a runtim	e instance of it:	
	U La	unch an	Eclipse	application		
Callere Orde Orelie	O La	unch a F	AP App	lication		
(Acc.)   <b>JAAA COOG PEEK2</b>	称 <u>La</u>	unch an	Eclipse	application in [	<u>Debug mode</u>	
JAVA 2 JAVA DEVELOPERS RESOURCE CENTER	校 La	unch a F	RAP App	lication in Deb	ug mode	

Figure 7.11: Running Application

## 7.10 Create New Part

Now, we will see how to add our own Part in the application. As shown in the picture, click on the product name - New - Other:



Figure 7.12: Add New Part

Select New Part Class and click Next

🖨 New		-		×
Select a wizard				$\diamond$
Create an Eclipse 4 part class				4
Wizards:				
type filter text				
<ul> <li>✓ General</li> <li>☐ File</li> <li>☐ Folder</li> <li>☑ Project</li> <li>☑ Untitled Text File</li> <li>✓ Eclipse 4</li> <li>☑ Eclipse 4 Application Project</li> <li>✓ Classes</li> <li>☑ New Addon Class</li> <li>☑ New Handler Class</li> <li>☑ New Part Class</li> <li>☑ New Tool Control Class</li> </ul>	ct	Java 2 Java Developers re	Geeks Source Center	~
? Kack	Next >	Finish	Canc	el

Figure 7.13: New Part Class

Enter class name and click Finish

New contribution of	class	
New Part		0
Create a new part clas	S	9
Source folder	com.jcg.rcp.ex1/src	Browse
Package	com.jcg.rcp.ex1.parts	Browse
Name	Orders	
PostContruct Method	postConstruct	
Predestroy Method	preDestroy	
Focus Method	onFocus	
Persist Method	save	
	Java Code Geeks	
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

Figure 7.14: Part Class Name

New Part is created with two annotations @Inject and @PostConstruct. @Inject marks a constructor, method, or field as being available for injection and methods annotated with @PostConstruct are called after an object has been fully injected.

a com.jo	:g.rcp.ex1.product 🚺 Orders.java 🔀
1	
2 pa	ckage com.jcg.rcp.ex1.parts;
3	
4⊕ im	port javax.inject.Inject;
7	
8 pu	blic class Orders {
90	@Inject
10	<pre>public Orders() {</pre>
11	
12	}
13	
140	@PostConstruct
15	<pre>public void postConstruct(Composite parent) {</pre>
16	
17	}
18	
19	17-14
2.0	( not only only only only only only only only
21	
22 }	JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

Figure 7.15: New Part Class

## 7.11 Add New Part

New Part created above should be attached to the stack to view. Open Application.e4xmi and navigate to *Part Stack*. Click *Add* button.

Image: Comjcg.rcp.ex1 > Image: System Library [JavaSE-1.8]	> BindingTables	Part Stack
<ul> <li>Plug-in Dependencies</li> <li>src</li> <li>css</li> <li>cons</li> <li>META-INF</li> <li>Application.e4xmi</li> <li>build.properties</li> <li>com.jcg.rcp.ex1.product</li> <li>plugin.xml</li> </ul>	<ul> <li>Commanas</li> <li>Command Categories</li> <li>Windows and Dialogs</li> <li>Trimmed Window - com.jcg.rcp.ex</li> <li>Main Menu</li> <li>Handlers</li> <li>Windows and Dialogs</li> <li>Controls</li> <li>Serspective</li> <li>Windows and Dialog</li> <li>Controls</li> <li>Serspective</li> <li>Windows and Dialog</li> <li>Controls</li> </ul>	ID Accessibility Phrase Selected Element Container Data Parts Part On Add Remove Dow Part - Sample Part
	> Part - San Shared Elements > TrimBars Part Descriptors	To Be Rendered G Visible Visible Visib

Figure 7.16: Attach Part

New Part form is opened. Enter Part name and click Find button of Class URI. Select the new part Orders just created and click

#### OK.

Find Contribution Find Contribution	on Class					
				plication.e4xmi 🔀		
Scope 💿 Pro	oject Only O Project and References	⊖ Workspace ⊖ Target	Platform	Part Part		
Scope Filter 🛛 No	one Bundle Package Locatio	n			com.jcg.rcp.e	x1.part.order
Class Name	to start search			Label	Order	
Type of the type of type of the type of type o				Accessibility Phrase		
9.0	om.jcq.rcp.ex1.handlers.OpenHandler -	com.jcq.rcp.ex1 - Jay		Tooltin		
9	om.jcg.rcp.ex1.parts.Orders - com.jcg.rc	p.ext - Java		icomp		
Ga	om.icg.rcp.ex1.parts.SamplePart - com.	ico.rcp.ex1 - Java		Icon URI		G Find
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			Class URI		🔾 Find
?	Clear Model Cache	OK	Cancel	Container Data		(2)
		winaows an	a Dialogs	ToolBar		$\smile$
p.ex1.parts		✓ Controls	-	Closeable		Jovo Codo Cooko
Part.java		🗸 📄 Part Sash	Container	To Be Rendered	Allee)	JAVA LUUE GEEKS
nplePart		V D Part	Stack	Visible		JAVA 2 JAVA DEVELOPERS RESOURCE CENTER
		> 🗆 P	art - Sample I	Binding Contexts		Y

