**CACIE Tool #NN** – **Solid Waste Release Reduction Tool (reducer.py)**

**Version** **1.1**

**QA**: **QA**

# Description and Purpose

The Solid Waste Release (SWR) reduction tool reduces the solid waste contaminant inventory release rates (i.e., flux) for a waste site from an original dataset of approximately 10,000 timesteps and fluxes to a dataset consisting of a reduced number of timesteps and corresponding fluxes. The reduced dataset represents the original inventory release rates and released mass/activity, considering user-defined criteria for maximum relative total mass error, the number of reduced data pairs, the number of reduction iterations, and additional post reduction correction steps.

The Ramer-Douglas-Peucker algorithm is implemented for the reduction of the original datasets. Inputs to the Ramer-Douglas-Peuker algorithm consist of the original dataset (ordered by timestep) and a distance dimension ε > 0. The algorithm recursively divides a line represented by the original dataset. Initially it is given all the points between the first and last point. It automatically marks the first and last point to be kept. It then finds the point that is farthest from the line segment with the first and last points as end points; this point is obviously farthest on the curve from the approximating line segment between the end points. If the point is closer than ε to the line segment, then any points not currently marked to be kept can be discarded without the simplified curve being worse than ε.

If the point farthest from the line segment is greater than ε from the approximation, then that point must be kept. The algorithm recursively calls itself with the first point and the farthest point and then with the farthest point and the last point, which includes the farthest point being marked as kept. When the recursion is completed, the reduced dataset is generated, which consists of all and only those points that have been marked as kept.

Following the reduction using the Ramer-Douglas-Peuker algorithm, additional corrections can be performed if user-defined. Additional timesteps can be added during the reduction when gaps between timesteps increase relative error. Additional timesteps can be added following the final reduction when the difference in cumulative mass exceeds the acceptance criteria for total mass. Additionally, the reduced total mass error (original – reduced) will be reduced to meet acceptance criteria by adjusting the non-peak flux values of the reduced dataset.

Implementation of the reduction methodology and the resulting reduced datasets will be documented in an environmental calculation file, which will document the user-defined inputs for the reduction and the corresponding outputs and results.

# Functional Requirements

The following are the functional requirements (FR) of the SWR Reduction tool:

FR-1: Read in user-defined input values from a JSON-formatted file (.JSON filename provided as an argument to the python script)

FR-2: Read in solid waste release inventory from comma-delimited file

FR-3: Generate summary file of reduced dataset results

FR-4: Reduce original timestep-flux dataset for each waste site-COPC, meeting acceptance criteria for maximum number of data pairs and error thresholds

FR-5: Generate an individual file of reduced timesteps and corresponding flux for each waste site-COPC

FR-6: Generate individual plots of original and reduced flux and original and reduced cumulative mass versus time for each waste site-COPC

FR-7: Adjust nonpeak flux rates to reduce error to within acceptance criteria if reduction iterations are exceeded and the error is greater than the error threshold

FR-8: Adjust fluxes less than a user-defined zero-below threshold to zero prior to reduction

FR-8: Insert additional timesteps to fill gaps between timesteps following reduction to minimize error (if specified by user)

FR-9: Insert additional timesteps to minimize cumulative mass error after reduction (if specified by user). This may result in datasets exceed the maximum number of datapoints.

# Software Requirements Specifications

The programming Language and required modules/libraries for the SWR reduction tool are:

Python 3.5

Python Standard Libraries:  
argparse  
datetime   
json   
logging  
os  
sys

Python Libraries:  
matplotlib.pyplot   
numpy  
pandas  
scipy.signal  
scipy.integrate  
pathlib [Path]

Python Modules:  
pylib\vzreducer\config.py [config, parse\_args]  
pylib\vzreducer\constants.py  
pylib\runner\constants.py [LOGGER\_KEY, LOG\_LEVEL\_MAP]  
pylib\config\config.py [read\_config]  
pylib\autoparse\autoparse.py [config\_parser]  
pylib\autoparse\constants.py  
pylib\vzreducer\constants [contains the python variables assigned to the names of the name/value pairs in the JSON-formatted input file]  
pylib\vzreducer\parse\_input\_file.py [parse\_input\_file]  
pylib\vzreducer\read\_solid\_waste\_release.py [SolidWasteReleaseData]  
pylib\vzreducer\reduce\_dataset.py [reduce\_dataset]  
pylib\vzreducer\summary\_file [get\_summary\_file, summary\_info]  
pylib\timeseries\timeseries.py [TimeSeries]  
pylib\timeseries\timeseries\_math.py  
pylib\vzreducer\reduce\_flux.py  
pylib\vzreducer\plots.py  
pylib\datareduction\rdp  
pylib\datareduction\reduction\_result.py [ReductionResult]  
pylib\pygit\git.py [get\_version]

# Software Design Description

The following is a brief description of the required arguments and the output generated by the SWR Reducer tool.

Arguments:

* Positional Arguments:
  + Input\_File: Path and filename to input file (JSON formatted)
  + Output\_Directory: Path to output folder
* Optional Arguments:
  + -h, --help
  + –loglevel {I,D} verbosity of log: (I)nfo, (D)ebug; default=I
  + --logfile LOGFILE path to a log file (default is stdout)
  + –logfilemode {a,w} Log file mode: (a)ppend or (w)rite; default=w
* Shell file configuration:

python [directory path]/pylib/vzreducer.py [optional arguments—see above] Input\_File Output\_Directory

Input File:

* User-defined input values (JSON-formatted file) example:

{  
 "Source Files":{  
 "200 E":"<path\_filename>.csv",   
 "200 W":"<path\_filename>.csv"  
 },

"Zero Below":"0",  
 "SUMMARY\_FILE\_NAME": "<user\_defined\_filename>.csv",  
 "SUMMARY\_TEMPLATE":"{copc},{site},{N},{ix},{used\_eps:.2g},{orig\_total\_mass:.7e},{reduced\_total\_mass:.7e},{unbal\_mass\_err:.2g},{unbal\_rel\_err:.2g},{bal\_mass\_err:.2g},{bal\_rel\_err:.2g}",  
 "SUMMARY\_HEADER":["COPC","SITE","N reduced","N Iterations","Epsilon","Original Total Mass","Reduced/Rebalanced Total Mass","Unbalanced Total Mass Error (Ci) (Original-Reduced)","Total Mass Relative Percent Error [before rebalance]","Rebalanced Total Mass Error(Ci) (Original-Reduced)","Total Mass Relative Percent Error [after rebalance]"

],  
 "SUMMARY\_MODE":"a/w",  
 "COPCs": [<comma-delimited string(s) in double quotes>],  
 "Waste Sites": [<comma-delimited strings(s) in double quotes>],  
 "Mass Threshold":"<float>",  
 "Output Lower Error Threshold":"<float>",  
 "Output Upper Error Threshold":”<float>",  
 "Lower Reduced Datapoint Limit":"<integer>",  
 "Upper Reduced Datapoint Limit":"<integer>",  
 "Maximum Iterations":"<integer>",  
 "Maximum Error Iterations":"<integer",  
 "Epsilon":"<float>",  
 "Close Gaps":"<Boolean>",  
 "Gap Delta": "<integer>",  
 "Gap Steps":"<integer>",  
 "Diff Mass Correction":"<Boolean>"  
}

Output Files:

Summary File (user-defined\_filename.csv): summarizes the following for each reduced dataset

* COPC
* Waste site
* Number of reduced data pairs
* Number of reduction iterations performed
* Epsilon value used in final reduction
* Original total cumulative mass (Ci)
* Reduced and rebalanced total cumulative mass (Ci)
* Unbalanced total mass error (Ci) (original-reduced),
* Total mass relative percent error [before rebalance],
* Rebalanced total mass error (Ci) (original-reduced),
* Total mass relative percent error [after rebalance]

Reduced dataset file (WasteSite\_COPC.csv) generated for each waste-COPC with non-zero inventory:

* Site Name
* Date Created
* Script Version (git SHA-1 tag)
* COPC
* Reduced timestep (Year)
* Reduced Activity Release Rate (Ci/year)

Plot .png file (COPC\_WasteSite.png) generated for each waste-COPC with non-zero inventory:

* Site Name
* COPC
* Plot of original (input) flux, reduced flux, and flux difference (original – reduced) versus time
* Plot of original (input) cumulative mass (Ci), reduced cumulaltive mass (Ci), and cumulative mass difference (original – reduced) versus time

Tool Runner:

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

python {directory path to the repositoyr}/pylib/vzreducer/reducer.py [optional arguments—see above] Input\_File Output\_Directory

Code Review:

A code review was performed by Mitchell Tufford on 03/04/2020. A summary is provided in Appendix C.

# Requirements Traceability Matrix

The requirements traceability matrix for the Solid Waste Release Reduction tool is presented in Table 1.

| Table 1  Requirements Traceability Matrix | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case** |
| QA Level | CACIE-reducer.py -IT-1 | Installation Test |
| FR-1 | CACIE-reducer.py -TC-1 | Invoke SWR Reduction tool with subset of COPCs and waste sites |
| FR-2 |
| FR-3 |
| FR-4 |
| FR-5 |
| FR-6 |
| FR-7 |
| FR-8 | CACIE-reducer.py -TC-2 | Invoke SWR Reduction tool with and without zero-below threshold |
| FR-9 | CACIE-reducer.py -TC-3 | Invoke SWR Reduction tool with and without decreasing timestep gaps |
| FR-10 | CACIE-reducer.py -TC-4 | Invoke SWR Reduction tool with and without correcting cumulative mass difference |

# Installation Test Plan and Acceptance Test Plan Cases

The installation test plan for Solid Waste Release Reduction tool is presented in Table 2 and the acceptance test plan cases for Solid Waste Release Reduction tool are presented in Tables 3, 4, and 5.

| Table 2  **Solid Waste Release Reduction Tool Installation Test Plan** | | | |
| --- | --- | --- | --- |
| **Solid Waste Release Reduction Tool Installation Solid Waste Release Reduction Testing**  **CACIE-reducer.py– IT-1** | | **Date:** | |
| **Tool Runner File Location for this test:**  **[PUT LINK TO THE DIRECTORY HERE]** | | **Test Performed By: [FIRST & LAST NAME]** | |
| **Testing Directory: [PROVIDE LINK TO TESTING DIRECTORY]** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Tools Code Repository Directory: | | | |
| Navigate to the testing directory | | | |
| 1 | Invoke Tool runner and test the tool by entering the following:  *./runner\_run\_reducer.py\_IT-1\_reducer.py.bat* | | |
| 2 | Verify Tool Runner is invoked and executed. |  |  |
| 3 | Verify Solid Waste Release Reduction tool is invoked and executed. |  |  |

| Table 3  **Solid Waste Release Reduction Tool Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **Solid Waste Release Reduction Acceptance Testing**  **CACIE-reducer.py– TC-1** | | **Date:** | |
| **Tool Runner File Location for this test:**  **[PUT LINK TO THE DIRECTORY HERE]** | | **Test Performed By:** | |
| **Testing Directory: [PROVIDE LINK TO TESTING DIRECTORY]** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory\CA-CIE-reducer.py-TC-1 | | | |
| 1 | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-1\_windows.bat  Note:  Log files are saved to the Testing Directory\ CACIE-reducer.py-TC-1 folder  Output files (summary, dataset, and plots) are saved to Testing Directory\CACIE-reducer.py-TC-1\output\_CACIE-reducer.py-TC-1 | | |
| 2 | Open log file logfile\_CACIE-reducer.py-TC-1.txt in a text editor | | |
| 3  FR-1 | Verify that the text in the CACIE-reducer.py-TC-1\_input.JSON file corresponds to the log information on Line 3. | The text in the .JSON file is the same as the text on Line 3 of the log file |  |
| 4 | Search for “COPCS in” | Should correspond to two lines in log file | |
| 5  FR-2 | Verify that there are 16 COPCs listed on both lines and the filename is the same as the filename in the JSON file | 16 COPCs (U-235, Np-237, Tc-99, H-3, Th-230, Ra-226, C-14, I-129, U-233, U-236, U-238, Sr-90, Cl-36, U-232, Re-187, U-234)  Same filename in log file and .JSON file |  |
| 6 | Search for “Sites in” | Should correspond to two lines in log file | |
| 7 FR-2 | Verify there are 107 sites in 200E Area and 130 sites in 200W Area and the filename is the same as the filename in the JSON file | Number of sites correspond to expect waste site counts  Same filename in log file and .JSON file |  |
| 8 FR-3 | Verify summary file is generated and use summary file to verify steps 9 and 10 | CACIE-reducer.py-TC-1\_summary.csv |  |
| 9 FR-5 | Verify relative error meets acceptance criteria in the .JSON file | For total mass > 0.1 Ci: relative % error ≤ 0.1% For total mass ≤ 0.1 Ci: relative percent error ≤ 1% |  |
| 10 FR-5 | Verify number of data pairs meets acceptance criteria in .JSON file | Data pairs ≤ 50 |  |
| 11 FR-6 | Verify there is a dataset file and a plot file generated for each waste site-COPC listed in the .JSON file | There should be 9 dataset files and 9 plot files (3 waste sites and 3 COPCs are listed in .JSON file for testing purposes) |  |
| 12 | Using summary file, select a waste site that required nonpeak flux adjustments | Total Mass Relative Percent Error [before rebalance] ≠ Total Mass Relative Percent Error [after rebalance] | |
| 13 FR-7 | Using the dataset file (located in Testing Directory\CACIE-reducer.py-TC-1\output\_CACIE-reducer.py-TC-1) and the corresponding input file (located in Testing Directory\ CACIE-reducer.py-TC-1\inputs), verify that flux values at a peak timestep in reduced dataset are equal to the flux value at the same timestep in the input file | Reduced dataset flux = Input dataset flux |  |

| Table 4  **Solid Waste Release Reduction Tool Acceptance Test Plan Case 2** | | | |
| --- | --- | --- | --- |
| **Solid Waste Release Reduction Acceptance Testing**  **CACIE-reducer.py– TC-2** | | **Date:** | |
| **Tool Runner File Location for this test:**  **[PUT LINK TO THE DIRECTORY HERE]** | | **Test Performed By:** | |
| **Testing Directory: [PROVIDE LINK TO TESTING DIRECTORY]** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-2\_windows.bat  Note:  Log files are saved to the Testing Directory\ CACIE-reducer.py-TC-2 folder  Output files (summary, dataset, and plots) are saved to Testing Directory\CACIE-reducer.py-TC-2\output\_CACIE-reducer.py-TC-2  Make sure before running that the initial value of “Zero Below” in CACIE-reducer.py-TC-2\_input.json is empty string “”. | | |
| 2 | Open dataset file and based on the flux values in the reduced dataset, select a threshold value to zero out fluxes that are less than the selected value. | | |
| 3 | Rename and save file with a “\_nonzeroed” suffix | | |
| 4 | Open the CACIE-reducer.py-TC-2\_input.json file and enter selected value on the following line:  Your Value:  "Zero Below":"{Your Value Here}" | | |
| 5 | Save file with the same filename | | |
| 6 | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-2\_windows.bat | | |
| 7 FR-8 | Open dataset file and verify flux values are all greater than the threshold value | Reduced fluxes > threshold value |  |
| 9 | The zero below functionality can also be verified in log file as well | | |

| Table 5  **Solid Waste Release Reduction Tool** **Acceptance Test Plan Case 3** | | | |
| --- | --- | --- | --- |
| **Solid Waste Release Reduction Acceptance Testing**  **CACIE-reducer.py– TC-3** | | **Date:** | |
| **Tool Runner File Location for this test:**  **[PUT LINK TO THE DIRECTORY HERE]** | | **Test Performed By:** | |
| **Testing Directory: [PROVIDE LINK TO TESTING DIRECTORY]** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-3\_windows.bat  Note:  Log files are saved to the Testing Directory\ CACIE-reducer.py-TC-3 folder  Output files (summary, dataset, and plots) are saved to Testing Directory\CACIE-reducer.py-TC-3\output\_CACIE-reducer.py-TC-3  Make sure before running that the initial value of “Close Gaps” in CACIE-reducer.py-TC-3\_input.json is “True”. | | |
| 2 | Rename generated dataset and plot files with a “\_gapclosed” suffix | | |
| 4 | Open the CACIE-reducer.py-TC-3\_input.json file and enter False on the following line as shown here:  "Close Gaps":"False", | | |
| 5 | Save file with the same filename | | |
| 6 | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-3\_windows.bat | | |
| 7 | Rename generated dataset and plot files with a “\_gaps” suffix | | |
| 8 FR-8 | Open plot files (\_gapclosed and \_gaps) and verify the additional timesteps in \_gapclosed and the omission of the timesteps in \_gaps | \_gapclosed file has additional timesteps consistent with the parameters in the .JSON file |  |

| Table 6  **Solid Waste Release Reduction Tool Acceptance Test Plan Case 4** | | | | |
| --- | --- | --- | --- | --- |
| **Solid Waste Release Reduction Acceptance Testing**  **CACIE-reducer.py Tool– TC-4** | | | **Date:** | |
| **Tool Runner File Location for this test:**  **[PUT LINK TO THE DIRECTORY HERE]** | | | **Test Performed By:** | |
| **Testing Directory: [PROVIDE LINK TO TESTING DIRECTORY]** | | | | |
| **Test Step** | **Test Instruction** | | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | | |
| 1 | | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-4\_windows.bat  Note:  Log files are saved to the Testing Directory\ CACIE-reducer.py-TC-4 folder  Output files (summary, dataset, and plots) are saved to Testing Directory\CACIE-reducer.py-TC-4\output\_CACIE-reducer.py-TC-4  Make sure before running that the initial value of “Diff Mass Correction” in CACIE-reducer.py-TC-4\_input.json is “False”. | | |
| 2 | | Rename generated dataset and plot files with a “\_nodiffcorr” suffix | | |
| 4 | | Open the CACIE-reducer.py-TC-4\_input.json file and enter False on the following line as shown here:  "Diff Mass Correction":"True" | | |
| 5 | | Save file with the same filename | | |
| 6 | | Invoke the SWR Reducer tool by entering the following:  CACIE-reducer.py-TC-4\_windows.bat | | |
| 7 | | Rename generated dataset and plot files with a “\_diffcorr” suffix | | |
| 8 FR-9 | | Open plot files (\_diffcorr and \_nodiffcorr) and verify the additional timesteps in \_diffcorr and the mass difference in \_nodiffcorr | \_diffcorr file has additional timesteps consistent with the parameters in the .JSON file |  |

# Acceptance Test Report

To complete the Acceptance Testing use Appendix A. The test cases are described as follows:

* Acceptance Test Plan Case 1 is in Table A-1.
* Acceptance Test Plan Case 2 is in Table A-2.
* Acceptance Test Plan Case 3 is in Table A-3.
* Acceptance Test Plan Case 4 is in Table A-4.

Details of these tests, when they were conducted, by whom, and if they Passed or Failed are in each table of Appendix A. Test logs and supporting files have been archived.

# User Guide

The SWR Reduction tool is invoked using the optional and required arguments defined in Section 4 Software Design. The SWR Reduction tool requires the user to define the following parameters specified in the JSON-formatted input file:

* Source Files: the directory paths and filenames for input files (comma-delimited) containing the original solid waste contaminant inventory
* Zero Below: flux values less than this value are set to 0 [zero]; if not defined (i.e. “”), all flux values are retained
* SUMMARY\_FILE\_NAME: user-defined filename for the summary file generated
* SUMMARY\_TEMPLATE: user-defined formatting (python string) for the summary output file (enclosed in double quotes); user may edit the numerical formatting; example:
  + {copc},{site},{N},{ix},{used\_eps:.2g},{orig\_total\_mass:.7e},{reduced\_total\_mass:.7e},{unbal\_mass\_err:.2g},{unbal\_rel\_err:.2g},{bal\_mass\_err:.2g},{bal\_rel\_err:.2g}"
* SUMMARY\_HEADER: user-defined column headers (python list of strings) for summary output file (enclosed in brackets); example:
  + ["COPC","SITE","N reduced","N Iterations","Epsilon","Original Total Mass","Reduced/Rebalanced Total Mass","Unbalanced Total Mass Error (Ci) (Original-Reduced)","Total Mass Relative Percent Error [before rebalance]","Rebalanced Total Mass Error(Ci) (Original-Reduced)","Total Mass Relative Percent Error [after rebalance]"]
* SUMMARY\_MODE: user-defined; “a” = append, “w” = write
* COPCs: a user-defined subset of COPCs; each COPC enclosed in double quotes (“”) and comma-delimited; if not defined (i.e., [ ]), all COPCs listed in a solid waste inventory release input file are reduced.
* Waste Sites: a user-defined subset of waste sites; each waste site enclosed in double quotes (“”) and comma-delimited; if not defined (i.e., [ ]), all waste sites in a solid waste inventory release input file are processed through the reduction.
* Mass Threshold: user-defined threshold used to determine corresponding error threshold for a reduced dataset.
* Output Lower Error Threshold: acceptable relative error threshold for the reduced “total released mass” relative to the original “total released mass“;
  + inventory> Mass Threshold, relative error ≤ Output Lower Error Threshold.
* Output Upper Error Threshold: acceptable relative error threshold for the reduced “total released mass” relative to the original “total released mass“;
  + inventory ≤ Mass Threshold, relative error ≤ Output Lower Error Threshold
* Lower Reduced Datapoint Limit: minimum number of datapoints in the reduced dataset.
* Upper Reduced Datapoint Limit: maximum number of datapoints in the reduced dataset (note: this value can be exceeded if error threshold cannot be met with in the maximum number of reduction iterations specified—see below)
* Maximum Iterations: maximum number of reduction iterations to be performed
* Maximum Error Iterations: maximum number of error correction iterations to be performed
* Epsilon: beginning tolerance value for reduction algorithm
* Close Gaps: user-defined if gaps in timesteps in reduced dataset following reduction are to be minimized; “True” or “False”
* Gap Delta: user-defined tolerance for maximum timestep gap following reduction
* Gap Steps: number of timesteps to be inserted if gaps are to be minimized
* Diff Mass Correction: user-defined if additional correction to be applied when maximum relative error in cumulative mass (original – reduced) is greater than applicable error threshold; note this correction may result in additional timesteps greater than Upper Reduced Datapoint Limit

# Tool Versions

This section details changes incorporated into each version of the **Solid Waste Release Reduction** tool.

* 1.0 – Initial release.
* 1.1 – Revise to remove requirement to invoke reducer from within a folder in git repository; impacted reduce\_dataset.py (added pathlib library and additional code)

# Appendix A

**Completed Acceptance Test Cases**

# Appendix C

**Code Review**

|  | **Table C-1. Data Reducer Issues/Improvements** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Issue #** | **Code Line** | **Comment** | **Function Impact** | **Suggested Change** | **Resolution** |
| 1 | constants.py line 13-14 | Inconsistent use of tabs and spaces in file. | Some python environments may not execute this file correctly. | Change tabs to spaces |  |

**Table C-2. Data Reducer  
Code Review History**

| **Date** | **Reviewer** | **Comments** |
| --- | --- | --- |
| **03/04/2020** | Mitchell Tufford | Issue #1 identified. No impacts to other repository tools or library dependencies were discerned. The HSSMBuilder tool refers to the Timeseries library, but the update in this revision was unit tests only, not the library code. |