# NFCDD Node Point Table python script

This tool was coded to automate creation of the NFCDD Node Point Table. When uploading flood model results for the EA National Flood Coastal Defense Database one of the deliverables is the Modelled Node Point Measurement MS Access database table. This table is simply a list of the model nodes with the peak levels and peak flow results from the model 1D outputs tabulated. There are several filler columns including the model node point name, a code derived from the node ID and information supplied by the EA for the project.

This document is a user guide for a python script that will create this table for you.

## To use this script you will need the following:

Python (Anaconda 2.7) installed

Access to the python script; S:\Sheffield\_Library\Knowledge and Tools\Innovations\Python\_Scripts\NFCDD\_scripts\NODEPOINTMEASUREMENT\_002.py

ISIS 1D results files exported to the correct format

An ISIS LNG file, or any text file with a list of node names (one per line in the file) which are identical in format to those in the 1D results file.

A modelled Flood Group code set. Different EA regions have different naming systems but you should be provided with a Modelled Flood Group code, Modelled Flood Group name and system for naming individual nodes.

## Naming conventions

Computers are not intelligent, they are simply elaborate calculators that take data in process it and spit a result out based on instructions. I have written this program to take specific formats of data in. If these exact naming conventions are not used the script is not clever enough to work.

An ISIS node should be labelled RIVER\_0000b, where RIVER is the river name, and 0000 is the chainage, and b indicates a structure or some such edit. The key element is the underscore. Some of the processes within the script look for the underscore, if this is missing the script might not work or produce a strange result.

The ISIS 1D results files (exported using ISIS or Flood Modeller from the ZZN file) must be labelled;

RIVERNAME\_0000\_99\_F or RIVERNAME\_0000\_99\_H

River name is the river name, this **must not** contain an underscore!

0000 is the chainage, this must be all digits and have an underscore on either side.

99 is the model iteration number. The script doesn’t actually look at this but it does count the underscore after it.

H or F, meaning Head or Flow. The script identifies whether a file is a head(level) or flow results file by looking at the last letter. If the letter is H it adds the file to the head results list, if F to the flow list.

The script will identify the return period of the results file from the results file name. It does this by counting across the underscores from the end of the file path and taking the 3rd item and converting it into an integer. Consequently if there are fewer or more underscores or if there is an alphabetical character with the return period number the script will not work. Remember climate change scenarios should be labeled 101. 101=100+CC.

## Problem solving

If for some reason the script will not work with your results contact [simon.desmet@capita.co.uk](mailto:simon.desmet@capita.co.uk). The script can be fairly easily edited to resolve any changes in naming convention or data format. If you wish to attempt editing of the code yourself (not as hard as it sounds) use notepad++. I have added code line numbers in italics at various points so you can easily ID the bit of code that needs changing for a given issue.

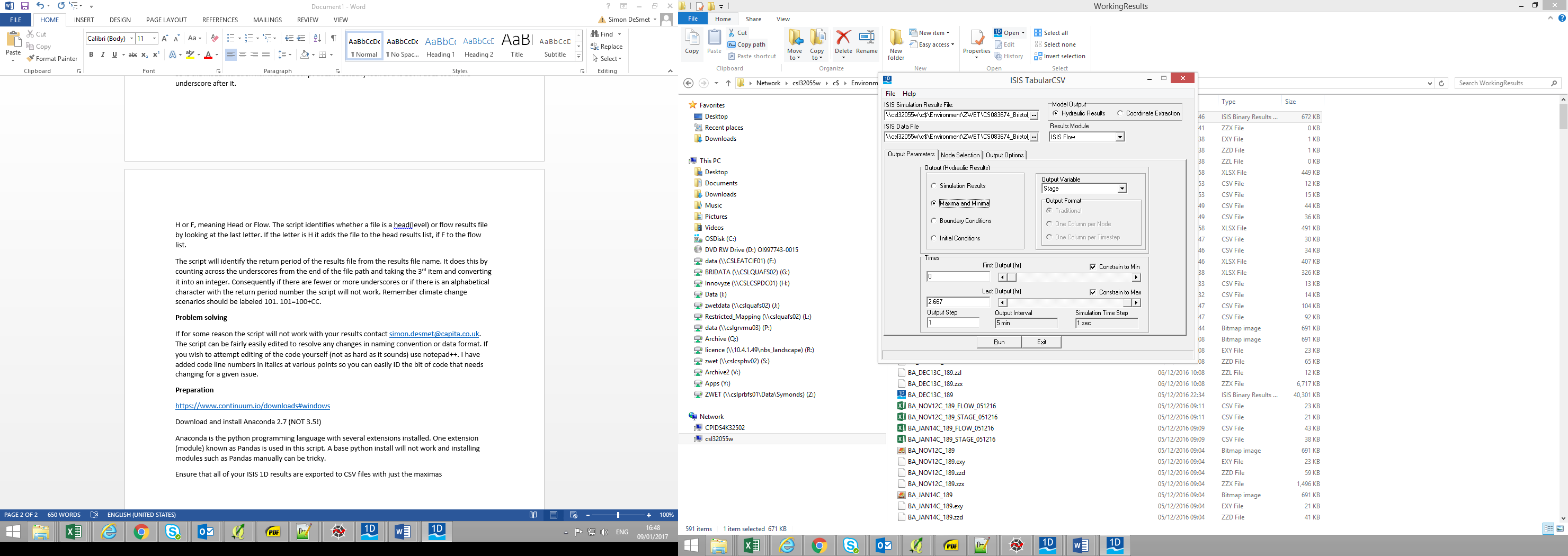
## Preparation

<https://www.continuum.io/downloads#windows>

Download and install Anaconda 2.7 (NOT 3.5!)

