**CACIE Tool #NN** – **CA-IPP Tool**

**Version** **1.0**

**QA**: **QA**

# Description and Purpose

The CA-IPP tool’s purpose is to create a comprehensive data set consisting of radionuclide and aqueous volume releases as a function of time for Central Plateau sites. The data set consists of a modified compilation of a series of data sources, consisting of:

a. **SIMv2 Release**: The data set provides the radionuclide and liquid inventory release estimates from site inception to closure.

b. **Solid Waste Release**: The data set consists of modified (i.e., reduction of number of time steps through interpolation) output from the Composite Analysis Solid Waste Release Model. The Solid Waste Release data files consist of a summary file, and the associated *wastesite\_radionuclide.csv* files (e.g.,*T31\_Tc-99.csv*).

c. **Rerouted Inventory Release:** The data set provides the rerouted radionuclide and liquid release originating from the U-10 and B-3 Pond sites, and the releases from the SALDS site.

d. **SAC Liquid Release**: The data set contain SAC liquid releases for non-contaminated effluent discharges.

The waste and liquid-only sites included in the comprehensive release data set are all part of the **VZEHSIT**, a compilation of waste sites and their vertices corresponding to their footprint boundary extents.

This iteration of qualifying the inventory preprocessor, CA-IPP, will omit qualifying the chemical inventory option (i.e. the chemical inventory argument will be “none” for this set of acceptance tests).

This tool’s specific task is to parse out the relevant information for the CA waste and liquid-only sites to assemble a site list containing radionuclide and water release over time. During the execution of CA-IPP, uranium isotopes will be converted from activity to mass and summed together to provide a total uranium waste inventory on a site-by-site, year-by-year basis.

# Functional Requirements

The following are the functional requirements (FR) of the CA-IPP Tool:

FR-1: Radionuclide and liquid releases are included only if the site is part of the VZEHSIT data set

FR-2: Read in all **SIMv2 Release** records with temporal data (excluding records with no year recorded)

FR-3: Read in all **Rerouted Inventory Release** information, replacing **SIMV2 Release** information where applicable

FR-4: Read in all **Solid Waste Release** information, replacing **SIMv2 Release** output and appending new sites as applicable

FR-5: Convert **SIMv2 Release** records’ source type (e.g. “Solids” vs “Liquid”) to “Liquid” where the “Inventory Module” has the matching string “entrained” (case-insensitive match)

FR-6: Read in the **SAC Liquid Release** only if the site record is not already included. For these sites, *only* the liquid release information is read in.

FR-7: Convert uranium from activity to mass for all isotopes for sites with a “Liquid” waste type

FR-8: **Solid Waste Release** types should be assigned the waste type “Solid Release Series”. All remaining records in the CA-IPP output should have waste type “Liquid”

# Software Requirements Specifications

The Perl programming language was used to develop this script. No libraries outside of the standard distribution of the Perl v5.18.2 interpreter were used.

# Software Design Description

Arguments:

The tool is executed from the command line in a Linux terminal in the following manner (positional argument numbers are explained below by the corresponding numbered list):

$ perl ca-ipp.pl [1] [2] [3] [4] [5] [6] [7] [8]

1. File path (including file name and extension) to the **VZEHSIT** data file
2. File path (including file name and extension) to the **SIMv2 Release** data file
3. File path (including file name and extension) to the **SAC Liquid Release** data file
4. File path (including file name and extension) to the chemical inventory file (for CA applications, using “none” to exclude)
5. Directory path to the **Solid Waste Release** data files
6. File path (including file name and extension) to the **Solid Waste Release** index/summary file
7. File path (including file name and extension) to a file containing a newline-delimited list of file paths (each including the file name and extension) for the **Rerouted Inventory Release** information
8. Output filename and associated file path (not including the extension) for output file naming pattern and location

Input Files:

Each input file will be described in the same order as their corresponding positional argument for the script:

1. The waste site vertices file is a comma-separated file with a single header line (skipped). Unique values (except for Null or empty strings) are taken from the first column of the file.
2. The **SIMv2 Release** file is a comma-separated file with 4 header lines (2 are comment lines, the 3rd line in the 7th column is a unit descriptor). Column headings are taken from the 4th line in the file. The header row column rows are as follows:
   1. Inventory Module
   2. SIMV2 site name
   3. CA site name
   4. Source Type
   5. Volume [m3]
   6. Discharge/decay-corrected year
   7. C-14
   8. Cl-36
   9. H-3
   10. I-129
   11. Np-237
   12. Re-187
   13. Sr-90
   14. Tc-99
   15. U-232
   16. U-233
   17. U-234
   18. U-235
   19. U-236
   20. U-238
   21. Th-230 (decay only)
   22. Ra-226 (decay only)
3. “none” is provided for this version of the script, which skips parsing any chemical inventory file
4. The **SAC Liquid Release** file is a comma separated file. The first line is a header line whose second column describes the number of waste sites in the file. Each line that has a waste site in the first column also has the number of condition changes in the second column. Each line after that has a year in the first column constitutes a “condition change”.
5. The path to the top-level directory of the solid waste release files (passed as a string)
6. The index file for the **Solid Waste Release** is a comma separated file with the first line as a header (the header is skipped). The columns are as follows:
   1. COPC
   2. SITE
   3. N reduced
   4. N Iterations
   5. Epsilon
   6. Original Total Mass (Ci)
   7. Reduced/Rebalanced Total Mass (Ci)
   8. Unbalanced Total Mass Error (Ci) (Original-Reduced)
   9. Total Mass Relative Percent Error [before rebalance]
   10. Total Mass Relative Percent Error (after rebalance)
7. The newline-delimited list of rerouting information is a text file with Unix-style line endings (“\n”). Each line contains a single path to a file
8. This is not a file, but an argument indicating the base file name for the outputs.

Output Files:

Three files are produced by this tool:

1. <base file name>.csv
   1. This file is the final output. It is a comma-delimited file with 11 header rows containing meta information. The 12th row contains the column names of the file. The 13th row contains the units for each corresponding column (if applicable).
   2. The remainder of the file is a combination of each input file and reflects the functional requirements described/tested in this document.
2. <base file name>.log
   1. Contains more meta information printed/logged by the script processes. This information is not to be QA’d but is useful information for debugging the script.
3. <base file name>-summary.csv
   1. Contains summary information of the total inventory by waste site and some breakdown of the integrated mass over specific time frames. This information, like the log file, is not to be QA’d but is useful information for evaluating the script and its output.

Tool Runner:

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

{directory path to repository}/\tools\ca-ipp \ca-ipp.pl “$EHSIT $RADINV $CHEMINV $LIQINV $SWRDIR $SWRIND $REDFIN $OUTPUT”

Each of these shell script variables (denoted by the “$”) will be set in the shell script with the corresponding variable input.

Code Review:

Neil Powers performed an independent code review on 3/31/2020. No impacts to other repository tools or library dependencies were identified for the CA-IPP tool.

# Requirements Traceability Matrix

The requirements traceability matrix for the CA-IPP tool is presented in Table 1.

| Table  Requirements Traceability Matrix | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case** |
| QA Level | CACIE-ca-ipp.pl -IT-1 | Installation Test |
| FR-1 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will compare all site names of the output from ca-ipp.pl against **VZEHSIT**. If no differences are found, the checking script will print out “##QA-PASS (FR-1)”. |
| FR-2 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will compare all information from the **SIMv2 Release Inventory** file against the ca-ipp.pl output. If no differences are found, the checking script will print out “##QA-PASS (FR-2)”. |
| FR-3 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will compare all records from **Rerouted Inventory Releases** against the ca-ipp.pl output. If no differences are found, the checking script will print out “##QA-PASS (FR-3)”. |
| FR-4 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will compare all information from the **Solid Waste Release** files against the ca-ipp.pl output. If no differences are found, the checking script will print out “##QA-PASS (FR-4)”. |
| FR-5 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will verify that all records whose “Inventory Module” has the matching string “entrained” is listed as a “Liquid” waste type (relevant to the **SIMv2 Release Inventory**) in the ca-ipp.pl output. If no differences are found, the checking script will print out “##QA-PASS (FR-5)”. |
| FR-6 | CACIE- ca-ipp.pl -TC-1 | Execute the checking script (common step for test case). The script will compare all records from the **SAC Liquid Release** file (parsing only liquid inventory) against the ca-ipp.pl output. If no differences are found, the checking script will print out “##QA-PASS (FR-6)”. |
| FR-7 | CACIE- ca-ipp.pl -TC-2 | Perform a spot-check of the conversion from activity to mass for uranium isotopes. |
| FR-8 | CACIE- ca-ipp.pl -TC-3 | Verify that all waste types are assigned correctly. |

