StreamStats Data Preparation Tools 4.0.0 documentation

[StreamStats Data Preparation Tools](#)

4.0.0

Contents:

* [StreamStats\_DataPrep ESRI Toolbox](index.html#document-StreamStats_DataPrep)
  + [Database Setup](index.html#database-setup)
  + [Make Elevation Data Index](index.html#make-elevation-data-index)
  + [Extract Polygons](index.html#extract-polygons)
  + [Check For NoData Cells](index.html#check-for-nodata-cells)
  + [Fill NoData Cells](index.html#fill-nodata-cells)
  + [Project and Scale](index.html#project-and-scale)
  + [TopoGrid (optional)](index.html#topogrid-optional)
  + [Bathymetic Gradient Setup](index.html#bathymetic-gradient-setup)
  + [Coastal DEM Processing (optional)](index.html#coastal-dem-processing-optional)
  + [Hydro-Enforce DEM](index.html#hydro-enforce-dem)
  + [Adjust Accumulation](index.html#adjust-accumulation)
  + [Adjust Accumulation Simple](index.html#adjust-accumulation-simple)
  + [Post Hydrodem](index.html#post-hydrodem)
* [StreamStats DataPrep Library](index.html#document-modules)
  + [databaseSetup Module](index.html#document-databaseSetup)
  + [elevationTools Module](index.html#document-elevationTools)
  + [make\_hydrodem Module](index.html#document-make_hydrodem)
  + [topo\_grid Module](index.html#document-topo_grid)
* [Example Scripts](index.html#document-example_scripts)
  + [Initial Database Setup](index.html#document-ex_databaseSetup)
  + [Make Digital Elevation Model Index](index.html#document-ex_makeELEVDATAindex)
  + [Extract Watershed Polygon](index.html#document-ex_extractPoly)
  + [Fill NoData Values](index.html#document-ex_FillNoData)
  + [Project and Scale a Digital Elevation Model](index.html#document-ex_ProjScale)
  + [Check Inner and Outer Walls](index.html#document-ex_check_walls)
  + [Check For NoData Cells](index.html#document-ex_CheckNoData)
  + [Topogrid (optional)](index.html#document-ex_topogrid)
  + [Bathymetric Gradient](index.html#document-ex_bathymetricGradient)
  + [Hydro-Enforce a Digital Elevation Model](index.html#document-ex_hydroDEM)
  + [Adjust Accumulation Grid](index.html#document-ex_adjust_accum)
  + [Adjust Accumulation Grid (Simple)](index.html#document-ex_adjust_accum_simple)
  + [Post Hydro-Enforcement](index.html#document-ex_post_hydrodem)

[StreamStats Data Preparation Tools](#)

* [Docs](#) »
* StreamStats Data Preparation Tools 4.0.0 documentation

# Welcome to StreamStats Data Preparation Tools Documentation![¶](#welcome-to-streamstats-data-preparation-tools-documentation)

A Python package to pre-process and hydro-enforce digital elevation models using hydrography features for use in the U.S. Geological Survey (USGS) StreamStats project.

Version 3.10 tools can be accessed [here](https://code.usgs.gov/StreamStats/datapreptools/-/archive/v3.10/datapreptools-v3.10.zip). See below for installation instructions for Version 4 tools.

## About[¶](#about)

The StreamStats Data Preparation Tools aid in the processing of digital elevation models (DEMs) and hydrography data for ingestion into the USGS StreamStats project and web-application. The tools and associated work flow examples can be used to prepare DEM and hydrography subsets for local StreamStats folders, prepare those data for use in hydro-enforcement, hydro-enforce the digital elevation model, and process the resulting flow accumulation and flow direction grids for use in the ESRI ArcHydro data model.

## Citation[¶](#citation)

Please cite these tools and documentation as:

Barnhart, T.B., Smith, M., Rea, A., Kolb, K., Steeves, P., and McCarthy, P., 2020, StreamStats Data Preparation Tools, version 4: USGS Software Release, <https://doi.org/10.5066/P9UM2NUL>.

## Dependencies[¶](#dependencies)

This toolbox has few dependencies; however, it must either be started through ESRI ArcMap 10.6.1 or ESRI ArcPro 2.5.1 or via a Python 2 or 3 executable that is aware of ESRI ArcPy. These tools can be run either via their ESRI ArcToolbox wrappers or as functions via Python scripts to facilitate processing of larger domains.

The post-hydrodem() function requires ESRI ArcHydro. Please use version 10.6.0.51 of ESRI ArcHydro 64-bit, available here: <http://downloads.esri.com/archydro/ArcHydro/Setup/10.6/>

## Structure[¶](#structure)

This package is contained in several Python files, which will need to be installed correctly to function (see below).

* root
  + **documentation:** Documentation of the tools and functions. Open documentation/build/html/index.html to access.
  + **StreamStats\_DataPrep.pyt:** ESRI ArcPy Toolbox wrapper file.
  + **StreamStats\_DataPrep.py:** ESRI ArcPy Toolbox definition file.
  + **databaseSetup.py:** Python module for setting up the local folders for a processing domain.
  + **elevationTools.py:** Python module for inspecting DEMs, reprojection, and scaling values to integers.
  + **make\_hydrodem.py:** Python module for DEM hydro-enforcement.
  + **\*.xml:** ESRI ArcPy Toolbox documentation files.
  + **examples:** Folder of Python script examples of work flows.
  + **source:** Folder containing documentation source files.

## Installation[¶](#installation)

Clone these tools onto your machine using the git clone commands. Or download the repository using the link in the upper right of the repository page here: <https://code.usgs.gov/StreamStats/datapreptools>

Once downloaded, the data preparation ESRI ArcGIS toolbox can be accessed from the ArcCatalog pane in ESRI ArcMap or navigated to in ESRI ArcPro. The toolbox is compatible with both ESRI ArcMap (i.e. Python 2) and ESRI ArcPro (i.e. Python 3), except for the final processing step, Post Hydrodem, which relies on ESRI ArcHydro and only works with ESRI ArcMap (Python 2) at this time (2020) due to ESRI ArcHydro compatibility issues with ESRI ArcPro (i.e. Python 3).

The ESRI ArcGIS toolbox is built from a set of Python libraries that can be called from the command line or a scripting environment to facilitate processing large volumes of data. Please refer to the documentation of the [StreamStats DataPrep Library](index.html#modules-label) and [Example Scripts](index.html#examples-label) for information and examples on the usage of the tools on the command line. The tools run fastest via ESRI ArcPro or Python 3 (see caveat above), but can still be used with ESRI ArcMap and Python 2.

## Documentation[¶](#documentation)

Tool and function library documentation can be found by opening ./documentation/build/html/index.html with a web browser after the tools have been cloned/downloaded to your local machine.

## Reporting Issues and Problems with the Tools[¶](#reporting-issues-and-problems-with-the-tools)

Please log problems with the tools or function libraries in the issues portion of this repository. **Please do not email me.** Logging problems in this way allows other users to see the discussion and, hopefully, the solution to problems. Please be sure to check out the repository documentation as well before submitting an issue.

