

# GeMS Tools for Arc10: Tools for creating, manipulating, and transforming GeMS-style geologic map databases

Version of 2 February 2021

By Ralph A. Haugerud

## Introduction

This document describes a toolbox of Python scripts (“tools” in ArcGIS-speak) for creating, manipulating, and transforming GeMS-style geologic map databases. Use of these tools should make it significantly easier to make, edit, and validate such databases.

The GeMS schema is a standard, supported by the National Cooperative Geologic Mapping Program of the U.S. Geological Survey and the Association of American State Geologists, for digital publication and archiving of single geologic maps. The schema prescribes both the content of a digital geologic map publication and how such content should be encoded in an ESRI ArcGIS geodatabase. Further information on the standard and its development is online at <http://ngmdb.usgs.gov/Info/standards/GeMS/>.

The tools are written to work with ArcGIS version 10. The underlying scripts call the **arcpy** module, rather than the **arcgisscripting** module provided with earlier version of ArcGIS. Many of the tools require ArcGIS version 10.1 or higher. Note that this documentation is for the

ArcMap version of the GeMS toolbox. Most of the documentation will be relevant to the ArcPro version.

## Installing the toolbox

To install this toolbox, unzip (if necessary) and place the folder *gems-tools-arcmap-X.X* (where *X.X* is the version number, 1.3 at the time this document was last revised) in a locale of your choice. Start ArcCatalog or ArcMap, open the Arc Toolbox window, right click on empty space in the Arc Toolbox window, and select "Add Toolbox". Then navigate to the *gems-tools-arcmap-X.X* folder and select the file *GeMS\_ToolsArc105.tbx* (if you are running ArcGIS 10.5) or *GeMS\_ToolsArc10.tbx* (if you are running ArcGIS 10.0 – 10.4).

Right-click again on empty space in the Arc Toolbox window and select "Save settings" and then "Default" to have the GeMS toolbox available next time you open ArcCatalog or ArcMap. Maybe.

Two of the tools (**.docx to DMU** and **DMU to .docx**) require the lxml package to function. Installing this package requires Administrator rights on the host computer and a willingness to navigate rather oblique installation instructions. If you cannot install the lxml package, the remainder of the toolbox will still be functional. The **Purge Metadata** tool requires the USGS EGIS tools be installed. Again, without the EGIS tools the other tools will still function.

## Naming convention

Scripts are named by SchemaVersion\_Toolname\_ArcVersion, e.g.,

GeMS\_CreateDatabase\_Arc10.py

Within a given schema version and Arc version, script versions are identified by the "versionString" constant contained within each script file, which contains the date of the last revision. Note that files *GeMS\_Definition.py* and *GeMS\_utilityFunctions.py* have no dependency on ArcGIS, thus the version of Arc is not specified in the file name.

This naming convention assumes that, within the world of GeMS schema users, there will be only one CreateDatabase script for given versions of ArcGIS and the schema. In other words, if you decide to write your own CreateDatabase script, please give it a different name! In order to retain some predictability in filenames, it is recommended that you rename it by merely inserting something in place of “<new>”, as follows: *GeMS\_CreateDatabase<new>\_Arc10.py*.

## About the code

GeMS Tools was written in Python 2.7 by Ralph Haugerud, Evan Thoms, and others. Scripts have been developed and tested on Windows 7 and 10, using various versions of ArcGIS (9.1 - 10.7).

This software is preliminary or provisional and is subject to revision. It is being provided to meet the need for timely best science. The software has not received final approval by the U.S. Geological Survey (USGS). 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. The software is provided on the condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from the authorized or unauthorized use of the software.

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

The code for these tools is of variable quality. **.docx To DMU, DMU to .docx**, and **Make Polygons** have not been extensively tested.

Feel free to modify the scripts and to extend the toolbox by including additional scripts. If you do so, consider sharing your changes with others, either via email to the GeMS schema authors (*gems@usgs.gov*) and (or) email to the Digital Mapping Techniques list-serve (*digimap@listserv.ua.edu*). Even better, find the GitHub repository for gems-tools-arcmap, clone the master branch, make your modifications, and submit a pull request.

The scripts, and this report, are not subject to U.S. copyright. The scripts are made available under the terms of the Creative Commons CC0 1.0 Public Domain Dedication: You

can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission. See <https://creativecommons.org/publicdomain/zero/1.0/>

File docxModified.py is an exception; it is largely copyrighted by others. See the head of docxModified.py for further information.

## Tool documentation

The tool documentation given here is an extended version of that included within the ArcGIS toolbox. Names in bold (**.docx to DMU**) are tool names presented to the user by ArcGIS. Names in italics (*GeMS\_DocxToDMU\_Arc10.py*) are the corresponding script filenames. Tables in the documentation below identify and describe input parameters for each tool/script.

Most of the scripts call module *GeMS\_utilityFunctions.py*. Several call module *GeMS\_Definition.py*. Both modules are included in directory *gems-tools-arcmap/Scripts*.

### **.docx to DMU** *GeMS\_DocxToDMU\_Arc10.py*

**.docx to DMU** extracts DMU paragraphs from a Microsoft Word document, calculates values of HierarchyKey, and partially fills in table DescriptionOfMapUnits. Non-DMU paragraphs (the rest of the map text) are ignored. The Word document must be formatted using the paragraph styles in USGS Pubs template *MapManuscript\_v1-0\_04-11.dotx*, which is included with this report in folder *gems-tools-arcmap-X.X/Docs*.

DMU paragraphs in the manuscript are those tagged with styles DMU-Heading1, DMU-Heading2, DMU Headnote, DMU Paragraph, DMU Unit 1, DMU Unit 1 (1st after heading), DMU Unit 2, etc. Some character formatting (bold, italic, superscript, subscript, DMU Unit Label character style, FGDCGeoAge font) is recognized and preserved in DescriptionOfMapUnits.

If a paragraph is a unit description, UnitLbl in the map manuscript is related to field Label in the database table DescriptionOfMapUnits. If there is no match, UnitLbl is matched to the field MapUnit in DescriptionOfMapUnits. If the paragraph is a heading, the heading text is

matched to the Name field in DescriptionOfMapUnits. If a headnote, the first 20 characters of the headnote text is matched to the Description field. If no matching row is present in DescriptionOfMapUnits, a new row is created. If a new row is created, the UnitLabel is used to fill both MapUnit and Label fields in DescriptionOfMapUnits.

Values of UnitLabel in the manuscript must be unique.

Parameter	Explanation	Data Type
DMU_manuscript_file	Microsoft Word .docx file formatted according to USGS template MapManuscript_v1-0_04-11.dotx.	File
Geologic_map_geodatabase	GeMS-style geodatabase with DescriptionOfMapUnits table. DMU table may be empty or partly complete.	Workspace

Significant dependencies:

- The lxml package must be present on the host computer. Easiest to install using the Python pip utility. Note that you may want to install it for both 64-bit and 32-bit Pythons (e.g., C:\Python27\ArcGISx6410.5 and C:\Python27\ArcGIS10.5).
- *docxModified.py*, which is included in the GeMS toolbox

### **Attribute By Key Values** *GeMS\_AttributeByKeyValues\_Arc10.py*

**Attribute By Key Values** steps through an identified subset of feature classes in the GeologicMap feature dataset and, for specified values of an independent field, calculates values of multiple dependent fields. It is useful for translating single-attribute datasets into GeMS format, and for using GEMS to digitize in single-attribute mode.

Many geologic map database schemas have characterized features--especially lines--with a single attribute such as "contact--approximate". GEMS characterizes lines by multiple attributes, such that

LTYPE="contact--approximate"

might, depending on map scale and the author's intentions, translate to

Type="contact"  
 IsConcealed="N"  
 LocationConfidenceMeters=150  
 ExistenceConfidence="certain"  
 IdentifyConfidence="certain"  
 Symbol="01.01.03"

This tool simplifies the translation from such schemas into GeMS.

