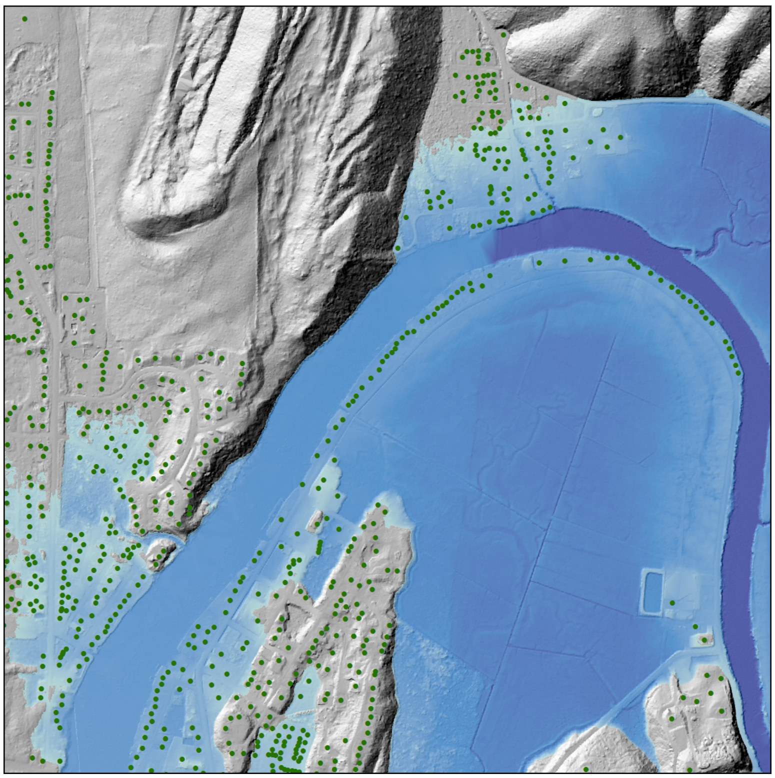
State of Oregon
Oregon Department of Geology and Mineral Industries
Brad Avy, State Geologist

OPEN-FILE REPORT O-18-04

ARCGIS PYTHON SCRIPT ALTERNATIVE TO THE HAZUS-MH FLOOD MODEL FOR USER-DEFINED FACILITIES

USER GUIDE

by John M. Bauer[1]



2018

[1]Oregon Department of Geology and Mineral Industries, 800 NE Oregon Street, Suite 965, Portland, OR 97232

DISCLAIMER

Cover image: 100-year flood depth grid (in blue) with buildings represented as points on a lidar-derived digital terrain model, Pacific City, Oregon

Oregon Department of Geology and Mineral Industries Open-File Report O-18-04
Published in conformance with ORS 516.030

For additional information:
Administrative Offices
800 NE Oregon Street, Suite 965
Portland, OR 97232
Telephone (971) 673-1555
http://www.oregongeology.org
http://oregon.gov/DOGAMI/

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACCOMPANYING DATA AND PROGRAMS

See the digital publication folder for files.

DOGAMI Hazus Flood Script v3p2.py

DDF_Hazus4p0_LookupTables

folder, containing the following CSV (comma-separated values) files:

```
Building_DDF_CoastalA_LUT_Hazus4.0.csv
Building_DDF_CoastalV_LUT_Hazus4.0.csv
Building_DDF_Riverine_LUT_Hazus4p0.csv
Content_DDF_CoastalA_LUT_Hazus4p0.csv
Content_DDF_CoastalV_LUT_Hazus4p0.csv
Content_DDF_Riverine_LUT_Hazus4p0.csv
flBldgContDmgFn.csv
flBldgEconParamOwnerOccupied.csv
flBldgEconParamRecaptureFactors.csv
flBldgEconParamRental.csv
flBldgEconParamSalesAndInv.csv
flBldgEconParamWageCapitalIncome.csv
flBldgInvDmgFn.csv
flBldgStructDmgFn.csv
flDebris_LUT.csv
flRsFnGBS_LUT.csv
Inventory_DDF_LUT_Hazus4p0.csv
```

Note: "_LUT" appended to a table name indicates that the table is a modification of the original SQL Server table.

1.0 OVERVIEW

Hazus® is a geographic information system (GIS)-based tool and set of methods for loss estimation from natural hazards. Hazus is developed and supported by the Federal Emergency Management Agency (FEMA). The Oregon Department of Geology and Mineral Industries (DOGAMI) ArcGIS® Python® Script Alternative (hereafter, “script”) is intended to complement a structure-level Hazus analysis of flood risk by providing rapid estimates of damage to building, content, and inventory, building debris, and building repair/replacement times, for a given flood depth grid or set of flood depth grids. Users may specify particular depth-damage functions (DDF) for a particular user-defined facility (UDF), or let the script choose the standard (default) DDF. With the rapid turnaround, users can more quickly evaluate their UDF parameters for accuracy and pursue in-depth sensitivity analyses. The script is targeted for users who have developed flood depth grids outside of the Hazus-MH flood model, especially for users with high-resolution flood depth grid(s) derived from lidar-based digital elevation models.

The script achieves a significant improvement in performance by avoiding the creation of redundant copies and unnecessary geoprocessing of the flood depth grid(s), and bypasses the Comprehensive Data Management System (CDMS) UDF import process. It simply queries for the flood depth at all UDF points and implements the Hazus-MH flood loss methods to calculate loss estimates. In addition, the UDF per-record processing is about 10 times faster than the Hazus-MH flood model.

Coastal flooding is supported in a straightforward manner—a simple optional attribute identifies the UDF as being placed in a Coastal AE or a V/VE Zone. If the attribute is not supplied, or is blank, the script assumes a Riverine flooding scenario. The script will then apply the appropriate DDF for that UDF.

The script provides additional information that is not currently available with the existing Hazus-MH flood model: building debris and building repair/replacement time estimates. Debris is calculated by following the methods outlined in the Hazus Flood Technical Manual (FEMA, 2011). The script provides estimates on the time needed to repair or replace the damaged building, providing more compelling information on the impact of a flood. The temporal element is derived following the Hazus methods specified by FEMA (2011).

While the script implements all functionality of the Hazus-MH flood model for UDFs, it does not implement all features of the Hazus-MH flood model, such as analysis of Essential Facilities or non-building structures such as water treatment plants.

An analyst with moderate Python programming language skills can add additional functionality. We encourage users to modify the script for their needs, keeping in mind that any new estimates or capabilities should always be validated against the Hazus-MH flood model itself, when possible, or rigorously hand-checked.

This document is written primarily as a user guide, providing instructions on installing and running the script and information on the script’s output. The appendices provide more detailed information on implementation and validation.

2.0 SOFTWARE REQUIREMENTS

- Esri® ArcGIS® Desktop 10.x Basic License, with ArcGIS Spatial Analyst extension. A Hazus-MH installation is **not** required. Python 2.7 should be installed as part of the ArcGIS installation.
- A copy of the accompanying Python script and supporting contents stored at a fixed location. The script, supporting contents, and user data, including flood depth grids, can reside on a network drive.
- Appendix A provides instructions on integrating the script as an ArcToolBox® Python script. The following guide assumes the user has done so.

3.0 INPUT REQUIREMENTS

3.1 User-Defined Facility Geodata

The user supplies a user-defined facilities (UDF) GIS pointfile to the script. Seven attributes are required and must be fully populated (**Table 14-1**). Six other attributes are optional, with two of the six having no Hazus-MH flood equivalent. The input attributes must conform to Hazus naming and capitalization listed in **Table 14-1**. Python variable names are case-sensitive. If an optional attribute's name or capitalization is at variance with what is specified in **Table 14-1**, the script will assume the attribute was not supplied. Note the minor capitalization inconsistency with *BldgDamageFnID* suffix, compared to *ContDamageFnId*, and *InvDamageFnId*. This is per Hazus naming convention, which we retained for compatibility purposes. The UDF Input file should be a file or personal geodatabase feature class. The shapefile format limits attribute names to 10 characters, and many attribute names in **Table 14-1** exceed 10 characters in length.

In practice, Python is capable of handling other data types—for example, *FoundationType* can be an Integer type. However, we encourage conformance to the Hazus-MH data type conventions.

As with the Hazus-MH flood model, if a building's content cost (*ContentCost*) attribute is supplied and that value is not null, the script will use the provided value when computing content loss. Otherwise, it will calculate content cost based on user-supplied building cost, using the ratios specified by FEMA (2011, Table 14.6).

The *BldgDamageFnID*, *ContDamageFnId*, and *InvDamageFnId* are optional fields. If they are present in the UDF input file as an attribute, and, for a given UDF record, if they are populated with a legitimate value, the script will use the specified DDF. Otherwise, the script will use the standard (default) DDF based on the UDF's specific *OccupancyClass, FoundationType*, and *NumStories* values. Section 13.1 provides additional details on the definition of standard (default) DDFs.

Two attributes (*flC*, *InvCost*) have no equivalent in the Hazus-MH flood model. We created the *flC* attribute to provide a more convenient, straightforward method for implementing the Hazus loss estimation for structures in zones designated as subject to coastal flooding. A simple overlay with a Special Flood Hazard Area map can identify buildings (UDFs) in zones subject to coastal flooding. The particular record's *flC* attribute can then be assigned a 'V', 'VE', or 'CAE' (for Coastal AE Zone) value. Any other values, including white space, are ignored by the script. If a user-supplied DDF ID is not supplied, and the *flC* attribute is assigned with 'V', 'VE', or 'CAE', the script will use the standard coastal flooding DDF. If a user-supplied DDF is not supplied, and the *flC* attribute is some other value, or not present in the input file, the script will use the standard (default) riverine DDF.

Inventory Cost (*InvCost*) allows a user to directly specify an inventory cost, if available. If the *InvCost* attribute is not present in the input UDF file, or if the *InvCost* attribute is present in the UDF file but has a value null for that building record, the script will compute the inventory cost per the standard Hazus methods (FEMA, 2011, Eqn. 14-4).

The script ignores any other attributes in the UDF input file that do not match the names in **Table 14-1**, including the following attributes that typically exist with a Hazus-MH UDF input file. If they happen to be in the UDF input file, they are simply copied to the output file:

Name, *Address*, *City*, *Statea*, *Zipcode*, *Contact*, *PhoneNumber*, *YearBuilt*, *BackupPower*, *ShelterCapacity*, *Latitude*, *Longitude*, *Comment*, *BldgType*, *DesignLevel*, *FloodProtection*

If a user's organization has their own UDF field naming conventions that differ from the Hazus naming/capitalization conventions, the user can modify the script's code. Search for the `specify your names here` text string.

3.2 User-Supplied Depth-Damage Functions

Users may supply their own custom depth-damage functions by appending to the appropriate CSV files in the full DDF lookup table library:

- Building Damage: `flBldgStructDmgFn.csv`
- Content Damage: `flBldgContDmgFn.csv`
- Inventory Damage: `flBldgInvDmgFn.csv`

The damage function identifier (*BldgDamageFnID*, *ContDamageFnId*, *InvDamageFnId*) must be unique from all other values in the table. We suggest the user gives the identifier a longer string name in order to clearly distinguish a user-supplied DDF, such as `50001`. The DDF table values are in percent and must be filled out for all depths, from minus 4 foot to plus 24 foot. If the user's DDF information does not have information up to 24-foot flood depth level, we advise the user to simply extend the damage percentage from the deepest flood depth available, up to the 24-foot depth.

3.3 Flood Depth Grids

One or more flood depth grids must be supplied as input to the script. The flood depth values must be in feet. Flood depth grids can be in any Esri-supported raster format and in any projection.

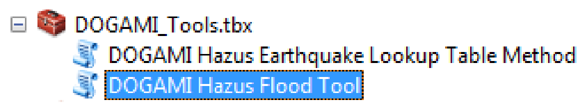
The script's execution times are considerably faster if the flood depth grids are in TIF, IMG, or Esri file geodatabase format, versus the standard Esri grid format. The reason is that the legacy Esri grid format stores 4 bytes for each `NoData` pixel. For typical flood depth grids containing large areas of `NoData`, the grids will be significantly inflated in size and thus slow down geoprocessing times. Modern raster formats efficiently handle the storage of `NoData` pixels using various lossless compression techniques. While Hazus-MH v4.2 now permits the user to specify flood depth grids in other formats, its internal processing and storage continues to use the legacy Esri grid format.

