**CACIE Tool** – **ca-sumdosedb**

**Version** **1.0**

**QA**: **QA**

# Description and Purpose

This tool is equivalent to ca-sumdose except that it uses a database implementation to accommodate larger data sets than is possible with the ca-dumdose tool. Where possible, we recommend using this tool over ca-sumdose.

The ca-sumdosedb tool aggregates the dose from various contaminants of potential concern (COPC) for an exposure pathway at each point in space and time in a MODFLOW/MT3D model.

Given:

1. A control file specifying paths to input files (each of these are assumed to be outputs of a ca-dosecalc for a COPC)

This utility loads each of the input files and computes the total dose for each exposure pathway by summing dose contributions from all COPCs at each row, layer, and column for every timestep.

# Functional Requirements

Ca-dosecalc has the following functional requirements (FR):

FR-1: Consume the output of ca-dosecalc

FR-2: Consume an input control file that specifies dose outputs for each COPC of interest

FR-3: Compute the total dose for an exposure pathway by summing the pathway dose for each COPC specified in the input control file. This sum should be computed for every row, layer, column, timestep, and exposure pathway.

FR-4: Output results to a comma-separated value (CSV) file

# Software Requirements Specifications

Where there are no explicit restrictions on Windows provided dependencies are installed, it is assumed that the tool will operate on a Linux (Ubuntu) machine. The following modules/libraries must be installed:

* Python 3.6+
* PostgreSQL 10+ (12 is preferred)

# Software Design Description

This tool is intended to be invoked by the CA/CIE Tool Runner on a Linux machine. The command signature looks like:

$ python3 *[path to repository]*/pylib/casumdose/sumDoseDB.py inputControlFile.json

Arguments:

This tool accepts one input argument: a string indicating the path to the input control file.

Input Files:

**Input control file**

This file provides the tool with important input, such as where to find the dose data. It must be in JavaScript Object Notation (JSON) format and have the following structure:

{

"outputFile":"/absolute/path/to/output/outputFile.csv",

“dbname”:”aNameForTheDatabase”,

"doseFiles":[

{"copc":"trit", "fpath":"/absolute/path/to/dose/trit/dose.csv"},

{"copc":"U235", "fpath":"/absolute/path/to/dose/u235/dose.csv"},

{"copc":"U238", "fpath":"/absolute/path/to/dose/u238/dose.csv"}

]

}

The field names are required and must be lower case. Users may specify additional fields, such as comments, but they are ignored by this tool.

* outputFile
  + desired absolute path to the file produced by this tool. The user postgres must have write access to the directory where this file will be located.
* dbname
  + the name of the database that will be created during the calculation. Generally this is not important, but if you anticipate executing multiple simultaneous runs of this tool you must provide a unique name for each instance of this tool.
* doseFiles
  + A list of objects that define the COPC and location of the input dose file(s). Each entry in the list must be an object with the following fields (additional fields are ignored):
    - copc
      * A label used to identify the dose data
    - fpath
      * An absolute path to the dose data. This is expected to be the output of ca-dosecalc; there is generally one dose file per COPC.

Output Files:

This tool produces a single output CSV file as defined in the outputFile parameter in the input control file. The structure of the file is like the output of ca-dosecalc except that there is an additional column containing the pathway dose for each COPC and a final column denoting the total pathway dose from all COPCs. Each row in the file corresponds to a unique timestep, row, layer, column and exposure route.

The first row is header text with column names. The number of columns depends on the number of COPCs specified in the input control file.

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Column name** | **Type** | **Description** |
| 1 | Pathway | Text | The exposure pathway |
| 2 | elapsed\_tm | Integer | The elapsed time in days |
| 3 | cell\_layer | Integer | Grid index (1-based) |
| 4 | cell\_row | Integer | Grid index (1-based) |
| 5 | cell\_column | Integer | Grid index (1-based) |
| 6 | model\_date | Text | The calendar date in YYYY-MM-DD format corresponding to elapsed\_tm. The calendar date is calculated as the model start date (determined by the modeldate input parameter) plus elapsed\_tm calendar days. Leap years are observed. |
| 7 through (6 + NCOPC) | [COPC name] | Float | The pathway dose for each COPC specified in the input control file. Units are identical to the units in input dose files. There is one column for every COPC up to the total number of input files (NCOPC) specified in the input control file |
| 6+NCOPC + 1 | Dose | Float | The calculated total dose for a pathway. Units are identical to the units in the input dose file. The value of this column equals the sum of the previous NCOPC columns |

File size depends strongly on the number of grid cells, timesteps, and exposure pathways; for the output produced by saturated zone model, output is typically on the order of 8GB.

