Configuring Test Configurations and Rules for Policies

This topic explains how to create and share custom Test Configurations (and any rule files or rule mapping files that they depend on) across the team.

Sections include:

- About Test Configurations
- Setting the Favorite Test Configuration
- Creating a Custom Test Configuration
- Deploying Test Configurations Across the Team

About Test Configurations

Every test run of C++test —both in the GUI or from the command line interface—is based on a test configuration that defines a test scenario and sets all related test parameters. You can perform testing with built-in test configurations shipped with C/C++test that are based a variety of popular test scenarios. Alternatively, you can create a custom test configuration by duplicating one of the built-in test configurations and modifying the duplicate, or by creating a new test configuration from scratch; see Creating a Custom Test Configuration. Test configurations that are created and stored on your DTP server can only be customized directly on DTP; see DTP Test Configurations.

For your convenience, you can configure a Favorite test configuration that you plan to use most frequently. The Favorite test configuration defines the default test scenario and can easily be run from the **Parasoft** menu, with the **Test Using** tool bar button, or from the command line interface; see Setting the Favorite Test Configuration.

Setting the Favorite Test Configuration

Setting a test configuration as Favorite makes it easily accessible in the Parasoft menu and the toolbar in your IDE to facilitate execution of your most frequent scenarios. In addition, the Favorite test configuration defines the default test scenario for command line execution.

You can specify more than one Favorite test configuration to configure easy access to test configurations you frequently use. C/C++test assumes that the test configuration with the highest position in the Favorite hierarchy is the default for command line use and for the **Run Tests** toolbar button.

To specify the Favorite test configuration:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Right-click the test configuration you want to set as Favorite.
3. Choose **Set As Favorite** from the context menu.
4. Specify the position of the test configuration:
   - **F**: indicates the highest priority and the default scenario; this test configuration will be marked with the 'F' icon
   - **1-3**: indicates the order in which the Favorite test configurations will appear in the GUI (**Test Using** buttons, the **Run Tests** button menu)

Creating a Custom Test Configuration

You cannot directly modify built-in test configurations shipped with C/C++test. To create a custom test configuration, duplicate an existing test configuration and modify the duplicate in the User-defined category.

ⓘ Test configurations that are created and stored on your DTP server can only be customized directly on DTP.

To create a custom test configuration:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Review the available test configurations and do one of the following:
   - To create a custom test configuration based on an existing test configuration: select and right-click the existing test configuration, then choose **Duplicate**.
   - To create a custom test configuration from scratch: click **New**.

   The duplicated or newly created test configuration will be added to the 'User-defined' category in the test configuration tree.
3. Select the new test configuration.
4. Modify the settings as needed; see Configuring Test Configurations for information about settings available in each tab.
5. (Optional) Set the Test Configuration as the Favorite test configuration; see Setting the Favorite Test Configuration.
6. Click **Apply**.
7. Click **Close**.

"Grayed-Out" Test Configurations = Incompatible Test Configurations
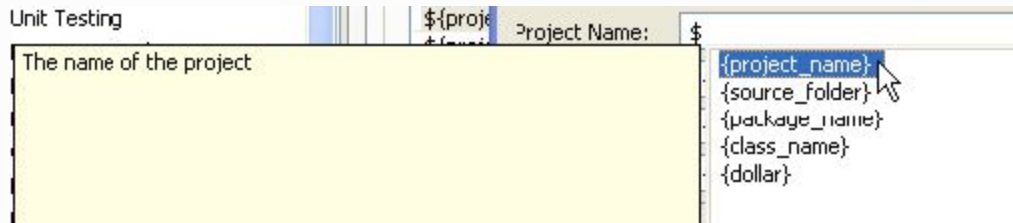
If a test configuration is "grayed out," this indicates that it was created with an incompatible C/C++test version, and cannot be edited or run with the current version.

✓

Using Variable Assist

When you type $, the variable assist feature automatically displays hints which variables you can use when specifying variables in the 'Test Configuration' panel.

Deploying Test Configurations Across the Team

You can deploy test configurations across the team via DTP, a primary platform for sharing and configuring team configurations, via Team Server, or by manually exporting and then importing a test configuration.

