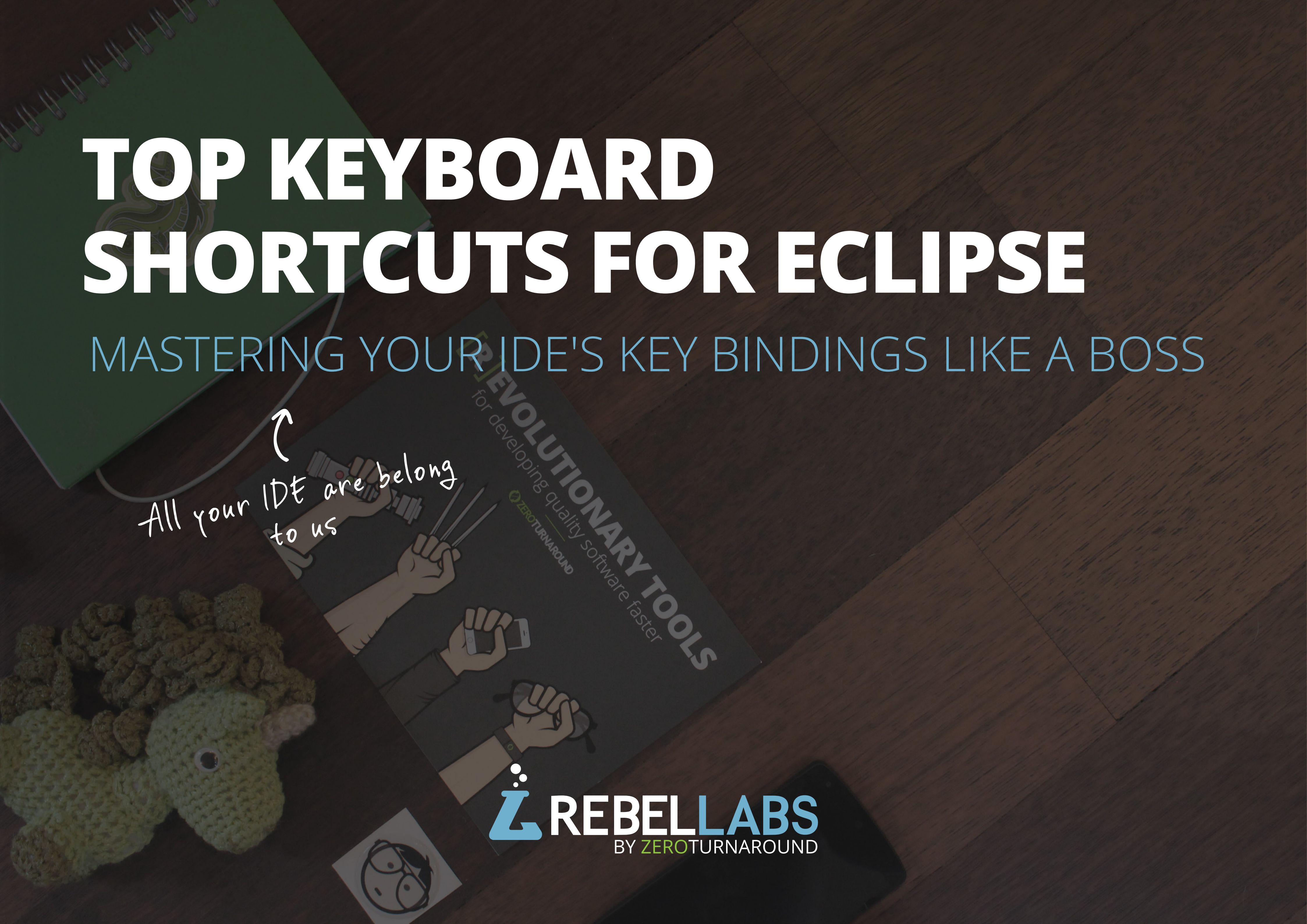
TOP KEYBOARD SHORTCUTS FOR ECLIPSE

MASTERING YOUR IDE'S KEY BINDINGS LIKE A BOSS



All your IDE are belong to us

REBELLABS
BY ZEROTURNAROUND

TABLE OF CONTENTS

INTRODUCTION TO YOUR NEXT GENERATION TEXT EDITOR

> “*Tools of the trade can be a source of frustration or a source of enjoyment. Mastery of your tools doesn't just make you more productive, it makes the whole process more fun. Those who think otherwise have never used vi. ZZ*”

