Attachment D

IQST Consolidated Tool Package Template

This page intentionally left blank.

**IQST Consolidated Tool Package Template**

***The CHD Package Generation* Tool**

***chdgenerate.py***

**Version 1.0**

**Validated (DATE TOOL IS APPROVED): MM/DD/YYYY**

1. **Description and Purpose**

The Time-Variant Specified Head Package Generation Tool provides a method to create a MODFLOW compliant input file that, when used as part of a MODFLOW simulation, approximates the interaction the suprabasalt aquifer at the Hanford Site with groundwater just outside the model domain. This code reads input files that describe the hydraulic head data as measured at wells near model boundaries along with model grid cell information at designated model boundaries.  Hydraulic head data are assigned to cell locations based on user input in order to create a MODFLOW input file for the Time-Variant Specified Head Package of MODFLOW.

1. **Functional Requirements**

The following are the functional requirements (FR) of the CHD Package Generation Tool:

| Table 1. Functional Requirements for The CHD Package Generation Tool. | | |
| --- | --- | --- |
| **Unique Number** | **Requirement** | **Source** |
| FR-1 | Read the Grid Location file for information on the cells in the model. | Identified Modeling Need |
| FR-2 | Read the Hydraulic Head Elevation files for information on well elevation. | Identified Modeling Need |
| FR-3 | Read the Model Elevation Arrays to get the information on the elevation for each cell in each layer. | Identified Modeling Need |
| FR-4 | Read the Model IBOUND arrays (one for each layer) that contain the information as to which cells are active. | Identified Modeling Need |
| FR-5 | For each of the cells in the Grid Location file, and for each layer of that cell, compare its elevation to the elevation of its corresponding well. If the cell’s elevation is lower, and that cell is active, it is added to the CHD package with this information: layer, row, column, well’s elevation, well’s elevation. | Identified Modeling Need |
| FR-6 | Run FR-5 for each of the remaining stress periods in the Hydraulic Head Elevation Files. | Identified Modeling Need |
| FR-7 | Output the CHD package to a file in 10-space delimited format. | Identified Modeling Need |

1. **Software Requirements Specifications**

The Software requirements are what is required on the computer to run the CHD Generation tool. They are listed in the table below.

| Table 2. Software Requirements for The CHD Package Generation Tool. | | |
| --- | --- | --- |
| **Unique Number** | **Requirement** | **Objective Verification Method** |
| SR-1 | Windows or Linux | Confirm chdgenerate.py runs on both Windows and Linux without administration rights. |
| SR-2 | Python 3.0 | Confirm that chdgenerate.py runs using Python 3. |
| SR-3 | Python Library: Pandas version 1.5.2 | Confirm that chdgenerate.py works with Pandas 1.5.2 |

1. **Software Design Description**

Flow:

Breakdown of the source code flow.

1. Parse command line arguments.
2. Open param.json and load in parameters.
3. Reads in the model elevation array files into memory.
4. Reads in the model IBOUND array files into memory.
5. Read in hydraulic head elevation files into memory.
6. Read in grid location file (from command line argument) into pandas dataframe.
7. Find which well corresponds with each cell.
8. Starting from the first cell, calculate where the position of the cell would be in the model elevation arrays and IBOUND arrays using the equation: columns\*(row-1) + column -1 where columns is the total number of columns in the grid(specified by the param.json file.)
9. Go through the IBOUND arrays to find all the active layers in that cell’s position.
   1. For each active layer for that cell:
      1. Find the elevation of that cell using the elevation arrays. Using the current layer, go to the corresponding elevation array and use the position of the cell to find the elevation.
      2. If the elevation of that cell’s well is higher than the elevation of the cell
         1. Add that cell to the temporary CHD package. The cell data in the CHD package should contain the layer, row, column, well’s elevation, and well’s elevation.
10. Add the total number of entries for that stress period to the final CHD package.
11. Add all the data from the temporary CHD package (CHD cells from this stress period) to the final CHD package.
12. Repeat steps 6-9 but with all the remaining stress periods.
13. Write final CHD package to a file by the name specified in the command line arguments.