Figure 7.17: New Part Class

Now, newly created Part has been attached with the application. Click Ctrl+S to save and run the application as explained in step 8. You can see the new Part has been attached next to *Sample Part* 



Figure 7.18: New Part Attached

## 7.12 Add Controls on Part

We will use WindowBuilder Editor to add controls on Part. To open WindowBuilder Editor, right click on the newly created class name *Orders* and open with WindowBuilder Editor.



Figure 7.19: WindowBuilder Editor

Click on Design tab

💼 com.jcg.rcp.ex1.product 🛛 🕽	Orders.java	Application.e	4xm i	er Orders.java ⊠	- 8
Structure ====	9 🖻 🗃	25 8 0	13	《 ☴   🚱 ▾	
🎦 Components 🛛 🕀 🖃	Palet	te			
parent in postConstruct()	🗁 System		^		
	Selection	□ Marquee			
	Choose co.	. 3+2 Tab Order			
	🔁 Composite	s			
	Composite	📄 Group			
	ScrolledCo.	. SashForm			
	TabFolder	Tabltem			
	CTabFolder	CTabltem			
🔲 Properties 🛛 🔭 💀	UiewForm	CBanner			
	🗁 Layouts				
	W H Absolute I	FillLayout			
	# GridLayout	F FormLayout			
	H RowLayout	StackLayout			
<no properties=""></no>	Here FlowLayout	BoxLayout			
	BorderLay			10	
	🗁 Controls				Cooke
~	🔁 Label	🏥 Text	~	< JAWA 2 JAWA DEVEROPERS RESS	NIRCE CENTERS
🗐 Source 📰 Design 📎 Bindings					



Now, your Part can be decorated with required controls and design elements.

## 7.13 Export application

Yes, finally we want our application to be executed as a separate application away from Eclipse platform. RCP application can be exported as a separate product and executed out of eclipse platform. To export click on the *Eclipse Product export wizard* hyperlink from the same *Overview* tab.

🔂 com.jcg.	rcp.ex1 📰 Application.e4xmi 📄 com.jcg.rcp.ex	x1.product 🛛 🗖 🗖		
This section	on describes general information about the product.	·		
ID:				
Version:	1.0.0.qualifier			
Name:	com.jcg.rcp.ex1			
The pro	oduct includes native launcher artifacts	Java Code Geeks		
Product D	efinition	JAVA 2 JAVA DEVELOPERS RESOURCE CENTER		
This section	on describes the launching product extension identifier and	d application.		
Product:	com.jcg.rcp.ex1.product	~ New		
Applicatio	tion: org.eclipse.e4.ui.workbench.swt.E4Application			
The <u>produ</u>	uct configuration is based on:	atures		
Testing		Exporting		
1. <u>Synch</u> plug-	nronize this configuration with the product's defining in.	Use the <u>Eclipse Product export wizard</u> to package and export the product defined in this configuration.		
2. Test t	he product by launching a runtime instance of it:			
0	Launch an Eclipse application	To export the product to multiple platforms see the <u>Cross</u> <u>Platform Wiki Page</u> .		
Overview C	Contents Configuration Launching Splash Branding Cu	ustomization Licensing Updates		

Figure 7.21: Export Application

Export pop-up window appears. Enter destination directory path where this stand-alone application needs to be exported. Click *Finish* 

duct - Eclipse	🖨 Export — 🗆 🗙	
lp	Eclinse product	
868 -	Use an existing Eclipse product configuration to export the product in one of	
om.jcg.rc	the available formats.	- E
Version: 1.0	Product Configuration	
Name: co	Configuration: /com.jcg.rcp.ex1/com.jcg.rcp.ex1.product ~ Browse	
The prod	Root directory: eclipse	
Product Def	Synchronization	
This section Product:	Synchronization of the product configuration with the product's defining plug-in ensures that the plug-in does not contain stale data.	~ New
Application:	Destination	~
me <u>product</u>	Directory: F:\eclipsercp\ex1      Browse	
Testing	O Archive file: V Browse	o package and export
plug-in.	Export Options	n.
2. Test the	Export source: Generate source bundles	
	Generate p2 repository	orms see the <u>Cross</u>
₩ La	Allow for binary cycles in target platform	
校 <u>La</u>	(Log) Java Code Geeks	
Overview Cor	JAVA 2 JAVA DEVELOPERS RESOURCE CENTER	

Figure 7.22: Product Configuration

## 7.14 Completed Application

The application is built and by default the windows version of this application is copied in the destination directory. Click on the *eclipse* icon to run the application away from eclipse platform.