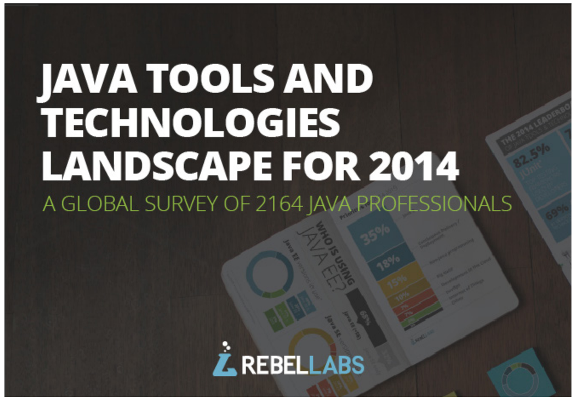
JEVGENI KABANOV,

CEO of ZeroTurnaround, makers of JRebel

Brief Overview of the IDE landscape

If you're just joining the conversation, then you probably know quite well that IDE stands for **I**ntegrated **D**evelopment **E**nvironment, and is most likely one the first tools you learned how to use (unless vi was more your style). IDEs emerged in order to give developers working on more complex applications more of a feature-rich experience. At least, more than a glorious notepad capable of opening multiple documents at the same time.

As you can see in the image from Java Tools and Technologies Landscape 2014, in Java we have about a dozen (or maybe more) options for developers to give their gorgeous code life: Eclipse, IntelliJ IDEA, NetBeans, Spring Tool Suite, IBM RAD, MyEclipse, JBoss Dev Studio and Codenvy (formerly "eXo IDE") are some products designed to fulfill our IDE requirements, but predecessors to what many consider "modern IDEs" -- such as vi/vim, Emacs and Notepad++ -- are still in use by small segments of hardcore fans. If you are using Eclipse, you are in good company.