Arguments:

The arguments for the program are passed in command line using flags:

--c Grid Location File

--o Output file

The program is called as so:

python chdgenerate.py –c <Grid Location file> --o <Output file>

Input Files:

* + params.json

params.json is a file that is automatically read in by the program. It contains parameters and inputs that are not specified by the command line inputs. It contains these variables:

* numRows – number of rows in the model
* numCol – number of columns in the model
* rowColumn – which column is the row specified in the Grid Location file
* columnColumn – which column is the column specified in the Grid Location file
* wellColumn – which column is the well specified in the Grid Location file
* timeColumn – which column is the time value in the Hydraulic Head Elevation file
* elevationColumn – which column is the flow value in the Hydraulic Head Elevation file
* maxChdCells – how much memory allocated should be printed on top of the CHD package
* cbbUnitNumber – how much max memory per entry should be printed on top of the CHD package
* bot\_ref\_files – list of the model elevation array files (one for each layer)
* ibnd\_inf\_files – list of the model IBOUND array files (one for each layer)
* numLayers – number of layers in the model
* numWells – number of wells in the model
* numStressPeriods – number of stress periods in the well files
* well\_files – list of hydraulic head elevation csv files. All should have identical stress periods.
* wellID – a list of the identifications used for the well in the cell information file. Each well IDs’ position on the list is respective to the well in the same position on the well\_files’ list.

The param.json file is in standard json format.

Example:

{

"numRows": 7,

"numColumns": 7,

………….

}

* + Grid Location File

The Grid Location File is a csv file containing the information about the cells. After the header, each row in this file is data about a cell. The information the program uses from this file are the row, column, and which well is corresponds to. Which columns each of these datums are in is specified in params.json.

* + Output file
    - The output file is where you want the generated CHD package to be. The program will create a file by that name and will write a 10-space delimited file without headers but with the columns: layer, row, column, well’s elevation, and well’s elevation.

Output Files:

The output file is where you want the generated CHD package to be. The program will create a file by that name and will write a csv file without headers but with the columns: layer, row, column, stage, well’s elevation, well’s elevation. If file by that name already exists, it will be overwritten.

Execution:

The program is a python program. It is a command line program that can be run in both Windows and Linux. To run, navigate to the directory containing the program, and type:

python chdgenerate.py --c <Grid Location file> --o <Output file>

The output file will appear in the same directory by the name specified.

Code Review:

A code review was performed by [REVIEWER NAME] on [DATE CODE REVIEW PERFORMED]. No impacts to other repository tools or library dependencies were identified for the [TOOL NAME] tool.

1. **Requirements Traceability Matrix**

A requirements traceability matrix for the tool will be documented in the table below. At a minimum, the matrix will include the following (as applicable):

