**CACIE Tool #NN** – **Inventory Preprocessor Tool**

**Version** **1.0**

**QA**: **QA**

# Description and Purpose

The Inventory Preprocessor tool’s purpose is to create a comprehensive data set consisting of radionuclide and/or chemical aqueous volume releases as a function of time for Central Plateau sites. Solid waste releases (chemical and/or radionuclide) may optionally be included. The input file set consists of the following types:

1. **Site-Specific Inventory**: This dataset represents any site-specific information. Any number of site-specific sources may be included.
2. **Solid Waste Release**: The data set consists of modified (i.e., reduction of number of time steps via interpolation) output from solid waste release model(s).
3. **SIMv2 Release**: The data set consists of a single CSV file containing radionuclide and liquid inventory release estimates.
4. **Chemical Inventory Release**: The data set consists of a single CSV file containing chemical and liquid release estimates.
5. **SAC Liquid Release**: The data set consists of a text file (comma-delimited) containing estimates for liquid discharges. Only water releases are included from this source file.

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 tool’s specific task is to parse out the relevant information for the waste sites found in the **VZEHSIT** file to assemble a site list containing chemical/radionuclide and water releases over time.

# Functional Requirements

The following are the functional requirements (FR) of the Inventory Preprocessor Tool:

FR-1: Accept arguments at the command line. Arguments will include:

* Required: VZEHSIT file path (including file name)
* Optional: Chemical Inventory file path (including file name)
* Optional: SAC file path (including file name)
* Optional: Solid Waste Release directory path (including the directory name)
* Optional: Solid Waste Release file path (including file name)
* Optional: Site-Specific Inventory file path (including file name)
* Optional: SIMv2 file path (including file name)
* Optional: List of analytes to treat as chemicals: array of strings
* Optional: List of encodings to try when parsing input files: array of strings
* Optional: List of analytes to include in output file: array of strings
* Optional: When set to "True", SIMv2 entrained solids will be included in the output as liquid release(s): boolean
* Optional: Name for the output file: string
* Optional: When set to "True", output will be recorded in the "Legacy" output format. If set to "False", output will be in standard format: boolean
* Optional: The name of the log file: string
* Optional: Path of directory in which to save output files (including directory name)
* Optional: The number of significant digits to preserve in the output: integer
* Optional: List of possible names for site name columns: array of strings
* Optional: Adjusts the level of detail recorded in the log file: string
* Optional: List of possible names for water columns: array of strings
* Optional: List of possible names for year columns: array of strings

FR-2: Only sites whose site name is found in the **VZEHSIT** file will be included in the output

FR-3: Sites from the **SAC** are included if no other information is had from another input file. Sites with “241-“ in its site name are excluded. The exception to the “241-“ exclusion rule are sites with “241-C” in the site name (case-insensitive), which are included in the final output. Only water releases are considered from the **SAC**.

FR-4: The user may identify a list of analytes to process from the input files (including whether to process water or not).

FR-5: The user may identify a list of analytes to be treated as chemicals. The designation of whether an analyte is a chemical will determine whether the analyte(s) are parsed from the **SIMv2** or **Chemical Inventory** files (no other input files are affected by the grouping). Designating analytes as chemicals also determines the formatting of the output headers for each analyte.

FR-6: If the “entrain solids” option is set to “true”, the tool will convert **SIMv2** records’ source type (e.g. “Solids” vs “Liquid”) to “Liquid” where the “Inventory Module” has the matching string “entrained solids” (case-insensitive).

FR-7: After parsing user-specified analytes from all input files provided (whose sites are found in the VZEHSIT list), the records are merged into a single file. The rules for merging the various input files into one output file are thus:

1. Records parsed from the **Site-Specific Inventory** file(s)
2. Records from **Solid Waste Release** file(s), excluding site(s) found in the **Site-Specific Inventory** file(s)
3. Records from **SIMv2**, excluding site(s) found in the **Site-Specific Inventory** file(s)
4. Records from **Chemical Inventory Release**, excluding site(s) found in the **Site-Specific Inventory** file(s)
5. Records from the **SAC** if site(s) has not been listed in any other source

FR-8: When writing the output file, waste release information will be grouped on a site-by-site, analyte-by-analyte, and year-by-year basis

FR-9: The Source/Inventory Module column in the output will include which file(s) contributed to each site record for every year of waste release included, separated by an underscore character (e.g. “Chemical-Inventory\_SIMv2”).

FR-10: Supports a standard and a legacy mode. The two output modes are explicitly concerned with how to format the output file. If the “legacy\_mode” option is set to “false” then the output will reflect the “standard” formatting, and vice versa if the option is set to “true”.

FR-11: The user may specify the number of significant digits reported, the default is six significant digits.

FR-12: The user may modify the default file encoding to accommodate input files with special characters.

FR-13: The user may specify string patterns for input file columns corresponding to the site name, year, and water column. Multiple patterns may be supplied, allowing flexibility in the input files to use different naming conventions for their site name, year, and water columns.

FR-14: After compiling all of the information, the user-specified number of significant figures will be preserved, rounding to the final digit. The rounding method employed will always break ties in favor of the next-greatest number. Otherwise the next closest digit is selected. A tolerance of error of “one” is reserved for any given value at the final significant digit (e.g. 3.14159 +/-0.00001).

# Software Requirements Specifications

Version 3.6 of the Python programming language was used to develop this script. The libraries implemented by this tool consist of the following:

* argparse
* copy
* logging
* os
* pandas (version 0.24.2)
* re
* math
* pathlib

All but the “pandas” library are native to the Python v3.6 release. Additional software requirements are dependencies on upstream work products that are parsed by the Inventory Preprocessor tool. The work products of interest include the following files: **Site-Specific Inventory**, **Solid Waste Release**, **SIMv2 Release**, **Chemical Inventory Release**, and **SAC Liquid Release**. The dependencies in the case of these files refers to the formatting of each file which is described in section 4 under Input files.

# Software Design Description

Arguments:

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

$ python inventory\_pp.py --VZEHSIT /path/to/VZEHSIT [other optional flags and input values]

The only required argument to run the script is the **VZEHSIT** file. However, providing this argument in isolation will not produce any output and will crash the script. The remaining arguments, though optional, may be combined to produce a single output with a selective set of inputs. Other flags will modify the output format or the parsing behavior. The optional arguments will be grouped into two categories: input files and auxiliary options.