# Installation Test Plan and Acceptance Test Plan Cases

The installation test plan for CA-IPP is presented in Table 2 and the acceptance test plan cases for CA-IPP is presented in Table 3, Table 4, and Table 6. Output from acceptance test case 1 will be used in subsequent acceptance test cases.

| Table  **CA-IPP Installation Test Plan** | | | |
| --- | --- | --- | --- |
| **CA-IPP Installation Testing**  **CACIE-CA-IPP – IT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **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 using *runner\_run\_IT-1.sh* as follows:  *./runner\_run\_IT-1\_CA-IPP.sh* | | |
| 2 | Verify Tool Runner is invoked and executed. | Should see exact string: “QA Status: QUALIFIED : /opt/tools/pylib/runner/runner.py” |  |
| 3 | Error messages from the tool should also be recorded in the same log file. | Should see at least one line reading “Use of uninitialized value” |  |

| Table  **CA-IPP Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the “reroute-list” file and modify the path from “USER” to the user’s corresponding username. | | |
| 2 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./runner\_run\_AT-1\_CA-IPP.sh* | 4 files should be created:   1. ***runner\_run\_AT-1\_CA-IPP\_log.txt*** 2. ***preprocessed\_inventory.csv*** 3. ***preprocessed\_inventory.log*** 4. ***preprocessed\_inventory-summary.csv*** |  |
| 3 | Inside of a Linux terminal, invoke the checking script: *./ca-ipp\_check.sh* | The script should produce a file called ***ca\_ipp\_check.log***. |  |
| 4 | FR-1 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-1): The ca-ipp.pl output only has sites listed in the VZEHSIT.”  If this exact string is present, FR-1 has been satisfied by the **CA-IPP** tool. |  |
| 5 | FR-2, FR-5, FR-6 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-2, FR-5, and FR-6)”  If this exact string is present, then FR-2, FR-5, and FR-6 have been satisfied by the **CA-IPP** tool. |  |
| 6 | FR-3 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-3)”.  If this exact string is present, FR-3 has been satisfied by the **CA-IPP** tool. |  |
| 7 | FR-4 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-4)”.  If this exact string is present, FR-4 has been satisfied by the **CA-IPP** tool. |  |

| Table  **CA-IPP Acceptance Test Plan Case 2** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-2** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the ***preprocessed\_inventory.csv*** file generated in Acceptance Test Plan Case 1 (see Table 3). Select three or more rows that have uranium inventory and copy them into a new workbook or text file. | | |
| 2 | Using Equation 1 and the tabulated constants (see Table 5) for the corresponding uranium isotopes, calculate the expected mass of uranium (summing the contribution of each isotope). | | |
| 3 | Compare the calculated mass against the mass produced by the script for the selected rows. | There should be an exact match to the 5th significant digit between the script output (the column with “U” as the header) and the calculated sums. |  |

