Viewing Results

This topic explains how to view C/C++test results and customize their presentation.

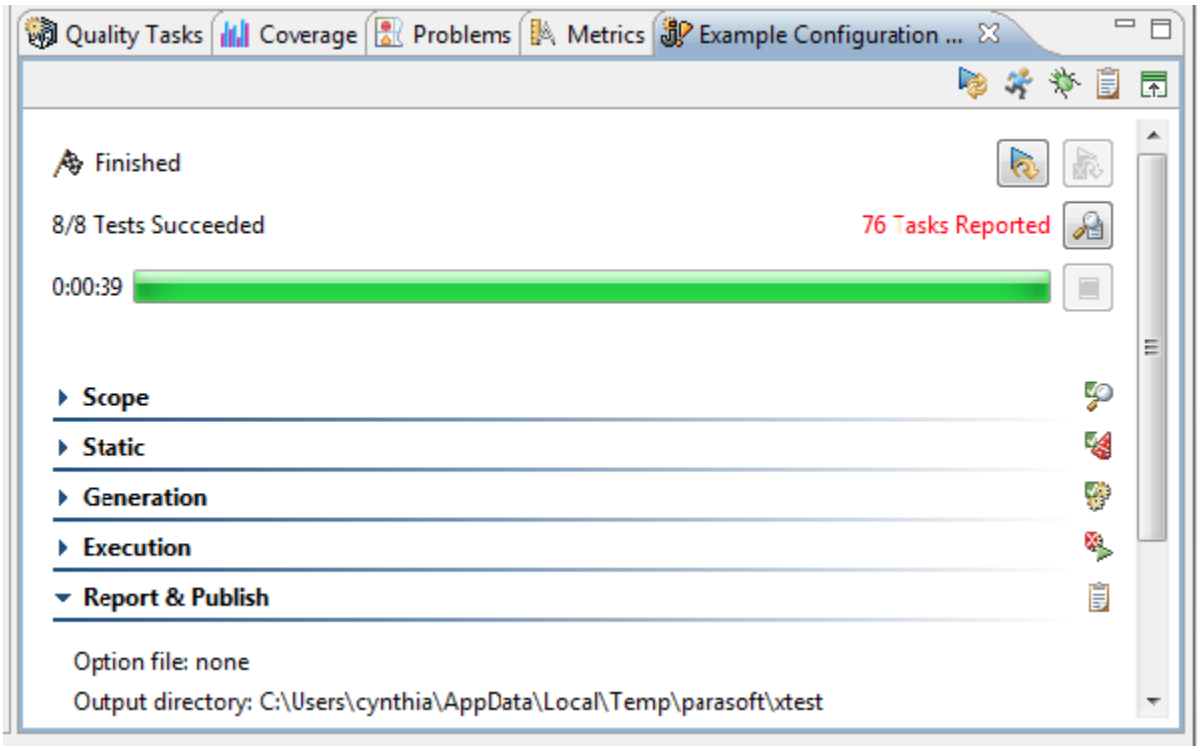
Sections include:

Accessing Results

Results can be accessed from a variety of locations in the GUI, as well as from command-line reports.

Results from Tests Run in the GUI

The **Test Progress** view reports test progress and status.



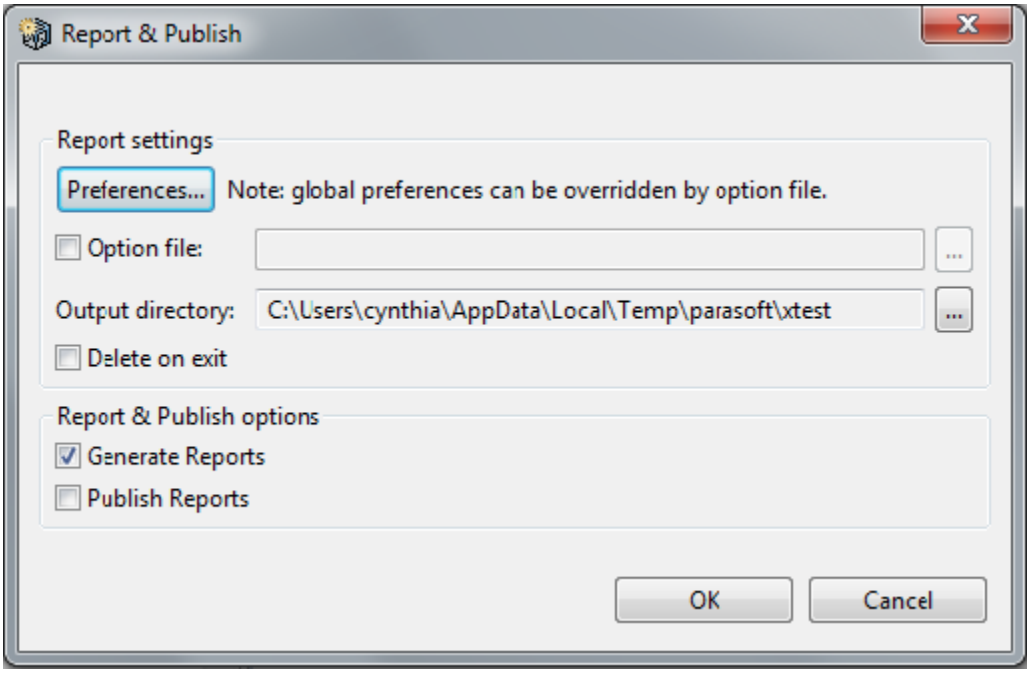
Note that:

