

Version 1.6

Copyright © 2007 - 2009 Lars Vogel

30.12.2009

Eclipse Java IDE

This article describes the usage of Eclipse as a Java IDE. It describes the installation of Eclipse, the creation of Java programs, the usage of external jars, quick fix and content assist and the usage of the Eclipse update manager.

This article is based on Eclipse 3.5 (Eclipse Galileo).

Table of Contents

- 1. Eclipse Overview
- 2. Getting started
 - 2.1. Installation 2.2. Start Eclipse
- 3. Eclipse UI Overview
 - 3.1. Workspace
 - 3.2. Perspective
 - 3.3. Views and Editors
- 4. Create your first Java program
 - 4.1. Create project
 - 4.2. Create package
 - 4.3. Create Java class
 - 4.4. Run your project in Eclipse
 - 4.5. Run your Java program outside Eclipse (create jar file)
 - 4.6. Run your program outside Eclipse
- 5. Content Assists and Quick Fix
 - 5.1. Content assist
 - 5.2. Quick Fix
- 6. Using jars (libraries)
 - 6.1. Adding external library (.jar) to the Java classpath
 - 6.2. Show source code for jar
 - 6.3. Add the Javadoc for a jar
- 7. Updates and Installation of Plugins
 - 7.1. Eclipse Update Manager
 - 7.2. Manual installation of plugins (dropins folder)
- 8. More Tips
 - 8.1. Problems view



Eclipse Java IDE - Tutorial

- 8.2. Important Preference Settings
- 8.3. Task Management
- 8.4. Working Sets
- 8.5. Synchronize package explorer with code display
- 8.6. Code Templates

9. Next steps

- 10. Thank you
- 11. Questions and Discussion
- 12. Links and Literature

12.1. Source Code

- 12.2. Eclipse Resources
- 12.3. Other Resources

1. Eclipse Overview

Eclipse an open source community whose projects building tools and frameworks for creating general purpose application.

The most popular usage of Eclipse is as a Java development environment which will be described in this article.

UML Plugin for Eclipse

Modeling Java, Code Engine, UML 2, Use Case, ERD & Hibernate, Try Now! www.visual-paradigm.com

< >

Ads by Google

2. Getting started

2.1. Installation

Download "Eclipse IDE for Java Developers" from the website Eclipse Downloads and unpack it to a directory. This is sufficient for Eclipse to be used; no additional installation procedure is required.

Use a directory path which does not contain spaces in its name.

2.2. Start Eclipse

To start Eclipse double-click on the file eclipse.exe in your installation directory.

The system will prompt you for a workspace. The workspace is the place there you store your Java projects (more on workspaces later). Select a suitable (empty) directory and press Ok.



Eclipse will start and show the Welcome page.



Close the welcome page by press in little x besides the Welcome.

3. Eclipse UI Overview

Eclipse provides perspectives, views and editors. Views and editors are grouped into perspectives. All projects are located in a workspace.

3.1. Workspace

The workspace is the physical location (file path) you are working in. You can choose the workspace during startup of eclipse or via the menu (File-> Switch Workspace-> Others).

All your projects, sources files, images and other artifacts will be stored and saved in your workspace.



3.2. Perspective

A perspective is a visual container for a set of views and editors.

You can change the layout within a perspective (close / open views, editors, change the size, change the position, etc.)



For Java development you usually use the "Java Perspective".

3.3. Views and Editors

A view is typically used to navigate a hierarchy of information or to open an editor. Changes in a view are directly applied.

Editors are used to modify elements. Editors can have code completion, undo / redo, etc. To apply the changes in an editor to the underlying resources, e.g. Java source file, you usually have to save.

4. Create your first Java program

The following will describe how to create a minimal Java program using Eclipse. It will be the classical "Hello World" program. Our program will write "Hello Eclipse!" to the console.

