

TOP JAVA IDE KEYBOARD SHORTCUTS FOR ECLIPSE, INTELLIJ IDEA & NETBEANS

MASTERING YOUR IDE'S KEY BINDINGS LIKE A BOSS

↑
All your IDE are belong to us



TABLE OF CONTENTS

INTRODUCTION to your next generation Text Editor	1
BRIEF OVERVIEW of the IDE landscape	2-9
MOST POPULAR PLUGINS and useful keyboard shortcuts	10-26
IDE SHORTCUT Cheatsheets	27-33
GOODBYE COMIC and Best Wishes	34

View code changes
instantly



THIS REPORT IS SPONSORED BY:

JRebel

INTRODUCTION

TO YOUR NEXT GENERATION TEXT EDITOR

“

We believe that smart tools enable creative work with any project, no matter how big or small it is, be it a fresh startup or something that's been in development for a long time.

”

MAX SHAFIROV,
CEO of JetBrains, makers of IntelliJ IDEA

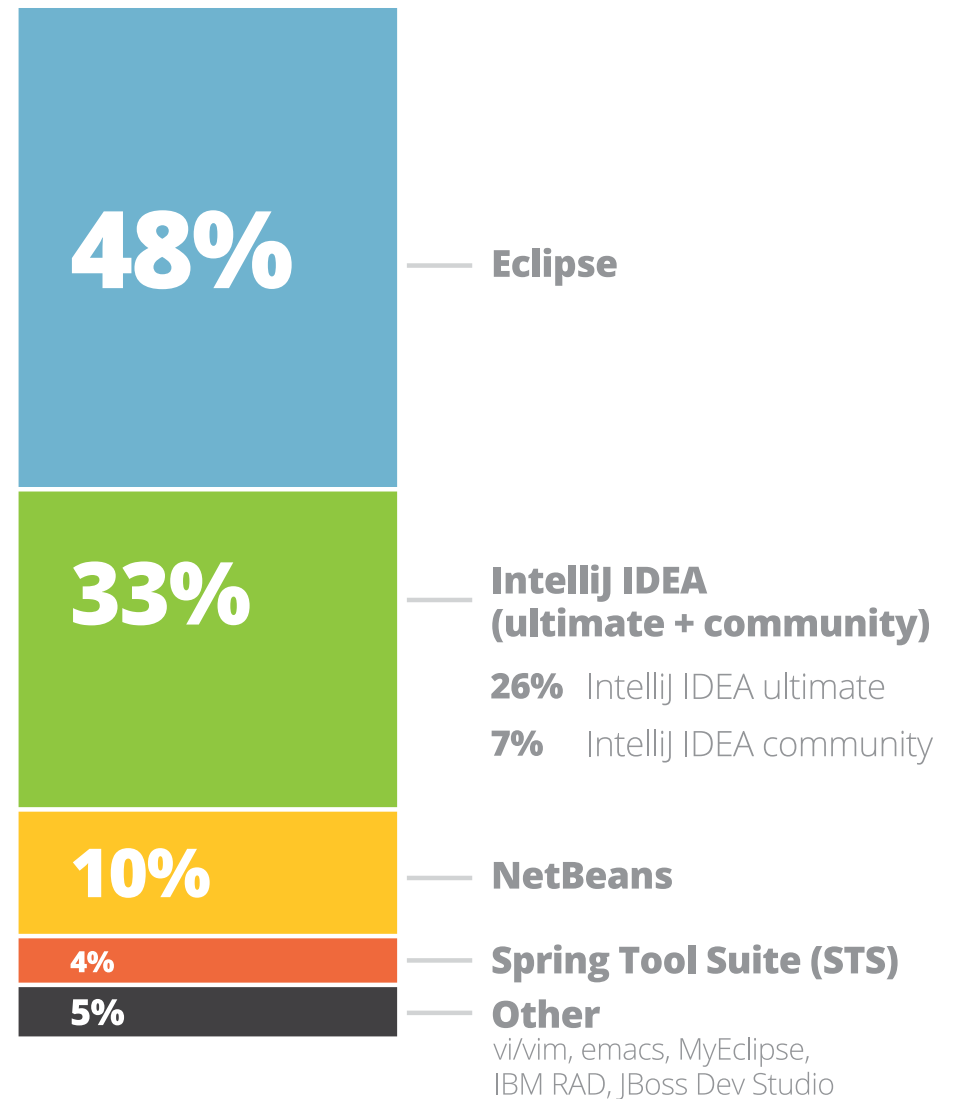
Brief Overview of the IDE landscape

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.