- When a test runs, the view label changes from "Test Progress" to "Testing [Test Configuration name].
- Clicking the **Review tasks** button displays the results in the **Quality Tasks** view.

- A results summary for each analysis category is available in expandable sections.

- The toolbar buttons in the upper right corner of the **Test Progress** view allow you to generate reports.

This opens the **Report** dialog from which you can configure Report Preferences.



Quality Tasks View

Your assigned quality tasks generated during interactive testing or imported from command-line tests are shown in the **Quality Tasks** view. If this view is not available, choose **Parasoft> Show View> Quality Tasks** to open it. To see additional details, drill down into the **Quality Tasks** view tree. To toggle through the items reported in this view, use the arrow buttons in the view's toolbar.

The results are presented as a task list that helps you determine how to proceed with testing and code improvement. Tasks are organized by author, category, then by severity (if assigned). Severity levels range from 1 to 5— Severity 1 tasks are estimated to have the greatest chance of eliminating or preventing a critical bug, while Severity 5 tasks are estimated to have the least chance of eliminating or preventing a critical bug.

Source Code Markers

For tests that were run on source files, results are also reported at the source code level.

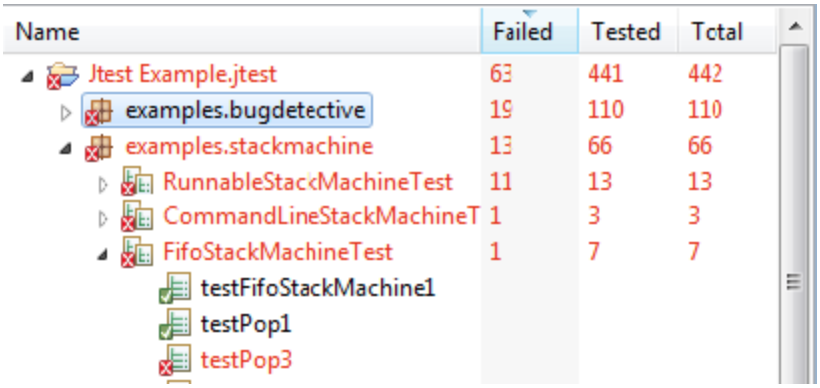
If you open the editor for a tested source file, markers will be placed next to the source code responsible for problems found. For static code analysis violations, markers are placed next to the line of code responsible for the violation. For unit test errors, markers are placed on the first line of the stack trace that matches the tested class. For unit test failures or for errors where the tested class is not known, markers are placed on the first line of the stack trace that matches the unit test class. To learn what problem a particular marker indicates, place your mouse over the marker and review the information in the popup window. Or, to go directly to the related **Quality Tasks** view message, right-click the source code responsible for the problem, choose **Show In> Quality Tasks** (for Eclipse) or **Parasoft> Show in Quality Tasks**

Console View

To see testing details, open the **Console** view during test execution. Testing details are reported here when a test is in process, and remain there until they are cleared or until another test is run.

Test Case Explorer View

The **Test Case Explorer** indicates the status of all available test cases. Red is used to mark the complete path to a failure.



To view any tasks related to a test listed in the Test Case Explorer, right-click that test's Test Case Explorer node, then choose **Show in Tasks**. Any tasks related to that test will then be shown in the Quality Tasks view.

