**CACIE Tool #04a** – ***RET-2-STOMP (CA\_RET2STOMP.py)***

**Version** **1.0**

**QA**: **QA**

# Description and Purpose

The RET-2-STOMP tool generates the meteoric recharge boundary condition cards for the STOMP model input file using RET output. In addition to the boundary condition cards, the RET-2-STOMP script summarizes the recharge specified in the boundary condition cards in one table for verifying flow balances in the STOMP model. For details on the approach and methodology of the RET-2-STOMP tool, please refer to *Application of the Recharge Evolution Tool (RET) to Prepare Spatially and Temporally Variable Recharge Boundary Conditions for Hanford Site Composite Analysis Vadose Zone Models* ECF-HANFORD-18-0074.

# Functional Requirements

The functional requirements for the RET-2-STOMP tool are as follows:

FR-1: Read in user-specified \*.nij, \*.sij file and \*.top file

*Acceptance Criteria* – test demonstrates the software can correctly read the STOMP grid definition files

FR-2: Read all RET shapefiles within a directory (including subdirectories)

*Acceptance Criteria* – all shapefiles in the user-specified directory are used by the RET-2-STOMP script

FR-3: Assign high-recharge features that may not align with the grid-node centers to the intersecting grid surface(s)

*Acceptance Criteria* – high recharge polygons found in the RET that do not coincide with the node are assigned to the intersecting node surface(s)

FR-4: Create text files for the summary table (also known as the integration table), steady-state and transient boundary conditions cards, and surface files (referenced by the steady-state and transient boundary conditions cards)

*Acceptance Criteria* – text files are produced for the 2 boundary conditions cards as well as the associated surface files and the summary table

FR-5: Export shapefiles of the STOMP grid nodes as points and surfaces with attributes for the selected recharge rate taken from the RET

*Acceptance Criteria* – two folders should be generated, each including a set of files comprising a complete shapefile

FR-6: Filter out features less than 0.01 and buffer all remaining features by 0.1 meters. This addresses a deficiency in the upstream RET output where slivers were present in the calculation.

*Acceptance Criteria* – 2 lines for 1978 in group\_00052.dat of the “…ret/ca\_tr\_boundary\_card.dat” with “-8.5” and “-63” listed in sequence (-8.5 comes before -63).

# Software Requirements Specifications

Python Environment Configuration, using at least Python version 3.6:

| Package Name | Version |
| --- | --- |
| geopandas | 0.4.1 |
| numpy | 1.16.2 |
| pandas | 0.24.2 |
| psutil | 5.6.2 |
| scipy | 1.2.1 |
| shapely | 1.6.4 |

# Software Design Description

The RET-2-STOMP script is written in Python using libraries native to Python. The only expected argument is a single positional argument containing a valid path to a text file containing necessary information for the RET-2-STOMP script to execute.

The input file contains (in order): the directory of the RET shapefiles; the locations for the \*.nij, \*.sij, and \*.top files, respectively; the output folder path and name, and the number of analytes (used to format the STOMP boundary conditions cards). All these user-provided inputs are then accessed by the RET-2-STOMP script for data processing.

On completion of the script, outputs are saved to the specified output folder. The outputs generated (text files and shapefiles) are created using standard methods found in the Python environment.

The following is the shell script configuration that will be passed as an argument to the Tool Runner for qualified runs:

* python ../tools/ca-ret/CA\_RET2STOMP.py ret\_input.txt

# Requirements Traceability Matrix

Table 1 presents the requirements traceability matrix for the RET-2-STOMP tool.

| **Table 5-1. RET-2-STOMP Tool Requirements Traceability Matrix** | | |
| --- | --- | --- |
| **Functional Requirement** | **Acceptance Test** | **Test Case** |
| FR-1  FR-2  FR-3  FR-4  FR-5  FR-6 | CACIE-CA\_RET2STOMP.py-TC-1 | Run the RET-2-STOMP tool to completion and verify that the outputs match outputs documented/archived in ECF-HANFORD-18-0074 (provided for the test plan) |

# Test Plan and Cases

Table 6-1 presents the test plan for the RET-2-STOMP tool. Test cases have been designed to test the functional requirements described in Section 2. The functional requirements corresponding to each test are shown in Table 5-1. In this case, all functional requirements are tested with only one test.