## Known Issues[¶](#known-issues)

* **Networked storage drives:** When working on ESRI ArcGIS projects stored on network Attached Storage (NAS) devices, additional configuration may be required. See:
  + <https://community.spiceworks.com/topic/1389064-performance-and-locking-issues-with-synology-nas-and-arcgis>
  + <https://support.esri.com/en/technical-article/000012722>

## Acknowledgments[¶](#acknowledgments)

The authors thank Moon Kim (USGS) for his comments on an early version of this code.

## Disclaimers[¶](#disclaimers)

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

This software has been approved for release by the U.S. Geological Survey (USGS). Although the software has been subjected to rigorous review, the USGS reserves the right to update the software as needed pursuant to further analysis and review. No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. Furthermore, the software is released on condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from its authorized or unauthorized use.

## License[¶](#license)

This work is published with the MIT Open Source License:

Copyright 2020 Theodore Barnhart, Martyn Smith, Alan Rea, Katherine Kolb, Peter Steeves, and Peter McCarthy, U.S. Geological Survey.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### StreamStats\_DataPrep ESRI Toolbox[¶](#streamstats-dataprep-esri-toolbox)

#### Database Setup[¶](#database-setup)

*class* StreamStats\_DataPrep.databaseSetup[¶](#StreamStats_DataPrep.databaseSetup)

Set up the workspaces needed to process elevation and hydrography data.

This tool is a wrapper on [databaseSetup.databaseSetup()](index.html#databaseSetup.databaseSetup).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.databaseSetup.getParameterInfo)(self) | Database Setup inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.databaseSetup.getParameterInfo)

Database Setup inputs.

|  |  |
| --- | --- |
| Parameters: | **Output Workspace**DEWorkspace (File System)  Folder-type workspace for local folders and geodatabase to be created.  **Main ArcHydro Geodatabase Name**GPString  Name of the geodatabase to be created in “Output Workspace.”  **Hydrologic Unit Boundary Dataset**DEShapefile or DEFeatureClass  Polygon vector defining local processing units. Should have columns for outwalls and inwalls, see below.  **Outwall Field**Field  Field in “Hydrologic Unit Boundary Dataset” used to determine local folders and outwalls.  **Inwall Field**Field  Field in “Hydrologic Unit Boundary Dataset” used to determine inwalls.  **Hydrologic Unit Buffer Distance (m)**GPString  Distance to buffer local folder polygons.  **Input Hydrography Workspace**DEWorkspace  Path to folder type workspace with National Hydrography Dataset geodatabases.  **Elevation Dataset Template**DERasterBand  Raster dataset to pull projection information from, works best as an ESRI grid.  **Alternative Outwall Buffer**GPString (optional)  Distance for alternative outwall buffer. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Make Elevation Data Index[¶](#make-elevation-data-index)

*class* StreamStats\_DataPrep.makeELEVDATAIndex[¶](#StreamStats_DataPrep.makeELEVDATAIndex)

Create a seamless raster mosaic dataset from input digital elevation tiles.

This tool is a wrapper on [elevationTools.elevIndex()](index.html#elevationTools.elevIndex).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.makeELEVDATAIndex.getParameterInfo)(self) | Make ELEV data index inputs |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.makeELEVDATAIndex.getParameterInfo)

Make ELEV data index inputs

|  |  |
| --- | --- |
| Parameters: | **Output Geodatabase**DEWorkspace (Geodatabase)  Path to the geodatabase that will hold the output raster mosaic dataset.  **Output Raster Mosaic Dataset Name**GPString  Name of raster mosaic dataset (RMD) to output, defaults to IndexRMD.  **Coordinate System**GPCoordinateSystem  Coordinate system of input grids and raster mosaic dataset.  **Input Elevation Data workspace**DEWorkspace (Folder)  Path to folder holding input digital elevation models to be included in the raster mosaic dataset. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Extract Polygons[¶](#extract-polygons)

*class* StreamStats\_DataPrep.ExtractPoly[¶](#StreamStats_DataPrep.ExtractPoly)

Extract a hydrologic unit from a digital elevation model based on a clipping polygon.

This tool is a wrapper on [elevationTools.extractPoly()](index.html#elevationTools.extractPoly).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.ExtractPoly.getParameterInfo)(self) | Extract Polygon inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.ExtractPoly.getParameterInfo)

Extract Polygon inputs.

|  |  |
| --- | --- |
| Parameters: | **Output Workspace**DEWorkspace (Folder)  Path to folder to work in.  **ELEVDATA Raster Mosaic Dataset**DEMosaicDataset  Path to the raster mosaic dataset holding the elevation data.  **Clip Polygon**GPFeatureLayer  Feature class of the watershed boundary being used for clipping.  **Output Grid**GPString  Name of the output ESRI grid, defaults to dem\_dd. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Check For NoData Cells[¶](#check-for-nodata-cells)

*class* StreamStats\_DataPrep.CheckNoData[¶](#StreamStats_DataPrep.CheckNoData)

Check for no data cells in a digital elevation model.

This tool is a wrapper on [elevationTools.checkNoData()](index.html#elevationTools.checkNoData).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.CheckNoData.getParameterInfo)(self) | Check for no data inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.CheckNoData.getParameterInfo)

Check for no data inputs.

|  |  |
| --- | --- |
| Parameters: | **InputGrid**DERasterBand  Path to raster dataset to examine.  **Workspace**DEWorkspace (Geodatabase)  Geodatabase-type workspace.  **Output Feature Layer**GPString  Name of output feature class, defaults to DEM\_NoDataSinks. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Fill NoData Cells[¶](#fill-nodata-cells)

*class* StreamStats\_DataPrep.FillNoData[¶](#StreamStats_DataPrep.FillNoData)

Fill no data cells in a digital elevation model.

This tool is a wrapper on [elevationTools.fillNoData()](index.html#elevationTools.fillNoData).

Notes

This tool can be run iteratively to fully fill no data areas that are larger than one cell.

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.FillNoData.getParameterInfo)(self) | Fill no data inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.FillNoData.getParameterInfo)

Fill no data inputs.

|  |  |
| --- | --- |
| Parameters: | **Workspace**DEWorkspace (Folder)  Path to workspace folder.  **Input Grid**DERasterBand  Path to raster dataset with no data values to be filled, defaults to DEM\_NoDataSinks.  **Output Grid**GPString  Path to write out filled raster dataset to, defaults to DEM\_filled. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Project and Scale[¶](#project-and-scale)

*class* StreamStats\_DataPrep.ProjScale[¶](#StreamStats_DataPrep.ProjScale)

Project and scale a digital elevation model.

This tool is a wrapper on [elevationTools.projScale()](index.html#elevationTools.projScale).

Notes

After scaling, this tool attempts to set the correct z-units; however, if your vertical units are different from your horizontal units it is advised to check the z-units manually.

