# FAST – Flood Assessment Structure Tool

FAST is a python-based structure level assessment tool for floods. It has a pre-processing tool that helps the user prep the data for the analysis tool. FAST was built starting with the ArcGIS Python script developed by DOGAMI: https://www.oregongeology.org/pubs/ofr/O-18-04/O-18-04_user_guide.pdf. The tool was developed in an open source framework with an addition of a pre- processing tool to help users assign Depth Damage Functions (DDFs) for structure, content and inventory depth damage functions (DDFs) based on Occupancy Type, Foundation Type and number of stories and flood type if the user does not already have the DDF IDs identified. Analysis is rapid, approximately 10K records per second based on simple input and output .csv files.

**Prerequisites to setup the tool:**

**Python Packages:** GDAL version 2.4.0, numpy version 1.16.2, tkinter version 8.6, utm version 0.4.2

**Folders:** ../Rasters, ../UDF, ../Log

**Rasters**
The program will look for rasters in the .tif, .tiff format in the 'rasters' in folder in the main directory. The user may add or remove depth grids to this hazard data library folder.

The default depth grid rasters included with this demo are>
Rasters:

- Honolulu_GAT.tif: Depth grid developed for Honolulu City/County based on potential inundation from the Great Aleutian tsunami scenario.
- NYC_rpd100.tif: Depth grid created for the 5 Boroughs of New York City based on the FEMA 100-year mapping.
- ND_Minot_PNNL100.tif, ND_Minot_PNNL500.tif, ND_Minot_rpd100.tif, ND_Minot_rpd500.tif: Representing multiple flood hazard sources including 100- and 500-year depth grids from the Pacific Northwest National Labs (PNNL) and 100 and 500 year depth grids developed by Hazus using 100 and 500 year discharges for the Souris River.

**UDF:**
This folder contains some sample UDF input files for Honolulu County, New Yor City and Minot, ND:

Each row must have columns corresponding to the fields below:

| **Input** | **Required?** |
|---|---|
| A UDF in a .csv file-format | Yes |
| User Defined Flty Id: | Yes |
| Occupancy Class: | Yes |

| | |
|---|---|
| One of 33 Hazus-defined types, e.g., {RES1, RES2, COM3, IND4, AGR1, GOV2, REL1}. Script will skip row if not specified, or if an unrecognized value is provided. | |
| Cost:<br>Replacement Cost of Structure, in U.S. dollars. Records with '0' cost: the script will accept a zero value, but any estimated dollar damage to the structure will be 0. Consider correcting the UDF record or deleting it. | Yes |
| Number of Stories:<br>Number of stories of building. Must be an integer. | Yes |
| Foundation Type:<br>Foundation Type of the building. Text type, per Hazus-MH Flood Model convention. Must be an integer from 1 to 7, inclusively. | Yes |
| First Floor Height:<br>Must be a float greater than 0. | Yes |
| Area:<br>Total Area for the structure, in square feet. Used for Inventory Loss calculation when Inventory Cost is not supplied. Used for debris estimates. Must be greater than 0. | Yes |
| Coastal Flooding attribute (flC):<br>Used by pre-processing script to assign default DDFs. Riverine is used by default, but user should change to Coastal A or V if coastal scenario. If user has already assigned DDFs this setting will not impact results. However, losses will increase based on Coastal DDFs and should be used for coastal scenarios. Since most buildings do not have unique Coastal A zone DDFs, Coastal V will be used by default. As a result, there are minimal differences between Coastal A and V losses. | Yes |
| Content Cost:<br>If attribute is supplied, script will use the attribute value; otherwise, the script will assume Content Cost is 50% or 100% or 150% of Building Cost, depending on Occupancy Class. Must be greater than or equal to 0. | No |
| Inventory Cost:<br>Hazus estimates are provided based on Occupancy Class and Area unless provided by the user. Must be greater than or equal to 0. | No |
| Building DDF:<br>If not provided by the user, defaults will be assigned based on Hazus methodology by computing Specific Occupancy ID based on Occupancy Type, Foundation Type, num stories and flood type. If populated by user, the script will check to ensure that only valid DDFs are used. | No |
| Content DDF:<br>If not provided by the user, defaults will be assigned based on Hazus methodology by computing Specific Occupancy ID based on Occupancy Type, Foundation Type, num stories and flood type. If populated by user, the script will check to ensure that only valid DDFs are used. | No |
| Inventory DDF: | No |