Tips

- Many tree nodes report the line number at which an error or possible problem occurred. To view the related code, double-click the node that shows the line number, or right-click that node and choose **Go to** from the shortcut menu. The related editor will then open and highlight the designated line of code.
- You can use **Ctrl + C** to copy findings in the **Quality Tasks** view, then paste them into another document.
- You can filter and search in Test Case Explorer results as described in About the Test Case Explorer.

Results from Tests Run from the Command Line

For tests run from the command line, results are recorded in the generated report. If results were sent to DTP, results can be imported into the GUI as described in Importing Results into the UI. You can then review the results as if the test had been performed in the GUI.

Filtering Results

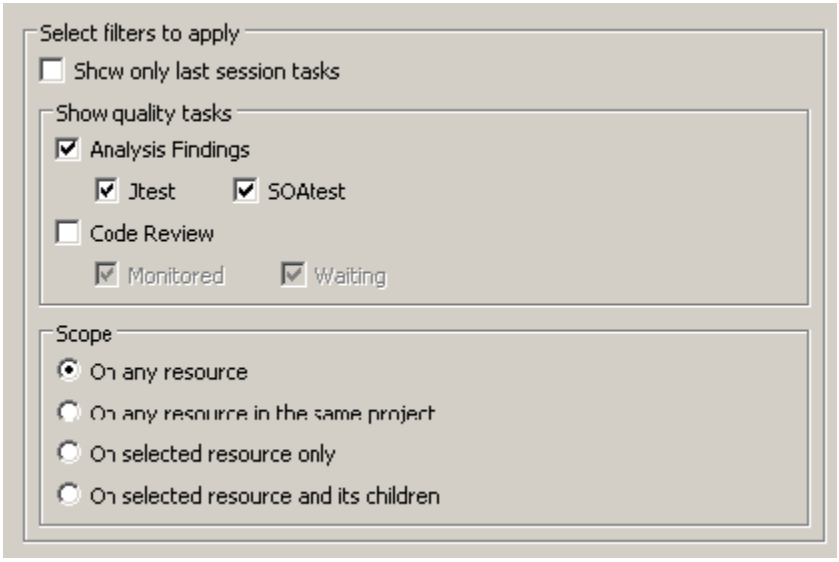
By default, the **Quality Tasks** view shows cumulative results for all tested resources. For example, if you imported results to your IDE, and then ran two tests from the GUI, the **Quality Tasks** view would show all imported tasks, plus all results from the subsequent two tests.

If you prefer to see only results from the last test session or from selected resources, you can filter results.

To filter results:

1. Click the filters button in the **Quality Tasks** view's toolbar.

2. Set the desired filter options in the dialog that opens.



Matching a Task to the Responsible Test Configuration

For any task reported in the **Quality Tasks** view, you can open the Test Configuration which caused that task to be reported. This is particularly useful if:

- You want to review or modify the setting that caused that task to be reported.
- You imported results from server execution and you want to know which Test Configuration was run when this task when generated.

To view the Test Configuration that caused a specific task to be reported:

- Right-click that task, then choose **View Test Configuration**.

This will open the appropriate Test Configuration and go directly to the Test Configuration controls that are related to this task being generated. For instance, if a static analysis task was selected, the **Static** tab will be opened and the corresponding rule will be highlighted.

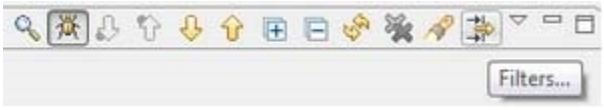
Customizing the Results Display

There are several ways to customize the results display to preferences and needs for reviewing quality tasks.

