

### Core

Ctrl+1	Quickfix
Ctrl+3	Open "Quick Access" search
Ctrl+_	Content Assist
Alt+⇧Shift+Z	Surround with (do, for, if, try, ..)
Ctrl+Z	Undo
Ctrl+⇧Shift+Z	Redo

### Testing

Alt+⇧Shift+X, T	execute Tests	
Ctrl+R	execute Tests	<a href="#">moreUnit</a>
Ctrl+J	jump to test or implementation	<a href="#">moreUnit</a>

### Comment

Ctrl+/	Comment / uncomment line or selection (adds '//')
Ctrl+⇧Shift+7	Add Block Comment around selection (adds '/* ... */')
Ctrl+⇧Shift+ß	Remove Block Comment
Alt+⇧Shift+J	Add Element Comment (adds '/* ... */' JavaDoc)

### Text Editing

Ctrl+⇧Shift+Y	lowercase selected text
Ctrl+⇧Shift+X	uppercase selected text
Ctrl+D	Delete line
Ctrl+⇧Shift+Del	Delete until end of line
Ctrl+Del	Delete next element
Ctrl+⌫	Delete previous element
Ctrl+Alt+↓/↑	Copy line below/above
Alt+/	Propose word completion - after typing at least one letter. Repeatedly press until reaching correct name
Alt+⇧Shift+A	Column (block) selection, Multiline block editing
⇧Shift+Enter	Adds a blank line below the current line
Ctrl+⇧Shift-+Enter	Adds a blank line above the current line

### Generate

Alt+⇧Shift+S, R	Generate Getters and Setters
-----------------	------------------------------

### Movement

Alt+↑/↓	Move current line or selection up or down
Ctrl+←/→	Jump one word to the left / right
Ctrl+⇧Shift+↑/↓	Jump to previous / next method
Ctrl+L	Jump to Line Number. To hide/show line numbers, press Ctrl+F10 and select 'Show Line Numbers'
Ctrl+Q	Jump to last location edited
Ctrl+.,/	Jump to next / previous warning or error
Ctrl+↑	Scroll up
Ctrl+↓	Scroll down
Ctrl+⇧Shift+P	Go to the matching bracket
Alt+⇧Shift+↑/↓	Select the previous / next syntactical element
Alt+⇧Shift+↑/↓/←/→	Extending / reducing the selection of the previous / next syntactical element

**Tip:** Use [bookmarks](#). Add them on left margin of the code view (where you set the breakpoints) with "right click → Add bookmark". Add the Bookmarks-View (Window → Show View → Bookmarks).

### Selection

Alt+⇧Shift+↑	Expand selection to current element
Alt+⇧Shift+↓	Reduce previously expanded selection by one step

### Search / Information

Ctrl+⇧- Shift+T	Open Type
Ctrl+⇧- Shift+H	Open Type In Hierarchy
Ctrl+⇧- Shift+R	Open Resource
Ctrl+O	Show Outline ( <b>second press shows the inherited members</b> )



### Search / Information (cont)

F2	Open class, method or variable information (tooltip text)
F3	Open Declaration: Jump to Declaration of selected class, method or parameter
F4	Open Type Hierarchy window for selected item
Ctrl+T	Show Quick Type Hierarchy ( <b>toggles between supertypes and subtypes when pressed again</b> )
Ctrl+U	Find occurrences of expression in current file
Ctrl+mouse over method	Open Declaration or Implementation
Ctrl+⇧+L	List of shortcuts (depending on the context)
Ctrl+⇧+_	Method cursor is located at the parameter Parameter specification for method reference Hints
Ctrl+⇧+G	Search workspace for references to selected method or variable
Ctrl+G	Search workspace for declarations of to the item under the cursor
Ctrl+F	Quick search
Alt+⇧	Show properties (in Package or Project Explorer)

### Cleanup

Ctrl+⇧+O	Organize Imports
Ctrl+⇧+F	Format Source
Ctrl+I	Intend selected source
Ctrl+⇧+M	Add Import / also static imports

### Refactoring

Alt+⇧+R	Rename Element
Ctrl+2, R	Rename locally (in file), faster than Alt+⇧+R
Alt+⇧+L	Extract to local variable
Alt+⇧+M	Extract to method
Alt+⇧+T	Opens the context-sensitive refactoring menu
Alt+⇧+R	Rename
Alt+⇧+V	Move
Alt+⇧+C	Change Method Signature
Alt+⇧+M	Extract Method
Alt+⇧+L	Extract Local Variable
Alt+⇧+I	Inline

