**DRAFT REPORT**

**REENGINEERING SWMM/EPANET   
USER INTERFACE APPLICATION**

**Phase I Test Report**

**Revision 0**

**EPA Contract Number: EP-G15H-01189**

**Task Order Numbers: 001, 002**

**Prepared For:**



**U.S. Environmental Protection Agency**

**Office of Research and Development**

**26 West Martin Luther King Drive**

**Cincinnati, Ohio 45268**

**Prepared by:**

**HydroGeoLogic, Inc.**

**11107 Sunset Hills Road, Suite 400**

**Reston, Virginia 20190**

**and**

**RESPEC, Inc.**

[**150 E. Ponce de Leon Ave., Suite 355**](x-apple-data-detectors://2/0)[**Decatur, Georgia 30030**](x-apple-data-detectors://2/0)

**May 2017**

**TABLE OF CONTENTS**

**Page**

[1.0 INTRODUCTION 1-1](#_Toc453305718)

[1.1 OBJECTIVES 1-1](#_Toc453305719)

[1.2 REPORT ORGANIZATION 1-1](#_Toc453305720)

[2.0 TESTING GUIDELINES AND ENVIRONMENT 2-1](#_Toc453305721)

[2.1 TESTING GUIDELINES 2-1](#_Toc453305722)

[2.1.1 Quality Assurance Plan 2-1](#_Toc453305723)

[2.1.2 Application Feature Requirements 2-1](#_Toc453305724)

[2.1.3 Coding Style Standard 2-1](#_Toc453305725)

[2.2 TESTING AND DEVELOPMENT ENVIRONMENT 2-1](#_Toc453305726)

[3.0 UNIT TESTING 3-1](#_Toc453305728)

[3.1 METHODOLOGY 3-1](#_Toc453305729)

[3.2 UNIT TESTINGASSESSMENT 3-1](#_Toc453305730)

[4.0 REGRESSION/INTEGRATION TESTS 4-1](#_Toc453305731)

[4.1 METHODOLOGY 4-1](#_Toc453305732)

[4.2 ASSESSMENT OF REGRESSION/INTEGRATION TESTS 4-2](#_Toc453305733)

[5.0 COVERAGE TESTING 5-1](#_Toc453305734)

[6.0 STYLE TESTING 6-1](#_Toc453305735)

[7.0 USER INTERFACE(UI) TESTING 7-1](#_Toc453305736)

[8.0 REFERENCES 8-1](#_Toc453305737)

**LIST OF FIGURES**

**Page**

[Figure 7.1 Sample Page of a UI Test Dialog Box 7-2](#_Toc453255433)

**LIST OF ACRONYMS AND ABBREVIATIONS**

AFRD Application Features Requirement Document

API Application Programming Interface

ASCII American Standard Code for Information Interchange

EPA U.S. Environmental Protection Agency

HGL HydroGeoLogic, Inc.

IDE integrated development environment

MTP Minimum Testable Product

OS Operating System

PEP Python Enhancement Proposal

QAPP Quality Assurance Project Assurance Plan document

SWMM Storm Water Management Model

UI User Interface

*This page was intentionally left blank.*

**REENGINEERING SWMM/EPANET   
USER INTERFACE APPLICATION**

**REPORT FOR SUMMARIZING PHASE I TESTS**

# INTRODUCTION

Storm Water Management Model (SWMM) and EPANET are EPA software for evaluating storm water collection, distribution and piping systems. Recently, RESPECT was tasked by EPA to develop a modular and extensible User Interface (UI) allowing “Plug-in” and scripting supports so that new application features created by the EPA, third-party developers, and end users can be deployed within the application. For this project, HydroGeoLogic, Inc. (HGL) and RESPEC, Inc.performed testing on a number of aspects of software development. The tasks performed include:

* Unit testing: Tasks include:
  + Develop unit tests for code modules
  + Perform unit tests and record results
  + Provide documentation detailing the unit test results
* Regression testing: Tasks include:
* Develop appropriate user interface test sequences.
* Perform test sequences and document results.

## OBJECTIVES

The objective of this document is to provide a summary of the tests conducted within the scope work of this particular version of software deliverable. Test objectives, methodologies, results, and the recommendations are summarized and provided for future references.

## REPORT ORGANIZATION

This report consists of seven chapters. Chapter 1 gives an introduction and overview of the report. Chapter 2 summarizes the test activities including the test environment, tools, rules and standards. Chapters 3 and 4 present the methodology and assessment processes for unit tests and regression/integration tests. Chapter 5 addresses coverage testing. Chapter 6 discusses style testing evaluation. Chapter 7 presents results from interface testing.

*This page was intentionally left blank.*

# TESTING GUIDELINES AND ENVIRONMENT

This section provides a summary of the testing environment and general guidelines.

## TESTING GUIDELINES

### Quality Assurance Plan

The Quality Assurance Project Assurance Plan document (QAPP) (Aqua Terra Consultants, 2015a) requires the software to be developed in an iterative developing-testing cycle such that most tests are performed in parallel to development. Detailed testing standards are provided in this document.

### Application Feature Requirements

Agile development methods are implemented for this project and the development activities are carried out under three sets of Minimum Testable Products (MTPs) defined in the Application Features Requirement Document (AFRD) (Aqua Terra Consultants, 2015b). Tests are designed to meet the requirements listed for each MTP in the AFRD. In addition, User’s guides for the command versions of SWMM (Rossman, 2015) and EPANET (Rossman, 2000) are frequently referenced to ensure that the UI is backward compatible and meets the requirements of SWMM and EPANET users.

### Coding Style Standard

Widely-used Python Enhancement Proposal (PEP) 8 coding standard (Van Rossum, 2001) is used for this project as coding standard.

## TESTING AND DEVELOPMENT ENVIRONMENT

Currently testing environment is “Windows 7, 64-bit, administrative account”. The development of the tests is carried out under Python IDE - Pycharm community edition 64-bit. Detailed installation guide for setting up the development environment is documented on SWMM-EPANET GitHub Wiki page (<https://github.com/USEPA/SWMM-EPANET_User_Interface/wiki/Getting-Started-as-a-Developer>). The test development environment is set up with small variations because of additional test tools installed (See Appendix A: Test Environment). GitHub integrated within Pycharm IDE is used for source control. GitHub issue tracker is used along with spreadsheets to document and report issues.

