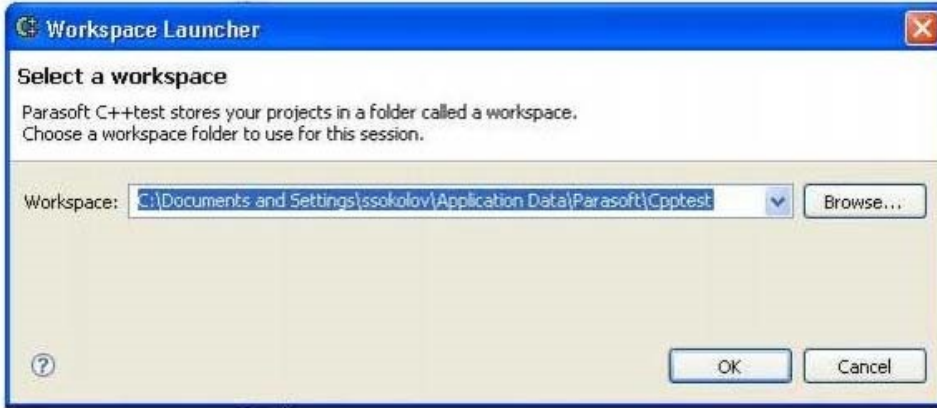


# Variable Definitions

This topic defines the variables that can be used in C++test.

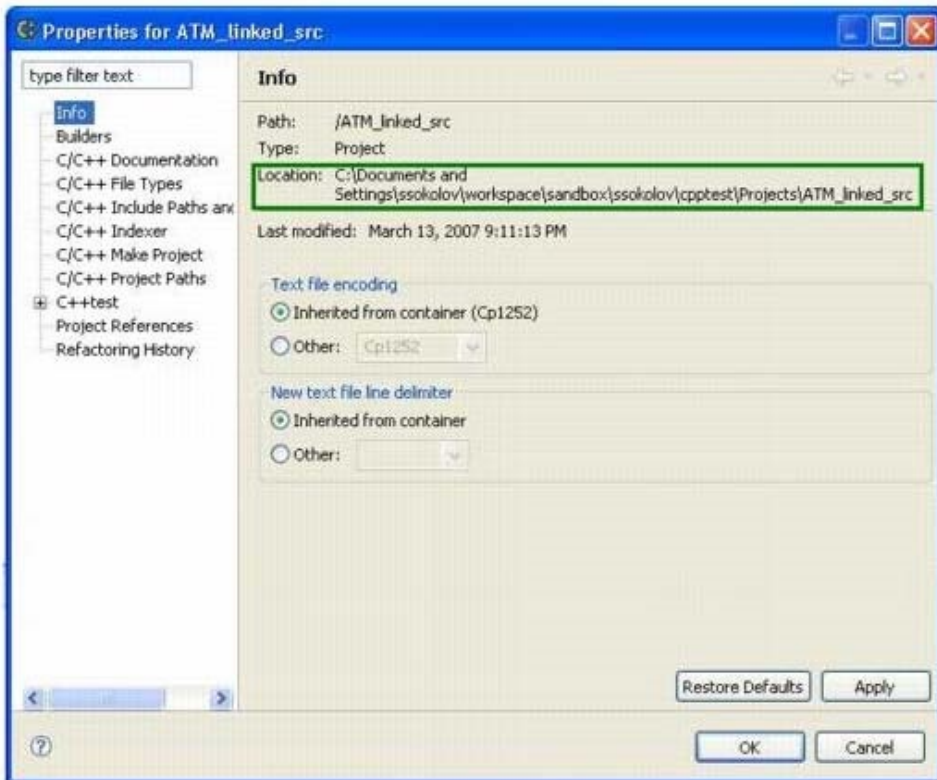
## General

- `$(workspace_loc-)` Location of the workspace. This location is displayed in the workspace selector (to open this, choose **File> Workspace**).



- `$(project_loc-)` Location of the tested project.