The input file flags consist of the following (each flag is preceeded by two hyphen “-“ characters when entered at the command line:

| Table  Accepted User Arguments List | | | |
| --- | --- | --- | --- |
| **Flag** | **Group** | **Behavior** | **Usage Notes** |
| CHEMINV | Input File | **Optional: Chemical Inventory** file path (including the file name) |  |
| CLEANINV | Input File | **Optional: SAC** file path (including the file name) |  |
| RCASWR\_dir | Input File | **Optional: Solid Waste Release** directory path (including the directory name) |  |
| Site\_Specific | Input File | **Optional: Site-Specific Inventory** file path (including the name of the file) | May specify multiple files with this flag, e.g. --Site\_Specific ./file1.csv ./data/file2.csv |
| VZEHSIT | Input File | **Required: VZEHSIT** file path (including the name of the file) | **Requires** at least one of the other Optional input files to run correctly, as specified in the preceding section. |
| VZINV | Input File | **Optional: SIMv2** file path (including the name of the file) |  |
| chem\_copcs | Auxiliary | **Optional:** List of analytes to treat as chemicals | May specify multiple analytes with this flag, e.g. --chem\_copcs CN Cr U NO3  Default: CN, Cr, U, NO3  NOTE: If a radionuclide name is wholly contained by a chemical analyte name, or vice versa, it is possible for the user to inadvertently cause the tool to cross-select chemicals and analytes. An example of this would be a radionuclide called “U” and a chemical called “USER-CHEM”. Although “U” is designated a radionuclide, its name is wholly contained in “USER-CHEM” and can be identified as a chemical. For best use of this feature, ensure that analyte names are independent of one another as much as possible. |
| codec\_list | Auxiliary | **Optional:** List of encodings to try when parsing input files | May specify multiple codecs with this flag, e.g. --codec\_list utf-8 iso-8859-1  Default Value: utf-8, iso-8859-1 |
| COPC | Auxiliary | **Optional:** List of analytes to include in output file | May specify multiple analytes with this flag, e.g. --COPC water h-3 sr-90 u cn  Default Value: WATER H-3 I-129 SR-90 TC-99 U CR NO3 CN |
| entrain\_sim\_solids | Auxiliary | **Optional:** When set to "True", **SIMv2** entrained solids will be included in the output as liquid release(s). | Default: True |
| ipp\_output | Auxiliary | **Optional:** Name for the output file | Default: preprocessed\_inventory.csv |
| legacy | Auxiliary | **Optional:** When set to "True", output will be recorded in the "Legacy" output format. If set to "False", output will be in standard format | Default: True |
| logger | Auxiliary | The name of the log file | Default: inventory\_pp.log |
| output | Auxiliary | **Optional:** Path of directory in which to save output files | Default: execution directory |
| sig\_figs | Auxiliary | **Optional:** The number of significant digits to preserve in the output | Default: 6 |
| site\_keys | Auxiliary | **Optional:** List of possible names for site name columns | May specify multiple site column names, e.g. --site\_keys site\_name "cie site name" "ca site name". If column names have spaces it is necessary to use double quotes around the column header name (single quotes will not work).  Default: SITE\_NAME “CIE SITE NAME”, “CA SITE NAME” |
| verbosity | Auxiliary | **Optional:** Adjusts the level of detail recorded in the log file, defaults to [ALL] | Default: ALL |
| water\_keys | Auxiliary | **Optional:** List of possible names for water columns | May specify multiple water column names, e.g. --water\_keys water volume liquid "volume mean [m3]"  Default: WATER, VOLUME, LIQUID, VOLUME MEAN [M3], VOLUME [M3] |
| year\_keys | Auxiliary | **Optional:** List of possible names for year columns | May specify multiple year column names, e.g. --year\_keys "discharge/decay-corrected year" YEAR  Default: DISCHARGE/DECAY-CORRECTED YEAR, YEAR |

Input Files:

All lines starting with a hashtag “#” will be considered comments and will not impact the parsing methods of the Inventory Preprocessor tool. All files are expected to be comma-delimited files.

The Inventory Preprocessor tool expects that all **VZEHSIT** file is a file with a single header line (skipped). Unique values (except for Null or empty strings) are taken from the first column of the file.

The Inventory Preprocessor tool expects that all **Site-Specific Inventory** file(s) have:

* A header line containing at least 3 columns (with corresponding rows of data in subsequent lines), the columns do not have to be in order (bracketed columns are descriptive of the type of column, not to be used verbatim, e.g. [Column])::
  + [Site name column]
  + [Year column]
  + [Analyte column] (may have multiple unique columns of analytes)

The Inventory Preprocessor tool expects that all **Solid Waste Release** file(s) have:

* A file name with a site name and analyte separated by an underscore (e.g. 200-E-30\_Sr-90.csv). File endings are irrelevant, but the file naming pattern must be [site name]\_[analyte].[file ending]
* Each file is expected to have 4 header lines (comments, with or without a hashtag) preceding the data header line (found on line 5).
* A header line (line 5) with 2 columns (exactly):
  + Reduced Year
  + Reduced Activity Release Rate (Ci/year)

The Inventory Preprocessor tool expects that all **SIMv2 Release** file(s) have:

* Requires three comment lines (not distinguished with hashtag characters), these lines are skipped by the tool
* The fourth line is the header line, expecting the following columns (bracketed columns are descriptive of the type of column, not to be used verbatim, e.g. [Column]):
  + Inventory Module
  + Source Type
  + [Site name column]
  + [Year column]
  + [Analyte Column] (one or more of these, including water)

The Inventory Preprocessor tool expects that all **Chemical Inventory Release** file(s) have:

* The fourth line is the header line, expecting the following columns (bracketed columns are descriptive of the type of column, not to be used verbatim, e.g. [Column]):
  + [Site name column]
  + [Year column]
  + [Analyte Column] (one or more of these, including water)

The Inventory Preprocessor tool expects that all **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 I 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”.

