IDEA 12 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.

What is IntelliJ IDEA "Leda"?

IntelliJ IDEA "Leda" is major version of IntelliJ IDEA, released in December of 2012.

Download IntelliJ IDEA "Leda" 12.0.4

IntelliJ IDEA 12.0.4 has been released. Please download it from http://www.jetbrains.com/idea/download/

Download IntelliJ IDEA "Leda" Community Edition 12.0.4

IntelliJ IDEA 12.0.4 has been released. Please download it from http://www.jetbrains.com/idea/download/

Plugin Development Kit: Since the open-source Community Edition was released, we no longer provide the plugin development kit as a separate download. Please see http://www.jetbrains.org/display/IJOS/Writing+Plug-ins for instructions on setting up a plugin development environment.

Release Notes

IntelliJ IDEA 123.168 Release Notes
IntelliJ IDEA 123.165 Release Notes
IntelliJ IDEA 123.150 Release Notes
IntelliJ IDEA 123.139 Release Notes
IntelliJ IDEA 123.116 Release Notes
IntelliJ IDEA 123.111 Release Notes
IntelliJ IDEA 123.100 Release Notes
IntelliJ IDEA 123.95 Release Notes
IntelliJ IDEA 123.67 Release Notes
IntelliJ IDEA 123.63 Release Notes
IntelliJ IDEA 123.45 Release Notes
IntelliJ IDEA 123.23 Release Notes
IntelliJ IDEA 123.4 Release Notes
IntelliJ IDEA 122.813 Release Notes
IntelliJ IDEA 122.746 Release Notes
IntelliJ IDEA 122.694 Release Notes
IntelliJ IDEA 122.639 Release Notes
IntelliJ IDEA 122.592 Release Notes
IntelliJ IDEA 122.519 Release Notes
IntelliJ IDEA 122.395 Release Notes
IntelliJ IDEA 122.327 Release Notes
IntelliJ IDEA 122.264 Release Notes
IntelliJ IDEA 122.202 Release Notes
IntelliJ IDEA 122.100 Release Notes
IntelliJ IDEA 122.29 Release Notes
IntelliJ IDEA 120.305 Release Notes
IntelliJ IDEA 120.152 Release Notes
IntelliJ IDEA 120.11 Release Notes

Major changes in IntelliJ Idea 12

Core features:

- JDK 8 "Project Lambda" support (closures and extension methods) (work in progress)
- Bytecode viewer plugin
- Rearranger plugin is now bundled (work in progress)
- Recent Files (Ctrl-E) popup redesigned
- Breakpoints dialog redesigned
- Middle matching in completion
- Completion includes static methods and fields
- Faster project content indexing
- Native help viewer under Mac OS X
- Rewritten annotation processing configuration
- Compilation subsystem rewrite: "External make" feature
  - The new make is by default off, can be enabled on a per-project basis from "Settings | Compiler | Use out-of-process build for project compilation" checkbox.
  - Currently supported: incremental compilation of Java, Groovy, resource copying, UI Designer forms, Artifacts, Android, annotation processing, not-null instrumentation
  - Highlights:
    - all compilation tasks are performed in a separate process, fully independent from IDE's VM, no interference with AWT thread, no IDE resources consumption during build
    - lower memory requirements for the IDE.
    - in the make process javac compiler is called in "in-process" mode via java API, hence faster java sources compilation
    - java make: completely rewritten java classes dependency management for faster and more accurate search of dependent classes
    - 'automake' mode: automatic make triggered by events from IDE's virtual file system. If such make resulted in errors, those are highlighted in Project View and editor tabs
    - faster project rebuild
- New run configuration features
  - Possibility to run another configuration or external tool as "before run" action
  - Possibility to forbid running multiple copies of a run configuration
- New intentions and quickfixes
  - Split string on line breaks
  - Convert color between hex and decimal
  - Add type parameter
  - Call 'toString()'
  - Replace with cast variable
  - Assign all unassigned parameters to fields
  - Unwrap/Remove in boolean expressions
  - Replace assignment with setter call
  - Make class implement interface
  - Add ternary operation to avoid NPE
  - Convert array access to collection access
  - Move initializer to setUp()
  - Remove extra argument in method call
  - Convert if statement to ternary operator
  - Split 'try' statement with multiple resources
  - Merge nested try statements
  - Merge sequential method calls into call chain
  - Make package-local, Make protected, Make private
  - Push condition inside call
- New inspections
  - Non-final field in enum
  - 0 is used as first parameter for setXXX" for JDBC PreparedStatement
  - String concatenation missing whitespace
- Highlighting of recursive calls with gutter icons

Version control:

- Subversion: tree conflicts visualization and resolve (for some variants)
- Perforce: more reliable offline mode and authentication problem handling, ignored files support
- Git:
  - Commit & update optimizations.
  - Cherry-pick usability enhancements.
  - Show Diff for directories from history; Compare with Branch for directories.
  - Show Diff for merge commits allows to select a parent to compare with.
  - Pre-commit check if CRLF line separators are about to be committed to the repository.
- Github: usability improvements for Gists.
- Mercurial:

- Repository View.
- Show files affected in commit from file history and annotations.
- Diff and Merge tools:
    - UI improvements.
    - Ability to modify editor settings: enable/disable soft wraps, line numbers, whitespaces and indent guides.
    - Diff: show line separators indicator if they are different.
    - Merge: highlight applied changes; allow to apply both sides of a conflict.
    - External merge tools support.

Play Framework

- Play 1.2.5
    - Custom tags, navigation between views and controllers, annotations, list and field tags support.
- Play 2.0
    - Scala templates code assistance: navigation, code completion, find usages, etc.

