Run/Debug Configuration: JUnit

JUnit run/debug configurations define how the unit tests that conform to the JUnit testing framework should be run.

The dialog box consists of the following tabs:

- Configuration tab
- Code Coverage tab
- Logs tab

Click here for the description of the options that are common for all run/debug configurations.

Item	Description
Name	Specify the name of the run/debug configuration.

Configuration tab

Item	Description
All in package	Select this option to run all unit tests in a certain package. Specify the package in the **Package** field. To do that, type the package name or click ... and select the required package in the **Choose Package** dialog. Specify where in your project IntelliJ IDEA should look for the test classes related to the current package: - **In whole project:** IntelliJ IDEA will look for the test classes of the current package in all project modules. - **In single module:** IntelliJ IDEA will look for the test classes of the current package only in the module selected in the Use classpath or JDK of module field. - **Across module dependencies:** IntelliJ IDEA will look for the test classes of the current package only in the module selected in the Use classpath or JDK of module field and the modules that depend on that module.
Class	Select this option to run all tests in a class. Specify the fully qualified name of the class to be launched (passed to the JRE) in the **Class** field. To do that, type the class name or click ... and select the desired class in the **Choose Test Class** dialog. When doing so, you can search for the desired class by name, or search through the project.
Method	Select this option to run an individual test method. Specify the fully qualified name of the class to be launched (passed to the JRE) in the **Class** field. To do that, type the class name or click ... and select the desired class in the **Choose Test Class** dialog. When doing so, you can search for the desired class by name, or search through the project. Specify the method to be launched (passed to the JRE) in the **Method** field. To do that, type the method name or click ... and select the desired method in the **Choose Test Method** dialog.

Item	Description
Pattern	Select this option to run a set of test classes. This set may include the classes located in the same or different directories, packages or modules. Specify the necessary classes in the **Pattern** field. Each class in this field should be represented by its fully qualified name. The class names should be separated with `||`. You can type the class names in the **Pattern** field, or you can use + to the right of the field (`Shift+Enter`) to add classes to the set. When you click +, the **Choose Test Class** dialog is displayed. In this dialog, you can search for the class to be added by name, or search through the project.
VM options	If necessary, specify the string to be passed to the VM. This string may contain the options such as `-mx`, `-verbose`, etc. When specifying the options, follow these rules: - Use spaces to separate individual options, for example, `-client -ea -Xmx1024m`. - If an option includes spaces, enclose the spaces or the argument that contains the spaces in double quotes, for example, `some" "arg` or `"some arg"`. - If an option includes double quotes (e.g. as part of the argument), escape the double quotes by means of the backslashes, for example, `-Dmy.prop=\"quoted_value\"`. The `-classpath` option specified in this field overrides the classpath of the module.
Test runner parameters	If necessary, specify the list of arguments to be passed to the test runner. This list is specified in the same way as if you were entering these parameters in the command line. Use the same rules as for specifying the VM options.
Working directory	Specify the directory which will act as the current directory when running the test. This will be the root directory for all relative input and output paths. By default, the directory where the project file resides is used as a working directory. You can use `Ctrl+Space` to let IntelliJ IDEA help you fill in this and other fields. (In the editor, `Ctrl+Space` is used for code completion.)
Use classpath and JDK of the module	Select the necessary module from the list of your project modules. When executing the run/debug configuration, the classpath and JDK of the selected module will be used.

Item	Description
Use alternative JRE	Select this option to use a JRE other than the one associated with the project or module. Select the necessary JRE from the list or click … and select the JRE directory in the **Choose Path** dialog.

Code Coverage tab

Use this tab to configure code coverage monitoring options.

Item	Description
Choose code coverage runner	Select the desired code coverage runner. By default, IntelliJ IDEA uses its own coverage engine with the Sampling mode. You can also choose JaCoCo or Emma for calculating coverage.
Sampling	Select this option to measure code coverage with minimal slow-down.
Tracing	Select this option to collect accurate branch coverage. This mode is available for the IntelliJ IDEA code coverage runner only.
Track per test coverage	Select this check box to detect lines covered by one test and all tests covering line. If this check box is selected, becomes available on the toolbar of the coverage statistic pop-up window. This option is only available for the *Tracing mode* of code coverage measurement for the testing run/debug configurations. Refer to the section Viewing Code Coverage Results.
Merge data with previous results	When you run your unit testing or application configuration several times, use this item to calculate statistics in the **Project View**, taking into account the statistics of each time you have run the configuration. Finally, the line is considered *covered* if it is covered at least once.
Packages and classes to record code coverage data	Click and buttons to specify classes and packages to be measured. You can also remove classes and packages from the list by selecting them in the list and clicking the — button.
Enable coverage in test folders.	If this check box is selected, the folders marked as test are included in the code coverage analysis.

Logs tab

Use this tab to specify which log files generated while running or debugging should be displayed in the console, that is, on the dedicated tabs of the Run or Debug tool window.