Note that the Esri Spatial Analyst cannot process rasters whose full pathname contains any spaces.

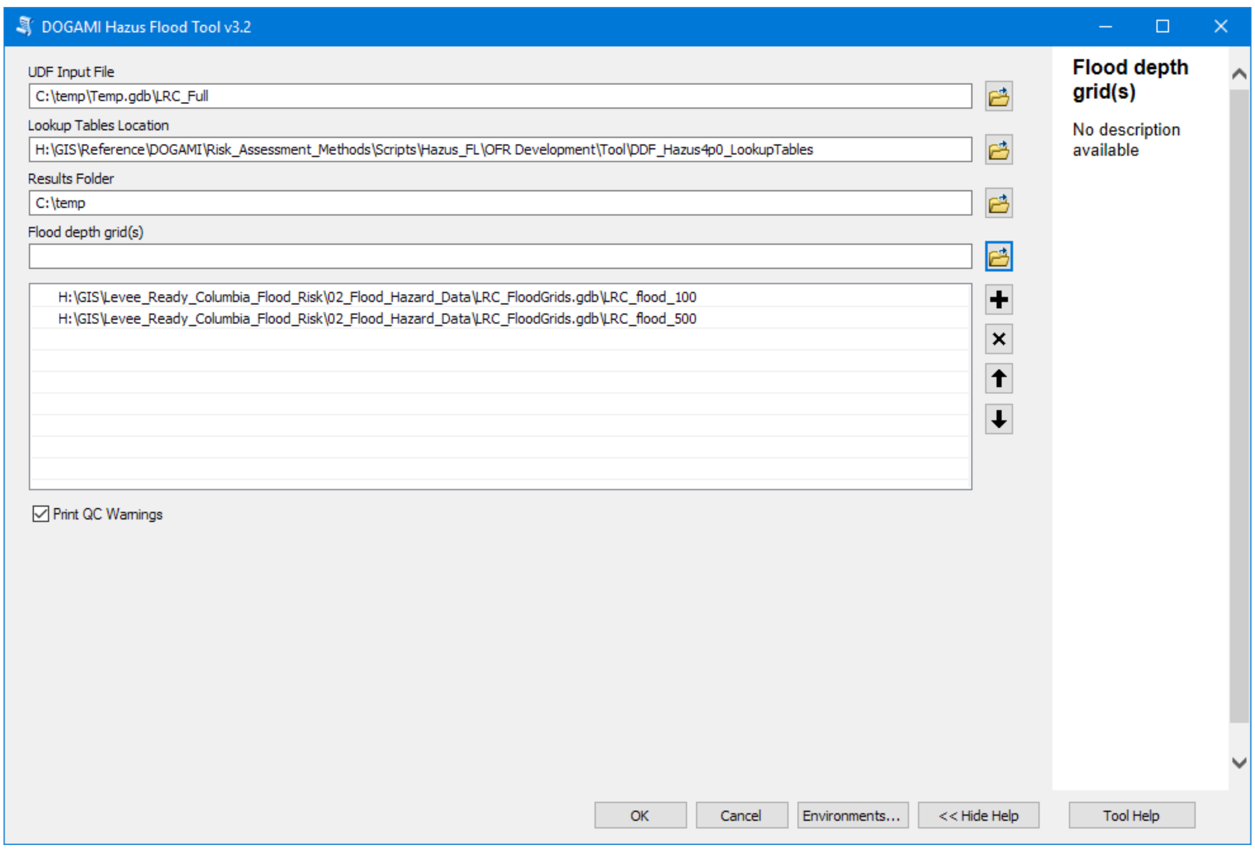
4.0 RUNNING THE SCRIPT

The script can be run in a command window with arguments passed to it on the command line. More conveniently, one can import the script into an ArcToolbox and run it within an Esri ArcMap®/ArcCatalog® session (see ArcToolbox Installation section in Appendix A).

Click on the `Hazus Flood` tool (or the name you gave it during installation):

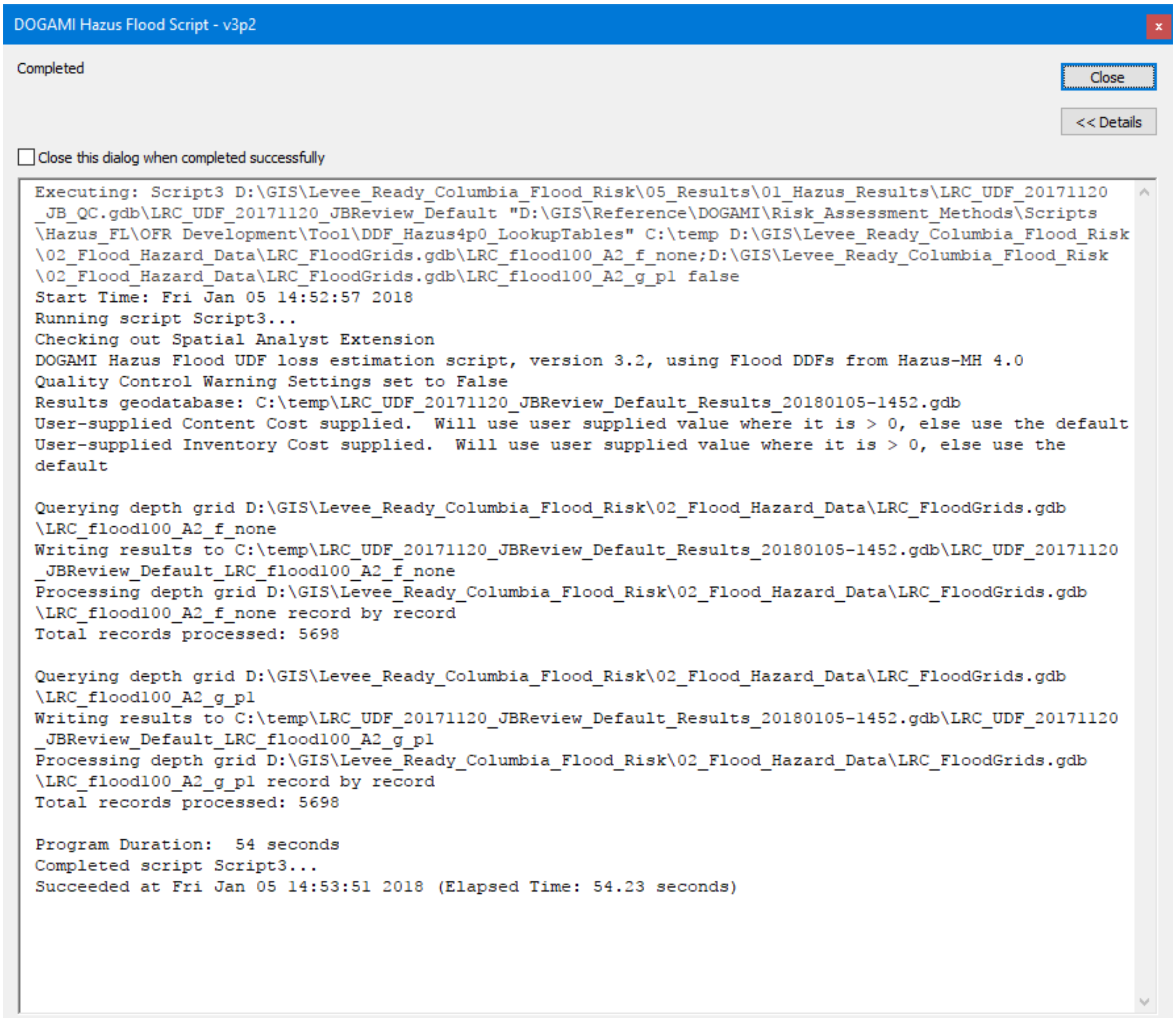


The interface should look similar to this (note that spaces are not permitted in grid path names):



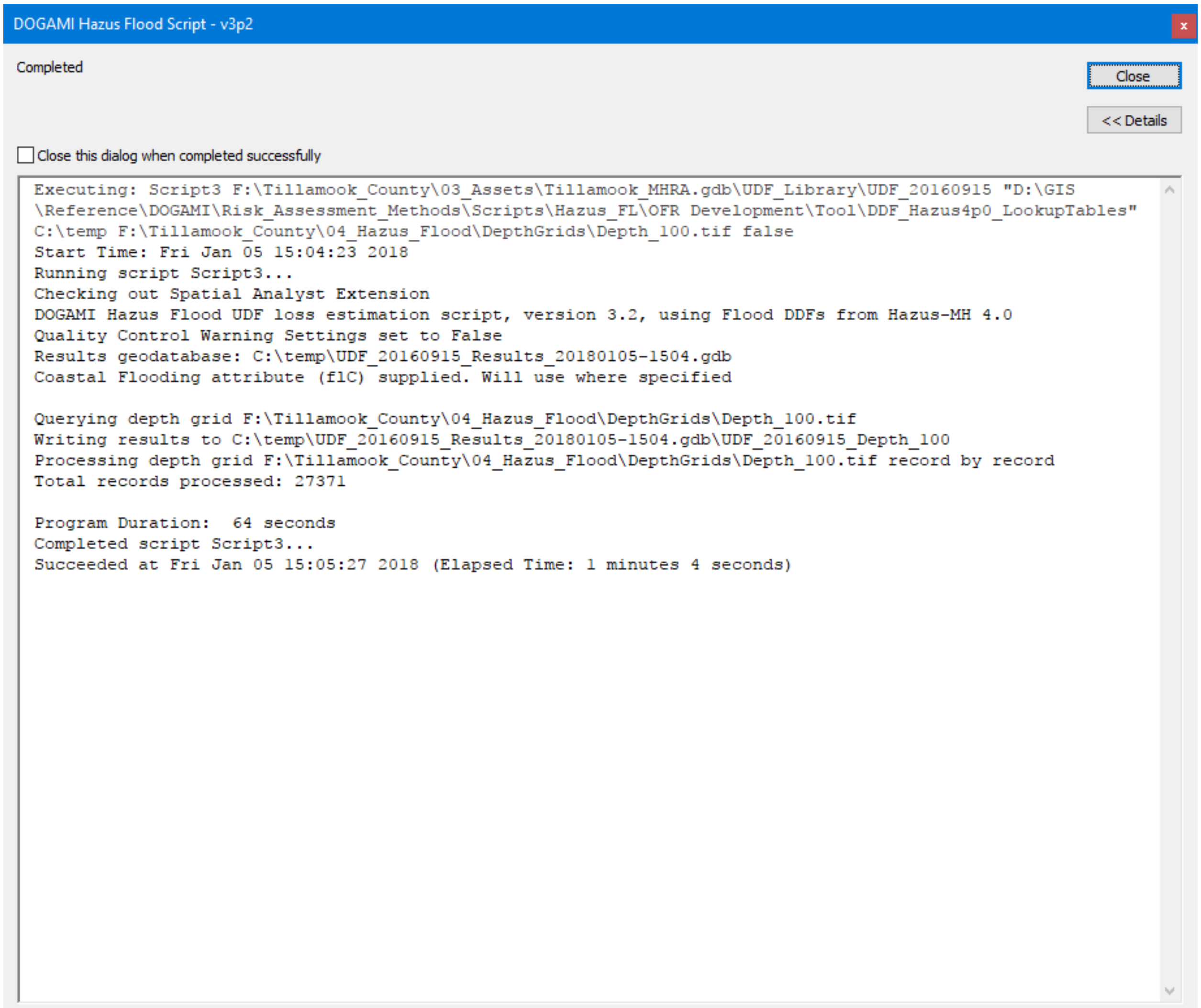
The Lookup Tables Location field should be filled in by default. Your lookup tables location will differ from what is listed here. Specify your UDF input file, Flood depth raster grid(s), and Results Folder location using full path names. You can also specify if Quality Control warnings should be printed or suppressed.