| | |
|---|---|
| If not provided by the user, defaults will be assigned based on Hazus methodology by computing Specific Occupancy ID based on Occupancy Type, Foundation Type, num stories and flood type. If populated by user, the script will check to ensure that only valid DDFs are used. | |

HI_Honolulu_UDF.csv: A sample csv file containing (>80K) buildings with required attributes for Honolulu County for use with the raster depth grid for Honolulu.
NY_NYC_UDF.csv: A sample csv file containing (>800K) buildings with required attributes for use with the NYC depth grid.
ND_Minot_UDF.csv: A sample csv file containing (>12K) building with required attributes for use with the Minot depth grids.

## Lookup Tables

The program will look for the lookup tables in the .csv format in the 'lookuptables' folder in the main directory.

## 1. How to start:

Double click on OpenHazus_POC.bat in the main directory.

A windowed GUI should launch with field inputs.
A console log should also launch; check here for errors.

## 2. The GUI:

The GUI of the program allows for custom field mapping and checking. If valid input UDF (in a .csv format) is not selected, the fields will be color coded as RED.

If an input UDF is selected the program will search through the input UDF's field names and cross-check them against what is currently in the corresponding text entry box. It also checks against the default name of the field, according to its field name on the left of the entry box.
If the field is colored YELLOW, that field is has not been successfully mapped, but is NOT critical as the program will develop and assign defaults. *NOTE: We encourage the user to customize field-names to be recognized by default by changing the code in gui_program.py.*

If the field is colored RED and a UDF is already selected, the field has not been usefully mapped and IS critical. If the attribute exists in the users data, the field name should be manually typed in box.

If the field is colored GREEN, the field has been mapped successfully.

## 3. Input:

CSV file with fields corresponding to program requirements.

You will be asked to browse for this file. The included UDF folder has some ready-to-go samples with pre-mapped fields. You must be sure to select a depth grid that spans the coordinates in the UDF, otherwise depth calculations will not results in any loss calculations. *NOTE: multiple grids can be selected if you want to run them sequentially; they may be selected by shift or control clicking multiple selections.*

