

# Generating Reports

You can generate HTML, PDF, or custom XSL reports. In this section:

- [Generating Reports from the GUI](#)
  - [Generating the Test Execution Details Report](#)
  - [Uploading the Report to DTP](#)
  - [Uploading the Report to Team Server](#)
- [Generating Reports from the Command Line](#)

## Troubleshooting Issues with Eclipse Report Display

A known bug in older versions of Eclipse may cause the IDE to crash when displaying reports. If you are experiencing this issue, here are some workarounds:

- Update to the latest version of Eclipse.
- 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.

## Generating Reports from the GUI

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



2. Complete the **Report** dialog that opens. The **Report** dialog allows you to configure the following options:

- **Preferences:** See [Configuring Reporting Settings](#).
- **Options file:** If you want to override the report settings specified in the GUI's **Preferences** panel, you can specify a .properties file configured with your custom report settings. Also see [Configuring LocalSettings](#)
- **Report location:** Specify where to output the report file. By default, reports are created in the `<user_home_dir>\Local Settings\Temp\parasoft\xtest` directory.
- **Open in browser:** Enable/disable opening the report file in the browser automatically.
- **Delete on exit:** Enable/disable deleting the report upon exiting C++test.
- **Generate reports:** Enable/disable generating a report. You cannot proceed if this setting is disabled.
- **Publish reports to Team Server:** Enable this option if C++test is connected to DTP and you are using Team Server to manage shared reports and configurations.

3. Click **OK**. The report will then open. For details on the format and contents, see [Understanding Reports](#).

## Tip

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

### **Report & Publish**

Option file: none

Report location: C:\Users\annstu\AppData\Local\Temp\parasoft\xtest

Generate Reports

Publish Reports to Team Server

Publishing to DTP can be enabled in Preferences > Development Testing Platform > Enable reporting results to DT

[Proceed](#)  [Configure...](#)

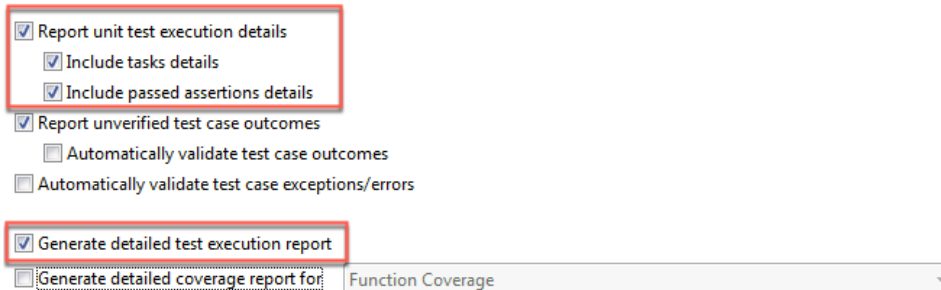
**Proceed** immediately generates the report using the existing options.

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

## Generating the Test Execution Details Report

You can generate an additional Test Execution Details report linked to your regular report (see [Understanding Reports](#) for details). To generate this report, configure the following settings in the test configuration before executing it:

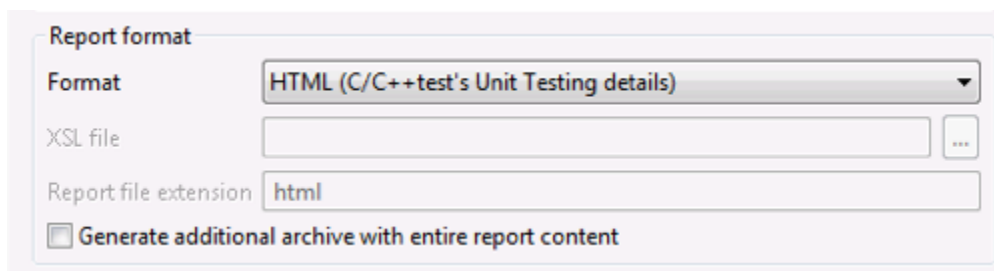
1. Choose **Parasoft> Test Configurations...** and select a test configuration.
2. Open the **Execution> Runtime** tab and enable the following options:
  - **Report unit test execution details**
  - **Include tasks details**
  - **Include passed assertion details**
  - **Generate detailed test execution report**



The screenshot shows the 'Runtime' tab of the test configuration dialog. The following options are checked and highlighted with red boxes:

- Report unit test execution details
  - Include tasks details
  - Include passed assertions details
- Report unverified test case outcomes
  - Automatically validate test case outcomes
  - Automatically validate test case exceptions/errors
- Generate detailed test execution report
- Generate detailed coverage report for: Function Coverage

3. (Optional) Enable the **Generate detailed coverage report for** option and choose a coverage metric.
4. Apply changes and close the test configuration.
5. Choose **Parasoft> Preferences... > Reports**.
6. In the **Report Format** section, choose **HTML (C/C++test's Unit Testing details)** from the drop-down menu.



The screenshot shows the 'Reports' section of the Parasoft Preferences dialog. The 'Report format' section is expanded, showing the following settings:

- Format:** HTML (C/C++test's Unit Testing details)
- XSL file:** (empty field with a browse button)
- Report file extension:** html
- Generate additional archive with entire report content

Click **Generate Report** after the test completes (see [Generating the Report](#)). The link to the **Test Execution Details** report will be available in the **Test Execution Details** column included in the **Additional Reports** section at the bottom of main report (see [Understanding Reports](#)).

You can also generate the Test Execution Details report in XML format. This enables you to apply additional post-processing workflows to the test execution results. The XML file is available in the same location as the original Test Execution Details HTML file, for example:

```
<REPORT_DIR>/<EXECUTION_DIR>/report_TestSuiteFoo_cpp.xml
```

```
<REPORT_DIR>/<EXECUTION_DIR>/report_TestSuiteFoo_cpp.html
```

In addition, XML Schema for Test Execution Details XML file is available:

```
<CPPTEST_INSTALL_DIR>/manuals/examples/ut_details_report_schema.xsd
```

## Uploading the Report to DTP

To upload the report to DTP:

- Ensure that C++test is connected with a DTP Server (See [Connecting to DTP](#)).
- Go to **Parasoft> Preferences... > DTP** and enable the **Enable reporting results to DTP** option in the **Reports** area.

Publishing reports to DTP requires the "DTP Publish" or "Automaton" license feature to be enabled.

## 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 to Team Server** before clicking **OK**.

**i** 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).

## Generating Reports 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, use the `-publishteamsserver` option.

To upload the report to DTP, use the `-publish` option.

Command-line interface testing details are explained [Testing from the Command Line Interface](#). This topic also discusses how to setup and configure email notifications.