Output Files:

Two files are produced by this tool:

1. Preprocessed Inventory File
   1. If the legacy output option is set to “True”, the file will be a comma-delimited file with 11 header rows preceding the 12th row which contains the column names of the file. The 13th row contains the units for each corresponding column (if applicable). Fixed column names (regardless of the analytes selected) include: Inventory Module, SIMV2 site name, CA Site Name, Source Type
   2. If standard output is selected (legacy = “False”), then no leading header rows will be printed. Instead, the first row will contain the column names.
   3. The remainder of the file (in both legacy and standard output formats) is a combination of each input file and reflects the functional requirements described/tested in this document.
2. Log File
   1. Contains meta information printed/logged by the script processes. This information is not to be QA’d but is useful information for understanding the tool’s 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}/\pylib\inventory\_pp\inventory\_pp.py “—VZEHSIT $VZEHSIT --\* $\*”

Each of these shell script variables (denoted by the “$”) will be set in the shell script with the corresponding variable input. The “--\*” and “$\*” symbols represent zero or more flags and corresponding shell script variables, depending on how the tool is executed

Code Review:

# Requirements Traceability Matrix

The requirements traceability matrix for the Inventory Preprocessor tool is presented in Table 1.

| Table  Requirements Traceability Matrix | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case** |
| QA Level | CACIE-cie-ipp.pl -IT-1 | Installation Test |
| FR-1 | All test cases | Execute the script and verify that the various command line inputs are accepted. |
| FR-2 | CACIE- Inventory Preprocessor -AT-1 | Execute the tool with various input file configurations/combinations and verify that the rules for merging the disparate input files are obeyed as specified in the FR. |
| FR-3 | CACIE- Inventory Preprocessor -AT-1 | Execute the tool with various input file configurations/combinations and verify that only the sites in the accepted site list are included in the final output. |
| FR-4 | CACIE- Inventory Preprocessor -AT-2 | Execute the tool with a specified list of analytes (including some custom analyte names) and verify that the output file contains only the analytes requested and that the output file includes the custom analytes. |
| FR-5 | CACIE- Inventory Preprocessor -AT-2 | Execute the tool with a specified list of analytes (including some custom analyte names) and verify that the output file contains only the analytes requested and that the output file includes the custom analytes. Also verify that the custom allocation of chemical analytes is honored (both in the header and the values parsed). |
| FR-6 | CACIE- Inventory Preprocessor -AT-2 | Execute the tool with the option to entrain solids set to “False”. Verify that the solids were not included. Rerun the tool with the same option set to “True” and verify that the solids were included. |
| FR-7 | CACIE- Inventory Preprocessor -AT-1 | Execute the tool with various input file configurations/combinations and verify that the rules for merging the disparate input files are obeyed as specified in the FR. |
| FR-8 | CACIE- Inventory Preprocessor -AT-1 | Execute the tool with various input file configurations/combinations and verify that the output files have grouped each waste site and their time series as described in the FR. |
| FR-9 | CACIE- Inventory Preprocessor -AT-4 | Execute the the tool and verify that the source files are properly referenced by their respective sites in the output file. |
| FR-10 | CACIE- Inventory Preprocessor -AT-5 | Execute the tool and verify that the output format when the “legacy mode” is set to “True” outputs the appropriate format with the additional lines and columns (described in Section 4). |
| FR-11 | CACIE- Inventory Preprocessor -AT-6 | Execute the tool and verify that only the specified number of significant digits were preserved. |
| FR-12 | CACIE- Inventory Preprocessor -AT-7 | Execute the tool and verify that the selected codec fails to allow the code to parse an input file. Rerun the tool with a different codec specified and verify that the output is generated correctly from the supplied input file. |
| FR-13 | CACIE- Inventory Preprocessor -AT-8 | Execute the tool with a specified set of expected column names for the year, site name, and water columns. Verify that the information was parsed correctly and saved in the output. |
| FR-14 | CACIE- Inventory Preprocessor -AT-6 | Execute the tool and verify that only the specified number of significant digits were preserved and that the rounding scheme was properly applied (refer to the FR for details). |

# Installation Test Plan and Acceptance Test Plan Cases

The installation test plan for Inventory Preprocessor is presented in Table 3 and the acceptance test plan case for Inventory Preprocessor is presented in Table 4