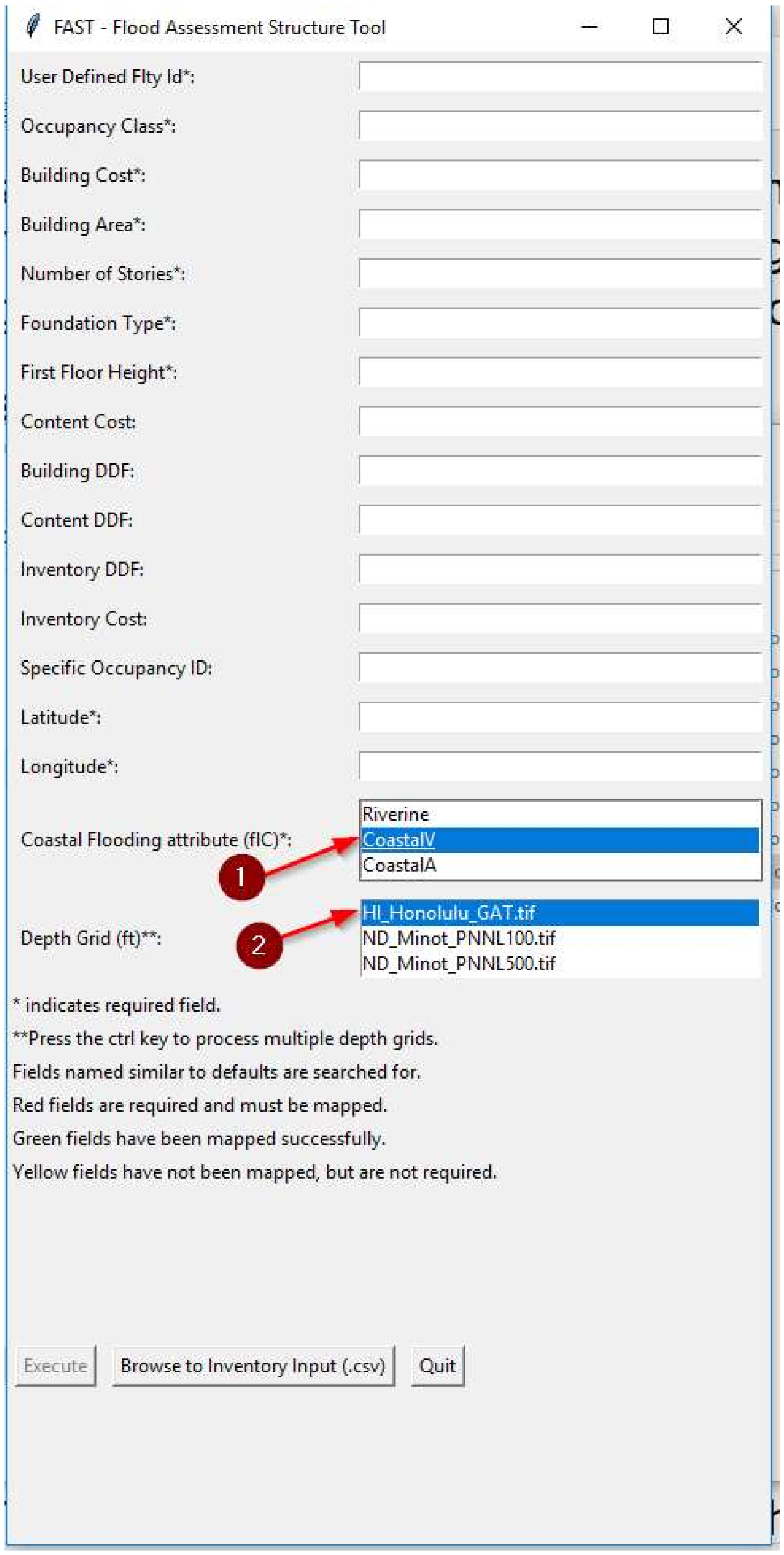
4. **Output:**

If the program runs successfully, you will find the final product in .csv file-format in the same location of the original input .csv file. The name should be the original name of the .csv file with an added _RASTERNAME to the end.

