Troubleshooting and FAQs

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

- What Do I Do If C/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 or later?
- Why do I get notified that Parasoft Plugin or its components are slowing down Visual Studio 2017 or 2019 on startup?
- What if Visual Studio reports syntax errors on code that is correct in test suites generated with C/C++test?
- What if C++test does not launch in the command line mode?
- What can I do if C/C++test reports a timeout during static analysis?
- Why are suppressions of some rules no longer available on DTP after C/C++test was upgraded to a newer version?
- How can I prevent my machine ID from floating?
- What if some characters fail to be properly displayed in C/C++test reports?
- If I'm having problems, what information should I send to the C++test support team?

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

You cancustomize the amount of allocated memory with the CPPTEST_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:
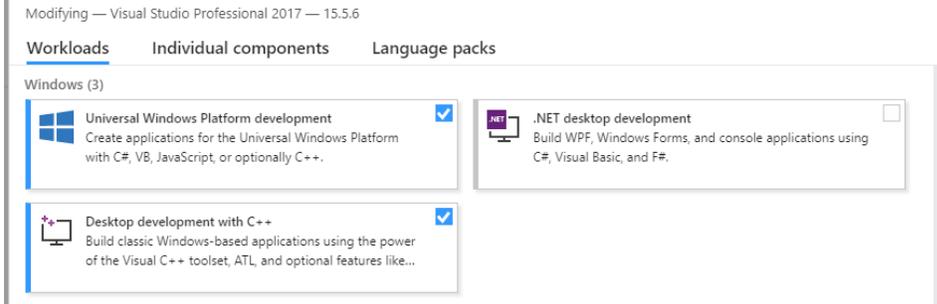
1. In the Visual Studio 2017 main menu, go to **Tools> Options**.
2. Expand **Projects and Solutions** and select **VC++ Project Settings**.
3. Set the **Enable Project Caching** option to **No** to disable project caching.

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

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 or later:

- *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.

Why do I get notified that Parasoft Plugin or its components are slowing down Visual Studio 2017 or 2019 on startup?

In Visual Studio 2017 and 2019, performance management monitors and manages extensions that can affect startup time. As a result, a pop-up alert may appear when Parasoft Plugin is installed to indicate that Visual Studio startup time has increased. You can click the 'Manage performance' link in the pop-up window to review the startup times and disable displaying the alert.

What if Visual Studio reports syntax errors on code that is correct in test suites generated with C/C++test?

Visual Studio may mark correct code elements in test suites with a red underline to indicate potential errors when IntelliSense, Visual Studio's native code completion, is unable to recognize C/C++test-specific syntax. To prevent this, you can do one of the following:

- Configure IntelliSense to stop reporting potential syntax errors. Note that this will globally disable error recognition–in test suites and in all other files.
- Wrap the code in your test suite (s) with the following directives:

```
#ifndef __INTELLISENSE__
//this code block will not be analyzed by IntelliSense
#endif
```

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.

What can I do if C/C++test reports a timeout during static analysis?

To ensure the analysis completes, you can do one of the following:

- Increase the timeout limit by configuring the following advanced option:
  `cpptest.analyzer.source.timeout=<TIMEOUT_IN_SECONDS>`
  See Configuring Advanced Options .
- Disable the most time-consuming rules or metrics in your test configuration. The information about the times is included in the timeout message displayed on the console and in the report.
- Contact Parasoft Support.

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.

ⓘ 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):

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

For example:

```
CPPTEST_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`.

How can I prevent my machine ID from floating?

Changes in the network environment may affect the interface that is used to compute your machine ID and result in machine ID instability. You can use the PARASOFT_SUPPORT_NET_INTERFACES environment variable to specify a stable interface and prevent the machine ID from floating.

1. Set up the **PARASOFT_SUPPORT_NET_INTERFACES** environment variable.
2. Set the variable value to a stable Ethernet network interface. Do not use virtual, temporary or loopback interfaces.
   - On Windows: Set the value to the MAC address of your network card. You can use the `ipconfig -all` command to obtain the address. For example:

   ```
   SET PARASOFT_SUPPORT_NET_INTERFACES=00-10-D9-27-AC-85
   ```

   - On Linux: Set the value to one of the network interfaces from the "inet" or "inet6" family. You can use the `ifconfig` command to obtain the list of available interfaces. For example:

   ```
   export PARASOFT_SUPPORT_NET_INTERFACES=eth1
   ```

If the problem persists, you can obtain diagnostic information by setting up the environment variable **PARASOFT_DEBUG_NET_INTERFACES** and setting its value to true. This will print to the standard output the checking procedure that can be shared with technical support, as well as the interface that is used to compute your machine ID. The interface will be marked with the [SELECTED] prefix.

What if some characters fail to be properly displayed in C/C++test reports?

Reports generated by Parasoft products require a sans-serif font to be available in your environment. If your report fails to correctly display some characters, such as national characters, ensure that a sans-serif font is installed on your system.

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

See Preparing a Support Archive.