4.1. Create project

Select from the menu File -> New-> Java project. Maintain "de.vogella.eclipse.ide.first" as the project name. Select "Create separate source and output folders".

New Java Project	
Create a Java Project	
Create a Java project in the workspace or in an o	external location.
Project name: de vogella eclinse ide first	
Contents	
• Create new project in workspace	
Create project from existing source	
Directory: C:\Documents and Settings\d034	797\Desktop\Documents\16_Eclips Browse
JRE	
• Use an execution environment JRE:	JavaSE-1.6
OUse a project specific JRE:	jdk1.6.0_13
OUse default JRE (currently 'jdk1.6.0_13')	Configure JREs
Project layout	
○ <u>U</u> se project folder as root for sources and	class files
⊙ ⊆reate separate folders for sources and cl	ass files <u>Configure default</u>
Working sets	
Add project to working sets	
Working sets: Birt	Select
?	k Next > Einish Cancel

Press finish to create the project. A new project is created and displayed as a folder. Open the folder "de.vogella.eclipse.ide.first"

4.2. Create package

Create now a package. A good convention is to use the same name for the top package as the project. Create therefore the package "de.vogella.eclipse.ide.first".

Select the folder src, right mouse click on it and select New -> Package.

😑 🔁 de.v	ogella.eclipse.ide.first			www.lic.clas
	New		•	/ Java Project
	Open in New Window			(S) Web Application Project
···· 🗴 .	Open Type Hierarchy	F4		🎦 Project
	Show In	Alt+Shift+W	•	🕂 Package

Eclipse Java IDE - Tutorial

😂 New Java	Package		
Java Packag Create a new .	je Java package.		
Creates folders	corresponding to packages.		
Source fol <u>d</u> er:	de.vogella.eclipse.ide.first/src		Browse
Na <u>m</u> e:	de.vogella.eclipse.ide.first		
?		Einish	Cancel

4.3. Create Java class

Right click on your package and select New -> Class

😑 😂 de.vogella.eclipse.ide.fir:	st		
🖮 进 src			
de.vogella.eclips	e.ide.first		
🗄 🗁 .settings	New	•	🖄 Java Project
.classpath	Open in New Window		🛞 Web Application Project
.project		F4	Project
	Open Type Hierarchy	F4	
	Show In	Alt+Shift+W	🖶 Package
	📄 Сору	Ctrl+C	Class

Create MyFirstClass, select the flag "public static void main (String[] args)"

😂 New Java Clas	35				
Java Class	Java Class				
Create a new Java	class.	\bigcirc			
Source fol <u>d</u> er:	de.vogella.eclipse.ide.first/src	Browse			
Pac <u>k</u> age:	de.vogella.eclipse.ide.first	Bro <u>w</u> se			
Enclosing type:		Browse			
Name:	MyEirstClass	1			
Modifiers:					
	abstract final static				
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>			
Interfaces:		<u>A</u> dd			
Which method stubs	would you like to create?				
l l	public static void main(String[] args)				
	Constructors from superclass				
Do you want to add	Interited abstract methods comments? (Configure templates and default value here)				
Do you want to add	Generate comments				
0					
0	Einish				

Maintain the following code.

package de.vogella.eclipse.ide.first; public class MyFirstClass {

public static void main(String[] args) {

4.4. Run your project in Eclipse

😑 📂 de.vogella.eclipse.ide.first public class MyFirstClass { 🚊 🗁 进 src 🚊 🖶 de.vogella.eclipse.ide.first public static void main(🗄 🕕 MyFirstClass.java New • System.out.println(" 🗄 🗁 .settings 🗴 .classpath F3 Open .project • Open With Open Type Hierarchy F4 Show In Alt+Shift+W ۲ 📄 Сору Ctrl+C 🗎 Copy Qualified Name 💼 Paste Ctrl+V Delete 💢 Delete 🕭 Remove from Context Ctrl+Alt+Shift+Down Build Path Alt+Shift+S • Source Alt+Shift+T • Refactor 🚵 Import... 🚮 Export... References ۲ Declarations ۲ Find Bugs • 🦑 Refresh F5 Assign Working Sets... Run As 🕨 📙 1 Run on Server Alt+Shift+X, R Debug As • 🗊 2 Java Application Alt+Shift+X, J

Now run your code. Right click on your Java class and select Run-as-> Java application

Finished! You should see the output in the console.



