

Reviewing Results

This topic explains how to view SOA test results and customize their presentation. Sections include:

- [Filtering Results](#)
- [Matching a Task to the Responsible Test Configuration](#)
- [Customizing the Results Display](#)
- [Clearing Messages](#)
- [Generating Reports](#)
- [Reviewing SOA test Results](#)

Filtering Results

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

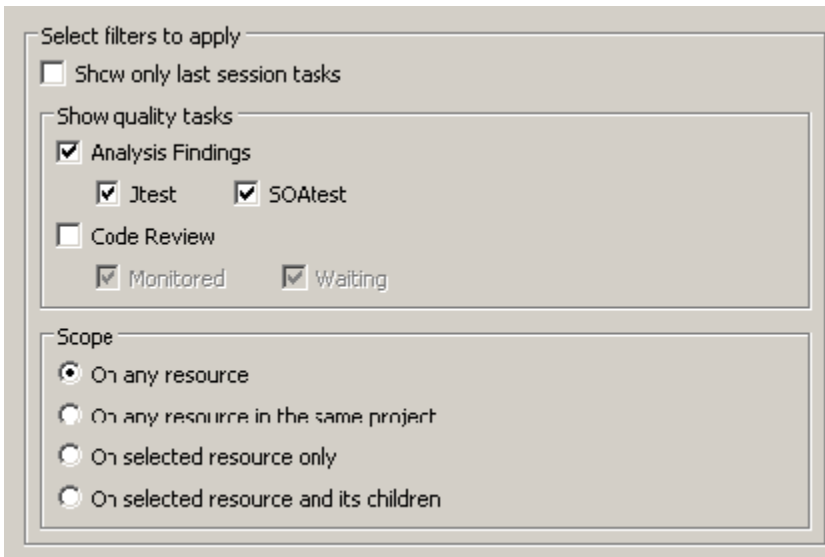
If you prefer to see only results from the last test session or from selected resources, you can filter results.

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 was 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.

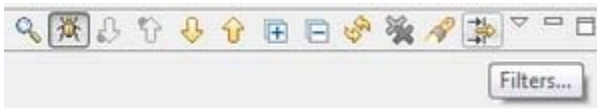
Customizing the Results Display

There are several ways to customize the results display to preferences and needs for reviewing quality tasks and/or peer code review tasks.

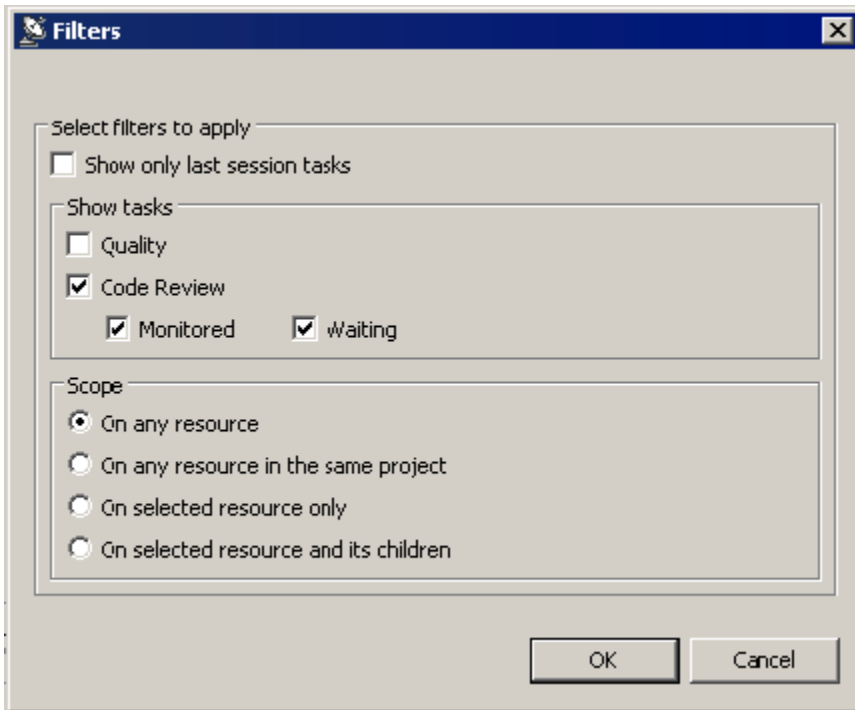
Customizing the Display Contents

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](#)