The results window contains miscellaneous information. In the run shown below, the user specified two depth grids. No user-supplied Coastal Flooding attribute or damage function attributes were supplied:

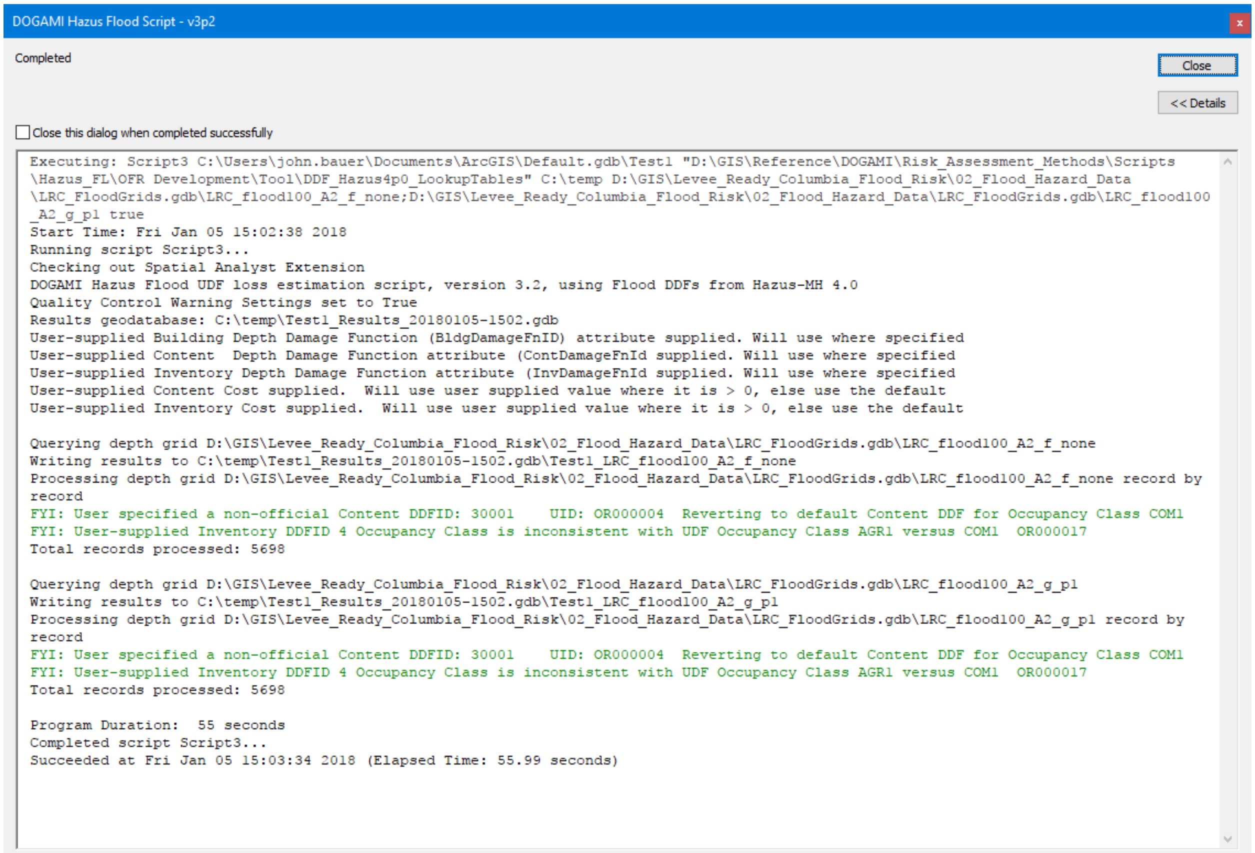



```
Executing: Script3 D:\GIS\Levee_Ready_Columbia_Flood_Risk\05_Results\01_Hazus_Results\LRC_UDF_20171120
_JB_QC.gdb\LRC_UDF_20171120_JBReview_Default "D:\GIS\Reference\DOGAMI\Risk_Assessment_Methods\Scripts
\Hazus_FL\OFR Development\Tool\DDF_Hazus4p0_LookupTables" C:\temp D:\GIS\Levee_Ready_Columbia_Flood_Risk
\02_Flood_Hazard_Data\LRC_FloodGrids.gdb\LRC_flood100_A2_f_none;D:\GIS\Levee_Ready_Columbia_Flood_Risk
\02_Flood_Hazard_Data\LRC_FloodGrids.gdb\LRC_flood100_A2_g_pl false
Start Time: Fri Jan 05 14:52:57 2018
Running script Script3...
Checking out Spatial Analyst Extension
DOGAMI Hazus Flood UDF loss estimation script, version 3.2, using Flood DDFs from Hazus-MH 4.0
Quality Control Warning Settings set to False
Results geodatabase: C:\temp\LRC_UDF_20171120_JBReview_Default_Results_20180105-1452.gdb
User-supplied Content Cost supplied.  Will use user supplied value where it is > 0, else use the default
User-supplied Inventory Cost supplied.  Will use user supplied value where it is > 0, else use the
default

Querying depth grid D:\GIS\Levee_Ready_Columbia_Flood_Risk\02_Flood_Hazard_Data\LRC_FloodGrids.gdb
\LRC_flood100_A2_f_none
Writing results to C:\temp\LRC_UDF_20171120_JBReview_Default_Results_20180105-1452.gdb\LRC_UDF_20171120
_JBReview_Default_LRC_flood100_A2_f_none
Processing depth grid D:\GIS\Levee_Ready_Columbia_Flood_Risk\02_Flood_Hazard_Data\LRC_FloodGrids.gdb
\LRC_flood100_A2_f_none record by record
Total records processed: 5698

Querying depth grid D:\GIS\Levee_Ready_Columbia_Flood_Risk\02_Flood_Hazard_Data\LRC_FloodGrids.gdb
\LRC_flood100_A2_g_pl
Writing results to C:\temp\LRC_UDF_20171120_JBReview_Default_Results_20180105-1452.gdb\LRC_UDF_20171120
_JBReview_Default_LRC_flood100_A2_g_pl
Processing depth grid D:\GIS\Levee_Ready_Columbia_Flood_Risk\02_Flood_Hazard_Data\LRC_FloodGrids.gdb
\LRC_flood100_A2_g_pl record by record
Total records processed: 5698

Program Duration:  54 seconds
Completed script Script3...
Succeeded at Fri Jan 05 14:53:51 2018 (Elapsed Time: 54.23 seconds)
```

The run shown below had a coastal flooding attribute specified. No user-supplied DDFs were provided.



The run shown below had two flood depth grids specified. The user supplied custom DDF IDs. Warnings indicating the user-supplied DDF ID was inconsistent with the Occupancy Class are displayed. One of the DDF IDs was not recognized, and it was discarded. The messages are repeated twice because the input UDF file was evaluated twice.



5.0 MODEL OUTPUT

5.1 Results File Geodatabase and Feature Class

The script creates a file geodatabase in the folder you specified. The name of the file geodatabase will be the root name of your UDF input file along with a date and time stamp suffix; e.g., for a run made on December 18, 2017, at 14:49:

UDF_Salem_20171218-1449.gdb

In the output file geodatabase, one feature class is created for each flood depth grid you specify. The feature class name is the UDF input file name appended with the specific flood depth grid name. The UDF input file is copied into the output file geodatabase, with 18 additional attributes added and populated by the script (**Table 14-2**). Five of the output attributes are identical to the Hazus-MH flood model result file. We added 13 additional attributes that are not present in a Hazus-MH flood results file. We maintain that they have informative and archival value, such as the actual depth grid sampled at the UDF point and the specific DDF ID that the script used for accessing the lookup tables.

5.2 Building Repair/Replacement Times

The script does not fully implement the direct economic loss factors; instead, it provides a minimum and maximum repair or restoration time per structure, given the depth-in-structure and its occupancy class. The repair times form the basis for other direct economic loss estimates specified by FEMA (2011), namely, relocation expenses ($RT_{i,j}$ in FEMA Eqn. 14-6), capital-related income loss ($LOF_{i,j}$ in FEMA Eqn. 14-7), wage loss ($LOF_{i,j}$ in FEMA Eqn. 14-7), and rental income loss ($RT_{i,j}$ in FEMA Eqn. 14-8).

Displaced population can be calculated in a temporal dimension as follows: If the user has an estimate of the number of permanent residents per residential structure, the user can simply multiply the permanent residents of the building with its estimated repair/replacement time. The user can choose either the minimum or the maximum value for building repair/replacement time. We note that the minimum repair time should be used judiciously for such calculations. While a residential building may not sustain any direct loss, authorities may restrict access to it due to the building being surrounded by floodwaters.

6.0 MISCELLANEOUS NOTES

The script was developed using ArcMap 10.4 and Python 2.7.10. We also tested it with ArcGIS Pro 2.0.1 and Python 3.5.3. No differences were observed in the results between the two versions when using a common UDF dataset and flood depth grids. However, our more extensive validation (Appendix B) was done using ArcMap 10.4 and Python 2.7.10 and not in the ArcGIS Pro environment.

7.0 KNOWN DIFFERENCES WITH THE HAZUS-MH FLOOD MODEL

Performance

The script processes UDF records at about 10,000 per minute. The Hazus-MH flood model, even after the flood depth grids are imported and processed, processes UDF records at a rate of about 1,000 per minute. Such a performance difference may be compelling in certain applications.

Coastal Flooding Support

The script provides straightforward support for UDFs in coastal flooding zones by simply identifying the structure is in a V/VE Zone or Coastal A zone. There is no equivalent in the Hazus-MH flood model. As of Hazus-MH v4.0, users had to provide their own DDF ID for structures in a coastal flooding zone (FEMA, 2017). As of Hazus-MH v4.2, the user release notes (FEMA, 2018, p. 5) indicate the following, but due to publication timing constraints, we were not able to validate this nor incorporate it into our script:

> A UDF without an assigned DDF by the user will now use Coastal DDFs for Coastal hazards using a type of depth approach similar to Hazus GBS. For structures at a depth greater than 2 feet, a Coastal V DDF is used and for structures in 2 feet or less of flood depth, a Coastal A DDF is used. For Content and Inventory, 4 feet is the boundary between using Coastal A and V DDFs.

Neither the Hazus-MH flood model nor the script performs any additional adjustment to depth-in-structure for UDFs in coastal flooding zones. Some users expressed the need for an additional 1-foot depth adjustment in coastal zones (a freeboard adjustment), in order to be more conservative, but neither one currently makes such an adjustment.

Flood Depth Grid Processing

The Hazus-MH flood model removes negative values in the user-provided flood depth grid by setting those cell values to `NoData`. This has an effect of creating apparent islands. The script does no such processing. Negative values of depth grids, if present, are retained. The depth-in-structure calculation remains the same. The effect can sometimes be seen in coastal flooding areas, where houses on "islands" that may have a negative depth may appear to be above the flood level, but because the DDF begins at −4 feet, the house will record some damage. In practice, this difference will likely be limited to very few structures.

Latent Bugs

The Hazus-MH flood model (v4.0) uses an obsolete default DDF for riverine RES1, 1-story, Basement ('R11B'), leading to moderate underestimate of loss for such structures. See Appendix B for more details.

The script implements the Inventory Loss calculation correctly. Note that prior to Hazus-MH v4.2, the inventory loss calculation was overestimated by the Hazus model.

Minor Processing Differences

For extreme flood depths, where flood depth is greater than 24 feet, the script first subtracts the *FirstFloorHt* from the depth to derive depth-in-structure; then, if the depth-in-structure is still more than 24 feet, the script caps the depth to 24 feet. The Hazus-MH flood model flips the order: it caps the depth grid to 24 feet, then does the *FirstFloorHt* adjustment to get depth-in-structure. We maintain the script implementation to be more correct, though in practice, the impact should be relatively minor, with very few structures impacted.

The Hazus-MH flood model rounds the *FirstFloorHt* to the nearest 0.5 foot prior to the depth-in-structure adjustment, then does the DDF calculation. The script does no such rounding and uses the full precision of the *FirstFloorHt* to determine depth-in-structure. The overall effect is likely neutral; individually, though, loss percentages may slightly vary between the script and the Hazus-MH flood model, depending if the user supplies fractional *FirstFloorHt* values. See Appendix B for more details.

Quality Checks and Support

The script checks for consistency between a user-supplied DDF ID and the Occupancy Class. For example, if the user specifies an *OccupancyClass* of COM3 and a *BldgDamageFnID* = 397 for a particular record, the Hazus-MH flood model ignores the user-specified *BldgDamageFnID* and reverts to using default DDF for COM3, because *BldgDamageFnID* = 397 is for COM8 Occupancy Class. The Hazus-MH flood model gives no such warning of this substitution, however. The script will use whatever DDF ID is given to it, regardless of *OccupancyClass*, and if the DDF ID is inconsistent with the *OccupancyClass*, the script will provide a warning in the output window but will proceed with using the specified DDF. We maintain this functionality offers more flexibility to a user who may have reasons to assign a specific DDF to a particular structure, regardless of its Occupancy Class.

