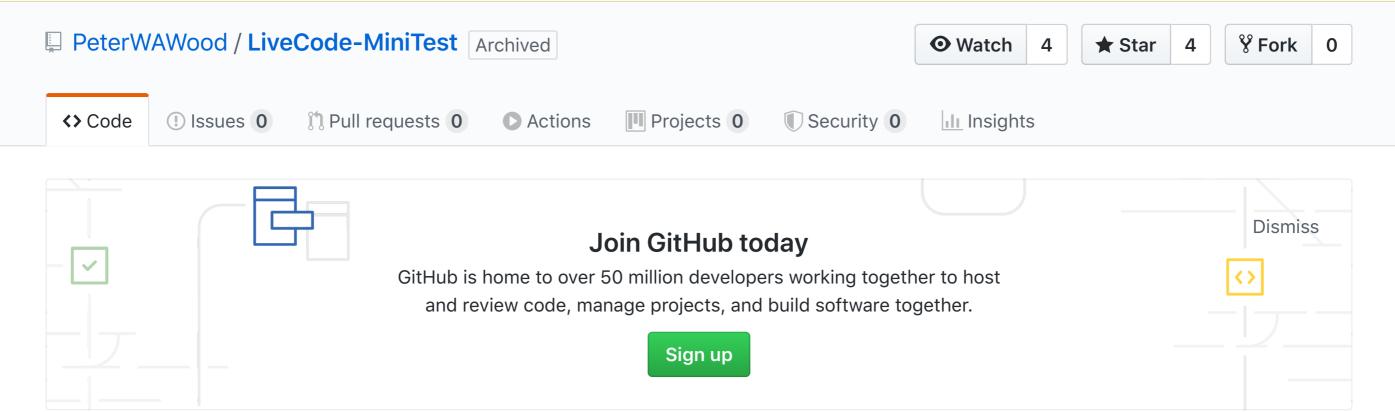
This repository has been archived by the owner. It is now read-only.



A minimal test framework for LiveCode

| -0- 18 commits | 1 branch | 🗇 0 packages | 🗘 0 releases | 1 contributor | ាំ្ BSD-3-Clause | |
|----------------------------|---------------|--|---------------------|---------------------------------------|-------------------------|--|
| Branch: master ▼ | | | | Find file | Clone or download ▼ | |
| PeterWAWood Update READ.ME | | | | Latest commit 88634ef on Oct 14, 2018 | | |
| test | a test with n | a test with no assertions is now reported as failing | | | 5 years ago | |
| LICENSE | Initial comm | Initial commit | | | 5 years ago | |
| MiniTest.livecode | Removed in | correctly encoded charcter | rs from a comment | | 5 years ago | |
| | | | | | | |

2 years ago

III README.md

README.md

LiveCode-MiniTest

Mini Test is a small, easy to use testing toolkit for LiveCode. It can be used in both GUI scripts and server scripts. Thanks to LiveCode's messaging, it can be used to test GUI applications as well as commands and handlers.

Warning

LiveCode Mini Test is no longer being developed or maintained.

Update READ.ME

How It Works

needed to run a MiniTest test script. The test script itself and MiniTest. Normally you would also include one or more stacks that you wish to test unless you are testing elements of the LiveCode language.

Mini Test consists of a set of handlers which you can use to create automated test scripts. There are at least two scripts

There are two levels of tests in MiniTest. The highest level is a Test Run which may contain individual tests or Test Files. The

Testing with Mini Test

lower level is a Test File which contains a individual tests. (A Test File cannot be included in a Test File). Actually, MiniTest is quite flexible and is happy with any number of test runs and test files in a single script. It is also happy to process only Test Runs (without Test Files) and only Test Files (without Test Runs).

At the start of your test script you simply include MiniTest and the stack you wish to test. MiniTest is in a file named

A Mini Test script is written by simply interspersing Mini Tests commands in your testing code.

MiniTest.LiveCode. The code to include in your testing script is: start using stack "<insert file path here>/MiniTest.livecode"

An indivdual test consist of four elements:

The Mini Test start test command

- Mini Test assert and refute commands to check that the test is as expected

Testing code

- The Mini Test end test command
- The testing code and Mini Test assert and refute commands can be freely inter-mingled.

Mini Test Handlers

Mini Test consists of the handlers listed below. The name of all the Mini Test the handlers is pre-fixed with "MT." which should practically reduce the chance to nil of a Mini Test handler having the same name as one of your handlers.

###MT.setTestReport pTestReporter This command prepares Mini Test to run in "GUI mode". It requires the name of a LiveCode Field to be supplied for it to use to report the test results. For example, if you want Mini Test to display it's result in a

Field called "Mini Test Report", you would use: MT.setTestReport "Mini Test Report"

This command prepares Mini Test to perform a test run.

MT.startTestRun

MT.startTestRun

MT.startTestFile pTestFileName

This command informs Mini Test that a new test file is starting. You need to provide a name for the test file to Mini Test.

MT.startTestFile "My First Mini Test File"

MT.startTest pTestName This command signfies the start of an individual test. A name is required so that Mini Test can report which test fails (if any

This command checks that the supplied actual value is true.

MT.startTest "Test 1"

do).

MT.assert pActual

MT.assert "a" = "A"

MT.assertEqual pExpected, pActual This command checks that the supplied actual value is equal to the supplied expected value.

MT.assertEqual 2, 1 + 1

expected value.

MT.refuteEqual "z", the first character of "abcdefgh"

###MT.refuteEqual pExpected, pActual This command checks that the supplied actual value is not equal to the supplied

MT.endTest This command signifies the end of a test. Mini Tests checks if there were any assert or refute commands which failed. It

MT.endTest

reports any failures.

This command signifies the end of a test file. Mini Test will print totals for the test file.

MT.endTestFile

MT.endTestFile

MT.endTestRun

This command signifies the end of a test run. Mini Test will print totals for the test run.

MT.endTestRun

MT.refute pActual

This command checks that the supplied actual value is false.

© 2020 GitHub, Inc. Terms Privacy Security Status Help

MT.refute "a" = "b"