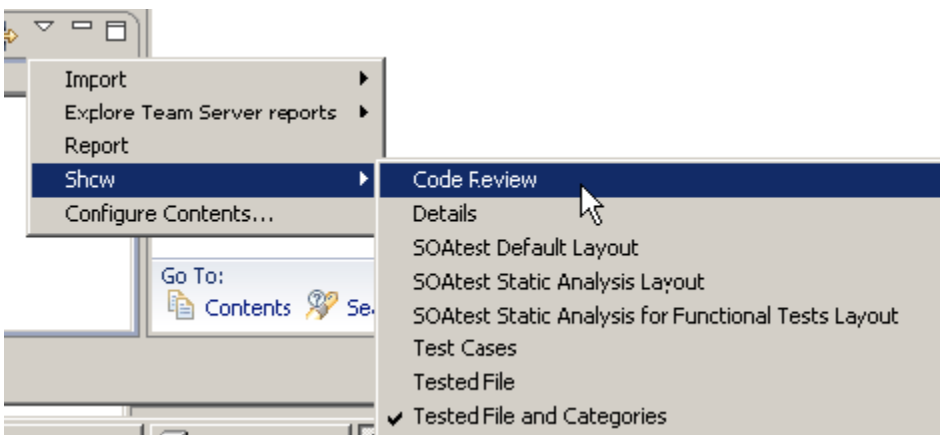
Selecting Layout Templates

There are several available layout templates:

- **Code Review:** For peer code review tasks.
- **Details:** Shows category, subcategory, task type, package or namespace, and location; organized by task type.
- **SOAtest Default Layout:** For functional testing.
- **SOAtest Static Analysis Layout:** For running static analysis against source code (e.g., from the Scanning perspective).
- **SOAtest Static Analysis for Functional Tests Layout:** For running static analysis by executing a test suite (e.g., a test suite that contains a Browser Testing tool or a Scanning tool).
- **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 right of 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

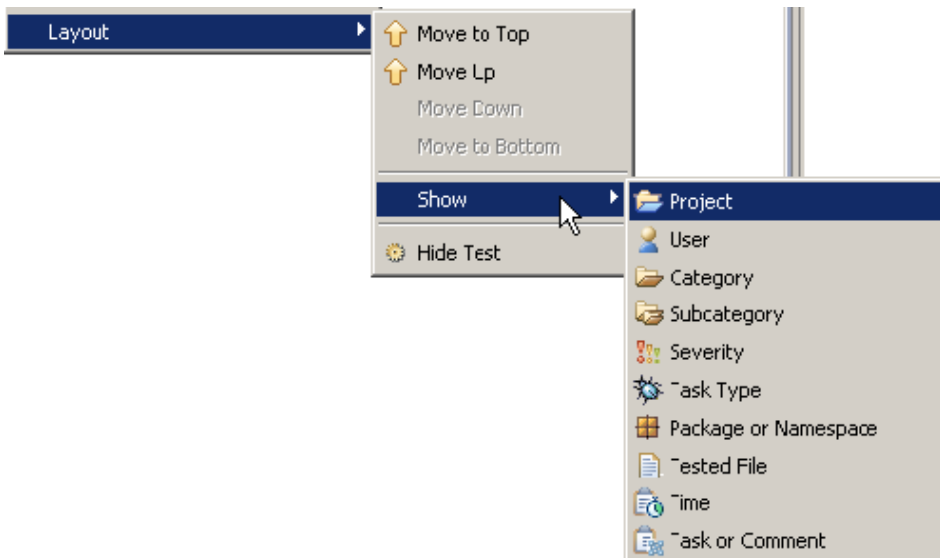
To add a new layout template:

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 **Comments** 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.