The Hazus-MH flood model does not provide a permanent record of what DDF ID it used. In such cases as mentioned in the previous paragraph, where, say, a *BldgDamageFnID* was explicitly provided but ignored by Hazus, the record can be misleading. Further, the standard DDF library has changed with time, and may change in the future. We contend that a permanent record of the particular DDF IDs used for each UDF record has archival value. Such information is contained in the *BDDF_ID*, *CDDF_ID*, and *IDDF_ID* attributes in the results file.

Feature Support

The script provides debris estimates on a per-building basis. The Hazus-MH flood model currently provides no debris estimate at the structural level.

The script provides a minimum and a maximum restoration time estimate per building. The Hazus-MH flood model currently provides no such estimates for UDF analysis. For General Building Stock-based loss estimation, the Hazus-MH flood model uses the values to compute a direct economic impact loss but does not provide any sense of temporal range for the recovery/rebuild time.

Hazus-MH flood model estimates flooding damage to the following entities using particular depth-damage functions. The DOGAMI script currently does not provide direct support for the following:

- Essential Facilities (EFs can be modeled, as with Hazus, as a UDF)
- Transportation Facilities: Bridges
- Utilities (typically consisting of non-building structures such as water treatment plants)
- Vehicles
- Agricultural crop damage
- Average annualized loss
- Flood warning times (reductions in Content and Inventory loss)

8.0 RECOMMENDATIONS

The script does not calculate direct economic losses such as relocation expenses, capital-related income loss, wage loss, rental income loss, nor does it estimate average annualized loss estimates for buildings (FEMA, 2011, Chapter 14). Such information may be required for certain projects. For convenience, the lookup tables from the Hazus SQL database were exported and are provided in the lookup table library ([`flbldgEconParam*`] tables).

Additional quality checks can be added to the script to check for illegal or poorly specified input data and to cleanly handle such cases.

The script could be used as a basis to implement the Hazus flood damage model using open source software. Only one processing step—obtaining the flood depth grid value at the building point—requires geospatial tools; all other operations are tabular. An example of using the Hazus-based functions within an open source context is given by Gutenson and others (2017), whose Flood Damage Wizard built on the Hazus-MH R (statistical computing language) package constructed by Gopi Goteti (Damage functions from FEMA's HAZUS software for use in modeling financial losses from natural disasters, https://cran.r-project.org/package=hazus) .

9.0 ACKNOWLEDGMENTS

We thank Cynthia McCoy, FEMA Region X, for arranging formal feedback from Hazus developers and expert users, who provided helpful background and encouragement. James Mawby, Dewberry, Inc., provided insightful observations on the script's behavior and performance. Jesse E. Morgan II, Atkins Global, gave the Python code an in-depth technical review and suggested a number of improvements, which greatly enhanced the code's flexibility and maintenance. Doug Bausch, Pacific Disaster Center, provided technical guidance for implementing the debris calculation. Bill Bohn, Sobis, Inc., provided spreadsheets that were helpful in implementing the economic loss and debris calculations. Lastly, I thank my DOGAMI colleagues, Matthew Williams and Fletcher O'Brien, who provided encouragement and clarification on flooding methods and terminology, especially in a National Flood Insurance Program context.

10.0 REFERENCES

Federal Emergency Management Agency (FEMA), 2011, Hazus®-MH 2.1 Technical manual, Flood model: Washington, D.C., 569 p. https://www.fema.gov/media-library-data/20130726-1820-25045-8292/hzmh2_1_fl_tm.pdf

Federal Emergency Management Agency (FEMA), 2017, Hazus-MH software: FEMA's tool for estimating potential losses from natural disasters, version 4.0: Washington, D.C. https://www.fema.gov/media-library-data/1493315287435-68e5171cc8856bf36651f1ce9ba2e6fe/Hazus.4.0.User.Release.Notes.pdf

Federal Emergency Management Agency (FEMA), 2018, Risk MAP CDS Hazus® 4.2 user release notes, ver. 1.0, January 29, 2018, 16 p. Available as part of Hazus 4.2 software package.

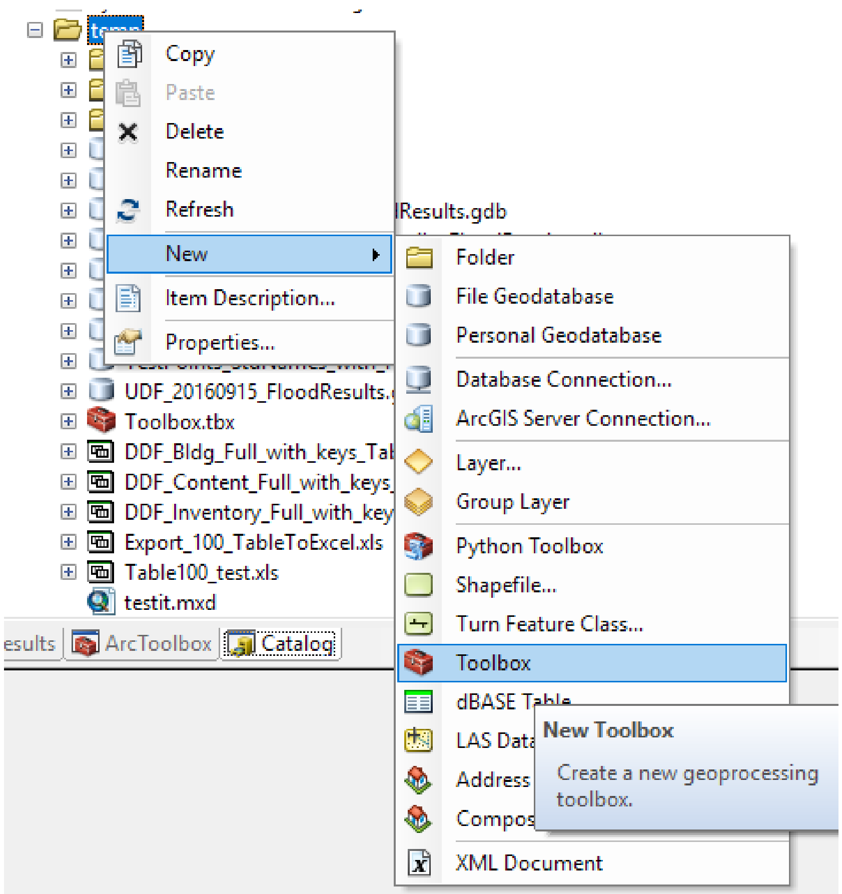
Gutenson, J. L., Ernest, A. N. S., Oubeidillah, A. A., Zhu, L., Zhang, X., and Sadeghi, S. T., 2017, Rapid flood damage prediction and forecasting using public domain cadastral and address point data with fuzzy logic algorithms: Journal of the American Water Resources Association (JAWRA), v. 54, no. 1, 104–123, https://doi.org/10.1111/1752-1688.12556

11.0 APPENDIX A: ARCTOOLBOX INSTALLATION

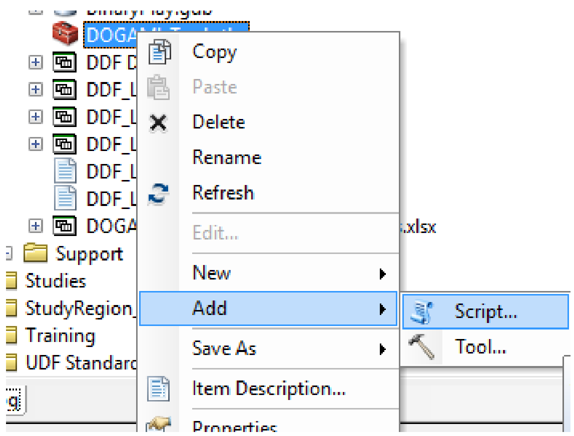
The script can be run on a Python command line, requiring five arguments. However, that invocation method is tedious, given the requirement for full pathnames in the arguments. We recommend installing the script in ArcToolbox.

First, find a fixed location for the DDF Library—in the download zip package, this is the folder `DDF_Hazus4p0_LookupTables`. This folder can be stored on a network drive or local drive, but the folder should be in a location that is stable. Record the pathname to the folder.

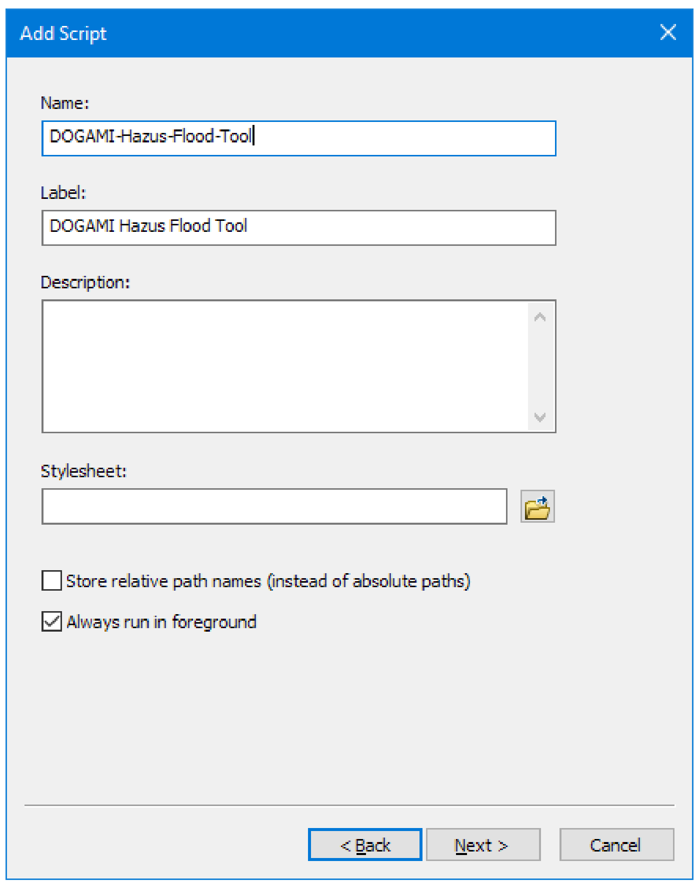
To install the `DOGAMI_Hazus_Flood_v3p2.py` as a script in an ArcToolbox: as with the DDF library, identify a stable drive location to store the script. The script can reside on a network drive or local drive. Open Esri ArcCatalog. If you do not already have a toolbox you use in the ArcGIS context, then in ArcCatalog create a toolbox by right-clicking on a folder or an existing file geodatabase (note: this is **not** a Python toolbox):



Add a script to that toolbox, with right-click:

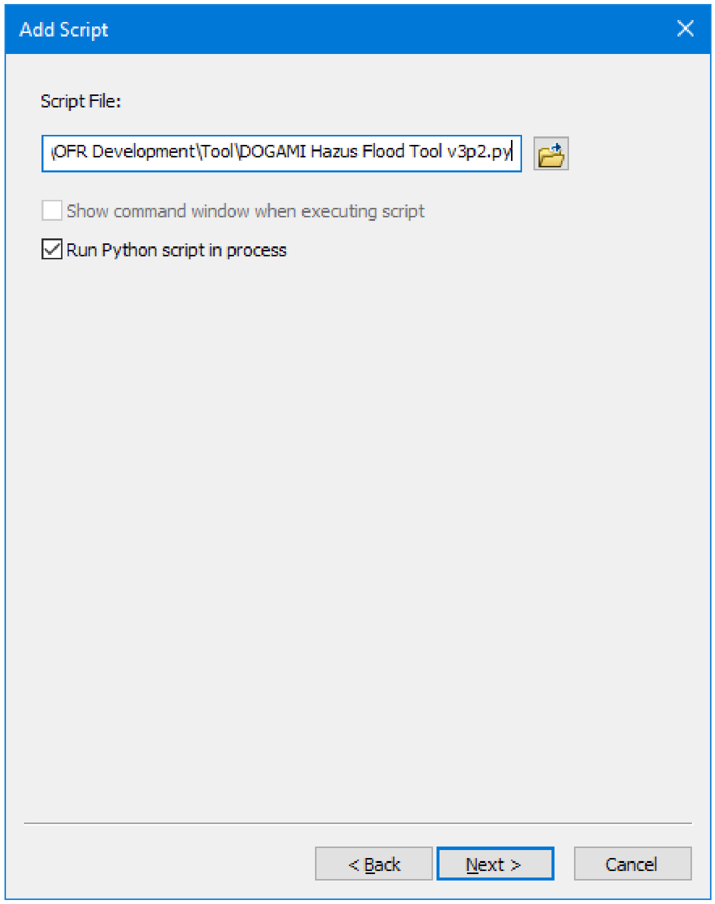


The following window appears. Give the script a [Name] and [Label] (suggested names are provided). Note that ArcToolbox has character restrictions on the script Name itself, i.e., no underscores or spaces in the Name field. Spaces are allowed in the Label field.