*This page was intentionally left blank.*

# UNIT TESTING

This section lists and describes the unit testing of EPANET and SWMM model input sections (components) within the scope of MTPs 1 and 2 defined in the Application Features Requirement Document (AFRD) (Aqua Terra Consultants, 2015b).

## METHODOLOGY

The QAPP document (Aqua Terra Consultants, 2015a) requires each public method of a software component (model input section) to have an associated unit test method and recommends using Python’s built-in “unittest” framework. In this development, each model input section (component) is represented by a Python class in the core section of the source code with public methods *set\_text* and *get\_text* which are the primary focus of unit testing. The *set\_text* method reads a section of the input file from text into the Python class, where it is accessible to the user for editing in the user interface and to the script or plug-in author as class attributes. The *get\_text* method goes the other direction, turning the Python class representation of the section into text suitable for writing to a model input file. In a subsequent development in 2016, the *set\_text* and *get\_text* methods were refactored and turned into classes in module inp\_reader\_project, inp\_reader\_sections, inp\_writer\_project and inp\_writer\_sections. The *set\_text* method was replaced by *read* method of each reader class, and the *get\_text* method was replaced by the *as\_text* method of each writer class. For each section, test examples are assembled from the most recent user manuals (Rossman, 2000 and 2015) and from the provided example input sequences. Data input options listed in the manuals are used as guidelines to formulate tests. Many options and some formatting variations are tested.

Each unit test of a model input section starts with a test string that is a complete example of this section as it would appear in the text input file. The test uses the *read* method of the individual *reader* class for this section to read the test string into a class object, then tests are conducted, including comparing the data attributes of the class object with the data in the text string, and using the *match* method to test whether the *as\_text* method produces output that matches the original test string. The *match* method compares the test string and the output string while ignoring insignificant differences such as tabs, spaces, order of lines, and capitalization. Testing of a section may include testing of one or more examples depending on the complexity of the section and the variety of valid inputs.

Unit tests are designed as automated tests. Separate test reports are generated for SWMM and EPANET each time the tests are executed. The latest combined unit test reports are always available in the repository as test/core/ test\_results\_core.html by running test/core/ unittest\_all\_with\_cmd.py. Issues discovered during test development have been reported to the development team and most have been addressed. A spreadsheet has been developed to record the bugs and theirs conditions. An example of the spreadsheet is also attached in Appendix B. Recently, we have experimented with reporting issues via the GitHub issue tracker and may increase the usage of that in the next phase.

## UNIT TESTINGASSESSMENT

The last unit test reports are included in Appendices B-1 (Unit Test Report), and B-2 (Sample Bug Report). An overall assessment of the Unit testing is summarized below:

* All 25 tests for EPANET core classes within the scope of MTPs 1 and 2 passed.
* All 135 tests for SWMM core classes within the scope of MTPs 1, 2 and 3 passed.
* Additional tests within the scope of MTP 3 may be developed in the future if needed.

# REGRESSION/INTEGRATION TESTS

Integration and regression testing exercises the entire process of reading and writing a complete input sequence and running the model for a variety of model input sequences. This is integration testing because it tests all of the core classes at the project and section levels as they read and write their contents and it serves as regression testing because it is easily extensible if a new example input file is found that triggers a problem. These tests are performed for all examples provided for the development team as well as additional examples downloaded from the EPA SWMM website (https://www.epa.gov/water-research/storm-water-management-model-swmm#resources). A total of 54 test examples of SWMM and 5 examples of EPANET have been tested. Detailed methodologies for SWMM and EPANET are provided as Pseudo-code below individually due to small variations between the two models.

## METHODOLOGY

For SWMM and EPANET, the test methodology includes the following steps:

Loop through all examples:

use ProjectReader.read\_file to read example's .inp file

use ProjectWriter.writer\_file to write a copy .inp\_copy

using each of these two files as inputs, run swmm5.exe or epanet2d.exe (from \Externals\epanet or swmm\model)

compare outputs:

If error occurs when executing the exe file using the original .inp file:

Test passes with message "----", as it is not a problem in the code but invalid original input.

If the original .inp file produces binary output but the copy does not:

Test fails.

If the two rpt files differ in non-white space and not-exempted lines:

Test fails.

Otherwise:

Test passes.

Note: A custom method to compare\_two\_analysis\_blocks is used to compare rpt files, where only lines between “Analysis begun” and “Analysis ended” are compared and all empty lines, spaces, tabs, and lines that contain model running time stamps or error messages (error message numbers may differ) are exempted from the comparison. Binary output files contain run-specific information and are not compared.

The test examples are stored under test/core/epanet/examples/ and test/core/swmm/examples/. A base class RegressTestBase is created for the regression tests of both SWMM and EPANET to reduce code redundancy. Regressions tests are performed by running test/core/epanet/ epanetregressiontest.py and test/core/swmm/ swmmregressiontest.py. The latest regression/integration test reports are provided as test/core/epanet/Test\_Example\_Summaries\_EPANET.html and test/core/swmm/Test\_Examples\_Summaries\_SWMM.html.

## ASSESSMENT OF REGRESSION/INTEGRATION TESTS

The regression/integration test reports are included in Appendices C-1 (EPANET) and C-2 (SWMM). An overall assessment of the tests is summarized below:

* All 5 EPANET tests passed.
* All 54 SWMM tests passed

These sets of tests helped build confidence in the backward compatibility and the integrated performance of the system.

# COVERAGE TESTING

Test coverage is a measure of how much of the source code is exercised during a test. Coverage is a good way to see which code is not yet tested by the current tests. A coverage report can find a whole module or method that is not covered by tests and can also find methods containing some lines that are not exercised by tests. Coverage of 100% indicates that every line of code was executed during testing. Complete coverage is always a goal, but this is not always achievable. In some cases, code is only executed in conditions that are difficult or impossible to duplicate during a test. For example, some code may handle an error condition that other code currently prevents from ever happening. While coverage is valuable in measuring the breadth of testing, it is not a guarantee that testing is truly comprehensive. When a test runs a line of code once, it is only testing one set of conditions. In practice, our tests run the same code more than once with different inputs to test how well it functions in a variety of conditions.

The Python standard Coverage.py tool is used to report on the coverage of the entire suite of automated unit and integration tests described in the sections above. Coverage reports are generated by running the tests and coverage module from the command line.