Anaconda is the python programming language with several extensions installed. One extension (module) known as Pandas is used in this script. A base python install will not work and installing modules such as Pandas manually can be tricky.

Ensure that all of your ISIS 1D results are exported to CSV files with just the maximas and minimas;



Ensure all of these results are copied into a folder, and that this folder contains NO other CSV files (the script will try and open all csv files in the folder.

Ensure the results files are named correctly.

Ensure you have an LNG file. And LNG file is an ISIS file created by using the node selection tab in the ISIS tabularCSV window. Simply select the desired nodes and save the list. Alternatively create a text document with a list of the nodes you want in it. If you open the LNG file is should look like this;

HINB\_03567i

HINB\_03533i

HINB\_03502

HINB\_03456i

HINB\_03410

HINB\_03342

HINB\_03251

HINB\_03251i

HINB\_03150

HINB\_03150i

HINB\_03044

HINB\_02953

You list of nodes should include all river sections and potentially all interpolates. Structures such as weir and bridge units and junctions should all be removed. In some cases some interpolates will not be used, if in doubt use all interpolates as the data can be deleted later.

## Running the Script

Double click on the following file:

*S:\Sheffield\_Library\Knowledge and Tools\Innovations\Python\_Scripts\NFCDD\_scripts\NODEPOINTMEASUREMENT\_002.py*

A command line window should open

If .py files are not associated with python you will need to right click on the .py file and select properties. Then choose the ‘change’ button next to ‘opens with’. Scroll down the list until you can select ‘look for another app on this pc’, and then navigate to your python.exe e.g. *C:\Anaconda2\python.exe*

In the command line the script will ask the following questions;

*"file path for LNG file please:"*

Enter the complete file path to your LNG file (or text file) e.g.

*S:\Sheffield Based Jobs\CS078708\_NW Package 3\Hydraulics\NFCDD\_PROCESSING\WRAY\FILES\WRAY.lng*

*"EA Region-Area-Catchment code please, e.g. ea01207"*

*"River Name, e.g. Wray"*

*"Modelled Flood Group Name, e.g. SFRM\_2015"*

These three questions all pertain to the Node Point Name and Modelled Flood Group Code. Enter text as indicated in the examples. If you project has a different EA naming scheme this section of code will need to be rewritten, see the final section of the document for more details.

*“FOLDER path for ISIS Flow and Head results please:”*

Enter the path to the folder containing your 1D ISIS result CSV files. E.g.

*S:\Sheffield Based Jobs\CS078708\_NW Package 3\Hydraulics\NFCDD\_PROCESSING\WRAY\1DRESULTS*

The model will then create a new folder called NPM in the results folder you have supplied the path too and write the NodePointTable into this folder. Usually the process takes a matter of seconds, a big model may take longer.

## Editing the Script

In this section I will give a brief description of the code works and what you might need to edit.

Step 1 the code creates lists for the different fields in the Node Point Table. (lines 29-40). Each of these lists is essentially a column in the final table.

Step 2, the code asks the user the questions noted above and gets its input LNG file and ISIS results, lines 48-77. The Head and Flow results are identified by files ending in H.csv and F.csv respectively and attached to separate lists (lines 70 and 74). If you have labelled your files differently as long as the level flow or head level appears at the end of the file you can edit these lines to identify your files.

Step 3, the LNG list is edited to remove \n from each line. This is a quirk of python and how it handles new lines in a text file. As each node label is on its own line all the line returns have to be taken out (lines 70-74)

Step 4, associate NFCDD IDs with each node label. This might be unnecessary as the EA should do this on results upload. In this case you can just delete this column from the final table. The code assigns a unique number (starting with 1) to each ISIS node. It labels them in the order they are found in the LNG file (lines 91-96).

Step 5, find and write the Head (level) results. This is the most complex section of code (lines 101-120). The code opens the list of head results files and opens the first file. It then splits up the file name by underscore (\_) to find the return period and then loads in the file line by line (lines 105/106). For each line it splits out the first item (the node label) and compares it to the LNG list, if there is a match it copies the node label and head result into the appropriate lists (columns in the eventual table). The level number is rounded to 2dp (lines 110/111). The code also fills out the NodePointName (line 112), ModelledFloodGroupCode (113) using the node label and the user inputs. Finally the filler columns for supporting, downloaded, For\_upload etc are filled with False or True (lines 115-119). The NFCDD\_ID is also filled in (line 120). The script loops this procedure over every results file in the list, comparing each node in the results file to the LNG file.

Step 6, find and write flow results. This is a simplification of the above, but this time with the flow results. Several steps are left out as the majority of the table columns have already been filled. In the unlikely event that the nodes appear in a different order the flow results from the level results a checkcode column is written out with the node for each flow result being copied in.

Step 7, the columns are assembled into a table using the Pandas module. The table is sorted b the NodePointName column and the columns in the table are ordered correctly. This step and step 8 are the only steps that use Pandas, everything else is core python. If for any reason you cannot get Anaconda or Pandas to work and have only regular python you could delete these lines (lines 140-158) and instead write print ’column name’ for each column. The code would then write each column to the command line window. You could then copy the data out of the window and into a spreadsheet.

Step 8, the code writes the resulting table to a csv using the supplied 1D results path but adding a new folder called NPM the model group code and .csv (line 162). If you wish to specify a different folder here this is the line to edit.

Name: Simon DeSmet  
Job: Graduate Modeller

Team: ZTWE Flood Risk and Water Management

Tel: 01228673774  
Email: simon.desmet@capita.co.uk