IDE used most often



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.

Considering that over 90% of Java devs out there are using Eclipse, IntelliJ IDEA or NetBeans, and Spring Tool Suite is an implementation of Eclipse with mostly the same shortcuts, we've decided to focus on these three IDEs.

This report is ultimately a quick guide for learning the shortcuts and, to an extent, a bit more about the features, of Eclipse, IntelliJ IDEA and NetBeans. Our goal here is to help you learn your current IDE better, so that you can be more fluent or learn new features. We also hope you'll get a decent snapshot of other IDEs and how things are done elsewhere, in case you decide to ever switch some day.

So, let's find out a little bit more about each of them, starting with our market leader, Eclipse...





Latest version: **4.4.1 (as of Oct 2014)**

First introduced: **Nov 2001**

Release frequency: **Annual**

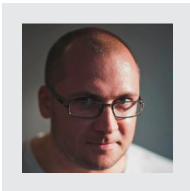
The Eclipse IDE, with its alternative implementations like MyEclipse, Spring Tool Suite, JBoss Dev Studio and IBM RAD, has been the market leader for some years—at least since 2011, when we first asked.

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.

“

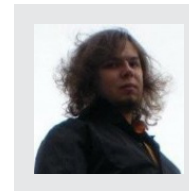
Eclipse is the **best** RCP platform out there, very well documented and well-designed. The ability to run non-compilable code is cool and I also like the shortcut assist Shift+Cmd/Ctrl+L. But I think that Ctrl+3 in Eclipse is the best feature in IDEs evah! It works as you expect it to work, unlike what 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. Too bad IntelliJ does not have it.



ARNEL PÄLLO

Engineer at ZeroTurnaround

”



Latest version: **14 (as of Nov 2014)**

First introduced: **Jan 2001**

Release frequency: **Annual**

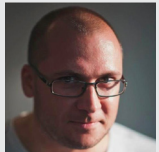
IntelliJ IDEA is the second most popular IDE in the Java development market, with a combined 33% share (26% Ultimate Edition and 7% Community Edition) according to the [Java Tools and Technologies Landscape for 2014](#). In addition, IntelliJ IDEA was chosen by 49% of developers as the "IDE they would rather use", making it the most popular choice of IDE for developers.

IntelliJ IDEA was first released in 2001 and has continued to making annual major releases around December, with minor updates throughout the year.

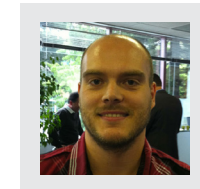
Although JetBrains is best known for their flagship and first product, they provide a family of 7 IDEs in total, plus another 10 proprietary tools and technologies for development.

“

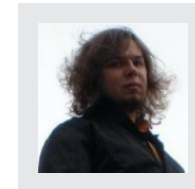
Autocompletion in IntelliJ is awesome, especially the smart auto-completion. However, I'd love to see it simplified so that novice users could enjoy those features too, without going through the learning curve. One real differentiator is the “productivity guide” feature, which helps you understand just how efficient you are with IntelliJ, and how can you improve to become even more productive. Oh, I like not having to press Ctrl/Cmd+S in IntelliJ too!



ANTON ARHIPOV
Product Manager
at ZeroTurnaround



MICHAEL RASMUSSEN
Engineer at
ZeroTurnaround



ARNEL PÄLLO
Engineer at
ZeroTurnaround

My top features include postfix-completion, the ability to write an expression, and then for instance add `.for<tab>` so that it automatically completes that expression into an enhanced for-loop. Also, the usage-dropdown, the ability to have the cursor on basically any member or identifier, press the hotkey, and a drop down appears with all the usages of that item throughout the project, to quickly look over and/or select from.

I quickly grew fond of IntelliJ's one project-per-window way of things (instead of having a hundred projects in a single workspace). This is especially nice for large, multi-module Maven projects. Also, IntelliJ's File Watcher plugin is a rockstar for me and makes using preprocessors (such as typescript to javascript or SASS to CSS) a breeze.

”



NetBeans

Latest version: **8.0.1 (as of Oct 2014)**

First introduced: **1997**

Release frequency: **Annual**

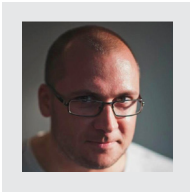
NetBeans continues to be a minority player in the IDE market, even as IntelliJ IDEA grows to chip away at Eclipse's dominance.

At 10% developer use, NetBeans holds the third position in the market. NetBeans is one of the first modern IDEs on the scene, coming out of a 1996 program by Prague's Charles University before being acquired by Sun Microsystems in 1999.

