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:

Test Configuration Basics

About Test Configurations

Every test run of C++test —whether it is performed 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 (e.g., for static analysis, test execution, code review scanning, runtime error detection, and so on). To change how a test is performed, you can modify the settings for the Test Configuration that you plan to use. For example, to change the rules that are checked during static analysis, you modify the settings in the related static analysis Test Configuration.

C++test provides built-in Test Configurations that are based on a variety of popular test scenarios. However, because development projects and team priorities differ, custom Test Configurations are often desirable.

The default Test Configurations, which are in the Built-in category, cannot be modified. The recommended way to create a custom Test Configuration is to copy a Built-in Test Configuration to the User-defined category, then modify the copied Test Configuration to suit your preferences and environment. Alternatively, you could create a new Test Configuration "from scratch", then modify it as needed.

If C++test is connected to DTP, you can perform analysis with Test Configurations that are created and stored in your DTP server. Such configurations cannot be modified in your IDE, but they are configurable in DTP.

The Favorite Configuration should be set to the custom Test Configuration that you plan to use most frequently. By setting your preferred Test Configuration as the Favorite Configuration, you can easily run it from the **Parasoft** menu, the **Test Using** tool bar button, or from the command line interface.

Creating a Custom Test Configuration

To create a custom Test Configuration:

1. Open the Test Configurations panel by choosing **Parasoft> Test Configurations**.
2. Review the available Test Configurations to determine which (if any) you want to base your custom Test Configuration on.
    - Each product's built-in Test Configurations are described in the related product user's guide.
3. Do one of the following:
    - If you want to base a custom Test Configuration on a built-in Test Configuration or an available Team Test Configuration, right-click that Test Configuration, then choose **Duplicate**.
    - If you want to create a custom Test Configuration from scratch, click **New**.
4. Select the new Test Configuration, which will be added to the **User-defined** category.
5. Modify the settings as needed. Settings are described in product user guides, with the exception of code review settings, which are described in C onfiguring and Running PreCommit Code Review Scans and Configuring and Running Post-Commit Code Review Scans.
6. (Optional) Set the Test Configuration as a Favorite Test Configuration by right-clicking it, choosing **Set as Favorite** from the shortcut menu, then specifying what "favorite" position you want it to take (default, 1, 2, or 3). The configuration will then be set as a Favorite Configuration; the "favorite" icon will be added to that configuration in the Test Configurations tree.
7. Click **Apply**, then **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++test version, and cannot be edited or run with the current version.

Tip - Importing and Exporting to Share Test Configurations

If you are not using Parasoft Team Server to share test settings across your team, you can share custom Test Configurations with team members by exporting each Test Configuration that you want to share, then having your team members import it. See Importing/Exporting Test Configurations for details.

Changing the Favorite Test Configuration

The Favorite Configuration defines the test scenario used by default when a test is run. For example, if a test is started by clicking the **Run Tests** button, C++test will run that test based on the parameters defined in the Favorite Configuration. In addition to settings the Test Configuration that is used as the default, you can also mark other commonly-used Test Configuration as your favorites; this configures easy access to them.

To indicate which Test Configuration you want to set as the Favorite Configuration:

1. Open the Test Configurations panel by choosing **Parasoft> Test Configurations** or by choosing **Test Configurations** in the drop-down menu on the **Test Using** toolbar button.
2. Right-click the Test Configuration you want to set as a Favorite Configuration, choose **Set As Favorite** from the shortcut menu, then specify what "favorite" position you want it to take (default, 1, 2, or 3).

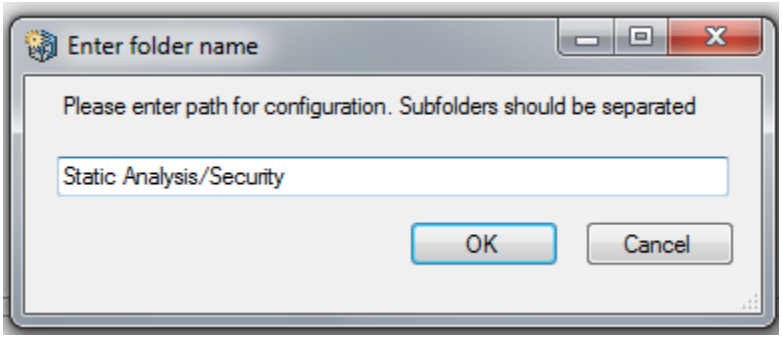
The configuration will then be set as a Favorite Configuration; the "favorite" icon will be added to that configuration in the Test Configurations tree.