| **Table 6‑1. RET2STOMP Test Plan** | | |
| --- | --- | --- |
| **TEST ID** | **Test Case** | **Test Result  (Pass/Fail)** |
| *Note [*Test\_Repo\_Name*] in acceptance test report*  *Note [Testing\_Directory]\CA-CIE-Tools-Testing\ret2stomp (code repository)* | | |
| CACIE- CA\_RET2STOMP.py-TC-1 | Unzip testing materials to the [Testing\_Directory], which should expand into two folders and corresponding sub-folders:   1. commondata (folder)    1. RET\_shps (folder)       1. 1943 (folder)       2. …       3. 2570 (folder)       4. RET\_shps\_hash.txt 2. ret2stomp\_test (folder)    1. archived\_outputs       1. bc\_gdf (folder)       2. grid\_surfs (folder)       3. ca\_integration\_table.dat       4. ca\_ss\_boundary\_card.dat       5. error.log       6. group\_00000.dat       7. group\_00001.dat       8. …       9. group\_00203.dat       10. mem\_log.txt       11. polys\_to\_restore.txt    2. ss (folder)       1. input.nij       2. input.sij       3. input.top    3. ret\_input.txt | |
| Invoke the following command from inside the [Testing\_Directory]\ret2stomp\_test folder:   * python /opt/tools/pylib/runner/runner.py --logfile CA\_RET2STOMP.py\_TC-1\_logfile.txt --logfilemode w "python" "../CA-CIE-Tools-Testing/tools/ca-ret/CA\_RET2STOMP.py ../../../ret2stomp\_test/ret\_input.txt" > screen.log 2>&1 & | |
| Verify that the “screen.log” file has a line stating: “Obtained parameters and finished setting environment variables” |  |
| A user may monitor the progress of this script using traditional system utilities or they may choose to follow the script’s progress by watching the script auxiliary outputs like the “ret/mem\_log.txt” file. This is not a necessary part of the test and is for informational purposes only but may be helpful to the user. | |
| Once the script has completed, compare output files in the “ret” folder against the “archived\_outputs” folder. Apart from differences in headers (file path differences and dates specifically), the files should be identical.   1. The files that must match include every file beginning with the word “group” and any file beginning with “ca”. Apart from differences in the headers of the “ca” files, all other aspects should be identical.    1. For the boundary conditions cards, lines 10 through the end should be identical in all aspects.    2. The integration table should have identical matches in lines 8-11 and 21 through the end. 2. The files in ret/grid\_surfs/…, ret/bc\_gdf/…, “mem\_log.txt”, “polys\_to\_restore.txt”, and “error.log” do not need to be verified against the corresponding files in “rest\_outputs”, but should be produced by the script. These files are considered auxiliary outputs and do not impact downstream work products. |  |
|  | Open ./ret/ca\_tr\_boundary\_card.dat and search for “group\_00052.dat”. Verify that there are two entries for 1978, one with “-8.5” and another with “-63.0” in the line. |  |

# Acceptance Test Report

All acceptance tests produced the expected results and no deviations from the test plan were needed. The complete test log is included as an attachment to this document. Testing procedures and results are described in Section 7.1 through 7.2. The test was completed by Jose Lopez of INTERA as of January 20th, 2020.

**7.1 Comparing Integration Table and Boundary Condition Cards**

The integration table and boundary condition cards produced by the test run were compared to the archived files using a file compare program. The headers for each file was removed before comparing the files. Figures 7.1 through 7.3 shows the results of the program indicating the test and archived files are identical.