Figure 7.23: Run Separate Application

#### 7.15 Conclusion

We have shown you a way to define the general design of an application in a consistent way using Eclipse 4 RCP API. The Eclipse 4 Application Platform provides you the foundation to build whatever you want as a plug-in or as a stand alone application. In this example we have seen how to create our own view Part and attach with the application.

# **Chapter 8**

# How to Install and Use the Eclipse Marketplace Plugin

In this example, we will see how to install and use Eclipse Marketplace Client Plugin from within Eclipse IDE.

#### 8.1 Introduction

The Eclipse community has many third-party plugins and these plugins can be added to the individual Eclipse installation. But, in earlier version of Eclipse, this was not an easy way to discover and install these solutions from within Eclipse.

The Eclipse Foundation operates a website, called Eclipse Marketplace, the *App Store* for Eclipse apps, that provides a listing of Eclipse-based solutions. The listings allow each solution provider to specify a P2 repository for their solution. Eclipse users now have a central catalog to find Eclipse solutions but the install process is still not tightly integrated with the Eclipse workspace.

To make this process simple, Eclipse MarketPlace Client (MPC) provides the tight install integration between the Eclipse workspace and Eclipse Marketplace, plus other third party solution listings. It is a new feature that allows Eclipse users to discover and install Eclipse solutions directly into their Eclipse installation.

#### 8.2 Install Eclipse MarketPlace Client

From Eclipse Juno, the MPC is already included. But, it would be a great idea if we can have MPC on older versions also.

Please follow the steps below to configure market place in the older versions of Eclipse IDE.

Open  $Help \rightarrow Install New Software$ 

Help	2			
0	Android IDE		- 🗢 🗢	• → →   =
<b>?</b>	Help Contents Search Dynamic Help			
	Key Assist Tips and Tricks Cheat Sheets	Ctrl+Shift+L		
	Check for Updates		1	Java Oada Oaaka
	Install New Software		Rag)	JAVA COUL GEEKS
	About ADT			
	Help	Help         Image: Android IDE         Image: Android IDE         Image: Help Contents         Search         Dynamic Help         Key Assist         Tips and Tricks         Cheat Sheets         Check for Updates         Install New Software         About ADT	Help         Image: Provide the stress of t	Help   Image: Problem in the pice of

Figure 8.1: Install New Software

And select *Kepler - https://download.eclipse.org/releases/kepler*, or *Helios - https://download.eclipse.org/releases/helios* from the *Work with* field according to the version of your IDE.

Then select General Purpose Tools - Marketplace Client as shown below.

() Install		
Available Software Check the items that you wish to install.		
Work with: Kepler - http://download.eclipse.org/releases/kepler	~	<u>A</u> dd
	Find more software by working with the <u>"Available Software Site:</u>	_ preferences
type filter text		
Name	Version	^
🔲 🖗 Eclipse Plug-in Development Environment	3.9.1.v20140221-1700	
🔲 窷 Local Terminal (Incubation)	0.2.300.201307231220	
🔲 窷 m2e - Maven Integration for Eclipse	1.4.0.20130601-0317	
🗌 🙀 m2e - clf4j ever legback logging (Optional)	1.4.0.20130601-0317	
🗹 🎲 Marketplace Client	1.2.1.v20140219-1000	
Remory Analyzer	1.3.1.201401071412	
🔲 🌆 Memory Analyzer (Charts) [optional]	1.3.1.201401071412	
🗌 🌆 OCL End User SDK	4.1.1.v20140210-1137	~
Select All Deselect All 1 item selected		
Details		
The Eclipse Marketplace Client provides access to extension catalogs.		More.
Show only the latest versions of available software	✓ Hide items that are already installed	
Group items by category	What is already installed?	
Show only software applicable to target environment		
	Java Code Geeks	
(?)	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 8.2: Select MPC

Accept terms and install the plug-in. You will be asked to restart the IDE. Please do so.

Now, you will see a new menu *Eclipse Marketplace*... under help menu as shown below. Click on it. .Marketplace Client Menu Marketplace Client Menu

#### 8.3 Eclipse MarketPlace Wizard

Yes, you have Eclipse Marketplace within your IDE now. You can install plugins as per your requirement from the Marketplace shown in the wizard.

