

Importing an Existing Visual Studio 6.0 Project

- [Importing a Project with the GUI Wizard](#)
- [Importing a Project from the Command Line](#)
- **Example 1:** You have a single project with sources inside the same directory as the .dsp file
- **Example 2:** You have a VS workspace .dsw file with many .dsp projects with sources in many different folders
- **Example 3:** You want to import all projects in a specified <projects_dir>

You can import existing Visual Studio 6.0 projects from the GUI or from the command line.

Importing a Project with the GUI Wizard

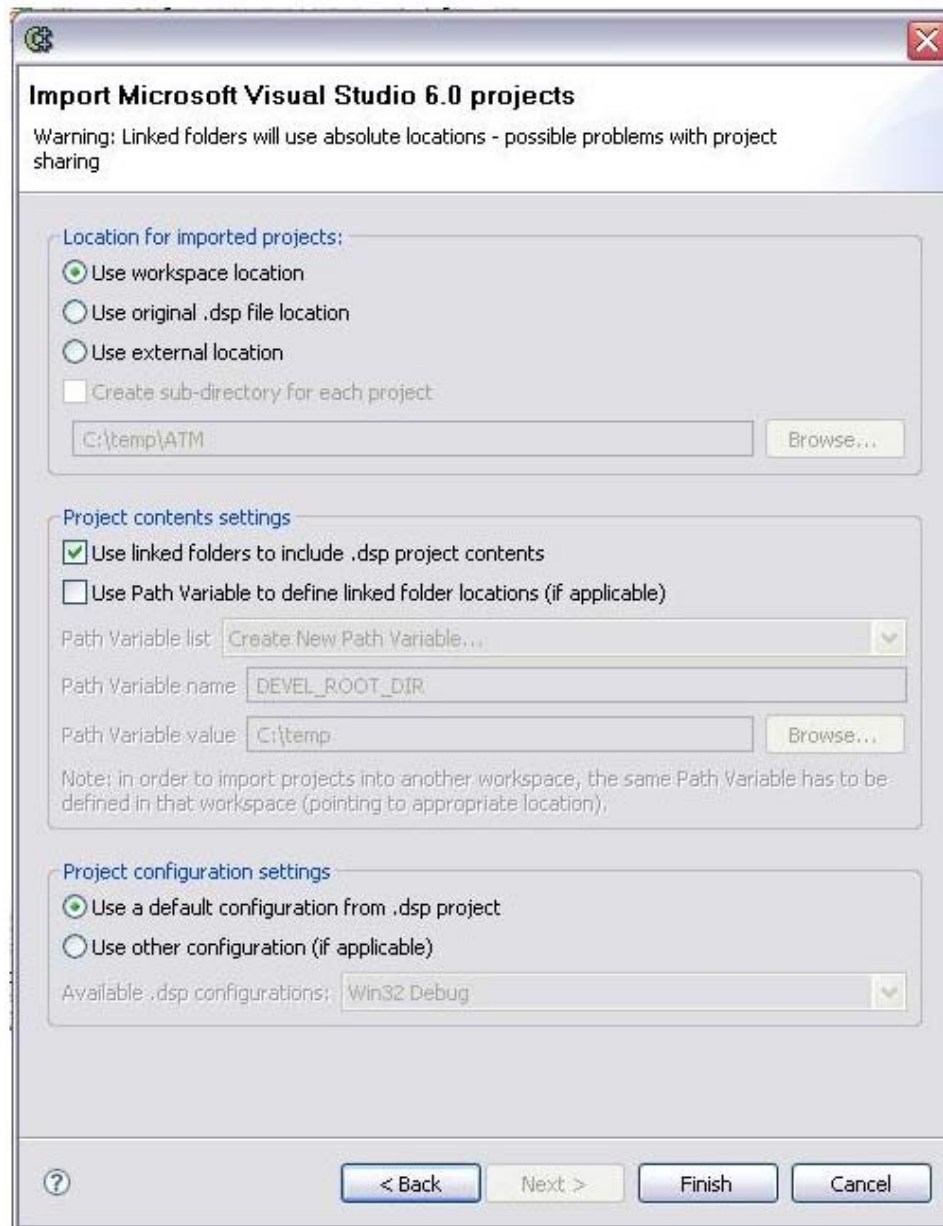
To create a C++test project from a Visual Studio 6.0 project:

1. Choose **File> New> Project**.
2. Choose **C++test> Import Microsoft Visual Studio 6.0 projects**.
3. Click **Next**. The Import Microsoft Visual Studio 6.0 projects wizard will open.
4. In the text field at the top of the wizard, specify the Microsoft Visual Studio 6.0 project file (.dsp), Microsoft Visual Studio 6.0 workspace file (.dsw), or root directory where you want C++test to search for Microsoft Visual Studio 6.0 projects. The projects found will be listed in the "Projects to be imported" tree.
5. Check the projects that you want to import, then click **Next**.