* **Unique Number**: A unique identification number containing the general category of the requirement assigned in ascending order.
* **Requirement**: The requirement statement.
* **Source of Requirement**: The requirement source (Identified Modeling Need, Conference; Configuration Control Board; Task Assignment, etc.)
* **Software Requirements Specification (SRS)**: The section in this document that references the requirement.
* **Design Specification**: Design document where the requirement is defined. Not applicable to acquired software, per CPCC-PRO-IRM-309, *Controlled Software Management*, Section 3.1.7 Software Requirements Identification and Management, Number 5.
* **Program Module**: The software module satisfying the requirement.
* **Test Case Number**: The test case number referencing where the requirement is tested.
* **Successful Test Verification**: Indicate Pass or Fail to indicate successful verification testing of satisfying the requirement.
* **Modification of Requirement**: If requirement was changed, eliminated, or replaced, indicate disposition and authority for modification.
* **Remarks**: Provide any pertinent remarks or notes.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3. Requirements Traceability Matrix for The CHD Package Generation Tool. | | | | | | | | | | |
| **Unique  Number** | **Requirement** | **Source of Requirement** | **Software Requirements Specification** | **Design Specification** | **Program Module** | **Test Case** | **Successful Test Verification Pass / Fail** | **Modification of Requirement** | **Remarks** |
| **Objective 1:Read in the Grid Location file, the Hydraulic Head Elevation files, the Model Elevation array files and the Model IBOUND array files.** | | | | | | | | | | |
| **RTM-1.1** | Read the Grid Location File for information on the cells in the model. | Identified Modeling Need | Section 2  FR-1 | Section 4  Step 6 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **RTM-1.2** | Read the Hydraulic Head Elevation files to get the elevation of each hydraulic head at each specific stress period | Identified Modeling Need | Section 2  FR-2 | Section 4  Step 5 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **RTM-1.3** | Read the Model Elevation Array files to get the information on the elevation for each cell in each layer. | Identified Modeling Need | Section 2  FR-3 | Section 4  Step 3 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **RTM-1.4** | Read the Model IBOUND array files (one for each layer) that contain the information as to which cells are active. | Identified Modeling Need | Section 2  FR-4 | Section 4  Step 4 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **Objective 2: Calculate the CHD cells and output in the correct format to the output file.** | | | | | | | | | | |
| **RTM-2.1** | For each of the cells in the Grid Location file, compare its elevation to the elevation of its corresponding well. If the cell’s elevation is higher, that cell is added to the CHD package with this information: layer, row, column, well’s elevation, well’s elevation. | Identified Modeling Need | Section 2  FR-5 | Section 4  Step 7-11 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **RTM-2.2** | Run FR-5 for each of the remaining stress periods in Well Elevation Files. | Identified Modeling Need | Section 2  FR-6 | Section 4  Step 12 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |
| **RTM-2.3** | Output the CHD package to a file in 10-space delimited format. | Identified Modeling Need | Section 2  FR-7 | Section 4  Step 13 | chdgenerate.py | IQST-CHD Package Generation Tool-ATC-1 |  | No | None |

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 (See Section 8.1.4 of the IQST Utility Code ISMP for TEST ID naming conventions) will be referenced in the RTM and ATR. An installation test 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. (See Section 8.1.4 of the IQST Utility Code ISMP for identification conventions for the test cases).

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

| Table 4. [TOOL ACRONYM] Installation Test Plan | | | | |
| --- | --- | --- | --- | --- |
| **IQST-The CHD Package Generation Tool** **Installation Testing Case 1**  **Description: Tests the correct installation of the CHD Package Generation Tool.** | | **Installation Test Case #:**  **IQST-The CHD Package Generation Tool-ITC-1** | **Date:** | |
| **System Attributes available in the Tool Runner File. Location for this test:** | | **Test Performed By:** | | |
| **Testing Directory:** | | | | |
| **Test Step** | **Test Instruction** | **Test Step** | | **Test Instruction** |
| Navigate to the testing directory | | | | |
| 1 | Make sure python is installed on the computer. | | | |
| 2 | Make sure that params.json is in the same folder as the program. | | | |
| 3 | All the parameters including the Model Elevation Arrays, Model IBOUND arrays are specified in the params.json. The complete list of parameters is specified earlier in the document. | | | |
| 4 | All the input files, including the ones specified in params.json, exist. | | | |