Click **Next.**

In the Script File field, point to the fixed location where you are storing the Python script (`DOGAMI_Hazus_Flood_v3p2.py`). Include the complete pathname and script name with file extension:



Click **Next.**

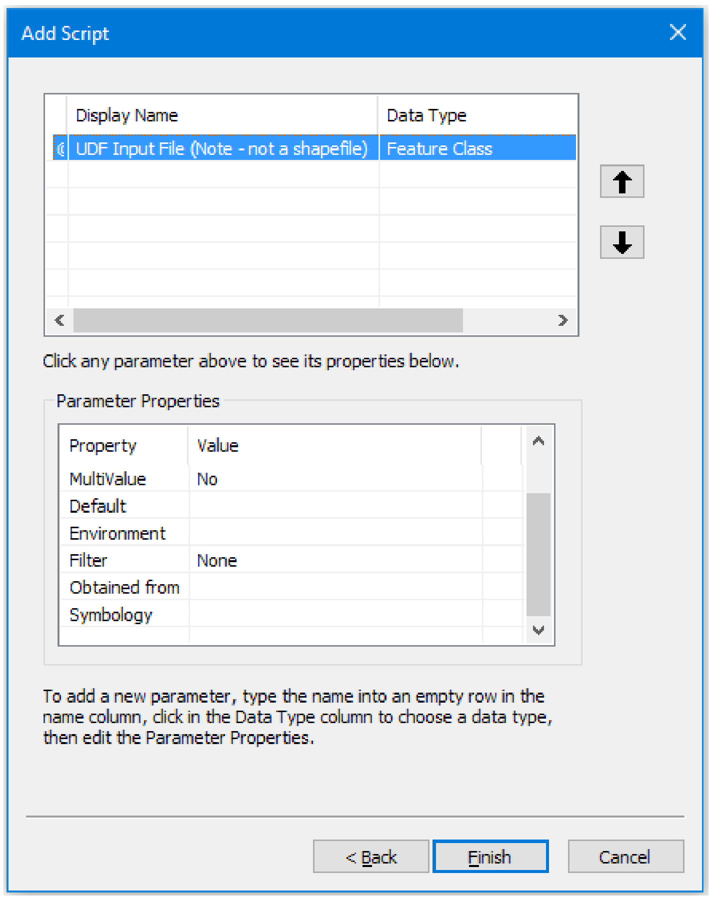
Define the five inputs to the script. The order of the inputs is fixed, as shown in the series of images below. Keep clicking in the Display Name, Data Type, and Parameter Properties fields to define all five inputs before clicking **Finish**.

First input line: The UDF Input File. In the first row, click in the **Display Name** column. We suggest the text:

`UDF Input File (not a shapefile)`

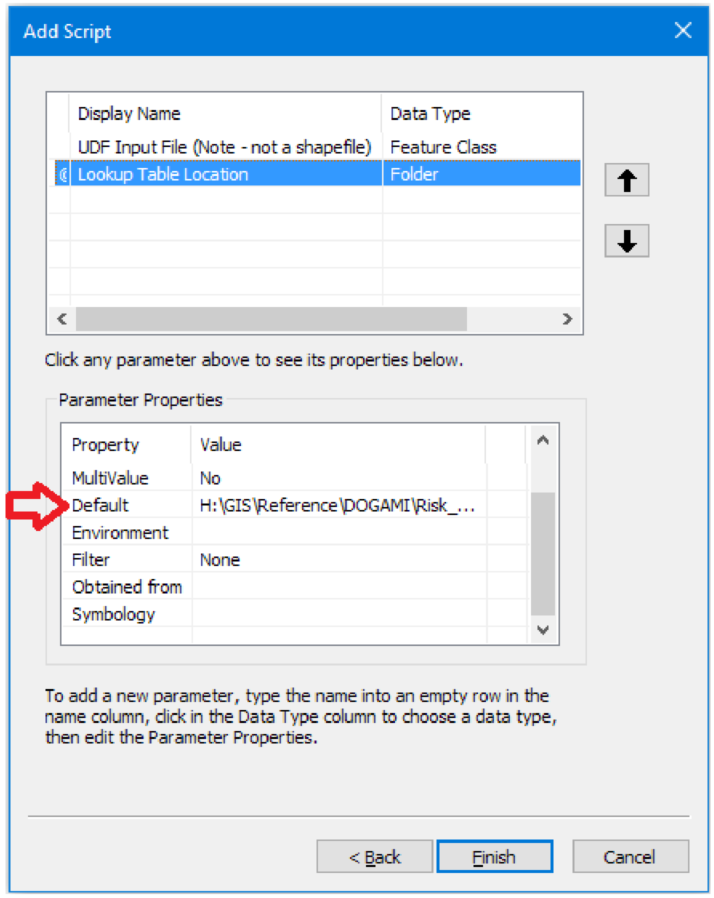
to emphasize to the user that the input must be a file or personal geodatabase feature class (due to the shapefile's 10-character limit on attribute names.)

In the **Data Type** column, use the pull-down menu to select Data Type = `Feature Class`.



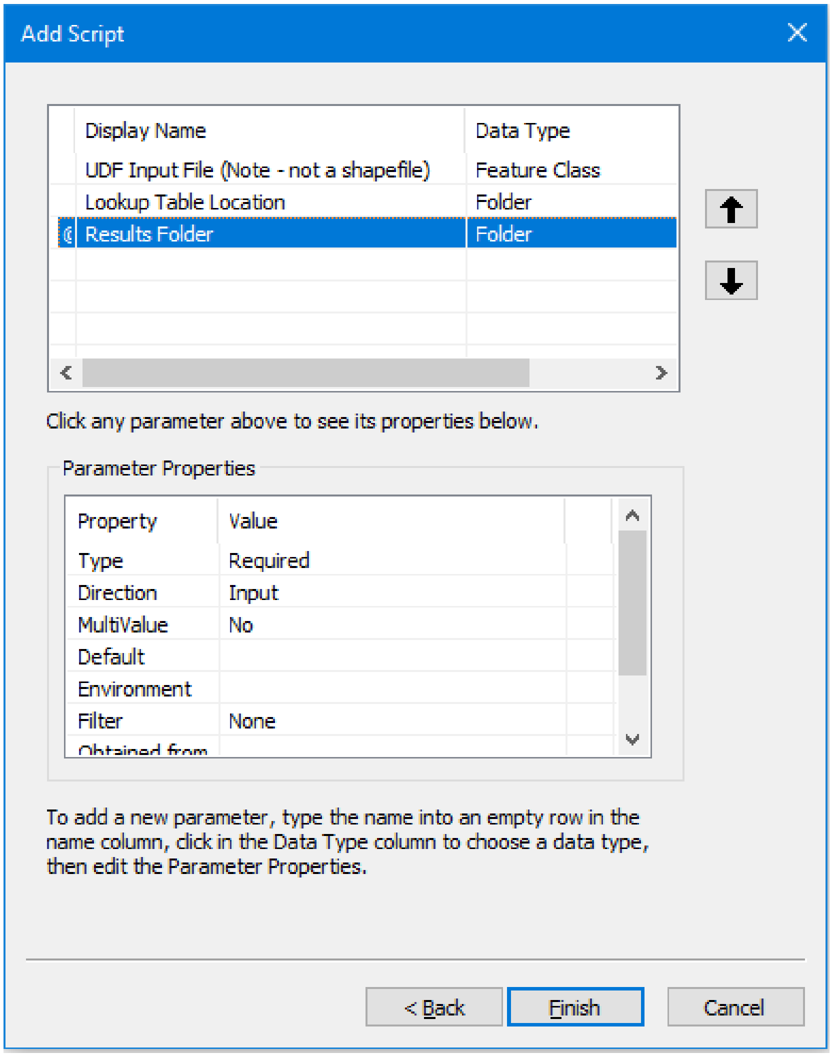
Do *not* click **Finish.**

Second input line: The DDF lookup table location is specified as a **Folder** type. We used the Display Name `Lookup Tables Location`, but you can change that name. Set the **Default** value to the folder where you stored the DDF library. Networked folder locations are permissible. Your pathname will differ from what is shown.



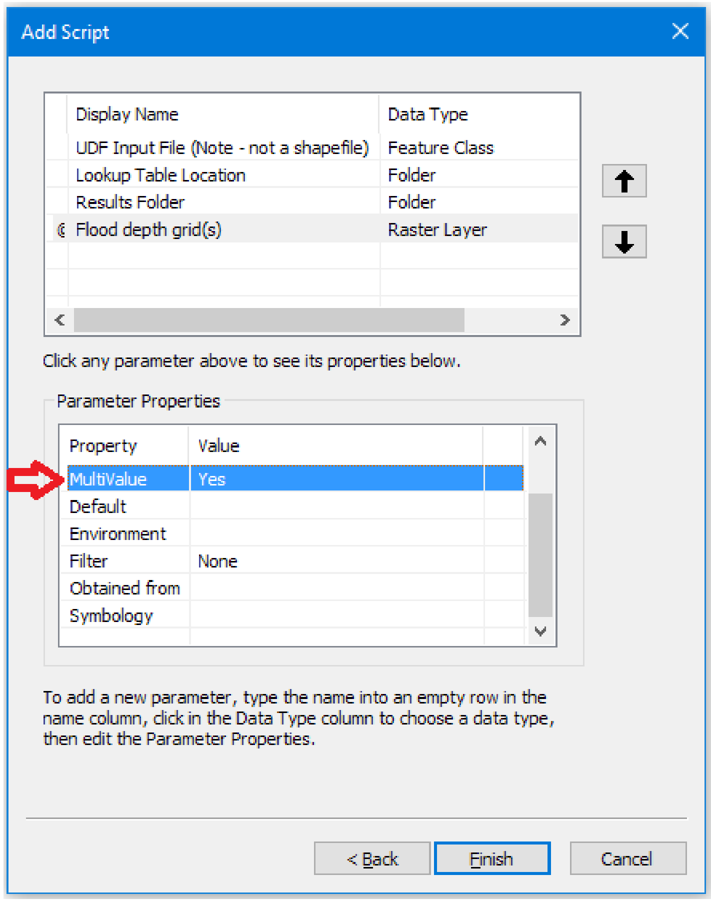
Do *not* click **Finish.**

Third input line: The Results folder is specified as the third parameter: **Folder** type. The direction is set to `Input`. Again, you can change the `Display Name` to what you wish.



