Adding User-defined Test Cases

C++test provides several ways to add user-defined test cases:

- Test Case Editor - Use the Test Case Editor to create test cases without writing source code. The editor operates on predefined steps that represent typical actions inside the test case. It enables you to build the test case structure, specify input parameters, and define output expectations. You can save and modify test cases created with the Test Case Editor to easily maintain your tests.
- Test Case Wizard - Use the Test Case Wizard for black box testing of new code. It provides a simple way to specify inputs and expected outputs, and it offers a better analysis of the tested code's data dependencies (e.g. global variables), which facilitates test case setup. Using the Test Case Wizard is the only way to automatically generate a data source template for a test case.
    - A potential drawback of the Test Case Wizard is that you can only create one test at a time.
- Automated test case generation - This is designed to facilitate rapid development of a large number of customizable/extensible tests in bulk. This approach supports characterization testing on volumes of existing code, where tests are generated for many functions at once, and results of tests need to be captured automatically.
    - A potential drawback to automated test case generation is that editing is required to control test case values.
- Test Suite Wizard - This enables you to quickly create a form for writing complex test code that complies with the C++test test framework.
    - A potential drawback of using the Test Suite Wizard is that the code needs to be added to the template.

This chapter describes how to extend the test suite by adding new test cases via a graphical editor. The following information is included:

- Adding Test Suites and Test Cases with the Test Case Editor
- Adding Test Suites and Test Cases with the Graphical Test Case Wizard
- Adding Test Suites and Test Cases Manually
- Modifying Automatically-Generated Test Cases
- Using Setup and Teardown Functions
- Using Different Tests and/or Stubs for Different Contexts
- Specifying Custom Compiler Options for Standalone Test Suites
- Measuring and Validating Response Time in Real-time Systems
- Available Macros
- Testing Functions/Methods with Endless Loops
- Testing Functions from Unnamed (Anonymous) Namespaces
- Available Test Functions
- Example: Test Case Modification
- Maintaining the Test Suite

The test suite can also be extended with your existing unit test cases (i.e., CppUnit test cases); this is described in Executing Manually-Written CppUnit Test Cases.