Command Reference: CreateRegressionTestCommandFile()

Create a command file to run software regression tests

General Command

Version 3.08.02, 2010-01-06

The `CreateRegressionTestCommandFile()` command is used for software testing (or certification of processes used in operations) and creates a command file that includes a `StartRegressionTestResultsReport()` and multiple `RunCommands()` commands. A starting search folder is provided and all files that match the given pattern (by convention *Test_*.StateDMI*) are assumed to be command files that can be run to test the software. The resulting command file is a test suite comprised of all the individual tests and can be used to verify software before release. The goal is to have all tests pass before software release.

The following table lists tags that can be placed in # comments in command files to provide information for testing, for example:

```
#@expectedStatus Failure
```

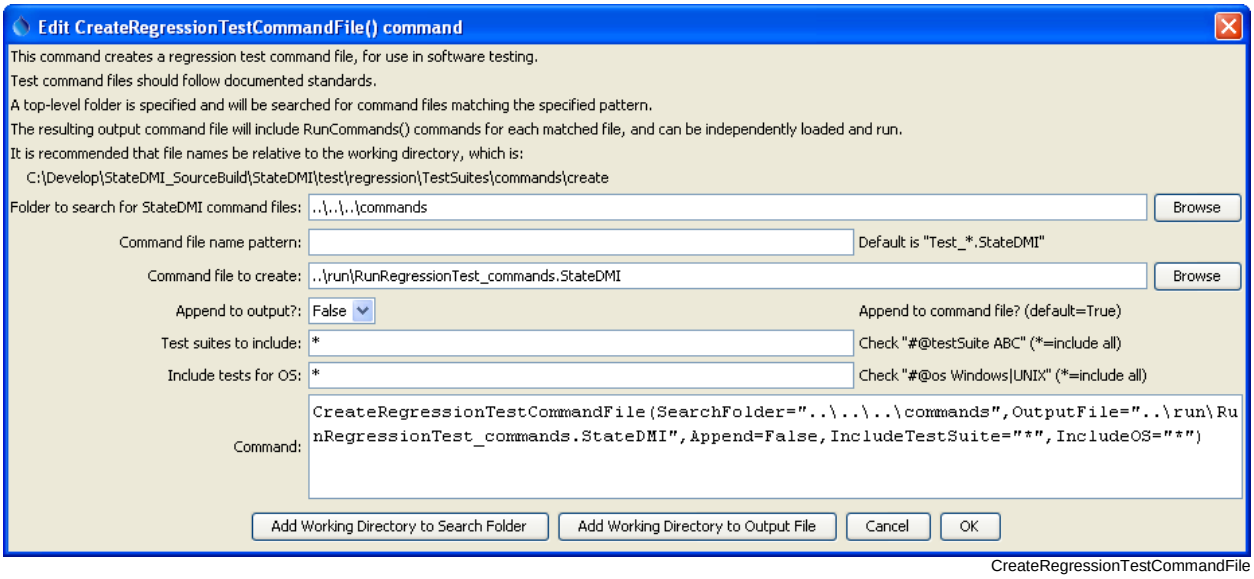
Command # Comment Tags

Parameter	Description

`@expectedStatus Failure`
 	
`@expectedStatus Warning`
	The `RunCommands()` command `ExpectedStatus` parameter is by default `Success`. However, a different status can be specified if it is expected that a command file will result in `Warning` or `Failure` and still be a successful test. For example, if a command is obsolete and should generate a failure, the expected status can be specified as `Failure` and the test will pass. Another example is to test that the software properly treats a missing file as a failure.

`@os Windows` `@os UNIX`	The test is designed to work only on the specified platform and will be included in the test suite only if the `IncludeOS` parameter includes the corresponding operating system (OS) type. This is primarily used to test specific features of the OS and similar but separate test cases should be implemented for both OS types. If the OS type is not specified as a tag in a command file, the test is always included (see also the handling of included test suites).
`@testSuite ABC`	Indicate that the command file should be considered part of the specified test suite, as specified with the `IncludeTestSuite` parameter. The test is included in all test collections if the tag is not specified; therefore, for general tests, do not specify a test suite. This tag is useful if a group of tests require special setup, for example connecting to a database. The suite names should be decided upon by the test developer.