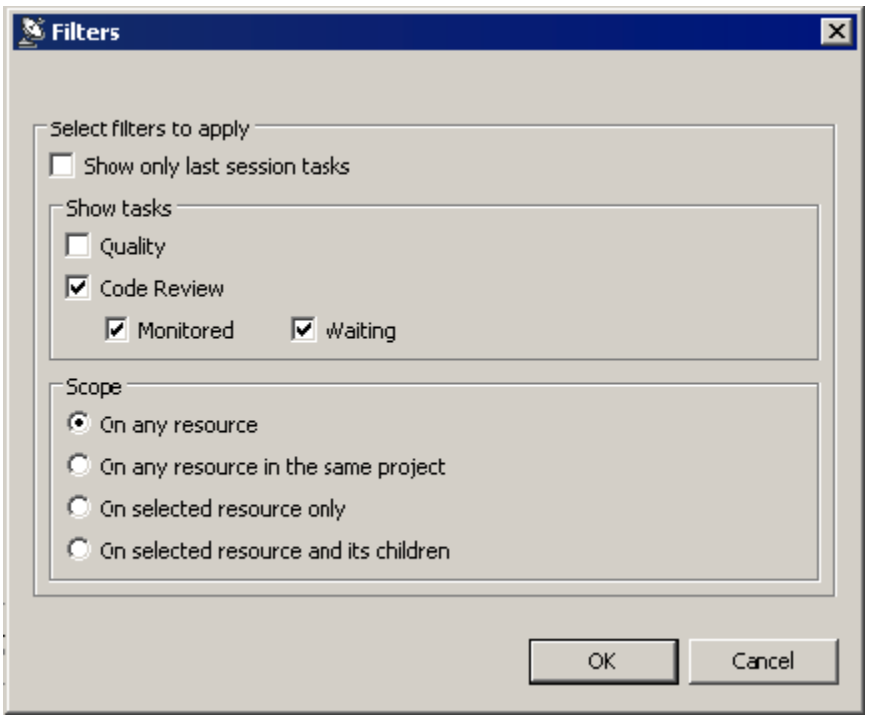
Customizing the Display Contents

To customize which tasks are displayed:

1. Click the **Filters** button in the toolbar.



2. In the dialog that opens, specify which content you want shown.



Changing the Display Format and Contents

You can change the **Quality Tasks** view's format and contents by:

- Selecting Layout Templates
- Customizing Layout Templates
- Adding New Layout Templates
- Changing Categories from the Quality Tasks View
- Clearing Selected Messages
- Clearing All Messages

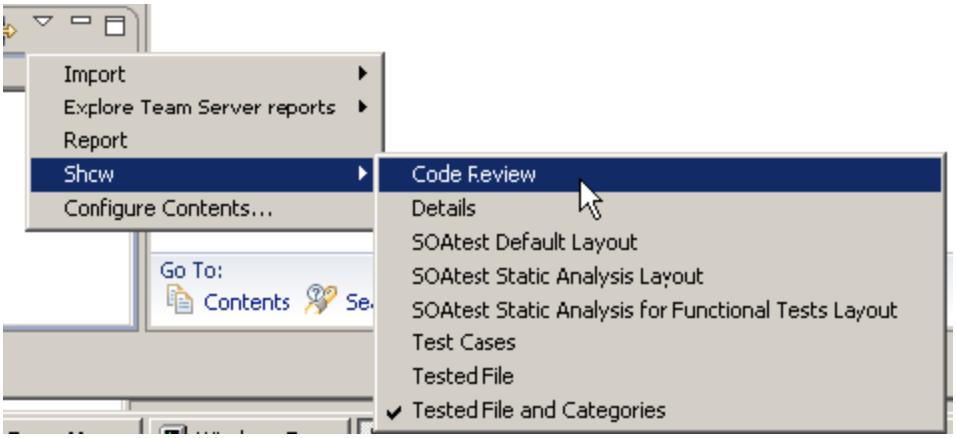
Selecting Layout Templates

There are several available layout templates:

- **Details:** Shows category, subcategory, task type, package or namespace, and location; organized by task type.
- **Test Cases:** Shows tests; organized by test names.
- **Tested Files:** Shows the location of the issue detected; organized by file names.
- **Tested File and Category:** Shows the category and location of the issue detected; organized by file names.

To select the layout best suited to your current goal:

1. Open the pull-down menu on the top rightof the **Quality Tasks** view.



2. Choose one of the available formats from the **Show** shortcut menu that opens.

Customizing Layout Templates

To customize one of these preconfigured layouts:

1. Open the pull-down menu on the top right of the **Quality Tasks** view.
2. Choose **Configure Contents**.
3. In the dialog that opens, specify how you want that layout configured. Note that **Comment** shows the comments that were entered upon source control commit.