Parameter	Explanation	Data Type
Input_geodatabase	An existing GeMS-style geodatabase with a GeologicMap feature dataset.	Workspace
Key_Value_file	A pipe (   ) -delimited text file that describes mapping from unique values of an independent attribute to values of multiple dependent attributes. See file Dig24K_KeyValues.txt (should be located in folder GeMS_Toolbox/Resources) for an example with format instructions.	Text File
Overwrite_existing	If unchecked, existing values of dependent attributes are not altered. If checked, existing values are overwritten.Compact	Boolean

Comments:

**Attribute By Key Values** requires an accessory keyvalue file. You must create a plain-text file (use gedit, Notepad or Wordpad, save as .txt) that looks like:

#### ContactsAndFaults

```
LTYPE|Type| LocationConfidenceMeters| ExistenceConfidence| IdentityConfidence| Symbol
contact| contact| 20| certain| certain| 01.01.01
contact-approximate| contact| 150| certain| certain| 01.01.03
```

...

#### OrientationPoints

```
STYPE|Type| LocationConfidenceMeters|OrientationConfidenceDegrees|IdentityConfidence|
Symbol
```

bedding| bedding| 20| 5| certain| 06.01.01

...

Bolding is used for emphasis here. It should not appear in your file. An example keyvalue file is provided with the toolbox, in the folder *gems-tools-arcmap-X.X\Resources*. Important aspects of the keyvalue file are: (a) Lines shown in bold above are header lines. Header lines come in pairs: the first line identifies a feature class within the GeologicMap feature dataset. The second line names the independent attribute within that feature class and then the dependent attributes whose values will be calculated based upon the independent attribute. (b) All lines after a pair of header lines are definitions for the respective values specified in that header line until the next pair of header lines is encountered. (c) Values within lines are separated by the pipe symbol (|). This permits the use of commas within values. Trailing (or leading) spaces are acceptable, but not required.

This file can also be created and edited with a spreadsheet program (e.g., LibreOffice Calc, Microsoft Excel), saving as a .csv file and setting the delimiter to “|”.

Attribute By Key Values will

- Read the keyvalue file for instructions
- Open the GeMS-format geodatabase and calculate unassigned feature attributes (values of <NULL>, zero-length string, or zero) based on key values (LTYPE, STYPE, or other). Note that if any attributes have already been assigned they will NOT be changed WITH THE EXCEPTION OF NUMERIC FIELDS WITH ZERO (0) VALUE. This allows you to somewhat easily override the default attributes
- Write short messages to the script output window if unknown key values are encountered

Tool **Attribute By Key Values** can be run multiple times during the course of building a geodatabase (recommended) or just once at the end.

## Compact and Backup *GeMS\_CompactAndBackup\_Arc10.py*

Compact and Backup compacts an existing database and copies it to an archive version. The archive version is named with the current date.

Parameter	Explanation	Data Type
Input geodatabase	Existing geodatabase.	Workspace
Message for log file (optional)	Optional message to be written, with timestamp and username, to file 00log.txt inside the geodatabase directory. If this is a personal geodatabase, the write should fail gracefully and this message will not be recorded.	String

## Create New Database *GeMS\_CreateDatabase\_Arc10.py*

**Create New Database** creates a new GeMS-style geodatabase.

Note that with default settings this tool creates only the minimum required feature dataset, feature classes, and tables. Check the appropriate boxes to add OrientationPoints, GeologicLines, etc. See the GeMS documentation for the purposes of these optional elements. Change the number of cross sections to 1 (or more) to create feature dataset(s) for cross sections.

Running this script from ArcMap with GeMS feature classes in the active data frame AND "Enable edit tracking" checked will generate errors. Avoid such errors by closing ArcMap and running this script from ArcCatalog.

This tool may take several minutes to run.

Parameter	Explanation	Data Type
Output_Workspace	Name of a directory. Must exist and be writable.	Folder
Name_of_new_geodatabase	Name of a file or directory to be created; must not exist in output workspace. Use .gdb extension to create a file geodatabase, .mdb extension to create a personal geodatabase. If no extension is given, will default to .gdb.	String



Spatial_reference_system	May select an ESRI projection file, import a spatial reference from an existing dataset, or define a new spatial reference system from scratch.	Coordinate System
Optional_feature_classes__tables__and_feature_datasets (Optional)	<p>Select items from this list as needed. Note that if you later discover you need an additional feature class, table, or feature dataset, you may</p> <ul style="list-style-type: none"> <li>Define this element from scratch in ArcCatalog, or</li> <li>Run this tool again, creating a new geodatabase with the same spatial reference system, and creating the needed feature class(es), table(s), or feature dataset(s). Then copy and paste the additional elements into your existing geodatabase.</li> </ul>	Multiple Value
Number_of_cross_sections	An integer in the range 0 to 26.	Long
Enable_edit_tracking	<p>If checked, enables edit tracking on all feature classes. Adds fields created_user, created_date, last_edited_user, and last_edited_date. Dates are recorded in database (local) time. Default is checked.</p> <p>This parameter is ignored if the installed version of ArcGIS is less than 10.1.</p>	Boolean
Add_fields_for_cartographic_representations	<p>Default is unchecked.</p> <p>If checked, adds a representation, fields RuleID and Override, and representation rules to feature classes ContactsAndFaults, GeologicLines, and OrientationPoints, and equivalent feature classes in any CrossSection feature datasets that are created. Also adds coded-value domains that tie RuleID values (consecutive integers) to FGDC symbol identifiers (e.g., 1.1.7 for a dotted contact).</p> <p>The representation rules (aka symbols) are from the Arizona Geological Survey and are a subset of the FGDC symbology with a few additional symbols. Rules and coded-value domains are copied from a geodatabase and .lyr files in the Scripts\CartoRepsAZGS directory within the GeMS toolbox directory.</p>	Boolean
Add_LTYPE_and_PTTYE	If checked, adds LTYPE field to ContactsAndFaults and GeologicLines and adds PTTYPE field to	Boolean

OrientationPoints. Useful for digitizing purposes, or for ingesting ALACARTE-style data sets. Default is unchecked.

With the use of Feature Templates for digitizing, values of LTYPE may be helpful proxies for clusters of Type - IsConcealed - LocationConfidenceMeters - ExistenceConfidence - IdentityConfidence - Symbol values.

Add\_standard\_confidence\_values

Default is checked. If checked:

Boolean

1. Attaches standard values of "certain" and "questionable" as a coded-value domain to all ExistenceConfidence, IdentityConfidence, and ScientificConfidence fields
2. Adds definitions and definition source for "certain" and "questionable" to the Glossary table
3. Adds the definition source (FGDC-STD-013-2006) to the DataSources table

The use of these particular values, or of only 2 values, is not required. You may use other values, but they must be defined in the Glossary table.

Significant dependencies:

- *GeMS\_Definition.py*

### **Deplanarize CAF** *GeMS\_Deplanarize\_Arc10.py*

Deplanarize CAF removes excess nodes from arcs in the ContactsAndFaults feature class of the GeologicMaps feature dataset of a GeMS-style geodatabase. **Note: This script has not been extensively tested. PLEASE back up your geodatabase before running this script. Examine the results for correctness.**

Arcs are IDENTITYed with MapUnitPolys to ascertain their bounding MapUnits. If 2 arcs meet at a node and have identical values for attributes Type, IsConcealed,

ExistenceConfidence, IdentityConfidence, LocationConfidenceMeters, DataSourceID, Label, and Notes, they are merged. If 3 arcs meet at a node, the HierarchyKey values of the bounding map units are used to determine which pair of arcs bound the youngest polygon and thus should be continuous across that node if their values of attributes Type, IsConcealed, ExistenceConfidence, IdentityConfidence, LocationConfidenceMeters, DataSourceID, Label, and Notes are identical. If 4 arcs meet at a node, one of the arcs should be IsConcealed = 'Y'. The remaining 3 arcs are then treated as a 3-arc node.

Input geodatabase is assumed to have a GeologicMap feature dataset that contains feature classes ContactAndFaults and MapUnitPolys and a DescriptionOfMapUnits table.

This script will fail if the geodatabase has a Topology class that involves ContactsAndFaults.