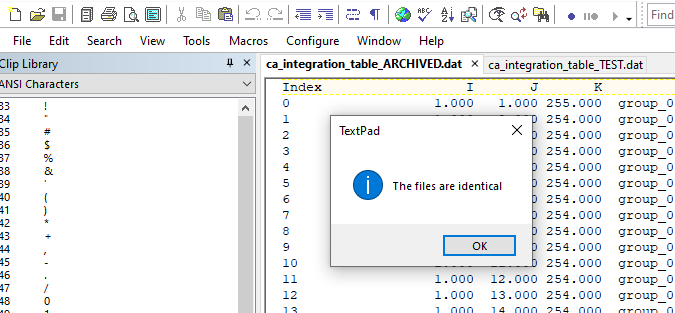


Figure 7.1 Comparing Integration Tables

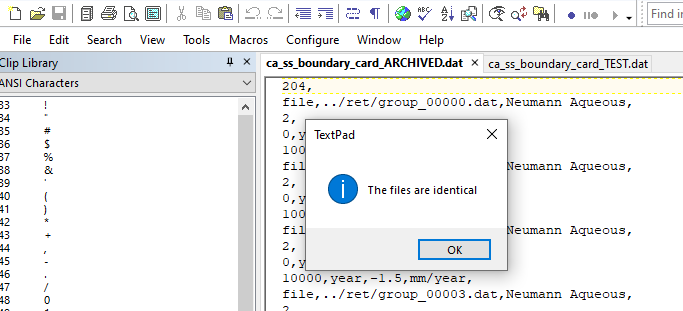


Figure 7.2 Comparing SS Boundary Condition Cards

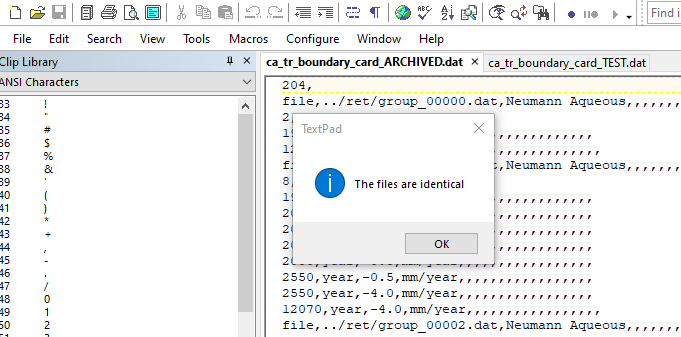


Figure 7.3 Comparing TR Boundary Condition Cards

**7.2 Comparing Group Files**

Group files (group\_xxxxx.dat) produced by the test run were compared to the archived files using the Fingerprinter tool. The group files were placed in their own respective folders to only fingerprint the group files. Fingerprints for test and archived group files were identical. Log files of the Fingerprinter tool are included as an attachment to this document.

# User Guide

The following is a guide for using this script:

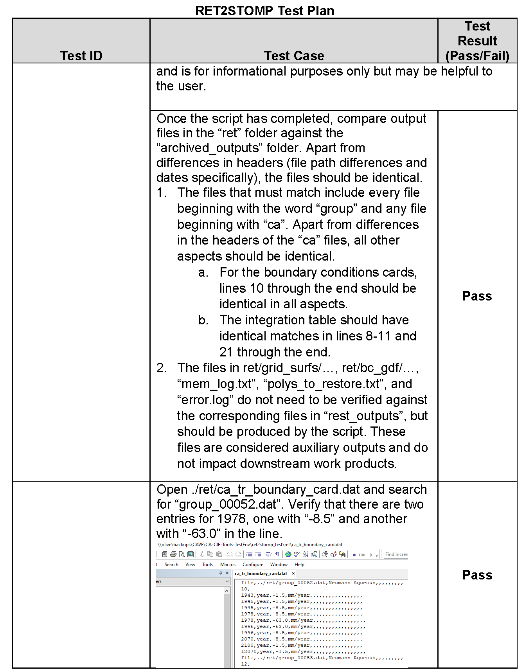
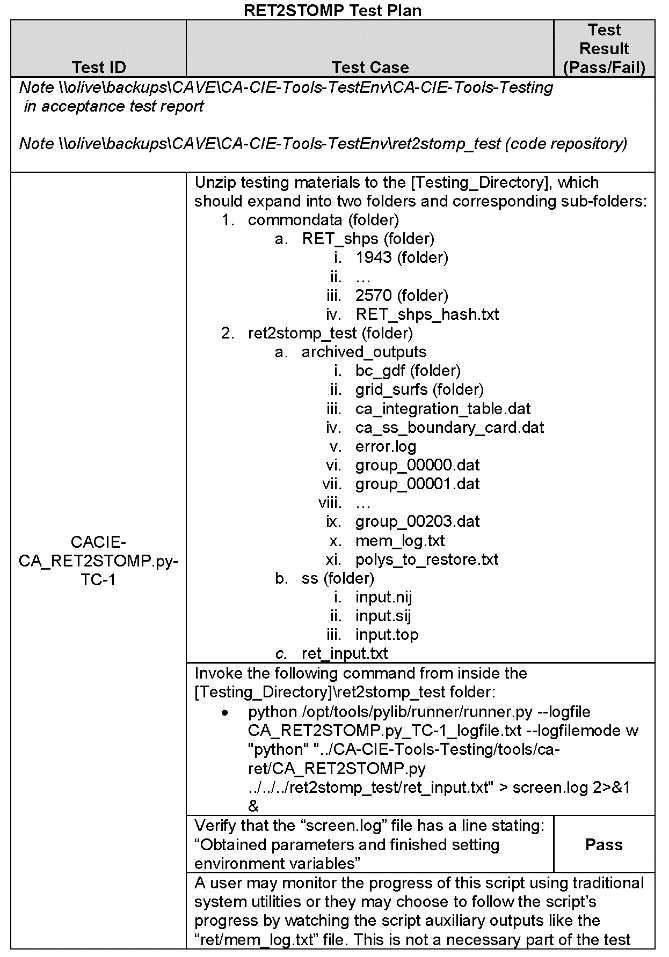
1. Copy the following from the CAVE/v4-2/shells directory to the main model directory, CAVE/v4-2/ModelName
2. Script ***run\_RET2STOMP.sh***
3. Text file ***ret\_input.txt***
4. A few notes before executing the RET script
5. Ensure there is enough space on Olive for the ***run\_RET2STOMP.sh*** script
6. Make sure to delete or move any existing ret folder in the CAVE/v4-2/ModelName directory
7. Run ***run\_RET2STOMP.sh*** script using a linux command window
8. This will automatically create a ret folder in the model directory.