4.5. Run your Java program outside Eclipse (create jar file)

To run your Java program outside of Eclipse you need to export it as a jar file. Select your project, right click on it and select "Export".

⊟ 2 de.vogella.eclips ⊟ 2 src ⊟ ⊕ de.vogel	New Go Into	•
😨 🗾 MyFii 🗈 🛋 JRE System l	Open in New Window Open Type Hierarchy Show In	F4 Alt+Shift+W ▶
	Сору	Ctrl+C
	🗎 Copy Qualified Name	
	💼 Paste	Ctrl+V
	💢 Delete	Delete
	& Remove from Context Build Path	Ctrl+Alt+Shift+Down ▶
	Source	Alt+Shift+S
	Refactor	Alt+Shift+T 🕨 🕨
	🚵 Import	
	🛃 Export	

Select JAR file, select next. Select your project and maintain the export destination and a name for the jar file. I named it "myprogram.jar".

Eclipse Java IDE - Tutorial

Export		$\overline{\mathbf{X}}$
Select Export resources into a JAR file on the local I	ile system.	1
Select an export destination:		
type filter text		
Java JAR file Javadoc GRUNNABLE JAR file Development Development Development Development Development Development Development Development		
? < Back Next >	Finish Cancel	
😂 JAR Export		
JAR File Specification Define which resources should be exported i	nto the JAR.	
Select the resources to export:		
	 Classpath Image: Market with the second second	
Export generated class files and resource	es	
Export all output folders for checked pro	jects	
Export Java source files and resources		
Export refactorings for checked projects	. <u>Select refactorings</u>	
Select the export destination:		
JAR file: c:\temp\myprogram.jar		Browse
Options:		
Compress the contents of the JAR file		
Add directory entries		
Overwrite existing files without warning		
?	< Back Next >	Finish Cancel

Press finish. This will create a jar file in your select output directory.

4.6. Run your program outside Eclipse

Open a command shell, e.g. under Microsoft Windows select Start -> Run and type in cmd. This should open a consle.

Switch to your output directory, e.g. by typing cd path, e.g. if you jar is located in "c:\temp" type "cd c:\temp".

To run this program you need to include the jar file into your classpath. See Classpath and Java JAR Files for details.

java -classpath myprogram.jar de.vogella.eclipse.ide.first.MyFirstClass

C:\temp>java -classpath myprogram.jar de.vogella.eclipse.ide.first.MyFirstClass Hello Eclipse!

Congratulations! You created your first Java project, a package a tiny Java program and you ran this program inside Eclipse and outside

5. Content Assists and Quick Fix

5.1. Content assist

The content assistant allows you to get input help in an editor. It can be invoked by CTRL + Space.

For example type syso and then press [Ctrl + Space] and it will be replaced by System.out.println(""). Or if you have an object, e.g. Person P and need to see the methods of this object you can type p. (or press CTRL + Space) which activates also the content assist.



5.2. Quick Fix

Whenever there is a problem Eclipse will underline the problematic place in the coding. Select this and press (Ctrl+1)

For example type "myBoolean = true;" If myBoolean is not yet defined, Eclipse will highlight it as an error. Select the variable and press "Ctrn+1", then Eclipse will suggest to create a field or local variable.

Quick Fix is extremely powerful, it allows you to create new local / field variables, new methods, classes, put try and catch around your exceptions, assign a statement to a variable etc.