### Editor Window

F12	Jump to Editor Window
Ctrl+P- gDn/PgUp-	Next / previous Editor - in the <b>order of their tabs</b> in the editor area
Ctrl+M	Maximize or un-maximize current Editor Window
Ctrl+E	Show list of open Editors. Use arrow keys and enter to switch
Alt+←/→	Backward / Forward editor navigation history
Alt+-	Open Editor Window Option menu
Ctrl+F6	Next Editor - if released it immediately, Eclipse will take you to the <b>last active editor</b>



### Editor Window (cont)

Ctrl+⇧Shi-ft+F6	Previous Editor
Ctrl+E	Quick Switch Editor
Alt+F7	<a href="#">Multipage editor</a> - next Page
Alt+⇧Shift+F7	<a href="#">Multipage editor</a> - previous Page
Ctrl+-	<b>Toogle Split Editor</b> <b>old:</b> Ctrl+_ (Horizontal) - since Luna
Ctrl+<	<b>Toogle Split Editor</b> <b>old:</b> Ctrl+{ (Vertical) - since Luna
Ctrl+⇧Shi-ft+"Num Keypad /"	Collapse all code blocks
Ctrl+⇧Shi-ft+"Num Keypad *"	Open all code blocks
⇧Shift (and hover over element)	Show source (in tooltip)    settings: Window → Preferences → Java → Editor → Hovers

### Outline View

Reorder class members with Drag & Drop	Disable "Sort Order" - otherwise not possible  Automatic reordering: Select the class, Source → Sort Members
--	--

### Package / Project Explorer

Alt+F5	Maven → Update Project...
--------	---------------------------

### Problems View

Configure to view only warnings and errors associated with a particular resource or group of resources	<a href="#">Problems View</a> - <a href="#">Configure Contents</a> e.g. select "on any element in same project" to restrict to current selected project
--	---

### Templates (Java → Editor → Templates)

test	Created test method
for	Creates for loop
new	Creates new instance
try	Creates try..catch

### Templates (Java → Editor → Templates) (cont)

sysout, syserr, systace	Creates System.out...
strf	Format string <a href="#">here</a>
if	if null/if not null
for	iterate over <b>map</b>
static_final	static final field
before	<b>junit</b> before method
after	<b>junit</b> after method
beforeclass	<b>junit</b> beforeclass method
afterclass	<b>junit</b> afterclass method
quote	surround selection with quotes <a href="#">here</a>

**Tip:** It is possible to select text and use completion (Ctrl+\_ ) including typing of template name

### Changed Eclipse Keybindings

#### Windows → Preferences → General → Keys

Required modifications due to the german QWERTZ keyboard layout:

1. Add Block Comment - **new:** Ctrl+⇧Shift+7, **old:** Ctrl+⇧Shift+/  
2. Remove Block Comment - **new:** Ctrl+⇧Shift+ß, **old:** Ctrl+⇧Shift+\
3. Toogle Split Editor (Horizontal) - **new:** Ctrl+⇧Shift+-, **old:** Ctrl+\_
4. Toogle Split Editor (Vertical) - **new:** Ctrl+⇧Shift+<, **old:** Ctrl+{

**Update 12/2016:** had to add ⇧Shift to 3. and 4. because of new [Eclipse Neon Zoom functionality](#) with Ctrl+-

### Project Specific Configuration

1. Import project **Java formatter configuration**
2. Download **Checkstyle configuration** file from sonar server and reference the config file as a file resource url
3. Download **Findbugs configuration** file from sonar server and add the config file under "include filter files"
4. Download **PMD configuration** file from sonar server and first "clear all" existing rules and then import the config file

### Plugins

Testing Support	<a href="#">moreUnit</a> <a href="#">Infinittest</a>
Test Coverage	<a href="#">EclEmma</a>
Source Code Analyzer	<a href="#">FindBugs</a> <a href="#">Checkstyle</a> <a href="#">PMD</a> <a href="#">Update-Site</a> <a href="#">Usus</a>

### Plugins (cont)

⚡ SonarCube or SonarLint  
Ignore certain warnings with //NOSONAR or @SuppressWarnings("squid:S00-112")