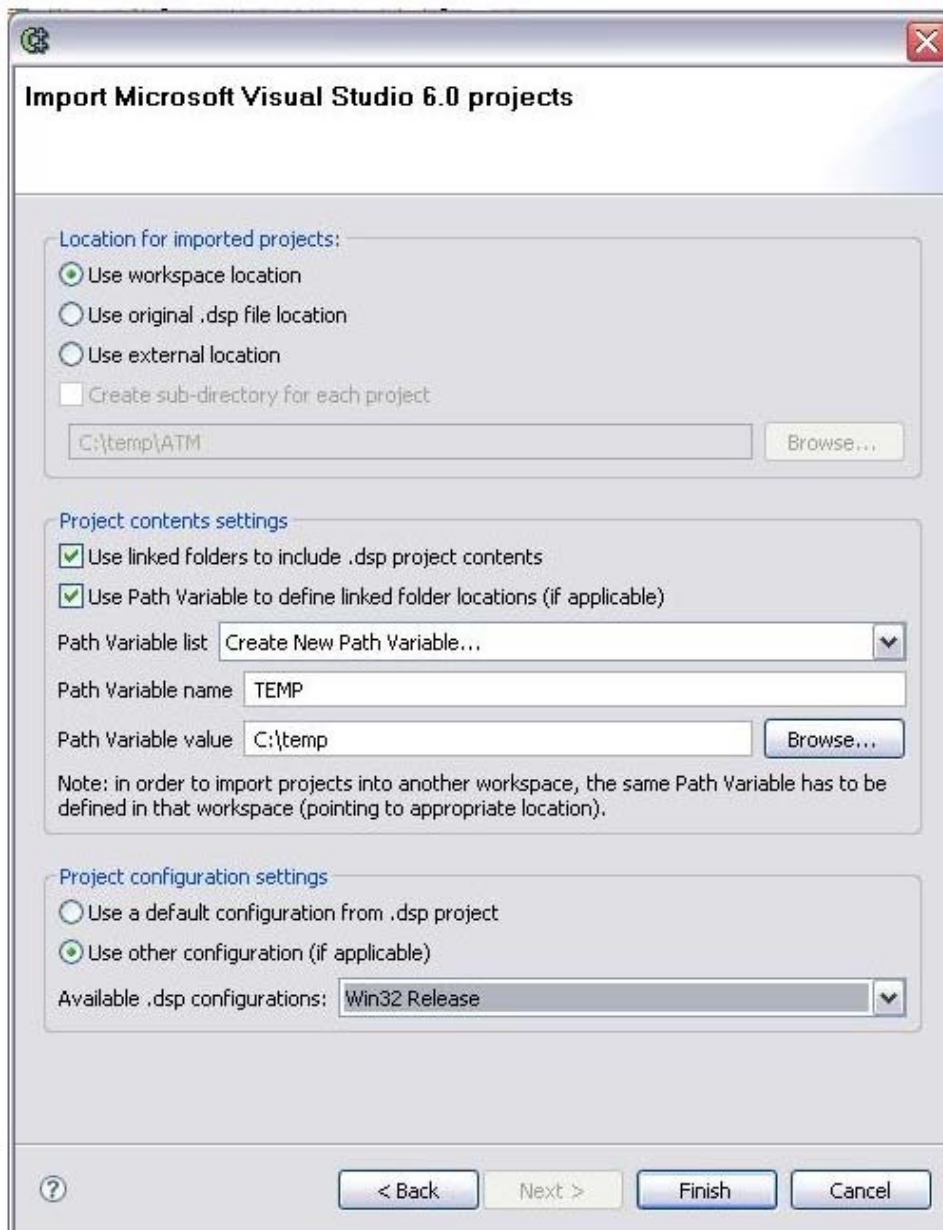


6. Specify the location for the imported projects.
 - To use the workspace location, choose **Use workspace location**.
 - To use the original .dsp file location, choose **Use original .dsp file location**.
 - To use a custom location, choose **Use external location**, then specify the desired location.

- If you want the project(s) imported into subdirectories created in the specified external location, enable **Create sub-directory for each project**.
- If you are importing only one project and you want it imported directly into the specified external location, clear **Create sub-directory for each project**.



7. If you want the .dsp project source folders to be linked into the created C++-test project, enable **Use linked folders to include .dsp project contents**.
8. If you want to specify path variables to be used in linked folder locations (e.g., to make project sharing possible), enable **Use Path Variable to define linked folder locations**, then specify the variables that you want to use. If this setting is false (the default), path variables will not be used while creating linked folders.
9. In the project configuration settings area, indicate the .dsp project configuration that you want used in the imported project.
 - To use the default project configuration from the .dsp project, enable **Use a default configuration from .dsp project**.
 - To use an alternative project configuration, enable **Use other configuration (if applicable)**, then select the desired configuration from the drop-down menu (which lists the configurations that are applicable to the specified projects).



10. Click **Finish**.

If an existing project in the workspace has the same name as the imported project, the **Finish** button will be disabled and a warning message will be reported.