This script can take several hours to run. To verify the script is still running check the log file in the CAVE/v4-2/ModelName/ret directory and use the “top” command in a linux command window logged into the Olive server.

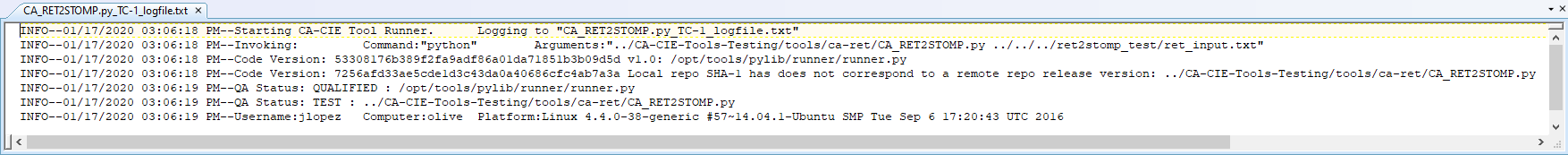
# References

ECF-HANFORD-18-0074, 2020, *Application of the Recharge Evolution Tool (RET) to Prepare Spatially and Temporally Variable Recharge Boundary Conditions for Hanford Site Composite Analysis Vadose Zone Models*, CH2M Hill Plateau Remediation Company, Richland, Washington.