Adding New Layout Templates

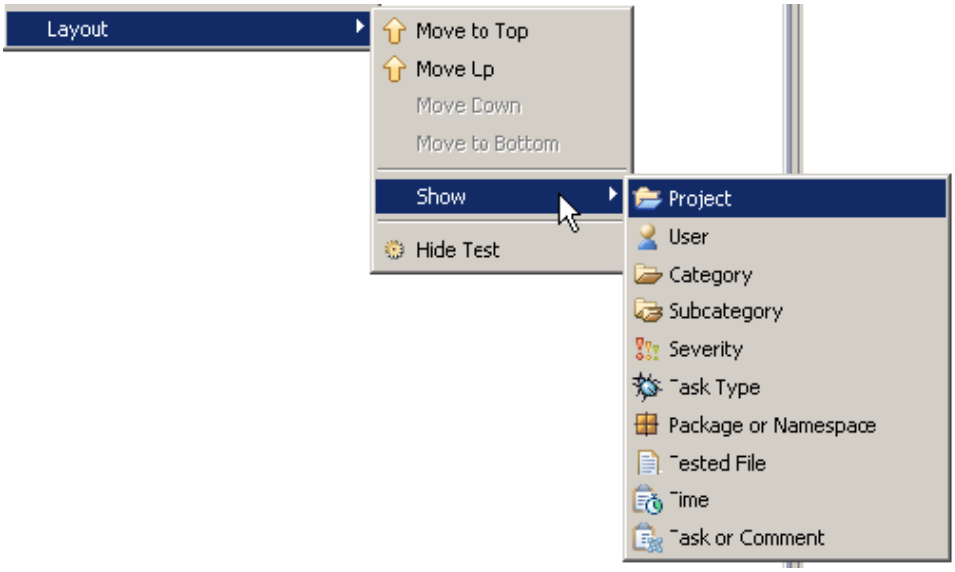
To add a new layout template:

1. Open the pull-down menu on the top right of the **Quality Tasks** view.
2. Choose **Configure Contents**.
3. Click the **New** button on the bottom left of the dialog that opens.
4. Select (and rename) the added template, then specify how you want that layout configured. Note that **Comment** shows the comments that were entered upon source control commit.

Changing Categories from the Quality Tasks View

To re-order, hide, and remove categories directly from the **Quality Tasks** view:

1. Right-click the item in the **Quality Tasks** view.
2. Choose from the available **Layout** menu options.



Clearing Messages

You might want to clear messages from the **Quality Tasks** view to help you focus on the findings that you are most interested in. For example, if you are fixing reported errors, you might want to clear each error message as you fix the related error. That way, the **Quality Tasks** view only displays the error messages for the errors that still need to be fixed.

Messages that you clear will only be removed temporarily. If the same findings are achieved during subsequent tests, the messages will be reported again.

You can clear individual messages, categories of messages represented in the **Quality Tasks** view, or all reported messages.

Clearing Selected Messages

To clear selected messages shown in the Quality Tasks view:

1. Select the message(s) or category of messages you want to delete. You can select multiple messages using **Shift + left click** or **Ctrl + left click**.
2. Right-click the message(s) you want to delete, then choose **Delete**.

The selected messages will be removed from the **Quality Tasks** view.

Clearing All Messages

To clear all messages found:

- Click the **Delete All** icon at the top of the **Quality Tasks** view.

Quality Tasks Categories

In the **Quality Tasks** view, C/C++test results are presented as a task list that helps you determine how to proceed in order to ensure the quality of your system.

Tasks are organized into the following categories:

- **Fix Static Analysis Violations:** This category contains static analysis violations that should be corrected or suppressed. It also includes problems identified by Flow Analysis.
- **Fix Unit Test Problems:** This category contains unit test problems—including functional test failures, unexpected exceptions, and timeouts—that need to be addressed.
- **Review Unit Test Outcomes:** This category contains unverified outcomes for test cases that were created during automated test case generation. Unverified outcomes are reported when C/C++test executes automatically-generated or user-defined test cases with postconditions that have not yet been converted to assertions The outcome might be the expected behavior, or it might indicate a problem. Further review and verification is required. If you determine that the outcome reflects the expected behavior, you verify it. If not, you specify the correct outcome.
- **Fix Runtime Error Detection Violations:** This category contains runtime errors detected when executing unit test cases or running the application.

Exploring and Resolving Errors

For details on how to explore and address errors reported for a specific type of analysis, see the following topics:

- **For Flow Analysis results:** See Reviewing Flow Analysis Results.
- **For static code analysis results:** See Reviewing Static Code Analysis Results.
- **For unit test results:** See Reviewing Test Execution Results.
- **For metrics results**: See Reviewing and Responding to Metrics Measurements.
- **For runtime error detection results:** See Runtime Error Detection.

Using Quick Fix (R) to Automate Error Resolution

The Quick Fix (R) feature can be used to automate actions commonly performed while reviewing and responding to unit test findings. Any finding that has a Quick Fix is marked with a yellow lightbulb icon. To automatically fix a problem marked with this icon, right-click the part of the message marked with the Quick Fix icon, then choose one of the available Quick Fix commands (marked with yellow lightbulb icons) from the shortcut menu.

Tip

- For details about using Quick Fix for unit test results, see Using Quick Fix (R) to Respond to Test Execution Findings**.**