6. Using jars (libraries)

6.1. Adding external library (.jar) to the Java classpath

The following describes how to add external jars to your project.

The following assumes you have a jar available.

If you need an example for working with jars you can use JFreeChart Tutorial

Create a new Java project "de.vogella.eclipse.ide.jars". Create a new folder called "lib" (or use your existing folder) by right click on your project and selecting New -> Folder



From the menu select File -> Import -> File system. Select your jar and select the folder lib as target.

Select your project, right mouse click and select properties. Under libraries select "Add JARs".

The following example shows how the result would look like if junit-4.4.jar would be added to a project.

🖶 Properties for JUnitTest		_ 🗆 🗙
type filter text	Java Build Path	$\leftarrow \bullet \bullet \bullet$
Resource Builders Java Build Path	Ø Source Projects A Libraries Order and Ex JARs and class folders on the build path:	port
⊡ Java Code Style	🕀 🗍 junit-4.4.jar - JUnitTest/lib	Add JARs
⊡-Java Compiler ⊡-Java Editor	i → IRE System Library [jre1.6.0_02]	Add External JARs
– Javadoc Location – Project References		Add Variable
- Run/Debug Settings		Add Library
Task Repository Task Tags		Add Class Folder
Validation		Edit
		Remove
		Migrate JAR File
	,	
0		OK Cancel

6.2. Show source code for jar

To browse the source of a type contained in library you can attach a source archive or source folder to this library. The editor will then show the source instead of a the decompiled code. Setting the source attachment also allows source level stepping with the debugger.

The Source Attachment dialog can be reached via:

Open the Java Build Path page of a project (Projects > Properties > Java Build Path). On the Libraries page expand the library's node and select the Source attachment attribute and press Edit

Maintain the location to the source attachement.

Properties for Documentatio	n	
type filter text	Java Build Path	
Resource BeanInfo Path Builders	Source Projects Libraries Order and E JARs and class folders on the build path:	xport
Java Build Path III Java Code Style	🗉 📓 ant.jar - C:\Documents and Settings\D034797\Des	Add JARs
iava Compiler iava Editor	The System Library [jre1.6.0_01] Access rules: No rules defined	Add External JARs
- Javadoc Location	Native library location: (None) Taylor and Files' lavalite1.6.0	Add Variable
	rt.jar - C:\Program Files\Java\jre1.6.0_01\lib	Add Library
Refactoring History Run/Debug Settings	Javadoc location: http://java.sun.com/jav	Add Class Folder
SVN Info Task Tags	Access rules: (No restrictions) - non modifi	Edit
L. Validation		Remove
	Charsets.jar - C:\Program Files\Java\jre1.6.0_ Sara - C:\Program - C:\Program Files\Java\jre1.6.0_ Sara - C:\Program - C:\Program Files\Java\jre1.6.0_ Sara - C:\Program - C:\Progr	Migrate JAR File
< >		
0		OK Cancel

In the Location path field, enter the path of an archive or a folder containing the source.

6.3. Add the Javadoc for a jar

Download the javadoc of the jar and put it somewhere in your filesystem.

Open the Java Build Path page of a project (Projects > Properties > Java Build Path). On the Libraries page expand the library's node and select the Javadoc location attribute and press Edit

Maintain the location to the api.

Properties for Documentation	n	
type filter text	Java Build Path	
Resource BeanInfo Path Builders Java Build Path Java Code Style Java Compiler Java Editor Profile Compliance and Validatic Project References Refactoring History Run/Debug Settings SVN Info Task Tags Validation	Java Build Path Source Projects Libraries Order and E JARs and class folders on the build path: Order and E JARs and class folders on the build path: Image: Source Image: Source attraction of the state of the	xport Add JARs Add External JARs Add Variable Add Library Add Class Folder Edit Edit Remove Migrate JAR File
<	<	
٢		OK Cancel

7. Updates and Installation of Plugins

7.1. Eclipse Update Manager

Eclipse provides functionality via so-called features (which contain plugins). Eclipse 3.5 contains a Software Update Manager which allows you to update existing plugins and to install new plugins.