For evaluating unit tests coverage, the following commands are run under /test/core:

*coverage unittest\_all\_with\_cmd.py*

*coverage report >> Report\_coverage\_unittest.txt*

Resulting coverage report is saved as Report\_coverage\_unittest.txt

For evaluating unit + regression tests coverage, the following commands are run under /test/core:

*coverage run all\_unit\_regress\_cmd.py*

*coverage report >> Report\_coverage\_all.txt*

Resulting coverage report is saved as Report\_coverage\_all.txt

The combined coverage report is attached as Appendix D. For all tests and core modules that have been exercised in the unit testing and regression/integration testing, the overall coverage rate is 88%, for exercising the unit testing only, the overall coverage rate is 77%.

*This page was intentionally left blank.*

# STYLE TESTING

Automated testing is recommended for checking the code adherence to the PEP 8 standard. "Pycodestyle" (referred to as the “pep8” utility) has been recommended by the QAPP document (Aqua Terra Consultants, 2015a) for checking the coding styles. The package is implemented, but rather than denoting the test as failed if a format check is not passed, style discrepancies are reported to the developers and fixed upon further discretion.

*This page was intentionally left blank.*

# USER INTERFACE(UI) TESTING

UI tests are difficult to automate. Exploring automated open-source UI test tool may be in the list of actions for the future tasks. The initial user interface testing effort is focused on building tests as a list of steps that can be handled by the current manual testing method and can also be adapted as tests in a future automated testing tool. Current UI tests are designed such that a tester will perform a set of actions in a checklist. Closely-related actions are grouped into one text file and displayed as one page of the test (Figure 7.1). Once a set of actions have been completed, the tester clicks ‘Ok’ to reach the next set. A ‘Skip’ button is provided at the bottom of the page to skip the entire page of actions when not testing those items. Each step successfully tested is checked off by the tester. Steps not checked are recorded as failures. A text box is provided in each page for testers to write additional notes that will be carried to the final test reports. A total of 12 pages of actions have been created and each contains several actions. Some issues have been identified and reported using GitHub issue trackers. An example of a test text file is attached in Appendix E.

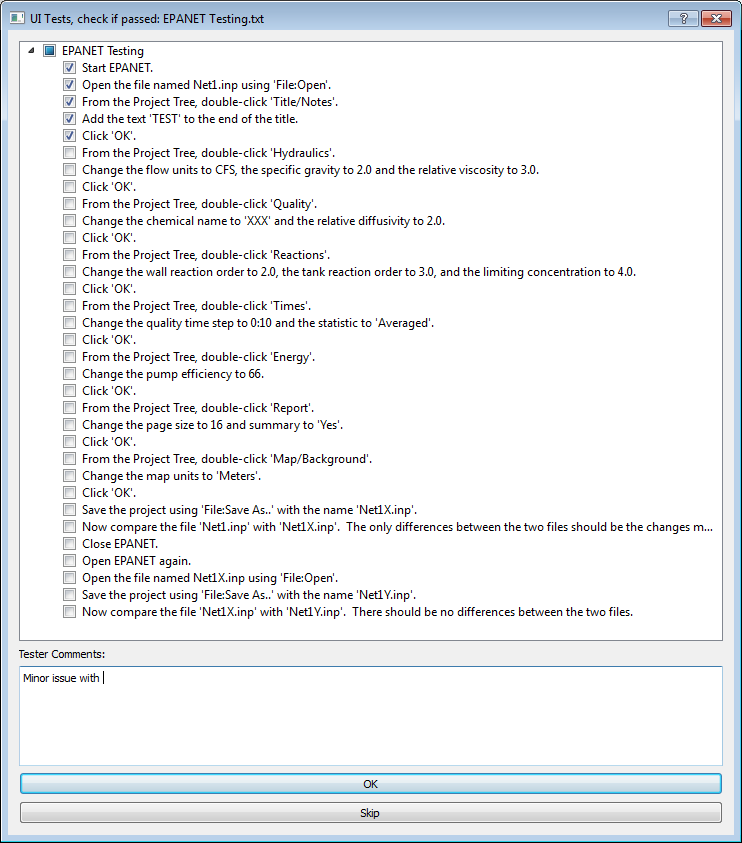


Figure 7.1 Sample Page of a UI Test Dialog Box

# REFERENCES

Aqua Terra Consultants, 2015a. Reengineering SWMM/EPANET User Interface Application Software Architectures, QAPP-16 Revision 0, Prepared For U.S. Environmental Protection Agency Office of Research and Development, 26 West Martin Luther King Drive, Cincinnati, Ohio 45268, November 2015

Aqua Terra Consultants, 2015b. Reengineering SWMM/EPANET User Interface Application Software Architectures: Application Features Requirements Document, RSI-2577 Revision 0, Prepared For U.S. Environmental Protection Agency Office of Research and Development, 26 West Martin Luther King Drive Cincinnati, Ohio 45268, December 2015

Rossman, L.A., 2000. EPANET2 User’s Manual, Office of Research and Development, Water Supply and Water Resources Division, Environmental Protection Agency (EPA), September 2000

Rossman, L.A., 2015. Storm Water Management Model User’s Manual Version 5.1, Office of Research and Development, Water Supply and Water Resources Division, EPA, September. 2015

Van Rossum, G. and Warsaw, B., PEP: 8 Style Guide for Python Code Version: 68852 Created: 05-Jul-2001， Post-history 01-Aug-2013

*This page was intentionally left blank.*

**APPENDIX A**

**TEST ENVIRONMENT**

*This page was intentionally left blank.*

**APPENDIX A**

**TEST ENVIRONMENT**

Create virtual environment for SWMM/EPANET using Anaconda

Install Anaconda, then create the virtual environment by type the following in command line window or shell:

>>conda create -n swmm python=2.7 anaconda

>>activate swmm

* With [swmm] at the beginning of the command line type the following

>>conda install pyqt

>>conda install qtconsole

>>conda install enum34

>>pip install doxypypy

* Python packages required for testing:

>>pip install coverage

>>pip install pycodestyle

*This page was intentionally left blank.*

**APPENDIX B-1**

**UNIT TEST REPORT: EPANET AND SWMM**

*This page was intentionally left blank. .*

**SWMM-EPANET Core Test Report**

**Start Time:** 2017-03-31 15:20:13

**Duration:** 0:00:00.884000

**Status:** Pass 160

Unit test results

Show [Summary](javascript:showCase(0)) [Failed](javascript:showCase(1)) [All](javascript:showCase(2))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Group/Test case** | **Count** | **Pass** | **Fail** | **Error** | **View** |
| test.core.epanet.test\_title.SimpleTitleTest: Test Title section | 5 | 5 | 0 | 0 | [Detail](javascript:showClassDetail('c1',5)) |
| test.core.epanet.test\_options.SimpleOptionsTest: Test Options section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c2',2)) |
| test.core.epanet.test\_report.SimpleReportTest: Test Report section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c3',3)) |
| test.core.epanet.test\_patterns.SimplePatternTest: Test one pattern | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c4',2)) |
| test.core.epanet.test\_curves.SimpleCurveTest: Test Curves section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c5',2)) |
| test.core.epanet.test\_energy.SimpleEnergyTest: Test Energy section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c6',2)) |
| test.core.epanet.test\_demands.SimpleDemandsTest: Test Demands section | 1 | 1 | 0 | 0 | [Detail](javascript:showClassDetail('c7',1)) |
| test.core.epanet.test\_quality.SimpleQualityTest: Test Quality section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c8',2)) |
| test.core.epanet.test\_reactions.SimpleReactionsTest: Test Reaction section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c9',2)) |
| test.core.epanet.test\_sources.SimpleSourcesTest: Test Sources section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c10',2)) |
| test.core.epanet.test\_backdrop.SimpleBackdropTest: Test Backdrop section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c11',2)) |
| test.core.swmm.test\_backdrop.SimpleBackdropTest: Test BACKDROP section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c12',3)) |
| test.core.swmm.test\_options\_dates.OptionsDatesTest: Test OPTIONS: Dates | 1 | 1 | 0 | 0 | [Detail](javascript:showClassDetail('c13',1)) |
| test.core.swmm.test\_options\_dynamicwave.OptionsDynamicWaveTest: Test OPTIONS section: Dynamic Wave | 1 | 1 | 0 | 0 | [Detail](javascript:showClassDetail('c14',1)) |
| test.core.swmm.test\_files.SimpleFilesTest: Test FILES section | 6 | 6 | 0 | 0 | [Detail](javascript:showClassDetail('c15',6)) |
| test.core.swmm.test\_options\_general.OptionsGeneralTest: Test OPTIONS section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c16',2)) |
| test.core.swmm.test\_map.SimpleMapTest: Test MAP section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c17',2)) |
| test.core.swmm.test\_report.SimpleReportTest: Test REPORT Section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c18',2)) |
| test.core.swmm.test\_options\_timesteps.OptionsTimestepTest: Test OPTIONS: Time steps | 1 | 1 | 0 | 0 | [Detail](javascript:showClassDetail('c19',1)) |
| test.core.swmm.test\_evaporation.EvaporationTest: Test all options provided in EVAPORATION section | 10 | 10 | 0 | 0 | [Detail](javascript:showClassDetail('c20',10)) |
| test.core.swmm.test\_temperature.TemperatureTest: Test TEMPERATURE section | 8 | 8 | 0 | 0 | [Detail](javascript:showClassDetail('c21',8)) |
| test.core.swmm.test\_adjustments.AdjustmentsTest: Test ADJUSTMENT section in climatology | 4 | 4 | 0 | 0 | [Detail](javascript:showClassDetail('c22',4)) |
| test.core.swmm.test\_aquifers.SimpleAquifersTest: Test AQUIFERS section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c23',2)) |
| test.core.swmm.test\_lid\_controls.SimpleLIDControlTest: Test LID\_CONTROLS section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c24',3)) |
| test.core.swmm.test\_raingages.SimpleRainGageTest: Test RAINGAGES section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c25',2)) |
| test.core.swmm.test\_snowpack.SimpleSnowPackTest: Test snowpack section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c26',3)) |
| test.core.swmm.test\_subcatchments.SimpleSubcatchmentTest: Test SUBCATCHMENTS section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c27',3)) |
| test.core.swmm.test\_infiltrations.InfiltrationTest: Test INFILTRATION section | 6 | 6 | 0 | 0 | [Detail](javascript:showClassDetail('c28',6)) |
| test.core.swmm.test\_groundwater.SimpleGroundwaterTest: Test GROUNDWATER section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c29',2)) |
| test.core.swmm.test\_lid\_usage.SimpleLIDUsageTest: Test LIDUSAGE section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c30',2)) |
| test.core.swmm.test\_coverages.SimpleCoverageTest: Test COVERAGES section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c31',3)) |
| test.core.swmm.test\_loadings.SimpleLoadingTest: Test LOADINGS section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c32',2)) |
| test.core.swmm.test\_hydrographs.SimpleHydrographsTest | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c33',2)) |
| test.core.swmm.test\_conduits.SimpleConduitTest: Test CONDUIT section, MTP3 | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c34',2)) |
| test.core.swmm.test\_xsections.SimpleCrossSectionTest: Test XSECTIONS section | 7 | 7 | 0 | 0 | [Detail](javascript:showClassDetail('c35',7)) |
| test.core.swmm.test\_xsection.XsectionTest | 1 | 1 | 0 | 0 | [Detail](javascript:showClassDetail('c36',1)) |
| test.core.swmm.test\_transects.SimpleTransectTest: Test TRANSECTS section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c37',3)) |
| test.core.swmm.test\_junctions.SimpleJunctionTest: Test JUNCTIONS section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c38',3)) |
| test.core.swmm.test\_outfalls.SimpleOutfallTest: Test OUTFALLS section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c39',2)) |
| test.core.swmm.test\_dividers.SimpleDividerTest: Test DIVIDER section | 5 | 5 | 0 | 0 | [Detail](javascript:showClassDetail('c40',5)) |
| test.core.swmm.test\_inflows.SimpleInflowTest: Test INFLOWS section | 5 | 5 | 0 | 0 | [Detail](javascript:showClassDetail('c41',5)) |
| test.core.swmm.test\_dwf.SimpleDWITest: Test DryWeatherInflow(DWF) Section | 4 | 4 | 0 | 0 | [Detail](javascript:showClassDetail('c42',4)) |
| test.core.swmm.test\_RDII.SimpleRDIITest: Test RDII section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c43',2)) |
| test.core.swmm.test\_treatment.SimpleTreatmentTest | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c44',3)) |
| test.core.swmm.test\_pollutants.SimplePollutantTest: Test POLLUTANTS section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c45',2)) |
| test.core.swmm.test\_buildup.SingleBuildupTest: Test BUILDUP section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c46',2)) |
| test.core.swmm.test\_washoff.SimpleWashoffTest: Test WASHOFF section | 2 | 2 | 0 | 0 | [Detail](javascript:showClassDetail('c47',2)) |
| test.core.swmm.test\_landuses.SimpleLanduseTest: Test LANDUSES section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c48',3)) |
| test.core.swmm.test\_title.SimpleTitleTest: Test TITLE section | 6 | 6 | 0 | 0 | [Detail](javascript:showClassDetail('c49',6)) |
| test.core.swmm.test\_timeseries.SimpleTimeSeriesTest: Test TIMESERIES section | 4 | 4 | 0 | 0 | [Detail](javascript:showClassDetail('c50',4)) |
| test.core.swmm.test\_patterns.SimplePatternTest: Test PATTERNS section | 6 | 6 | 0 | 0 | [Detail](javascript:showClassDetail('c51',6)) |
| test.core.swmm.test\_curves.SimpleCurveTest: Test CURVERS section | 3 | 3 | 0 | 0 | [Detail](javascript:showClassDetail('c52',3)) |
| **Total** | **160** | **160** | **0** | **0** |  |