Feature class ContactsAndFaults is assumed to have attributes Type, IsConcealed, ExistenceConfidence, IdentityConfidence, LocationConfidenceMeters, DataSourceID, Label, Notes, ContactsAndFaults\_ID, and Symbol. Feature class MapUnitPolys is assumed to have attribute MapUnit. Table DescriptionOfMapUnits is assumed to have attributes MapUnit and HierarchyKey. HierarchyKey must be populated and populated correctly.

Nodes are named by their XY coordinates recorded to within 0.01 map units. We assume no nodes are so close that they have the same name.

Parameter	Explanation	Data Type
Input_geodatabase	Should be a GeMS-style geodatabase.	Workspace

## **DMU to .docx** *GeMS\_DMUtoDocx\_Arc10.py*

**DMU to .docx** reads table DescriptionOfMapUnits in a GeMS-style geodatabase and creates "Description of Map Units" as a Microsoft Word .docx file using paragraph styles defined in USGS Pubs template *MapManuscript\_v1-0\_04-11.dotx*. The resulting file is likely to need minor editing, particularly finding and replacing all instances of “--” with em dashes.

**DMU to .docx** supports a minimal set of markup tags in text within the Description field:

- `<b> ... </b>` bold

- `<i>... </i>` italic
- `<g> ... </g>` FGDCGeoAge font
- `<sup>...</sup>` superscript
- `<sub>...</sub>` subscript
- `<br>` break (start new paragraph)

Parameter	Explanation	Data Type
Source_geodatabase	An existing GeMS-style geodatabase. Table DescriptionOfMapUnits should be (at least partially) populated, and HierarchyKey values should be present, as DMU table is sorted on HierarchyKey before translation to .docx file	Workspace or Feature Dataset
Output_workspace	Directory in which output file will be written. Should exist and be writable.	Folder
Output_filename	Name of output file. File will be overwritten if it already exists. If filename does not end with ".docx", ".docx" will be appended to file name.	String
Use_MapUnit_as_UnitLabl	Values in fields <i>MapUnit</i> or <i>Label</i> can be used for UnitLabl. Default (use <i>MapUnit</i> ) is recommended for constructing and proofing the DMU table. It may be useful to use <i>Label</i> (box unchecked) for creating a final MSWord file of the DMU.	Boolean
List_of_Map_Units	If checked, creates a <b>List of Map Units</b> which omits map unit descriptions.	Boolean

#### Significant dependencies:

- The lxml package (see <http://lxml.de>; look for Windows binary to install) must be present on the host computer. Easiest to install using the Python pip utility. Note that you may want to install it for both 64-bit and 32-bit Pythons (e.g., C:\Python27\ArcGISx6410.5 and C:\Python27\ArcGIS10.5)

- *docxModified.py*, which is included in the *GeMS\_Toolbox/Scripts* directory
- *MSWordDMUtemplate*, which is a directory within the NCGMP09 toolbox Scripts directory. This directory provides essential elements of a Microsoft Word document that uses the paragraph styles defined in USGS Pubs template *MapManuscript\_v1-0\_04-11.dotx*

## **FGDC Metadata, step 1** *GeMS\_FGDC1\_Arc10.py*

**FGDC Metadata, step 1** helps elaborate FGDC CSDGM2-style metadata for all elements of a GeMS-style geodatabase.

This script:

- Exports existing ArcGIS metadata to FGDC-format XML.
- Adds (or replaces) the branch with a statement that this is a GeMS-style database.
- Creates a branch if none is present.
- Adds branch with statement referring to ExistenceConfidence and IdentityConfidence attributes within database.
- Adds branch with statement referring to LocationConfidenceMeters attributes within database.
- Adds branches, one per datasource, that correspond to rows within DataSources table within database.
- Adds (or replaces) branch with statement that summarizes objects (tables, feature datasets, feature classes) within database.
- Writes resulting modified metadata to a standalone XML file named - metadata.xml.

Before running this script, ensure that database is complete and clean by running Validate Database tool. Run this script. After running this script, use a standalone XML metadata editor (e.g., USGS Metadata Wizard) to complete the XML metadata. In particular

- Citation will need completing or improving. Note that most geologic-map databases are not published by themselves; usually they are part of a larger work (a map report). Identify the larger work. The title of the database probably should not be identical to (but likely includes) the title of the larger work. The originator(s) of the database may or may not be the same as the originator(s)/author(s) of the larger work.
- Data quality/Data sources will need significant editing, parsing the concatenated "Source" and "Notes" fields from the DataSources table, now in the datasource title field, into the various fields required by the CSDGM schema.
- If you used the Topology Check and Validate Database scripts, you may want to add a statement about the logical consistency of the database.
- You may wish to describe the process steps relevant to creation of the database.

Then validate the edited XML metadata. Revise as needed. When complete, run script *FGDC Metadata, step 2* to create standalone XML metadata for each table, feature dataset, and feature class within the database. Edit/validate/revise the resulting metadata files, and save. Run script *FGDC Metadata, step 3* which imports all the standalone XML metadata back into the database.

Parameter	Explanation	Data Type
GeMS-style_geodatabase		Workspace

## **FGDC Metadata, step 2** *GeMS\_FGDC2\_Arc10.py*

**FGDC Metadata, step 2** helps elaborate FGDC CSDGM2-style metadata for all elements of a GeMS-style geodatabase. This script copies XML metadata for GeMS-style database as a whole and modifies copies to create formal metadata for objects (tables, feature datasets, feature classes) in the database. It does not create metadata for rasters.

Modifications to input XML metadata are:

- Updating branch using entity descriptions, enumerated value domains, and other domain descriptions taken from the GeMS documentation (transcribed in file `GeMS_Definitions.py` in Scripts subdirectory of the `gems-tools-arcmap-X.X` directory) and the input database. These descriptions may be supplemented by a workgroup- or project-specific definition file, most easily created by editing a copy of `my_GeMS_Definitions.py`, found in the Resources subdirectory of the `gems-tools-arcmap-X.X` directory.
- Pruning branches to include only those appropriate to the object
- Replacing and branches with those provided automatically for the object by ArcGIS (but not for non-spatial tables)
- Modifying `<title>` and text

After running this tool, look at file `-metadataLog.txt` to see what elements remain undescribed. You may provide descriptions by hand editing, or add descriptions to the `definonExtensions` file and re-run this script.

Input metadata are expected to conform to the FGDC CSDGM standard. Output metadata approximate this standard. Output may need further editing (e.g., with the USGS Metadata Wizard). Use Peter Schweitzer's metadata parser (`mp.exe`, available as tool USGS MP Metadata Translator (Conversion) in ArcGIS) to check your work.

At present the modified metadata are written to stand-alone XML files. After running this script and performing any subsequent hand editing, you should (1, optionall) purge the geoprocessing history from the database and (2) import the XML metadata into the database with script *FGDC Metadata, step 3*.

Parameter	Explanation	Data Type
GeMS-style_geodatabase		Workspace
XML metadata	Stand-alone XML metadata for input GeMS-style database. Typically prepared with script CSDGM	File

definitionExtensions (Optional)

metadata, step 1, and then extensively modified using an XML metadata editor such as the USGS FORT Metadata Wizard. Note that we do NOT extract this XML from the input database. This is because--at least at version 10.5 of ArcGIS--the FGDC metadata import-export cycle is not conservative: you do not get out what you put in.

Optional: file with extensions to definitions given in GeMS\_Definitions.py. Useful for persistent entity and field descriptions, amongst other things. See file *my\_GeMSDefinitions.py*, in Scripts directory of gems-tools-arcmap-X.X, for an example and further directions.

File

### FGDC Metadata, step 3 *GeMS\_FGDC3\_Arc10.py*

**FGDC Metadata, step 3** helps elaborate FGDC CSDGM2-style metadata for all elements of a GeMS-style geodatabase. The script imports standalone XML metadata files into a GeMS-style geologic map database. XML files may be prepared with tools CSDGM Metadata, step1 and CSDGM Metadata, step2 and hand editing with a standalone metadata editor such as the USGS Metadata Wizard.