Under Oracle's stewardship of Java, NetBeans has benefitted from the closeness to the helm, and was the first Java IDE to become Java SE 8 compliant, followed closely by IntelliJ IDEA.

“

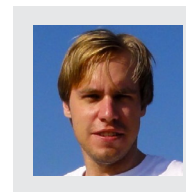
I think Maven integration in NetBeans is the most practical one among all the Java IDEs, and I absolutely love the "Services" view in NetBeans. It is simple and easy to access - what else would you need?



ANTON ARHIPOV

Product Manager at ZeroTurnaround

I like the simplicity of the NetBeans configuration the most. You don't have to search through a jungle of options to do what you want and then it just works needed. Oh, and occasionally I can even hack the PHP with the same excellent support!



ANTON PELESHEV

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. Or a clean and solid NetBeans that doesn't have any performance issues because you know what plugins you have installed there. Or an IntelliJ IDEA settings file that is accurately backed up every week.

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 (note that there is a difference!) for Eclipse, IntelliJ IDEA and NetBeans. For each of them we look at the most installed plugins for Java and JVM development, but we try to make sense of it and take into account differences between them, like the fact that JetBrains has different products for languages like Python, Ruby, PHP etc, whereas Eclipse and NetBeans do not.



Eclipse

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! Who installs it anyway?

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.

IntelliJ IDEA

IntelliJ IDEA is often regarded as an IDE with batteries included. Most of the functionality that you have to get with plugins in Eclipse is bundled in IntelliJ IDEA by default. Naturally, we're examining the Ultimate Edition (commercial) of the IDE because frankly it doesn't make sense to compare the Community Edition (free but with less features) version of one tool to the mainstream branch of the others. Don't get us wrong: IntelliJ IDEA Community Edition is a great IDE, but given that Ultimate is as easily obtainable at least for a trial, it's a no-brainer.

There are plugins that can enhance your IntelliJ experience and they reside in a [plugin repository](#) curated by JetBrains. Note that IntelliJ IDEA is the only IDE that we look at in this report that has counterparts for other languages. For example, a Python IDE would be the [PyCharm](#). This suggests that we might not find Python plugin as popular as for Eclipse.

Without further ado, here are the top plugins in the repository.

10 Top Downloads

1. [Scala](#) (2278989)
2. [Markdown](#) (1035994)
3. [Symfony2 Plugin](#) (911600)
4. [JRebel Plugin](#) (893228)
5. [BashSupport](#) (716212)
6. [IdeaVim](#) (654144)
7. [NodeJS](#) (568619)
8. [.gitignore support](#) (491378)
9. [Eclipse Code Formatter](#) (435729)
10. [Database Navigator](#) (422184)

Scala - Scala is the most desired Object Oriented and Functional JVM language, and very popular among IntelliJ IDEA users.

Markdown - a text-to-HTML conversion language with a lightweight syntax. Perfect for documentation, blogging and casual writing for the web.

Symfony2 Plugin - how did this PHP framework plugin get the most popular download in the Java JVM?

JRebel - ok, yeah it's here again -- what can we do? ;-) JRebel is better than a bacon-wrapped Ferrari that eliminates unnecessary redeploys and let's you view code changes instantly. I would really like if these edits made it into the report.

BashSupport - shell scripts are an essential part of our lives. The most used deployment tool and an incomparable automation solution for everything.

IdeaVim - vi/vim bindings for IntelliJ IDEA.

NodeJS - with server-side JavaScript, you can forget the "party in the front, business in the back" motto and have a party everywhere in your code.

.gitignore support - do you remember from the top of the head how to ignore all .class files in your git repo? Syntax highlighting, inspections and completion support can help you.

Eclipse Code Formatter - IntelliJ IDEA acknowledges that Eclipse is the most-used IDE for Java, making it only natural that your teammates want to have a single code formatting strategy.

Database Navigator - helps to manage database connections, execute SQLs and debug your database objects right from your IDE.

You can see that there are less VCS, language and framework plugins here, which supports the idea that the "batteries-included" approach allows you to focus on small user experience improvements that make your IDE extremely personal instead of just customizable. Additionally, note that JetBrains accepts feature requests for plugins you want and you can vote on them to make their implementation a reality faster.

NetBeans

NetBeans plugins are available for submission and download from the [NetBeans plugin portal](#), where you can find about 850 useful plugins hanging out. Honestly, NetBeans is the most mysterious IDE out of the big three that we looked at in this report, so we cannot predict in advance which plugins will make to the top here.

