

IDEA 2016.1 EAP

What is Early Access Program

We at JetBrains believe that making tools for developers should greatly involve listening to developers. Our Early Access Program lets development community closely participate in discussions devoted to IntelliJ IDEA and influence development planning, from early stages onwards. Early Access Program allows you to try pre-release versions of our software to evaluate features that will be added to the next release.



Be Careful

It is important to distinguish EAP from traditional pre-release software. Please note that the quality of EAP versions may at times be way below even usual beta standards.



Feedback

Please report errors and feature requests to our [issue tracker](#) or discuss them in [EAP users forum](#). Please do not add comments on this page directly.

What is IntelliJ IDEA 2016.1?

IntelliJ IDEA 2016.1 is update of IntelliJ IDEA released 17.03.2016.

Download IntelliJ IDEA 2016.1

IntelliJ IDEA 2016.1.4 RC (build 145.2070.6), released on Aug 03, 2016

Download	Description	Sha256 checksum
ideaIU-2016.1.4.exe	Windows installer (includes bundled JRE 1.8u77)	ideaIU-2016.1.4.exe.sha256
ideaIU-2016.1.4.win.zip	Windows ZIP archive (includes bundled JRE 1.8u77)	ideaIU-2016.1.4.win.zip.sha256
ideaIU-2016.1.4.tar.gz	Linux archive with x64 Java runtime (includes bundled JRE 1.8u76 custom build)	ideaIU-2016.1.4.tar.gz.sha256
ideaIU-2016.1.4-no-jdk.tar.gz	Unix archive without Java runtime	ideaIU-2016.1.4-no-jdk.tar.gz.sha256
ideaIU-2016.1.4.dmg	Mac installer (includes bundled JRE 1.8u76 custom build)	ideaIU-2016.1.4.dmg.sha256

Download IntelliJ IDEA 2016.1 Community Edition

Download	Description	Sha256 checksum
ideaIC-2016.1.4.exe	Windows installer (includes bundled JRE 1.8u77)	ideaIC-2016.1.4.exe.sha256
ideaIC-2016.1.4.tar.gz	Linux archive with x64 Java runtime (includes bundled JRE 1.8u76 custom build)	ideaIC-2016.1.4.tar.gz.sha256
ideaIC-2016.1.4-no-jdk.tar.gz	Unix archive without Java runtime	ideaIC-2016.1.4-no-jdk.tar.gz.sha256
ideaIC-2016.1.4.dmg	Mac installer (includes bundled JRE 1.8u76 custom build)	ideaIC-2016.1.4.dmg.sha256
ideaIC-2016.1.4-src.tar.bz2	Source tarball	ideaIC-2016.1.4-src.tar.bz2.sha256

Recent Changes

IntelliJ IDEA 2016.1 145.2070.6

[Release Notes](#)

IntelliJ IDEA 2016.1 145.1617.8

[Release Notes](#)

IntelliJ IDEA 2016.1 145.1616.11

[Release Notes](#)

IntelliJ IDEA 2016.1 145.1503.17

[Release Notes](#)

IntelliJ IDEA 2016.1 145.972.3

[Release Notes](#)

IntelliJ IDEA 2016.1 145.970.4

[Release Notes](#)

IntelliJ IDEA 2016.1 145.969.6

[Release Notes](#)

IntelliJ IDEA 2016.1 145.844.1

[Release Notes](#)

IntelliJ IDEA 2016.1 145.596.7

[Release Notes](#)

IntelliJ IDEA 2016.1 145.258.11

[Release Notes](#)

IntelliJ IDEA 2016.1 145.257.12

[Release Notes](#)

IntelliJ IDEA 2016.1 145.256.33

[Release Notes](#)

IntelliJ IDEA 16 145.184.1

[Release Notes](#)

IntelliJ IDEA 16 144.4199.23

If you're stepping over the code that doesn't match the running code, the IDE will warn you.

Now you can also disable inspections in the Diff Viewer for less distraction.