**DEMO INSTRUCTIONS:**

Open the program using FAST.bat,

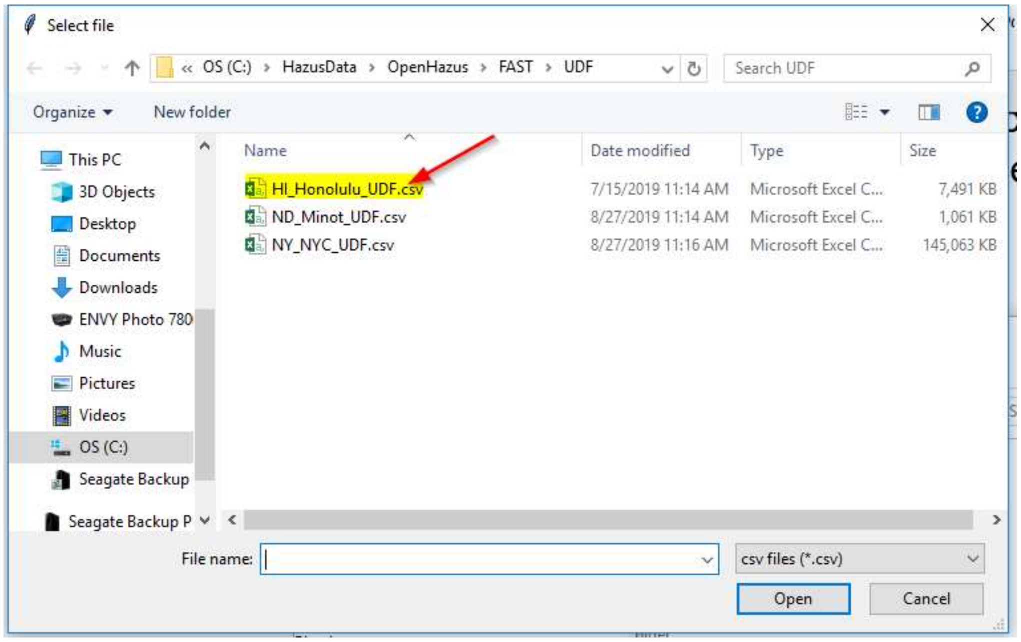
1. Leave all the entry text fields blank select the Honolulu_GAT.tif raster and since this is a coastal scenario, select Coastal V. The window should look like this:


FAST - Flood Assessment Structure Tool

| | |
|---|---|
| User Defined Flty Id*: | |
| Occupancy Class*: | |
| Building Cost*: | |
| Building Area*: | |
| Number of Stories*: | |
| Foundation Type*: | |
| First Floor Height*: | |
| Content Cost: | |
| Building DDF: | |
| Content DDF: | |
| Inventory DDF: | |
| Inventory Cost: | |
| Specific Occupancy ID: | |
| Latitude*: | |
| Longitude*: | |
| Coastal Flooding attribute (flC)*: | Riverine<br>CoastalV<br>CoastalA |
| Depth Grid (ft)**: | HI_Honolulu_GAT.tif<br>ND_Minot_PNNL100.tif<br>ND_Minot_PNNL500.tif |

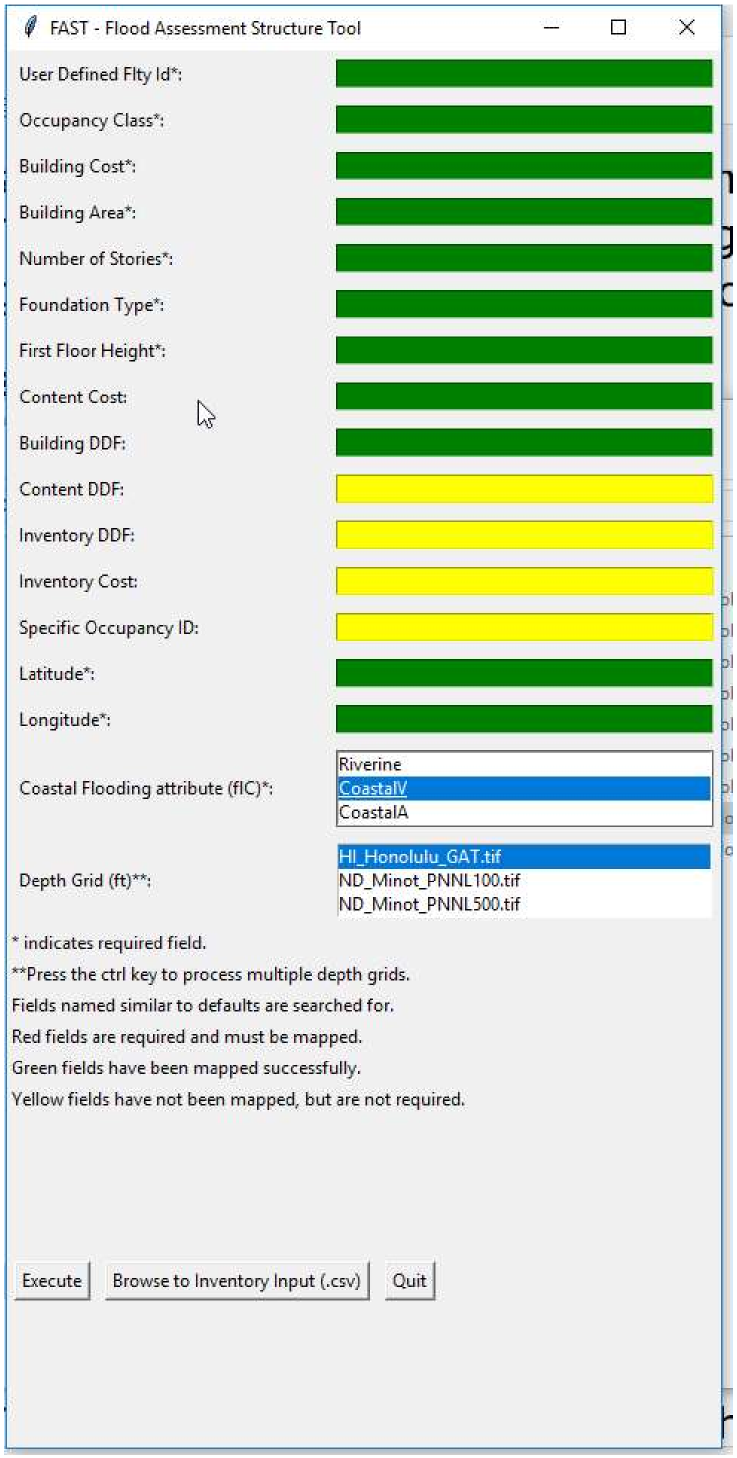
* indicates required field.
**Press the ctrl key to process multiple depth grids.
Fields named similar to defaults are searched for.
Red fields are required and must be mapped.
Green fields have been mapped successfully.
Yellow fields have not been mapped, but are not required.