Our experience with the modern releases of NetBeans suggests that things like build tools integrations, application servers support and VCS tooling are extremely solid and have smooth user experience in an out-of-the-box installation of NetBeans. So we guess plugins for that functionality are not strictly necessary. Let's see the NetBeans plugin portal top.

Show 20 entries			
Plugin name	Category	NetBeans versions	Description
➡ iReport	Databases, Enterprise Applications, Editing	7.4, 7.2, 7.1, 7.0, 6.9, 6.8, 6.7, 6.5, 6.1, 6.0	iReport designer for JasperReports
➡ NetBeans OpenGL Pack		6.7, 6.0	The NetBeans OpenGL Pack provides an easy to use OpenGL development environment integrated into NetBeans.
➡ Ruby and Rails	Programming Languages, Web Applications, Editing	8.0, 7.4, 7.3, 7.0, 6.9	Ruby for Netbeans provides all the basics expected from a full-featured Ruby IDE. Ruby for Netbeans also provides support for the Ruby on Rails web framework.
➡ Sublime Theme	Web Applications, Version Control Systems, Utilities	8.0, 7.3	Sublime Text default theme for Netbeans IDE.
➡ NBAndroid	Mobile Applications	8.0, 7.4, 7.3, 7.2	Interested in developing Android applications in the NetBeans IDE? This plugin provides support for Android applications development cycle.
➡ NetBeans UML		6.0	With NetBeans UML modeling, your designers focus on the design of the application, and your developers focus on the code. First, designers and analysts design applications using UML, a standard modeling language, then developers generate source code from the UML model. Of course it works both ways: You...
➡ Codename One	Mobile Applications, Graphical User Interface, Fun Stuff	8.0, 7.4, 7.3, 7.2, 7.1, 7.0	Develop true native mobile applications in Java for iOS, Android, Windows Phone 7, Blackberry etc.
➡ Struts2 Support Repack for NB 7.x + XWork	Web Applications, Enterprise Applications, Libraries	7.4, 7.3, 7.2, 7.1, 7.0	Repack of "Struts2 Support" plug-in available for NetBeans version 6.9. Updated plug-in runs on Netbeans versions 7.x. With new core Struts 2 library wrapper module version 2.3.15.3. Added experimental support for XWork validation features.
➡ JDateChooser 1.2	Graphical User Interface, Desktop Applications, Libraries	8.0, 7.3, 7.0, 6.0	JDatePicker. JavaBean components(calendar panel, combobox, dialog). Built-in visual appearance editor, external skins support. Single, period, multiple or null selection. Table cell editor & renderer.

iReport - a visual report designer for JasperReports

NetBeans OpenGL Pack - an easy to use OpenGL development environment

Ruby and Rails - integration supporting the Ruby IDE and Rails framework

Sublime theme - how can we make a top without any UI theme customizations?

NBAndroid - support for Android application development

NetBeans UML - UML modelling tools for professionals and amateurs alike

Codename One - support for developing mobile applications: iOS, Android, Windows Phone 7, Blackberry

Struts 2 support - framework support for Struts 2 (still in use by 6% of devs out there). Pretty straightforward, huh?

JDateChooser - a visual editor and renderer for JavaBean components, like calendar panels, dialogs and comboboxes.

Pretty all over the place, isn't? This supports the claim that NetBeans provides an excellent development environment from the start and people use different plugins according to their needs.

Keyboard shortcuts and productivity with IDEs

Ok, here we get to the juicy bits. Ask Ladi, our graphic designer, and she'll confirm that it's an ungrateful business to map out and compare keyboard shortcut preferences. But we're probably creative enough to do this in a manner that offers some insight rather than sparks a flamewar. We decided to look at a bigger picture across the top 3 Java IDEs and separate the most used shortcuts into sections they belong to. Here are the 8 sections we discerned:

SEARCH

FILE NAVIGATION

WINDOWS ACTIONS

TEXT EDITING

CODE NAVIGATION

REFACTORING

CODE COMPLETION

UNIVERSAL ACCESS

Note that we do not attempt to drive any conclusions about the productivity of an IDE based on the default keyboard shortcuts map, because you can easily change that and set key mappings for all imaginable (and even some unimaginable) actions. But we do think that it might hint that some use-cases are more important to the IDE development teams than others. Let's find out.



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. All three IDEs give you plenty of options for locating where a given Java symbol is referenced or declared. IntelliJ IDEA also offers a **Shift-Shift** combo for the unique “search everywhere” feature that combines looking through your code, menus and files. Magnificent!