Tool Runner:

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

$ python3 *[path to tools]*/pylib/casumdose/sumDoseDB.py inputControlFile.json

Code Review:

Code walkthrough was performed by Neil Powers on 08/20/2020. No impacts to other repository tools or shared library dependencies were identified for the sumDoseDB tool.

# Requirements Traceability Matrix

The requirements traceability matrix for the ca-sumdosedb tool is presented in Table 1.

| Table  Requirements Traceability Matrix | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case – step** |
| QA Level | CACIE-ca-dosecalc-IT-1 | 1-1: Tool was invoked and executed |
| FR-1 | CACIE- ca-dosecalc-AT-1 | 1-1, 1-3: Consume as inputs files in the format produced by ca-calcDose |
| FR-2 | CACIE-ca-dosecalc-AT-1 | 1-4: Verify the input control file works as expected |
| FR-3 | CACIE-ca-dosecalc-AT-1 | 1-5, 1-6: verify the total pathway dose is computed for each row, layer, column, timestep and exposure pathway |
| FR-4 | CACIE-ca-dosecalc-AT-1 | 1-5: Verify output is CSV |

# Installation Test Plan and Acceptance Test Plan Cases

This tool will be tested on the machine that it will be used on. The installation test and the acceptance tests are therefore the same test and presented in Table 2.

| Table 2  **ca-sumdosedb Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **ca-sumdosedb Acceptance Testing**  **CACIE-ca-sumdosedb – AT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **~/dose/test-sumDoseDB/test/outputs/runlog.txt** | | **Test Performed By:** | |
| **Testing Directory: ~/dose/test-sumDoseDB/test** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Remote into the machine holding the tool as the “ca” user using the password provided by the developer  Navigate to the Testing Directory | | | |
| 1 | Run the script by typing ‘./runAT1.sh’ into the command shell and let the program run to completion. | The program runs to completion without an error |  |
| 2 | type ls outputs -alh and press enter | The output directory contains two files:   * runlog.txt * dose.csv |  |
| 3 | Type  cat inputs/testControlAT1.json  and press enter | Text is displayed. The field “outputFile” contains the path to the dose.csv reported in the previous test step |  |
| 4 | Type  cat inputs/testControlAT1.json  and press enter | There are three “copc” entries: d1, d2, and d3 |  |
| 5 | Open outputs/dose.csv and verify the following: | The first line contains “d1,d2,d3,dose” as the last four column headers in the first line of the output |  |
| Every entry in the last column is equal to the sum of the previous three columns; assume zero if there is a blank entry in a previous column |  |
| The file is a CSV |  |
| 6 | Open outputs/dose.csv and verify the following for model date 1/1/2019: | There is one row for pathway “A Route” (column 1) at layer=1, row=1, column=1, with doses of 6,80,5 for d1,d2,d3 |  |
| There is one row for pathway “B Route” (column 1) at layer=1, row=1, column=1, with a dose of 7 for d1 |  |
| There is one row for pathway “A Route” (column 1) at layer=2, row=1, and column=1, with doses of 12,99,12 for d1,d2,d3 |  |
| There is one row for pathway “A Route” (column 1) at layer=1, row1, column=2, with a dose of 18 for d1 |  |
| Open outputs/dose.csv and verify the following for model date 1/2/2019: | There is one row for pathway “A Route” (column 1) at layer =1, row=2, column=1, with doses of 24,110,15 for d1,d2,d3 |  |
| Open outputs/dose.csv and verify the following for model date 1/3/2019: | There is one row for pathway “A Route” (column 1) at layer=1, row1, column=1, with a dose of 30 for d1 |  |

# Acceptance Test Report

To complete the Acceptance Testing use Appendix A. Acceptance Test 1 is in Table A-1.

Details of these tests, when they were conducted, by whom, and if they Passed or Failed are in each table of Appendix A.

# User Guide

Ca-sumdose is intended to be invoked on the command line by the CA/CIE tool runner. It aggregates (sums) multiple files produced by runs of ca-dose. Depending on the size of files involved, this tool may require significant resources to run efficiently.

In general, the size of the output file will roughly equal the largest dose input file; typically, these are on the order of 16GB.