Item	Description
Is Active	Select check boxes in this column to have the log entries displayed in the corresponding tabs in the Run tool window or Debug tool window.
Log File Entry	The read-only fields in this column list the log files to show. The list can contain: ■ Full paths to specific files. ■ Ant patterns that define the range of files to be displayed. ■ Aliases to substitute for full paths or patterns. These aliases are also displayed in the headers of the tabs where the corresponding log files are shown. If a log entry pattern defines more than one file, the tab header shows the name of the file instead of the log entry alias.
Skip Content	Select this check box to have the previous content of the selected log skipped.
Save console output to file	Select this check box to save the console output to the specified location. Type the path manually, or click the browse button and point to the desired location in the dialog that opens.
Show console when standard out changes	Select this check box to activate output console and bring it forward, if associated process writes to Standard.out.
Show console when standard error changes	Select this check box to activate output console and bring it forward, if associated process writes to Standard.err.
	Click this button to open the Edit Log Files Aliases dialog where you can select a new log entry and specify an alias for it.
	Click this button to edit the properties of the selected log file entry in the Edit Log Files Aliases dialog.
	Click this button to remove the selected log entry from the list.
	Click this button to edit the select log file entry. The button is available only when an entry is selected.

Toolbar

Item	Shortcut	Description
	Alt+Insert	Click this button to add new configuration to the list.
	Alt+Delete	Click this button to remove the selected configuration from the list.
	Ctrl+D	Click this button to create a copy of the selected configuration.

Item	Shortcut	Description
	Edit defaults	Click this button to edit the default configuration templates. The defaults are used for the newly created configurations.
 or	`Alt+Up` or `Alt+Down`	Use these buttons to move the selected configuration or group of configurations (folder) up and down in the list. The order of configurations or folders in the list defines the order, in which configurations appear in the **Run/Debug** drop-down list on the main toolbar.
		Use this button to create a new folder. If one or more run/debug configurations have the focus, then the selected run/debug configurations are automatically moved to the newly created folder. If only a category has the focus, then an empty folder is created. Move run/debug configurations to a folder using drag-and-drop, or  buttons.

Common options

Item	Description
Name	In this text box, specify the name of the current run/debug configuration. This field does not appear for the default run/debug configurations.
Defaults	This node in the left-hand pane of the dialog box contains the default run/debug configuration settings. Select the desired configuration to change its default settings in the right-hand pane. The defaults are applied to all newly created run/debug configurations.
Share	Select this check box to make the run/debug configuration available to other team members. If the directory-based project format is used, the settings for a run/debug configuration are stored in a separate xml file in the `.idea\runConfigurations` folder if the run/debug configuration is shared and in the `.idea\workspace.xml` file otherwise. If the file-based format is used, the settings are stored in the `.ipr` file for shared configurations or in the `.iws` file for the ones that are not shared. This check box is not available when editing the run/debug configuration defaults.

Item	Description
Before launch	Specify which tasks should be carried out before starting the run/debug configuration. The specified tasks are performed in the order that they appear in the list. **Item** | **Keyboard shortcut** | **Description**

Item	Item	Keyboard shortcut	Description
		Alt+Insert	Click this icon to add a task to the list. Select the task to be added: ■ **Run External tool.** Select this option to run an application which is external to IntelliJ IDEA. In the dialog that opens, select the application or applications that should be run. If the necessary application is not defined in IntelliJ IDEA yet, add its definition. For more information, see Configuring Third-Party Tools and External Tools. ■ **Make.** Select this option to have the project or module compiled. The Make Module command will be carried out if a particular module is specified in the run/debug configuration, and the Make Project command otherwise. If an error occurs during the compilation, IntelliJ IDEA won't attempt to start the run/debug configuration. ■ **Make, no error check.** The same as the **Make** option but IntelliJ IDEA will try to start the run/debug configuration irrespective of the compilation result. ■ **Build Artifacts.** Select this option to have an artifact or artifacts built. In the dialog that opens, select the artifact or artifacts that should be built. See also, Configuring Artifacts. ■ **Run Ant target.** Select this option to have an Ant target run. In the dialog that opens, select the target to be run. For more information, see Ant. ■ **Generate CoffeeScript Source Maps.** Select this option to have the source maps for your CoffeeScript sources generated. In the dialog that opens, specify where your CoffeeScript source files are located. For more information, see CoffeeScript Support. ■ **Run Maven Goal.** Select this option to have a Maven goal run. In the dialog that opens, select the goal to be run. For more information, see Maven. ■ **Run Remote External tool**: Add a remote SSH external tool. Refer to the section Remote SSH External Tools for details.
		Alt+Delete	Click this icon to remove the selected task from the list.

See Also

Concepts:

- Run/Debug Configuration

Procedures:

- Creating Run/Debug Configuration for Tests
- Testing

Language and Framework-Specific Guidelines:

- Testing Frameworks

Reference:

- Debugger

Web Resources:

- Developer Community