Organizing User and Team Test Configurations into Subdirectories

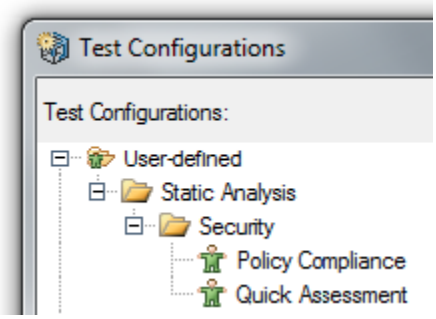
If desired, you can organize your user and team Test Configurations into user-defined subdirectories.

To move a user or team Test Configuration into a user-defined subdirectory:

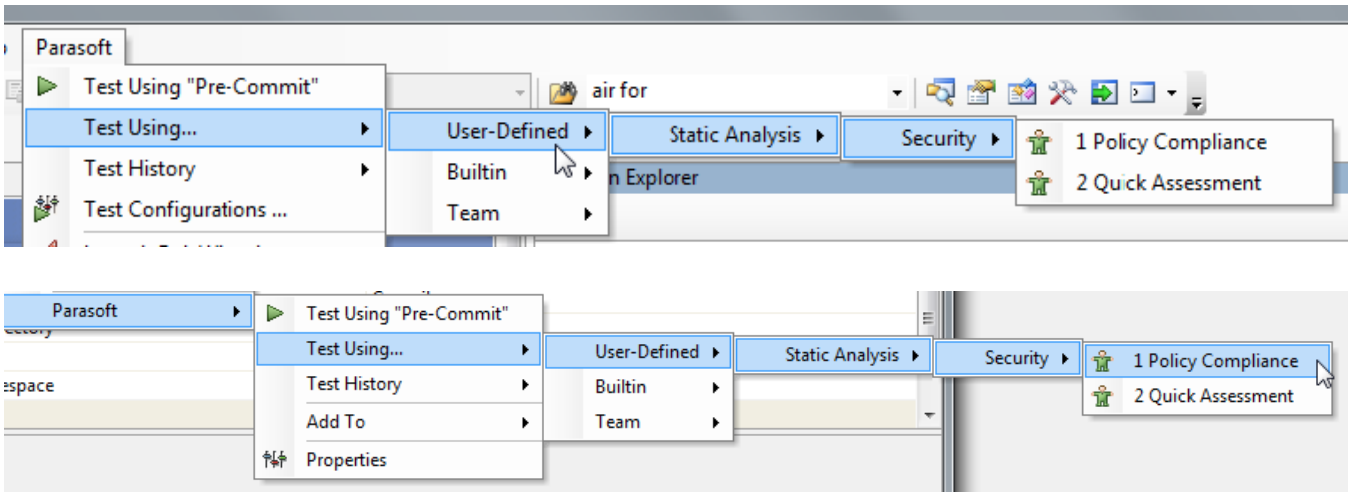
1. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations** or by choosing **Test Configurations** in the drop-down menu on the **Test** toolbar button.
2. Right-click the Test Configuration you want to move into a subdirectory, choose **Set Folder** from the shortcut menu, then specify the desired subdirectory or subdirectories.
    - For instance, if you want a Test Configuration to be presented under **Static Analysis> Security**, you would enter the following:



3. Click **OK**. The specified subdirectory will be created if needed, and the Test Configuration will be moved into it.



The created subdirectory will be available in the Test Using directories.



The command line mode invocation is NOT affected by subdirectories. Here, you would continue to refer to the Test Configuration by only its name (without paths). For example: `-config "team://Policy Compliance"`

Deploying Test Configurations Across the Team

About Team Test Configurations

Team Test Configurations are the Test Configurations that apply the team's designated test settings (for instance, the static analysis rules that your team has decided to follow, the test generation settings you prefer to use, etc.). When all team members use the designated Test Configurations, code will be checked consistently, and the team's quality and style guidelines will be applied consistently across the complete code base.

Deploying Team Test Configurations

Once a team Test Configuration is added to Team Server, it will be accessible on all connected team C++test installations. If a Test Configuration uses custom rules and/or rulemappings, they can be added to Team Server, then automatically accessed by all connected team C++test installations.

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

1. If you have not already done so, create a user-defined Test Configuration that applies the designated team settings.
    - See Creating a Custom Test Configuration for instructions.
2. Upload that configuration to Team Server as follows:
    a. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations**.
    b. Right-click the Test Configurations category that represents the Test Configuration you want to upload.
    c. Choose **Upload to Team Server** from the shortcut menu.

You can configure multiple team Test Configurations (for instance, one for static analysis, one for unit test generation, one for regression testing, etc.).

Tip

- If your Team Test Configuration uses custom rules or rule mappings, the related files can be shared as described later in this topic.

Modifying Team Test Configurations

Team Test Configurations can be directly edited from C++test.

1. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations** or by choosing **Test Configurations** in the drop-down menu on the **Test** toolbar button.
2. In the left pane, select **Team> [your_team_Test_Configuration]**.
3. Modify the settings as needed.
4. Click either **Apply** or **Close** to commit the modified settings.

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

Alternate Update Method

You can update a Team Test Configuration by modifying the User-Defined Test Configuration it was based on, then repeating the Deploying Test Configurations Across the Team procedure to re-upload the modified Test Configuration.

Setting the Team Favorite Test Configuration

The Team Favorite Configuration defines the test scenario used by default when a Team Server-connected team member runs a test. For example, if a test is started by clicking the **Test** button, C++test will run that test based on the parameters defined in the Team Favorite Configuration.

To set the Team Favorite Test Configuration:

1. Choose **Parasoft> Explore> Team Server**. The Browsing dialog will open.
2. Open the **Configurations** tab of the Browsing dialog.
3. Select the Test Configuration that you want to serve as the Team Favorite Test Configuration.
4. Click the **Set as Team Favorite** button.

Deploying Custom Rule Mappings Across the Team

Rule mapping is a key part of configuring C++test to enforce your team's or organization's coding policy (e.g. 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++test from a machine from which you can access the `rulemap.xml` file that you want to share.
2. Choose **Parasoft> Explore> Team Server**. The Browsing dialog will open.
3. Open the **Rules** tab of the Browsing dialog.
4. Click the **Upload** button. A file chooser will open.
5. Select the `rulemap.xml` file that you created, then click **Open**. The `rulemap.xml` file that you just uploaded should now be listed in the Browsing dialog's **Rules** tab. The rule configurations specified in this file will be available on all C++test installations connected to Team Server.
6. Click **Done**, click **Apply**, and then close the Parasoft Preferences dialog.
7. Restart the program. You do not have to stop the server first.
8. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations** or by choosing **Test Configurations** in the drop-down menu on the **Test Using** toolbar button.
9. Select any Test Configuration and open the **Static** tab. The new rule settings should be applied.

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++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++test installations will be updated automatically.

The architect (or other designated team member) performs the following procedure on one 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 any new rules should belong to a new category, create a new category as follows:
    a. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations** or by choosing **Parasoft> Test Configurations** in the drop-down menu on the **Test Using** toolbar button.
    b. Select any Test Configurations category.
    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 **Oper** . 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++test installations that are connected to Team Server and licensed for the applicable Parasoft product. 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++test:

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

Test Configurations: Advanced Topics

Specifying Test Configuration Inheritance

If you want multiple Test Configurations to share some of the same 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 present.

Inheritance is recursive; in other words, you could have the MyConfig2 Test Configuration inherit the settings from MyConfig1, and have MyConfig3 inherit the settings from MyConfig 2. MyConfig3 will thus inherit some MyConfig1 settings as it inherits MyConfig2 settings.

You can create a child Test Configuration from a Test Configuration shown in the Test Configuration panel, or by specifying a Test Configuration URL (for Test Configurations available via HTTP).

To create a child from a Test Configuration shown in the Test Configuration panel:

1. Open the Test Configurations panel.
2. Right-click the desired parent Test Configuration, then choose **New Child** from the shortcut 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, then choose **New Child** from the shortcut menu.
3. 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. Test Configurations that inherit from a parent must be created that way from the start using the "New Child" action.
- 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 no more than one parent configuration. Multiple inheritance is not supported.

Comparing Test Configurations

If you want to see the differences between two Test Configurations, you can compare them so that the differences are highlighted. For example, you might want to compare Test Configurations if:

- You customized a built-in static Test Configuration from a previous product version and want to see what new rules have been added and enabled in the most current version of that built-in Test Configuration.
- You want to know what settings will be impacted if you run a given Test Configuration in "Quick Mode."
- You want to zero in on the differences between a child Test Configuration and its parent.

To compare any two Test Configurations shown in the Test Configuration Manager

- Select those two Test Configurations, then right-click the selection and choose **Compare**.