Android Support

- UI Designer
- XML refactorings: Extract/Inline style, Extract/Inline layout, Use style where possible
- Application artifacts
- Android-style XML formatter

Groovy

- Static type-checking support (Groovy 2.0)
- Convert to Java refactoring
- 'Method can be static' inspection

Frameworks and Technologies

- Spring Core
    - Spring model: improved performance, multi-module (dependent) spring projects, mixed configurations (XML-, annotation- based)
    - application contexts navigation http://www.screenr.com/zoK8
    - unresolved bean: smart quickfixes for creating custom namespace elements
    - quickfixes for adapting Spring Facet/Model for unresolved but existing bean

- Spring Security support improvements
    - Security-namespace:
        - improve autocompletion/navigation support
        - support new/updated elements from 3.1 XSD
        - improved highlighting for configuration errors/problems
    - support OAuth1/2 (M6): XML highlighting, Spring Beans resolving
    - JSP Taglib: update completion
    - support for referencing properties of (custom) configured UserDetails, Config XML / Taglib
    - inject SQL into <jdbc-user-service> "*query" attributes (configurable via IntelliLang --> dialects)
    - inject PointcutExpression into <protect-pointcut> "expression"
    - JSP: resolve login/logout URL(s) provided by Spring Security

- Spring Batch (new)
    - Library/Framework wizard
    - navigation, completion and smart highlighting for <batch> namespace (2.1)
    - predefined SpringEL variables

- Spring Integration
    - 2.0.x-XSD: improved and updated autocompletion/highlighting, new inspection for highlighting deprecated elements
    - 2.1/2.2 XSD: updated, add support for AMQP, Gemfire, JPA and Redis
    - smart language injection (Spring EL, Groovy, XPath)
    - updated library wizard versions
    - <int-sec> namespace: integration with Spring Security

- Spring MVC
    - <mvc>-namespace updated to 3.1
    - Tiles: pattern-based "definitions" configuration, improve completion
    - <form>/<spring> taglib: completion for attributes with static values, CSS class references
    - improved completion presentation
    - @PathVariable <-> @RequestMapping handling extended

- Spring Web Flow
    - support 2.3-config XSD
    - JSF: support for spring-faces.xsd configuration elements
    - dedicated icon & structure view for *-flow.xml files
    - improve completion variants presentation

- support SpringEL or OGNL (Struts 2 plugin needed) instead of "plain" EL
- support <binding> bean property resolving
- support *-state "parent" resolving (flow#parentState)
- integrate with Spring Security authorities
- "Goto Related File" & gutter icons for related *-flow.xml files (parent & inheriting flows, <subflow-state> "subflow") and <bean-import> "resource"
- "Go To Parent" gutter icons for state-elements
- "Go To Symbol" for key navigation elements (view-state, end-state, ...)
- Compiler->Validation: validate flow.xml files during Make
- view-resolving (Tiles & anything via Spring MVC)
- performance improvements for complex (multi-file) flows
- completely rewritten Graph/Designer view
  - show Graph for selected or all flow.xml-files in module
  - improve graph presentation & layout, highlighted start-state ("link" style, blue bold)
  - show/hide dependencies (parent flow(s) and "parent"-relations between states)
  - show/hide sub-flows/event ("on-...") nodes
  - show/hide details for nodes (sub-elements, properties)
  - highlight all outgoing paths from selected node;
  - create/delete states and edges (transition, parent)

- Spring Data Jpa
  -
    - Java-based configuration of repositories (@EnableJpaRepositories) - http://www.screenr.com/On58

- CDI (JSR-299)
  - Extended CDI dependencies diagram: editing, navigation, DnD, etc. (done) - http://www.screenr.com/WKr8 and http://www.screenr.com/9Ar8
  - support beans.xml + configuration inspections (done)
  - simplify adding CDI in project + actual libraries (done)

- JBoss Drools support
  - rule completion: http://www.screenr.com/o0H7
  - constraints, bean properties - rename, navigation, etc. : http://www.screenr.com/p0H7
  - classes - navigation, search, rename: http://www.screenr.com/eOH7
  - download assistant: http://www.screenr.com/JXH7
  - structure: http://www.screenr.com/zXH7
  - rule file templates: http://www.screenr.com/daB7

- jBPM 5.0 (BPMN 2.0) - initial support (viewer, navigation, completion, simple error highlighting) - http://www.screenr.com/daG8
- Play! 1.2.4
  - navigation to related: Controller Methods <==> Views (http://www.screenr.com/YRA8)
  - FastTags support: http://www.screenr.com/uzA8
  - list tag implicit variables : http://www.screenr.com/5eK8
  - field tag support: http://www.screenr.com/5Nm8
  - Interceptor method annotations (@Before, @After, @Finally) : http://www.screenr.com/nXm8
- Hibernate 4.x
- JSF
  - @FacesComponent usages, rename, navigation(http://www.screenr.com/0G27), composite attributes processing (http://www.screenr.com/AC27)
  - component libraries extended support
    - RichFaces (4.4.2)
    - IceFaces (3.1.0)
    - PrimeFaces(3.4)
- Application Servers
  - Application server run configuration launch is made asynchronous -- IDEA UI will not hang while a run configuration is prepared to launch (all servers)
  - Correct deployment status handling of Tomcat integration -- JMX is utilized for the status detection now
  - Hot swap workflow is improved -- how swap will not fail after redeployment (all servers)
  - TomEE server intergation

- ActionScript / Flex support
  - AIR Mobile development: iOS Simulator support
  - UML diagrams improvements

- Cucumber Support
  - Cucumber language support is now independent from Ruby and is included in the IntelliJ IDEA Ultimate distribution
  - Cucumber for Java
  - Cucumber for Groovy