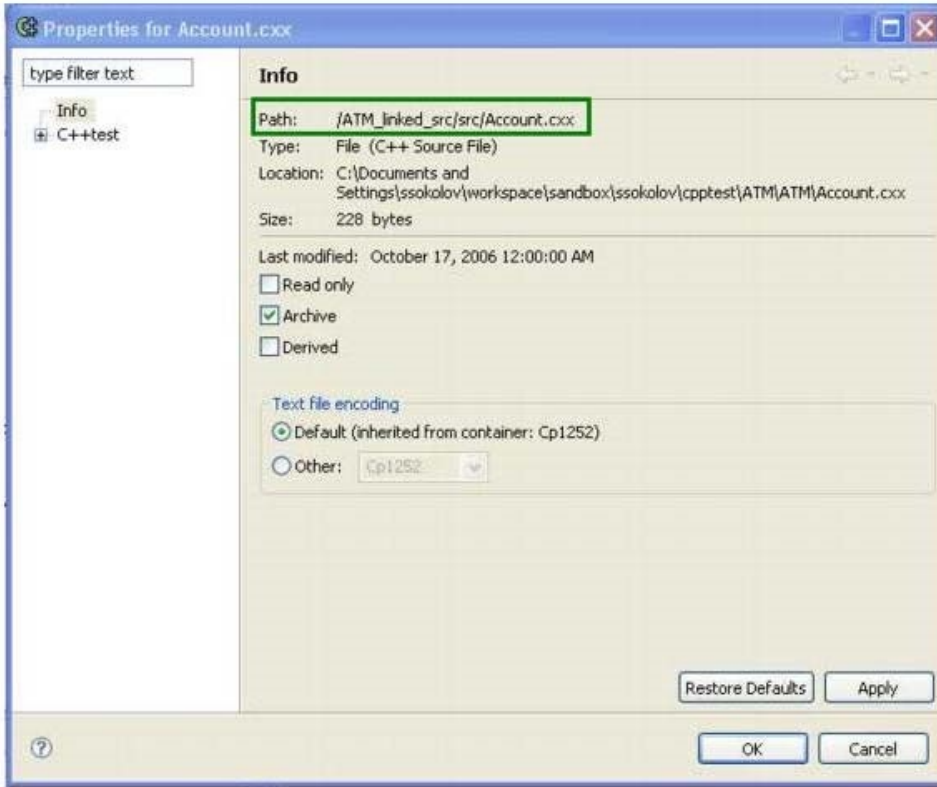
To see the location for a project, right-click it, choose **Properties**, and open the **Info** page. The location will be listed in the **Location** field.



- `$(project_name-)` Tested project name.
- `$(resource_loc:<RESOURCE_PATH>-)` Location of the specified project resource. This should be used in referencing a specific project resource in any applicable settings, such as the make run directory for projects with linked directories, or the path to .dsp file for VC6 projects, or similar. Use the resource\_loc variable to specify logical paths with respect to the project as it is registered in the workspace.

To determine the path for a project, right-click it, choose **Properties**, and open the **Info** page. Path corresponds to the <RESOURCE\_PATH> variable in

`$(resource_loc:<RESOURCE_PATH>)`. For example, if Path is /ATM\_linked\_src, then you would use `$(resource_loc:/ATM_linked_src)`. For a lower-level resource, this changes accordingly:



- `$(env_var:<ENV_VAR_NAME>-)` Value of the specified environment variable. This can be used for referencing environment variables from the user environment which determines physical file locations. For example, most typically, the root of a project location is an environment variable.
- `$(<ENV_VAR_NAME>-)` Value of the specified environment variable. This can be used for all components that expect disk locations
- `$(CPPTTEST_SCAN-)` Automatically initialized to the C++test options extractor utility with options; set in the environment when C++test runs the Build command. Used in the Build command line when configuring a Makefile-based project (**Project> Properties> Parasoftware>C++test> Build options**). Can be used directly in Makefiles and build scripts to replace the compiler/linker (when make or build scripts are run by C++test).
- `$(cpptest:original_options-)` Options from original (tested) project collected from monitoring the build or reading project files (Visual Studio or MULTI). Used in compiler / linker options in project / file properties.
- `$(cpptest:project_lib_files-)` Dependent libraries collected from the Visual Studio project/solution definition. Used in compiler / linker options in project / file properties.

## For Test Suite Files

### Key

- file = The source/header file where the tested function is defined.
- source file = The source file that defines a compilation unit where the tested function is defined.

## For Test Suite Output Layout Only

The following variables are used only in the **Test suite output file and layout** (in the Test Configuration's **Generation> Test suite** tab).

- `$(test_ext)` - C++test specific extension of a test suite file (.cpp).
- `$(file_name)` - File name.
- `$(file_base_name)` - File name without extension.
- `$(file_ext)` - File extension.
- `$(file_loc)` - File location.
- `$(file_loc_rel)` - File location relative to the project root.
- `$(file_uid)` - File unique identifier.
- `$(function_name)` - Tested function name.
- `$(function_uid)` - Tested function unique identifier (hash-code computed from the function signature/mangled name).

## For Test Suite Output Layout and Import C++test 6.x Test Suites Wizard

The following variables are used in the Import C++test 6.x Test Suites wizard, as well as in the **Test suite output file and layout** (in the Test Configuration's **Generation> Test** suite tab).

