

Extending and Modifying the Test Suite: Overview

This topic provides an overview of the different ways that you can extend and modify the automatically generated unit test suite.

Use the following table as a reference to determine which test suite extension/modification methods are best suited to your goals

To achieve this	Do this	Reference
Add new test cases to check specific unit-level functionality requirements or to improve coverage	Use the Test Case Editor or Test Case Wizard to create test cases graphically, or add code to test case templates	Adding User-Defined Test Cases
Modify automatically-generated test cases to check specific unit-level functionality requirements or to improve coverage	Edit the related test files	Adding User-Defined Test Cases
Modify test generation or execution settings	Configure settings in the Test Configuration panel's Generation and Execution tabs	Generation Tab Settings: Defining How Test Cases are Generated and Execution Tab Settings - Defining How Tests are Executed
Remove test cases and disable outcome checks or test cases that are not currently of concern to you	To remove a test suite: Right-click its Test Case Explorer node and choose Delete To remove a test case: Right-click its Test Case Explorer node and choose Delete To disable checking of a specific outcome: Right-click the unverified outcome in the Quality Tasks view, then choose Ignore Outcome from the shortcut menu To disable a complete test case: Right-click its Test Case Explorer node and choose Disable	Deleting and Disabling Tests
Prevent the testing of certain classes or methods	Specify the resources that you want to include or exclude	Testing a User-Defined Set of Resources
Convert automatically generated tests into a "functional snapshot" for regression testing (to identify changes/problems introduced by code modifications)	If the code is behaving correctly, right-click the unverified outcome node, then choose Verify Outcome from the shortcut menu	Verifying Test Cases for Regression Testing
Access data source values during testing	Configure test cases to access values stored in a data source.	Using Data From Data Sources to Parameterize Test Cases
Use standard I/O data in test cases	Add C++test Stream API calls for redirecting standard input/output stream	Using Data From Standard IO
Define custom stubs (to specify what values an external method/function returns to the class under test)	Use the Stub wizard to create a stubs framework, then customize it.	Adding and Modifying Stubs
Execute existing unit test cases (i.e., CppUnit test cases) using C++test	Ensure that your preferred test execution Test Configuration can find the test cases	Executing Manually-Written CppUnit Test Cases