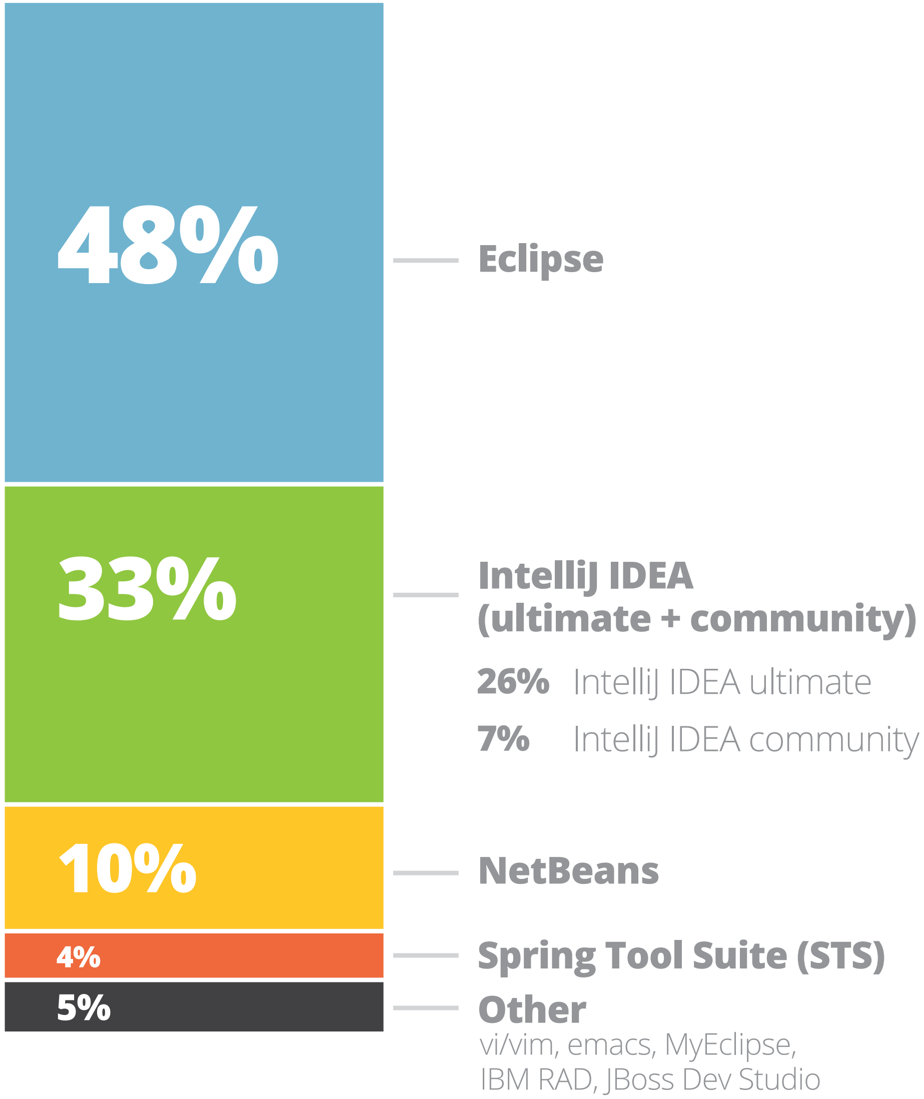


JAVA TOOLS AND TECHNOLOGIES LANDSCAPE FOR 2014

A GLOBAL SURVEY OF 2164 JAVA PROFESSIONALS

REBELLABS

IDE used most often



According to our 2014 survey, it is by far the most popular IDE for Java developers.

These days, IDEs are more or less ubiquitous, with 97% of developers surveyed out there using one according to recent surveys by RebelLabs.

In modern IDEs, opening multiple projects, figuring out dependencies between them, integrating the build systems, navigating type systems, and even the plain automatic compilation on save is something so common in the Java world that we don't even consider these actions as features. They are must-haves that are basically non-negotiable. Indeed, many developers don't even consider using a programming language for any serious work if the IDE support for it is lacking.

This report is ultimately a quick guide for learning the shortcuts and, to an extent, a bit more about the features, of Eclipse. Our goal here is to help you learn your current IDE better, so that you can be more fluent or learn new features.

48% of developers
choose Eclipse as their IDE

eclipse

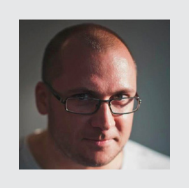
Latest version: **4.4.1 (as of Oct 2014)**
First introduced: **Nov 2001**
Release frequency: **Annual**

The Eclipse IDE, with its alternative implementations like MyEclipse, Spring Tool Suite, JBoss Dev Studio and IBM RAD, has been the market leader for some years—at least since 2011, when we first asked.

Standing at **48%** share of the developer market, Eclipse is a widely-used platform with a massive library of plugins and extensions via the Eclipse Marketplace to support nearly any programming environment.

Eclipse is one of a dozen of high-level projects maintained by the Eclipse Foundation, a non-profit organization supported by corporate sponsors and organizations in addition to the development community.

Eclipse is the **best** RCP platform out there, very well documented and well-designed. The ability to run non-compilable code is cool and I also like the shortcut assist Shift+Cmd/Ctrl+L. But I think that Ctrl+3 in Eclipse is the best feature in IDEs evah! It works as you expect it to work, unlike what you get with other IDEs.



ANTON ARHIPOV
Product Manager at ZeroTurnaround

Eclipse has a Maven dependency hierarchy viewer which is great for troubleshooting why wrong versions of some libraries are sometimes used in your project.

ARNEL PÄLLO
Engineer at ZeroTurnaround

MOST POPULAR PLUGINS AND USEFUL KEYBOARD SHORTCUTS

Personalizing your IDE is an important and time consuming action, which boils down to considering which plugins to install or remove, and updating the keymap to reflect your habits. So let's get productive!

A bit about plugins

An IDE is one of the most personal items in your developer toolbelt. Many of us have a thoroughly honed Eclipse installation folder with a history that goes back several years.

Given the worldwide popularity of the Java ecosystem, most of us have different ideas, preferences and requirements for what an IDE should do (and know how to do). It comes as no surprise that all that functionality cannot be pre-built into the IDE from the beginning and for everybody. Luckily, software gurus figured out the problem of extensibility a long time ago, and the current solution is to introduce new functionality via plugins.

