Configuring Report Settings

You can configure the report settings in the GUI or from the command line interface (using localsettings). In addition to generating local reports, you can also send reports to Parasoft Development Testing Platform (DTP) or to Parasoft Project Center (legacy). Sending reports to DTP enables you to apply sophisticated analytics to better understand the risk associated with the application under test and your overall development and testing processes. Sending reports to Project Center enables centralized task management.

Sections include:

From the GUI

You can specify the report settings for all tests—whether they are run from the command-line interface or the UI–using the GUI controls. Before configuring report settings, you should review the settings on the following preference pages to ensure that task authorship is being calculated correctly, that results are being sent to the appropriate system, that the correct email host is used, etc.:

- Configuring Email Settings
- Connecting to Parasoft Development Testing Platform
- Connecting to Parasoft Team Server
- Connecting to Project Center
- Configuring Task Assignment and Code Authorship Settings
- Connecting to Your Source Control Repository
- Licensing

The settings specified in the UI can be fully or partially overwritten by those specified via localsettings.

To specify reporting settings from the GUI:

1. Choose **Parasoft> Preferences**. The Preferences dialog will open.
2. Select **Parasoft> Reports**.
3. Specify the appropriate settings. Available settings are described in Report Configuration Settings below.
4. If you have not already configured e-mail settings (sender address, host name, etc.) in either the GUI or from the command line, do so now in **Parasoft> E-mail.**
5. Select **Parasoft> Reports> E-mail Notifications**.
6. Specify the appropriate e-mail notification settings. Available settings include:
    - **Send reports by e-mail:** Specifies whether reports are sent via e-mail.
    - **E-mail subject:** Specifies the subject for e-mails containing reports.Specifies the subject line of the emails sent. The default subject line is "[Product Name] Report." For example, if you want to change the subject line to "Functional Testing Report for Project A", you would enter `Functional Report for Project A`
    - **Send manager reports to:** Specifies where to send manager reports. You can enter a single address or a semicolon-separated list.
    - **Send reports without tasks:** Specifies whether reports are sent when zero tasks are reported. You can enter a single address or a semicolon-separated list.
    - **Send developer reports to:** Specifies where to send developer reports. You can enter a single address or a semicolon-separated list.
    - **Send 'Unknown' developer reports to:** Specifies where to send developer reports for tasks assigned to "unknown" (tasks that could not be traced back to a specific developer). These reports can be sent to only one user / address. A semicolon-separated list of addresses is not valid.

Report Configuration Settings

Reports Content

Setting	Description
Detailed report for developers	Determines whether customized, detailed reports are generated for each team member (in addition to a summary report for managers). These reports contain only the tasks assigned to that specific team member.
Overview of tasks by authors	Determines whether the report includes an overview of the number and type of tasks assigned to each developer.

Overview of checked files and executed tests	Specifies whether the report provides details about all checked files and executed tests. For static analysis, this results in a list of all the files that were checked. For each file, it lists the number of rule violations and the number of suppressed violations. If the file has a violation, it also lists the line number, rule name, and rule ID for that violation. For test execution, this results in a list of all executed test cases and their outcomes (pass or fail). For each test suite, it lists the total number of test cases and the number of passed test cases. If a task is reported for a test case, additional details (stack trace, outcome, etc.) are presented. For code review, this results in a list of all pending issues with messages between author - reviewer - monitor (filtered by session tag by default).
Task details	Determines whether the report includes details about all of the reported tasks.
Test case details	Determines whether the report includes details about all executed test cases.
Suppression details	Specifies whether the report lists suppressed messages.
Requirement/defect details	Specifies whether the report shows requirements, defects, tasks, and feature requests that are associated with a test. **Only top-level test suites** (below) must be disabled for this setting to take effect.
Only top-level test suites	Determines whether the Test Suite Summary report section only lists the .tst files (with this option enabled) or displays a tree-like view of the individual tests in each .tst file (with this option disabled).
Active static analysis rules	Determines whether the report lists the static analysis rules that were enabled for the test.
Only tests that failed	Specifies whether the report lists only tests that failed.
Generate formatted reports in command-line mode	Determines whether formatted reports are generated for tests run in command line mode.
Cutoff date for graphs	Specifies the start date for trend graphs that track different categories of tasks over a period of time.