Given a database named DB.gdb, XML files are assumed to be named

- DB-metadata.xml
- DB\_Glossary-metadata.xml
- DB\_GeologicMap-metadata.xml
- DB\_ContactsAndFaults-metadata.xml

and similar. Any error messages from the import process are deleted; to preserve them, edit the script to set debug = True.

Parameter	Explanation	Data Type
GeMS-style_geodatabase		Workspace



Xml directory (optional)	Directory which contains XML metadata files. Default is the directory which hosts the GeMS_style_geodatabase.	Folder
--------------------------	---	--------

### **Fix Strings** *GeMS\_FixStrings\_Arc10.py*

**Fix strings** cleans up text strings in a GeMS database. Removes leading and trailing spaces and fixes bad values.

Parameter	Explanation	Data Type
Input_geodatabase	An existing GeMS-Style geodatabase	Workspace

### **Geologic Names Check** *GeMS\_GeolexCheck\_Arc10.py*

Geologic Names Check automates some of the steps in a geologic names review as required by USGS publication policy. It searches within the DescriptionOfMapUnits table for names and usages found in the U.S. Geologic Names Lexicon (Geolex) and provides a report template in spreadsheet form for the author and reviewer to use during the review process. The tool reports the Geolex names found within the map unit name, the usages associated with those names, and whether or not the author's choice of geographic extent matches that found in Geolex. Comparisons of age and status (formal vs informal) are not at this time considered.

Parameter	Explanation	Data Type
DMU_Table	A GeMS-compliant DescriptionOfMapUnits table. Must contain the fields HierarchyKey, MapUnit, Fullname, and Age, but they may be in any order, need not be CamelCase, and there are no special constraints on how the values are formatted. Values in Name are searched for Geolex names. If no Geolex name is found, Fullname is searched as well.	Table
States_extent	One or more (comma separated) state or territory abbreviations. Examples are "WA", or "ID,OR,WA".	String
open_report_when_completed? (optional)	Should the Excel report file be opened when the script has finished running?	Boolean

## **Inclination Numbers** *GeMS\_InclinationNumbers\_Arc10.py*

Creates point feature class OrientationPointLabels with dip and plunge numbers for appropriate features within OrientationPoints. Adds the new feature class to your map composition.

If this script fails because of locking issues, try (a) Stop Editing, or (b) save any edits and save the map composition, exit ArcMap, and restart ArcMap. Maybe the script will then run satisfactorily.

Parameter	Explanation	Data Type
Feature_dataset	The feature dataset with class OrientationPoints for which inclination annotation is to be created.	Feature Dataset
Map_scale_denominator	Denominator of the map scale fraction. For a 1:24,000-scale map, enter 24000.	Double

## **Make Polygons** *GeMS\_MakePolys3\_Arc10.py*

Make Polygons:

- Creates (or recreates) feature class MapUnitPolys from lines in ContactsAndFaults, excluding lines for which IsConcealed='Y'
- Attributes polygons using temporary label points created from any polygons in the pre-existing MapUnitPolys and any label points in feature class MapUnitPoints.
- Flags (a) polygons with multiple, conflicting, label points, (b) multiple, conflicting label points within a single polygon, (c) polygons with blank or null MapUnit values, and (d) contacts that separate polygons of the same MapUnit. These errors are written to feature classes errors\_multilabelPolys, errors\_multilabels, errors\_unlabeledPolys, and errors\_excessContacts.

- IDENTITYs the new MapUnitPolys with a temporary copy of the old MapUnitPolys. Any polygons--or fragments of polygons--that have changed their MapUnit are saved in feature class edit\_ChangedPolys
- Optionally, saves old feature class MapUnitPolys to MapUnitPolysNNN, where NNN is a successively higher integer.
- When used within ArcMap, selecting a MapUnitPolys layer will cause the data source of the layer to be changed to the newly created polygon feature class.with no change in customization.

Note that if you remove a contact or fault that separates two existing polygons with different MapUnit values and run **Make Polygons**, the resulting multi- (temporary) label polygon is correctly flagged as an error. Re-running **Make Polygons** will (incorrectly) cause this error to disappear! Moral: Run **Make Polygons** once and FIX THE PROBLEMS.

While running, **Make Polygons** writes (overwrites) and deletes temporary feature classes xxxlabel, xxxpolys, and xxxtlabels, all within the GeologicMap feature dataset.

Parameter	Explanation	Data Type
input_geodatabase	An existing GeMS-style geodatabase.	Workspace
Save_old_MapUnitPolys (Optional)	If checked, saves old MapUnitPolys feature class to feature class MapUnitPolysNN, where NN is a successively higher zero-padded integer. Default is checked (true)..	Boolean
Saved-layer_directory (Optional)	Directory in which .lyr files are saved for any map layers with sources MapUnitPolys, errors_excessContacts, errors_multilabelPolys, errors_multilabels, and errors_unlabeledPolys. These .lyr files are deleted when script completes.  Must have write permission.  Default is the directory that hosts the input geodatabase.	Folder
Label_points_feature_class (Optional)	An optional point feature class with attribute MapUnit (and perhaps other attributes), which may used to	Feature Class

label polygons. Familiar to those who used workstation ArcInfo, in which such features were necessary. ArcGIS does not \_require\_ label points; polygons can be created and attributed without them.

### **Make Topology** *GeMS\_MakeTopology\_Arc10.py*

Creates and validates a topology feature class within a GeMS-style feature dataset. The new topology class is named GeologicMap\_Topology (for the GeologicMap feature class) or xxx\_Topology (for all other feature classes, where xxx is the prefix for the ContactsAndFaults-equivalent feature class within that feature dataset). Any existing topology with this name will be deleted. The input feature dataset should contain feature classes xxxContactsAndFaults and xxxMapUnitPolys (where xxx may be null).

Esri topology rules applied are:

Must Not Overlap (Line)

Must Not Self-Overlap (Line)

Must Not Self-Intersect (Line)

Must Be Single Part (Line)

Must Not Have Dangles (Line)

Must Not Overlap (Area)

Must Not Have Gaps (Area)

Boundary Must Be Covered By (Area-Line)

Parameter	Explanation	Data Type
Input_feature_dataset		Dataset
use_MUP_rules	Default = checked.	Boolean

	<p>If checked, adds rules that involve MapUnitPolys feature class (no gaps, no overlaps, boundaries must be covered by ContactsAndFaults).</p> <p>In some cases it is useful to build a topology that does not incorporate MapUnitPolys, largely so that this topology need not be deleted before (re)making polygons.</p>	
--	--	--

## MapOutline *mapOutline\_Arc10.py*

**MapOutline** calculates a map boundary and tics for rectangular (in latitude-longitude space) areas. Locations of boundary and tics are projected to the specified output coordinate system.

Locations of boundary and tics may be specified in either NAD27 or NAD83, independently of the output datum and projection. This may be useful if, for example, it is desirable that a map boundary coincides with that of a published USGS quadrangle map (most of which are located at even latitude-longitude values in NAD27) but because the mapper is working with GPS coordinates or dense lidar data, both of which are commonly in NAD83-based projections, the map database should be in a NAD83-based projection.

Output feature classes MapOutline and Tics are written to the top level of the output geodatabase. Any existing feature classes with these names will be overwritten.