The following dialog is used to edit the command and illustrates the syntax for the command.



CreateRegressionTestCommandFile

CreateRegressionTestCommandFile() Command Editor

The command syntax is as follows:

```
CreateRegressionTestCommandFile(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
`SearchFolder`	The folder to search for regression test command files. All subfolders will also be searched.	None – must be specified.
`OutputFile`	The name of the command file to create, enclosed in double quotes if the file contains spaces or other special characters. A path relative to the command file containing this command can be specified.	None – must be specified.
`SetupCommandFile`	The name of a StateDMI command file that supplies setup commands, and which will be prepended to output. Use such a file to open database connections and set other global settings that apply to the entire test run.	Do not include setup commands.
`FilenamePattern`	Pattern for StateDMI command files, using wildcards.	`Test_*.StateDMI`
`Append`	Indicate whether to append to the output file (`True`) or overwrite (`False`). This allows multiple directory trees to be searched for tests, where the first command typically specifies `False` and additional commands specify `True`.	`True`
`IncludeTestSuite`	If *, all tests that match `FilenamePattern` and `IncludeOS` are included. If a test suite is specified, only include tests that have `@testSuite` tag values that match a value in `IncludeTestSuite`. One or more tags can be specified, separated by commas.	* – include all test cases.
`IncludeOS`	If *, all tests that match `FilenamePattern` and	* – include all test

Parameter	Description	Default
	`IncludeTestSuite` are included. If an OS is specified, only include tests that have `@os` tag values that match a value in `IncludeTestSuite`. This tag is typically specified once or not at all.	cases.

See the `RunCommands()` documentation for how to set up a regression test. The following command file illustrates how to create a regression test suite.

```
CreateRegressionTestCommandFile(SearchFolder="..\..\..\commands\general",
    OutputFile="..\run\RunRegressionTest_commands_general.StateDMI",
    Append=False)
```

An example of the output file from running the tests is:

```
# File generated by...
# program:      StateDMI 3.08.02 (2009-09-29)
# user:         sam
# date:         Wed Sep 30 13:26:41 MDT 2009
# host:         SOPRIS
# directory:    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\TestSuites\commands\run
# command line: StateDMI
#
# The test status below may be PASS or FAIL.
# A test can pass even if the command file actual status is FAILURE, if failure is expected.
#     Test   Commands   Commands
#     Pass/  Expected   Actual
# Num Fail   Status     Status     Command File
#-------------------------------------------------------------------
   1  PASS   SUCCESS    SUCCESS
C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\AggregateWellRights\Test_AggregateWellRights_rg2007part.StateDMI
   2  PASS   SUCCESS    SUCCESS    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateDiversionDemandTSMonthly\
                                    Test_CalculateDiversionDemandTSMonthly.StateDMI
   3  PASS   SUCCESS    SUCCESS    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateDiversionDemandTSMonthlyAsMax\
                                    Test_CalculateDiversionDemandTSMonthlyAsMax.StateDMI
   4  PASS   SUCCESS    SUCCESS    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateDiversionStationEfficiencies\
                                    Test_CalculateDiversionStationEfficiencies.StateDMI
   5  PASS   warning    WARNING    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateStreamEstimateCoefficients\
                                    Test_CalculateStreamEstimateCoefficents_cm2005.StateDMI
   6  PASS   warning    WARNING    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateStreamEstimateCoefficients\
                                    Test_CalculateStreamEstimateCoefficients_gm2004.StateDMI
   7  PASS   warning    WARNING    C:\Develop\StateDMI_SourceBuild\StateDMI\test\regression\commands\CalculateStreamEstimateCoefficients\
                                    Test_CalculateStreamEstimateCoefficients_rg2007.StateDMI
```

This page is intentionally blank.