A well-maintained plugin ecosystem is what makes an IDE a true platform and allows it to conquer all your possible scenarios without becoming a huge slow monster. Many productivity gains also come from the plugins directly, such as <SHAMEFUL PLUG>JRebel, a developer productivity-enhancing plugin that lets you view code changes instantly without having to restart or redeploy (now supporting changes to superclasses and implemented interfaces in JRebel 6) </SHAMEFUL PLUG>

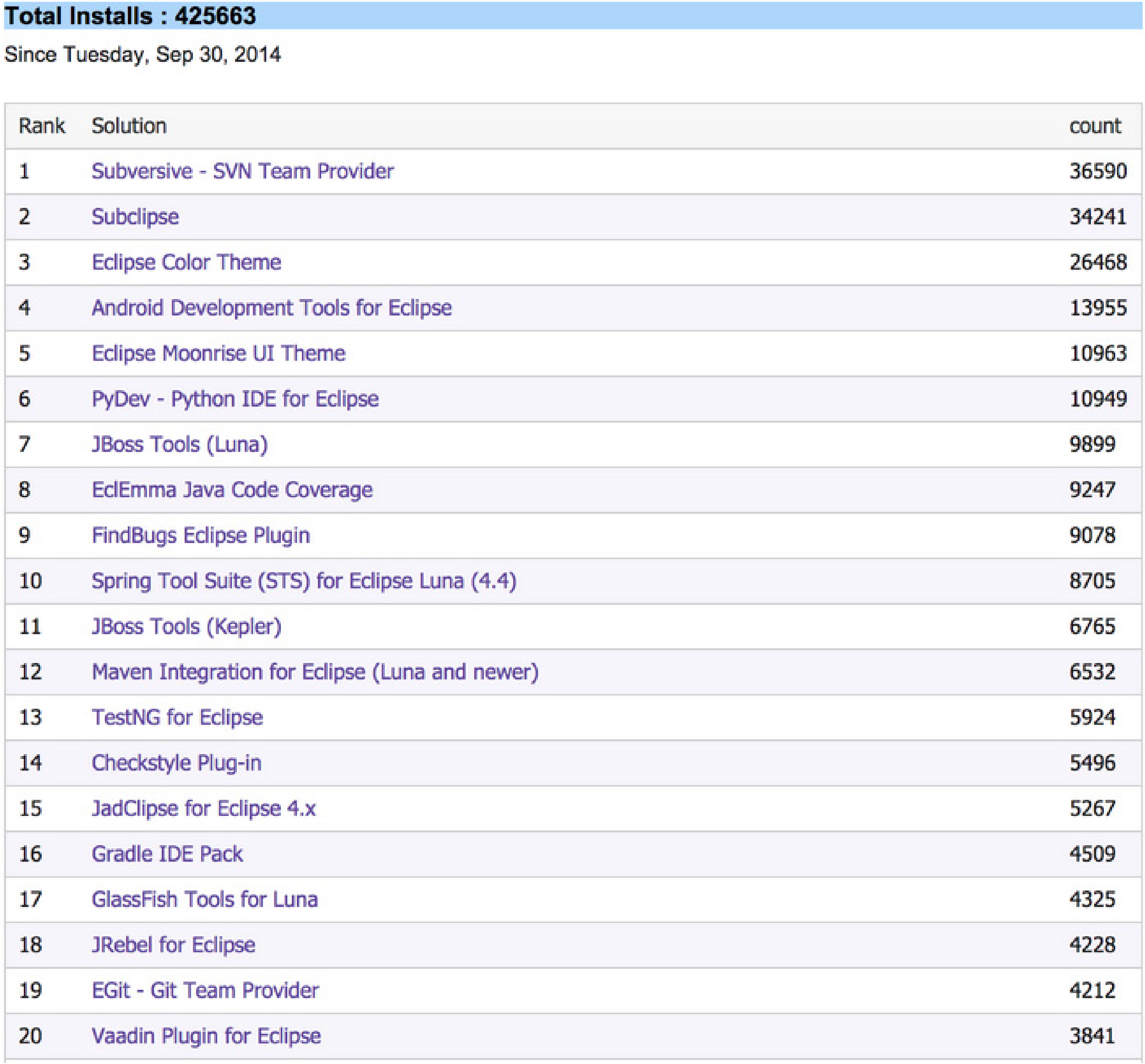
In this next part, we review some of the top rated and top downloaded plugins for Eclipse. We'll look at each of the most installed plugins for Java and JVM development, and see what plugins other Eclipse users install that may be valuable to you as well.