Parameter	Explanation	Data Type
SE_longitude	Longitude of SE corner of map rectangle. May be in decimal degrees (-120.625), degrees - decimal minutes (-120 37.5) or degrees - minutes - seconds (-120 37 30). West longitudes (all of US) are negative.	String
SE_latitude	Latitude of SE corner of map rectangle. May be in decimal degrees (42.375), degrees - decimal minutes (42 22.5) or degrees - minutes - seconds (48 22 30). South latitudes (none in US) are negative..	String

width__longitudinal_extent__	E-W extent of map rectangle. Values <= 5 are assumed to be in degrees. Values >5 are assumed to be in minutes. Enter 600 to obtain a 10 degree extent.	Double
height__latitudinal_extent__	N-S extent of map rectangle. Values <= 5 are assumed to be in degrees. Values >5 are assumed to be in minutes. Enter 600 to obtain a 10 degree extent.	Double
tic_spacing	Value is in decimal minutes	Double
Is_NAD27	Check to locate map boundary and tics at NAD27 lat-long positions. Uncheck to locate at NAD83 positions.	Boolean
output_geodatabase	Must be an existing, writable geodatabase. Please do not select a feature dataset or feature class (though ArcGIS will let you do so).	Workspace or Feature Dataset
output_coordinate_system	Browse to select coordinate system from ArcGIS-provided coordinate system definitions, to import a coordinate system from an existing data set (RECOMMENDED) or define a coordinate system from scratch..	Coordinate System
scratch_workspace	Directory that is writable. Files xxxbox.csv, xxxtics.csv, xxx1.dbf, xxx1.dbf.xml will be written to this directory and then deleted. Existing files with these names will be lost!	Folder

## Project Map Data to Cross Section *GeMS\_ProjectCrossSectionData\_Arc10.py*

**Project Map Data to Cross Section** generates backdrop feature classes useful in constructing a geologic cross section. Inputs include the GeologicMap feature dataset of a GEMS-style geodatabase, a cross-section line (feature class, feature layer, or selection), and a DEM. The cross-section line need not be straight.

Final output is written to feature dataset ***CrossSectionxx***, where xx is `Output_name_token`. This feature dataset will be created if it does not exist.

By default, all feature classes in the GeologicMap feature dataset are projected into the cross section feature dataset, with the exception of feature classes whose names begin with `errors_` and `ed_`. Alternately, you may choose to project specified feature classes.

- Polygons are projected to line segments that follow the topographic profile of the section line. Projected lines for MapUnitPolys (or any other polygon feature class that spans the entire section line) constitute the topographic profile.
- Lines are projected to short vertical line segments located at the point where lines cross the section line. Line segments for ContactsAndFaults are below the topographic profile. All other line feature classes are projected to line segments above the topographic profile.
- Points are projected to points at the appropriate elevation in the vertical plane of the section line.

For point feature classes the distance from the section line and the local azimuth of the section line at the projection point are recorded in fields DistanceFromSection and LocalCsAzimuth. If a point feature class has fields Azimuth and Inclination (e.g., OrientationPoints), new fields ApparentDip, Obliquity (angle between Azimuth and LocalCsAzimuth), and PlotAzimuth (rotation field for symbolization) are calculated.

Output feature classes are named **ed\_CsxxInputFeatureClass**, where xx is Output\_name\_token. Existing feature classes with these names will be overwritten.

This script also creates empty GEMS feature classes CSxxContactsAndFaults, CSxxMapUnitPolys, and CSxxOrientationPoints if they are not already present in the output feature dataset. You may find it useful to load data from the appropriate ed\_CSxxxx feature class into these classes.

Parameter	Explanation	Data Type
GEMS-style_geodatabase	An existing GEMS-style geodatabase. May be .gdb or .mdb. Must have a GeologicMap feature data set.	Workspace
Project_all_features_in_GeologicMap	Default = true (checked). If checked, all point, line, and polygon feature classes within GeologicMap feature dataset (except for certain edit classes) are projected into cross section. If checked, the Feature_classes_to_Project parameter is ignored. If unchecked, you should specify which feature classes should be projected with the Feature_classes_to_Project parameter	Boolean

Feature_classes_to_Project (Optional)	Used only if Project_all_features_in_GeologicMap is False (unchecked). Specify which point, line, and polygon feature classes to project. Note that the tool will happily project feature classes that are not within the GeologicMap feature dataset. This may not be meaningful!	Multiple Value
DEM	Digital elevation model that encompasses section line and buffered selection polygon. Z units should be equal to XY units of GeologicMap feature dataset.	Raster Dataset
Section_line	Feature class, feature layer, or selection that contains ONLY ONE element. Section line need not be straight. Zigs are OK and you can make a section along a stream course. Points to be projected onto section line are selected by buffering the section line with FLAT line ends. To project points that are beyond the ends of the section line, extend the section line. Points are projected to nearest point on section line using <i>LocateFeaturesAlongRoutes</i> .	Feature Layer
Start_quadrant	Where does section start? (This may not control the section orientation. If you don't get the results you expect, try flipping the section line.)	String
Output_name_token	Short text token used to name output feature dataset and feature classes.  The output feature dataset will be named <i>CrossSectionOutput_name_token</i> . If this feature dataset does not exist it will be created.  Output feature classes will be named <i>ed_CSOutput_name_tokenInput_featureclass_name</i>  Suggested values are A, B, C, ... If the cross section has vertical exaggeration that is not 1, set the token to B5x, or C10x, or ...	String
Vertical_exaggeration	Default is 1. If you change to another value, suggest you incorporate this value in the output name token.	Double
Selection_distance	Distance, in GeologicMap XY units, within which point features are projected onto the cross section plane.	Double



Add_LTYPE_and_PTTYE	Feature classes CSxxMapUnitPolys, CSxxContactsAndFaults, and CSxxOrientationPoints will be created in the output feature dataset if they are not already present. Check this box to add LTYPE field to CSxxContactsAndFaults and PTTYPE field to CSxxOrientationPoints.	Boolean
Force_exit	If checked, use sys.exit() to force an exit with error. Allows re-run of tool without re-entry of all parameters. Default is false (unchecked).	Boolean
Scratch_workspace (Optional)	If blank, output feature dataset will be used as scratch workspace. Temporary feature classes have names that begin with 'xxx'. Existing feature classes with these names will be overwritten.	Workspace or Feature Dataset
Save_intermediate_data	Default = NO (unchecked).  This script creates temporary tables in the input geodatabase and temporary feature classes in the scratch workspace (default is the output feature dataset). If this box is unchecked, these tables and feature classes will not be deleted when the script finished.  Check this box if you need these temporary data for troubleshooting. Note that using the default scratch workspace, saving intermediate data and then running the tool to create another feature dataset will not work. You must first delete the temporary data files within the first feature dataset.	Boolean

## **Project Points to Cross Section** *GeMS\_ProjectPtsToCrossSection\_Arc10.py*

**Project Points to Cross Section** projects points within a specified horizontal distance of a cross-section line into the vertical cross-section plane. Output is a feature class with attributes className\_ID, ID, and DistanceFromSection. The source point feature class may then be joined to the output feature class--using ID and \_ID as the join fields--to make the entire set of source featureclass attributes available. The cross-section line must be straight!

If points are for orientation data (i.e., have an Azimuth attribute), the apparent dip (or plunge) of each orientation is calculated, as well as the obliquity of the measurement to the section line and the angle (PlotAzimuth) at which a symbol should be rotated.

Script *Project Map Data To Cross Section* is likely to be more useful. It will project data to a zig-zag or curved section line.

Parameter	Explanation	Data Type
Featureclass_that_contains_cross-section_line	Line feature class that contains the cross-section line. Typically, the CartographicLines feature class within the GeologicMap feature dataset.	Feature Class
Cross-section_Label	One line needs to be selected from the feature class that contains cross-section lines. Selection is on the value of the "Label" field. Cross-section lines should have unique Label values.	String
Point_class_to_be_projected	Point feature class that is to be projected. May be well points, OrientationPoints, or other.	Feature Class
Key_field	Field in point class to be projected that serves as a primary key. If there is an _ID field in the point feature class, Key_field will be automatically reset to this _ID field	Field
Vertical_exaggeration	Desired vertical exaggeration. A real number. Default value is 1.0.	Long
DEM	Digital elevation model that overlaps the points to be projected. Vertical units should be same as horizontal units, or the vertical exaggeration will be incorrect.	Raster Dataset
Max_projection_distance	Distance from cross-section line within which points will be projected. Value is in map units.	Long
Output_feature_dataset	Location of output data. Suggest use of one of the GEMS-defined feature datasets CrossSectionA, CrossSectionB, ... If appropriate feature dataset is not present, quit and run Create New Database tool to create empty cross-section feature datasets, which can then be copied and pasted into the existing map geodatabase.	Feature Dataset
Output_feature_class_name	Name of the output featureclass. Any existing featureclass with this name will be overwritten.	String

Note that featureclass names must be unique within a geodatabase. This suggests that, if there may be multiple cross sections, the name of the enclosing feature dataset should be incorporated within the featureclass name. For example:

```
MyGeodatabase.gdb
  GeologicMap
    CrossSectionA
      CSAWellPts
      CSAOrientationPts
    CrossSectionB
      CSBWellPts
      CSBOrientationPts
```

Scratch\_workspace (Optional)

If left blank, defaults to the output feature dataset. Several feature classes (xxx1, xxx2, and xxx4) will be written to this workspace and then deleted. Any existing feature classes with these names will be deleted.