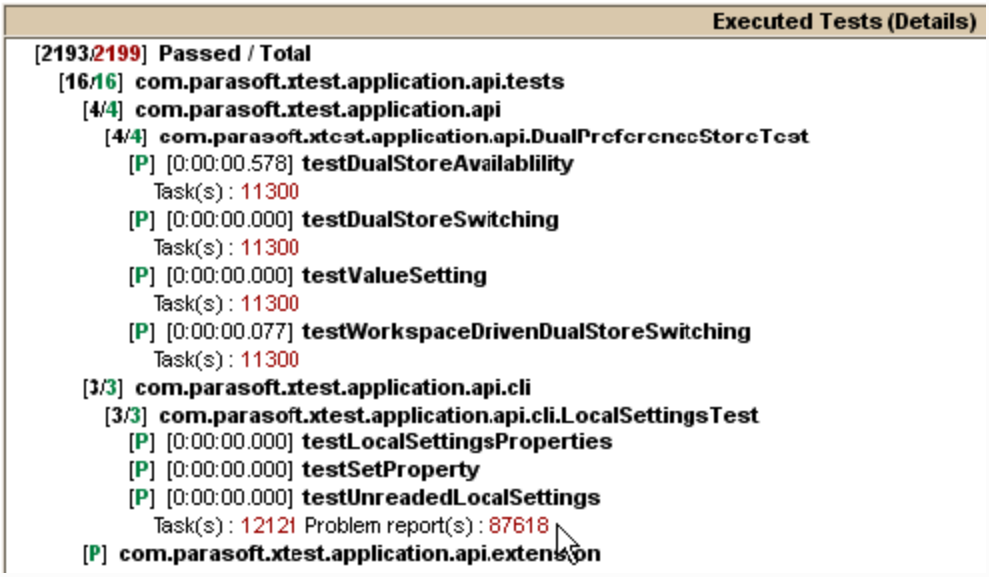
HTML Report Hyperlinks for Requirement/Defect Details

If you are specifying a URL for the artifact (e.g., as a SOAtest test property), that URL will be available as a hyperlink in HTML reports.

If want to create hyperlinks and you are specifying artifact associations using only the artifact ID as specified in Indicating Code and Test Correlations, you can use local properties to specify how the ID maps to a URL.

For example:

- report.assoc.url.pr=https://bugzilla.parasoft.com/show_bug.cgi?id=[%ID%]
- report.assoc.url.jira=https://jira.parasoft.com/browse/[%ID%]
- report.assoc.url.fr=https://bugzilla.parasoft.com/show_bug.cgi?id=[%ID%]
- report.assoc.url.task=http://concerto.parasoft.com.pl:8080/grs/jsf/planning/task/edit_task.jsf?entityId=[%ID%]



Report Format

Setting	Description
Format	Specifies the desired report format: • HTML - includes XML source data file • PDF • XML SATE - format specific to NIST SAMATE Static Analysis Tool Exposition • XSL Custom • xUnit - a format that is like JUnit, with the addition of nested test suites. This format is supported by Jenkins, Hudson, Eclipse JUnit view, etc.. It is applicable only with **Overview of checked files and executed tests** enabled. This format is supported for language tools: dotTEST, C++test, and Jtest.
XSL file	If you chose custom XSL as the report format, specify the path to the XSL file that defines your custom format.
Report file extension	If you want to use a file extension other than the default .html extension, specify that extension here.
Generate additional archive with entire report content	Enable this option to generate an additional compressed archive (.zip) file in the specified report location. The ZIP file contains all the files generated to build the report. This option can generate an archive for any report format (e.g., HTML, CSV, PDF, etc.). By generating an archive, you can also perform custom transformations of the report because all of the elements are generated to the specified destination folder.

Advanced Settings

Setting	Description
Add absolute file paths to XML data	Specifies whether absolute file paths are added to XML data. This needs to be enabled on the Server installation if you want to relocate tasks upon import to desktop installations (as described in Relocating Tasks During Import: Requirements and Limitations).

