IntelliJ IDEA Project

IntelliJ IDEA Project runner allows you to build a project created in IntelliJ IDEA.
TeamCity versions up to 6.0 had Ipr (deprecated) which is now superseded by IntelliJ IDEA Project runner.

Supported IntelliJ IDEA features

TeamCity IntelliJ IDEA runner supports subset of IntelliJ IDEA features:

Feature	Status	Notes, limitations
Java	✅	Runner is able to compile Java projects
JUnit 3.x/4.x	✅, with limitations	• Test runner parameters are not supported • running of the Ant or Maven before tests start is not supported • alternative JRE is not supported
TestNG	✅, with limitations	• Test runner parameters are not supported • running of the Ant or Maven before tests start is not supported • running of the tests from group is not supported • alternative JRE is not supported
Application run configuration	✅, with limitations	• running of the Ant or Maven before tests start is not supported • altrenative JRE is not supported
J2EE integration	✅	runner is able to produce WAR and EAR archives with necessary descriptors
JPA	✅	runner adds necessary descriptors in produced artifacts
GWT	✅	runner can invoke GWT compiler and add compiler result to artifacts
Groovy	✅, with limitations	runner is able to compile projects with Groovy code and run tests written in Groovy, Groovy script run configurations are not supported
Android	✅	
Flex	➖	
Coverage	➖, if specified in run configurations	IntelliJ IDEA based coverage can be configured separately on the runner settings page
Profiling plugins	➖	

IntelliJ IDEA Project Settings

Option	Description
Path to the project	Use this field to specify the path to the project file (.ipr) or path to the project directory (root directory of the project that contains `.idea` folder). This information is required by this build runner to understand the structure of the project. ⓘ Specified path should be relative to the checkout directory.

Detect global libraries and module-based JDK in the *.iml files	If this option is checked, all the module files will be automatically scanned for references to the global libraries and module JDKs when saved. This helps you ensure all references will be properly resolved. Warning When this option is selected, the process of opening and saving the build runner settings may become time-consuming, because it involves loading and parsing all project module files.
Check/Reparse Project	Click to reparse the project and import build settings right from the IDEA project, for example the list of JDKs. If you update your project settings in IntelliJ IDEA - add new jdks, libraries, don't forget to update build runner settings by clicking Check/Reparse Project.
Working directory	Enter a path to a build working directory, if it differs from the build checkout directory. The license could not be verified: License Certificate has expired! Optional, specify if differs from the checkout directory.

Unresolved Project Modules and Path Variables

This section is displayed, when an IntelliJ IDEA module file (.iml) referenced from IPR-file:

- cannot be found
- allows you to enter the values of path variables used in the IPR-file.

To refresh values in this section click Check/Reparse Project.

Option	Description
<path_variable_name>	This field appears, if the project file contains path macros, defined in the Path Variables dialog of IntelliJ IDEA's Settings dialog. In the Set value to field, specify a path to project resources, to be used on different build agents.

Project JDKs

This section provides the list of JDKs detected in the project.

Option	Description
JDK Home	Use this field to specify JDK home for the project. When building with the Ipr runner, this JDK will be used to compile the sources of corresponding IDEA modules. For Inspections and Duplicate Finder builds, this JDK will be used internally to resolve the Java API used in your project. To run the build process itself the JDK specified in the `JAVA_HOME` environment variable will be used.

JDK Jar File Patterns	Click this link to open a text area, where you can define templates for the jar files of the project JDK. Use Ant rules to define the jar file patterns.
	The default value is used for Linux and Windows operating systems:
	`jre/lib/*.jar`
	For Mac OS X, use the following lines:
	`lib/*.jar`
	`../Classes/*.jar`

IDEA Home	If your project uses the IDEA JDK, specify the location of IDEA home directory
IDEA Jar Files Patterns	Click this link to open a text area, where you can define templates for the jar files of the IDEA JDK.

You can use references to external properties when defining the values, like `%system.idea_home%` or `%env.JDK_1_3%`. This will add a requirement for the corresponding property.

Java Parameters

Option	Description
JDK home path	Use this field to specify the path to your custom JDK which should be used to run the build. If the field is left blank, the path to JDK Home is read either from the `JAVA_HOME` environment variable on agent computer, or from `env.JAVA_HOME` property specified in the build agent configuration file (buildAgent.properties). If these both values are not specified, TeamCity uses Java home of the build agent process itself.
JVM command line parameters	Specify the desired Java Virtual Machine parameters, such as maximum heap size or parameters that enable remote debugging. These settings are passed to the JVM used to run your build. Example: `-Xmx512m -Xms256m`

Compilation settings

Option	Description
Only compile classes required to build artifacts and execute run configurations	Select whether to compile all classes in the project or only those classes which are required by run configurations or for building artifacts.

Artifacts

Option	Description
Artifacts to Build	Specify here names of the artifacts to be build that are configured in IntelliJ IDEA project.

Run configurations

Option	Description
Run configurations to execute	Specify here names of IntelliJ IDEA run configurations configured in the project to execute inside TeamCity build. Supported configuration types are: JUnit, TestNG and Application. Note, that run configurations specified here should be shared (via "Share" checkbox in IntelliJ IDEA Run/Debug Configurations dialog) and checked in to the version control.

Test Parameters

- To learn more about Run recently failed tests first and Run new and modified tests first options, please refer to the Running Risk Group Tests First page.
- Run affected tests only (dependency based) option will take build changes into account. With this option enabled runner will compute modules affected by current build changes and will execute only those run configurations which depend on affected modules directly or indirectly.

Code Coverage

Specify code coverage options, for the details, refer to IntelliJ IDEA Code Coverage page.

See also:

Administrator's Guide: IntelliJ IDEA Code Coverage