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.ProjScale.getParameterInfo)(self) | Project and scale digital elevation model inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.ProjScale.getParameterInfo)

Project and scale digital elevation model inputs.

|  |  |
| --- | --- |
| Parameters: | **InWorkSpace**DEWorkspace (Folder)  Path to the workspace folder.  **InGrid**DERasterBand  Path to the raster dataset to project and scale.  **OutGrid**GPString  Name for the projected and scaled raster, defaults to dem\_raw.  **OutCoordSys**GPCoordinateSystem  Coordinate system with which to project or preproject the input raster.  **OutCellSize**analysis\_cell\_size  Output cell size to project the input raster to, defaults to 10 horizontal map units.  **RegPt**GPString  Registration point for the projected raster, defaults to “0 0”.  **scaleFact**GPString  Scale factor to use to convert the projected raster to integers, defaults to 100. Consider using a larger scale factor as cell-size decreases. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### TopoGrid (optional)[¶](#topogrid-optional)

*class* StreamStats\_DataPrep.TopoGrid[¶](#StreamStats_DataPrep.TopoGrid)

Condition an input DEM using a flowline dendrite prior to hydro-enforcement.

This tool is a wrapper on [topo\_grid.topogrid()](index.html#topo_grid.topogrid).

Notes

This function turns the input DEM into a 3D point cloud, thinned using the VIP algorithm so that not all points are retained from the original DEM. The point cloud is used in conjunction with the supplied flowlines to re-interpolate a DEM that is aware of the location of the flowlines and their flow direction.

This is a computationally intensive function. Running it via ESRI ArcPro or Python 3 will be faster than using ESRI ArcMap or Python 2.

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.TopoGrid.getParameterInfo)(self) | TopoGrid inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.TopoGrid.getParameterInfo)

TopoGrid inputs.

|  |  |
| --- | --- |
| Parameters: | **Output Workspace**DEWorkspace (Geodatabase)  Path to a geodatabase workspace.  **Dissolved HUC8 boundary**DEFeatureClass or DEShapefile  Feature class to use in bounding the topogrid conditioning process.  **Topogrid Buffer Distance**GPDouble  Distance to buffer the input HUC8 boundary, in the units of the HUC8 boundary.  **12 Digit Hydrologic Unit Datasets if dissolved HUC8 boundary failed**DEFeatureClass or DEShapefile (Optional)  List of HUC12 boundaries to split up TopoGrid computations if the whole domain fails.  **Dendritic Flowline Features**DEFeatureClass or DEShapefile  Dendrite used for enforcing flow direction in topogrid.  **Buffered and Projected Elevation Data**DERasterBand or DERasterDataset  Input digital elevation model to be conditioned using topogrid.  **Output Cell Size**GPString  Cell size for output digital elevation model, defualts to 10 horizontal map units.  **VIP Percentage**GPString  Thinning value used in the Very Important Points (VIP) algorithm to decide how many points from the original raster are retained, defaults to 5 percent.  **SnapGrid**DERasterBand (Optional)  Raster to snap output grid to. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Bathymetic Gradient Setup[¶](#bathymetic-gradient-setup)

*class* StreamStats\_DataPrep.SetupBathyGrad[¶](#StreamStats_DataPrep.SetupBathyGrad)

Prepare bathymetric gradient inputs for use in hydro-enforcement.

This tool is a wrapper on [make\_hydrodem.bathymetricGradient()](index.html#make_hydrodem.bathymetricGradient).

Notes

The bathymetric gradient refers to generating a sloping area around the flowline dendrite that ensures the landscape around the dendrite flows to the stream. This also adds a sloping surface to double-line streams and waterbodies to help insure proper drainage after hydro-enforcement.

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.SetupBathyGrad.getParameterInfo)(self) | Setup Bathymetric Gradient inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.SetupBathyGrad.getParameterInfo)

Setup Bathymetric Gradient inputs.

|  |  |
| --- | --- |
| Parameters: | **Output Workspace**DEWorkspace (Geodatabase)  Path to a geodatabase workspace.  **Digital Elevation Model (used for snapping)**DERasterBand  Path to a digital elevation model to use for aligning output grids to the rest of the project.  **Dissolved Eight-Digit Hydrologic Unit Code (HUC8) Dataset**DEFeatureClass  Feature class of the local folder boundary.  **NHD Area**DEFeatureClass  Feature class of NHD double line streams.  **NHD Dendrite**DEFeatureClass  Feature class of the flowline dendrite.  **NHD Waterbody**DEFeatureClass  Feature class of the NHD water bodies.  **Cell Size**GPString  Output grid cell size, defaults to 10 horizontal map units. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

Notes

This tool expects that the NHD Dendrite and NHD Area features have an attribute column with the name “FType” populated with feature type codes. In the newer NHD High-Resolution data-sets this attribute is called “FTYPE.” The query used to select features is case sensitive; as such, this attribute needs to be renamed to “FType” for NHD High-Resolution data.

#### Coastal DEM Processing (optional)[¶](#coastal-dem-processing-optional)

*class* StreamStats\_DataPrep.CoastalDEM[¶](#StreamStats_DataPrep.CoastalDEM)

Prepare coastal areas for hydro-enforcement.

This tool is a wrapper on [make\_hydrodem.coastaldem()](index.html#make_hydrodem.coastaldem).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.CoastalDEM.getParameterInfo)(self) | Coastal digital elevation model processing inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.CoastalDEM.getParameterInfo)

Coastal digital elevation model processing inputs.

|  |  |
| --- | --- |
| Parameters: | **Workspace**DEWorkspace (Folder)  Path to a folder-type workspace.  **Input raw DEM**DERasterBand  Original digital elevation model to be corrected for coastal areas, defaults to dem\_raw.  **Input LandSea polygon feature class**DEFeatureClass  Feature class indicating areas of land and sea.  **Output DEM**DERasterBand  Output digital elevation model name, defaults to dem\_sea.  **Sea Level**GPString  Value to insert into areas identified as the sea, defaults to -60000 vertical map units. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Hydro-Enforce DEM[¶](#hydro-enforce-dem)

*class* StreamStats\_DataPrep.HydroDEM[¶](#StreamStats_DataPrep.HydroDEM)

Hydro-Enforce a DEM.

This tool is a wrapper on [make\_hydrodem.hydrodem()](index.html#make_hydrodem.hydrodem).

Notes

AGREE defaults should not be changed as this can lead to alignment issues between the flowlines and the resultant hydro-enforced DEM and subsequent products (FDR and FAC).