Equation . Activity to Mass

Table . Constants for Uranium Isotopes for Conversion from Activity to Mass

|  |  |  |  |
| --- | --- | --- | --- |
| **Radionuclide** | **Atomic Mass a** | **Half Life (years) b** | **Half Life (Seconds)** |
| U-232 | 232.037 | 6.89000E+01 | 2.17432E+09 |
| U-233 | 233.040 | 1.59200E+05 | 5.02397E+12 |
| U-234 | 234.041 | 2.45500E+05 | 7.74739E+12 |
| U-235 | 235.044 | 7.04000E+08 | 2.22166E+16 |
| U-236 | 236.046 | 2.34200E+07 | 7.39079E+14 |
| U-238 | 238.051 | 4.46800E+09 | 1.40999E+17 |
| 1. First 6 significant figures for each isotope taken from physics.nist.gov on March 24th, 2020 2. Half-lives taken from EMDT-DE-0006 Rev. 1 | | | |

| Table  **CA-IPP Acceptance Test Plan Case 3** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-3** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **(NA)** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the ***preprocessed\_inventory.csv*** file in Excel and click “File” then “save as” an Excel workbook (“***preprocessed\_inventory.xlsx***”). | | |
| 2 | With the recently copied/created Excel workbook copy of the preprocessed inventory, create a new sheet. | | |
| 3 | Open and copy the “Waste\_Site” column from: "S:\PSC\!HANFORD\ICF\Prod\SWSDIST\v1.0\data\CA\_SolidWasteSite\_Centroids.csv" into the recently created sheet (refer to the previous step). | | |
| 4 | Go to “Sheet 1” of the ***preprocessed\_inventory.xlsx*** workbook. Create a copy of the waste site list that uses upper case letters for all of the sites. Consider using the “Upper()” formula from Excel. | | |
| 5 | Highlight the range of the uppercase waste site name column. Name this range “SWR\_Sites” by clicking in the “Name Box” to the left of the formula bar and typing the desired range name. | | |
| 6 | Go to the “preprocessed\_inventory” sheet of the ***preprocessed\_inventory.xlsx*** workbook. Insert a column after the “Source Type” column and name the new column “Type Verification”. | | |
| 7 | Under “Type Verification” enter the following Excel formula for all rows:  =IF(IFNA(VLOOKUP(C14,SWR\_Sites,1,FALSE),"Liq\_Site")="Liq\_Site",IF(D14="Liquid","Correct","Type\_Mismatch"),IF(D14="Solid Release Series","Correct",IF(AND(F14>0,D14="Liquid"),"Correct","Type\_Mismatch"))) | | |
| 8 | Looking at unique values present for the entire column, all records corresponding to waste site records should be listed as “Correct” under the “Type Verification” column. | All records with a recorded waste stream should have a value of “Correct” in the “Type Verification” column. |  |

# Acceptance Test Report

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

* Acceptance Test 1 is in Table A-1. It is successful and qualified to use
* Acceptance Test 2 is in Table A-2. It is successful and qualified to use
  + Excel program was used to verify the steps listed in Table A-2. The file Table\_T-2\_check located in the testing directory \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp
* Acceptance Test 3 is in Table A-3. It is Successful
  + Excel program was used to verify the steps listed in Table A-2. The file preprocessed\_inventory.xlsx located in the testing directory \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp

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

Refer to Section 4 of this software management plan for a full list of required inputs. It is recommended that a shell script be created to ease the burden of putting the command into a single command line argument. The recommended structure of this shell script is shown below:

TOOL=<path/to/ca-ipp.pl>

EHSIT=<path/to/waste/sites/file>

RADINV=<path/to/radionuclide/inventory/file>

CHEMINV=none

LIQINV=<path/to/liquid/inventory/file>

SWRDIR=<path/to/solid/waste/release/directory>

SWRIND=<path/to/solid/waste/release/index/file>

REDFIN=<path/to/file/containing/list/of/files/for/rerouting/information>

OUTPUT=preprocessed\_inventory

perl $TOOL $EHSIT $RADINV $CHEMINV $LIQINV $SWRDIR $SWRIND $REDFIN $OUTPUT

# Tool Versions

This section details changes incorporated into each version of the **CA-IPP** tool.

* 1.0 – Tool was developed.