| Table  **Inventory Preprocessor Installation Test Plan** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Installation Testing**  **CACIE-Inventory Preprocessor – IT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp** | | | |
| **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 as follows:  *./runner\_run\_IT-1\_Inventory Preprocessor.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  **Inventory Preprocessor Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-1** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-1** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-1.log*** 2. ***step-1\_AT-1\_ipp.log*** 3. ***step-1\_AT-1\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-1.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-1.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the user arguments were accepted by comparing two files:   * ***run\_AT-1\_step-1.sh*** * ***step-1\_AT-1\_ipp.log*** | Open the files indicated in the test instruction column of this step and verify that each argument was accepted. A user verifies that each argument was accepted by ascertaining that what was specified in the shell script is printed within the first 20 lines of the log file.  Note: file paths specified in the shell script are relative paths or have the “~” shorthand. When accepted/written by the script to its log file the paths are written as absolute paths.  This partially satisfies FR-1 |  |
| 4 | Verify that each source file was merged into the output file. | Open ***step-1\_AT-1\_ipp.csv***  in a text editor (or Excel) and search for the following words/phrases:   * SAC-Water * Chemical-Inventory * AT-1\_Site-Specific-Inventory * SIMV2 * Solid-Waste-Release   If all of these bulleted words/phrases are present, this partially satisfies FR-3 and FR-7. |  |
| 5 | Verify that only sites in the accepted site list are included. | Open ***step-1\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites are in the second column:   * 241-C-105 * CHM-1 * COMMON-SITE * SAC-1 * SIM-1 * SSI-1 * SWR-1   If the listed sites are present in the output file and no other site name is found under “SITE\_NAME” in the second column, this satisfies FR-2.  This also partially satisfies FR-3, FR-7, and FR-8. |  |
| 6 | Verify that the right site was excluded while parsing from the SAC. | Open ***step-1\_AT-1\_ipp.log*** in a text editor and search for the following strings inside double quotes:   * “##Excluded sites with a substring of ‘241-‘ except for ‘241-C’ (1):” * “241-A-101”   The second string, “241-A-101”, should immediately follow the first. If both strings are present, this partially satisfies FR-3. |  |
| 7 | Verify that the data merged into the final output file is correct. | Open ***step-1\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 1 in the water, tritium, iodine, strontium, technetium, uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium * “SSI-1” has a value of 1 in the water, tritium, iodine, strontium, technetium, uranium, chromium, nitrate, and cyanide columns * “SWR-1” has a value of 2 in the iodine, strontium, and uranium columns   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 8 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-8.sh* | 2 files should be created:   1. ***step-8\_AT-1\_ipp.log*** 2. ***step-8\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 9 | Verify that the data merged into the final output file is correct. | Open ***step-8\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 3 in the water, tritium, and technetium columns. * “COMMON-SITE” has a value of 2 in the iodine, strontium, and uranium columns * “COMMON-SITE” has a value of 4 in the chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium * “SWR-1” has a value of 2 in the iodine, strontium, and uranium columns   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 10 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-10.sh* | 2 files should be created:   1. ***step-10\_AT-1\_ipp.log*** 2. ***step-10\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 11 | Verify that the data merged into the final output file is correct. | Open ***step-10\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 3 in the water, tritium, iodine, strontium and technetium columns. * “COMMON-SITE” has a value of 4 in the uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 12 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-12.sh* | 2 files should be created:   1. ***step-12\_AT-1\_ipp.log*** 2. ***step-12\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 13 | Verify that the data merged into the final output file is correct. | Open ***step-12\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 14 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-14.sh* | 2 files should be created:   1. ***step-14\_AT-1\_ipp.log*** 2. ***step-14\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 15 | Verify that the data merged into the final output file is correct. | Open ***step-14\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “COMMON-SITE” has a value of 5 in the water column * “SAC-1” has a value of 5 in only the water column   If the values specified are present as indicated per waste site, this, in conjunction with the rest of this acceptance test, satisfies FR-3, FR-7, and FR-8. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 2** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-2** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-2\runner\_AT-2.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-2** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-2\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-2.log*** 2. ***step-1\_AT-2\_ipp.log*** 3. ***step-1\_AT-2\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-2.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-2.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the user-selected analytes and list of chemical analytes were accepted by the script.  NOTE: If a radionuclide name is wholly contained by a chemical analyte name, or vice versa, it is possible for the user to inadvertently cause the tool to cross-select chemicals and analytes. An example of this would be a radionuclide called “U” and a chemical called “USER-CHEM”. Although “U” is designated a radionuclide, its name is wholly contained in “USER-CHEM” and can be identified as a chemical. For best use of this feature, ensure that analyte names are independent of one another as much as possible. | If the shell script was left unmodified by the tester (tester should feel free to modify the shell script for their own testing, preferably maintining a copy), the following strings should be found in ***step-1\_AT-2\_ipp.log***:   * chem\_copcs : ['U', 'CR', 'NO3', 'CN', 'USER-CHEM'] * copcs : ['WATER', 'H-3', 'U', 'USER-RAD', 'USER-CHEM']   This partially satisfies FR-4 and FR-5.  This also partially satisfies FR-1. |  |
| 4 | Verify that the data merged into the final output file is correct. | Open ***step-1\_AT-2\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 3 in the water and tritium columns * A value of 4 in the uranium column * A value of 9 in the “USER-RAD” column * A value of 8 in the “USER-CHEM” column   Finally, verify that uranium and “USER-CHEM” have units of “(kg/year)” in the column header line and that tritium and “USER-RAD” have units of “(Ci/year)”.  If all of these values are present as specified, this, in conjunction with the rest of this acceptance test, satisfies FR-4 and FR-5. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 3** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-3** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-3\runner\_AT-3.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-3** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-3\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-3.log*** 2. ***step-1\_AT-3\_ipp.log*** 3. ***step-1\_AT-3\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-3.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-3.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The shell script executed in step 1 did not include entrained solids. Verify that the script was set to not include entrained solids. | Open ***step-1\_AT-3\_ipp.log*** and verify that the following text within double quotes is present:  “entrain\_sim\_solids : False”  If the string inside double quotes is present in the file, this partially satisfies FR-1 and FR-6. |  |
| 4 | Verify that the correct data was pulled from the SIMv2 source file into the final output. | Open ***step-1\_AT-3\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 4 in the water, tritium, iodine, strontium, and technetium columns   If the values are present as specified, this partially satisfies FR-6. |  |
| 5 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-3\_step-5.sh* | 2 files should be created:   1. ***step-5\_AT-3\_ipp.log*** 2. ***step-5\_AT-3\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-3.log*** |  |
| 6 | The shell script executed in step 5 included entrained solids. Verify that the script was set to include entrained solids. | Open ***step-5\_AT-3\_ipp.log*** and verify that the following text within double quotes is present:  “entrain\_sim\_solids : True”  If the string inside double quotes is present in the file, this partially satisfies FR-1 and FR-6. |  |
| 7 | Verify that the correct data was pulled from the SIMv2 source file into the final output. | Open ***step-5\_AT-3\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 4 in the water column * A value of 7 in the tritium, iodine, strontium, and technetium columns   If the values are present as specified, this, in conjunction with the rest of this acceptance test, satisfies FR-6. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 4** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-4** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-4\runner\_AT-4.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-4** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-4\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-4.log*** 2. ***step-1\_AT-4\_ipp.log*** 3. ***step-1\_AT-4\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-4.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-4.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that each site has the appropriate “Source” value, which refers to the corresponding input file the site information came from. Where multiple sites contributed to the site information, the source files are concatenated with an underscore character “\_”. | Open ***step-1\_AT-3\_ipp.csv*** and verify that the following sources are attributed to the corresponding site:   * CHM-1 should have “Chemical-Inventory” for its source * COMMON-SITE should have “Chemical-Inventory\_SIMV2\_Solid-Waste-Release” for its source * SAC-1 should have “SAC-Water” for its source * SIM-1 should have “SIMV2” for its source * SSI-1 should have “AT-4\_Site-Specific-Inventory” for its source   If the source(s) for each site is present as specified, this satisfies FR-9. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 5** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-5** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-5\runner\_AT-5.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-5** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-5\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-5.log*** 2. ***step-1\_AT-5\_ipp.log*** 3. ***step-1\_AT-5\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-5.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-5.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The shell script executed in step 1 will format the output in the “Legacy” format. Verify that formatting is correct. | Open ***step-1\_AT-3\_ipp.csv*** in Excel or a text editor and verify that the following strings are present:   * There should be 11 lines at the top of the file where each line says “# Header String” * The 12th line of the file should have the following string (in double quotes): “Inventory Module,SIMV2 site name,CA Site Name,Source Type,Discharge/decay-corrected year,Volume [m3],H-3,I-129,Sr-90,Tc-99,U,Cr,No3,Cn” * The 13th line should read as follows (in double quotes): “,,,,year,m^3,Ci,Ci,Ci,Ci,kg,kg,kg,kg” * The 14th line should read as follows (in double quotes): “Chemical-Inventory,,CHM-1,Liquid,1961.0,4.0,,,,,4.0,4.0,4.0,4.0” * The 15th line should read as follows (in double quotes): “Chemical-Inventory\_SIMV2\_Solid-Waste-Release,,COMMON-SITE,Liquid,1961.0,3.0,3.0,2.0,2.0,3.0,2.0,4.0,4.0,4.0” * The 16th line should read as follows (in double quotes): “SAC-Water,,SAC-1,Liquid,1961.0,5.0,,,,,,,,” * The 17th line should read as follows (in double quotes): “SIMV2,,SIM-1,Liquid,1961.0,3.0,3.0,3.0,3.0,3.0,,,,” * The 18th line should read as follows (in double quotes): “AT-5\_Site-Specific-Inventory,,SSI-1,Liquid,1961.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0”   If the text is present as specified, this satisfies FR10.  This also partially satisfies FR-1. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 6** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-6** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-6\runner\_AT-6.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-6** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-6\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-6.log*** 2. ***step-1\_AT-6\_ipp.log*** 3. ***step-1\_AT-6\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-6.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-6.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the precision flag was used and given a value of 5 | Open ***step-1\_AT-6\_ipp.log*** and verify that the line “sig\_figs : 5” is in the file.  If the line is in the file as stated, the partially satisfies FR-1 and FR-11. |  |
| 4 | The tool default precision is 6 significant digits. For this acceptance test the output precision was set to 5 significant digits.  The rounding used in this tool will break ties by rounding up in all cases. This differs from many standard practices which attempt to reduce bias in the rounding by rounding to the nearest even or odd number when breaking ties. | Open ***step-1\_AT-6\_ipp.csv*** in a text editor (may use Excel, but take care to display only the reported digits) and verify the following (in order from left to right):   * Water has a value of “1.0” * Tritium has a value of “100000.0” * Iodine has a value of “100010.0” * Strontium has a value of “100010.0” * Technetium has a value of “1234600.0” * Uranium has a value of “100010.0” * Chromium has a value of “100020.0” * Nitrate has a value of “100020.0” * Cyanide has a value of “1.0”   If the values are recorded as specified, this satisfies both FR-11 and FR-14. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 7** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-7** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-7\runner\_AT-7.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-7** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-7\_step-1.sh* | 2 files should be created:   1. ***runner\_run\_AT-7.log*** 2. ***step-1\_AT-7\_ipp.log*** |  |
| 2 | Open the ***runner\_run\_AT-6.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-6.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The codec supplied in the shell script was not correctly matched. Verify that the specified codec was accepted by the script and that it attempted to use it to parse the file (and failed). | Open ***step-1\_AT-6\_ipp.log*** and verify that the line “Unsuccessful attempt to parse /home/[USER]/CAVE/CA-CIE-Tools-TestEnv/inventory\_pp/data/AT-7/F\_CP-61786\_R1\_sorted\_mar42020.csv using codec: utf-8” is in the file. Replace “[USER]” with the username corresponding with the account used to test the tool.  If the line is in the file as stated, the partially satisfies FR-1 and FR-12. |  |
| 4 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-7\_step-4.sh* | 2 files should be created:   1. ***step-4\_AT-7\_ipp.csv*** 2. ***step-4\_AT-7\_ipp.log***   1 file should have been modified:   1. ***runner\_AT-7.log*** |  |
| 5 | Verify that the new input for a different codec was accepted by the tool | Open ***step-4\_AT-7\_ipp.log*** in a text editor and verify that the following line is present in the file: “codec\_list : ['iso-8859-1']”  If the specified text is present in the file, this partially satisfies FR-1 and FR-12. |  |
| 6 | Verify that the file was parsed and output was generated. | Open ***step-4\_AT-7\_ipp.csv*** in a text editor (or Excel) and verify the following:   * The output should have the following header line: “Source,SITE\_NAME,YEAR,WATER(m^3/year),H-3(Ci/year),I-129(Ci/year),SR-90(Ci/year),TC-99(Ci/year)” * There should be only one unique value under the “Source” column: “SIMV2” (for lines 2 through 55) * There should be only one unique value under the “SITE\_NAME” column: “200-E-100” (for lines 2 through 55).   If the bulleted items are true then this satisfies FR-12. |  |