Methods

|  |  |  |  |
| --- | --- | --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.HydroDEM.getParameterInfo)(self) | |  |  | | --- | --- | | Parameters: |  | |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.HydroDEM.getParameterInfo)

|  |  |
| --- | --- |
| Parameters: | **Output Workspace**DEWorkspace (geodatabase)  Geodatabase-type workspace where output raster will be saved.  **Scratch Workspace**DEWorkspace (folder)  Folder-type scratch workspace.  **HUC layer**DEFeatureClass  Polygon feature class the bounds the local folder you are working in.  **Digital Elevation Model**DERasterBand  Digital elevation model to be hydro-enforced.  **Stream Dendrite**DEFeatureClass  Polyline feature class describing where streams are on the landscape.  **Snap Grid**DERasterBand  Raster grid used to align output DEM with other related grids or adjacent local folders.  **NHD Waterbody Grid**DERasterDataset (optional)  Grid representing waterbodies from the bathymetric gradient step.  **NHD Flowline Grid**DERasterDataset (optional)  Grid representing flowlines from the bathymetric gradient step.  **Inner Walls**DEFeatureClass (optional)  Polyline feature class used to inforce internal drainage.  **Cell Size**GPString  Output cell size, defaults to 10 horizontal map units.  **Drain Plugs**DEFeatureClass (optional)  Point feature class representing sink locations.  **HUC buffer**GPDouble (optional)  Distance to buffer the HUC layer, defaults to 50 horizontal map units.  **Inner Wall Buffer**GPDouble (optional)  Distance to buffer the inwall, defaults to 15 horizontal map units.  **Inner Wall Height**GPDouble (optional)  Inwall height, defaults to 150000 vertical map units.  **Outer Wall Height**GPDouble (optional)  Outer wall height, defaults to 300000 vertical map units.  **AGREE buffer**GPDouble (optional)  Defaults to 60 horizontal map units.  **AGREE Smooth Drop**GPDouble (optional)  Defaults to -500 vertical map units.  **AGREE Sharp Drop**GPDouble (optional)  Defaults to -50000 vertical map units.  **Bowl Depth**GPDouble (optional)  Defaults to 2000 vertical map units. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Adjust Accumulation[¶](#adjust-accumulation)

*class* StreamStats\_DataPrep.AdjustAccum[¶](#StreamStats_DataPrep.AdjustAccum)

Adjust flow accumulation grids following hydro-enforcement.

This tool is a wrapper on [make\_hydrodem.adjust\_accum()](index.html#make_hydrodem.adjust_accum).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.AdjustAccum.getParameterInfo)(self) | Adjust flow accumulation grid inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.AdjustAccum.getParameterInfo)

Adjust flow accumulation grid inputs.

|  |  |
| --- | --- |
| Parameters: | **Downstream Accumulation Grid**DERasterDataset  Downstream raster dataset to correct.  **Downstream Flow Direction Grid**DERasterDataset  Downstream flow direction grid.  **Upstream Flow Accumulation Grids**DERasterDataset  Upstream flow accumulation grids to correct the downstream grid with.  **Upstream Flow Direction Grids**DERasterDataset  Upstream flow direction grids corresponding to the grids listed above.  **Workspace**Workspace (Geodatabase)  Geodatabase to work in. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Adjust Accumulation Simple[¶](#adjust-accumulation-simple)

*class* StreamStats\_DataPrep.AdjustAccumSimp[¶](#StreamStats_DataPrep.AdjustAccumSimp)

Simply adjust a flow accumulation grid.

This tool is a wrapper on [make\_hydrodem.adjust\_accum\_simple()](index.html#make_hydrodem.adjust_accum_simple).

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.AdjustAccumSimp.getParameterInfo)(self) | Simple flow accumulation grid adjustment inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.AdjustAccumSimp.getParameterInfo)

Simple flow accumulation grid adjustment inputs.

|  |  |
| --- | --- |
| Parameters: | **Inlet Point**GPFeatureLayer  Point feature class indicating the inlet to the downstream hydrologic unit.  **Flow Direction Grid**DERasterBand  Flow direction grid of the downstream hydrologic unit.  **Flow Accumulation Grid**DERasterBand  Flow accumulation grid of the downstream hydrologic unit.  **HydroDEM**DERasterBand  Downstream hydrologic unit hydro-enforced digital elevation model.  **Output FAC**GPRasterDataLayer  Corrected flow accumulation grid.  **Adjustment Value**GPString  Upstream flow accumulation value to correct the downstream hydrologic unit with, defaults to 150000 grid cells. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

#### Post Hydrodem[¶](#post-hydrodem)

*class* StreamStats\_DataPrep.posthydrodem[¶](#StreamStats_DataPrep.posthydrodem)

ESRI ArcHydro processing using the hydro-enforced DEM and resultant flow direction and flow accumulation grids.

This tool is a wrapper on [make\_hydrodem.postHydroDEM()](index.html#make_hydrodem.postHydroDEM).

Notes

This tool only functions with ESRI ArcMap / Python 2, ESRI ArcPro / Python 3 are currently not supported.

Methods

|  |  |
| --- | --- |
| [getParameterInfo](#StreamStats_DataPrep.posthydrodem.getParameterInfo)(self) | Post hydro-enforcement processing inputs. |

getParameterInfo(*self*)[¶](#StreamStats_DataPrep.posthydrodem.getParameterInfo)

Post hydro-enforcement processing inputs.

|  |  |
| --- | --- |
| Parameters: | **Workspace**DEWorkspace (Geodatabase)  Path to a geodatabase-type workspace.  **hydrodemfac**DERasterDataset  Hydro-enforced flow accumulation grid.  **hydrodemfdr**DERasterDataset  Hydro-enforced flow direction grid.  **str threshold**GPLong  Stream threshold in raster cells.  **str900 threshold**GPLong  str900 grid threshold, in raster cells.  **Sink Link**DERasterBand  Sink link raster name. |
| Returns: | **parameters**list  List of input parameters passed to the execute method. |

### StreamStats DataPrep Library[¶](#streamstats-dataprep-library)

#### databaseSetup Module[¶](#module-databaseSetup)

databaseSetup.check\_walls(*dendrite*, *inwall*, *points*, *outwall=None*)[¶](#databaseSetup.check_walls)

Intersect dendrite with inwall and outwall to check for errors.

|  |  |
| --- | --- |
| Parameters: | **dendrite**str  File path to stream dendrite.  **inwall**str  File path to inwall polygons.  **points**str  File path to output intersection points.  **outwall**str (optional)  File path to outwall polygons. |
| Returns: | **points**Feature class or shapefile  Intersection points between the dendrite, inwall, and outwall. |

databaseSetup.databaseSetup(*output\_workspace*, *output\_gdb\_name*, *hu\_dataset*, *hu8\_field*, *hu12\_field*, *hucbuffer*, *nhd\_path*, *elevation\_projection\_template*, *alt\_buff*, *version=None*)[¶](#databaseSetup.databaseSetup)

Set up the local folders and copy hydrography data into input geodatabases.