**APPENDIX B-2**

**SAMPLE BUG REPORT**

*This page was intentionally left blank.*

| MTP | Tested Class | Tested functions/features | Procedures | Results | Expected Results (if different from text\_test) | Pass? | If Bug, Bug Severity | Date | Bug Status |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
| MTP2 | Curve | Storage curve | test\_text = **"AC1 STORAGE 0 1000 2 2000 4 3500 6 4200 8 5000"** | "AC1 STORAGE 0 1000" |  | No | Severe | 2016/5/5 | Fixed, 5/12/2016 |
|  |  | Pump1 curve | test\_text = **"PC1 PUMP1\nPC1 100 5 300 10 500 20"** | "PC1 PUMP1\nPC1 100 5" |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| MTP2 | TimeSerie | Time Series data | test\_text = "TS1 6-15-2001 7:00 0.1 8:00 0.2\nTS1 6-21-2001 4:00 0.2 5:00 0.0" | Exported as the multi-line input format. actual = '"TS1 6-15-2001 7:00 0.1  TS1 8:00 0.2  TS1 6-21-2001 4:00 0.2  TS1 5:00 0.0 '" |  | Yes? | According to manual, this is OK |  |  |
|  |  |  |  | matches() Not match |  |  |  |  |  |
|  |  |  | Input using the multi-line input format. Compare actual\_text produced by get\_text() but removing the white spaces test\_text = "TS1\t6-15-2001\t7:00\t0.1\n" \  "TS1\t\t8:00\t0.2\n" \  "TS1\t6-21-2001\t4:00\t0.2\n" \  "TS1\t\t5:00\t0.0" | Compare actual\_text produced by get\_text() but removing the white spaces, same |  | Yes |  |  |  |
|  |  |  |  | matches() Not match |  |  | Moderate, issue is in match(), not in data model of TimeSerie | 2016/5/5 | Fixed, 5/12/2016 |
|  |  | Time Series from file | test\_text = **"TS1 FILE myfile.txt"** | matches() Match |  | Yes |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| MTP2 | Pattern |  |  | matches() not found |  |  | It is not inherited from Section, do not have the Match method | 2016/5/5 | Fixed, 5/12/2016 |
|  |  | DAILY | test\_text = "D1\tDAILY\t1.0\t1.0\t1.0\t1.0\t1.0\t0.5\t0.5" | D1 DAILY 1.0 1.0 1.0 1.0 1.0 0.5   D1 0.5 ' |  | No | Moderate tab between D1 and DAILY is being replaced by spaces | 2016/5/5 | Fixed, 5/12/2016 |
|  |  | MONTHLY | Data cross multiple rows |  |  |  | Same as above |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Hydrograph |  | test\_text = "UH101\tRG1\n" \  "UH101\tALL\tSHORT\t0.033\t1.0\t2.0\t0.033\t1.0\t2.0\n" \  "UH101\tALL\tMEDIUM\t0.300\t3.0\t2.0\t0.033\t1.0\t2.0\n" \  "UH101\tALL\tLONG\t0.033\t10.0\t2.0\t0.033\t1.0\t2.0\n" \  "UH101\tJUL\tSHORT\t0.033\t0.5\t2.0\t0.033\t1.0\t2.0\n" \  "UH101\tJUL\tMEDIUM\t0.011\t2.0\t2.0\t0.033\t1.0\t2.0" | Compare actual\_text produced by get\_text() but removing the white spaces , same |  | Yes |  |  |  |
|  |  |  |  | matches() Match |  |  |  |  |  |
|  | Transect | Normal input with GR on two lines | test\_text ="NC\t0.016\t0.016\t0.016\n" \  "X1\tFull\_Street\t7\t0.0\t0.0\t0.0\t0.0\t0.0\t0.0\t0.0\n" \  "GR\t1.3\t-40\t0.5\t-20\t0\t-20\t0.8\t0\t0\t20\n" \  "GR\t0.5\t20\t1.3\t40" | matches() Match |  | Yes |  |  |  |
|  | Transects | Normal input with two transects |  | Does not match and failed visual inspection |  | No | Severe, Output does not match input, only one transect was kept and GRs gets combined | 2016/5/5 |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Coverages | Single coverage test | test\_text = "S2 Residential\_1 27" | match each field |  | No | Severe, set\_text set the input test to value, not individual field | 2016/5/16 |  |
|  |  |  |  | matches() does not match |  | No |  |  |  |
|  | Loading | Single loading test | test\_text = "SB1 TSS 0.1 Lead 0.01 " | matches() does not match |  | No | Severe, set\_text missing the second pair of chemical/initial build up | 2016/5/16 |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Inflow | Single inflow test | test\_text = "NODE2 FLOW N2FLOW " | match each field, the fourth field - I have trouble finding the parameter FLOW |  | No | Moderate, need to find the field for the fourth field | 2016/5/16 |  |
|  |  |  |  | matches() does not match, because defaults not provided in test\_text, but written in output |  |  |  |  |  |
|  |  |  | test\_text = "NODE65 BOD N65BOD MASS 126" | match each field, the fourth field - I have trouble finding the parameter |  |  |  |  |  |
|  |  |  |  | matches() does not match, because defaults not provided in test\_text, but written in output |  |  |  |  |  |
|  |  |  | test\_text = "N176 FLOW FLOW176 FLOW 1.0 0.5 12.7 FlowPat" | match each field, the fourth field - I have trouble finding the parameter |  |  |  |  |  |
|  |  |  |  | matches() match |  | Y |  |  |  |
|  | Dividers | Single divider test | test\_text = "NODE10 0 LK1 OVERFLOW 0 0 0 0" | actual\_text = 'NODE10 0 LK1 OVERFLOW 0 0' |  | N | Severe, set\_text missing the DivLink field. | 2016/5/16 |  |
|  |  |  |  | match each field. Missing DivLink field, then everything gets shifted. |  | N |  |  |  |
|  |  |  |  | matches(), no |  | N |  |  |  |