The Eclipse Marketplace

Eclipse is much more of a platform rather than just an IDE. The default place for all plugin distribution is the Eclipse Marketplace where around 1830 plugins are available right now. You can install a standalone plugin just by copying its jar into the correct directory, but the amount of plugins distributed this way is probably negligible compared to the standard distribution mechanism of the marketplace.

The marketplace contains two top lists, one that counts **total installations** over time, and the other the number of **users favorites** among the plugins.

The top installed plugins gives the statistics for the last month (which we looked at) and for all time, but you can see for yourself that not very much changes between these lists. Eclipse is indeed a very mature and stable platform. Also, note that for other IDEs we look at the top 10 and not top 20 like here. However, given the abundance of SVN and Maven plugins in the Eclipse top, there only so many interesting entries in the top 20 downloads. SPOILER ALERT: you'll be unhappy if you're looking for something novel like 2 plugins for Maven, 3 for Spring, 2 for Subversion and 2 color themes...



Total Installs : 425663

Since Tuesday, Sep 30, 2014

Rank	Solution	count
1	Subversive - SVN Team Provider	36590
2	Subclipse	34241
3	Eclipse Color Theme	26468
4	Android Development Tools for Eclipse	13955
5	Eclipse Moonrise UI Theme	10963
6	PyDev - Python IDE for Eclipse	10949
7	JBoss Tools (Luna)	9899
8	EclEmma Java Code Coverage	9247
9	FindBugs Eclipse Plugin	9078
10	Spring Tool Suite (STS) for Eclipse Luna (4.4)	8705
11	JBoss Tools (Kepler)	6765
12	Maven Integration for Eclipse (Luna and newer)	6532
13	TestNG for Eclipse	5924
14	Checkstyle Plug-in	5496
15	JadClipse for Eclipse 4.x	5267
16	Gradle IDE Pack	4509
17	GlassFish Tools for Luna	4325
18	JRebel for Eclipse	4228
19	EGit - Git Team Provider	4212
20	Vaadin Plugin for Eclipse	3841

You can see that all the plugins that were downloaded the most can be easily divided into several groups by functionality. Some of these groups are pretty basic, like UI color themes. Maybe it means that the average Eclipse user would prefer to pre-bundle some of these enhancements before downloading?

Subversive / Subclipse - SVN plugins, for those who haven't migrated to git or mercurial yet.

EGit - Git has won, no wonder the plugin is popular.

Eclipse color theme / Moonlight UI - woohoo, we all like things that look pretty, don't we?

Maven integration - Maven is used by 64% of Java developers, so perhaps it could be added into the bundle?

Gradle IDE pack - Gradle might very well rule the world eventually. Nice to know it gets traction and the tooling catches up.

Android development tools - Eclipse is still the official IDE for Android development.

PyDev - Python is flexible, dynamic and installed everywhere by default.

Spring Tool Suite (STS) - Spring Framework is an umbrella project for tons of useful libraries and making your IDE aware of them is a smart step -- you'll notice that STS is also in use by 4% of the survey respondents from the introduction, so it's not easy to ignore.

Vaadin framework - Vaadin is an interesting web framework with pure Java components, beautiful widgets, flexibility.

JBoss Tools (both Luna and Kepler) - umbrella project to work with all things Red Hat, including JBoss, which is considered by some to be the best Java Application Server there is.

GlassFish Tools for Luna - Oracle has cut commercial support of the GlassFish, but it still is the Reference Implementation of Java EE server.

EclEmma - a very well-known code coverage tool for Java.

FindBugs - a very popular open source, static code analysis tool.

TestNG - JUnit is certainly used more than TestNG, but it doesn't mean that other testing frameworks cannot top it in terms of quality, usability or features.

CheckStyle - code quality analysis tool focused on the looks of code. Make your team comply with a chosen code standard and enjoy more readable diffs.

JadClipse - a well-known Java Bytecode decompiler.