| For all the testing, some test input files are included. It contains a 7 row by 5 column cell model. Throughout testing, this can be used to modify input cells and test outputs.  Table 5. [TOOL ACRONYM] Acceptance Test Plan Case 1 | | | | |
| --- | --- | --- | --- | --- |
| **IQST-The CHD Package Generation Tool Acceptance Test Case 1**  **Description: Tests all the requirements in the requirements traceability matrix.** | | **Test Case #:**  **IQST-The CHD Package Generation Tool-ATC-1** | **Date:** | |
| **System Attributes available in the Tool Runner File. Location for this test:** | | **Test Performed By:** | | |
| **Testing Directory:** | | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | | **Test Result  (Pass/Fail)** |
| 1 | Ensure that all the needed files are there. (Grid Location File, all the Hydraulic Head elevation files, all the Model Elevation Files, and all the Model IBOUND Array files. For the last 2, which are specified in params.json, there is one for each layer. | All the files exist and are in proper format. | |  |
| 2 | Execute the program using the format specified earlier in this document. | Program executes without error. | |  |
| 3 | 1. Modify the input files and the model elevation arrays so that all the cells in the grid location file would be included in the CHD package. This can be done by putting 1 in all the cells for the IBOUND file for layer 1, 0 in all the cells for the other IBOUND layer files, and 0 in all the cells for the first elevation array file. 2. Make sure all the well elevation files only have one stress period and the hydraulic head of the wells is above 1. 3. Update info in params.json. 4. Execute the program. 5. Confirm that the number of cells in the generated CHD package is the same number as in the grid location file. | Number of cells in the CHD package should be the same as the number of cells in the grid location file. | |  |
| 4 | 1. Modify the hydraulic head elevation files and the params.json file to have a certain number of stress periods. 2. Execute the program. 3. Confirm that the CHD package has the information for all the stress periods. | The CHD package has the groups of cells for all the stress periods in the well elevation files. | |  |
| 5 | 1. Execute the program with some input files. 2. Open the CHD package and find a specific cell and look at the layer. (Preferably a cell in the first few rows to make it more convenient.) 3. Go to the corresponding IBOUND file and change that cell to 0, and make sure the next layer is 1 for that cell. 4. Execute the program again. 5. Confirm that cell’s layer has changed to the next layer. 6. Note: If a cell disappears after the IBOUND is changed, that means that the next layers elevation is now under that cell’s well. 7. Note 2: The program automatically checks that it reads in “numLayers” IBOUND files where numLayers is specified in params.json. | Confirming that changing the IBOUND files changes the results confirms that the program is reading in the IBOUND files. | |  |
| 6 | 1. Execute the program with a set of inputs. 2. Open the CHD package and choose a cell in there. 3. Looking at the well elevation of that cell, change that cell’s elevation so that it is lower than the well elevation. This can be done by looking at the layer, row and column, finding the relevant position among the cell elevation files, and changing the value. 4. Run the program again and confirm that the cell is not in the CHD package anymore for that stress period. | The cell is not in the CHD package anymore if their elevation is above the relevant well elevation. | |  |
| 7 | 1. Execute the program with a set of inputs. 2. Confirm that the output first has the max CHD cells number specified in params.json., then the cbb Unit Number. 3. Confirm that the next line has the number of cells for that stress period. 4. Confirm that all the cells afterwards are in this format: layer, row, column, well’s elevation, well’s elevation. 5. Confirm that everything is in 10-space delimited format. | The output CHD package is in the correct format. | |  |

**Acceptance Test Summary**

Control Procedures details in Section 9.1.3 of 66771, Rev. 1, *Software Management Plan for IQST-UTILS* was followed during the testing of this IQST-UTILS tool. All testing was performed without administrator level privileges. During testing the tester observed no indication of degradation of the overall system nor any unintended functions of the software and handling of abnormal conditions/events.

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.

Make sure params.json is set and the Model Elevation Arrays and the Model IBOUND arrays exist(as described earlier in this document)

Open a command line terminal and run:

python chdgenerate.py –c <Grid Location file> --o <Output file>

The output file will appear in the same directory by the name specified.

1. **Tool Versions**

This section details changes incorporated into each version of the [Keywords] tool.

* 1.0 – Tool was developed.

1. **Attachments**

* Attachment A – IQST-UTILS Code Custodian Qualification Template and Completed Forms
* Attachment B – Code Walkthrough Template and Completed Forms
* Attachment C – Unit Testing Template and Completed Form