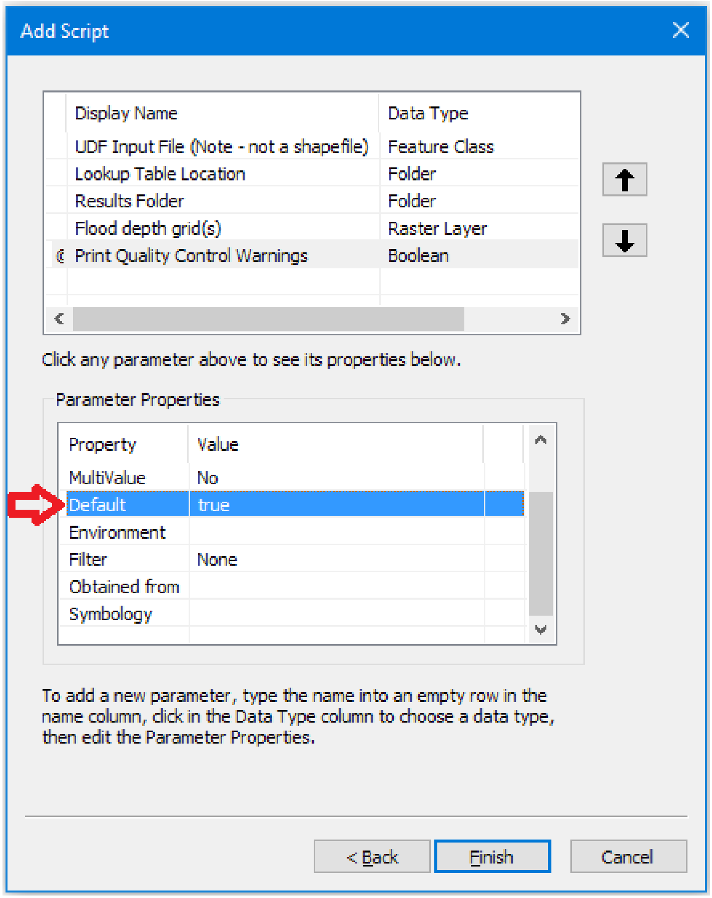
Do *not* click **Finish.**

Fourth input line: The Flood depth grid(s) are specified as **Raster Layers,** with MultiValue set to `Yes` (you may wish to provide a warning to users in the 'Display Name' field that spaces in the flood depth grid pathnames are not permitted):



Do *not* click **Finish.**

Fifth input line: Finally, the Quality Control Messages Enabled switch is set as a **Boolean** data type. We suggest its Default setting be set to `True`.



Click **Finish**.

The script should now be installed. Navigate to your ToolBox and test it out with your UDF file and flood depth grid(s).

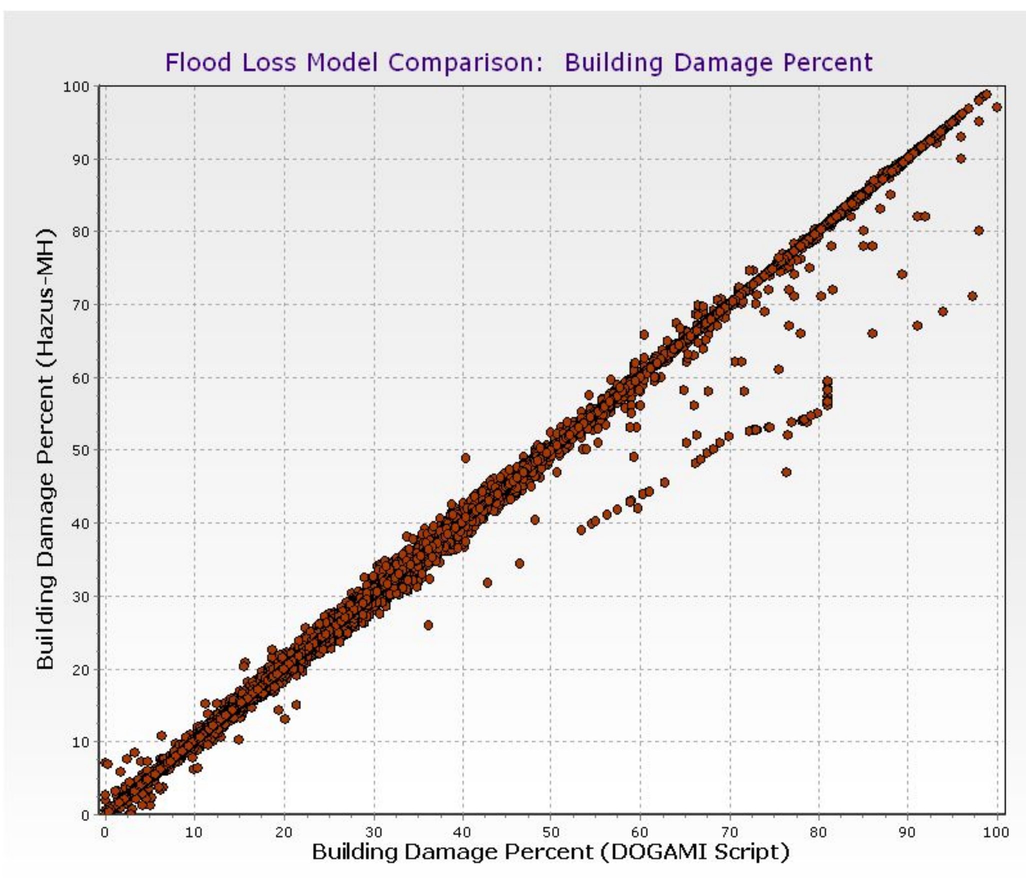
12.0 APPENDIX B: VALIDATION

To establish confidence in the script's results, we created a test scenario, wherein 40,000 UDFs were randomly placed within a depth grid that varied between 0 and 30 feet in depth. The UDFs were randomly assigned an Occupancy Class, Foundation Type, Number of Stories (Integer), and First Floor Height (Float). Bounds were set on the input parameters to assure testing within reasonable and meaningful ranges. Cost and Area were set to $100,000 and 1,000 square feet, respectively, for all structures. The UDF input file and test grid were first run with Hazus-MH v4.0 and then with the script. Loss estimates between the two methods were then directly compared by joining the results tables using the *UserDefinedFltyID* key.

Expert users are likely familiar with the slight shift the Hazus-MH flood model introduces to the flood depth grid and UDF data during the import process. The comparison between the two approaches took this slight shift into account, with the script using the Hazus-shifted UDF and flood depth grid data.

A scatter plot comparison quickly highlights any differences between the two methods. The building loss percentage estimate (*BldgDmgPct*), which varies between 0 and 100, is plotted for each point, with the script's calculation on the x-axis and the Hazus-MH flood model's calculation on the y-axis. We should see a straight 1:1 relationship. However, the scatter plot comparison using the *BldgDmgPct* output attribute suggests significant differences between the two tools (**Figure 12-1**).

Figure 12-1. Comparison of Building Damage Percent calculation between the script and the Hazus-MH flood model (*n* = 40,000 points).



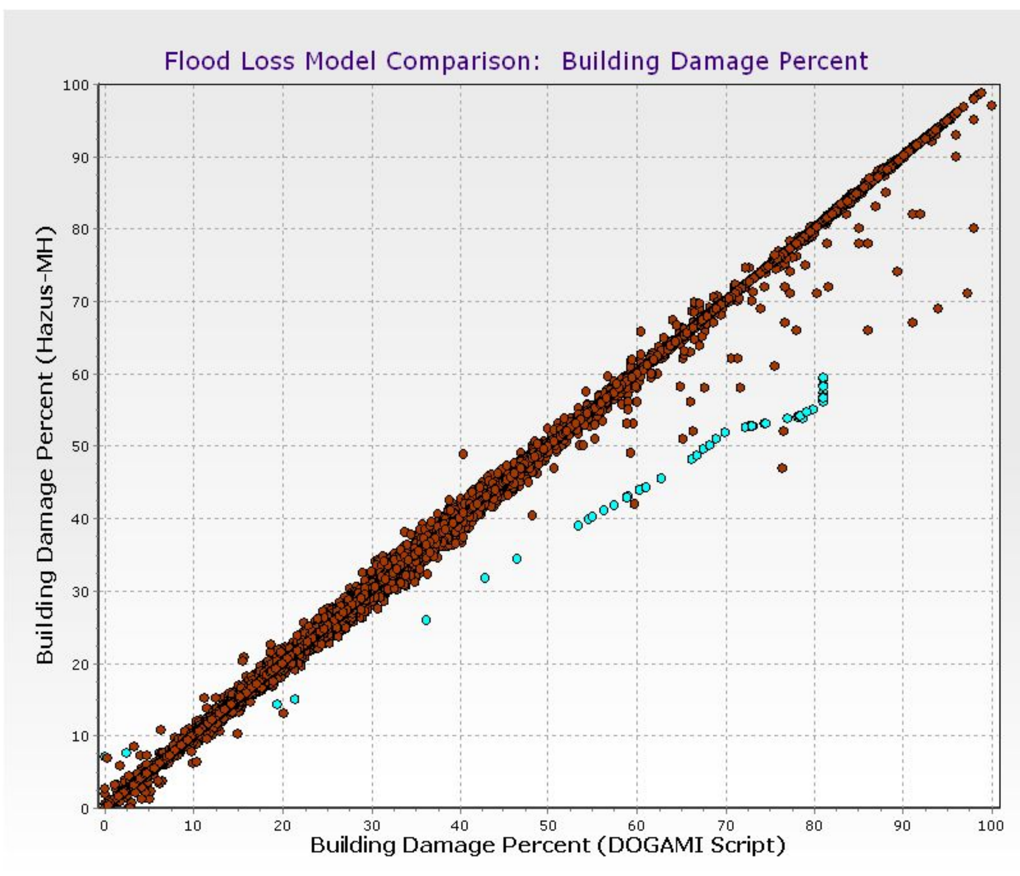
We noted no changes to the default Depth Damage Function library with the release of Hazus-MH v4.2 (tables `[dbo].[flBldgContDmgXRef]`, `[dbo].[flBldgInvDmgXRef]`, and `[dbo].[flBldgStrucDmgXRef]`). Thus what we report here for Hazus-MH v4.0 flood model should be the same as what would be obtained from a Hazus-MH v4.2 flood model.

Investigation and queries to Hazus-MH developers revealed the following:

- The Hazus-MH flood model (v4.0) uses an obsolete DDF for 'R11B' for building damage. It should use DDFID = 704. This is a legacy bug with the Hazus-MH flood model. The script performs the correct calculation.
- The Hazus-MH flood model rounds the First Floor Height to the nearest 0.5 foot, then does the depth-in-structure calculation. The script does no such rounding. The rounding does not introduce any particular bias to the loss estimation and is considered not a bug but a minor, inconsequential difference.
- The Hazus-MH flood model truncates the flood depth to 24 feet, then does the depth-in-structure calculation using First Floor Height. The script does the depth-in-structure calculation first, then caps the depth to 24 feet. We believe the script is more correct, though such cases are likely few in practice.

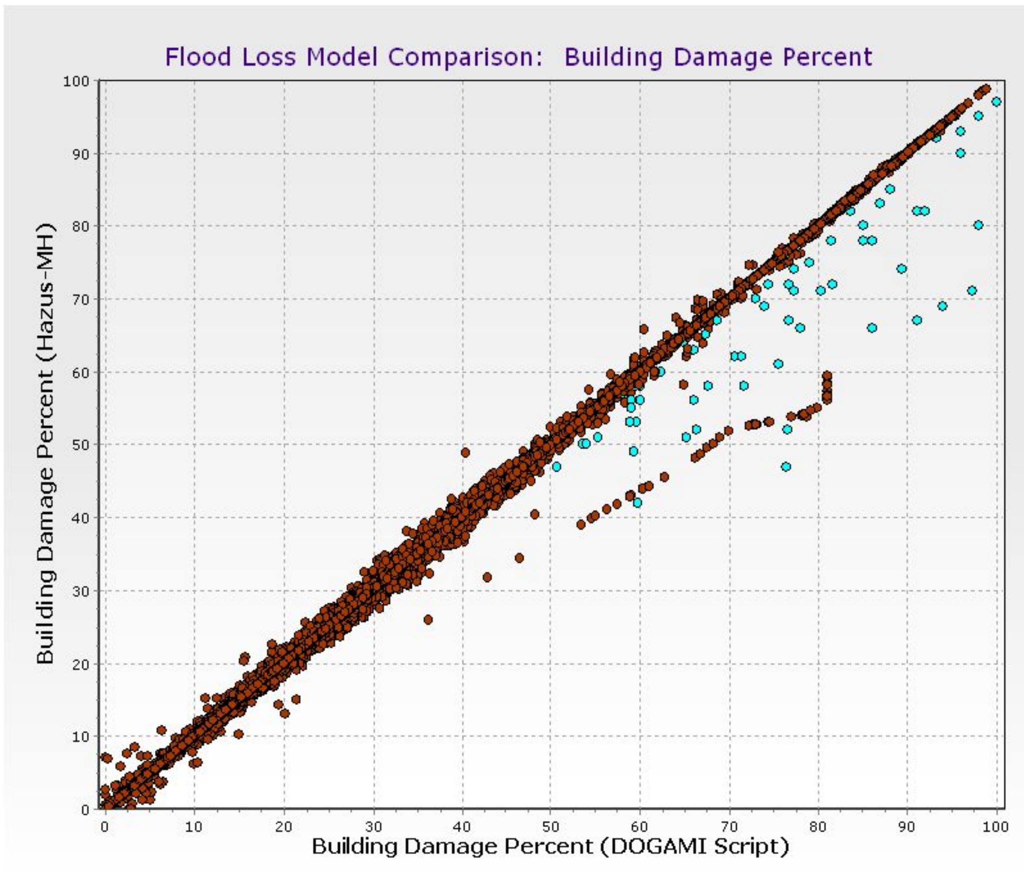
Graphically demonstrating the differences, we select the 'R11B' from the scatter plot data (**Figure 12-2**), and we see a simple systematic difference due to the usage of two different DDFs.