JRebel - a developer productivity tool, allows you to view code changes instantly, which enables developers to get more done in the same amount of time. Become 17% more productive immediately. More effective than a double espresso in the morning.

Keyboard shortcuts and productivity with IDEs

Okay, here we get to the juicy bits. Our graphic designer Ladi has spent a lot of time trying to map out keyboard shortcut preferences in a visually interesting way. Hopefully this is insightful!

Please note that these are the default keyboard shortcuts for Eclipse, but you can easily change and remap key mappings for all imaginable (and even some unimaginable) actions. But, we do think the defaults might hint that some use-cases are more important to the Eclipse development team.

SEARCH

Searching for text is the most useful basic feature that an editor can provide. However, given that the code has structure and dependencies, more advanced lookups are possible. Find usages is pretty useful, it makes refactoring a lot easier. Awesome!

eclipse

ACTION	WINDOWS	OS X
Find usages	Ctrl + Shift + G	⌘ + Shift + G
Find / Replace in file	Ctrl + F	⌘ + F
Find / Replace in projects	Ctrl + H	Ctrl + H
Find next	Ctrl + J	⌘ + K

eclipse

ACTION	WINDOWS	OS X
Open resource / Navigate to file	Ctrl + Shift + R	⌘ + Shift + R
Open type	Ctrl + Shift + T	⌘ + Shift + T
Go to line	Ctrl + L	⌘ + L
Tab / File switcher	Ctrl + E	⌘ + E

FILE NAVIGATION

File navigation shortcuts are highly-used key bindings in the IDE. Open Java class, open resources like .xml, properties files or your favorite *gradle.build* file. Eclipse's open resource and open type actions are super useful and will help you find class definitions very quickly. Eclipse also has the **Cmd / Ctrl + O** shortcut for opening a quick outline that allows you to find a method in the current class.

WINDOWS ACTIONS

Window actions are about as basic as it gets, but useful to know how to easily navigate between open editors back and forth, open recent files, etc.

ACTION	WINDOWS	OS X
Maximize active window	Ctrl + M	⌘ + M
Next view (editor)	Ctrl + F6	⌘ + F6
Quick switch editor	Ctrl + E	⌘ + E
Back	Alt + Left	Alt + Left
Forward	Alt + Right	Alt + Right
Activate editor	F12	F12

TEXT EDITING ACTIONS

Using shortcuts to edit your code, such as how you move and delete lines, format indentation and so on, are generally part of a strong developer’s requirements. Generally the shortcuts should be familiar to you, even if you're fresh to Eclipse as they're standard amongst most text editors as well.

ACTION	WINDOWS	OS X
Move lines	Alt + Up / Down	Alt + Up / Down
Delete lines	Ctrl + D	⌘ + D
Copy / Duplicate lines	Alt + Shift + Up / Down	⌘ + Alt + Up / Down
Select identifier	Alt + Shift + Up	Alt + Shift + Up
Format code	Ctrl + Shift + F	⌘ + Shift + F
Correct indentation	Ctrl + I	⌘ + I
Structured selection	Alt + Shift + Up	Alt + Shift + Up

CODE NAVIGATION

Code navigation is similar to file navigation, has it's own structure, independent of file directories. Eclipse recognizes that code navigation is very important, so they make it easy.

eclipse

ACTION	WINDOWS	OS X
Find usages / References in workspace	Ctrl + Shift + G	⌘ + Shift + G
Quick outline / File structure	Ctrl + O	⌘ + O
Inspect code hierachy	Ctrl + T	⌘ + T
Open / Navigate to declaration	F3	F3
Open / Navigate to type hierarchy	F4	F4

eclipse

ACTION	WINDOWS	OS X
Show quick refactoring menu	Ctrl + Alt + T	⌘ + Shift + T
Rename	Ctrl + Alt + R	⌘ + Alt + R
Surround with	Ctrl + Alt + Z	⌘ + Alt + Z
Extract local variable	Ctrl + Alt + L	⌘ + Alt + L
Inline	Ctrl + Alt + I	⌘ + Alt + I
Extract method	Ctrl + Alt + M	⌘ + Alt + M

REFACTORING

Refactoring shortcuts immensely help you when making changes to legacy code, helping you extract pieces of code into different entities (variables, fields). Using this actually serves as a basis for writing the code in a right-to-left manner, without declaring the variables, but specifying their contents first and automagically extracting them into variables later.