Workspace or  
Feature Dataset

## Purge Metadata *GeMS\_PurgeMetadata\_Arc10.py*

Purges metadata of geoprocessing history (and probably any other elements that don't have a place within the FGDC CSDGM2 metadata schema). Steps through all feature datasets, feature classes within feature datasets, tables, and the dataset as a whole; for each item this script:

1. Exports existing metadata as FGDC CSDGM2 metadata
2. Clears metadata with USGS EGIS Clear Metadata tool
3. Imports exported CSDGM2 metadata

Note that feature classes that are not within a feature dataset are not processed. The USGS EGIS tools must be installed for this script to work.

This script will leave the directory that hosts the geodatabase with a multitude of .xml metadata files

Parameter	Explanation	Data Type
Input_geodatabase		Workspace
Output_directory (Optional)	Where .xml metadata files are created. If not specified, defaults to directory that hosts input geodatabase.	Folder

## Relationship Classes *GeMS\_RelationshipClasses1\_Arc10.py*

A GeMS geodatabase has numerous implicit relationships. For example,

- table DescriptionOfMapUnits references table DataSources via the DefinitionSourceID
- table DescriptionOfMapUnits references table GeoMaterialDict via GeoMaterial
- feature class MapUnitPolys references table DescriptionOfMapUnits via MapUnit
- feature class MapUnitPolys references table DataSources via DataSourceID
- feature class MapUnitPolys references table Glossary via IdentifyConfidence

These relationships may be made explicit as joins or relates (which live in an .mxd document) or as relationship classes which live in the geodatabase. **Relationship Classes** creates a number (though in many cases not all useful or all possible) of relationship classes within a GeMS-style database.

Note that:

- Relationship classes are named by the referencing table and field, e.g., DMU\_DefinitionSource, DMU\_GeoMaterial, MUP\_MapUnit, MUP\_DataSource, MUP\_IdentityConfidence
- Any existing relationship class with the same name is overwritten
- This script creates a RelationshipClasses feature dataset. If this feature dataset already exists it is overwritten. Newly-created relationship classes are NOT placed in this feature dataset, though you may do so afterwards

Parameter	Explanation	Data Type
GeMS_geodatabase	Name of geodatabase to which relationship classes will be added. Note that any existing relationship classes with the same names will be deleted.	Workspace

### Set ID values *GeMS\_reID\_Arc10.py*

GeMS-style geodatabases use `_ID` values as primary keys; these values are repeated as ID values in other tables where they serve as foreign keys to tie tables together. **Set ID values** generates `_ID` values while preserving any links established by existing `_ID` and ID values. As an option, GUIDs may be substituted for plain-text `_ID` and ID values.

The script makes a copy of the input geodatabase and works on the copy. If you are satisfied with the results, delete the input geodatabase and rename the output.

Parameter	Explanation	Data Type
Input_GeMS-style_geodatabase	The geodatabase for which <code>_ID</code> values are to be created or recreated. Must exist. May be file geodatabase (.gdb) or personal geodatabase (.mdb).	Workspace
Output_geodatabase	Name of output. Must not exist. Should be same type (.gdb or .mdb) as input geodatabase.	String
Use_GUIDs (Optional)	Default is unchecked (false), which creates <code>_ID</code> values as several characters which denote the table (e.g., MUP for MapUnitPolys) followed by consecutive zero-padded integers: MUP0001, MUP0002, MUP0003, etc. If checked, creates GUIDs (Globally Unique IDs which are many-byte nonsense strings) for <code>_ID</code> values.	Boolean
Do_not_reset_DataSource_IDs	If unchecked, resets values of <code>DataSources_ID</code> and all <code>DataSourcesID</code> , <code>LocationSourceID</code> , <code>AnalysisSourceID</code> and similar that refer to <code>DataSources_ID</code> . Default is checked, which leaves these values unchanged.	Boolean

## **Set PlotAtScale values** *GeMS\_SetPlotAtScales\_Arc10.py*

Sets values of item PlotAtScale so that a definition query

$[\text{PlotAtScale}] \geq \text{MapScale}$

limits displayed features to those that can be shown without crowding at the specified map scale.

Note that MapScale is the DENOMINATOR of the scale ratio; that is, PlotAtScale of 24000 means feature can be plotted without crowding at map scales of 1:24,000 and larger.

Input feature class must have PlotAtScale field. If input feature class is named "OrientationPoints", point to be plotted of a too-close pair is biased towards (a) upright or overturned bedding (not bedding without facing direction), (b) bedding (not joint, foliation, lineation, ...), and (c) lower value of OrientationConfidenceDegrees. Large feature classes (>1,000) and large Maximum\_value\_of\_PlotAtScale (>100,000) lead to long calculation times. A more efficient algorithm might be in order.

Parameter	Explanation	Data Type
Feature_class	Must have PlotAtScale field.	Feature Class
Minimum_separation__mm__	Set this on basis of symbol diameter, in mm on the page. For FGDC structure symbols, 8 works OK.	Double
Maximum_value_of_PlotAtScale	Large values (e.g., >100,000) with large feature classes (e.g., > 1,000 features) can take a LONG time to calculate.	Double

## **Set Symbol Values** *GeMS\_SetSymbols\_Arc10.py*

**Set Symbol Values** sets the *Symbol* attribute for some features in a GeMS-style geodatabase to match symbol IDs in the GSC implementation of the FGDC Digital Cartographic Standard for Geologic Map Symbolization (FGDC-STD-013-2006).

Values for line symbols are calculated on the basis of map scale and the GEMS attributes *Type*, *IsConcealed*, *LocationConfidenceMeters*, *ExistenceConfidence*, and *IdentityConfidence*.

Values for orientation point symbols are calculated from *Type* and *OrientationConfidenceDegrees*, e.g.,

for Type = bedding

if OrientationConfidenceDegrees <= threshold, symbol = 06.02

if OrientationConfidenceDegrees > threshold, symbol = 06.33 (bedding symbol with open center)

This script calculates Symbol for feature classes ContactsAndFaults, GeologicLines, and OrientationPoints.

Symbols are only calculated for recognized Type values. Recognized *Type* values are given in file Type-FgdcSymbol.txt, which is located in the Resources folder of the GeMS toolbox. That file may be edited to reflect your choices for *Type* values, add additional *Type* values, or change the correlation between *Type* and symbol ID.

Parameter	Explanation	Data Type
Feature_dataset	The feature dataset with classes xxxContactsAndFaults, xxxGeologicLines, and xxxOrientationPoints for which Symbol values are to be calculated. xxx may be null (the GeologicMap feature dataset), "CSA" for CrossSectionA, or similar.	Feature Dataset
Map_scale_denominator	Denominator of the map scale fraction. For a 1:24,000-scale map, enter 24000.	Double
Certain_to_approximate_-threshold_mm_on_map	<p>The value of LocationConfidenceMeters, converted to mm on the map, at which lines change from "certain" (continuous) to "approximate" (dashed).</p> <p>If map scale denominator is 24,000 and the certain to approximate threshold is 1.0, lines with LocationConfidenceMeters &lt;= 24 will have continuous-line symbols. If map scale denominator is 100,000 and the certain to approximate threshold is 2.5, lines with LocationConfidenceMeters &lt;= 250 will have continuous-line symbols.</p>	Double
Use_inferred_short_dash_line_-symbols	FGDC-STD-013-2006 defines "approximate" (long dash) and "inferred" (short dash) symbols and distinguishes inferred on the basis of how-located, not how-well-located. Yet by the how-located	Boolean

