

Project Creation Overview

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.

General Procedure

Before you can test code with C++test, it must be available in a C++test project within the workbench (a temporary work area—unique for each user). This involves:

1. Creating or importing a project:
 - If you have **existing Eclipse CDT, Wind River Workbench, or ARM DS-5 projects**, simply configure your existing projects for use with C++test. Proceed to step 2.
 - If you have **a build system that runs from the command line**, use the `cpptestscan` utility to collect information from the build process, then have C++test automatically create a project based on the collected information. Projects can be created from the command line or using a GUI wizard. For details, see [Creating a Project Using an Existing Build System](#)
 - If you want to **import existing Visual Studio 6.0 projects**, use the procedure described in [Importing an Existing Visual Studio 6.0 Project](#)
 - If you want to **import existing Wind River Tornado projects**, use the procedure described in [Importing an Existing Wind River Tornado Project](#).
 - Otherwise, configure your project from the GUI as described in [Creating a Project from the GUI](#)
2. Configuring build settings within the C++test project options.
 - This is required for all C++test projects—including existing Eclipse CDT, Wind River Workbench, and ARM DS-5 projects that you want to test with C++test.
 - See [Setting Project and File Options](#).

These two tasks need to be performed only once. After one team member creates and configures a project, it can be checked into source control, then reused by all developers working on the code as described in [Sharing Projects](#).

Working with C++test Projects

C++test projects define a group of source and header files that are built to a single binary result: library or executable. Each project maps to a directory in the file system. All files and folders from this directory are a part of the project and are visible as the project subtree. Additionally, projects may contain linked folders that point to external locations in the file system. All files and subfolders of such locations are visible as a part of the project. Projects can be created :

- In the root directory of the source files structure. In this case, all source files will be visible as part of the project.
- In another location (e.g., in the workspace directory). In this case, linked folders pointing to the source/header directories will need to be created.

The C++test Standalone GUI is built using Eclipse. Consequently, some of the general Eclipse capabilities for creating and organizing projects are used in C++test. For more details about these Eclipse capabilities, see the relevant topics in the Eclipse Workbench User Guide (available by choosing **Help> Help Contents**, then opening the **Workbench User Guide** book). Some Eclipse functionality is not available for C++test projects because they are not complete Eclipse projects. For example, C++test projects will not support Build or Run options from the Eclipse menu.

Note - Sharing Projects with Linked Resources

Linked folders are defined as absolute paths in project definition (`.project`) files. To make it easier to share projects containing linked resources, linked folders can also be specified relative to Path Variables. To define a Path Variable, choose **Window> Preferences** from the main menu and go to **General> Workspace> Linked Resources**.

Tip - Where to Create a Project?

Creating a project in the root directory of the source files structure is recommended if the project should be shared and C++test files/folders can be added to the project root directory. If C++test files cannot be added to the original file structure, then the project should be created in the workspace/external location and linked folders should be created to include the source/header files locations in the project.