# Appendix

**Completed Acceptance Test Cases**

**Testing Process Description**

The CA-IPP tool is checked using another script verifying only the first six functional requirements. The checking script is written in Python v3.6 and uses one library outside of the standard release called “Pandas”. The general process of the checking script is to first parse all of the input files into separate hashed dictionaries. The dictionaries are later combined into a final dictionary containing the expected result. Comparisons between the CA-IPP output (excluding the log and summary files) and the in-memory dictionary are made to verify compliance with the first 6 functional requirements. If the check passes, the script will print corresponding text to the log file for each functional requirement. In the event the checking script finds deviations from the functional requirements the output details which sites and waste streams were found to be out of compliance.

**Tool Runner Log**

INFO--03/27/2020 08:54:05 AM--Starting CA-CIE Tool Runner. Logging to "runner\_run\_AT-1\_CA-IPP\_log.txt"

INFO--03/27/2020 08:54:05 AM--Code Version: 0b4478192d987adbff818006239cda369278912a v2.2: /opt/tools/pylib/runner/runner.py<--1bcfd6779e9cbdb82673405873a8e5e81514ae27

INFO--03/27/2020 08:54:05 AM--Code Version: a27b8bc997309317b6f2f818dfd51c40276d2c6a Local repo SHA-1 has does not correspond to a remote repo release version: /home/pallena/CAVE/CA-CIE-Tools-TestRepos/ca\_ipp/CA-CIE-Tools/tools/ca-ipp/ca-ipp.pl<--1df1cd98a7c2755771e6f8427217047706466801

INFO--03/27/2020 08:54:05 AM--QA Status: QUALIFIED : /opt/tools/pylib/runner/runner.py

INFO--03/27/2020 08:54:05 AM--QA Status: TEST : /home/pallena/CAVE/CA-CIE-Tools-TestRepos/ca\_ipp/CA-CIE-Tools/tools/ca-ipp/ca-ipp.pl

INFO--03/27/2020 08:54:05 AM--Invoking Command:"perl" with Arguments:"/home/pallena/CAVE/CA-CIE-Tools-TestRepos/ca\_ipp/CA-CIE-Tools/tools/ca-ipp/ca-ipp.pl /opt/ICF/Prod/VZEHSIT/v1.1/data/Original/WasteSites(ehsit)\_Geometry\_SIMV2\_CA-dos2unix.csv /opt/ICF/Prod/VZINV/v1.0/data/F\_CP-61786\_R1\_sorted\_mar42020.csv none /opt/ICF/Prod/CLEANINV/v1.0/data/inflow-04\_inv1-edited.res /opt/ICF/Prod/RCASWR/v1.0/data/ /opt/ICF/Prod/RCASWR/v1.0/data/CASWR\_Output\_20200219\_summary\_03.09.2020.csv ./reroute-list preprocessed\_inventory"

INFO--03/27/2020 08:54:05 AM--Username:pallena Computer:olive Platform:Linux 4.4.0-38-generic #57~14.04.1-Ubuntu SMP Tue Sep 6 17:20:43 UTC 2016

| Table A-1 **CA-IPP Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-1** | | **Date: 03-27-2020** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By: Praveena Allena** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the “reroute-list” file and modify the path from “USER” to the user’s corresponding username. | | |
| 2 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./runner\_run\_AT-1\_CA-IPP.sh* | 4 files should be created:   1. ***runner\_run\_AT-1\_CA-IPP\_log.txt*** 2. ***preprocessed\_inventory.csv*** 3. ***preprocessed\_inventory.log*** 4. ***preprocessed\_inventory-summary.csv*** | Pass |
| 3 | Inside of a Linux terminal, invoke the checking script: *./ca-ipp\_check.sh* | The script should produce a file called ***ca\_ipp\_check.log***. | Pass |
| 4 | FR-1 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-1): The ca-ipp.pl output only has sites listed in the VZEHSIT.”  If this exact string is present, FR-1 has been satisfied by the **CA-IPP** tool. | Pass |
| 5 | FR-2, FR-5, FR-6 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-2, FR-5, and FR-6)”  If this exact string is present, then FR-2, FR-5, and FR-6 have been satisfied by the **CA-IPP** tool. | Pass |
| 6 | FR-3 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-3)”.  If this exact string is present, FR-3 has been satisfied by the **CA-IPP** tool. | Pass |
| 7 | FR-4 Check (from ***ca\_ipp\_check.log***) | Open ***ca\_ipp\_check.log*** in a text editor and search for the string inside double quotes: “##QA-PASS (FR-4)”.  If this exact string is present, FR-4 has been satisfied by the **CA-IPP** tool. | Pass |