[Release Notes](#)

IntelliJ IDEA 16 144.3891.8

Those who work with Spring Boot projects should give this build a try because we've added a number of improvements in this area, including better coding assistance. In other news, Kotlin ships its first public RC build, which is now a part of IntelliJ IDEA 16 EAP.

Learn more by reading the [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.3600.7

Now when debugging a Java application, you can use Groovy expressions with Evaluate Expression and Watches. Previously you could do that only when debugging Groovy code. This is good because Groovy expressions are much shorter and more expressive (especially when you work with collections.)

Also, we've made it easier to debug multiple threads. Until now, resuming one thread meant IntelliJ IDEA would also resume all the other threads. Now you can change [this behavior](#) by enabling the Resume only the current thread option in Settings > Build, Execution, Deployment > Debugger > Stepping.

Git users will be happy to know that IntelliJ IDEA now supports [worktrees](#). This feature was introduced in Git 2.5 to make working with clones of a single repository simpler, because instead of making a repository clone you could create a lightweight worktree. The good news is that now IntelliJ IDEA supports those worktrees, so you can work with them just like you do with regular repositories.

The look and feel of Git Log has been updated with a better-looking toolbar and thinner splitters, and the table headers have been removed.

Last but not least, if you run any [code cleanup inspection](#) via Run Inspection by Name, you will now be offered to apply the quick-fix right away in the dialog.

To see the screenshots, check out the [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.3357.4

Sometimes it might be useful to move statements left or right: for changing the order of attributes of an XML element, method parameters, or elements in an array initializer. To make it possible we've added two new actions: Alt + Ctrl + Shift + Left/Right (Alt + Cmd + Shift + Left/Right for OS X).

Two new actions to help you work with Git branches, available via the Branches popup: Checkout with Rebase and Rename.

If you do an interactive rebase via the VCS | Git | Rebase main menu, IntelliJ IDEA now automatically stashes and unstashes local changes and assists with resolving conflicts.

The experience of applying patches has also been improved: you can now recover from the patches that could not be applied because of conflicts.

To learn more, read the corresponding [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.3143.6

Now the IDE suggests to add imports for static methods and constants in the very same way as it does for classes. If the Add unambiguous imports on the fly option is enabled in Settings | Editor | General | Auto Import, the IDE will import unambiguous static members on the fly. For more details, see the corresponding [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.2925.2

Built-in inspections that detect issues with use of concurrency annotations now support `javax.annotation.concurrent.*`.

The built-in Kotlin plugin in turn has been updated to [Beta 4](#). The new version has brought better incremental compilation performance along with minor changes to the language, libraries, and improved coding assistance in the IDE.

For more details, see the [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.2608.2

With this build IntelliJ IDEA and the whole IntelliJ platform migrates to Java 8.

The inspection that finds code duplicates now has an option to ignore generated code.

Also, we added a new inspection called 'Declaration access can be weaker'. When enabled it prevents making fields, methods and classes unnecessary public.

For Mercurial we made it possible to compare revisions of a file between branches.

To learn more, see the [blog post](#).

[Release Notes](#)

IntelliJ IDEA 144.2151.16

IntelliJ IDEA 16 adds the support for [RTL languages](#) (Arabic, Hebrew) to the editor.

The IntelliJ IDEA project model now supports Gradle custom source sets. This fixes a great number of [issues](#). Also, IntelliJ IDEA now can resolve EAR artifacts defined in your build scripts.

When you debug the code compiled without debug info, you will be provided with better name matching, and allowed to set and evaluate values for slot variables. The debugger is now aware of that your current thread is being blocked by another suspended thread, and suggests to resume it.

Merging with Git and other VCS got a little bit easier with the added by-word difference highlighting.

For those who would like to migrate their Guava code to Java 8 APIs, we've added a number of dedicated quick-fixes.

The [Android Studio 1.4](#) features are now available in the IntelliJ IDEA 16 EAP.

To learn more about IntelliJ IDEA 16 EAP, read the [announcement](#).