To update your existing installation select the menu Help -> Check for Updates. The system will verify if for the installed plugins updates are available or not.

To install new functionality, select Help-> Install New Software.

Eclipse Java IDE - Tutorial



Select from the list a update site from which you would like to install new software. For example if you want to install new plugins from Galileo select the Galileo Update Site.

(i)	
Sometimes you have to uncheck "Group items by category" – not all available Plugins are categorized. If they are not categorized	
they will not be displayed. See Eclipse bug .	

To add a new update site select, press the button "Add" and input the URL. This will then make this update site available and will allow you to install software from this site.

7.2. Manual installation of plugins (dropins folder)

If you're using Plugins where no Software Site is available, then you can use the Dropins folder in your Eclipse installation directory.

To do this put the plugin into Eclipse "dropins" folder and restart Eclipse. Eclipse should detect the new plugin and install it for you.

8. More Tips

8.1. Problems view

The problems view displays problems in your projects. You can open it via Windows -> Show View -> Problems

199 errors, 147 warnings, 0 others					
	Description	Resource	Path 🔺	Locat	Туре
ĺ	🖃 😣 Errors (199 items)				
	🔇 Bundle 'RfcCommon' cannot be resolve	MANIFEST.MF	de.vogella	line 7	Plug-in Prob
	🝳 Bundle 'TMRfcConnector' cannot be re	MANIFEST.MF	de.vogella	line 10	Plug-in Prob
	🝳 LaneSelection cannot be resolved to a :	ILaneDao.java	de.vogella	line 9	Java Problem
	🝳 The import selection cannot be resolve	ILaneDao.java	de.vogella	line 5	Java Problem
	🝳 LaneSelection cannot be resolved to a :	LaneDownloadDao.ja∨a	de.vogella	line 13	Java Problem

You can configure the problems view, e.g. if you only want to display the problems from the current selected project, select "Configure Contents".

Show	
Group By	• 🔼
Sort By	•
New Problems View	
Configure Contents	
Columns	
Preferences	
Pocus on Active Task	

Configure Contents		
Configurations:		▼ Scope
All Errors	New	🔿 On any element
VVarnings on Selection		On any element in same project
	Remove	On selected element only
		On selected element and its children
		○ On working set: Window Working Set
		Select
		▼ Description:
		Description contains
		Where severity is: 🗹 Error 🗌 Warning 🔲 Info
		▼ Types
		🗉 🗹 API Problems 🦳 Select All
		APT Problems
		Ant Buildfile Problem Deselect All
		Checkstyle Problem
		Classpart Dependency Validator Mess
		V Driver Problems
		EAR Validator Message
		EJB Validator Message
Match any configuration		
C match any conliguiation		

8.2. Important Preference Settings

Eclipse allows to set semicolons (and other elements) automatically.

🖶 Preferences	
type filter text	Typing $\leftrightarrow \mathbf{v} \Rightarrow \mathbf{v}$
Connectivity Help Install/Update Java Appearance Build Path Code Style Code Style Compiler	Automatically close Image: "Strings" Image: "Gradeses), [square] and <angle> brackets Image: "Gradeses" Image: "Gradeses"</angle>
⊕-Debug ⊖-Editor ⊕-Content Assist Folding	Automatically insert at correct position Semicolons Braces
Hovers Mark Occurren Save Actions Syntax Colorin Templates	Tabulators Tab key adjusts the indentation of the current line The tab display value (currently 4) and whether spaces are used to indent lines are configured on the <u>formatter preference page</u> . The current indentation mode uses tabs.
Typing ⊡-Installed JREs JUnit Properties Files Edi	When pasting Adjust indentation Update imports
⊕-Mylyn ⊕-Plug-in Development ⊕-Run/Debug ⊕-SQL Development	In string literals Image: string literal Image: string literal Image: string literal
	Restore Defaults Apply
0	OK Cancel

Eclipse allows to format the source code and to organize the imports at save.

