

Troubleshooting and FAQs

This topic helps you troubleshoot problems you might experience with C++test.

- [What Do I Do If C++test Runs Out of Memory?](#)
- [How do I analyze header files/what files are analyzed?](#)
- [How can I create a report with a list of active coding standard rules?](#)
- [How can I modify the verbosity level of the C++test Console?](#)
- [How can I import C++test 6.x Test Configurations?](#)
- [Why is C++test unable to generate test cases in Visual Studio 2017?](#)
- [Why are some Parasoft options not working in the Solution Explorer context menu in Visual Studio 2017?](#)
- [What if C++test does not launch in the command line mode?](#)
- [Why are suppressions of some rules no longer available on DTP after C/C++test was upgraded to a newer version?](#)
- [If I'm having problems, what information should I send to the C++test support team?](#)

What Do I Do If C++test Runs Out of Memory?

You can customize the amount of allocated memory with the CPPTTEST_ENGINE_EXTRA_ARGS option by specifying the value of the -Xmx setting. This may be particularly useful for when you perform static analysis or report results to DTP. See [Configuring Advanced Options](#) for details.

How do I analyze header files/what files are analyzed?

C++test analyzes C/C++ source files directly and header files indirectly. Based on the current selection, C++test will analyze all C/C++ source files and report violations for all source and header files from the selection (only for header files included by the source files).

For example:

- if the project root is selected, then all source files (and header files indirectly) will be analyzed.
- if a single source file is selected, then only that file will be analyzed (no header files will be analyzed).
- if single source file and a single header file is selected, then the source file (and header file if it's included by the source) will be analyzed.
- if only a header file is selected, then C++test will skip the analysis (header files are not analyzed directly).

How can I create a report with a list of active coding standard rules?

When generating a report, specify a report configuration file that contains the following entry:

```
results.report.active_rules=true
```

How can I modify the verbosity level of the C++test Console?

Choose **Parasoft > Preferences**, select **Console**, then select the desired verbosity level (High, Normal, Low).

	High Verbosity	Normal Verbosity	Low Verbosity
Basic info about the current step's name and status (done, failed, up-to-date)	Yes	Yes	Yes
Errors	Yes	Yes	Yes
Warnings	Yes	No	No
Command Lines	Yes	Yes	No
Violations printed out during static analysis and unit testing execution	Yes, full-format	Yes, short-format	No

How can I import C++test 6.x Test Configurations?

See [Migrating test assets from C++test 6.x](#).

Why is C++test unable to generate test cases in Visual Studio 2017?

C++test cannot generate test cases when project caching is enabled in Visual Studio 2017. To ensure that test cases can be generated, disable project caching:

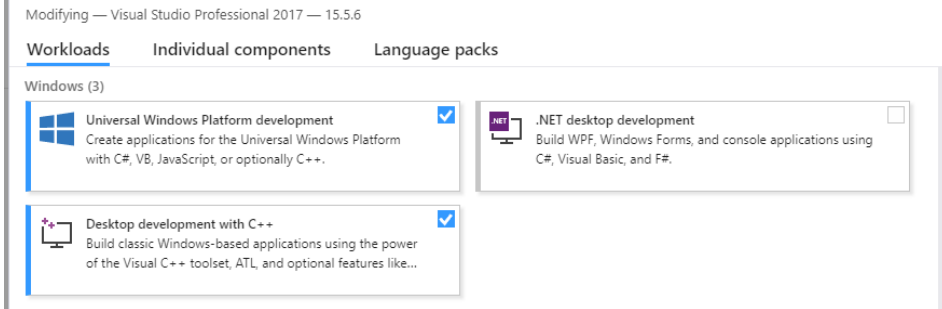
1. In the Visual Studio 2017 main menu, go to **Tools > Options**.
2. Expand **Projects and Solutions** and select **VC++ Project Settings**.
3. Set the **Enable Project Caching** option to **No** to disable project caching.

Why are some Parasoft options not working in the Solution Explorer context menu in Visual Studio 2017?

To ensure that all Parasoft options are working in the Solution Explorer context menu, the following Visual Studio Workloads must be installed with your Visual Studio 2017:

- *Desktop development with C++*
- *Universal Windows Platform development* or *.NET desktop development*

1. Run the Visual Studio Installer.
2. Select the appropriate Workloads when prompted.



What if C++test does not launch in the command line mode?

Installing or updating extensions or other Visual Studio components may corrupt Visual Studio's Component Model Cache, which can prevent C++test from launching. To clear the Component Model Cache, try to run `cpptestcli` with the `-clearcmc` option.

Why are suppressions of some rules no longer available on DTP after C /C++test was upgraded to a newer version?

Suppressions associated with rules whose messages changed between releases may not be available on DTP and the rules must be re-suppressed.

i You can restore legacy messages for BD category rules in version 10.4.1 and later by configuring the following advanced option (see [Configuring Advanced Options](#)):

```
CPPTTEST_ENGINE_EXTRA_ARGS="-property flowanalysis.legacy.messages.for.<rule_ID>=true"
```

For example:

```
CPPTTEST_ENGINE_EXTRA_ARGS="-property flowanalysis.legacy.messages.for.BD.PB.ARRAY=true -property flowanalysis.legacy.messages.for.BD.PB.ZERO=true"
```

To successfully restore the legacy messages, ensure that you remove the `.cpptest` folder that contains the local C/C++test cache data from following location: `<workspace_location>/.cpptest/.dtpengine/.cpptest`.

If I'm having problems, what information should I send to the C++test support team?

See [Contacting Parasoft Technical Support](#).