**APPENDIX C-1**

**REGRESSION TEST REPORT: EPANET**

*This page was intentionally left blank.*

EPANET REGRESSION TEST REPORT:

|  |  |  |
| --- | --- | --- |
| \Net1.rpt | Pass | Net1.rpt successful matching |
| \Net2.rpt | Pass | Net2.rpt successful matching |
| \Net2\_Morph.rpt | Pass | Net2\_Morph.rpt successful matching |
| \Net3.rpt | Pass | Net3.rpt successful matching |
| \Net3\_(BWSN-2)\_Morph\_Error\_Free.rpt | Pass | Net3\_(BWSN-2)\_Morph\_Error\_Free.rpt successful matching |

**APPENDIX C-2**

**REGRESSION TEST REPORT: SWMM**

*This page was intentionally left blank.*

SWMM REGRESSION TEST REPORT

|  |  |  |
| --- | --- | --- |
| \Example1.rpt | Pass | Example1.rpt successful matching |
| \Example1a.rpt | Pass | Example1a.rpt successful matching |
| \Example1b.rpt | Pass | Example1b.rpt successful matching |
| \Example1c.rpt | Pass | Example1c.rpt successful matching |
| \Example1d.rpt | Pass | Example1d.rpt successful matching |
| \Example1e.rpt | Pass | Example1e.rpt successful matching |
| \Example1f.rpt | Pass | Example1f.rpt successful matching |
| \Example1g.rpt | Pass | Example1g.rpt successful matching |
| \Example1h.rpt | Pass | Example1h.rpt successful matching |
| \Example1Nodes.rpt | Pass | Example1Nodes.rpt successful matching |
| \Example2.rpt | Pass | Example2.rpt successful matching |
| \Example3.rpt | Pass | Example3.rpt successful matching |
| \Example4.rpt | Pass | Example4.rpt successful matching |
| \Example4a.rpt | Pass | Example4a.rpt successful matching |
| \Example5.rpt | Pass | Example5.rpt successful matching |
| \Example6.rpt | Pass | Example6.rpt successful matching |
| \extran1\extran1.rpt | Pass | extran1.rpt successful matching |
| \extran10\extran10.rpt | Pass | extran10.rpt successful matching |
| \extran2\extran2.rpt | Pass | extran2.rpt successful matching |
| \extran3\extran3.rpt | Pass | extran3.rpt successful matching |
| \extran4\extran4.rpt | Pass | extran4.rpt successful matching |
| \extran6\extran6.rpt | Pass | extran6.rpt successful matching |
| \extran7\extran7.rpt | Pass | extran7.rpt successful matching |
| \extran8\extran8a.rpt | Pass | extran8a.rpt successful matching |
| \extran8\extran8b.rpt | Pass | extran8b.rpt successful matching |
| \extran9\extran9.rpt | Pass | extran9.rpt successful matching |
| \test1\test1.rpt | Pass | test1.rpt successful matching |
| \test2\test2.rpt | Pass | test2.rpt successful matching |
| \test3\test3.rpt | Pass | test3.rpt successful matching |
| \test4\test4.rpt | Pass | test4.rpt successful matching |
| \test5\test5.rpt | Pass | test5.rpt successful matching |
| \user1\user1.rpt | Pass | user1.rpt successful matching |
| \user2\user2.rpt | Pass | user2.rpt successful matching |
| \user3\user3.rpt | Pass | user3.rpt successful matching |
| \user4\user4.rpt | Pass | user4.rpt successful matching |
| \user5\user5.rpt | Pass | user5.rpt successful matching |
| \web\Example1-Adjustments.rpt | Pass | Example1-Adjustments.rpt successful matching |
| \web\Example1-DirectInflow-DryInflow-RDII-InitBuildup.rpt | Pass | Example1-DirectInflow-DryInflow-RDII-InitBuildup.rpt successful matching |
| \web\Example1-Post.rpt | Pass | Example1-Post.rpt successful matching |
| \web\Example1-Pre.rpt | Pass | Example1-Pre.rpt successful matching |
| \web\Example1-Transect.rpt | Pass | Example1-Transect.rpt successful matching |
| \web\Example1-Treatment.rpt | Pass | Example1-Treatment.rpt successful matching |
| \web\Example2-Post.rpt | Pass | Example2-Post.rpt successful matching |
| \web\Example3.rpt | Pass | Example3.rpt successful matching |
| \web\Example4.rpt | Pass | Example4.rpt successful matching |
| \web\Example5-3Aquifers.rpt | Pass | Example5-3Aquifers.rpt successful matching |
| \web\Example5-EMC.rpt | Pass | Example5-EMC.rpt successful matching |
| \web\Example5-EXP.rpt | Pass | Example5-EXP.rpt successful matching |
| \web\Example6-Final.rpt | Pass | Example6-Final.rpt successful matching |
| \web\Example6-Initial.rpt | Pass | Example6-Initial.rpt successful matching |
| \web\Example7-Final.rpt | Pass | Example7-Final.rpt successful matching |
| \web\Example7-Initial.rpt | Pass | Example7-Initial.rpt successful matching |
| \web\Example8.rpt | Pass | Example8.rpt successful matching |
| \web\Example9.rpt | Pass | Example9.rpt successful matching |