Session tag	Session tags are unique identifiers for test runs and are used to distinguish specific runs from similar runs in DTP. The results from each of the team's regular test runs should have a unique tag. Variables can be used as described in Using Preference Settings for Command-Line Execution. For instance, if your team runs static analysis, unit testing, and code review, you might use the session tags `Static`, `Execution`, and `Code Review`. Or, you might use variables such as `session.tag=${config_name}` or `session.tag=${analysis_type}`

From localsettings

See Reporting Settings for details.

Support for Custom Report Formats

You can customize locally-generated and emailed reports by building a custom XSL transformer that specifies how you want the results formatted. For instance, you could use a custom transformer to map data into the format you need to demonstrate compliance to an internal security policy.

To specify a custom report format, you need to:

1. Create an XSL file that specifies how you want to transform the XML data.
2. In the Preferences panel's Reports page, specify the location of this XSL file, as well as the extension that should be assigned to the resulting file.
    - Alternatively, you can specify this information in a localsettings file via the settings `(results.)report.custom.extension and (results.)report.custom.xsl.file.`

Available Parameters

The following parameters can be used in custom XSL files:

Parameter	Notes
report_type=disk_report|email_report	Determines if the generated report will be sent by email or saved on the local disk drive.
test_params	The command line that was used to start the product that generated this report. Example: jtestcli: -config team://xtest-static.properties -localsettings /home/nightly/localsettings.properties -publish -report /home/nightly /reports/report.html -resource myproject -concerto.autoconfig develop-ment@myserver.com:8080
test_config_name	The name of the Test Configuration that was executed to produce this report. Example: Parasoft's Recommended Rules
output_dir=[dir]	The directory where the report is created. This can be used to generate developer reports. Example: <xsl:value-of select="concat ($output_dir,$dev_reports_prefix,$authid,'.csv')"/>
rules_dir_path=[path]	The directory where rules documentation is saved. This can be used to generate rule popups/links. Example: <xsl:value-of select="concat('javascript:openWin ('',$qt,$rules_dir_path,$id,'.html',$qt,')')"/>
suppr_msgs=true|false	The value of the "Suppressions details" option. See Report Configuration Settings for details.
dev_errors=true|false	The value of the "Task details" option. See Report Configuration Settings for details.

dev_reports=true|false	The value of the "Detailed report for developers" option. See Report Configuration Settings for details.
show_active_rules=true|false	The value of the "Active static analysis rules" option. See Report Configuration Settings for details.
associations=true|false	The value of the "Requirement/defect details" option. See Report Configuration Settings for details.
dev_reports_prefix=[prefix]	The prefix used to name developer reports. Example: <xsl:value-of select="concat ($output_dir,$dev_reports_prefix,$authid,'.csv')"/>
exec_cvg_details=true|false	The value of the "Detailed report for" option. See Report Configuration Settings for details.
attachments=true|false	The value of the localsettings key report.mail.attachments. See Configuring Localsettings for details.
test_suites_only=true|false	The value of the localsettings key report.test_suites_only. See Configuring Localsettings for details.
test_cases_details=true|false	The value of the "Test case details" option. See Report Configuration Settings for details.
failed_tests_only=true|false	The value of localsettings key report.failed_tests_only. See Configuring Localsettings for details.
authors_details=true|false	The value of the "Overview of tasks by authors" option. See Report Configuration Settings for details.
contexts_details=true|false	The value of the "Overview of checked files and executed tests" option. See Report Configuration Settings for details.

Samples

For additional guidance, see the following files:

Local Reports

- XML Schema: reports.xsd
- Sample XML with a variety of results: rep_example.xml
  - Note that this report is used by all transformations below
- Sample transformation to a plain text CSV file
  - XSL file: csv.xsl
  - result: csv.txt
- Sample transformation to an HTML table with a violations list
  - XSL file: html_table.xsl
  - result: html_table.html
- Sample transformation to an HTML table with author/violations statistics
  - XSL file: stats_table.xsl
  - result: stats_table.html
- Sample developer report transformation to a CSV file
  - XSL file: csv_dev.xsl
  - result: csv_dev.csv
- Sample developer report transformation to an HTML file
  - XSL file: html_dev.xsl
  - result: html_dev.html