- DTP Test Configurations
- Team Test Configurations
- Importing/Exporting Test Configurations

DTP Test Configurations

If C/C++test is connected to DTP, you can analyze your code according to test configurations that are stored on the DTP server you have specified (see Connecting to DTP).

ℹ Test configurations that are created and stored on your DTP server can only be customized directly on DTP.

To customize a DTP test configuration:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Select the **DTP** category to view test configurations that are available on your DTP.
3. Click the test configuration you want to modify.
4. Click the **Configure in DTP** link to open the Test Configuration page in your DTP interface.
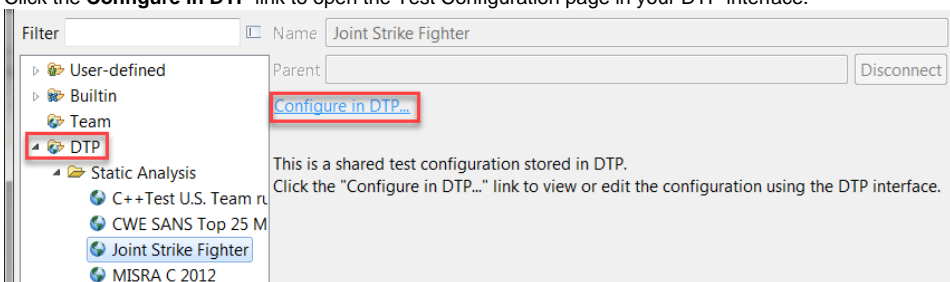
See the Test Configurations section in the Parasoft DTP User Guide for details about adding, configuring, and managing test configurations on DTP.

Team Test Configurations

Team test configurations are the test configurations that apply the team-specific test settings (for example, a selected sub-set of static analysis or custom test generation settings). Executing the same test configurations across the team allows you to consistently analyze your code and ensure that the team's quality and style guidelines are consistently applied across the code base.

Deploying Team Test Configurations

Once a team test configuration is added to Team Server, it is accessible from all connected team C/C++test installations. If the test configuration uses custom rules and/or rulemapping, they can be added to Team Server, then automatically accessed by all connected team C/C++test installations.

To share a team test configuration team-wide, the architect or manager performs the following procedure on a C/C++test installation that is already connected to Team Server:

1. Create a user-defined Test Configuration that applies the designated team settings; see Creating a Custom Test Configuration.
2. Choose **Parasoft> Test Configurations** to open the Test Configurations panel.
3. Right-click the test configuration you want to upload.
4. Choose **Upload to Team Server** from the context menu.

You can configure multiple team test configurations.

Modifying Team Test Configurations

Team test configurations can be directly edited in your IDE.

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Select **Team> [your_team_test_configuration]** in the left pane.
3. Modify the settings as needed.
4. Click either **Apply** or **Close** to commit the modified settings.

The settings will be updated on Team Server and shared across the team.

ⓘ Alternatively, you can first update the custom test configuration your test configuration was based on, and the re-upload that test configuration to Team Server.

> ⊘ **Setting the Team Favorite Test Configuration**
>
> Consider setting your team test configuration as the Favorite test configuration; see Setting the Favorite Test Configuration.

Deploying Custom Rule Mappings Across the Team

Rule mapping is a key part of configuring C/C++test to enforce your team's or organization's coding policy (for example, by customizing the built-in rule names, severities, and categories to match the ones defined in your policy).

You can use Team Server to ensure that all team members can access the `rulemap.xml` file you have created to customize Parasoft's rule categories and severity levels. For details on how to create this file, see Modifying Rule Categories, IDs, Names, and Severity Levels).

To upload a `rulemap.xml` file to the Team Server:

1. Launch C/C++test from a machine from which you can access the `rulemap.xml` file that you want to share.
2. Choose **Parasoft> Explore> Team Server** to open the Browsing dialog.
3. Open the **Rules** tab of the Browsing dialog.
4. Click the **Upload** button to open the file chooser.
5. Select the `rulemap.xml` file that you created, then click **Open**. The `rulemap.xml` file that you just uploaded will now be listed in the Browsing dialog's **Rules** tab. The rule configurations specified in this file will be available on all C/C++test installations connected to Team Server.
6. Click **Done**.
7. Click **Apply** and close the Parasoft Preferences dialog.
8. Restart the program. You do not have to stop the server first.