| Table  **Inventory Preprocessor Acceptance Test Plan Case 8** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-8** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-8\runner\_AT-8.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-8** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-8\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-8.log*** 2. ***step-1\_AT-8\_ipp.log*** 3. ***step-1\_AT-8\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-8.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-8.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the tool was passed the right custom keys to use and that the tool accepted them. | Open ***step-1\_AT-8\_ipp.log*** and verify that the following lines are present in the file (within the double quotes):   * “site\_keys : ['CUSTOM-SITECOL', 'SITE\_NAME']” * “water\_keys : ['CUSTOM-WATERCOL', 'WATER']” * “year\_keys : ['CUSTOM-YEARCOL', 'YEAR']”   If the lines are in the file as stated, the partially satisfies FR-1 and FR-13. |  |
| 4 | With the custom columns specified, this will allow the tool to parse the 2 source files for their respective sites: SSI-1 and SSI-2 | Open ***step-1\_AT-6\_ipp.csv*** in a text editor (may use Excel) and verify the following:   * Line 1 should have the following string (within double quotes): “Source,SITE\_NAME,YEAR,WATER(m^3/year)” * Line 2 should have the following string (within double quotes): “AT-8\_Site-Specific-Inventory\_1,SSI-1,1961.0,1.0” * Line 3 should have the following string (within double quotes): “AT-8\_Site-Specific-Inventory\_2,SSI-2,1961.0,2.0”   If the strings are recorded as specified, this, in conjunction with all previous acceptance tests satisfies both FR-1 and FR-13. |  |

# Acceptance Test Report

To complete the Acceptance Testing use Appendix A. This constitutes a single acceptance test that addresses all of the functional requirements listed in the traceability matrix.