Save Actions	ionfigure Project Spe	← ▼ → ▼
⊆ ✓ Perform the selected actions on save	ionfigure Project Spe	ecific Settinas
 Format source code Configure the formatter settings on the Form Organize imports Configure the organize imports settings on the Additional actions Add final modifier to private fields Add missing '@Override' annotations Add missing '@Deprecated' annotations Remove unnecessary casts 	<u>matter</u> page. he <u>Organize Imports</u>	2 page. Configure
	Destars Defeute	0 mmb i
		Capad
	 Organize imports Configure the organize imports settings on t Additional actions Add final modifier to private fields Add missing '@Override' annotations Add missing '@Deprecated' annotations Remove unnecessary casts 	 Organize imports Configure the organize imports settings on the Organize Imports Additional actions Add final modifier to private fields Add missing '@Override' annotations Add missing '@Deprecated' annotations Remove unnecessary casts Restore Defaults

8.3. Task Management

them again into your workspace.

If you use // TODO in the coding this indicates a task for eclipse and you find it in the task view of Eclipse.

For more advanced tasks you can use Eclipse Mylyn Tutorial .

8.4. Working Sets

A common problem in Eclipse is that your data in your workspace grows and therefore your workspace is not well structured anymore. You can use working sets to organize your displayed projects / data. To setup your working set select in the Package Explorer -> Show -> Working Sets.

🚝 Java - SixSigmaRCP/src/sixsigmarcp/actions/DeleteAction	ı.java - Eclipse SDK					
File Edit Source Refactor Navigate Search Project Run	Window Help					
] 📬 ▾ 📄 🛓 🌬 ▾ 🎋 ▾ 💽 ▾ 🎭 ▾ 🗍 🖑 🕾 🞯 ▾ .] 😂 🤒 😂 🛷] 🌛 😜] 🥹] 🞚] ½					
ቹ Package Explorer 🛛 👔 Hierarchy 🗧 🔿 💿 🔄 🙀	🝸 🗖 🕼 SixSigmaRCP 🛛 🔊 Requireme					
🗂 🖆 Cockpit	Top Level Elements					
DaliTest	Select Working Set					
Erroriest	Deselect Working Set					
i powerflower	Edit Active Working Set					
🐵 📾 > SixSigmaJars 655 [https://sixsigma.googlecode.com/svn, T	• 🔄 1 Window Working Set					
다 굶 > SixSigmaRCP 684 [https://sixsigma.googlecode.com/svn, T	1 Backage declarations					
$\square_{3} = 370000$ SrC 664	2 Static fields and methods					
⊡ clipboard 657	3 Synthetic members					
🐵 📠 contentProvider 684						
🗈 🖶 controller 657	Package Presentation					
Erandata 677	 Show 'Referenced Libraries' Node 					
⊡ database 004 ⊡ filter 677	🖉 Liek Mitte Editor					
🕀 🏭 messages 681						
🕀 🚓 reporting 681	PFocus on Active Task					

Press new on the following dialog to create a working set.



On the following dialog select java, select the source folder you would like to see and give it a name. You can now easily display only the files you want to see.



8.5. Synchronize package explorer with code display

The package explorer allows to display the associated file from the current selected editor. Example: if you working on foo.java and you change in the editor to bar.java then the display in the package explorer will change.

To activate this press "Link with Editor".

増 Package Explorer 🛛 🍣 Plug-ins	E <mark>⊈</mark> 💱 ▽ ─ 🗗
images 975 Marticle.xml 975	Link with Editor

8.6. Code Templates

If you have to type frequently the same code / part of the document you can maintain templates which can be activate via autocomplete (Ctrl + Space).