Figure 12-2. Comparison of Building Damage Percent calculation between the script and the Hazus-MH flood model. Single story RES1, no basement ('R11B') points are shown in cyan.



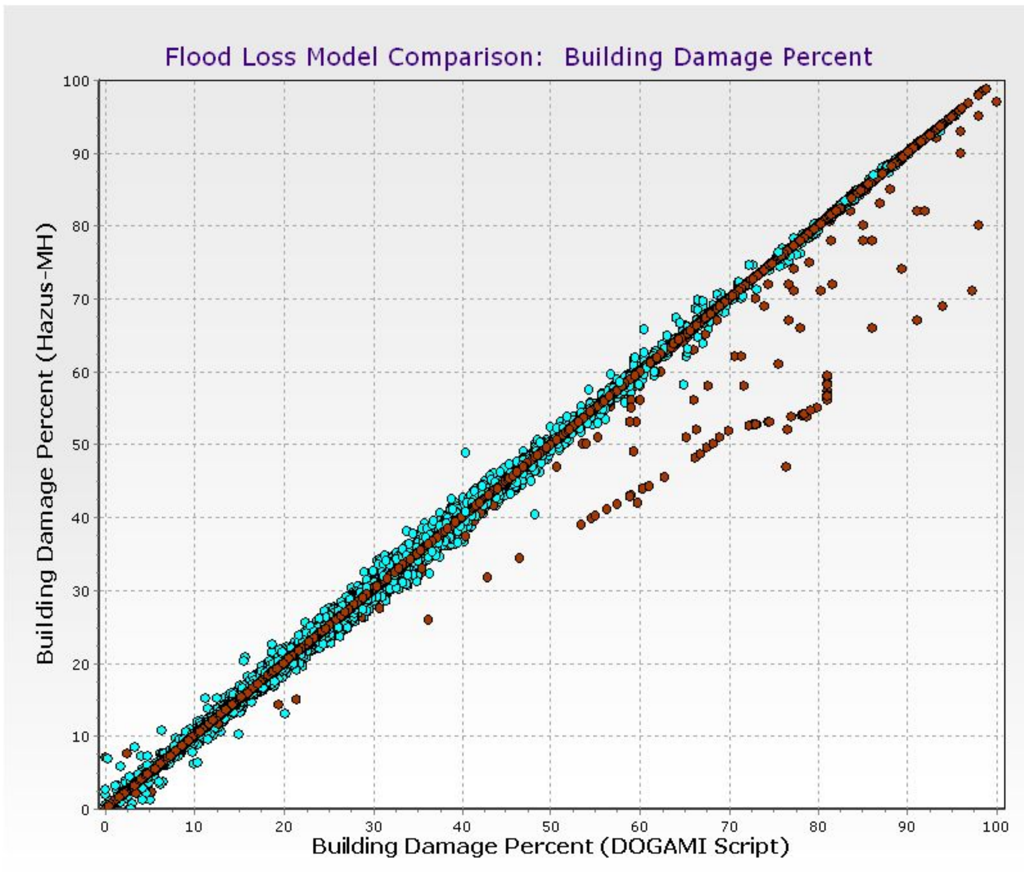
UDF points with depth grids in excess of 24 feet are shown in cyan in **Figure 12-3**. Note the Hazus-MH flood model underestimates the loss, due to the tool's rounding of the flood depth grid to 24 feet prior to calculating depth-in-structure.

Figure 12-3. Comparison of Building Damage Percent calculation between the script and the Hazus-MH flood model. UDF points with flood depth greater than 24 feet are shown in cyan.



Finally, we select UDFs with fractional values of First Floor Height (**Figure 12-4**). The Hazus-MH flood model, as mentioned previously, rounds to the nearest 0.5 foot, then performs the depth-in-structure calculation. The script does no such adjustment. The overall effect on damage loss estimation is considered minor and inconsequential.

Figure 12-4. Comparison of Building Damage Percent calculation between the script and the Hazus-MH flood model. UDF points with non-integer First Floor Height (e.g., 3.42 ft) are shown in cyan.



With all three variations taken into account, the results then compare exactly.

Similar results were observed with the Content loss estimates (**ContDmgPct**). We also validated that the script correctly calculates Content Cost (where not specified) as 0.5, 1.0, or 1.5 times the Building Cost, depending on the Occupancy Class.

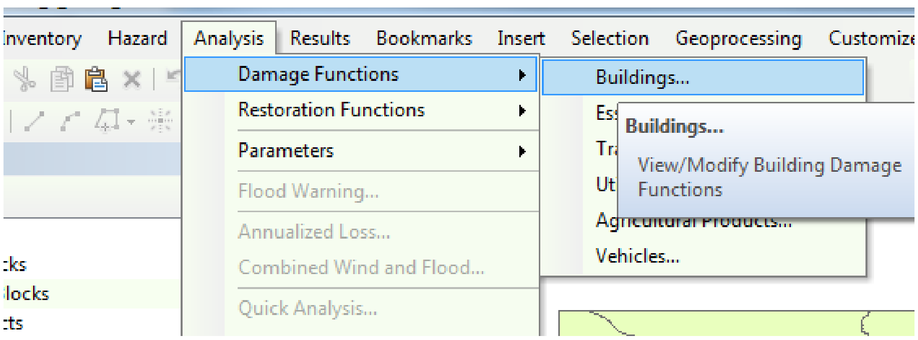
The Inventory loss is not comparable, given the incorrect method the Hazus-MH flood model v4.0 implements for computation (see the Known Differences with the Hazus-MH Flood Model section above).

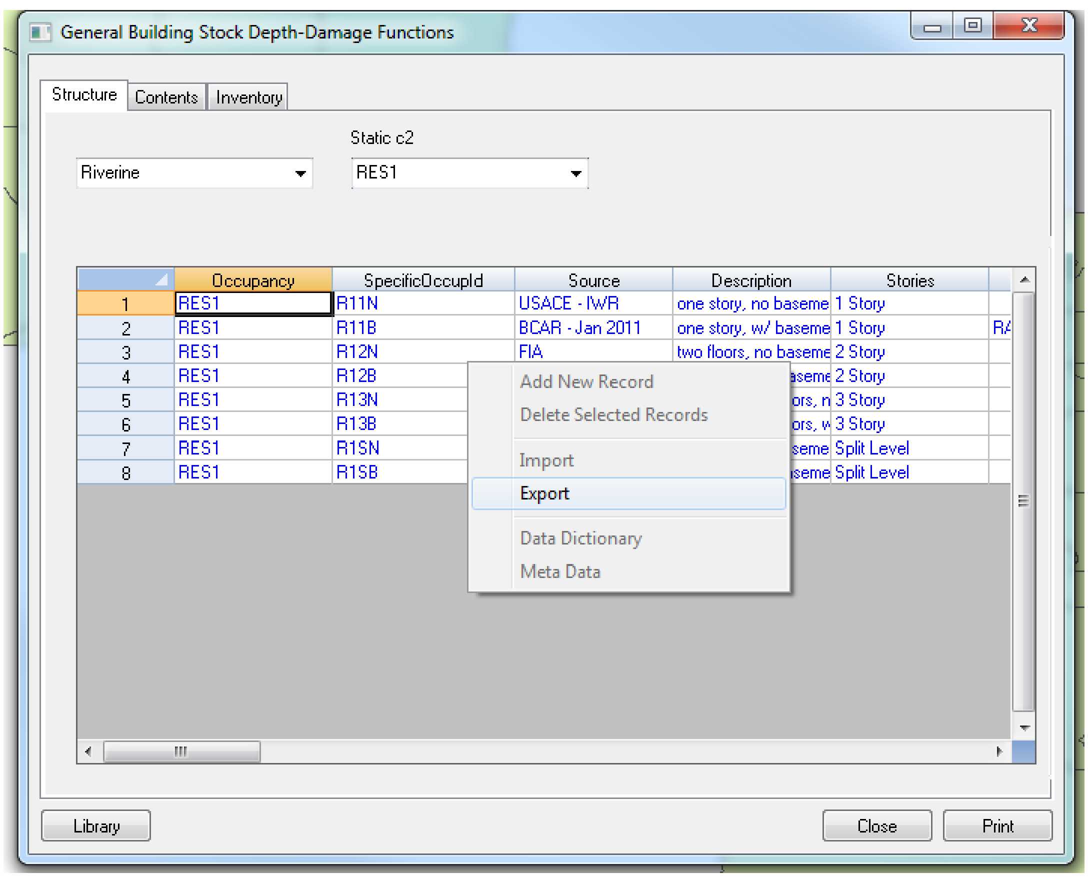
Debris and repair/restoration days were manually checked for a variety of Occupancy Classes and flood depths.

13.0 APPENDIX C: NOTES ON DEVELOPMENT

13.1 Building, Content, and Inventory Damage Functions

The CSV files used by the script were obtained by individually querying the damage function in Hazus-MH v4.0 (FEMA, 2017), and exporting each table (`Riverine` *and* `Coastal A` *and* `Coastal V`) as a TXT file (which is formatted as a CSV file). We noted no changes to the default Depth Damage Function library with the release of Hazus-MH v4.2 (tables `[dbo].[flBldgContDmgXRef]`, `[dbo].[flBldgInvDmgXRef]`, and `[dbo]. [flBldgStrucDmgXRef])`.





The CSV files for a particular category (e.g., 'Riverine', 'Building') were concatenated together, massaged with a text editor to remove redundant headers, and then imported into Microsoft® Excel®. In Excel, the headers were altered as follows:

FROM:	-4 ft	-3 ft	-2 ft	-1 ft	0 ft	1 ft	etc.,	up to	24 ft
TO:	m4	m3	m2	m1	p0	p1	etc.,	up to	p24

The "m" prefix stands for "minus," the "p" prefix for "plus." The script reads the adjusted flood depth and linearly interpolates between the specified values, using, as an example, the p3 and p4 as indices to access the DDF for a depth of 3.45 ft.

Once the headers were modified, the particular DDF collection was saved as a CSV file, which is then imported by the script. The `Hazus 4.0 Default DDF tables.xlsx` contains all default DDFs and also includes the full library of DDFs.

Close observation of the DDFs exported from the Hazus-MH flood model revealed that DDFs for Coastal AE and VE zones differ from Riverine DDFs only for certain residential structures. The script reverts to the Riverine default DDFs for all non-residential structures. DDFs for Coastal AE and V/VE zones differ slightly, thus we distinguish between the two. The unique DDF libraries for residential structures in Coastal AE and V/VE zones are named as follows in the lookup table library:

- `Building_DDF_CoastalA_LUT_Hazus4.0.csv`
- `Building_DDF_CoastalV_LUT_Hazus4.0.csv`
- `Content_DDF_CoastalA_LUT_Hazus4.0.csv`
- `Content_DDF_CoastalA_LUT_Hazus4.0.csv`

As of Hazus-MH v4.0, there are no unique inventory DDFs for buildings in coastal flooding zones. They are identical to the riverine DDFs.

The full set of DDFs is obtained by exporting the following SQL Server tables as CSV:

- `[dbo].[flBldgContDmgFn]`
- `[dbo].[flBldgInvDmgFn]`
- `[dbo].[flBldgStructDmgFn]`

As with the default DDFs, these tables were brought into Excel, headers were changed as mentioned above to a more useable, space-free, m/p header notation, and saved as separate CSV files.

To correctly calculate Inventory loss, the Inventory Economic Parameters table `[dbo].[flBldgEconParamSalesAndInv]` was exported from the Hazus 4.0 Microsoft SQL Server® database, and is stored in the LUT library as flBldgEconParamSalesAndInv.csv. No Consumer Price Index (CPI) adjustment was made to the values in the table.