Details of this test, when conducted, by whom, and if Passed or Failed are in 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 (this does not contain an exhaustive listing of possible combinations of options and is only meant to illustrate how to execute the tool using a shell script):

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

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

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

CHEMINV=<path/to/chemical/inventory/file>

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

REDFIN=<path/to/file/containing/list/of/files/for/site-specific/information>

OUTPUT=preprocessed\_inventory

python $TOOL $EHSIT $RADINV $CHEMINV $LIQINV $REDFIN $OUTPUT

# Tool Versions

This section details changes incorporated into each version of the **Inventory Preprocessor** tool.

* 1.0 – Tool was developed.

# Appendix

**Completed Acceptance Test Cases**

**Tool Runner Log**

| Table AT-1  **Inventory Preprocessor Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-1** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-1** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-1.log*** 2. ***step-1\_AT-1\_ipp.log*** 3. ***step-1\_AT-1\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-1.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-1.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the user arguments were accepted by comparing two files:   * ***run\_AT-1\_step-1.sh*** * ***step-1\_AT-1\_ipp.log*** | Open the files indicated in the test instruction column of this step and verify that each argument was accepted. A user verifies that each argument was accepted by ascertaining that what was specified in the shell script is printed within the first 20 lines of the log file.  Note: file paths specified in the shell script are relative paths or have the “~” shorthand. When accepted/written by the script to its log file the paths are written as absolute paths.  This partially satisfies FR-1 |  |
| 4 | Verify that each source file was merged into the output file. | Open ***step-1\_AT-1\_ipp.csv***  in a text editor (or Excel) and search for the following words/phrases:   * SAC-Water * Chemical-Inventory * AT-1\_Site-Specific-Inventory * SIMV2 * Solid-Waste-Release   If all of these bulleted words/phrases are present, this partially satisfies FR-3 and FR-7. |  |
| 5 | Verify that only sites in the accepted site list are included. | Open ***step-1\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites are in the second column:   * 241-C-105 * CHM-1 * COMMON-SITE * SAC-1 * SIM-1 * SSI-1 * SWR-1   If the listed sites are present in the output file and no other site name is found under “SITE\_NAME” in the second column, this satisfies FR-2.  This also partially satisfies FR-3, FR-7, and FR-8. |  |
| 6 | Verify that the right site was excluded while parsing from the SAC. | Open ***step-1\_AT-1\_ipp.log*** in a text editor and search for the following strings inside double quotes:   * “##Excluded sites with a substring of ‘241-‘ except for ‘241-C’ (1):” * “241-A-101”   The second string, “241-A-101”, should immediately follow the first. If both strings are present, this partially satisfies FR-3. |  |
| 7 | Verify that the data merged into the final output file is correct. | Open ***step-1\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 1 in the water, tritium, iodine, strontium, technetium, uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium * “SSI-1” has a value of 1 in the water, tritium, iodine, strontium, technetium, uranium, chromium, nitrate, and cyanide columns * “SWR-1” has a value of 2 in the iodine, strontium, and uranium columns   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 8 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-8.sh* | 2 files should be created:   1. ***step-8\_AT-1\_ipp.log*** 2. ***step-8\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 9 | Verify that the data merged into the final output file is correct. | Open ***step-8\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 3 in the water, tritium, and technetium columns. * “COMMON-SITE” has a value of 2 in the iodine, strontium, and uranium columns * “COMMON-SITE” has a value of 4 in the chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium * “SWR-1” has a value of 2 in the iodine, strontium, and uranium columns   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 10 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-10.sh* | 2 files should be created:   1. ***step-10\_AT-1\_ipp.log*** 2. ***step-10\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 11 | Verify that the data merged into the final output file is correct. | Open ***step-10\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 3 in the water, tritium, iodine, strontium and technetium columns. * “COMMON-SITE” has a value of 4 in the uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column * “SIM-1” has a value of 3 in the water, tritium, iodine, strontium, and technetium   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 12 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-12.sh* | 2 files should be created:   1. ***step-12\_AT-1\_ipp.log*** 2. ***step-12\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 13 | Verify that the data merged into the final output file is correct. | Open ***step-12\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “CHM-1” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “COMMON-SITE” has a value of 4 in the water, uranium, chromium, nitrate, and cyanide columns * “SAC-1” has a value of 5 in only the water column   If the values specified are present as indicated per waste site, this partially satisfies FR-3, FR-7, and FR-8. |  |
| 14 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-1\_step-14.sh* | 2 files should be created:   1. ***step-14\_AT-1\_ipp.log*** 2. ***step-14\_AT-1\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-1.log*** |  |
| 15 | Verify that the data merged into the final output file is correct. | Open ***step-14\_AT-1\_ipp.csv*** in a text editor (or Excel) and verify that the following sites have their corresponding values for the columns specified from 1961 through 1970:   * “241-C-105” has a value of 6 only in the water column * “COMMON-SITE” has a value of 5 in the water column * “SAC-1” has a value of 5 in only the water column   If the values specified are present as indicated per waste site, this, in conjunction with the rest of this acceptance test, satisfies FR-3, FR-7, and FR-8. |  |

**Tool Runner Log**

| Table AT-2  **Inventory Preprocessor Acceptance Test Plan Case 2** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-2** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-2\runner\_AT-2.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-2** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-2\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-2.log*** 2. ***step-1\_AT-2\_ipp.log*** 3. ***step-1\_AT-2\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-2.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-2.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the user-selected analytes and list of chemical analytes were accepted by the script.  NOTE: If a radionuclide name is wholly contained by a chemical analyte name, or vice versa, it is possible for the user to inadvertently cause the tool to cross-select chemicals and analytes. An example of this would be a radionuclide called “U” and a chemical called “USER-CHEM”. Although “U” is designated a radionuclide, its name is wholly contained in “USER-CHEM” and can be identified as a chemical. For best use of this feature, ensure that analyte names are independent of one another as much as possible. | If the shell script was left unmodified by the tester (tester should feel free to modify the shell script for their own testing, preferably maintining a copy), the following strings should be found in ***step-1\_AT-2\_ipp.log***:   * chem\_copcs : ['U', 'CR', 'NO3', 'CN', 'USER-CHEM'] * copcs : ['WATER', 'H-3', 'U', 'USER-RAD', 'USER-CHEM']   This partially satisfies FR-4 and FR-5.  This also partially satisfies FR-1. |  |
| 4 | Verify that the data merged into the final output file is correct. | Open ***step-1\_AT-2\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 3 in the water and tritium columns * A value of 4 in the uranium column * A value of 9 in the “USER-RAD” column * A value of 8 in the “USER-CHEM” column   Finally, verify that uranium and “USER-CHEM” have units of “(kg/year)” in the column header line and that tritium and “USER-RAD” have units of “(Ci/year)”.  If all of these values are present as specified, this, in conjunction with the rest of this acceptance test, satisfies FR-4 and FR-5. |  |