**Tool Runner Log**

(NA)

| Table A-2 **CA-IPP Acceptance Test Plan Case 2** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-2** | | **Date: 03-27-2020** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By: Praveena Allena** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the ***preprocessed\_inventory.csv*** file generated in Acceptance Test Plan Case 1 (see Table 3). Select three or more rows that have uranium inventory and copy them into a new workbook or text file. | | |
| 2 | Using Equation 1 and the tabulated constants (see Table 5) for the corresponding uranium isotopes, calculate the expected mass of uranium (summing the contribution of each isotope). | | |
| 3 | Compare the calculated mass against the mass produced by the script for the selected rows. | There should be an exact match to the 5th significant digit between the script output (the column with “U” as the header) and the calculated sums. | Pass |

**Tool Runner Log**

(NA)

| Table A-3 **CA-IPP Acceptance Test Plan Case 3** | | | |
| --- | --- | --- | --- |
| **CA-IPP Acceptance Testing**  **CACIE-CA-IPP – AT-3** | | **Date: 03-27-2020** | |
| **Tool Runner Log File Location for this test:**  **(NA)** | | **Test Performed By: Praveena Allena** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Open the ***preprocessed\_inventory.csv*** file in Excel and click “File” then “save as” an Excel workbook (“***preprocessed\_inventory.xlsx***”). | | |
| 2 | With the recently copied/created Excel workbook copy of the preprocessed inventory, create a new sheet. | | |
| 3 | Open and copy the “Waste\_Site” column from: "S:\PSC\!HANFORD\ICF\Prod\SWSDIST\v1.0\data\CA\_SolidWasteSite\_Centroids.csv" into the recently created sheet (refer to the previous step). | | |
| 4 | Go to “Sheet 1” of the ***preprocessed\_inventory.xlsx*** workbook. Create a copy of the waste site list that uses upper case letters for all of the sites. Consider using the “Upper()” formula from Excel. | | |
| 5 | Highlight the range of the uppercase waste site name column. Name this range “SWR\_Sites” by clicking in the “Name Box” to the left of the formula bar and typing the desired range name. | | |
| 6 | Go to the “preprocessed\_inventory” sheet of the ***preprocessed\_inventory.xlsx*** workbook. Insert a column after the “Source Type” column and name the new column “Type Verification”. | | |
| 7 | Under “Type Verification” enter the following Excel formula for all rows:  =IF(IFNA(VLOOKUP(C14,SWR\_Sites,1,FALSE),"Liq\_Site")="Liq\_Site",IF(D14="Liquid","Correct","Type\_Mismatch"),IF(D14="Solid Release Series","Correct",IF(AND(F14>0,D14="Liquid"),"Correct","Type\_Mismatch"))) | | |
| 8 | Looking at unique values present for the entire column, all records corresponding to waste site records should be listed as “Correct” under the “Type Verification” column. | All records with a recorded waste stream should have a value of “Correct” in the “Type Verification” column. | Pass |

# Appendix

**Completed Installation Test**

| Table B-1  **CA-IPP Installation Test Plan** | | | |
| --- | --- | --- | --- |
| **CA-IPP Installation Testing**  **CACIE-CA-IPP – IT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\ca-ipp** | | | |
| **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 using *runner\_run\_IT-1.sh* as follows:  *./runner\_run\_IT-1\_CA-IPP.sh* | | |
| 2 | Verify Tool Runner is invoked and executed. | Should see exact string: “QA Status: QUALIFIED : /opt/tools/pylib/runner/runner.py” |  |
| 3 | Error messages from the tool should also be recorded in the same log file. | Should see at least one line reading “Use of uninitialized value” |  |