Generating Reports

This topic explains how to generate HTML, PDF, or custom XSL reports for tests that you run from the GUI or command line.

Sections include:

- [From the GUI](#)
- [From the Command Line](#)



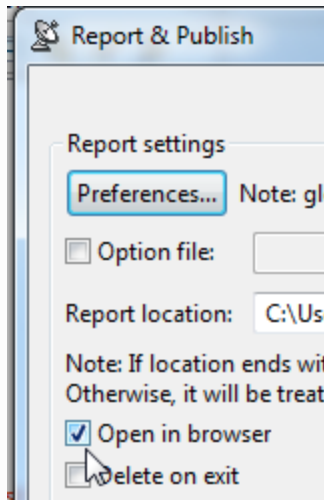
Understanding Report Categories and Contents

Report categories and contents vary from product to product. For details on the reports generated by a specific Parasoft product, see that product's user's guide.

Troubleshooting Issues with Eclipse Report Display

Due a known bug in Eclipse, Eclipse may crash when displaying reports. If you are experiencing this issue, here are some workarounds:

- Update to the latest version of Eclipse (the issue has been fixed).
- Install a fixed XULRunner plugin to Eclipse.
- Use an EPF (Eclipse Preferences File) to configure Eclipse to use an external browser; you need to change the "browser-choice" option—e.g.: `/instance/org.eclipse.ui.browser/browser-choice=1`
- (If the problem occurs only at the end of the testing process) Disable the **Open in browser** option in the Report and Publish dialog.



From the GUI

Generating the Report

To generate a report immediately after a test completes:

1. After the test has completed, click the **Generate Report** button that is available in the Test Progress panel's toolbar.

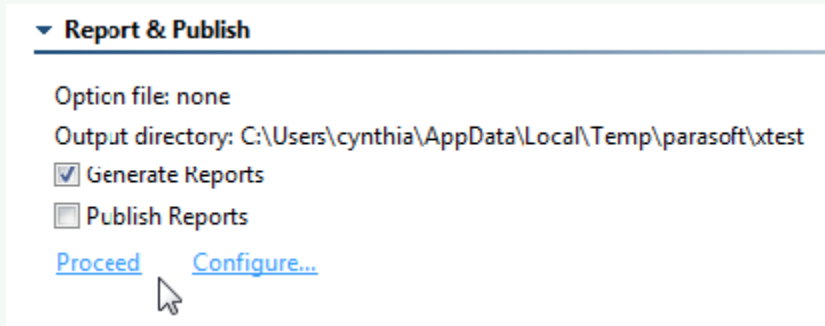


2. Complete the Report dialog that opens. The Report dialog allows you to specify:
 - **Preferences:** Report preferences (by clicking the Preferences button and specifying settings as explained in [Configuring Report Settings](#)).
 - **Options file:** Any localsettings/options that specify reporting settings you want to use. These will override settings specified in the GUI's Preferences panel). For details on configuring reports through localsettings, see [Configuring Localsettings](#).
 - **Report location:** The location of the report file (by default, reports are created at `<user_home_dir>\Local Settings\Temp\parasoft\xtest`).
 - **Open in browser:** Whether the file is automatically opened in a browser.
 - **Delete on exit:** Whether the report is deleted upon exit.
 - **Generate reports:** Whether a report should be created.
 - **Publish reports:** Whether the report should be uploaded to the Team Server (Server Edition only; requires Team Server).
 - **Publish code reviews:** Whether code review tasks/results should be uploaded to the Team Server (any edition; requires Team Server).
3. Click **OK**. The report will then open. For details on the format and contents, see the appropriate product's user's guide ([\[product_name\] User's Guide](#)> Setup and Testing Fundamentals> Reviewing Results> Understanding Reports).



Tip

You can also generate and configure reports from the bottom of the Test Progress panel.



Proceed immediately generates the report using the existing options.