**Tool Runner Log**

| Table AT-3  **Inventory Preprocessor Acceptance Test Plan Case 3** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-3** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-3\runner\_AT-3.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-3** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-3\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-3.log*** 2. ***step-1\_AT-3\_ipp.log*** 3. ***step-1\_AT-3\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-3.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-3.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The shell script executed in step 1 did not include entrained solids. Verify that the script was set to not include entrained solids. | Open ***step-1\_AT-3\_ipp.log*** and verify that the following text within double quotes is present:  “entrain\_sim\_solids : False”  If the string inside double quotes is present in the file, this partially satisfies FR-1 and FR-6. |  |
| 4 | Verify that the correct data was pulled from the SIMv2 source file into the final output. | Open ***step-1\_AT-3\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 4 in the water, tritium, iodine, strontium, and technetium columns   If the values are present as specified, this partially satisfies FR-6. |  |
| 5 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-3\_step-5.sh* | 2 files should be created:   1. ***step-5\_AT-3\_ipp.log*** 2. ***step-5\_AT-3\_ipp.csv***   1 file should have been modified:   1. ***runner\_run\_AT-3.log*** |  |
| 6 | The shell script executed in step 5 included entrained solids. Verify that the script was set to include entrained solids. | Open ***step-5\_AT-3\_ipp.log*** and verify that the following text within double quotes is present:  “entrain\_sim\_solids : True”  If the string inside double quotes is present in the file, this partially satisfies FR-1 and FR-6. |  |
| 7 | Verify that the correct data was pulled from the SIMv2 source file into the final output. | Open ***step-5\_AT-3\_ipp.csv***  in a text editor (or Excel) and verify that for each year of the solitary site, “COMMON-SITE”, the following values are recorded:   * A value of 4 in the water column * A value of 7 in the tritium, iodine, strontium, and technetium columns   If the values are present as specified, this, in conjunction with the rest of this acceptance test, satisfies FR-6. |  |

**Tool Runner Log**

| Table AT-4  **Inventory Preprocessor Acceptance Test Plan Case 4** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-4** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-4\runner\_AT-4.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-4** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-4\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-4.log*** 2. ***step-1\_AT-4\_ipp.log*** 3. ***step-1\_AT-4\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-4.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-4.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that each site has the appropriate “Source” value, which refers to the corresponding input file the site information came from. Where multiple sites contributed to the site information, the source files are concatenated with an underscore character “\_”. | Open ***step-1\_AT-3\_ipp.csv*** and verify that the following sources are attributed to the corresponding site:   * CHM-1 should have “Chemical-Inventory” for its source * COMMON-SITE should have “Chemical-Inventory\_SIMV2\_Solid-Waste-Release” for its source * SAC-1 should have “SAC-Water” for its source * SIM-1 should have “SIMV2” for its source * SSI-1 should have “AT-4\_Site-Specific-Inventory” for its source   If the source(s) for each site is present as specified, this satisfies FR-9. |  |

**Tool Runner Log**

| Table AT-5  **Inventory Preprocessor Acceptance Test Plan Case 5** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-5** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-5\runner\_AT-5.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-5** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-5\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-5.log*** 2. ***step-1\_AT-5\_ipp.log*** 3. ***step-1\_AT-5\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-5.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-5.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The shell script executed in step 1 will format the output in the “Legacy” format. Verify that formatting is correct. | Open ***step-1\_AT-3\_ipp.csv*** in Excel or a text editor and verify that the following strings are present:   * There should be 11 lines at the top of the file where each line says “# Header String” * The 12th line of the file should have the following string (in double quotes): “Inventory Module,SIMV2 site name,CA Site Name,Source Type,Discharge/decay-corrected year,Volume [m3],H-3,I-129,Sr-90,Tc-99,U,Cr,No3,Cn” * The 13th line should read as follows (in double quotes): “,,,,year,m^3,Ci,Ci,Ci,Ci,kg,kg,kg,kg” * The 14th line should read as follows (in double quotes): “Chemical-Inventory,,CHM-1,Liquid,1961.0,4.0,,,,,4.0,4.0,4.0,4.0” * The 15th line should read as follows (in double quotes): “Chemical-Inventory\_SIMV2\_Solid-Waste-Release,,COMMON-SITE,Liquid,1961.0,3.0,3.0,2.0,2.0,3.0,2.0,4.0,4.0,4.0” * The 16th line should read as follows (in double quotes): “SAC-Water,,SAC-1,Liquid,1961.0,5.0,,,,,,,,” * The 17th line should read as follows (in double quotes): “SIMV2,,SIM-1,Liquid,1961.0,3.0,3.0,3.0,3.0,3.0,,,,” * The 18th line should read as follows (in double quotes): “AT-5\_Site-Specific-Inventory,,SSI-1,Liquid,1961.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0”   If the text is present as specified, this satisfies FR10.  This also partially satisfies FR-1. |  |

