

# Creating a Project from the GUI

If you cannot (or prefer not to) use the previously-mentioned project creation/import strategies, create the project from the GUI as follows:

- If you want the test files (automatically-generated settings files, test files, stub definitions, etc.) stored in the same directory as the project source, see [Using the Source Directory as the Project Location](#)
- Otherwise, see [Using an External Location as the Project Location](#). This option must be used when:
  - More than one project needs to refer to the same source directory/directories, and/or
  - When you want to keep test artifacts completely separate from the source code directories.



## Tip - Referencing BUILD\_ROOT in project settings

In project settings, you can reference BUILD\_ROOT as `${BUILD_ROOT}` or `${env_var:BUILD_ROOT}`.

If you are creating a new project directly in the build directory, which is at the BUILD\_ROOT, you can specify the project location as absolute path or using a workspace path variable. Do not use `${BUILD_ROOT}` in a project path.

## Using the Source Directory as the Project Location

To create a project using the source directory as the project location:

1. Choose **File> New> Project**.
2. Choose **C> C Project** or **C++> C++ Project**.
3. Click **Next**. The projects wizard will open.
4. Enter a name for the project in the **Project name** field.
5. Disable the **Use default location** option.
6. Enter the path to the actual source directory in the **Location** field.
7. Under **Project Types**, select **Makefile project**.
8. Under **Toolchain**, select the appropriate Toolchain.
9. (Optional) If you want to use Eclipse to build your project, click **Next**, then complete the settings in the available tabs.
10. Click **Finish**.

## Using an External Location as the Project Location

To create a project using an external location as the project location:

1. Create an empty project as follows:
  - a. Choose **File> New> Project**.
  - b. Choose **C> C Project** or **C++> C++ Project**.
  - c. Click **Next**. The projects wizard will open.
  - d. Enter a name for the project in the **Project name** field.
  - e. Click **Finish**.
2. Link your source files to that project as follows:
  - a. Choose **File> New> Folder**.
  - b. Select the name of the project that you created in Step 1.
  - c. Click the **Advanced** button.
  - d. Enable the **Link to folder in file system** option.
  - e. Enter or browse to the location of your source files.
  - f. Click **Finish**.

The linked files will appear within the project folder in the C/C++ Projects view and Navigator view, but they will continue to "live" within their original location. C++test will not add any new files to the linked source location; any files that it generates will be saved in the project directory.

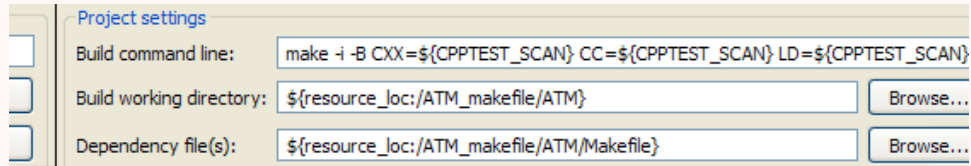


### Warning - Using Appropriate Build Settings

When configuring a project with a linked source folder, you need to modify any project build settings that refer to `${project_loc}`; these settings must be changed to use `resource_loc` instead. If you do not change these default settings, the project's source files will be skipped during testing.

To change the project's build settings:

1. Right-click the C/C++ Projects tree node for the project, then choose **Properties** from the shortcut menu. The Properties dialog will open.
2. Expand the **Parasoft> C++test** category in the left pane.
3. Select the **Build Settings** category.
4. Modify the **Build working directory** setting to `${resource_loc}/<project name>/<linked source dir>/<actual make working dir path from there>`.
5. Modify the **Dependency file(s)** setting to `${resource_loc}/<project name>/<linked source dir>/<actual make working dir path from there>/Makefile`.



See [Important note for projects with linked source folders](#) for details on how to make the appropriate modifications to your build settings.