**CACIE Tool #01** – ***CAST (???)***

**Version** **1.0**

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

1. **Description and Purpose**

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

The CAST tool builds basic site-specific STOMP input <???files???> using a template xml file. There is a template xml file for models located in the 200E Area and a template xml file for models located in the 200W Area. The tool acts as an interface to update the template to a model-specific xml file that documents the following information:

* the modeler generating the input files,
* the site-specific grid model Surfaces, and
* the LEAP frog-mapped site-specific materials for the model.

The CAST tool generates the following site-specific STOMP input files as well as a log file (input.buildLog):

* input
* input.zone
* input.sij
* input.nij
* input.top
* input.bot
* input.north
* input.south
* input.west
* input.east

1. **Functional Requirements**

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

The functional requirements for the CAST tool are as follows:

FR-1: Read in template xml file (200E-mdef.xml or 200W-mdef.xml)

FR-2: Update template xml file with modeler name and model-specific name (now model-specific xml file)

FR-3: Update model-specific xml file with site-specific grid model Surfaces

FR-4: Update model-specific xml file with site-specific materials.

FR-5: Document updates made to the model-specific xml file.

…

FR-X:

1. **Software Requirements Specifications**

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

JAVA

1. **Software Design Description**

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

Template xml files:

* 200E-mdef.xml for the 200-East
* 200W-mdef.xml for 200-West

Output files:

* ModelName.csv

.bat file: runCAST.bat

* java -Dsun.java2d.dpiaware=false -jar -Djava.library.path=..\..\tools\CAST\lib -Xmx7000m -Xms7000m ..\..\tools\CAST\ModelSetupFY18.jar

1. **Requirements Traceability Matrix**

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

Table 1 presents the requirements traceability matrix for the CAST tool.

| **Table 1. CAST Tool Requirements Traceability Matrix** | | |
| --- | --- | --- |
| **Functional Requirement** | **Acceptance Test** | **Test Result (Pass/Fail)** |
| FR-X |  |  |
| FR-X |  |  |
| FR-X |  |  |
| FR-X |  |  |

1. **Test Plan and Cases**

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

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

The test plan for the CAST tool is as follows.

| **Table 2. CAST Tool Test Plan** | | |
| --- | --- | --- |
| **TEST ID** | **Test Case** | **Test Result (Pass/Fail)** |
| IT-1 | Installation Test |  |
| ATC-X |  |  |
| ATC-X |  |  |

See attachments for the acceptance test case test logs.

1. **Acceptance Test Report**

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

1. **User Guide**

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