Troubleshooting and FAQs

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

How can I tell what version of C++test is installed?

To check which version of C++test is currently installed:

1. Choose **Help> About Eclipse** (for the plugin) or **Help> About Parasoft C++test** (for the standalone version).
2. Check if there is a **C++test** button available.
3. Click the **C++test** button. The About Features dialog will open.
4. Check the **C++test** version in the features table (see the **Version** column).

Why does Eclipse crash upon startup now that I installed the C++test plugin?

Because it cannot write to the location in the installation directory where configuration files are kept. This is common when Eclipse is set up to have multiple users share a single install area. To prevent this, store config files in your home directory (for details on how to do this, see the first bullet in Other Requirements).

What if C++test was not (re-) installed correctly?

Try to launch Eclipse (once) with the `-clean` option. This will force a refresh of an internal (Eclipse) registry/cache.

What if the Parasoft menu is not available?

Set C/C++ or C++test as the current perspective (by choosing **Window> Open Perspective> Other**, then choosing **C/C++** or **C++test** from the dialog that opens).

What if the compiler family is not auto-detected?

Ensure that you have correctly configured your environment for the given compiler before launching Eclipse (including. $PATH, $LD_LIBRARY_PATH etc.)

How do I work with Microsoft Visual C++ compilers?

Visual C++ 6.0

Import the project as described in To create a C++test project from a Visual Studio 6.0 project:

Other Versions of Visual C++

For other versions of Microsoft Visual C++, use the "Visual Studio Command Prompt" utility script (provided with the Visual Studio installation) to open a console with the Visual Studio environment. Then, launch Eclipse from that console to ensure that the environment is correctly configured. Note that C++test is also available as a plugin for Visual Studio 2005 and newer versions.

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

To prevent C++test from running out of memory, add memory parameters to the script or shortcut being used to start C++test. The two parameters are the initial size of the JVM (Xms) and the maximum size of the JVM (Xmx). Typically, both are set to the same size (for instance, 1024MB). However, if you have occasional problems but don't want to always allocate a large amount of memory, you can set the parameters to different sizes (for example, 1024MB as the initial size and 1400MB for the maximum size). The maximum size you can set depends on your OS and JVM.

Examples:

C++test standalone: `cpptest.exe -J-Xms1024m -J-Xmx1400m`

C++test plugin for Eclipse: `eclipse.exe -vmargs -Xmx1400m`

You canalsocustomize 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).

What does the "cannot open .pathtoeclipse file" message mean (when launching cpptestcli)?

The `.pathtoeclipse` file is automatically created when you install C++test with the `extinstall` utility. This file contains the location of your Eclipse installation. If the file is missing (as indicated by the above message), manually add it by creating a simple text file, adding a line with your Eclipse location, then saving it as `<C++test_install_dir>/.pathtoeclipse`.

Is it possible to create a project using command-line utility (cpptestcli)?

Yes. See Creating a Project Using an Existing Build System.

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.

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

See Contacting Parasoft Technical Support.