🚺 Eclipse Mark	ketplace — 🗆	×			
Eclipse Marketplace A					
Select solution: Press the inform	s to install. Press Finish to proceed with installation. mation button to see a detailed overview and a link to more information.	3			
Search Recen	t Popular Installed 🖓 December Newsletter				
Find:	All Markets      ✓ All Categories      ✓	<u>G</u> o			
Featured		^			
	Vaadin Plugin for Eclipse				
vaadin }>	<b>Promoted</b> - Vaadin is a powerful open source UI library for creating rich web user interfaces. It is a Java EE compatible component library that makes it easy <u>more info</u>				
	by Vaadin Ltd, Apache 2.0				
	java J2EE web ria java ee	1			
★ 44	Installs: 114K (3,340 last month) Install				
	Optimizer for Eclipse				
	<b>Promoted</b> - Your Eclipse is slow. Optimizer for Eclipse speeds up your IDE by finding and fixing common configuration issues in your Eclipse installation. These <b>more info</b>				
	by ZeroTurnaround, Commercial - Free				
	Eclipse slow startup performance faster optimizer				
+ 25	Installs: 79 4K (9 045 last month) Install	Υ.			
Markatala					
warketpia	ices				
	Java Code Geeks				
?	< <u>B</u> ack <u>Install Now &gt; Finish</u> Cancel				

Figure 8.3: Eclipse Marketplace Client

## 8.4 Search Eclipse Solutions

There is a very effective feature of searching the solutions in the MarketPlace.

In the existing IDE, there is no support for Java 8. Now, from this MarketPlace, we will install Java 8 support plugin.



#### Figure 8.4: Java 7 Latest

Please look at the picture given below. To enable Java 8 support in the existing IDE which was not there by default, enter Java 8 and click search icon.



Figure 8.5: Search Solution

In the wizard, you will see available plugins to enable Java 8 support. Click *Install* button of *Java 8 support for Eclipse Kepler SR2* plugin.



Figure 8.6: Java 8 Support

Select required features as shown below and click Confirm button.

Also, it is good to notice that you can install multiple plugins on the same flow by pressing Install More button.



Figure 8.7: Select Plugins

Accept terms and click Finish

?



Figure 8.8: Accept Terms

< Back

Next >

I do not accept the terms of the license agreement

Finish

Cancel

It may take some time to update your IDE with Java 8 support plug-in. You will be asked to restart the IDE. Now, after restarting the IDE if you look at the compiler version, you will be able to see Java 8 compliance level is enabled as depicted below.



Figure 8.9: Java 8 Support

#### 8.5 Search by Market

The search conditions can be still reduced by specifying Market or Category.

A							
🕖 Eclipse Mar	ketplace	- 0	×				
Eclinse Mar	ketnlace	5					
Compse main	verhidee	54	1				
Select solution	is to install. Press Finish to proceed with installati mation button to see a detailed oveniew and a l	ink to more information	1				
Fress the infor	mation button to see a detailed overview and a r		_				
Search Recer	nt Popular Installed December Newsletter						
Find:	🔍 All Markets 🗸 🗸	All Categories 🛛 🗸 🗸	Go				
	All Markets		0.00				
Featured	Eclipse Project		^				
reaturea	I ools BCD Applications						
	Vaadin Plug Long Term Support						
	vadan i neg song rem seppore						
	Promoted - Vaadin is a powerful open source	E UI library for creating rich web					
vaadin }>	easy more info	bonent library that makes it					
	hu Vandin Ital Annaha 2.0						
	iava IZEE web ria java ee						
-							
★ 44	Installs: 114K (3,340 last month)	Update Uninstall					
	Optimizer for Eclipse						
-	Promoted - Your Eclinse is slow Ontimizer fo	r Eclince sneeds up your IDE by					
	finding and fixing common configuration issues in your Eclipse installation.						
	These more info						
	by ZeroTurnaround, Commercial - Free						
	Eclipse slow startup performance faster optimi	zer					
-A-25			~				
	Installs / / 4K (91/45 last month)	Install					
Marketpl	ares						
marketph							
	1						
	(mg) law	ovan Conta					
	Uav	a vouo acono					
	These T has						
0			_				
(?)	< <u>B</u> ack <u>I</u> nstall Now >	<u>Finish</u> Cancel					

Figure 8.10: Search By Market

## 8.6 Conclusion

In this example, we have seen how to install Eclipse Marketplace Client plug-in used for browsing and installing the Eclipse based solutions listed on the Eclipse Marketplace portal from within the Eclipse IDE.