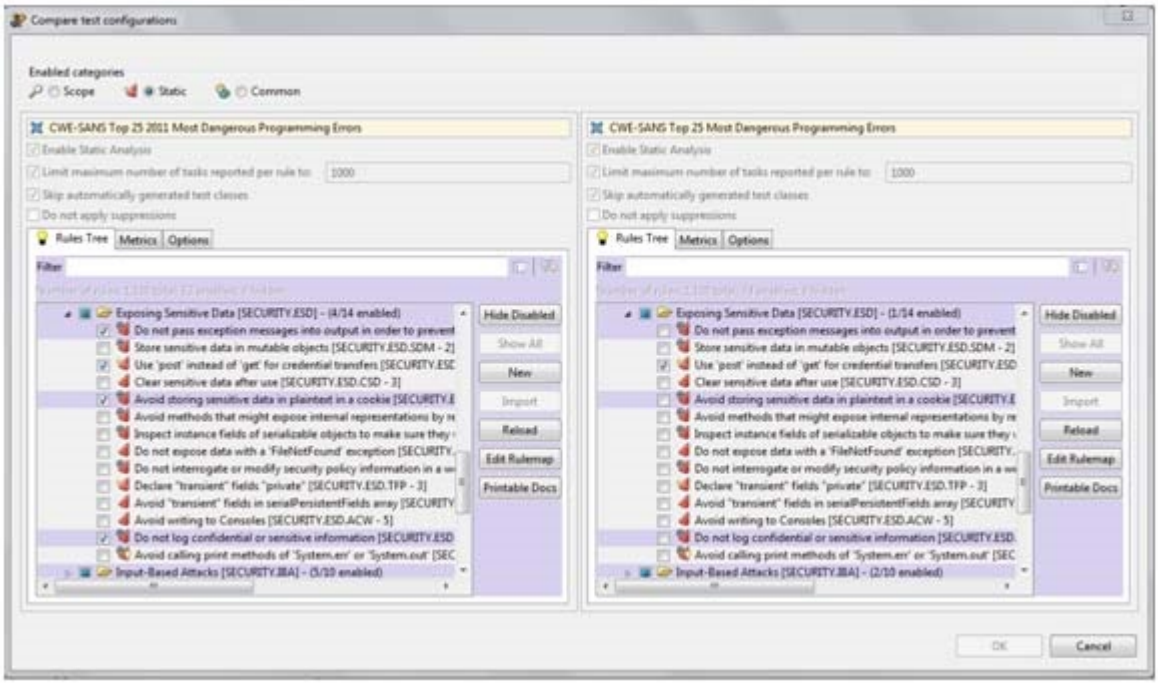
To compare a child Test Configuration with its parent:

- Right-click the child Test Configuration, then choose **Compare with> Parent Configuration**.

To compare a child Test Configuration with its Quick Mode equivalent:

- Right-click the child Test Configuration, then choose **Compare with> Quick Mode Configuration**.

Changes will be highlighted in the comparison editor that opens.



Note that the comparison identifies both differences that are apparent within the panel (e.g., a setting is disabled in one Test Configuration and enabled in another) as well as deeper level differences (e.g., a rule is parameterized differently in one Test Configuration).

Importing/Exporting Test Configurations

If you have created a Test Configuration that you want to share with team members or use in an upgraded version of C++test, you can export the Test Configuration into a properties file. That Test Configuration can then be added by importing the related properties file.

Exporting

To export a Test Configuration:

1. Open the Test Configurations panel by choosing **Parasoft> Test Configurations**.
2. Right-click the Test Configuration you want to export, choose **Export** from the shortcut menu, then use the file chooser to indicate where you want to save the properties file that will be created for this Test Configuration.

A properties file will then be saved in the designated location. A dialog box will open to confirm the location of the newly-created properties file.

Importing

To import a Test Configuration that was previously exported into a properties file:

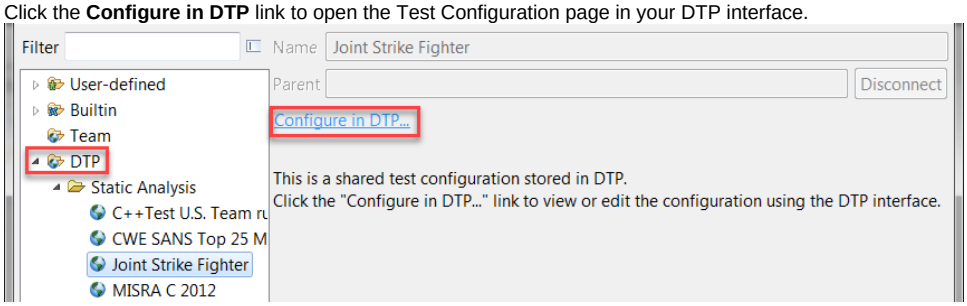
1. Open the Test Configurations panel by choosing **Parasoft> Test Configurations**.
2. Right-click the **User-defined** category, choose **Import** from the shortcut menu, then use the file chooser to select the appropriate properties file.

DTP Test Configurations

If 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). Test Configurations in DTP cannot be modified in the IDE, but they can be configured in DTP.

To customize a DTP Test Configuration:

1. Open the Test Configurations dialog by choosing **Parasoft> Test Configurations** or by choosing **Test Configurations** in the drop-down menu on the **Run Tests** toolbar button.
2. Select the **DTP** category to view Test Configurations that are available in your DTP and click the configuration you want to configure.
3. 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 Documentation for details.