**APPENDIX D**

**COVERAGE REPORT**

*This page was intentionally left blank.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Stmts | Miss | Cover - Unit + Regression | Miss | Cover - Unit only |
|  |  |  |  |  |  |
| \src\core\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\coordinate.py | 17 | 0 | 100% | 0 | 100% |
| \src\core\epanet\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\epanet\calibration.py | 313 | 265 | 15% | 265 | 15% |
| \src\core\epanet\curves.py | 16 | 0 | 100% | 0 | 100% |
| \src\core\epanet\epanet\_project.py | 71 | 5 | 93% | 34 | 52% |
| \src\core\epanet\hydraulics\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\epanet\hydraulics\control.py | 19 | 0 | 100% | 6 | 68% |
| \src\core\epanet\hydraulics\link.py | 52 | 3 | 94% | 22 | 58% |
| \src\core\epanet\hydraulics\node.py | 62 | 0 | 100% | 18 | 71% |
| \src\core\epanet\inp\_reader\_project.py | 95 | 5 | 95% | 50 | 47% |
| \src\core\epanet\inp\_reader\_sections.py | 536 | 64 | 88% | 256 | 52% |
| \src\core\epanet\inp\_writer\_project.py | 81 | 7 | 91% | 42 | 48% |
| \src\core\epanet\inp\_writer\_sections.py | 319 | 34 | 89% | 89 | 72% |
| \src\core\epanet\labels.py | 20 | 0 | 100% | 9 | 55% |
| \src\core\epanet\options\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\epanet\options\backdrop.py | 15 | 0 | 100% | 0 | 100% |
| \src\core\epanet\options\energy.py | 22 | 3 | 86% | 3 | 86% |
| \src\core\epanet\options\hydraulics.py | 46 | 0 | 100% | 0 | 100% |
| \src\core\epanet\options\map.py | 6 | 2 | 67% | 2 | 67% |
| \src\core\epanet\options\options.py | 11 | 0 | 100% | 0 | 100% |
| \src\core\epanet\options\quality.py | 49 | 11 | 78% | 11 | 78% |
| \src\core\epanet\options\reactions.py | 15 | 0 | 100% | 0 | 100% |
| \src\core\epanet\options\report.py | 25 | 2 | 92% | 2 | 92% |
| \src\core\epanet\options\times.py | 24 | 0 | 100% | 11 | 54% |
| \src\core\epanet\patterns.py | 7 | 0 | 100% | 0 | 100% |
| \src\core\epanet\title.py | 7 | 0 | 100% | 0 | 100% |
| \src\core\epanet\vertex.py | 5 | 2 | 60% | 2 | 60% |
| \src\core\indexed\_list.py | 107 | 68 | 36% | 74 | 31% |
| \src\core\inp\_reader\_base.py | 219 | 34 | 84% | 47 | 79% |
| \src\core\inp\_writer\_base.py | 155 | 21 | 86% | 76 | 51% |
| \src\core\metadata.py | 62 | 27 | 56% | 27 | 56% |
| \src\core\project\_base.py | 141 | 34 | 76% | 55 | 61% |
| \src\core\swmm\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\swmm\climatology.py | 67 | 0 | 100% | 0 | 100% |
| \src\core\swmm\curves.py | 19 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydraulics\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydraulics\control.py | 9 | 1 | 89% | 1 | 89% |
| \src\core\swmm\hydraulics\link.py | 144 | 9 | 94% | 31 | 78% |
| \src\core\swmm\hydraulics\node.py | 110 | 5 | 95% | 16 | 85% |
| \src\core\swmm\hydrology\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydrology\aquifer.py | 20 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydrology\lidcontrol.py | 51 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydrology\raingage.py | 30 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydrology\snowpack.py | 38 | 0 | 100% | 0 | 100% |
| \src\core\swmm\hydrology\subcatchment.py | 165 | 27 | 84% | 27 | 84% |
| \src\core\swmm\hydrology\unithydrograph.py | 18 | 0 | 100% | 0 | 100% |
| \src\core\swmm\inp\_reader\_project.py | 147 | 5 | 97% | 33 | 78% |
| \src\core\swmm\inp\_reader\_sections.py | 1322 | 88 | 93% | 314 | 76% |
| \src\core\swmm\inp\_writer\_project.py | 139 | 6 | 96% | 57 | 59% |
| \src\core\swmm\inp\_writer\_sections.py | 683 | 50 | 93% | 109 | 84% |
| \src\core\swmm\labels.py | 14 | 0 | 100% | 7 | 50% |
| \src\core\swmm\options\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\backdrop.py | 11 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\dates.py | 19 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\dynamic\_wave.py | 29 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\files.py | 18 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\general.py | 44 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\map.py | 22 | 8 | 64% | 8 | 64% |
| \src\core\swmm\options\report.py | 19 | 0 | 100% | 0 | 100% |
| \src\core\swmm\options\time\_steps.py | 15 | 0 | 100% | 0 | 100% |
| \src\core\swmm\patterns.py | 14 | 0 | 100% | 0 | 100% |
| \src\core\swmm\quality.py | 69 | 0 | 100% | 0 | 100% |
| \src\core\swmm\swmm\_project.py | 93 | 5 | 95% | 7 | 92% |
| \src\core\swmm\timeseries.py | 17 | 0 | 100% | 0 | 100% |
| \src\core\swmm\title.py | 6 | 0 | 100% | 0 | 100% |
| \src\core\utility.py | 11 | 4 | 64% | 6 | 45% |
| \test\HTMLTestRunner.py | 200 | 37 | 82% | 37 | 82% |
| \test\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| \_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| all\_unit\_regress\_cmd.py | 45 | 6 | 87% | ------- | ------- |
| epanet\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| epanet\epanetregressiontest.py | 15 | 2 | 87% | ------- | ------- |
| epanet\test\_all.py | 50 | 9 | 82% | 9 | 82% |
| epanet\test\_backdrop.py | 32 | 2 | 94% | 2 | 94% |
| epanet\test\_curves.py | 43 | 2 | 95% | 2 | 95% |
| epanet\test\_demands.py | 36 | 2 | 94% | 2 | 94% |
| epanet\test\_energy.py | 36 | 2 | 94% | 2 | 94% |
| epanet\test\_options.py | 48 | 2 | 96% | 2 | 96% |
| epanet\test\_patterns.py | 57 | 2 | 96% | 2 | 96% |
| epanet\test\_quality.py | 39 | 2 | 95% | 2 | 95% |
| epanet\test\_reactions.py | 31 | 2 | 94% | 2 | 94% |
| epanet\test\_report.py | 34 | 2 | 94% | 2 | 94% |
| epanet\test\_sources.py | 49 | 2 | 96% | 2 | 96% |
| epanet\test\_title.py | 44 | 2 | 95% | 2 | 95% |
| regression\_base.py | 116 | 28 | 76% | ------- | ------- |
| section\_match.py | 191 | 66 | 65% | 94 | 51% |
| swmm\\_\_init\_\_.py | 0 | 0 | 100% | 0 | 100% |
| swmm\swmmregressiontest.py | 15 | 2 | 87% | 2 | 93% |
| swmm\test\_RDII.py | 29 | 2 | 93% | 2 | 94% |
| swmm\test\_adjustments.py | 34 | 2 | 94% | ------- | ------- |
| swmm\test\_all.py | 191 | 9 | 95% | 9 | 95% |
| swmm\test\_aquifers.py | 59 | 2 | 97% | 2 | 97% |
| swmm\test\_backdrop.py | 35 | 4 | 89% | 4 | 89% |
| swmm\test\_buildup.py | 27 | 2 | 93% | 2 | 93% |
| swmm\test\_conduits.py | 27 | 2 | 93% | 2 | 93% |
| swmm\test\_coverages.py | 36 | 2 | 94% | 2 | 94% |
| swmm\test\_curves.py | 40 | 2 | 95% | 2 | 95% |
| swmm\test\_dividers.py | 52 | 2 | 96% | 2 | 96% |
| swmm\test\_dwf.py | 39 | 2 | 95% | 2 | 95% |
| swmm\test\_evaporation.py | 88 | 2 | 98% | 2 | 98% |
| swmm\test\_files.py | 89 | 7 | 92% | 7 | 92% |
| swmm\test\_groundwater.py | 27 | 2 | 93% | 2 | 93% |
| swmm\test\_hydrographs.py | 70 | 2 | 97% | 2 | 97% |
| swmm\test\_infiltrations.py | 92 | 2 | 98% | 2 | 98% |
| swmm\test\_inflows.py | 101 | 2 | 98% | 2 | 98% |
| swmm\test\_junctions.py | 33 | 2 | 94% | 2 | 94% |
| swmm\test\_landuses.py | 33 | 2 | 94% | 2 | 94% |
| swmm\test\_lid\_controls.py | 34 | 2 | 94% | 2 | 94% |
| swmm\test\_lid\_usage.py | 27 | 2 | 93% | 2 | 93% |
| swmm\test\_loadings.py | 33 | 2 | 94% | 2 | 94% |
| swmm\test\_map.py | 31 | 4 | 87% | 4 | 87% |
| swmm\test\_options\_dates.py | 17 | 2 | 88% | 2 | 88% |
| swmm\test\_options\_dynamicwave.py | 24 | 2 | 92% | 2 | 92% |
| swmm\test\_options\_general.py | 66 | 2 | 97% | 2 | 97% |
| swmm\test\_options\_interfacefiles.py | 23 | 14 | 39% | 14 | 39% |
| swmm\test\_options\_timesteps.py | 23 | 2 | 91% | 2 | 91% |
| swmm\test\_outfalls.py | 27 | 2 | 93% | 2 | 93% |
| swmm\test\_patterns.py | 66 | 2 | 97% | 2 | 97% |
| swmm\test\_pollutants.py | 29 | 2 | 93% | 2 | 93% |
| swmm\test\_raingages.py | 29 | 2 | 93% | 2 | 93% |
| swmm\test\_report.py | 39 | 2 | 95% | 2 | 95% |
| swmm\test\_snowpack.py | 36 | 2 | 94% | 2 | 94% |
| swmm\test\_subcatchments.py | 48 | 13 | 73% | 13 | 73% |
| swmm\test\_temperature.py | 93 | 29 | 69% | 29 | 69% |
| swmm\test\_timeseries.py | 42 | 2 | 95% | 2 | 95% |
| swmm\test\_title.py | 48 | 2 | 96% | 2 | 96% |
| swmm\test\_transects.py | 33 | 2 | 94% | 2 | 94% |
| swmm\test\_treatment.py | 33 | 2 | 94% | 2 | 94% |
| swmm\test\_washoff.py | 28 | 2 | 93% | 2 | 93% |
| swmm\test\_xsection.py | 177 | 2 | 99% | 2 | 99% |
| swmm\test\_xsections.py | 57 | 2 | 96% | 2 | 96% |
| unittest\_all\_with\_cmd.py | 35 | ------- | ------- | 4 | 89% |
| TOTAL | 9126 | 1119 | 88% | 2035 | 77% |