This tool creates folder corresponding to each local hydrologic unit, usually a HUC8, and fills those folders with the flowlines, inwalls, and outwalls that will be used later to hydro-enforce the digital elevation model for each hydrologic unit. This tool also creates a global geodatabase with a feature class for the whole domain.

|  |  |
| --- | --- |
| Parameters: | **output\_workspace**str  Output directory where processing will occur.  **output\_gdb\_name**str  Global file geodatabase to be created.  **hu\_dataset**str  Feature class that defines local folder geographic boundaries.  **hu8\_field**str  Field name in “hu\_dataset” to dissolve boundaries to local folder extents.  **hu12\_field**str  Field name in “hu\_dataset” from which inwalls are generated.  **hucbuffer**str  Distance to buffer local folder bounds in map units.  **nhd\_path**str  Path to workspace containing NHD geodatabases.  **elevation\_projection\_template**str  Path to DEM file to use as a projection template.  **alt\_buff**str  Alternative buffer to use on local folder boundaries.  **version**str  Package version number. |
| Returns: | None |

Notes

As this tool moves through each local hydrologic unit it searches the *nhd\_path* for a geodatabase with hydrography data with the same HUC-4 as the local hydrologic unit. If this cannot be found the tool will skip that local hydrologic unit. Non-NHD hydrography data can be used with this tool, but it must be named and organized exactly as the NHD hydrography.

#### elevationTools Module[¶](#module-elevationTools)

elevationTools.checkNoData(*InGrid*, *tmpLoc*, *OutPolys\_shp*, *version=None*)[¶](#elevationTools.checkNoData)

Generates a feature class of no data values.

|  |  |
| --- | --- |
| Parameters: | **InGrid**Raster  Input DEM grid to search for no data values.  **tmpLoc**str  Path to workspace.  **OutPoly\_shp**str  Name for output feature class.  **version**str, optional  StreamStats DataPrepTools version. |
| Returns: | **featCount**int  Count of no data features generated.  **OutPoly\_shp**feature class  No data feature class written to tmpLoc. |

elevationTools.check\_projection(*ds1*, *ds2*)[¶](#elevationTools.check_projection)

Check if the projections of the two files are the same.

|  |  |
| --- | --- |
| Parameters: | **ds1**str  Path to first geospatial file.  **ds2**str  Path to second geospatial file. |
| Returns: | **same**bool  Boolean indicating if the projections are the same or not. |

Notes

Slightly modified from Chris Wills’ post: <https://gis.stackexchange.com/questions/170088/checking-if-two-feature-classes-have-same-spatial-reference-using-arcpy>

elevationTools.compareSpatialRefUnits(*grd*)[¶](#elevationTools.compareSpatialRefUnits)

Compare horizontal and vertical units from a raster dataset. Returns True if units are the same, returns False if they are different.

|  |  |
| --- | --- |
| Parameters: | **grd**str  Path to raster dataset. |
| Returns: | **sameUnits**bool  True if units are the same, False if not. |

elevationTools.elevIndex(*OutLoc*, *rcName*, *coordsysRaster*, *InputELEVDATAws*, *version=None*)[¶](#elevationTools.elevIndex)

Make a raster mosaic dataset of the digital elevation model geospatial tiles supplied to the function.

|  |  |
| --- | --- |
| Parameters: | **OutLoc**str  Path to output location for the raster catalog.  **rcName**str  Name of the output mosaic raster dataset.  **coordsysRaster**str  Path to raster from which to base the mosaic dataset’s coordinate system.  **InputELEVDATAws**str  Path to workspace containing the elevation data to be included in the mosaic raster dataset. Rasters in this workspace should be either geoTiffs or ESRI grids. Rasters must be unzipped, but they may be in subfolders.  **version**str (optional)  StreamStats DataPrepTools version number. |
| Returns: | **OutFC**feature class  Output feature class of polygons with an attribute describing the full path to the elevation data tiles. |

elevationTools.extractPoly(*Input\_Workspace*, *nedindx*, *clpfeat*, *OutGrd*, *version=None*)[¶](#elevationTools.extractPoly)

Extracts watershed DEM from a raster mosaic dataset of tiles based on a watershed polygon feature.

|  |  |
| --- | --- |
| Parameters: | **Input\_Workspace**str  Path to the workspace to work in.  **nedindx**str  Path to the elevation data mosaic dataset.  **clpfeat**str  Path to the clipping feature.  **OutGrd**str  Name of the output grid to be generated in Input\_Workspace.  **version**str, optional  StreamStats DataPrepTools version number. |
| Returns: | **OutGrd**raster  Output extracted raster to Input\_Workspace. |

elevationTools.fillNoData(*workspace*, *InGrid*, *OutGrid*, *version=None*)[¶](#elevationTools.fillNoData)

Replaces NODATA values in a grid with mean values within 3x3 window around each NODATA value.

|  |  |
| --- | --- |
| Parameters: | **workspace**str  Path to the workspace to work in.  **InGrid**str  Name of the input grid to be filled.  **OutGrid**str  Name of the output grid.  **Version**str (optional)  Code version. |
| Returns: | **OutGrid**raster  Filled raster grid output to workspace. |

Notes

May be run repeatedly to fill in areas wider than 2 cells. Note that the output is floating point, even if the input is integer. This function will expand the data area of the grid around the outer edges of data, in addition to filling in NODATA gaps in the interior of the grid.

elevationTools.projScale(*Input\_Workspace*, *InGrd*, *OutGrd*, *OutCoordsys*, *OutCellSize*, *RegistrationPoint*, *scaleFact=100*, *version=None*)[¶](#elevationTools.projScale)

Projects a DEM grid to a user-specified coordinate system, handling cell registration and converts the output grid to integers using a scale factor.

|  |  |
| --- | --- |
| Parameters: | **Input\_Workspace**str  Path to input workspace.  **InGrd**str  Name of the grid to be projected and scaled.  **OutGrd**str  Name of the output grid.  **OutCoordsys**str  Path to the dataset to base the projection off of.  **OutCellSize**int or float  Cell size for output grid.  **RegistrationPoint**str  Registration point for output grid so all grids snap correctly. In the format “0 0” where the zeros are the x and y of the registration point.  **scaleFact**int  Scale factor to convert grid values to integers.  **version**str  StreamStats version number. |
| Returns: | **OutGrd**raster  Rescaled and projected raster file in the input workspace. |

#### make\_hydrodem Module[¶](#module-make_hydrodem)

make\_hydrodem.SnapExtent(*lExtent*, *lRaster*)[¶](#make_hydrodem.SnapExtent)

Returns a given extent snapped to the passed raster.

|  |  |
| --- | --- |
| Parameters: | **lExtent**str  ESRI Arcpy extent string.  **lRaster**str  Path to raster dataset. |
| Returns: | **extent**str  ESRI ArcPy extent string. |

make\_hydrodem.adjust\_accum(*facPth*, *fdrPth*, *upstreamFACpths*, *upstreamFDRpths*, *workspace*, *version=None*)[¶](#make_hydrodem.adjust_accum)

Adjust a downstream flow accumulation (FAC) raster based on upstream flow accumulation rasters.