Note

The imported project will be configured for C++test analysis only; it will not be configured for building in the Eclipse environment that C++test builds upon.

Importing a Project from the Command Line

To create a C++test project from a Visual Studio 6.0 project, do one of the following:

- Use the `-dsp <.dsp file | .dsw file | root location>` switch to `cpptestcli`.
- First, create a plain text options file with the desired `dsp.import.*` options (see [Local Settings \(Options\) Files](#) for details). Then, in your `cpptes tcli` invocation, use `-dsp <.dsp file | .dsw file | root location>` and `-localsettings <options.file>`.

If an existing project in the workspace has the same name as the imported project, the new project will not be imported.

Example 1: You have a single project with sources inside the same directory as the .dsp file

GUI

1. Open the Import Microsoft Visual Studio 6.0 projects wizard.
2. Select the .dsp project file, check the project you want imported, and click **Next**.
3. On the second wizard page, select **Use original .dsp file location**. This tells C++test to create the projects inside the .dsp projects directory.
4. Disable the **Use linked folders to include .dsp project contents** option. There is no need to link the directories, which are inside the .dsp directory.
5. Select **Use other configuration (if applicable)**, then select the desired Visual Studio project configuration from the **Available .dsp configurations** list.

Command Line

1. Prepare a plain text options file with the following settings:
 - `dsp.import.location=DSP`: This tells C++test to create the projects inside the .dsp projects directory.
 - `dsp.import.config=<desired config name>`: This tells C++test to use the specified Visual Studio project configuration, if applicable.
 - `dsp.import.linked=false`: This tells C++test not to use linked folders to include .dsp project contents.
2. In your `cpptestcli` invocation, use `-dsp <project.dsp file>` and `-localsettings <options.file>`.

Example 2: You have a VS workspace .dsw file with many .dsp projects with sources in many different folders

GUI

1. Open the Import Microsoft Visual Studio 6.0 projects wizard.
2. Select the .dsw workspace file, check all projects you want imported, and click **Next**.
3. On the second wizard page, select **Use workspace location**. This tells C++test to create the projects in the Eclipse workspace.
4. Select the **Use linked folders to include .dsp project contents** option. This tells C++test to link all source folders.
5. Select the **Use Path Variable to define linked folders locations** option to allow easier project sharing. This option will be deselected if it is not possible to use a path variable (because the linked folders don't have a common root folder). Additionally, a warning message will be shown to inform you there is no common root. You can still enable use of a path variable, but it will be used only when applicable.
6. Select the configuration which should be used in the imported projects. If the selected configuration is not applicable for a given project, then that project will use the default configuration.

Command Line

1. Prepare an options file with the following settings:
 - `dsp.import.location=WORKSPACE`: This tells C++test to create the projects in the Eclipse workspace.
 - `dsp.import.config=<desired config name>`: This tells C++test to use the specified configuration, if applicable.
 - `dsp.import.linked=true`: This tells C++test to use linked folders to include .dsp project contents.
2. (Optional) Add the following settings to the options file. If you add these settings, then the defaults will not be used. Otherwise a common root directory will be computed for all linked folders and the default variable name will be applied.
 - `dsp.import.pathvar.enabled=true`: This specifies that the path variable should be used.
 - `dsp.import.pathvar.name=<name>`: This specifies the path variable name.
 - `dsp.import.pathvar.value=<common_root_dir>`: This specifies the path variable value.
3. In your `cpptestcli` invocation, use `-dsp <project.dsw file>` and `-localsettings <properties.file>`.

Example 3: You want to import all projects in a specified <projects_dir>

GUI

1. Open the Import Microsoft Visual Studio 6.0 projects wizard.
2. Select <projects_dir>, check all projects you want imported, and click **Next**.
3. On the second wizard page, select **Use external location**. This tells C++test to create the projects in the specified directory.
4. Select the **Use linked folders to include .dsp project contents** option. This tells C++test to link all source folders.
5. Select the **Use Path Variable to define linked folders locations** option to allow easier projects sharing. As a path variable value, you can use <projects_dir>
6. Select the configuration which should be used in the imported projects. If the selected configuration is not applicable for a given project, then that project will use the default configuration.

Command Line

1. Prepare an options file with the following settings:
 - `dsp.import.location=<external_folder>`: This tells C++test to create the projects in an external folder.
 - `dsp.import.config=<desired config name>`: This tells C++test to use the specified configuration, if applicable.
 - `dsp.import.linked=true`: This tells C++test to use linked folders to include the .dsp project contents.
 - `dsp.import.pathvar.enabled=true`: This specifies that the path variable should be used.
 - `dsp.import.pathvar.name=<name>`: This specifies the path variable name.
 - `dsp.import.pathvar.value=<projects_dir>`: This specifies the path variable value.
2. In your `cpptestcli` invocation, use `-dsp <root_dir>` and `-localsettings <properties.file>`.



For a step-by-step guide to moving VS projects to Eclipse CDT, see <http://www-128.ibm.com/developerworks/opensource/library/os-ecl-vscdt/>.