**APPENDIX E**

**SAMPLE UI TEST FILE**

*This page was intentionally left blank.*

# Load Example1 and save to a new file

Start SWMM.

Open the file named 'Example1.inp' using 'File:Open'.

Save the project using 'File:Save As..' with the name 'Example1z7.inp'.

Save the project using 'File:Save As..' with the name 'Example1z8.inp'.

Compare files 'Example1z7.inp' with 'Example1z8.inp'. Should be no difference

Click 'OK'.

From the Project Tree, double-click 'Quality', 'Quality' expanded

# Test 'Quality'> 'Pollutants'

Double-click 'Pollutants' to activate 'Pollutants' Editor

Edit 1: In column 1, Set name='Cd', Units = 'UG\_per\_L', Init.Concenc = '0.101',Click 'OK' to close the Dialog

E1X1: Double-click 'Pollutants' to examine Edit 1, Click 'OK' to close the Dialog.

Edit 2: In column 2, Set Rain Concenc='0.0101', GW Concen = '0.101', Co-Pollutant = 'Cd',Click 'OK' to close the Dialog

E2X1: Double-click 'Pollutants' to examine Edit 2, Click 'OK' to close the Dialog.

# Test 'Quality'> 'Land Uses'

Click 'Land Uses', 'Residential' and 'Undeveloped' appear in the lower-left panel

Edit 3: Double-click 'Residential' to activate the 'Land Use' editor, in tab 'General', Set Interval='7', Availablity = '0.01', Last Swept = '3',Click 'OK' to close the Dialog (Cannot click 'OK', but the edits get saved???)

E3X1: Click 'Land Uses', double-click 'Residential' to examine Edit 3

Edit 4: Double-click 'Residential' to activate the 'Land Use' editor, in tab 'Build up', Set... ,Click 'OK' to close the Dialog (Cannot click 'OK')

E4X1: Click 'Land Uses', double-click 'Residential' to examine Edit 4

Click 'Land Uses', 'Residential' and 'Undeveloped' appear in the lower-left panel

Edit 5: Double-click 'Undeveloped' to activate the 'Land Use' editor, in tab 'Buildup', Set Function='EXP', Max. Build = '0.1',Click 'OK' to close the Dialog (Cannot click 'OK', but the edits get saved???)

E5X1: Click 'Land Uses', double-click 'Undeveloped' to examine Edit 5