

# Customizing Preference Settings

Load Test preferences can be customized in the Preferences panel, which can be accessed by choosing **File > Customize Preferences** (or pressing **Alt + F2**).

Once this panel is open, you can modify settings through the following tabs:

- [Components](#)
- [GUI](#)
- [Scripting](#)
- [Start-Up](#)
- [System Properties](#)

## Components

This page displays deployment details of a built in component (e.g., a jar file that drives the load tests). It displays component deployment properties such as the Java class name of the main component class as well as component validation messages.

## GUI

You can set the following settings in the GUI panel:

- **Themes:** Allows you to select the theme for the overall look and feel of the Load Test GUI.
- **Gradients:** Displays various gradient properties corresponding to the Theme selected. If **Custom** is selected from the **Themes** drop-down menu, you can customize Gradient settings.
- **Test Tree Colors:** Displays the color settings for Test Suite Labels and Test Suite Shared Artifacts Labels. If **Custom** is selected from the **Themes** drop-down menu, you can customize Test Tree Color settings.
- **Use Background Image:** Adds an image to the background of Load Test.
- **Fonts:** Allows you to select the font and font size of the labels that display for **Test Trees**, **Test Suites**, **Test Tree Tools**, **Load Test Trees**, and **Load Test Folders**.



### Note

Load Test must be restarted in order for GUI changes to take effect.

## Scripting

You can specify Jython and Java properties used for custom scripts in the Load Test Preferences panel's Scripting tab.

- **Java:** For Java, you can specify the Java home directory and the path to the `javac` compiler. You need to specify these parameters if you want to compile Java methods within Load Test's Editor.



### Note

The `javac` compiler is not included in the Load Test installation.

- **Java Home:** Specifies the Java installation directory.
- **Java Classpath:** Specifies the Java classpath.
- **Java Script:** If you create scripts using Jython or JavaScript, you can specify a script template in the **Script Template** field.
  - **Script Template:** Whatever code is specified in this field will be used as the default code for inlined scripting in the language with which the field is associated. This is primarily useful for setting default inputs and common global variables. Script templates apply to scripts used in Method tools; they do not apply to JavaScript tests that run in a browser context.
- **Jython:** For Jython, you can specify the `jython.home` and `jython.path` variables. Both variables are used to locate Jython modules, and Jython code that does not import any Jython modules can use the Jython scripting support without setting either variable. If you set the `jython.home` and `jython.path` variables, you need to restart Load Test before the changes will take effect.
  - **JythonHome:** Specifies the Jython installation directory. `jython.home` must be a single directory.
  - **Jython Path:** Used to add to your path modules that are not in your `jython.home/Lib` directory. Multiple paths can be listed in `jython.pathLoad Test`
  - **Script Template:** Jython code that does not import any Jython modules can use the Jython scripting support without setting either the `jython.home` or `jython.path`.

## Start-Up

You can set the following setting in the **Start-up** panel:

- **Start server on Parasoft Load Test start-up:** Tells Load Test to start the SOAtest server used for asynchronous tests in the SOAtest project. See [Configuring Asynchronous Tests](#).
- **Show welcome screen on Load Test start-up:** Tells Load Test to display the welcome screen upon start-up.

## System Properties

This panel lets you add JAR files and class folders to the classpath if needed. Use the available controls to add or remove JAR files and class folders. The specified JAR files and classpaths will be added to the system's classpath and the corresponding classes will be loaded into the JVM after Load Test is restarted.