

Run/Debug Configuration: JSR45 Compatible Server

Use this dialog to create or edit [run/debug configurations](#) for JSR45-compatible application servers.

Generally, you may want to use this type of run/debug configuration under the following conditions:

- The server you are going to work with supports [JSR-45](#).
- IntelliJ IDEA doesn't provide a dedicated plugin for integration with this server.

On this page:

- [Server tab](#)
- [Logs tab](#)
- [Code Coverage tab](#)
- [Startup/Connection tab](#)
- [Before Launch options](#)
- [Toolbar](#)

Server tab

Item	Local/Remote	Description
Application Server	Both	Select the server configuration to be used. If the run/debug configuration is intended for working with a remote server, the same server version should be configured locally and associated with the run/debug configuration. Click Configure to create a new server configuration or edit an existing one. (The Application Servers dialog will open.)
Start browser	Both	Select this check box to run the default Web browser to study your application output there.
With JavaScript debugger	Both	If this check box is selected, the Web browser is started with the JavaScript debugger enabled. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">JavaScript debugging is available for Firefox only. Note that when you debug your JavaScript in Firefox for the first time, the JetBrains Firefox extension will be installed.</div>
Startup page	Both	In this field, specify the URL the browser should go to when started. In most typical cases, this URL will correspond to the root of your Web application or its starting page.

Item	Local/Remote	Description
VM options	Local	<p>If necessary, specify the command-line options to be passed to the server JVM at the server start.</p> <p>If you need more room to type, click  next to the field to open the VM Options dialog where the text entry area is larger.</p> <p>When specifying the options, follow these rules:</p> <ul style="list-style-type: none"> ■ Use spaces to separate individual options, for example, <code>-client -ea -Xmx1024m</code>. ■ If an option includes spaces, enclose the spaces or the argument that contains the spaces in double quotes, for example, <code>some "arg or "some arg"</code>. ■ If an option includes double quotes (e.g. as part of the argument), escape the double quotes by means of the backslashes, for example, <code>-Dmy.prop=\"quoted_value\"</code>.
On 'Update' action	Local	Select the action to be performed when the application is updated in the Run or the Debug tool window ( .
Show dialog	Local	<p>Select this check box if you want to see the Update <application name> dialog every time you perform the Update action.</p> <p>The Update <application name> dialog is used to view and change the current update option (for example, Restart server) prior to actually updating the application.</p>
JSP's package	Both	Specify the Java package prefix to be used for jsp-to-servlet translation.
VM options variable	Local	If there is a variable which stores the command-line JVM options, you can specify the name of this variable in this field. In this way, you can pass the corresponding options to the server JVM at the server start.
Port	Both	Specify the HTTP server port.
Use JSP's line mapping model specific for WebSphere 5.1	Both	<p>If the run/debug configuration is intended for working with WebSphere Server 5.1, select this check box to be able to debug your JSPs.</p> <p>(To be able to debug JSPs, it's necessary to maintain relationships between the lines of the source files and the corresponding positions in the compiled code. All the servers except WebSphere 5.1 form such relationships similarly.)</p>
Host	Remote	The DNS name or the IP address of the server host (for example, <code>localhost</code> , <code>127.0.0.1</code> , etc.).

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	<p>The read-only fields in this column list the log files to show. The list can contain:</p> <ul style="list-style-type: none"> ■ 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. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>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.</p> </div>
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.
	<p>Click this button to edit the select log file entry.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>The button is available only when an entry is selected.</p> </div>

Code Coverage tab

Use this tab to configure [code coverage](#) monitoring options.

Note that this tab is not available for remote servers.

Item	Description
Choose code coverage runner	Select the desired code coverage runner.
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.
Packages and classes to record code coverage data	<p>If necessary, specify the classes and packages to be measured.</p> <p>Use  or  to add classes or packages to the list.</p> <p>To remove the classes or packages from the list, select the corresponding list items and click .</p>
Enable coverage in test folders.	Select this check box to include the test source folders in code coverage analysis.

Startup/Connection tab

Item	Local/Remote	Description
 Run  Debug	Both	Select Run or Debug to show settings either for the run or the debug mode.
 Run with Coverage	Local	Select Run with Coverage to show settings for run with coverage mode.