This function adjusts the FAC of a downstream HUC to include flow accumulations from upstream HUCs. Run this from the downstream HUC workspace. The function will leave the original FAC grids intact and will create a grid named “fac\_global” in the same directory as the original FAC raster. To get true accumulation values in HUCs downstream of other non-headwater HUCs, proceed from upstream HUCs to downstream HUCs in order, and specify the fac\_global grid for any upstream HUC that has one. (It is not essential that the fac\_global contain true global fac values, and in some cases it is not possible since the values get too large to be stored in a raster file. In practice, as long as the receiving cells have accumulation values larger than the stream definition threshold (150,000 cells for 10-m grids), then the ESRI ArcHydro data model will still function.

|  |  |
| --- | --- |
| Parameters: | **facPth**str  Path to downstream flow accumulation grid.  **fdrPth**str  Path to downstream flow direction grid.  **upstreamFACpths**list  List of paths to upstream flow accumulation grids.  **upstreamFDRpths**list  List of paths to upstream flow direction grids.  **workspace**str  local geodatabase to work in.  **version**str (optional)  Stream Stats datapreptool version number. |
| Returns: | **facGlobal**raster  Adjusted flow accumulation raster created in the same directory as fac. |

Examples

adjust\_accum(“./01010001/fac”, 2, [“./01010002/fac”, “./01010003/fac”])

make\_hydrodem.adjust\_accum\_simple(*ptin*, *fdrin*, *facin*, *filin*, *facout*, *incrval*, *version=None*)[¶](#make_hydrodem.adjust_accum_simple)

Simple flow accumulation grid adjustment.

Adds a value to the flow accumulation grid given an input point using a least-cost-path to cascade down through the flow direction grid.

|  |  |
| --- | --- |
| Parameters: | **ptin**str (feature class)  Point feature class representing one inlet to the downstream DEM.  **fdrin**str (raster)  Flow direction raster.  **facin**str (raster)  Name of the flow accumulation raster.  **filin**str (raster)  Burned DEM to use as cost surface.  **facout**str (raster)  Output name of adjusted FAC grid.  **incrval**int  Value to adjust the downstream FAC grid by.  **version**str  Stream Stats version number. |
| Returns: | **hydrodemfac\_global**raster  Adjusted FAC grid written to facout. |

make\_hydrodem.agree(*origdem*, *dendrite*, *agreebuf*, *agreesmooth*, *agreesharp*)[¶](#make_hydrodem.agree)

Function to adjust a DEM to match a vector.

|  |  |
| --- | --- |
| Parameters: | **origdem**Raster Object  Original DEM with the desired cell size.  **dendrite**Raster Object  Dendrite feature layer to adjust the DEM.  **agreebuf**float  Buffer smoothing distance (same units as horizontal map units).  **agreesmooth**float  Smoothing distance (same units as the vertical map units).  **agreesharp**float  Distance for sharp feature (same units as the vertical map units). |
| Returns: | **elevgrid**Raster Object  Conditioned elevation grid. |

Notes

Original function by Ferdi Hellweger, <http://www.ce.utexas.edu/prof/maidment/gishydro/ferdi/research/agree/agree.html>

make\_hydrodem.bathymetricGradient(*workspace*, *snapGrid*, *hucPoly*, *hydrographyArea*, *hydrographyFlowline*, *hydrographyWaterbody*, *cellsize*, *version=None*)[¶](#make_hydrodem.bathymetricGradient)

Generates the input datasets from hydrography features for enforcing a bathymetic gradient in hydroDEM (bowling).

|  |  |
| --- | --- |
| Parameters: | **workspace**str  Path to the geodatabase workspace.  **snapGrid**str  Path to the raster snap grid used for the project.  **hucPoly**str  Path to the bounding polygon for the local folder for which inputs are generated.  **hydrographyArea**str  Path to the double line stream features.  **hydrographyFlowline**str  Path to the flowline features.  **hydrographyWaterbody**str  Path to the waterbody features.  **cellsize**str  Output cell size to use for rasterization.  **version**str (optional)  Package version number. |
| Returns: | **hydro\_flowlines**raster  Grid representation of flowlines.  **hydro\_areas**raster  Grid representation of double line streams and flowlines. |

Notes

Outputs are written to the workspace.

make\_hydrodem.coastaldem(*Input\_Workspace*, *grdNamePth*, *InFeatureClass*, *OutRaster*, *seaLevel*)[¶](#make_hydrodem.coastaldem)

Sets elevations for water and other areas in digital elevation model.

|  |  |
| --- | --- |
| Parameters: | **Input\_Workspace**str  Input workspace, output raster will be written to this location.  **grdNamePth**str  Path to the input DEM grid.  **InFeatureClass**str  Path to the LandSea feature class.  **OutRaster**str  Output DEM grid name.  **seaLevel**float  Elevation at which to make the sea. |
| Returns: | **OutRaster**raster  Output raster with coastal areas corrected. |

Notes

Outputs are written to the workspace.

make\_hydrodem.hydrodem(*outdir*, *huc8cov*, *origdemPth*, *dendrite*, *snap\_grid*, *bowl\_polys*, *bowl\_lines*, *inwall*, *drainplug*, *buffdist*, *inwallbuffdist*, *inwallht*, *outwallht*, *agreebuf*, *agreesmooth*, *agreesharp*, *bowldepth*, *cellsz*, *scratchWorkspace*, *version=None*)[¶](#make_hydrodem.hydrodem)

Hydro-enforce a DEM using hydrography data sets.

This function is used by the National StreamStats Team as the optimal approach for preparing a state’s physiographic datasets for watershed delineations. It takes as input, a digital elevation model (DEM), and enforces this data to recognize the supplied hydrography as correct. Supplied watershed boundaries can also be recognized as correct if avaialable for a given state/region. This function assumes that the DEM has first been projected to a state’s projection of choice. This function prepares data to be used in the ESRI ArcHydro data model (the GIS database environment for National StreamStats).

|  |  |
| --- | --- |
| Parameters: | **outdir**DEworkspace  Working directory.  **huc8cov**DEFeatureClass  Local division feature class, often HUC8, this will be the outer wall of the hydroDEM.  **origdemPth**str  Path to the orignial, projected DEM.  **dendrite**str  Path to the dendrite feature class to be used.  **snap\_grid**str  Path to a raster dataset to use as a snap\_grid to align all the watersheds, often the same as the DEM.  **bowl\_polys**str  Path to the bowling area raster generated from the bathymetric gradient tool.  **bowl\_lines**str  Path to the bowling line raster generated from the bathymetric gradient tool.  **inwall**str  Path to the feature class to be used for inwalling.  **drainplug :**  Path to the feature class used for inserting sinks into the dataset.  **buffdist**float  Distance to buffer the outer wall, same units as the projection.  **inwallbuffdist :**  Distance to buffer the inner walls, same units as the projection.  **inwallht :**  Inwall height, same units as the projection.  **outwallht :**  Inwall height, same units as the projection.  **agreebuf :**  AGREE function buffer distance, same units as the projection.  **agreesmooth :**  AGREE function smoothing distance, same units as the projection.  **agreesharp :**  AGREE function sharp distance, same units as the projection.  **bowldepth :**  Bowling depth, same units as the projection.  **cellsz :**  Cell size, same units as the projection.  **scratchWorkspace**str  Path to scratch workspace.  **version**str (optional)  Package version number.  **Returns (saved to outDIR)**  **——-**  **filldem**raster  hydro-enforced DEM raster grid saved to outDir.  **fdirg**raster  HydroDEM FDR raster grid saved to outDir.  **faccg**raster  HydroDEM FAC raster grid saved to outDir.  **sink\_path**feature class  Sink feature class saved to outDir. |

make\_hydrodem.moveRasters(*source*, *dest*, *rasters*, *fmt=None*)[¶](#make_hydrodem.moveRasters)