ACTION	WINDOWS	OS X
Find usages	Ctrl + Alt + F7	⌘ + F7
Find usages (results)	Ctrl + Alt + Shift + F7	⌘ + Alt + F7
Find / Replace in file	Ctrl + F	⌘ + F / ⌘ + R
Find / Replace in projects	Ctrl + Shift + F	⌘ + Shift + F / ⌘ + Shift + R
Find next	F3	F3
Select in projects / files / favorites	-	-



ACTION	WINDOWS	OS X
Find usages	Ctrl + Shift + G	⌘ + Shift + G
Find usages (results)	-	-
Find / Replace in file	Ctrl + F	⌘ + F
Find / Replace in projects	Ctrl + H	Ctrl + H
Find next	Ctrl + J	⌘ + K
Select in projects / files / favorites	-	-



ACTION	WINDOWS	OS X
Find usages	Alt + F7	Ctrl + F7
Find usages (results)	Alt + Shift + U	Ctrl + Shift + U
Find / Replace in file	Ctrl + F/H	⌘ + F/R
Find / Replace in projects	Ctrl + Shift + F/H	⌘ + Shift + F/H
Find next	F3	⌘ + G
Select in projects / files / favorites	Ctrl + Shift + 1/2/3	⌘ + Shift + 1/2/3



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



ACTION	WINDOWS	OS X
Open resource / Navigate to file	Ctrl + Shift + N	⌘ + Shift + O
Open type	Ctrl + N	⌘ + O
Go to symbol	Ctrl + Alt + Shift + N	⌘ + Alt + G
Go to line	Ctrl + G	⌘ + L
Recent files	Ctrl + E	⌘ + E
Tab / File switcher	Ctrl + Tab	⌘ + Shift + [/]

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. Again, let us reiterate how powerful IntelliJ IDEA's **Shift-Shift** search everywhere is. Yes, it can find and open a method declaration for you. Eclipse has the **Cmd / Ctrl + O** shortcut for opening a quick outline that allows you to find a method in the current class.



ACTION	WINDOWS	OS X
Open resource / Navigate to file	Alt + Shift + O	Ctrl + Shift + O
Open type	Ctrl + O	⌘ + O
Go to symbol	-	-
Go to line	Ctrl + G	Ctrl+G
Recent files	Ctrl + Shift + T	Shift + ⌘ + T
Tab / File switcher	Ctrl + Tab	Ctrl + Tab



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
Show UML popup	-	-
Activate editor	F12	F12



NetBeans


ACTION	WINDOWS	OS X
Maximize active window	Shift + Escape	Shift + Escape
Next view (editor)	Ctrl + Alt + PageUp/ PageDown	Ctrl + Alt + PageUp/ PageDown
Quick switch editor	-	-
Back	Alt + Left	Ctrl + Left
Forward	Alt + Right	Ctrl + Right
Show UML popup	-	-
Activate editor	-	-




ACTION	WINDOWS	OS X
Maximize active window	Ctrl + Shift + F12	⌘ + Shift + F12
Next view (editor)	Alt + Left / Right	Ctrl + Left / Right
Quick switch editor	Ctrl + E	⌘ + E
Back	Ctrl + [⌘ + [
Forward	Ctrl +]	⌘ +]
Show UML popup	Ctrl + Alt + U	⌘ + Alt + U
Activate editor	Ctrl + Tab	Ctrl + Tab

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. As every IDE supports that we have included it more for the sake of completeness.

		
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

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


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. Moreover, the shortcuts here are somewhat universal and commonly follow the same ones used for the OS of the developer's workstation, with one exception: by default, the IntelliJ IDEA keymap Cmd / Ctrl + D *duplicates* the line instead of *deleting* it.


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

CODE NAVIGATION

Code navigation is somewhat like File navigation, but taking into account that most often we deal with structured code. NetBeans has the most consistent shortcut scheme, everything features F12... we know it's a matter of taste. Every IDE recognizes that code navigation is paramount, so they make it easy.