	<p>standard (not directly observed), most geologic lines are inferred, and this is not common usage.</p> <p>Check this box to use "inferred" symbols for lines with LocationConfidenceMeters greater than a threshold value.</p>	
Approximate_to_inferred_threshold_mm_on_map (Optional)	<p>The value of LocationConfidenceMeters, converted to mm on the map, at which lines change from "approximate" (long dash) to "inferred" (short dash). If threshold is 4 and map scale denominator is 24,000, lines with LocationConfidenceMeters &gt; 98 will be drawn with short dashes.</p>	Double
Use_approximate_strike-and-dip_symbols	<p>Default is checked. Uncheck to avoid use of open-center symbols for approximately-oriented bedding.</p>	Boolean
OrientationConfidenceDegrees_threshold	<p>FGDC-STD-013-2006 describes special symbols for some common orientation-data types to denote "when the measurement of strike and (or) dip value is approximate but the location of observation is accurate." For these types, if OrientationConfidenceDegrees &gt; threshold value, the approximate symbol is assigned.</p>	Double
Set_polygon_symbols_and_labels	<p>If checked, <i>Symbol</i> and <i>Label</i> values will be calculated for MapUnitPolys, using values from table DescriptionOfMapUnits. Default is checked.</p>	Boolean

### Symbol to RGB *GeMS\_WPGCMYK\_RGB.py*

Calculates values of AreaFillRGB in table DescriptionOfMapUnits of a GeMS-style geodatabase. Symbol values must be present and are assumed to reference the WPGCYMK color set, which is included in style file USGS Symbols2.style.

Parameter	Explanation	Data Type
Input_geodatabase		Workspace



Significant dependencies:

- Calls module *colortans.py*, which calls module *wpgdict.py*. Both are included in the *gems-tools-arcmap-X.X/Scripts* directory

## **Topology Check** *GeMS\_TopologyCheck\_Arc10.py*

**Topology Check** analyzes a map-like feature dataset within a GeMS database for topology appropriate for a geologic map. It assumes there is a *DescriptionOfMapUnits* table within the host geodatabase. Analysis does not alter the input geodatabase. Tool is very useful for catching digitizing errors related to poorly-controlled snapping. See head of the script file for more information.

Not all geologic-map topology errors will be identified. Some of the features identified may not be errors. Use your judgement!

Output is a folder *<inGdb>\_Topology*, located in the same folder as the input geodatabase, that contains geodatabase *<inGdb>\_TopologyCheck.gdb* and HTML file *<FDS>.html*, where *<FDS>* is the name of the analyzed feature dataset. Within *<inGdb>\_TopologyCheck.gdb/<FDS>* are several feature classes:

- **xxxContactsAndFaults**---copy of input *ContactsAndFaults*
- **xxxMapUnitPolys**---copy of input *MapUnitPoints*
- **xxxContactsAndFaults\_planarized**---planarized CAF, with left and right polygon info
- **xxxContactsAndFaults\_unplanarized**---unplanarized CAF
- **errors\_xxx\_BadNodes**---nodes with probable geologic-topology inconsistencies
- **errors\_xxx\_FaultFlipNodes**---nodes at which fault-line direction changes. Probably one fault line should be flipped
- **<FDS>\_topology**---ESRI topology class
- **errors\_<FDS>\_line**---ESRI topology errors

- errors\_<FDS>\_point---" "
- errors\_<FDS>\_poly---" "
- **xxxMissingConcealedCAF\_nodes**---nodes at which a concealed contact could be drawn

xxx is a short token that signifies the feature dataset. Feature classes errors\_<FDS>\_line, errors\_<FDS>\_point, and errors\_<FDS>\_poly are each only present if there are errors of that geometry.

For each feature class within which duplicate point features are detected, a table

- dups\_<FC>

is written, where <FC> is the name of the feature class.

BadNodes includes:

- 1-arc nodes that are not faults (i.e., non-fault dangles)
- 2-arc nodes at which there is a change in Type, or one arc is concealed and one is not
- 3-arc nodes at which:
  - 1 or 2 arcs are concealed
  - no two arcs have same Type
  - all arcs are concealed and not all adjoining map units are identical
  - all arcs are not faults and the youngest adjoining map unit is not bounded by arcs of same Type (note that for some maps this rule may not be applicable)
- 4-arc nodes at which:
  - opposite arcs don't have same Type
  - 0, 3, or 4 arcs are concealed
  - all arcs are faults

- fault marks transition from unconcealed to concealed crossing arcs
- any 5+ arc nodes

These rules stem from the following observations:

- Faults have offset
- Contacts are continuous
- The improbability of coincident truncations implies that there are no 4-arc junctions unless one arc is the concealed continuation of its opposite arc

The **Nodes** and **MapUnit adjacency** modules use functions *isFault* and *isContact*, imported from *GeMS\_utilityFunctions*. These functions may need to be modified for some geodatabases.

```
def isFault(lType):
```

```
    if lType.upper().find('FAULT') > -1:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def isContact(lType):
```

```
    lType = lType.upper()
```

```
    if lType.find('CONTACT') > -1:
```

```
        return True
```

```
    elif lType.find('FAULT'):
```

```
        return False
```

```
    elif lType.find('SHORE') > -1 or lType.find('WATER') > -1:
```

```
        return True
```

```
    elif lType.find('MAP') > -1: # is map boundary?
```

```

        return False

    elif lType.find('GLACIER') > -1 or lType.find('SNOW') > -1 or
lType.find('ICE') > -1:

        return True

    else:

        return False

```

Parameter	Explanation	Data Type
Feature dataset	A map-like feature dataset. Contains xxxContactsAndFaults (arcs) and xxxMapUnitPolys (polygon) feature classes.	Feature dataset
HKey test value	<p>MapUnits with HierarchyKey values less than HKey test value are deemed covering units beneath which one might choose to draw concealed faults and contacts. HKey test value is typically a string, something like '2', or '01-02'. Don't enter the enclosing quotes.</p> <p>See the DescriptionOfMapUnits table in the database you are analyzing to select a value.</p> <p>Correct identification of covering units requires that the youngest "map units" (water, permanent snow, unmapped area) have the lowest HKey values. Covering units may not be correctly identified in some cases in which map units are divided into rock-type silos (e.g., "Igneous Rocks").</p>	String

## Translate To Shapefiles *GeMS\_TranslateToShape\_Arc10.py*

**Translate to Shapefiles** converts a GeMS-style ArcGIS geodatabase to two shapefile packages:

- **OPEN--** Consists of shapefiles, additional .dbf files, and pipe-delimited text files. Field renaming is documented in output file *logfile.txt*. This package will be a complete transcription of the geodatabase without loss of any information.

- **SIMPLE**-- Consists of shapefiles alone. Tables Glossary, DataSources, and DescriptionOfMapUnits are joined to selected feature classes within feature dataset GeologicMap, long fields are truncated, and these feature classes are written to shapefiles. Field renaming is documented in output file *logfile.txt*. This package is a partial (incomplete) transcription of the geodatabase, but will be easier to use than the OPEN package.

Output is written to directories DBName-simple and DBName-open, where DBName is the name of the input geodatabase, without gdb or mdb suffix. If these directories already exist, any files within them will be deleted.

Parameter	Explanation	Data Type
Input_geodatabase	An existing geodatabase. May be a file (.gdb) or personal (.mdb) geodatabase.	Workspace
Output_workspace	Must be an existing folder. Output folders DBName-open and DBName-simple will be written here, as well as temporary geodatabase xxDBName.	Folder

### **Validate Database** *GeMS\_ValidateDatabase\_Arc10.py*

**Validate Database** audits a geodatabase for conformance with the GeMS schema and reports compliance as “may be LEVEL 1 COMPLIANT”, “is LEVEL 2 COMPLIANT”, or “is LEVEL 3 COMPLIANT”. It also runs mp (metadata parser) to check for formal errors in geodatabase-level FGDC metadata. Note that qualify as LEVEL 2 or LEVEL 3 compliant a database must also be accompanied by a peer-reviewed geologic names report.

Compliance criteria are:

#### **Level 1:**

- No overlaps or internal gaps in map-unit polygon layer

- Contacts and faults in single feature class
- Map-unit polygon boundaries are covered by contacts and faults lines

Databases with a variety of schema may meet these criteria. **Validate Database** cannot confirm LEVEL 1 compliance.