Compared to ca-dosecalc, ca-dosecalcDB uses a database implementation that reduces the amount of RAM needed. It should be possible to run several instances of ca-dosecalcDB simultaneously without exhausting the system’s available resources.

It is important that the user postgres be given write access to the directory where the output file will be written.

# Tool Versions

This section details changes incorporated into each version of the **ca-sumdosedb** tool.

* 1.0 – Tool was developed.

# Appendix

**Completed Acceptance Test Cases**

**Tool Runner Log**

INFO--08/20/2020 03:30:09 PM--Starting CA-CIE Tool Runner. Logging to "outputs/runlog.txt"

INFO--08/20/2020 03:30:09 PM--Code Version: 83fd29e41185e0f8b8560c5b83469c1e189a5931 v4.2: /home/ca/CA-CIE-Tools/pylib/runner/runner.py<--1bcfd6779e9cbdb82673405873a8e5e81514ae27

INFO--08/20/2020 03:30:09 PM--Code Version: 6e385e30e8fe573e0d2033124e7e1a6743c33d4d Local repo SHA-1 has does not correspond to a remote repo release version: /home/ca/dose/test-sumDoseDB/CA-CIE-Tools/pylib/casumdose/sumDoseDB.py<--3b36a233570ec1fee36942a339da8a24be87434b

INFO--08/20/2020 03:30:09 PM--QA Status: QUALIFIED : /home/ca/CA-CIE-Tools/pylib/runner/runner.py

INFO--08/20/2020 03:30:09 PM--QA Status: TEST : /home/ca/dose/test-sumDoseDB/CA-CIE-Tools/pylib/casumdose/sumDoseDB.py

INFO--08/20/2020 03:30:09 PM--Invoking Command:"python3" with Arguments:"/home/ca/dose/test-sumDoseDB/CA-CIE-Tools/pylib/casumdose/sumDoseDB.py inputs/testControlAT1.json"

INFO--08/20/2020 03:30:09 PM--Username:ca Computer:twotbbase Platform:Linux 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020

| Table 2  **ca-sumdosedb Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **ca-sumdosedb Acceptance Testing**  **CACIE-ca-sumdosedb – AT-1** | | **Date:** 8/20/2020 | |
| **Tool Runner Log File Location for this test:**  **~/dose/test-sumDoseDB/test/outputs/runlog.txt** | | **Test Performed By:** Christian Hall | |
| **Testing Directory: ~/dose/test-sumDoseDB/test** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Remote into the machine holding the tool as the “ca” user using the password provided by the developer  Navigate to the Testing Directory | | | |
| 1 | Run the script by typing ‘./runAT1.sh’ into the command shell and let the program run to completion. | The program runs to completion without an error | PASS |
| 2 | type ls outputs -alh and press enter | The output directory contains two files:   * runlog.txt * dose.csv | PASS |
| 3 | Type  cat inputs/testControlAT1.json  and press enter | Text is displayed. The field “outputFile” contains the path to the dose.csv reported in the previous test step | PASS |
| 4 | Type  cat inputs/testControlAT1.json  and press enter | There are three “copc” entries: d1, d2, and d3 | PASS |
| 5 | Open outputs/dose.csv and verify the following: | The first line contains “d1,d2,d3,dose” as the last four column headers in the first line of the output | PASS |
| Every entry in the last column is equal to the sum of the previous three columns; assume zero if there is a blank entry in a previous column | PASS |
| The file is a CSV | PASS |
| 6 | Open outputs/dose.csv and verify the following for model date 1/1/2019: | There is one row for pathway “A Route” (column 1) at layer=1, row=1, column=1, with doses of 6,80,5 for d1,d2,d3 | PASS |
| There is one row for pathway “B Route” (column 1) at layer=1, row=1, column=1, with a dose of 7 for d1 | PASS |
| There is one row for pathway “A Route” (column 1) at layer=2, row=1, and column=1, with doses of 12,99,12 for d1,d2,d3 | PASS |
| There is one row for pathway “A Route” (column 1) at layer=1, row1, column=2, with a dose of 18 for d1 | PASS |
| Open outputs/dose.csv and verify the following for model date 1/2/2019: | There is one row for pathway “A Route” (column 1) at layer =1, row=2, column=1, with doses of 24,110,15 for d1,d2,d3 | PASS |
| Open outputs/dose.csv and verify the following for model date 1/3/2019: | There is one row for pathway “A Route” (column 1) at layer=1, row1, column=1, with a dose of 30 for d1 | PASS |