		
ACTION	WINDOWS	OS X
Find usages / References in workspace	Ctrl + Shift + G	⌘ + Shift + G
Find usages results	-	-
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
Open / Navigate to member hierarchy	-	-

		
ACTION	WINDOWS	OS X
Find usages / References in workspace	Alt + F7	Alt + F7
Find usages results	Ctrl + Alt + Shift + F7	⌘ + Alt + Shift + F7
Quick outline / File structure	Ctrl + F12	⌘ + F12
Inspect code hierachy	Ctrl + Alt + H	Ctrl + Alt + H
Open / Navigate to declaration	Ctrl + Alt + B	⌘ + Alt + B
Open / Navigate to type hierarchy	Ctrl + H	Ctrl + H
Open / Navigate to member hierarchy	Ctrl + Shift + H	⌘ + Shift + H

		
ACTION	WINDOWS	OS X
Find usages / References in workspace	Alt + F7	Ctrl + F7
Find usages results	Alt + Shift + U	Ctrl + Shift + U
Quick outline / File structure	-	-
Inspect code hierachy	Alt + Shift + F12	Ctrl + Shift + F12
Open / Navigate to declaration	Ctrl + B	⌘ + B
Open / Navigate to type hierarchy	Alt + F12	Ctrl + F12
Open / Navigate to member hierarchy	Ctrl + F12	⌘ + F12



ACTION	WINDOWS	OS X
Refactor this	-	-
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
Extract / Assign to field	-	-
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.



ACTION	WINDOWS	OS X
Refactor this	Ctrl + Alt + Shift + T	⌘ + Alt + Shift + T
Show quick refactoring menu	-	⌘ + Shift + T
Rename	Ctrl + Alt + R	Shift + F6
Surround with	Ctrl + Alt + T	⌘ + Alt + T
Extract local variable	Ctrl + Alt + V	⌘ + Alt + V
Extract / Assign to field	Ctrl + Alt + F	⌘ + Alt + F
Inline	Ctrl + Alt + N	⌘ + Alt + N
Extract method	Ctrl + Alt + M	⌘ + Alt + M



ACTION	WINDOWS	OS X
Refactor this	-	-
Show quick refactoring menu	-	-
Rename	Ctrl + R	Ctrl + R
Surround with	Alt + Enter	Ctrl + Enter
Extract local variable	Alt + Shift + V	Ctrl + Shift + V
Extract / Assign to field	Alt + Shift + E	Ctrl + Shift + E
Inline	-	-
Extract method	Ctrl + Shift + M	Ctrl + Shift + M



ACTION	WINDOWS	OS X
Quick fix	Ctrl + 1	⌘ + 1
Code completion	Ctrl + Space	Ctrl + Space
Smart code completion	-	-
Live templates	-	-



ACTION	WINDOWS	OS X
Quick fix	Alt + Enter	Alt + Enter
Code completion	Ctrl + Space	Ctrl + Space
Smart code completion	-	-
Live templates	-	-




ACTION	WINDOWS	OS X
Quick fix	Alt + Enter	Alt + Enter
Code completion	Ctrl + Space	Ctrl + Space
Smart code completion	Ctrl + Shift + Space	Ctrl + Shift + Space
Live templates	Ctrl + J	⌘ + J


CODE COMPLETION


Everyone is on par here, but IntelliJ IDEA especially shines here: they've added shortcuts for using live templates and two modes of code completion, featuring a smart one that is aware of Java class types is an amazing feat.

UNIVERSAL ACCESS

Eclipse is definitely in the lead here, though all three IDEs provide support for accessing and searching the menus and actions via a shortcut. But where Eclipse goes further is in the ability to show a keybindings cheat sheet with a single touch of the keyboard.

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

		
ACTION	WINDOWS	OS X
Shortcuts cheatsheet	-	-
Quick access / search everywhere	Ctrl + Shift + A	⌘ + Shift + A

		
ACTION	WINDOWS	OS X
Shortcuts cheatsheet	-	-
Quick access / search everywhere	Ctrl+I	⌘ + I

IDE SHORTCUT CHEATSHEETS

In this part, we have combined all those groovy shortcuts into easy-to-print tables of useful cheatsheets. Print them for quick access around your work area and become the next local guru in your team...

SEARCH

ACTION	WINDOWS	OS X
Find usages	Ctrl + Shift + G	⌘ + Shift + G
Find usages (results)	-	-
Find / Replace in file	Ctrl + F	⌘ + F
Find / Replace in projects	Ctrl + H	Ctrl + H
Find next	Ctrl + J	⌘ + K
Select in projects / files / favorites	-	-

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
Show UML popup	-	-
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 symbol	-	-
Go to line	Ctrl + L	⌘ + L
Recent files	-	-
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
Find usages results	-	
Quick outline / File structure	Ctrl + O	⌘ + O
Inspect code hierarchy	Ctrl + T	⌘ + T
Open / Navigate to declaration	F3	F3
Open / Navigate to type hierarchy	F4	F4
Open / Navigate to member hierarchy	-	-