You can verify if the new rule setting are applied by in the Static tab of any test configuration.

> ⊘ **Tip**
>
> If you later modify the master `rulemap.xml` file, you must repeat the Deploying Custom Rule Mappings Across the Team procedure to upload the modified file; if the modified file is not uploaded, the modifications will not be shared.

Deploying Custom Rules Across the Team

You can use Team Server to ensure that all team members can access and check custom static analysis rules you have designed with the RuleWizard module. When Team Server manages a rule, all C/C++test installations connected to Team Server will automatically have access to the most recent version of the rule. If rule changes and the modified rule is uploaded to Team Server, the version on all team C-C++test installations will be updated automatically.

The architect (or other designated team member) performs the following procedure on one C/C++test that is already connected to Team Server:

1. Create one or more custom rules in RuleWizard.
2. Save each rule and assign it a `.rule` extension. You can save the rule in any location.
3. If you want a new rule to belong to a new category, create a new category as follows:
   a. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.

b. Select any test configuration.
c. Open the **Static> Rules Tree** tab.
d. Click the **Edit Rulemap** button.
e. Open the **Categories** tab.
f. Click **New**. A new entry will be added to the category table.
g. Enter a category ID and category description in the new entry. For instance, an organization might choose to use ACME as the category ID and ACME INTERNAL RULES as the description.
h. Note the location of the rulemap file, which is listed at the top of this dialog. You will need this information in step 9.
i. Click **OK** to save the new category.
4. Choose **Parasoft> Explore> Team Server**. The Browsing dialog will open.
5. Open the **Rules** tab of the Browsing dialog.
6. Click the **Upload** button. A file chooser will open.
7. Select one or more of the new `.rule` files that you created, then click **Open**. The `.rule` files that you just uploaded should now be listed in the Browsing dialog's **Rules** tab. All rules represented in this tab will be available on all C++test installations connected to Team Server.
8. Add additional team rules by repeating the previous two steps.
9. If you added any new rule categories or made any other changes to the rule mappings, click **Upload**, select the edited rulemap file, then click **Open**. The file that you just uploaded should now be listed in the Browsing dialog's **Rules** tab. This file will be available on the C/C++test installations that are connected to Team Server. This file controls how the team rules are categorized.
10. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations**.
11. Select any test configuration and open the **Static> Rules Tree** tab.
12. Click **Reload**. The new rule should be available in all available test configurations and classified under the Team category. The rule will be disabled by default.
13. If you want a Team Test Configuration to check these rules:
    a. Configure a new or existing Test Configuration to check these rules. The added rules will be disabled by default, so you will need to enable any rules that you want checked.
    b. Ensure that the modified Test Configuration available to the team as described in Deploying Test Configurations Across the Team. You must follow this procedure even if you are modifying a Test Configuration that is already shared.
14. Click either **Apply** or **Close** to commit the modified settings.

> ⊘ **Tip**
>
> - If your custom rule is visible in Test Configuration rules tree (for instance, if you imported it via the rules tree **Import** button), you can upload it to Team Server by simply right-clicking the rule, then choosing **Upload to Team Server** from the shortcut menu.
> - If you later modify a team rule, you must repeat the Deploying Custom Rules Across the Team procedure to upload the modified rule file; if the modified `.rule` file is not uploaded, the rule modifications will not be shared.

Removing Rules From Team Server

To remove a rule from Team Server, the architect (or another designated team member) performs the following procedure from C/C++test:

1. Choose **Parasoft> Explore> Team Server** to open the browsing dialog.
2. Open the **Rules** tab of the Browsing dialog.
3. Select the rule you want to remove.
4. Click **Delete**.
5. Click **Done**.

Importing/Exporting Test Configurations

If you want to share a custom test configuration with other team members or use it in an upgraded version of C/C++test, you can export this test configuration to a .properties file that can then be imported by another team member(s).

Exporting

To export a test configuration:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Right-click the test configuration you want to export.
3. Choose **Export** from the context menu, then use the file chooser to indicate where you want to save the .properties file that will be created for this test configuration.
4. Click **Save**. A dialog box will open to confirm the location of the newly created .properties file.

Importing

To import a test configuration from a .properties file:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Right-click the **User-defined** category.
3. Choose **Import** from the context menu to open a file chooser.
4. Navigate to the .properties file of the test configuration you want to import.
5. Click **Open**.