Refactoring JDeodorant

Subversion Subclipse

Eclipse

Team

Provider

Plugin

use Subclipse instead of Subversive because it is **much faster**

use JavaHL native/JNI provider

Task based working Mylyn

Nice: Task-focused Change Sets

Subclipse includes an optional Mylyn connector

JavaDoc JAutoDoc

Maven integration M2Eclipse (m2e)

add needed m2e connectors. follow pom.xml hints/problems. e.g:

m2e connector for build-helper-maven-plugin

m2e connector for the Eclipse JDT Compiler

m2e connector for the Maven Dependency Plugin

Groovy-Eclipse m2e integration

Decompiler Eclipse Class Decompiler

### XUbuntu specific Configuration Changes

Delete following keybindings in **XFCE Settings** → **Window Manager** → **Keyboard**:

1. Ctrl+Alt+↓/↑
2. Ctrl+F6
3. Alt+F5

If running **IBus-Applet** delete keybinding definition for "Ctrl+Space" (Content Assist) in IBus-Settings

Use **gnome-color-chooser** and change options:

1. This only works for Eclipse < 4.5, because Eclipse Mars 4.5 and higher uses **GTK3 by default (gtkrc-2.0 settings will not be used)**. You can force Eclipse 4.5 and higher to use **GTK2** with the following export or **change GTK3 theme options**: export SWT\_GTK3=0
2. Tooltip colors (background: yellow, foreground: black)
3. Global colors → Input field selected (foreground: black)

Disable "pink/purple highlights" in Xubuntu 14.10 and higher: disable custom highlight-color in **gtk-theme-config**

### Settings

show "." files in package explorer **"Package Explorer** → **View Menu** → **Filters** → **uncheck .\*\*** or "Package Explorer → Customize View... → Filters → uncheck .\*\*"

Package Explorer Option "Link with editor"

Favorites Imports Windows → Preferences → Java → Editor → Content Assist → Favorites, New Type... **x1**

Add types for Assertion library **Hamcrest** and/or **AssertJ**, **Mockito** and **JUnit-Assert**



By **Oliver Paulus (oli.p)**  
[cheatography.com/oli-p/](http://cheatography.com/oli-p/)  
[www.code-project.org](http://www.code-project.org)

Published 27th December, 2014.  
Last updated 25th January, 2017.  
Page 4 of 8.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

Settings (cont)	
Show line numbers	Window → Preferences → General → Editors → Text Editors → "Show line numbers"
Show print margin	Window → Preferences → General → Editors → Text Editors → "Show print margin"
Paste multiline strings	Window → Preferences → Java → Editor → Typing → "Escape text when pasting into a string literal"
Import code templates collection	x <sup>2</sup>
Import code template "surround with quotes"	x <sup>2</sup>
MoreUnit Configuration	Test source folder: "src/test/java"
	Test Type: "JUnit4"
	uncheck "Use test-prefix for test-methods"
	Rule for naming test files → Pattern: \${srcFile} (Test IT ITCase)
	<b>if you have legacy tests with Test prefix:</b> Rule for naming test files → Pattern: (Test)\${srcFile} (Test IT ITCase)

Settings (cont)	
Formatter Options	Window → Preferences → Java → Code Style → Formatter → Off/On Tags <b>for //@formatter:off and //@formatter:on</b>
	Check the <b>Never join lines</b> preference accessible on the Line Wrapping and the Comments tabs of the Java Formatter preference page
Improve Performance	Window → Preferences → Validation, Disable All
	Add <b>-Xverify:none</b> to eclipse.ini tell the VM not to validated all the .class files it is loading
	Window → Preferences → General → Startup and Shutdown remove plugins not needed
	increase -Xms and -Xmx heap memory options (start and max) in eclipse.ini e.g. -Xms512m -Xmx2048m
	increase -XX:MaxPermSize option (if java < 1.8) e.g. -XX:MaxPermSize=512m
Enable informative Eclipse Compiler Warnings	Window → Preferences → Java → Compiler → Errors/Warnings <b>set to Warning and keep sub checkboxes checked (if any)</b>
	<i>Potential Programming problems</i> Potential resource leak
	<i>Unnecessary code</i> Value of parameter is not used



Settings (cont)		
		Unused type parameter
		Unnecessary declaration of thrown exception
		Unnecessary 'else' statement
		Unnecessary cast of 'instanceof' operation
	<i>Generic types</i>	Redundant type arguments (1.7 and higher)
	<i>Annotations</i>	Unhandled token in '@SuppressWarnings' set to "Ignore"
	<i>Null analysis</i>	Redundant null check
Indent using spaces	Window → Preferences → XML Files → Editor	Select "Indent using spaces" and set "Indentation Size: 4"
	Window → Preferences → General → Editors → Text Editors	Set "Displayed tab width: 4" and select "Insert spaces for tabs"
Save Actions	Window → Preferences → Java → Editor → Save Actions	
	check "Perform the selected actions on save"	
	check "Format source code", "Format all lines"	
	check "Organize imports"	
	check "Additional actions" and <b>add the following:</b>	