REFACTORING

ACTION	WINDOWS	OS X
Refactor this	-	-
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
Extract / Assign to field	-	-
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
Smart code completion	-	-
Live templates	-	-

UNIVERSAL ACCESS

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 + Alt + F7	⌘ + F7
Find usages (results)	Ctrl + Alt + Shift + F7	⌘ + Alt + F7
Find / Replace in file	Ctrl + F	⌘ + F / ⌘ + R
Find / Replace in projects	Ctrl + Shift + F	⌘ + Shift + F / ⌘ + Shift + R
Find next	F3	F3
Select in projects / files / favorites	-	-

WINDOWS ACTIONS

ACTION	WINDOWS	OS X
Maximize active window	Ctrl + Shift + F12	⌘ + Shift + F12
Next view (editor)	Alt + Left / Right	Ctrl + Left / Right
Quick switch editor	Ctrl + E	⌘ + E
Back	Ctrl + [⌘ + [
Forward	Ctrl +]	⌘ +]
Show UML popup	Ctrl + Alt + U	⌘ + Alt + U
Activate editor	Ctrl + Tab	Ctrl + Tab

FILE NAVIGATION

ACTION	WINDOWS	OS X
Open resource / Navigate to file	Ctrl + Shift + N	⌘ + Shift + O
Open type	Ctrl + N	⌘ + O
Go to symbol	Ctrl + Alt + Shift + N	⌘ + Alt + G
Go to line	Ctrl + G	⌘ + L
Recent files	Ctrl + E	⌘ + E
Tab / File switcher	Ctrl + Tab	⌘ + Shift + [/]

TEXT EDITING ACTIONS

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

CODE NAVIGATION

ACTION	WINDOWS	OS X
Find usages / References in workspace	Alt + F7	Alt + F7
Find usages results	Ctrl + Alt + Shift + F7	⌘ + Alt + Shift + F7
Quick outline / File structure	Ctrl + F12	⌘ + F12
Inspect code hierarchy	Ctrl + Alt + H	Ctrl + Alt + H
Open / Navigate to declaration	Ctrl + Alt + B	⌘ + Alt + B
Open / Navigate to type hierarchy	Ctrl + H	Ctrl + H
Open / Navigate to member hierarchy	Ctrl + Shift + H	⌘ + Shift + H

REFACTORING

ACTION	WINDOWS	OS X
Refactor this	Ctrl + Alt + Shift + T	⌘ + Alt + Shift + T
Show quick refactoring menu	-	⌘ + Shift + T
Rename	Ctrl + Alt + R	Shift + F6
Surround with	Ctrl + Alt + T	⌘ + Alt + T
Extract local variable	Ctrl + Alt + V	⌘ + Alt + V
Extract / Assign to field	Ctrl + Alt + F	⌘ + Alt + F
Inline	Ctrl + Alt + N	⌘ + Alt + N
Extract method	Ctrl + Alt + M	⌘ + Alt + M

CODE COMPLETION

ACTION	WINDOWS	OS X
Quick fix	Alt + Enter	Alt + Enter
Code completion	Ctrl + Space	Ctrl + Space
Smart code completion	Ctrl + Shift + Space	Ctrl + Shift + Space
Live templates	Ctrl + J	⌘ + J

UNIVERSAL ACCESS

ACTION	WINDOWS	OS X
Shortcuts cheatsheet	-	-
Quick access / search everywhere	Ctrl + Shift + A	⌘ + Shift + A

SEARCH

ACTION	WINDOWS	OS X
Find usages	Alt + F7	Ctrl + F7
Find usages (results)	Alt + Shift + U	Ctrl + Shift + U
Find / Replace in file	Ctrl + F/H	⌘ + F/R
Find / Replace in projects	Ctrl + Shift + F/H	⌘ + Shift + F/H
Find next	F3	⌘ + G
Select in projects / files / favorites	Ctrl + Shift + 1/2/3	⌘ + Shift + 1/2/3

WINDOWS ACTIONS

ACTION	WINDOWS	OS X
Maximize active window	Shift + Escape	Shift + Escape
Next view (editor)	Ctrl + Alt + PageUp/ PageDown	Ctrl + Alt + PageUp/ PageDown
Quick switch editor	-	-
Back	Alt + Left	Ctrl + Left
Forward	Alt + Right	Ctrl + Right
Show UML popup	-	-
Activate editor	-	-