CODE COMPLETION

Eclipse has three types of code completion. First there's quick fix which helps you fix immediate problems that Eclipse has identified, "normal" completion, and a really robust macro system. Try typing "sysout" for an example of that.

eclipse

ACTION	WINDOWS	OS X
Quick fix	Ctrl + 1	⌘ + 1
Code completion	Ctrl + Space	Ctrl + Space

UNIVERSAL ACCESS

Eclipse has an awesome little keybindings cheatsheet included, it really helps

eclipse

ACTION	WINDOWS	OS X
Shortcuts cheatsheet	Ctrl + Shift + L	⌘ + Shift + L
Quick access / search everywhere	Ctrl + 3	⌘ + 3

SEARCH

ACTION	WINDOWS	OS X
Find usages	Ctrl + Shift + G	⌘ + Shift + G
Find / Replace in file	Ctrl + F	⌘ + F
Find / Replace in projects	Ctrl + H	Ctrl + H
Find next	Ctrl + J	⌘ + K

WINDOWS ACTIONS

ACTION	WINDOWS	OS X
Maximize active window	Ctrl + M	⌘ + M
Next view (editor)	Ctrl + F6	⌘ + F6
Quick switch editor	Ctrl + E	⌘ + E
Back	Alt + Left	Alt + Left
Forward	Alt + Right	Alt + Right
Activate editor	F12	F12

FILE NAVIGATION

ACTION	WINDOWS	OS X
Open resource / Navigate to file	Ctrl + Shift + R	⌘ + Shift + R
Open type	Ctrl + Shift + T	⌘ + Shift + T
Go to line	Ctrl + L	⌘ + L
Tab / File switcher	Ctrl + E	⌘ + E

TEXT EDITING ACTIONS

ACTION	WINDOWS	OS X
Move lines	Alt + Up / Down	Alt + Up / Down
Delete lines	Ctrl + D	⌘ + D
Copy / Duplicate lines	Alt + Shift + Up / Down	⌘ + Alt + Up / Down
Select identifier	Alt + Shift + Up	Alt + Shift + Up
Format code	Ctrl + Shift + F	⌘ + Shift + F
Correct indentation	Ctrl + I	⌘ + I
Structured selection	Alt + Shift + Up	Alt + Shift + Up

CODE NAVIGATION

ACTION	WINDOWS	OS X
Find usages / References in workspace	Ctrl + Shift + G	⌘ + Shift + G
Quick outline / File structure	Ctrl + O	⌘ + O
Inspect code hierachy	Ctrl + T	⌘ + T
Open / Navigate to declaration	F3	F3
Open / Navigate to type hierarchy	F4	F4

REFACTORING

ACTION	WINDOWS	OS X
Show quick refactoring menu	Ctrl + Alt + T	⌘ + Shift + T
Rename	Ctrl + Alt + R=	⌘ + Alt + R
Surround with	Ctrl + Alt + Z	⌘ + Alt + Z
Extract local variable	Ctrl + Alt + L	⌘ + Alt + L
Inline	Ctrl + Alt + I	⌘ + Alt + I
Extract method	Ctrl + Alt + M	⌘ + Alt + M

CODE COMPLETION

ACTION	WINDOWS	OS X
Quick fix	Ctrl + 1	⌘ + 1
Code completion	Ctrl + Space	Ctrl + Space