13.2 Debris and Building Repair/Restoration Times

The Hazus SQL Server database tables `[dbo].[fldebris]` and `[dbo].[RsFnGBS]` were exported and manually adjusted to facilitate table lookup in the Python scripting context. In the lookup table library, we appended "_LUT" to the table name to note that the table is a modification of the original SQL Server tables. To create the debris table lookup key, we took a least-common-denominator approach. The key was constructed using the Occupancy Class name, presence of basement using the Foundation Type ('N' is no basement, 'B' is basement), slab-on-grade ('SG') or footing ('FT') foundation type, and a truncated depth-in-structure (less than 0 foot, 0 to 1 foot, 1 to 4 foot, 4 to 8 foot, 8 to 12 foot, and more than 12 foot). For example, RES4NBFT4 is a RES4, no basement, footing foundation, with depth of flooding between 1 and 4 foot. The key for accessing the restoration time table was more straightforward: Occupancy Class appended with a simplified depth-in-structure suffix.

We noted several inconsistencies the tabular information presented in the Hazus Technical Manual (FEMA, 2011, Table 11.1 and Table 14.12) and the data presented in the two SQL Server tables. For example, in the Hazus Technical Manual, COM6 has no debris estimate from flooding, and COM10 has significant debris, even with 0-foot depth-in-structure. In constructing the lookup tables for use by the script, we deferred to the data in the SQL Server tables. Table 14.12 (FEMA, 2011) distinguishes between a RES1 with and without a basement, but the SQL table has no such distinction. We included the debris estimate for AGR1 as is presented in the Hazus Flood Technical Manual and in the SQL Server table, but we note it appears unduly pessimistic.

For repair and restoration times, we present both the minimum and the maximum restoration time that were present in the SQL table, and we note that the minimum restoration time is not present in FEMA Table 14.12. For modest levels of flooding, a zero-days minimum restoration time may be true in that the building will need no repairs from flood damage, but we caution that the building may not be accessible and that the business operations may be suspended or occupation of a residential structure may be forbidden due to presence of surrounding floodwaters. Thus, using only the minimum restoration times to estimate economic impact and displaced population may result in an overly optimistic assessment.

13.3 Miscellaneous Notes

Tables exported from Hazus SQL Server database, or from a Hazus-MH flood model results file, on occasion have extraneous whitespaces present in key fields. For example, the Occupancy Class field may have an extra space or two suffixed to the standard name. This can complicate table lookup/pattern matching. We removed extraneous whitespace in all CSV files in the lookup table library. When processing a user-supplied UDF file, the script removes all whitespace characters, using the Python string `strip()`, from the *OccupancyClass*, *FoundationType*, and *flC* attributes.

14.0 APPENDIX D. USER-DEFINED FACILITIES FILE ATTRIBUTE SPECIFICATIONS

Table 14-1. DOGAMI flood risk script user-defined facility (UDF) input attributes. Script can generally accept mixed data types — for example, FirstFloorHt can be Integer type in the UDF input file, and FoundationType can be integer type.

Required	Input Attribute	Data Type	Range	Text Length	Notes
✓	UserDefinedFltyId	Text		8	Must be populated with a unique 8-character identifier, e.g., OR000123.
✓	OccupancyClass	Text		5	One of 33 Hazus-defined types, e.g., {RES1, RES2, COM3, IND4, AGR1, GOV2, REL1}. Script will fail if not specified, or if an unrecognized value is provided.
✓	Cost	Long	> 0		Replacement Cost of Structure, in US dollars. Records with '0' cost: the script will accept a zero value, but the record is essentially useless, as any estimated dollar damage to the structure will be 0. Consider correcting the UDF record or deleting it.
✓	NumStories	Short	≥ 1		Number of Stories. Used directly in Hazus-FL.
✓	FoundationType	Text	{1,2,3,4,5,6,7}	1	Foundation Type of the building. Text type, per Hazus-MH Flood Model convention.
✓	FirstFloorHt	Float	≥ 0.0		First Floor Height, in feet. Height can be specified in fractional feet.
✓	Area	Long	> 0		Total Area for the structure, in square feet. Used for Inventory Loss calculation when Inventory Cost is not supplied.
	flC	Text	{V,VE,CAE}	3	**fl**ood **C**oastal. Identifies particular UDFs in a coastal flooding zone. If **BldgDamageFnID**, **ContDamageFnId** are not provided or populated, the script will use coastal flooding DDFs instead of Riverine DDFs. V, VE, are Coastal V zones. CAE is Coastal AE Zone. Definition of Coastal AE Zone is complex; consult your coastal flooding expert. Only V, VE, and CAE are recognized by the script. For all other values, Riverine DDFs will be used. **Note: Hazus-MH flood model has no equivalent.**
	ContentCost	Long	≥ 0		Content Cost. If attribute is supplied, DOGAMI 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.
	InvCost	Long	≥ 0		Inventory Cost. If user has better information than what Hazus estimates based on Occupancy Class and Square Foot. **Note: Hazus-MH flood model has no equivalent.**
	BldgDamageFnID	Text	'100'–'704'	3+	User-supplied Building Depth Damage Function ID. Used by Hazus-MH flood model and DOGAMI script to override the standard Damage Depth Functions for buildings. Text type, per Hazus convention. This is an optional attribute: if attribute exists, and the record is populated with a legitimate value, the script will use it, else the script will use the standard (default) DDF for the given Occupancy Class/Number of Stories/FoundationType. The 'ID' capitalization is per the Hazus naming convention. If the attribute is supplied, not all records need to be populated; supply with `NoData` or `""` (blank) where there is no need to override the standard (default) DDF assignment.
	ContDamageFnId	Text	'21'–'535'	3+	User-supplied Content Depth Damage Function ID. Used by Hazus-MH flood model and DOGAMI script to override the standard Damage Depth Functions for building content. Text type, per Hazus convention. This is an optional attribute: if attribute exists, and the record is populated with a legitimate value, the script will use it, else the script will use the standard (default) DDF for the given Occupancy Class/Number of Stories/FoundationType. If the attribute is supplied, not all records need to be populated; supply with `NoData` or `""` (blank) where there is no need to override the standard (default) DDF assignment.
	InvDamageFnId	Text	'1'–'116'	3+	User-supplied Inventory Depth Damage Function ID. Used by Hazus-MH flood model and DOGAMI script to override the standard Damage Depth Functions for building inventory. Text type, per Hazus convention. This is an optional attribute: if attribute exists, and the record is populated with a legitimate value, the script will use it, else the script will use the standard (default) DDF for the given Occupancy Class/Number of Stories/FoundationType. If the attribute is supplied, not all records need to be populated; supply with `NoData` or `""` (blank) where there is no need to override the standard (default) DDF assignment.

+ Hazus DDF libraries are one to three characters. Users can supply their own DDF IDs with more characters, if desired.

Table 14-2. DOGAMI flood risk script output attributes. The attributes are added to a copy of the user-defined facility (UDF) input file. Attributes marked as *Hazus-MH Flood Model Equivalent* have an equivalent attribute in a Hazus-MH flood UDF output file.

Output Attribute	Data Type	Range or Length	Hazus-MH Flood Model Equivalent	Notes
Depth_Grid	Float	≥ 0.0		Sampled Flood Depth Grid, in feet. Depending on the grid format, the value may be `-9999` or `NoData` for points outside of the flood depth grid.
Depth_in_Struc	Float	≥ 0.0		Depth-in-Structure Adjusted flood depth grid at the UDF point, in feet. Simple calculation: If the *Depth_Grid* is a `NoData` or `-9999` value, value is `-9999`. Else value is *Depth_Grid* − *FirstFloorHt*
flExp	Short	{0,1}		UDF is exposed to a flood. Simply 0 or 1. If the UDF is in a flood depth grid, then the value is 1, regardless of depth-in-structure.
SOID	Text	5		**S**pecific **O**ccupancy **ID**. The Hazus-MH Flood shorthand that compresses OccupancyClass, NumStories, and FoundationType into a concise 4- to 5-character code, e.g. R11N for a RES1, no basement, single story. Used to access the look-up tables where the user does not specify a DDF. XXXX for buildings not in the flood zone.
BDDF_ID	Text	3		**B**uilding **D**epth **D**amage **F**unction **ID**. If user specified a building DDF, and the value was legitimate, that value will be duplicated here. Otherwise, for UDFs in the flood zone, the value is the specific record used in the standard (default) DDF lookup table (Hazus-MH Flood does not provide this information in its results). If the UDF is not in the flood zone, the value is set to 0.
BldgDmgPct	Float	0 – 100	✓	Building Damage Percentage. Interpolated from the lookup tables, depending on flood depth. Value ranges between 0 and 100. For UDFs outside the flood zone, the value is set to 0.0.
BldgLossUSD	Long	≥ 0	✓	Loss, in US dollars, to the building. Formula: *Cost* × *BldgDmgPct*
ContentCostUSD	Long	≥ 0		Building Content Cost, in US dollars. If user supplied a *ContentCost* attribute, and the record's value is non-null, the value is *ContentCost*. Otherwise, depending on OccupancyClass, it is calculated at 0.5, 1.0, or 1.5 times the user-supplied building *Cost*. See Hazus-MH Flood Technical Manual (FEMA, 2011).
CDDF_ID	Text	3		**C**ontent **D**epth **D**amage **F**unction **ID**. If user specified a Content DDF, and the value was legitimate, that value will be duplicated here. Otherwise, for UDFs in the flood zone, the value is the specific record used in the DDF lookup table (note that Hazus-MH Flood does not provide this value). If the UDF is not in the flood zone, the value is set to 0.
ContDmgPct	Float	0 – 100	✓	Building Content Damage Percentage. Interpolated from the lookup tables, depending on flood depth. Value ranges between 0 and 100. For UDFs outside the flood zone, the value is set to 0.0.
ContentLossUSD	Long	≥ 0	✓	Loss, in US dollars, to the Building Content. Formula: *ContDmgPct* × *ContentCostUSD*
InventoryCostUSD	Long	≥ 0		Building Inventory Valuation, in US dollars. If the user supplied an *InvCost* attribute, and *InvCost* is greater than zero, the value is *InvCost*. Otherwise, depending on OccupancyClass, the Inventory Cost is calculated based on type of business and the building's square footage.
IDDF_ID	Text	3		**I**nventory **D**epth **D**amage **F**unction **ID**. If user specified a DDF, and the value was legitimate, that value will be duplicated here. Otherwise, for UDFs in the flood zone, the value is the specific record used in the DDF lookup table (note that Hazus-MH Flood does not provide this value). If the UDF is not in the flood zone, the value is set to 0.
InvDmgPct	Float	0 – 100		Building Inventory Damage Percentage. Interpolated from the lookup tables, depending on flood depth. Value ranges between 0 and 100. For UDFs outside the flood zone, the value is set to 0.0. Note that only certain types of OccupancyClass have a standard Inventory Loss function defined.
InventoryLossUSD	Long	≥ 0	✓	Loss, in US dollars, to the Inventory Content. If user supplied an inventory cost attribute, the value is *InvDmgPct* × *InventoryCostUSD*. (Note the significant discrepancy between the computed values and Hazus 4.0 loss estimates. Hazus 4.0 does not correctly implement the Inventory Loss calculation at the UDF level.)
Debris_Tot	Long	≥ 0		Total debris, in tons. Combines Finish, Structure, and Foundation debris estimates. Based on Occupancy Class, Square Footage, Foundation Type, and Depth-in-Structure.
Restor_Days_Min	Short	≥ 0		Restoration time, in days — Minimum bound. Note there is no direct Hazus equivalent. The name is identical to what is in the Hazus lookup table. Note that the restoration times assume, like the debris, that a 'substantially damaged' structure (one which experiences > 50% loss) is torn down and replaced.
Restor_Days_Max	Short	≥ 0		Restoration time, in days — Maximum bound. Note there is no direct Hazus equivalent. The name is identical to what is in the Hazus lookup table. Note that the restoration times assume, like the debris, that a 'substantially damaged' structure (one which experiences > 50% loss) is torn down and replaced.
GridName	Text	50		Name of flood depth grid. This may seem redundant, given the output file naming convention, but when concatenating multiple results files together (e.g., when doing a sensitivity test with multiple flood depth grids), the attribute is helpful for sorting and summarization purposes.