FILE NAVIGATION

ACTION	WINDOWS	OS X
Open resource / Navigate to file	Alt + Shift + O	Ctrl + Shift + O
Open type	Ctrl + O	⌘ + O
Go to symbol	-	-
Go to line	Ctrl + G	Ctrl+G
Recent files	Ctrl + Shift + T	Shift + ⌘ + T
Tab / File switcher	Ctrl + Tab	Ctrl + Tab

TEXT EDITING ACTIONS

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

CODE NAVIGATION

ACTION	WINDOWS	OS X
Find usages / References in workspace	Alt + F7	Ctrl + F7
Find usages results	Alt + Shift + U	Ctrl + Shift + U
Quick outline / File structure	-	-
Inspect code hierarchy	Alt + Shift + F12	Ctrl + Shift + F12
Open / Navigate to declaration	Ctrl + B	⌘ + B
Open / Navigate to type hierarchy	Alt + F12	Ctrl + F12
Open / Navigate to member hierarchy	Ctrl + F12	⌘ + F12

REFACTORING

ACTION	WINDOWS	OS X
Refactor this	-	-
Show quick refactoring menu	-	-
Rename	Ctrl + R	Ctrl + R
Surround with	Alt + Enter	Ctrl + Enter
Extract local variable	Alt + Shift + V	Ctrl + Shift + V
Extract / Assign to field	Alt + Shift + E	Ctrl + Shift + E
Inline	-	-
Extract method	Ctrl + Shift + M	Ctrl + Shift + M

CODE COMPLETION

ACTION	WINDOWS	OS X
Quick fix	Alt + Enter	Alt + Enter
Code completion	Ctrl + Space	Ctrl + Space
Smart code completion	-	-
Live templates	-	-

UNIVERSAL ACCESS

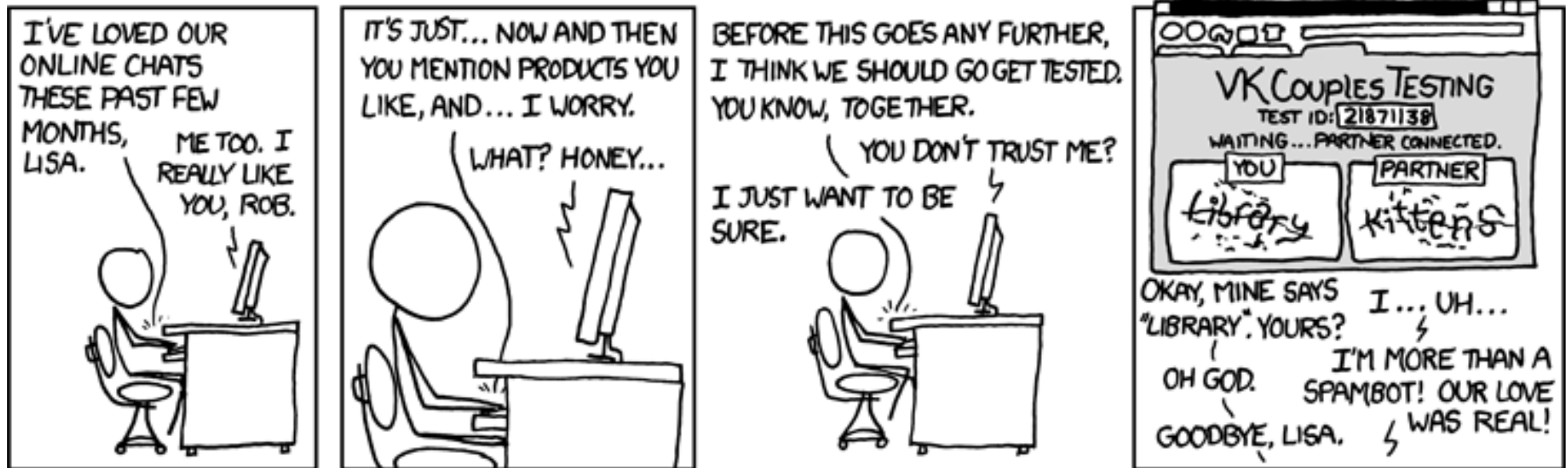
ACTION	WINDOWS	OS X
Shortcuts cheatsheet	-	-
Quick access / search everywhere	Ctrl+I	⌘ + I

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, we hope this report has helped shine a little light how to make your IDE experience considerably more enjoyable.

As we stated earlier, we want this report to be an accessible guide for learning the key bindings and cool shortcuts of the top IDEs in the market. Hopefully, you've learned a bit more about your current IDE, 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 ;-)



<http://xkcd.com/632/>



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)