Item	Local/Remote	Description
Startup script	Local	<p>Specify the script to be used to start the server. If necessary, you can also specify the script parameters and the options to be passed to the server JVM.</p> <p>You can provide all the necessary information right in the field, by typing. As an alternative:</p> <ul style="list-style-type: none"> ■ To specify the script, click  and select the desired script in the dialog that opens. ■ To specify the parameters, click  and specify the script parameters and VM options in the Configure VM and Program Parameters dialog. <p>When specifying the parameters and options, follow these rules:</p> <ul style="list-style-type: none"> ■ Use spaces to separate individual parameters and options, for example, <code>-client -ea -Xmx1024m</code>. ■ If a parameter or an option includes spaces, enclose the spaces or the argument that contains the spaces in double quotes, for example, <code>some" "arg</code> or <code>"some arg"</code>. ■ If a parameter or an option includes double quotes (e.g. as part of the argument), escape the double quotes by means of the backslashes, for example, <code>-Dmy.prop=\"quoted_value\"</code>.

Item	Local/Remote	Description
Shutdown script	Local	<p>Specify the script to be used to stop the server. If necessary, you can also specify the script parameters and the options to be passed to the server JVM.</p> <p>You can provide all the necessary information right in the field, by typing. As an alternative:</p> <ul style="list-style-type: none"> ■ To specify the script, click  and select the desired script in the dialog that opens. ■ To specify the parameters, click  and specify the script parameters and VM options in the Configure VM and Program Parameters dialog. <p>When specifying the parameters and options, follow these rules:</p> <ul style="list-style-type: none"> ■ Use spaces to separate individual parameters and options, for example, <code>-client -ea -Xmx1024m</code>. ■ If a parameter or an option includes spaces, enclose the spaces or the argument that contains the spaces in double quotes, for example, <code>some "arg</code> or <code>"some arg"</code>. ■ If a parameter or an option includes double quotes (e.g. as part of the argument), escape the double quotes by means of the backslashes, for example, <code>-Dmy.prop=\"quoted_value\"</code>.
Pass environment variables	Local	<p>To pass specific variables to the server environment, select this check box and specify the variables:</p> <ul style="list-style-type: none"> ■ To add a variable, click  and specify the variable name and value in the Name and Value fields respectively. ■ To remove a variable from the list, select the variable and click .
Port	Local	Use this field to change the debugger port (if necessary).
Debugger Settings	Local	Click this button to edit the debugger options on the Debugger page of the Settings dialog .

Item	Local/Remote	Description
Transport	Remote	<p>Specify the "transport" settings for the connection with the debugger. In technical terms, these are the parameters for the <code>-Xrunjdpw</code> command-line option:</p> <ul style="list-style-type: none"> ■ Socket. Specify the debugger port in the Port field. The combination of these two settings translates into <code>-Xrunjdpw:transport=dt_socket,address=<port>,suspend=n,server=y</code> ■ Shared memory. Specify the shared memory address in the corresponding field. The combination of these two settings translates into <code>-Xrunjdpw:transport=dt_shmem,address=<address>,suspend=n,server=y</code> <p>Note that as you change the transport settings, what follows <code>transport=</code> within <code>-Xrunjdpw</code> in the area above also changes. In this way you control the corresponding command-line debugger parameters which you cannot edit directly.</p>

Before Launch options

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	Shortcut	Description
------	----------	-------------

Item	Shortcut	Description
+	Alt+Insert	<p>Click this icon to add a task to the list. Select the task to be added:</p> <ul style="list-style-type: none"> ■ 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 Another Configuration. Select this option to have another run/debug configuration executed. In the dialog that opens, select the configuration to be run. ■ 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. Select this option to run a remote application which is external to IntelliJ IDEA. In the dialog that opens, select one or more remote applications to be run. If the necessary applications are not defined in IntelliJ IDEA yet, add definitions. For more information, see Configuring Third-Party Tools and External Tools.
-	Alt+Delete	Click this icon to remove the selected task from the list.
	Enter	Click this icon to edit the selected task. Make the necessary changes in the dialog that opens.

Item	Shortcut	Description
	Alt+Up/Alt+Down	Click these icons to move the selected task one line up or down in the list.
Show this page		Select this check box to have the run/debug configuration settings shown prior to actually starting the run/debug configuration.

Toolbar

Item	Shortcut	Description
	Alt+Insert	Use this icon or shortcut to create a new run/debug configuration.
	Alt+Delete	Use this icon or shortcut to delete the selected run/debug configuration.
	Ctrl+D	Use this icon or shortcut to create a copy of the selected run/debug configuration.
		Click this button to edit the default settings of the selected run/debug configuration. This button is disabled for defaults.
	Alt+Up/Alt+Down	Use these buttons to move the selected run/debug configuration up and down in the list. The order of configurations in the list defines the order in which the configurations appear in the corresponding list on the main toolbar. These buttons are disabled for defaults.
		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.

See Also

Concepts:

- [Run/Debug Configuration](#)

Procedures:

- [Creating and Editing Run/Debug Configurations](#)

Reference:

- [Debugger](#)

Web Resources:

