**CACIE Tool #NN** – ***Solid Waste Reduction Tool***

**Version** **1.0**

**QA**: **TEST** or **NA** or **QA**

1. **Description and Purpose**

One or two paragraphs describing the tool’s function and purpose.

The solid waste reduction tool reduces the solid waste contaminant inventory and release rates (i.e., flux) for a waste site from an original dataset to a dataset consisting of a reduced number of timesteps. The reduced dataset adequately represents the original inventory release rates and released mass/activity, considering established criteria for relative error for total mass etc. Implementation of the reduction and the resulting reduced datasets will be documented in an environmental calculation file, which will summarize user-defined inputs and the corresponding outputs and results.

User-defined parameters for the data reduction include:

* 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
* COPCs: a user-defined list of COPCs; each COPC enclosed in double quotes (“”); if not defined (i.e., [ ]), all COPCs listed in an inventory input file are processed through the reduction.
* Waste Sites: a user-defined list of waste sites; each waste site enclosed in double quotes (“”); if not defined (i.e., [ ]), all waste sites in the solid waste inventory files are processed through the reduction.
* SUMMARY\_TEMPLATE: user-defined formatting (python string) for the summary output file (enclosed in double quotes); example:
  + ["{copc},{site},{reduced\_number},{total\_mass\_error:.2g},{model\_mass\_error:.2g},{total\_mass:.3g},{mass\_error\_factor:.0f},{relative\_mass\_error:.1g}\n"
* SUMMARY\_HEADER: user-defined column headers (python list of strings) for summary output file (enclosed in brackets); example:
  + ["COPC", "Site", "N points", "Total Error", "Model Mass Error", "Total Mass", "Mass Error Factor"]
* Output Error Threshold: acceptable relative error threshold for the reduced “total released mass” relative to the original “total released mass.“
* 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)
* Minimum Iterations: minimum number of reduction iterations to be performed
* Maximum Iterations: maximum number of reduction iterations to be performed
* Peak Height: minimum difference between two datapoints in order to be considered a “peak” to eliminate differences due to noise at extremely low flux values.
* Flux Floor: PENDING
* Delta Slope Tolerance: placeholder (not currently used in code)
* Solve Type: “SMOOTH” or “RAW”; Specifying “SMOOTH” invokes a signal processing filter to reduce the background noise associated with the original dataset. Specifying “RAW” maintains the dataset as unfiltered.

1. **Functional Requirements**

The functional requirements of the tool will be documented in this section. Each requirement will have an ID, such as: FR-N, where N starts at 1 and increments for each Functional Requirement. Each of the Functional Requirement IDs will have a corresponding test ID listed in the RTM.

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

FR-2: Read in solid waste inventory (reference Ryan’s model and outputted files)—Source Files:{200 E: ; 200 W:}; provided as .csv files.

FR-3:

1. **Software Requirements Specifications**

The software requirements specification of the tool will be documented in this section.

Python 3.5

Python Standard Libraries:  
argparse  
datetime   
json   
logging  
os  
sys

Python Libraries:  
matplotlib.pyplot   
numpy  
pandas  
scipy.signal  
scipy.integrate

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 and ….<<<additional variables—not sure where they belong>>>

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 [reset\_summary\_file, summary\_info]

pylib\timeseries\timeseries.py [TimeSeries]

pylib\timeseries\timeseries\_math.py

pylib\vzreducer\reduce\_flux.py

pylib\vzreducer\plots.py

neils code imports...

pylib\datareduction\reduction\_result.py [ReductionResult]

pylib\pygit\git.py [get\_version]

1. **Software Design Description**

The software design description of the tool will be documented in this section. The results of a Code Walkthrough with an independent third party will be summarized in this section.

* Positional Arguments:
  + Input\_File: Path 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

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

{

"Source Files":{

"200 E":"directory\_path\\200East\_CASWR\_Ouput\_20190806.csv",

"200 W":"directory\_path\\200West\_CASWR\_Ouput\_20190806.csv"

},

"Zero Below":"0",

"SUMMARY\_TEMPLATE":"{copc},{site},{reduced\_number},{total\_mass\_error:.2g},{model\_mass\_error:.2g},{total\_mass:.3g},{mass\_error\_factor:.0f},{relative\_mass\_error:.1g}\n",

"SUMMARY\_HEADER":[

"COPC", "Site", "N points", "Total Error", "Model Mass Error", "Total Mass", "Mass Error Factor"

],

"COPCs": [],

"Waste Sites": [],

"Output Error Threshold":"1e-2",

"Lower Reduced Datapoint Limit":"30",

"Upper Reduced Datapoint Limit":"50",

"Minimum Iterations": "0",

"Maximum Iterations":"80",

"Maximum Error Iterations":"100",

"Flux Floor":"1e-15",

"Delta Slope Tolerance":"0.1",

"Solve Type": "SMOOTH"