# Test Configurations – Advanced Topics

## Organizing User and Team Test Configurations into Subdirectories

You can organize your user and team test configurations into user-defined subdirectories.

To move a user or team test configuration to a user-defined subdirectory:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Right-click the test configuration you want to move to a subdirectory.
3. Choose **Set Folder** from the context menu.
4. Specify the subdirectory or subdirectories and click **OK**. In the following example, the test configuration is moved to the 'Security' subdirectory:
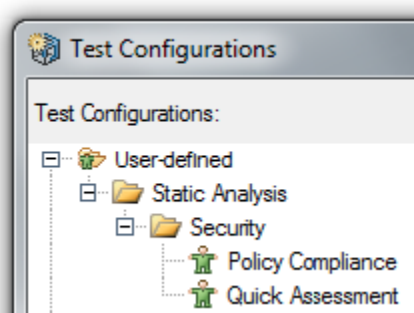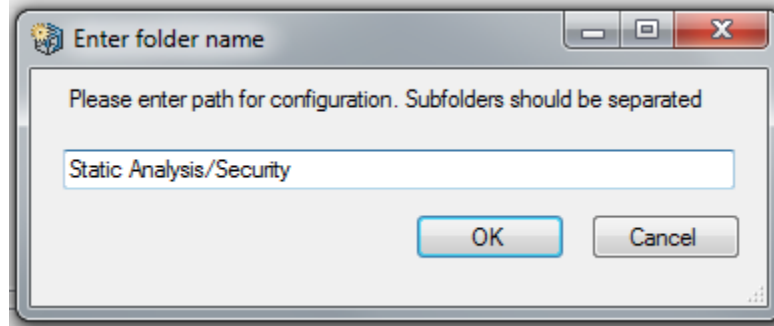
The created subdirectory will be available in the 'Test Using...' directories.
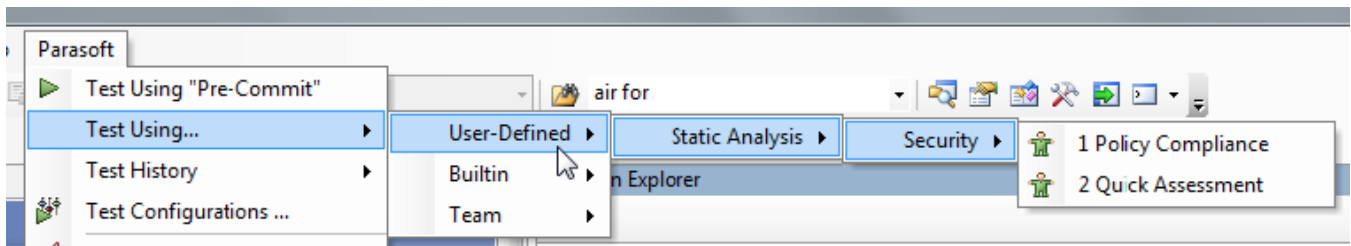
 Do NOT provide the path to the subdirectory in the command line mode. Subdirectories do not affect command line execution. In the above example, you can specify the following test configuration: `-config "team://Policy Compliance"`

Specifying Test Configuration Inheritance

If you want multiple test configurations to share some parameter settings (for example, if you want multiple test configurations to have the same rules enabled), you can create new child test configurations referring to one parent test configuration. A child test configuration will inherit the parent's settings; the value of each preference in the parent test configuration is used whenever the corresponding preference in the child test configuration is not configured.

Inheritance is recursive. For example, if MyConfig2 test configuration inherits the settings from MyConfig1 and MyConfig3 inherits the settings from MyConfig 2, then MyConfig3 will inherits some MyConfig1 settings, because it inherits MyConfig2 settings.

You can create a child test configuration from a test configuration displayed in the test configuration panel, or by specifying the URL of the test configuration (if the test configurations is available via HTTP).

To create a child from a test configuration displayed in the test configuration panel:

1. Open the Test Configurations panel.
2. Right-click the desired parent Test Configuration.
3. Choose **New Child** from the context menu.