## Level 2:

- 2.1 Has required elements: nonspatial tables DataSources, DescriptionOfMapUnits, GeoMaterialDict; feature dataset GeologicMap with feature classes ContactsAndFaults and MapUnitPolys
- 2.2 Required fields within required elements are present and correctly defined
- 2.3 GeologicMap topology: no internal gaps or overlaps in MapUnitPolys, boundaries of MapUnitPolys are covered by ContactsAndFaults
- 2.4 All map units in MapUnitPolys have entries in DescriptionOfMapUnits table
- 2.5 No duplicate MapUnit values in DescriptionOfMapUnit table
- 2.6 Certain field values within required elements have entries in Glossary table
- 2.7 No duplicate Term values in Glossary table
- 2.8 All xxxSourceID values in required elements have entries in DataSources table
- 2.9 No duplicate DataSources\_ID values in DataSources table

## Level 3

- Table and field definitions conform to GeMS schema
- All map-like feature datasets obey topology rules. No MapUnitPolys gaps or overlaps. No ContactsAndFaults overlaps, self-overlaps, or self-intersections. MapUnitPoly boundaries covered by ContactsAndFaults
- No missing required values
- No missing terms in Glossary

- No unnecessary terms in Glossary
- No missing sources in DataSources
- No unnecessary sources in DataSources
- No map units without entries in DescriptionOfMapUnits
- No unnecessary map units in DescriptionOfMapUnits
- HierarchyKey values in DescriptionOfMapUnits are unique and well formed
- All values of GeoMaterial are defined in GeoMaterialDict. GeoMaterialDict is as specified in the GeMS standard
- No duplicate `_ID` values
- No zero-length or whitespace-only strings

**Validate Database** checks for schema extensions: are there tables, feature datasets, feature classes or fields that are not defined by the standard?

**Validate Database** lists contents of tables DataSources, DefinitionOfMapUnits, Glossary, and (if present) MiscellaneousMapInformation in human-readable form.

**Validate Database** also inventories the database and reports the number of rows, fields, and field definitions for all tables and feature classes.

Output is written to several files in *Output\_workspace*: *Input.gdb-Validation.html*, *Input.gdb-ValidationErrors.html*, *Input.gdb-vFgdcMetadata.txt*, *Input.gdb-vFgdcMetadata.xml*, and *Input.gdb-vFgdcMetadataErrors.txt*. Topology errors are recorded in *Input\_Validation.gdb*.

If ArcMap is open, any joins--e.g., MapUnitPolys to DescriptionOfMapUnits--may need to be removed. If, when running this script from ArcMap, it fails to inventory some feature classes, try running it from ArcCatalog. If MapUnitPolys, ContactsAndFaults, or a similar feature class in another feature dataset participates in a relationship class--e.g., feature-linked annotation--ArcGIS may crash when the script attempts to copy the feature class into the Validation gdb to check topology.

Note that using this script with a geodatabase with a schema that differs significantly from GeMS may not yield a useful report.

Parameter	Explanation	Data Type
Input_geodatabase	A file geodatabase (.gdb). The .gdb extension must be included.	Workspace
Output_workspace (optional)	A directory that must exist and be writable. If no directory is specified, defaults to host directory for <i>Input_geodatabase</i> .	Folder
Refresh GeoMaterialDict	Databases built with earlier versions of the GeMS toolbox will generate numerous errors associated with GeoMaterialDict and GeoMaterial values. Check this box to replace the GeoMaterialDict table in the database with the current version. The GeoMaterials domain (available as a picklist while editing the DescriptionOfMapUnits table) is also replaced. <b><i>This option permanently modifies the geodatabase. Perhaps you should back it up before using this option.</i></b>	Boolean
Skip topology checks	If checked, potentially time-consuming topology checks will be skipped and database will FAIL level 2 and level 3 compliance checks. This may be useful when testing for other aspects of compliance with the GeMS schema.	Boolean
Delete unused Glossary and DataSources rows	Automatically delete any rows in Glossary and DataSources that describe Terms and Sources that are unused elsewhere in the database. If deleted rows have missing required values these still show up as errors. Rerun the Validate Database script to clear such errors. Deletion of a Glossary row may render a DataSource row unneeded. Rerun the Validate Database script to discover such errors. <b><i>This option permanently modifies the geodatabase. Perhaps you should back it up before using this option.</i></b>	Boolean



## Create and attribute geologic lines efficiently

Several colleagues have voiced concern that the GeMS schema could require additional work to set multiple attributes for each geologic line. This concern is significant: we do not want to make the generation of geologic maps more difficult. However, such additional work may be avoided by at least 3 different work-flows.

*Workflow 1* is the least satisfactory option. Ignore the suggestion in the GeMS schema documentation that “contact, certain” and “contact, approximate” should both be Type = ‘contact’, with different values of LocationConfidenceMeters.

- Define (in the Glossary table) Type values of “contact, certain”, “contact, approximate”, “contact, concealed”, etc. and use the GeMS schema as if it were ALACARTE. The Glossary definitions should include estimated values for LocationConfidenceMeters

Users of the database, or the author, can later re-calculate Type values and assign appropriate values of LocationConfidenceMeters. We suggest avoiding Type = “contact, queried” because this creates ambiguity about whether the contact is queried because ExistenceConfidence is uncertain, IdentityConfidence is uncertain, or both.

*Workflow 2* is especially appropriate to ArcGIS 9.3 and earlier.

- Add a temporary item (e.g., LTYPE) to the ContactsAndFaults feature class and populate this item with ALACARTE-like values of “contact, certain”, “contact, approximate”, “contact, concealed”, and “contact, queried”. It may be desirable to create a domain for the LTYPE field so that a drop-down list is available
- Symbolize the in-process map according to values of LTYPE. Type, IsConcealed, LocationConfidenceMeters, ExistenceConfidence, IdentityConfidence, and Symbol attributes are left empty

- At the end of each digitizing session, or after the map is completely digitized, use tool **Attribute By Key Values** to bulk-calculate appropriate values for the empty attributes

*Workflow 3* is recommended for ArcGIS 10.0 and subsequent versions.

- Add a temporary item (LTYPE) to ContactsAndFaults
- Set up symbolization for common assemblages of attributes (e.g., at 1:24,000 scale, LTYPE = “contact, approximate” : Type = “contact”, IsConcealed = “N”, LocationConfidenceMeters = 60, ExistenceConfidence = “certain”, LocationConfidence = “certain”, Symbol = “01.01.03”
- Create feature templates for these assemblages

Lines digitized using these templates will automatically have attributes set to appropriate values. The ArcGIS layer file ContactsAndFaults24K.lyr, located in the Resources folder, provides an example.

We suggest two modifications to this. First, instead of using ALACARTE-like LTYPE values of “contact, certain” and “contact, approximate”, use LTYPE values such as “contact 10m” and “contact 100m”. Second, in your .mxd file, create a ContactsAndFaults layer group that symbolizes these arcs in four different ways:

1. by LTYPE (or Symbol), using the appropriate FGDC symbol
2. By LocationConfidenceMeters:

Add ContactsAndFaults to the Table of Contents a second time.

Right-click on TOC entry for the new ContactsAndFaults layer and select Properties... Set the following properties:

Symbology tab

Show:

Quantities > Proportional symbols

Fields

Value: LocationConfidenceMeters

Normalization: none

Unit: Meters

Data represents

Distance from Center

Base Symbol

Double-click and set to a simple dark gray or black line. Set width = 0.5 or similar

General tab

Layer Name: "ContactsAndFaults—LocationConfidence"

"Visible" should be checked

Display tab

Transparent: 50%

"Scale symbols when a reference scale is set" should be checked

Click OK

3. by ExistenceConfidence

4. by IdentityConfidence

The last three may be turned off during routine digitizing but turned on when proofing one's work. Add the layer file ContactsAndFaults24K.lyr (in the Resources folder) to your ArcMap composition and set the Source property of each of the 4 sublayers to your ContactsAndFaults feature class to see an example of such symbolization.