UNIVERSAL ACCESS

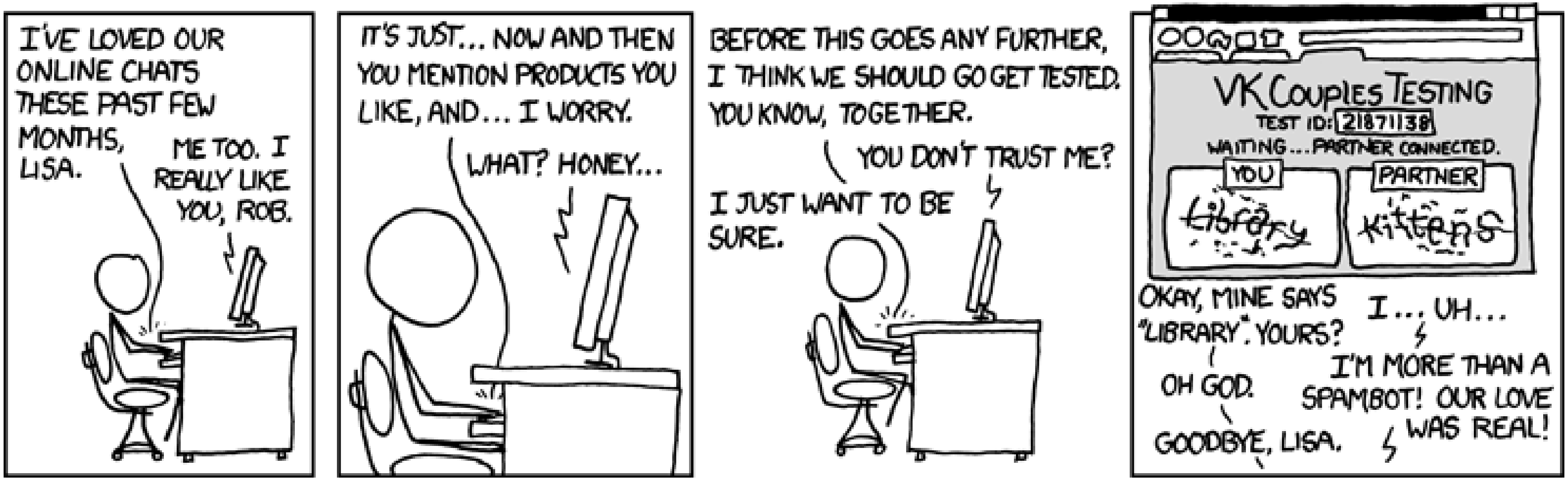
ACTION	WINDOWS	OS X
Shortcuts cheatsheet	Ctrl + Shift + L	⌘ + Shift + L
Quick access / search everywhere	Ctrl + 3	⌘ + 3

Goodbye Comic and Best Wishes

There's nothing better than getting a truly genuine experience with your most intimate of development tools, your trusted IDE. Whether you use Eclipse, Intellij IDEA, or NetBeans, coding while in the flow with your IDE is awesome, and keyboard shortcuts help keep you there. We hope this report has helped shine a little light how to make you Eclipse experience considerably more enjoyable and productive.

As we stated earlier, we want this report to be an accessible guide for learning the keybindings and cool shortcuts for Eclipse, the most popular IDE in the market. Hopefully, you've learned a bit more about it and you're well on your way to being more fluent.

So that's it ... want more? Too bad -- there are tons of resources out there ... which we can point you to if you send your search term to @RebelLabs via Twitter ;-)


I'VE LOVED OUR ONLINE CHATS THESE PAST FEW MONTHS, LISA.
ME TOO. I REALLY LIKE YOU, ROB.
IT'S JUST... NOW AND THEN YOU MENTION PRODUCTS YOU LIKE, AND... I WORRY.
WHAT? HONEY...
BEFORE THIS GOES ANY FURTHER, I THINK WE SHOULD GO GET TESTED. YOU KNOW, TOGETHER.
YOU DON'T TRUST ME?
I JUST WANT TO BE SURE.
VK COUPLES TESTING
TEST ID: 21871138
WAITING... PARTNER CONNECTED.
YOU
PARTNER
Library
kittens
OKAY, MINE SAYS "LIBRARY". YOURS?
I... UH...
OH GOD.
I'M MORE THAN A SPAMBOT! OUR LOVE WAS REAL!
GOODBYE, LISA.

http://xkcd.com/632/

Contact Us

Twitter: @RebelLabs

Web: http://zeroturnaround.com/rebellabs

Email: labs@zeroturnaround.com

Estonia
Ülikooli 2, 4th floor
Tartu, Estonia, 51003
Phone: +372 653 6099

USA
399 Boylston Street,
Suite 300, Boston,
MA, USA, 02116
Phone: 1(857)277-1199

Czech Republic
Jankovcova 1037/49
Building C, 5th floor,
170 00 Prague 7, Czech Republic
Phone:+420 227 020 130

Report Author:
Oliver White (@theotown), Oleg Shelajev (@shelajev)

Report Designer:
Ladislava Bohacova (@Ladislava)