- `$(src_file_name)` - Name of context (source) file. (A "context file" is a source file that describes the compilation unit in which the given tested function is defined).
- `$(src_file_base_name)` - Name of context (source) file without extension.
- `$(src_file_ext)` - Extension of context (source) file.
- `$(src_file_loc)` - Context (source) file location.
- `$(src_file_loc_rel)` - Context (source) file location (relative to the project root).
- `$(src_file_uid)` - Context (source) file unique identifier (hash-code computed from the source file location).

## For Import C++test 6.x Test Suites Wizard Only

The following variable is used only in the Import C++test 6.x Test Suites wizard.

- `$(unit_name)` = C++test 6.x test unit name for imported test suite file.

## Advanced

### For Temporary Files

`$(cpptest:testware_loc)` - C++test temporary files location. Used in:

- Test exec working directory
- Flow recipe XML

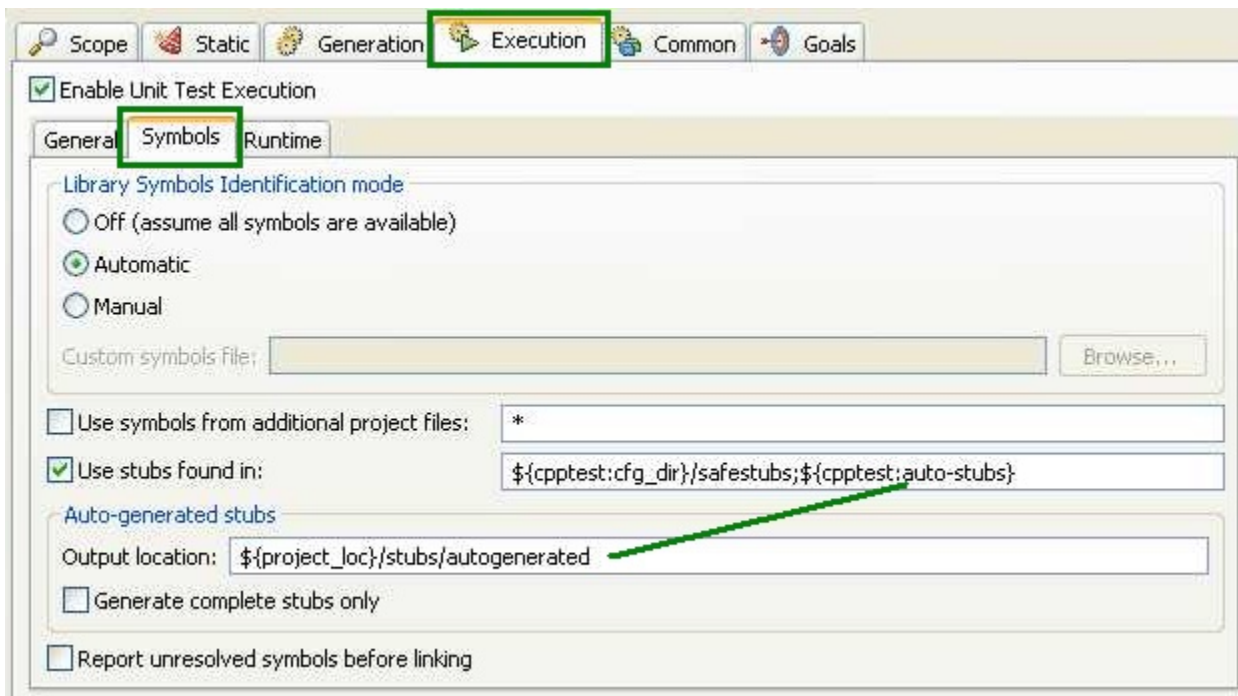
### For Configuration Files

`$(cpptest:cfg_dir)` - C++test configuration files location. Used in:

- Flow recipe XML
- Stub file locations
- Compiler / linker options in project / file properties

### For Automatically-Generated Stubs

`$(cpptest:auto-stubs)` - Auto-generated stub files location. This variable points to the value set in the **Test Configuration's Auto-generated stubs> Output** location, as shown in the screen shot below. This is basically a shortcut that synchronizes the directory where the stubs are generated and where they are picked up from.



Used in:

- Stub file locations-- only for Test Configurations that generate stubs.

## For Context-Specific Stubs

- `${ctx_name}` - name of the tested context file; e.g. 'foo.cpp'
- `${ctx_base_name}` - base name of the tested context file; e.g. 'foo'
- `${ctx_ext}` - extension of the tested context file; e.g. 'cpp'
- `${ctx_loc_rel}` - project-dependent location of the tested context file; e.g. 'src/subdir'

Used in:

- Stub file location ('Use symbols from additional project files') and auto-generated stub location ('Autogenerated stubs output location') for File-Scope Test Configurations.