

Sharing Projects and Test Assets Through Source Control

This topic explains how to share projects and related test assets by checking them in and out of your existing source control system.

Sections include:

- [Sharing Projects and Assets with any Source Control System](#)
- [C++test-Specific Sharing Tips](#)

Sharing Projects and Assets with any Source Control System

Key project artifacts are in ASCII. This makes it easy to maintain them via source control, and merge changes from multiple team members in a concurrent development environment.

To share projects and test assets using any source control system:

1. Ensure that your test assets (test cases, user-defined stubs, and so on) are stored within your project.
2. Add the project to source control.

Other team members can access the project files by checking them out from source control.

C++test-Specific Sharing Tips

1. Check in the necessary project definition files to ensure that the project can be re-created "from scratch". For example:
2. (Optional) Check in additional resources as desired. For example, you can check in:
 - The original project contents.
 - Tests
 - Stubs
 - Data sources

Do not check in temporary files

Do not check in temporary data/files (such as instrumented sources, compiled objects, linked test executables, and automatically-generated header files) with the project. By default, temporary files are in the

.cpptest directory at the same level as .sln or .vcproj files.

- These files should not be stored in source control because C++test can automatically re-generate these as needed. Moreover, they might include non-shareable information (like absolute paths).

- If this data is currently being stored within your project, change the C++test temporary files settings as explained in the C++test User's Guide.

3. If the project was configured to use options from a build data file, ensure that all users have the .bdf file in a consistent location (as set in the **Build data file** option in the project properties' build settings).
 - For example, you could use the project location (with files) for a single bdf. Or, you could use a workspace/temp area based on the project name.
 - Resource location, project location, or environment variables can be used.
 - There is no need to share the .bdf file. All users should generate it on their machines because of absolute paths used in the bdf file.

A checked in/checked out project can be imported by opening the appropriate xxx.sln /xxx.vcproj file in Visual Studio. Changes to the project structure (e.g. adding new stub files, or new test case folders) are saved in the original user's xxx.vcproj files.