To create a child from a test configuration available via HTTP:

1. Open the Test Configurations panel.
2. Right-click the **User-Defined** node.
3. Choose **New Child** from the context menu.
4. In the dialog that opens, enter the URL for the desired parent Test Configuration (http://config_address). For example: http://SOAtest.acme.com/configs/static.properties

To disconnect a child from its parent:

1. Open the Test Configurations panel.
2. Click the **Disconnect** button to the right of the **Parent** field.

> ⚠ **Important Notes**
>
>   - Once a parent-child relationship is set, that correlation cannot be modified. For example, if test configuration A is the parent of test configuration Z, you cannot switch test configuration Z's parent to test configuration B.
>   - Once a child test configuration is disconnected from its parent, it cannot be reconnected. All the inherited settings are applied directly in the child when disconnected.
>   - A given test configuration may have only one parent configuration. Multiple inheritance is not supported.

Comparing Test Configurations Using the Test Configurations Panel

If you want to see the differences between two test configurations, you can compare them to highlight the differences. You may want to compare test configurations if:

  - You customized a built-in static test configuration from a previous product version and want to verify what new rules have been added and enabled in the most current version of that built-in test configuration.
  - You want to verify what settings will be affected if you run a given test configuration in "Quick Mode."
  - You want to verify the differences between a child test configuration and its parent.

To compare any two test configurations displayed in the Test Configurations panel.

1. Select two test configurations.
2. Right-click the selection.
3. Choose **Compare**.

To compare a child Test Configuration with its parent:

1. Right-click the child test configuration.
2. Choose **Compare with> Parent Configuration**.

To compare a child test configuration with the corresponding test configuration in the Quick Mode:

1. Right-click the child test configuration.
2. Choose **Compare with> Quick Mode Configuration**.

Changes will be highlighted in the comparison editor that opens.
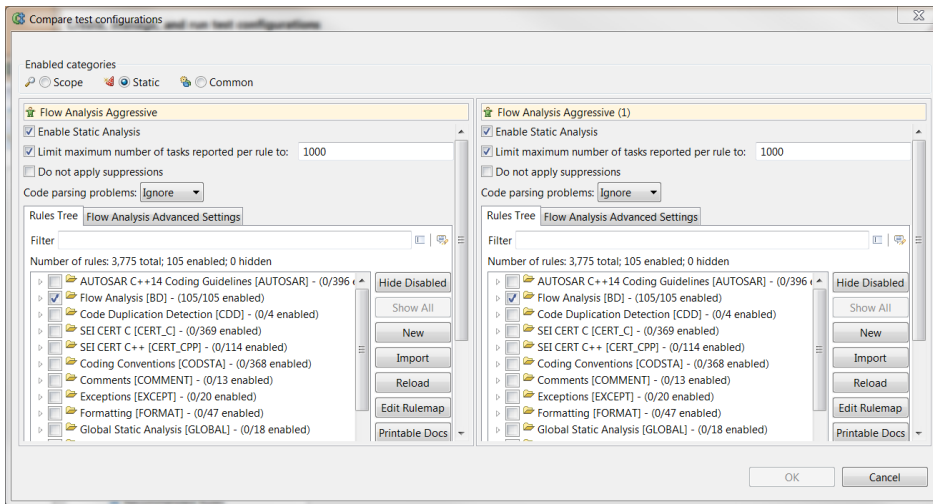
Comparint test configurations allows you to identify both the differences that are apparent within the panel (for example, when a setting is disabled in one test configuration and enabled in the other), as well as differences on a lower lever, such as different rule parametrization.

Comparing Test Configurations Using the .properties Files

To compare two test configurations:

1. Choose **Parasoft> Test Configurations** from the IDE menu or **Test Configurations** from drop-down menu on the **Run Tests** toolbar button to open the Test Configurations panel.
2. Right-click one of the test configurations you want to compare.
3. Choose **Export** from the context menu.
4. Navigate to the location where you want to save the `.properties` file (choose a folder that is in your workspace and available in the Package Explorer).
5. Repeat the above steps for the other test configurations you want to compare.
6. Select the two `.properties` files in the Navigator.
7. Right-click the selection.
8. Choose **Compare with> Each other**.

- Organizing User and Team Test Configurations into Subdirectories
- Specifying Test Configuration Inheritance
- Comparing Test Configurations Using the Test Configurations Panel
- Comparing Test Configurations Using the .properties Files