For example lets assume you are frequently creating "public void name(){}" methods. You could define a template which creates the method body for you.

To create a template for this select the menu Window->Preferences and Open Java -> Editor -> Templates

```
Eclipse Java IDE - Tutorial
```

Preferences								_ 🗆 🔀
type filter text	Т	emplate	s					⇔ • ⇒ • ▼
🕀 General 📃	C	reate, edit	or r	emove templates:				
🕀 Data Management		Name		Context	Description	Auto In	(<u>N</u> ew
⊕ Help		🖌 @auth	or	Javadoc	author name	on	≡	E dit
⊕ Install/Update		✓ 		Javadoc		on		
🗐 Java		<code:< td=""><td>></td><td>Javadoc</td><td><code></code></td><td>on</td><td></td><td>Remove</td></code:<>	>	Javadoc	<code></code>	on		Remove
Appearance		<i><i><i></i></i></i>		Javadoc	<i></i>	on		
😟 Build Path		<pre></pre>		Javadoc	<pre></pre>	on		
🕀 Code Style		🗹 addliste	B	SWT statements	add a listener to a			Restore Removed
		🗹 arraya	dd	Java statements	add an element to			
🕀 Debug 😑		🗹 arraym		Java statements	merge two arrays i			Revert to Default
= Editor		Browse	*	SWT statements	new Browser			
E Content Assist		🖌 Button		SWT statements	new Button			Import
Eolding		🖌 cast		Java statements	dynamic cast			
Houard		🖌 catch		Java	catch block			Export
Mark Organization		🖌 Combo		SWT statements	new Combo			
Mark Occurrences			cite	SW/T statements	new Composite wit		۳	
Save Actions		`					J	
- Syntax Coloring	Pr	revie <u>w</u> :						
Templates								
····· Typing								<u> </u>
FindBugs								
连 Installed JREs								
JUnit								
Properties Files Editor								
Java EE								~
🗄 Plug-in Development		<						3
⊕ Pvdev		-						
Report Design		Use code	e <u>F</u> ori	matter				
						Restore	Tefa	ults Apply
< · · · >						(Restore)	2010	
?							к	Cancel

Press New. Create the following template. \${cursor} indicates that the cursor should be placed at this position after applying the template.

ali Edit Ten	nplate 📃 🗌 🔀
Name:	npm Context: Java 🔽 🗹 Automatically insert
Description:	Creates a new public method
Pattern:	<pre>public void \$(cursor) () () </pre>
	Insert Variable
?	OK Cancel

This this example the name "npm" is your keyword.

Now every time you type the keyword in the Java editor and press Ctrl+Space the system will replace your text with your template.

	Job.scheduie();
8	public void () {
	}

9. Next steps

To learn how to debug Eclipse Java programs you can use Eclipse Debugging

To learn Java Web development you can use with Servlet and JSP development with Eclipse Web Tool Platform (WTP) - Tutorial . If you want to develop rich stand-alone Java clients you can use Eclipse RCP - Tutorial

. Check out Eclipse Plugin Development - Tutorial to learn how to develop your own plugins.

Good luck in your journey of learning Java!

10. Thank you

Thank you for practicing with this tutorial.

Please note that I maintain this website in my private time. If you like the information I'm providing please help me by donating.



11. Questions and Discussion

For questions and discussion around this article please use the www.vogella.de Google Group. Also if you note an error in this article please post the error and if possible the correction to the Group.

I believe the following is a very good guideline for asking questions in general and also for the Google group How To Ask Questions The Smart Way.

12. Links and Literature
12.1. Source Code
http://www.vogella.de/code/codeeclipse.html Source Code of Examples
12.2. Eclipse Resources
12.3. Other Resources
Eclipse.org Homepage
Articles about Java, Eclipse and Webdevelopment from www.vogella.de
Articles about Eclipse development from www.vogella.de
Articles about Web development from www.vogella.de