**Tool Runner Log**

| Table AT-6  **Inventory Preprocessor Acceptance Test Plan Case 6** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-6** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-6\runner\_AT-6.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-6** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-6\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-6.log*** 2. ***step-1\_AT-6\_ipp.log*** 3. ***step-1\_AT-6\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-6.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-6.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the precision flag was used and given a value of 5 | Open ***step-1\_AT-6\_ipp.log*** and verify that the line “sig\_figs : 5” is in the file.  If the line is in the file as stated, the partially satisfies FR-1 and FR-11. |  |
| 4 | The tool default precision is 6 significant digits. For this acceptance test the output precision was set to 5 significant digits.  The rounding used in this tool will break ties by rounding up in all cases. This differs from many standard practices which attempt to reduce bias in the rounding by rounding to the nearest even or odd number when breaking ties. | Open ***step-1\_AT-6\_ipp.csv*** in a text editor (may use Excel, but take care to display only the reported digits) and verify the following (in order from left to right):   * Water has a value of “1.0” * Tritium has a value of “100000.0” * Iodine has a value of “100010.0” * Strontium has a value of “100010.0” * Technetium has a value of “1234600.0” * Uranium has a value of “100010.0” * Chromium has a value of “100020.0” * Nitrate has a value of “100020.0” * Cyanide has a value of “1.0”   If the values are recorded as specified, this satisfies both FR-11 and FR-14. |  |

**Tool Runner Log**

| Table AT-7  **Inventory Preprocessor Acceptance Test Plan Case 7** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-7** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-7\runner\_AT-7.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-7** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-7\_step-1.sh* | 2 files should be created:   1. ***runner\_run\_AT-7.log*** 2. ***step-1\_AT-7\_ipp.log*** |  |
| 2 | Open the ***runner\_run\_AT-6.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-6.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | The codec supplied in the shell script was not correctly matched. Verify that the specified codec was accepted by the script and that it attempted to use it to parse the file (and failed). | Open ***step-1\_AT-6\_ipp.log*** and verify that the line “Unsuccessful attempt to parse /home/[USER]/CAVE/CA-CIE-Tools-TestEnv/inventory\_pp/data/AT-7/F\_CP-61786\_R1\_sorted\_mar42020.csv using codec: utf-8” is in the file. Replace “[USER]” with the username corresponding with the account used to test the tool.  If the line is in the file as stated, the partially satisfies FR-1 and FR-12. |  |
| 4 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-7\_step-4.sh* | 2 files should be created:   1. ***step-4\_AT-7\_ipp.csv*** 2. ***step-4\_AT-7\_ipp.log***   1 file should have been modified:   1. ***runner\_AT-7.log*** |  |
| 5 | Verify that the new input for a different codec was accepted by the tool | Open ***step-4\_AT-7\_ipp.log*** in a text editor and verify that the following line is present in the file: “codec\_list : ['iso-8859-1']”  If the specified text is present in the file, this partially satisfies FR-1 and FR-12. |  |
| 6 | Verify that the file was parsed and output was generated. | Open ***step-4\_AT-7\_ipp.csv*** in a text editor (or Excel) and verify the following:   * The output should have the following header line: “Source,SITE\_NAME,YEAR,WATER(m^3/year),H-3(Ci/year),I-129(Ci/year),SR-90(Ci/year),TC-99(Ci/year)” * There should be only one unique value under the “Source” column: “SIMV2” (for lines 2 through 55) * There should be only one unique value under the “SITE\_NAME” column: “200-E-100” (for lines 2 through 55).   If the bulleted items are true then this satisfies FR-12. |  |

**Tool Runner Log**

| Table AT-8  **Inventory Preprocessor Acceptance Test Plan Case 8** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Acceptance Testing**  **CACIE-Inventory Preprocessor – AT-8** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-8\runner\_AT-8.log** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp\tests\AT-8** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Navigate to the Testing Directory | | | |
| 1 | Inside of a Linux terminal, invoke the Tool Runner with the test input files as follows: *./run\_AT-8\_step-1.sh* | 3 files should be created:   1. ***runner\_run\_AT-8.log*** 2. ***step-1\_AT-8\_ipp.log*** 3. ***step-1\_AT-8\_ipp.csv*** |  |
| 2 | Open the ***runner\_run\_AT-8.log*** in a text editor. Look to verify that the runner tool is qualified and that the inventory preprocessor is “TEST” | The ***runner\_run\_AT-8.log*** should have 2 lines:   * QUALIFIED : /opt/tools/pylib/runner/runner.py * TEST: /home/[USER]/…/inventory\_pp.py   [USER] will be replaced by the user’s username and the ellipsis will be replaced by the full path to the file.  The key words are “QUALIFIED” and “TEST” as described. If present, this verifies that the tool and testing environment are valid. |  |
| 3 | Verify that the tool was passed the right custom keys to use and that the tool accepted them. | Open ***step-1\_AT-8\_ipp.log*** and verify that the following lines are present in the file (within the double quotes):   * “site\_keys : ['CUSTOM-SITECOL', 'SITE\_NAME']” * “water\_keys : ['CUSTOM-WATERCOL', 'WATER']” * “year\_keys : ['CUSTOM-YEARCOL', 'YEAR']”   If the lines are in the file as stated, the partially satisfies FR-1 and FR-13. |  |
| 4 | With the custom columns specified, this will allow the tool to parse the 2 source files for their respective sites: SSI-1 and SSI-2 | Open ***step-1\_AT-6\_ipp.csv*** in a text editor (may use Excel) and verify the following:   * Line 1 should have the following string (within double quotes): “Source,SITE\_NAME,YEAR,WATER(m^3/year)” * Line 2 should have the following string (within double quotes): “AT-8\_Site-Specific-Inventory\_1,SSI-1,1961.0,1.0” * Line 3 should have the following string (within double quotes): “AT-8\_Site-Specific-Inventory\_2,SSI-2,1961.0,2.0”   If the strings are recorded as specified, this, in conjunction with all previous acceptance tests satisfies both FR-1 and FR-13. |  |

# Appendix

**Completed Installation Test**

| Table B-1  **Inventory Preprocessor Installation Test Plan** | | | |
| --- | --- | --- | --- |
| **Inventory Preprocessor Installation Testing**  **CACIE-Inventory Preprocessor – IT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **\\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp** | | **Test Performed By:** | |
| **Testing Directory: \\olive\backups\CAVE\CA-CIE-Tools-TestEnv\inventory\_pp** | | | |
| **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 as follows:  *./runner\_run\_IT-1\_Inventory Preprocessor.sh* | | |
| 2 | Verify Tool Runner is invoked and executed. | Should see exact string: “QA Status: QUALIFIED : /opt/tools/pylib/runner/runner.py” |  |
| 3 |  |  |  |