

Windows Plugin

This topic explains how to install the C++test plugin into a working copy of Eclipse or Application Developer on Windows.

- For instructions on installing the plugin into Wind River Workbench, see [Overview of Wind River Workbench - C++test Integration](#).
- For instructions on installing the plugin into ARM DS-5, see [Overview of ARM DS-5 - C++test Integration](#).
- For instructions on installing the plugin into QNX Momentics, see [QNX Momentics Plug-in](#)

The section included:

- [About Multi-user Installations](#)
- [Prerequisites](#)
- [Installation](#)
- [Startup](#)
- [Licensing](#)

About Multi-user Installations

Each instance of Eclipse creates and writes to a configuration folder. If multiple instances of Eclipse/C++test are running in parallel, data corruption may occur. Eclipse provides a number of strategies for initializing the Eclipse configuration area in order to support multi-user installations. If you are implementing a multi-user installation of C++test, please choose the appropriate installation scenario:

Shared Configuration

In this scenario, users share not only an install area but also a master configuration area. By default, users must still have their own private writable configuration areas. A user's private configuration area is cascaded to the master configuration and will not contain any interesting data if the master configuration has been fully initialized and no changes to the set of plug-ins to be installed has occurred.

In this scenario, the system administrator initializes the master configuration (typically under the install location), and ensures the whole install and configuration areas are read-only to users. When users run the Eclipse-based product from the shared install location, since they do not have write access privileges to the configuration area under the install area, a local configuration area will be automatically computed and initialized.

Shared Installation

In this scenario, a single install area is shared by many users. The 'configuration' directory under the install area is home only to the config.ini as shipped with the product (it is not initialized). Every user has their own local standalone configuration location. This scenario requires making the install area read-only for regular users. When users start Eclipse, this causes the configuration area to automatically default to a directory under the user home dir.

Prerequisites

System Requirements

- One of the following platforms:
 1. Windows 7, Windows 8, Windows 10 (with an x86-compatible processor).
 2. Windows 7 x64, Windows 8 x64, Windows 10 x64, Windows Server 2008 x64 (with an x86_64-compatible processor), Windows Server 2012, Windows Server 2016
- 1.5 GHz or higher processor.
- 4 GB RAM (8 GB is recommended).
- A supported compiler or cross-compiler.
 - See [Supported Environments](#) for a list of supported compilers.

Windows and Windows Server notes

- Installation and initial run must be performed by a user with Administrator privileges.
- When running C++test, you must have write access to the directory containing the testing workspace and project. Specifically, if you are using a non-Administrator account, you will need to copy C++test example projects from the 'Program Files' location into the user directory and run the tests on the copied projects.

- The C++test CLI executable (cpptestcli.exe) should be executed from a directory with write access (because the working directory is the default location in which C++test generates the report after tests).
- If your machine-id is WIN32-0, remove the file located in <DRIVE>:\Users\<USER_NAME>\.parasoft\C++test\9.x\license.

IDE Requirements

- Eclipse IDE for C/C++ Developers 3.8, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 and a Java Runtime Environment (JRE) supported by Eclipse.
- Wind River Workbench 3.3.
- Renesas e2 Studio 5.1
- ARM DS-5 5.18 and higher
- Texas Instruments Code Composer Studio 5.x, 6.0, 7.4, 8.0

Other Requirements

- If earlier versions of C++test are installed on your system, the latest version of C++test must be installed in a new directory.
- Additional disk space is required for C++test project data.
- The recommended Japanese language encoding is Shift_JIS (ja_JP.PCK locale on Unix). Other encodings might cause font problems or prevent C++test from reading test results.

Alternative installation methods

For details on performing a silent installation see [Preconfiguration C++test During Installation](#).

Installation

About plugin and IDE architecture selection

You must install the appropriate plugin architecture for your IDE:

- Install 32-bit C++test to use with 32-bit Eclipse.
- Install 64-bit C++test to use with 64-bit Eclipse.

To install the C++test plugin for Eclipse on Windows:

1. In Windows Explorer, locate and double-click the self extracting archive.
2. Choose the setup language.
3. Click **Yes** when a dialog asks whether you want to install C++test.
4. Click **Yes** after you have read and agreed with the license information.
5. Choose the appropriate Eclipse/CDT target platform type, then click **Next**.
6. Enter your Eclipse installation directory, then click **OK**.
 - Choose the directory that contains the .eclipseproduct file.
7. Enter the desired destination directory for the C++test Extension files, then click **Next**.
 - If Parasoft Test 9.x Extension is not yet installed, you will be asked to select an installation directory for Parasoft Test's files.
 - If Parasoft Test 9.x Extension is already installed, C++test will be installed into it.
8. Enter the desired destination directory for the C++test Extension files, then click **Next**.
9. Close Eclipse if it is open, then click **OK** to close the dialog reminding you to close this program. C++test will then start copying files and installing the necessary files into the workbench. A dialog box with a progress indicator will open and indicate installation progress. When the installation is complete, a notification dialog box will open.
10. Click the **OK** button to close the notification dialog box.

Startup

Before launching C++test

For C++test to autodetect compiler and makefile settings, the necessary executables (compiler/linker, makefile, etc.) must be correctly configured. "Correctly configured" means different things for different compilers, but it typically involves ensuring that the executable is on the PATH.

Proper Compiler Configuration is Critical

In most cases, C++test needs to invoke the compiler and linker in order to perform static analysis and runtime testing tasks, which commonly involve preprocessing, compiling, and linking programs.

To access C++test's full functionality, the machine where C++test is run must have the complete development environment and compiler tool chain.

To launch the plugin:

- Launch your Eclipse-based IDE as normal.

Eclipse will automatically find the C++test plugin.

After Eclipse is launched, you should see a **Parasoft** menu added to the Eclipse menu bar. If you do not see this menu, choose **Window> Open Perspective> Other**, select **C++test**, then click **OK**.

If you suspect that C++test is not properly installed, see [Troubleshooting and FAQs](#) for help resolving some common installation problems.

Licensing

The license is configured through the centralized licensing framework (**Parasoft> Preferences> Parasoft> Licenses**). For details, see [Licensing](#).