Configure allows you to review and modify reporting options before generating the report.

Uploading the Report to Team Server

To upload the report to Team Server (Server Edition only):

- Follow the above procedure, but be sure to enable **Publish: Reports** before clicking **OK**.



How do I aggregate or separate results from multiple test runs?

Team Server uses the following criteria to identify unique reports:

- Host name
- User name
- Session tag
- Day - each day, only the last test run is used in trend graph

If your team performs multiple cli runs per day—and you want all reports included on Team Server—you need to use a different session tag for each run. You can do this in the Test Configuration's Common tab (using the **Override Session Tag** option).

Traffic Viewer SOAtest Report [02/18/14 15:07:13]

User configuration: Example Configuration
02/18/14 15:07:13

FUNCTIONAL TESTS

Test Suite Summary				
Name	Failed Tests	Successful Tests	Total Tests	Success %
Total	41	113	154	73%
SOAtest Tutorial/Calculator.tst	0	4	4	100%
SOAtest Tutorial/CheckTitleAlphabetization.tst	0	3	3	100%
SOAtest Tutorial/ClientTesterMethodExample.tst	1	0	1	0%
SOAtest Tutorial/CustomSOAPHeader.tst	0	1	1	100%
SOAtest Tutorial/DBToolParameterization.tst	2	0	2	0%
SOAtest Tutorial/DynamicDataSources.tst	0	2	2	100%
SOAtest Tutorial/DynamicTestVariableLogic.tst	0	3	3	100%
SOAtest Tutorial/EDI.tst	0	3	3	100%
SOAtest Tutorial/EndToEndSOAP.tst	3	0	3	0%
SOAtest Tutorial/EndToEndWeb.tst	3	0	3	0%
SOAtest Tutorial/Equation.tst	0	12	12	100%
SOAtest Tutorial/FixedLength.tst	2	0	2	0%
SOAtest Tutorial/FullExeTime.tst	0	1	1	100%
SOAtest Tutorial/HPQualityCenter.tst	4	4	8	50%
SOAtest Tutorial/ModifyDataBankValue.tst	0	3	3	100%
SOAtest Tutorial/MultipleDataSourceWorkAround.tst	0	30	30	100%
SOAtest Tutorial/RandomSelectResponseID.tst	0	2	2	100%
SOAtest Tutorial/RuleExamples.tst	23	0	23	0%
SOAtest Tutorial/SignAndVerify.tst	1	0	1	0%
SOAtest Tutorial/SingleTransformerMultiDiff.tst	0	1	1	100%
SOAtest Tutorial/SOAtestTutorial.tst	1	0	1	0%
SOAtest Tutorial/StockQuoteClient.tst	1	0	1	0%
SOAtest Tutorial/StoreTests.tst	0	21	21	100%
SOAtest Tutorial/TestFlowLogicAndRegression.tst	0	21	21	100%
SOAtest Tutorial/UseOfSecurityAssertion.tst	0	2	2	100%

Tasks by Author [Back to Top](#)

Author	Tasks	Failed Tests
cynthia	65	41

From the Command Line

To generate a report of command line test results, use the `-report %REPORT_FILE%` option with your cli command. To upload the report to Team Server, also use the `-publish` option.

Command-line interface testing details are explained in each Parasoft Test family product's user's guide ([product_name] User's Guide> Setup and Testing Fundamentals> Running Tests and Analysis> Testing from the Command Line Interface). This topic also discusses how to setup and configure email notifications.



Configuring Detailed Reports

If you want your report to show details about every test that ran, including tests that did not fail, configure reporting preferences with the **Only tests** that failed and **Only top-level test suites** options disabled.

Reviewing SOAtest Results

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

Functional Testing

Functional test results are organized by test suite. See [Reviewing Functional Test Results](#) for details.

Static Analysis

Static analysis results should be reviewed in one of the layouts designed especially for static analysis. For details on enabling these layouts and reviewing static analysis results, see [Reviewing Static Analysis Results](#).