}

Flux Floor USED IN CODE???? May move some from config.json here…TBD

| Header | | Header | | Header | Header | Header |
| --- | --- | --- | --- | --- | --- | --- |
| JSON Input File (user-defined) | | vzreducer\constants.py | | Description | Value type |  |
| Source Files | 200 E | SOURCE\_FILES\_KEY | \_200E\_KEY | 200E-specific solid waste source inventory file | csv filename |  |
| 200 W | \_200W\_KEY | 200W-specific solid waste source inventory file | csv filename |  |
| Zero Below | | ZERO\_BELOW\_KEY | |  | value |  |
| SUMMARY\_TEMPLATE | | SUMMARY\_TEMPLATE\_KEY | | Values for Output Files |  |  |
| SUMMARY\_HEADER | | ??????????????????????????????? | | Header Text for Output Files |  |  |
| COPCs | | COPCS\_KEY | | Specific COPCs to reduce (default is all COPCs) |  |  |
| Waste Sites | | WASTE\_SITES\_KEY | | Specific waste sites to reduce (default is all waste sites) |  |  |
| Output Error Threshold | | OUT\_ERROR\_THRESHOLD\_KEY | | Acceptable relative difference threshold ([original mass – reduced mass]/original mass) |  |  |
| Lower Reduced Datapoint Limit | | UPPER\_N\_KEY | | Maximum number of datapoints in reduced dataset |  |  |
| Upper Reduced Datapoint Limit | | LOWER\_N\_KEY | | Minimum number of datapoints in reduced dataset |  |  |
| Minimum Iterations | | MIN\_ITERATIONS\_KEY | | Minimum number of reduction iterations |  |  |
| Maximum Iterations | | MAX\_ITERATIONS\_KEY | | Maximum number of reduction iterations |  |  |
| Maximum Error Iterations | | MAX\_ERR\_ITERATIONS\_KEY | | Maximum number of iterations for distributing the difference in mass across the reduced dataset |  |  |
| Peak Height | | PEAK\_HEIGHT\_KEY | | Minimum difference between fluxes to be considered a peak or valley during data reduction |  |  |
| Flux Floor | | FLUX\_FLOOR\_KEY | | NOT SURE IF CURRENTLY USED IN CODE… |  |  |
| Delta Slope Tolerance (?) | | DELTA\_SLOPE\_TOL\_KEY (commented out) | |  |  |  |
| Solve Type | | SOLVE\_TYPE\_KEY | |  |  |  |
| JSON Config File (pylib\vzreducer\config.json) | | pylib\autoparse\constants.py | |  |  |  |
| LOGGER | |  | |  |  |  |
| ARGPARSE\_SECTION | | ARGPARSE\_SECTION\_KEY | |  |  |  |
| ARGUMENTS | | ARGUMENTS\_KEY | |  |  |  |
| ARG | | ARG | |  |  |  |
| SMOOTHING | | SMOOTH\_KEY | |  |  |  |
| BUTTERWORTH\_FACTOR | | BUTTER\_INDEX\_KEY | |  |  |  |
| CUTOFF\_FREQUENCY | | CUTOFF\_FREQUENCY | |  |  |  |
| PLOTS | | PLOTS\_KEY | |  |  |  |
| RAW | | RAW | |  |  |  |
| SMOOTHED | | SMOOTHED | |  |  |  |
| ERROR | | ERROR | |  |  |  |
| COLOR | | COLOR | |  |  |  |
| SYMBOL | | SYMBOL | |  |  |  |
| ERROR\_TITLE | | ERROR\_TITLE | |  |  |  |
| SIGNAL\_TITLE | | SIGNAL\_TITLE | |  |  |  |
| AVERAGE\_ERROR\_LABEL | | AVERAGE\_ERROR\_LABEL | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |
|  | |  | |  |  |  |

1. **Requirements Traceability Matrix**

A requirements traceability matrix for the tool will be documented in this section. At a minimum, the matrix will include IDs of: Functional Requirements and the corresponding Acceptance Test, along with an indication of the test result (Pass/Fail).

1. **Test Plan and Cases**

The test plan for the tool will be documented in this section. Each test will have a unique ID and criteria for determining if the test result is pass or fail. The TEST ID will be referenced in the RTM and ATR. An installation test, labeled **IT-1**, will be used by the Tool Runner to confirm the version of the tool being used is running correctly before launching it with the user’s parameters.

The Unit Testing done on the tool will be documented here, also.

1. **Acceptance Test Report**

The test report will state whether the tool is qualified for use, summarize test case results, and report all resolved incidents and resolution of unresolved incidents.

1. **User Guide**

A guide for using the tool will be documented in this section.