Move raster out of a working geodatabase to a destination folder.

|  |  |
| --- | --- |
| Parameters: | **source**str  Path to geodatabase containing the rasters.  **dest**str  Path to destination location.  **rasters**list  List of rasters to move from source to dest.  **fmt**str (optional)  Extension indicating the raster format the output without the leading period, e.g. “tif”. |
| Returns: | None |

make\_hydrodem.postHydroDEM(*workspace*, *facPth*, *fdrPth*, *thresh1*, *thresh2*, *sinksPth=None*, *version=None*)[¶](#make_hydrodem.postHydroDEM)

Generate stream reaches, adjoint catchments, and drainage points

|  |  |
| --- | --- |
| Parameters: | **workspace**str  database-type workspace to output rasters and feature classes.  **facPth**str  Path to the flow accumulation grid produced by hydroDEM.  **fdrPth**str  Path to the flow direction grid produced by hydroDEM.  **thresh1**int  Threshold used to produce the str grid, in raster cells, usually equal to 15,000,000 m$^2$.  **thresh2**int  Threshold used to produce the str900 grid, in raster cells, usually equal to 810,000 m$^2$.  **sinksPth**str (optional)  Path to the snklnk grid, optional.  **version**str (optional)  StreamStats DataPrepTools version to be printed. |
| Returns: | **str**raster  Stream raster where fac > 15,000,000 m$^2$.  **str900**raster  Stream raster where fac > 810,000 m$^2$.  **strlnk**raster  Raster with streams labeld with index values.  **lnk**raster  Merged stream and sink raster.  **cat**raster  Catchment raster.  **drainageLine**feature class  Vectorized streams.  **catchment**feature class  Vectorized catchments.  **adjointCatchment**feature class  Vectorized catchments for use in delineation.  **drainagePoint**feature class  Point located at the greatest flow accumulation value in each catchment. |

Notes

This tool requires archydro to be installed and currently only works with Python 2.

#### topo\_grid Module[¶](#module-topo_grid)

topo\_grid.topogrid(*workspace*, *huc8*, *buffdist*, *dendrite*, *dem*, *cellSize*, *vipPer*, *snapgrid=None*, *huc12=None*)[¶](#topo_grid.topogrid)

Regenerate a DEM based on supplied hydrography features.

|  |  |
| --- | --- |
| Parameters: | **workspace**str  Path to geodatabase workspace.  **huc8**str  Path to local watershed feature class.  **buffdist**int  Distance to buffer HUC8 in horizontal map units.  **dendrite**str  Path to flowline dendrite feature class.  **dem**str  Path to buffered, scaled, and projected DEM.  **cellSize**int  Output cell size.  **vipPer**int  VIP thining value.  **snapgrid**str (optional)  Path to snapgrid to use instead of input DEM.  **huc12**list (optional)  List of paths to HUC12 values if the HUC8 is too large to process in one pass. |
| Returns: | **topodem**raster  DEM generated from topo to raster. |

Notes

See <https://support.esri.com/en/technical-article/000004588>

### Example Scripts[¶](#example-scripts)

These scripts can be found in the examples folder.

#### Initial Database Setup[¶](#initial-database-setup)

Set up the database of local folders for use with Stream Stats.

import sys  
sys.path.append("..") # change environment to see tools  
from databaseSetup import databaseSetup  
  
output\_workspace = r"" # path to file type workspace  
output\_gdb\_name = "" # name for global gdb to be created  
hu\_dataset = r"" # path to local folder features  
hu8\_field = "" # field that identifies local folders / outer walls  
hu12\_field = "" # field that identifies inner walls  
hucbuffer = # buffer distance for local folders  
nhd\_path = r"" # path to folder containing NHD hydrography.  
elevation\_projection\_template = r"" # path to a digital elevation model from which to pull a projection.  
alt\_buff = # alternate buffer distance.  
  
databaseSetup(output\_workspace, output\_gdb\_name, hu\_dataset, hu8\_field, hu12\_field, hucbuffer, nhd\_path,elevation\_projection\_template,alt\_buff)

#### Make Digital Elevation Model Index[¶](#make-digital-elevation-model-index)

Make an index mosaic of digital elevation model tiles to allow seamless extraction of watershed polygons.

import sys  
sys.path.append("..") # change environment to see tools  
from elevationTools import elevIndex  
  
OutLoc = r"" # geodatabase output location  
rcName = "IndexRMD" # output raster mosaic dataset name  
coordsysRaster = 'GEOGCS["GCS\_North\_American\_1983",DATUM["D\_North\_American\_1983",SPHEROID["GRS\_1980",6378137,298.257222101]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]' # projection of rasters to be mosaic, here in ESRI WKT.  
InputELEVDATAws = r"" # file workspace containing DEM tiles to mosaic  
OutFC = "ELEVDATIndexPolys" # output feature class showing tile extents  
  
elevIndex(OutLoc, rcName, coordsysRaster, InputELEVDATAws)

#### Extract Watershed Polygon[¶](#extract-watershed-polygon)

Extract a watershed polygon from a digital elevation model.

import sys  
sys.path.append("..") # change environment to see tools  
from elevationTools import extractPoly  
  
Input\_Workspace = r"" # path to file type workspace  
nedindx = r"" # path to DEM index raster mosaic dataset  
clpfeat = r"" # path to buffered and reprojected local folder clipping feature.  
OutGrd = "dem\_dd" # name of grid to be output  
  
extractPoly(Input\_Workspace, nedindx, clpfeat, OutGrd)

#### Fill NoData Values[¶](#fill-nodata-values)

Fill NoData and/or missing values in a digital elevation model.

import sys  
sys.path.append("..") # change environment to see tools  
from elevationTools import fillNoData  
  
workspace = r"" # file type workspace  
InGrid = "dem\_dd" # name of grid to be filled  
OutGrid = "dem\_fill" # name of output grid  
  
fillNoData(workspace, InGrid, OutGrid)

#### Project and Scale a Digital Elevation Model[¶](#project-and-scale-a-digital-elevation-model)

Reproject a digital elevation model to the project’s target projection and scale the values to turn float grids to integers and save space.

import sys  
sys.path.append("..") # change environment to see tools  
from elevationTools import projScale  
  