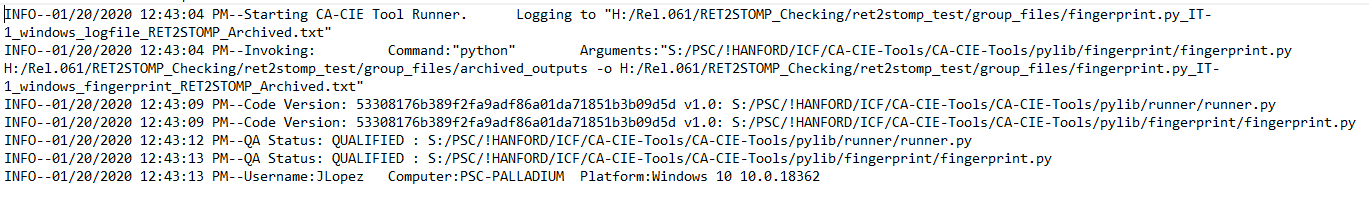
| **Table A-1. RET2STOMP Code Review Summary** | | | |
| --- | --- | --- | --- |
| **Code Line** | **Comment** | **Notes** | **Resolution** |
| -- | Remove Dead code | Remove code that is no longer called/utilized | Handle in next revision |
| -- | Parameterize values instead of hard coding | Put parameters that need to stay the same in a config file in the directory of the script so they are controlled/versioned together. preferable to have an extra optional input param that you can point to a custom config file as well. | Handle in next revision with input from facet lead |
| -- | Stream line code a bit. Lots and lots of looping inside of looping | Pre process columns before appending them to data frame can help | Handle in next revision |
| -- | Use python Log library for logging. | Currently you open a file append comment close the file. the log library will take care of that for you and you can still monitor the log file as the application runs. | Handle in next revision |
| -- | More comments | Some functions have no information as to the purpose of the function.  Some of the code could use comments as to why you are doing things in a specific way. IE Some calcs are done oddly but has to do with missing years or duplicate polys, etc | Handle in next revision |
| 292 | This does nothing as its not setting anything and its not part of an if statement | Param.num\_cocs == False | Delete in next revision |
| 342 | Missing a check if .SIJ file exists | You check for other files but miss this one | Add check for this file in next revision |
| 375 | See if you can replace some of the loops with Numpy functions | Part of reducing total number of loops | Evaluate vectorization options in next revision |
| 493 | When looping through an array looking for the first instance of a specific value, break after finding it, | This is so you don’t loop through extra times you don’t need to. Also may want to try numpy.where | Evaluate vectorization options in next revision, and improve for-loop efficacy by reducing unnecessary loops |



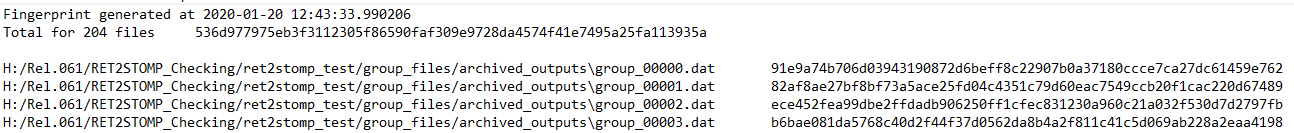
**Test Log A-1. RET2STOMP Acceptance Testing**



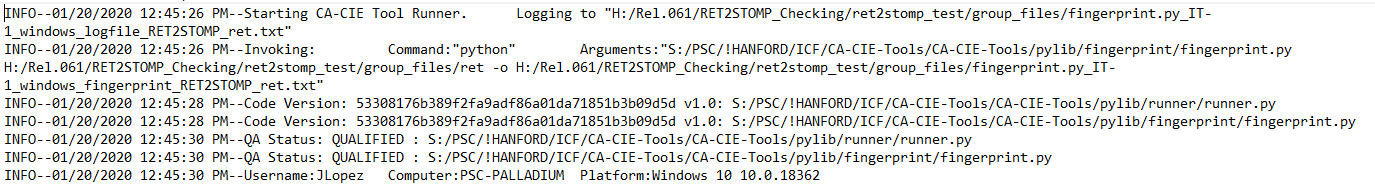
**Test Log A-2. RET2STOMP Acceptance Testing (Invoking Tool with Tool Runner)**



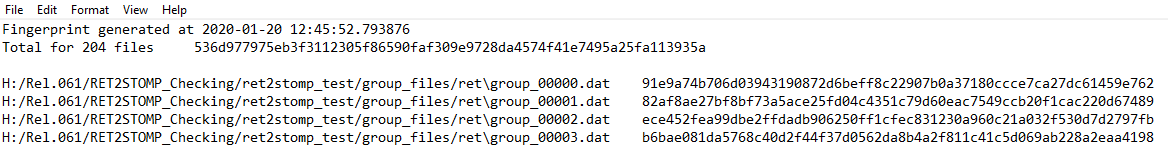
**Test Log A-3 Archived Files Tool Runner Log File**



**Test Log A-4 Archived Group Files Fingerprint**



**Test Log A-5 Test Files Tool Runner Log File**



**Test Log A-6 Test Files Fingerprint**