Ut Framework Users Guide

# Overview

The unit test tools provide a unit test framework and a collection of utilities that are designed to facilitate unit testing. The establishment of the FSW Reuse Library (FSRL) has made it obvious that the development and maintenance of unit tests is critical to the success of the FSRL. Unit tests are an essential part of the FSRL and they must be developed and maintained in a cost effective manner.

The unit test framework relieves developers from repeatedly implementing common unit testing functions and they enforce a common look-and-feel that helps test maintenance. These tools implement an assertion-based testing philosophy that requires the developer to explicitly write verification statements that assert whether a condition is true or false. This approach is much different than a unit test that creates a log file that requires a developer to analyze in order to determine whether a test passed or failed.

The use of a third party unit test tool such as cunit (<http://cunit.sourceforge.net/>) was considered. However, this option was rejected because it added unnecessary complexity to the FSRL unit testing. A tool such as cunit is written to accommodate many different users which increases its complexity and obscures the basic functionality required by the FSRL unit tests. For example, cunit supports the generation of XML reports which is not seen as a valuable feature for the FSRL. In addition, little effort is required to implement and maintain the unit test framework. Therefore by developing our own unit test framework we have complete control over the framework’s features, complexity, and growth path. We also do not rely on an outside party to implement enhancements and/or bug fixes.

Xunit

Self verifying

Regression testing

# The Big Picture

The Ut Framework is organized into the following core components:

# The Test Case

Collection of test methods

Setup and teardown functions

Add test case function

# The Test Method

# Verifying Test Results

# Asserts

# Custom Asserts

# The Test Suite

Mention that at a higher level it would be possible to integrate all unit tests in a system into a single test runner by calling all test suites. Show example.

# The Test Runner

Controlling the Level of Output

Verifying Test Coverage

# Naming Conventions

# File Organization

Tools

Tips

Getting Started

Frequently Asked Questions

Printf’s in unit tests

Controllong the level of output

Combining test suits together

Test case checklist