Input\_Workspace = r""%huc # folder type workspace  
InGrd = "dem\_dd"  
OutGrd = "dem"  
OutCoordsys = r"" # path to feature class from which to pull the output projection  
OutCellSize = 10 # output cell size  
RegistrationPoint = "0 0" # output registration point for raster alignment  
  
projScale(Input\_Workspace, InGrd, OutGrd, OutCoordsys, OutCellSize, RegistrationPoint)

#### Check Inner and Outer Walls[¶](#check-inner-and-outer-walls)

Check if the flowline dendrite intersect the inner and outer walls used for hydro-enforcement.

import sys  
sys.path.append("..") # change environment to see tools  
from databaseSetup import check\_walls  
  
dendrite = r"" # path to the flowline dendrite  
inwall = r"" # path to the inner wall feature class  
points = r"" # path to the intersection points to output  
outwall = r"" # path to the outer wall feature class (optional)   
  
check\_walls(dendrite, inwall, points, outwall = outwall)

#### Check For NoData Cells[¶](#check-for-nodata-cells)

Check a digital elevation model for cells missing data.

import sys  
sys.path.append("..") # change environment to see tools  
from elevationTools import checkNoData  
  
InGrid = r"" # path to the digital elevation model to check  
tmpLoc = r"" # path a geodatabase workspace  
OutPolys\_shp = "" # name of the output feature class  
  
checkNoData(InGrid, tmpLoc, OutPolys\_shp)

#### Topogrid (optional)[¶](#topogrid-optional)

Perform topogrid digital elevation model pre-conditioning before hydro-enforcement.

import sys  
sys.path.append("..") # change environment to see tools  
from topo\_grid import topogrid  
  
workspace = r""%(huc) # path to geodatabase type workspace  
huc8 = "huc8" # outerwall feature class name  
buffdist = "50" # buffer distance  
dendrite = "NHDFlowline" # dendrite feature class name  
dem = r""%(huc) # path to projected and buffered DEM to re-process  
cellSize = "10" # output cell size  
vipPer = "5" # threshould of points to keep based on VIP score.  
snapgrid = r"" # path to snap grid  
  
topogrid(workspace,huc8,buffdist,dendrite,dem,cellSize,vipPer, snapgrid = snapgrid)

#### Bathymetric Gradient[¶](#bathymetric-gradient)

Generate bathymetric gradient files for use in the hydrodem process.

import sys  
sys.path.append("..") # change environment to see tools  
from make\_hydrodem import bathymetricGradient  
  
workspace = r"" # path to geodatabase to use as a workspace  
snapGrid = r"" # path to snapping grid  
hucPoly = r"" # path to local folder polygon  
hydrographyArea = r"" # path to NHD area feature class  
hydrographyFlowline = r"" # path to NHD flowline feature class  
hydrographyWaterbody = r"" # path to NHD water body feature class  
cellsize = '' # cell size  
  
bathymetricGradient(workspace, snapGrid, hucPoly, hydrographyArea,  
 hydrographyFlowline, hydrographyWaterbody,cellsize)

#### Hydro-Enforce a Digital Elevation Model[¶](#hydro-enforce-a-digital-elevation-model)

Hydro-enforce a digital elevation model such that interal drainage and outer boundaries are correct.

import sys  
sys.path.append("..") # change environment to see tools  
from make\_hydrodem import hydrodem  
  
outdir = r"" # path to geodatabase type workspace  
huc8cov = "huc8" # name of local folder feature class  
origdem = r"" # path to DEM to be enforced  
dendrite = "NHDFlowline" # name of flowline dendrite to use  
snap\_grid = r"" # path to snap grid  
bowl\_polys = "nhd\_wbg"  
bowl\_lines = "wb\_srcg"  
inwall = "inwall\_edit"  
#drainplug = "sinkpoint\_edit"  
drainplug = None # path to drainplug feature class  
buffdist = 50 # outer wall buffer distance  
inwallbuffdist = 15 # inner wall buffer distance  
inwallht = 150000 # inner wall height  
outwallht = 300000 # outer wall height  
agreebuf = 60 # agree routine buffer  
agreesmooth = -500 # agree routine smooth drop  
agreesharp = -50000 # agree routine sharp drop  
bowldepth = 2000 # bowling depth  
cellsz = 10 # cell size  
scratchWorkspace = r"" # path to scratch workspace  
  
hydrodem(outdir, huc8cov, origdem, dendrite, snap\_grid, bowl\_polys, bowl\_lines, inwall, drainplug, buffdist, inwallbuffdist, inwallht, outwallht, agreebuf, agreesmooth, agreesharp, bowldepth, cellsz, scratchWorkspace)

#### Adjust Accumulation Grid[¶](#adjust-accumulation-grid)

Adjust a flow accumulation grid based on values from upstream regions.

import sys  
sys.path.append("..") # change environment to see tools  
from make\_hydrodem import adjust\_accum  
  
facPth = "" # path to downstream flow accumulation grid  
fdrPth = "" # path to downstream flow direction grid  
upstreamFACpths = [r"", r""] # list of paths to upstream flow accumulation grids.  
upstreamFDRpths = [r"", r""] # list paths to upstream flow direction grids, must be in the same order as the flow accumulation grids.  
workspace = r"" # path to geodatabase workspace.  
  
adjust\_accum(facPth, fdrPth, upstreamFACpths,upstreamFDRpths, workspace)

#### Adjust Accumulation Grid (Simple)[¶](#adjust-accumulation-grid-simple)

Adjust a flow accumulation grid using values from upstream regions using a simple method.

import sys  
sys.path.append("..") # change environment to see tools  
from make\_hydrodem import adjust\_accum\_simple  
  
ptin = r"" # path to the downstream inlet shapefile  
fdrin = r"" # path to the downstream flow direction grid  
facin = r"" # path to the downstream flow accumulation grid  
filin = r"" # path to downstream hydro-enforced DEM  
facout = r"" # path to ouput the downstream corrected flow accumulation grid.  
incrval = # value to correct the downstream grid with.  
  
adjust\_accum\_simple(ptin, fdrin, facin, filin, facout, incrval)

#### Post Hydro-Enforcement[¶](#post-hydro-enforcement)

Perform ArcHydro tasks after hydro-enforcemnt of the digital elevation model.

import sys  
sys.path.append("..") # change environment to see tools  
from make\_hydrodem import postHydroDEM  
  
workspace = r"" # geodatabase type workspace  
facPth = r"" # path to flow accumulation grid  
fdrPth = r"" # path to flow direction grid  
thresh1 = 1350000 # stream threshold 1  
thresh2 = 72900 # stream threshold 2  
sinksPth = None # path to sinks (optional)  
  
postHydroDEM(workspace, facPth, fdrPth, thresh1, thresh2, sinksPth = None)

# Indices and tables[¶](#indices-and-tables)

* [Index](genindex.html)
* [Module Index](py-modindex.html)
* [Search Page](search.html)

© Copyright 2020, Theodore Barnhart, Martyn Smith, Alan Rea, Katherine Kolb, Peter Steeves, and Peter McCarthy, U.S. Geological Survey.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/rtfd/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).