Settings (cont)		
Code Style	<ul style="list-style-type: none"> <li>"Use blocks in if/while/for/do statements"</li> <li>Use lambda where possible <b>if you use Java &gt;= 8</b></li> </ul>	
Add final modifier to private fields*	<ul style="list-style-type: none"> <li>Add final modifier to private fields</li> <li>Add final modifier to local parameter</li> <li>Add final modifier to local variables</li> </ul>	
Unused Code	<ul style="list-style-type: none"> <li>Remove unused imports</li> <li>Add missing '@Override' annotations</li> <li>Add missing '@Override' annotations to implementations of interface methods</li> <li>Add missing '@Deprecated' annotations</li> <li>Remove unnecessary casts</li> <li>Remove unnecessary '\$NON-NLS\$' tags</li> </ul>	
Limit maximum number of open editors	Window → Preferences → General → Editors	
	check "Close editors automatically"	
	choose value (e.g. 20) for "Number of opened editors before closing"	



### Settings (cont)

JDT Core/Null Analysis to avoid NPEs	Window → Preferences → Java → Compiler → Errors/Warnings, Section <i>Null analysis</i>	if you use Java < 8 (cannot use <b>Optional</b> ) and cannot use <b>Guava Optional</b> e.g. for legacy projects
	check "Enable annotation-based null analysis"	you can use the <b>Eclipse annotations</b> , the <b>JSR-305 annotations (old)</b> , custom ones or the <b>JSR-308 annotations from Checker Framework</b>

Maven	Window → Preferences → Maven → User Interface	
	check "Open XML page in the POM editor by default"	<b>Reason:</b> show pom.xml source code is faster than loading the "Overview" tab all the time

JUnit	Window → Preferences → Java → JUnit enable "Add 'ea' to VM arguments when creating new JUnit launch configuration"	enable JVM <b>assertions</b> by default
-------	---	--

Show heap memory status	Window → Preferences → General	enable "Show heap status"
----------------------------------	-----------------------------------	---------------------------

Enable "- Quick Diff"	Window → Preferences → General → Editors → Text Editors → Quick Diff	enable checkboxes and select correct reference source option
-----------------------------	--	--

**x¹:** TDD-Style mockito: [org.mockito.Mockito](http://org.mockito.Mockito), BDD-Style mockito:  
[org.mockito.BDDMockito](http://org.mockito.BDDMockito), [org.junit.Assert](http://org.junit.Assert), [org.hamcrest.Matchers](http://org.hamcrest.Matchers),  
[org.assertj.core.api.Assertions](http://org.assertj.core.api.Assertions)  
**x²:** Java → Editor → Templates

### General

Refresh all Projects (with some closed in between)	In <b>Package Explorer</b>	Ctrl+A, F5
Use <b>Working Sets</b> for better performance	Package Explorer → open the drop-down menu → Select Working Set	
	If Ctrl+⇧Shift+R and Ctrl+⇧Shift+T does not reflect active Working Set check the small arrow on the top right of the dialogue to select Working Set	

### Troubleshooting Projects

Maven dependency issues	<b>try without eclipse project resolution</b>	[Project] → Properties → Maven
-------------------------------	---	-----------------------------------

unchecked "Resolve depend-  
encies from Workspace  
projects" to check with jars  
from local ~/.m2 repository  
instead of eclipse projects  
within workspace/worki-  
ngset

**clean projects** don't mix usage of maven  
**CLI** and **m2e eclipse plugin**

if mixed - make a **mvn  
eclipse:eclipse (CLI)**  
and import again in eclipse

**Error message The  
archive: <pa -  
th\_ to\_ jar -  
\_fi le> which is  
referenced by  
the classpath,  
does not exist.**

make a **mvn eclipse -  
eclipse:eclipse (CLI)** and import  
again in eclipse

### Troubleshooting Projects (cont)

you can use this command to search for the projects referencing  
old jar: `find -name '*.class' -exec grep -Hin  
'example-1.2.3.jar' {} \;`



By **Oliver Paulus** (oli.p)  
[cheatography.com/oli-p/](http://cheatography.com/oli-p/)  
[www.code-project.org](http://www.code-project.org)

Published 27th December, 2014.  
Last updated 25th January, 2017.  
Page 8 of 8.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>