2. Press the "Browse to Inventory Input (.csv)" button and you will be brought to the UDF folder in the main directory; pick the HI_Honolulu_UDF.csv. It should look something like this:



| Name | Date modified | Type | Size |
|---|---|---|---|
| HI_Honolulu_UDF.csv | 7/15/2019 11:14 AM | Microsoft Excel C... | 7,491 KB |
| ND_Minot_UDF.csv | 8/27/2019 11:14 AM | Microsoft Excel C... | 1,061 KB |
| NY_NYC_UDF.csv | 8/27/2019 11:16 AM | Microsoft Excel C... | 145,063 KB |

3. ***Open*** it, the window should then look like this:

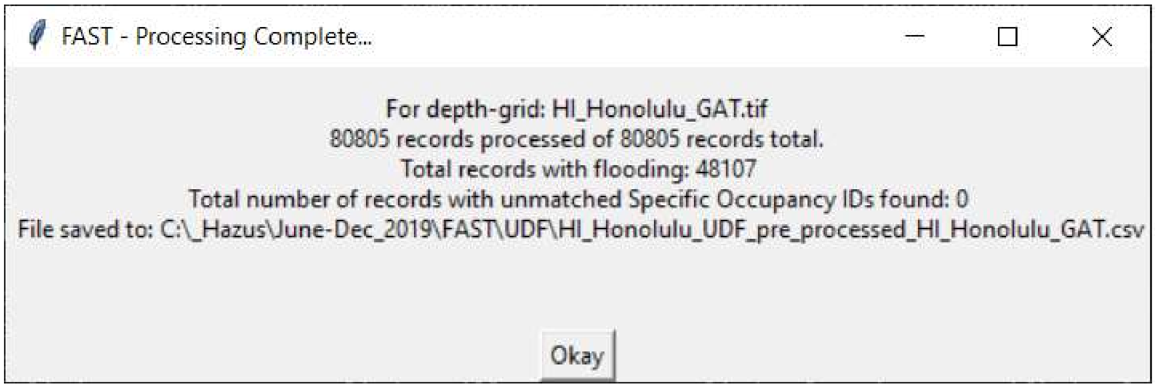


FAST - Flood Assessment Structure Tool

| | |
|---|---|
| User Defined Flty Id*: | 🟩 |
| Occupancy Class*: | 🟩 |
| Building Cost*: | 🟩 |
| Building Area*: | 🟩 |
| Number of Stories*: | 🟩 |
| Foundation Type*: | 🟩 |
| First Floor Height*: | 🟩 |
| Content Cost: | 🟩 |
| Building DDF: | 🟩 |
| Content DDF: | 🟨 |
| Inventory DDF: | 🟨 |
| Inventory Cost: | 🟨 |
| Specific Occupancy ID: | 🟨 |
| Latitude*: | 🟩 |
| Longitude*: | 🟩 |
| Coastal Flooding attribute (flC)*: | Riverine / [✓ CoastalV] / CoastalA |
| Depth Grid (ft)**: | [✓ HI_Honolulu_GAT.tif] / ND_Minot_PNNL100.tif / ND_Minot_PNNL500.tif |

* indicates required field.
**Press the ctrl key to process multiple depth grids.
Fields named similar to defaults are searched for.
Red fields are required and must be mapped.
Green fields have been mapped successfully.
Yellow fields have not been mapped, but are not required.

4. Then press ***Execute***. After the program finished you should have a summary window like this:



FAST - Processing Complete...

For depth-grid: HI_Honolulu_GAT.tif
80805 records processed of 80805 records total.
Total records with flooding: 48107
Total number of records with unmatched Specific Occupancy IDs found: 0
File saved to: C:_Hazus\June-Dec_2019\FAST\UDF\HI_Honolulu_UDF_pre_processed_HI_Honolulu_GAT.csv

Wait for the program to run (check the log window for any changes or errors),
then go to the UDF folder for the new processed .csv file.

**Troubleshooting:**

If the required fields aren't found using either the given or default field names, the program cannot run.

If a .csv file is not selected, the program cannot run.

NOTES:

Rasters used for processing must have geographic coordinate systems, or UTM. If UTM the program will reproject into WGS84 geographic. The input UDF must have Latitude and Longitude in decimal degree format.

NOTE: Coordinate transforms may result in very small changes to depth-based calculations. The Honolulu dataset below summarizes the typical differences. Small changes occur as a result of the transformations. These are very minimal with geographic transformation and slightly larger (<0.3%) when projecting to geographic based on a UTM depth grid input:

| Honolulu | | Diff | | | |
|---|---|---|---|---|---|
| $ | 14,611,808,813 | 0.0000% | Bldg losses, WGS84 geographic | | |
| $ | 14,612,663,276 | 0.0058% | Bldg losses, NAD83 geographic | | |
| $ | 14,568,790,444 | -0.2944% | Bldg losses, UTMZ4N projected | | |

Enhancements made to the tool as of 8/29/19

1. Error logging and Messaging – The application stores basic log in the ../Log/app.log
2. Field names and aliases can be modified by the user in the file ../Python_env/guiprogram.py to match with the input data so that the user does not have to type them at each run.
3. RES2 pre-processing fixed (was not assigning DDF_ID)
4. The Browse button now opens to the UDF folder

5. The records with depths less than or equal to 0 (from the depth grid raster) are not processed during loss calcs - Modified the DOGAMI code. This was resulting in large debris estimates when NODATA values of 0 are used.
6. Title modified to FAST - Flood Assessment Structure Tool
7. The final message now displays - Total records processed, Total records that had flooding, Total records that had unmatched specific occupancy Ids (these rows are presented in the results tables with "unmatched" building damage function so that the user can troubleshoot input errors.)
8. The location and name of the file that includes the results is now displayed on the final message
9. In addition to the normal results file there is another results file created with **_sorted** in the name which has results sorted on the Depth in structure field values
10. Special case for debris calculations handled for COM6 & RES2 with foundation type 4.
11. Minimized string hard coding as much as possible
12. The tool appears on the center of the screen
13. The Quit button now closes the complete app
14. The .bat background window is now minimized
15. The messages on the main screen are left justified at the bottom of the page
16. Adjusted for max float processing over integer
17. Account for missing values in input UDF – Checks added for NULL values in the input data
18. All background error messages fixed
19. Rowcount issues between the normal and sorted result file resolved

20. The number of structures flooded in the UI matches to the number of structure where flood exposure is > 1.