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

Version of 6 January 2018

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.

Installing the toolbox

To install this toolbox, unzip (if necessary) and place the folder *GeMS_Tools* 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* 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 NCGMP09 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 Toolbox was written in Python 2.7 by Ralph Haugerud, Evan Thoms, and others. Scripts have been developed and tested on Windows 7, using various versions of ArcGIS (9.1 - 10.5).

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. **Validate Database** could be rewritten for clarity and efficiency.

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@flagmail.wr.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, clone the dev branch, make your modifications, and submit a pull request.

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_Toolbox/Scripts*.

(re)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. **(re)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 _ID 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 _ID 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 _ID values.	Boolean
Do_not_reset_DataSource_IDs	If unchecked, resets values of DataSources_ID and all DataSourcesID, LocationSourceID, AnalysisSourceID	Boolean

	and similar that refer to DataSources_ID. Default is checked, which leaves these values unchanged.

.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/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, UnitLabl in the map manuscript is related to field Label in the database table DescriptionOfMapUnits. If there is no match, UnitLabl 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 UnitLabl is used to fill both MapUnit and Label fields in DescriptionOfMapUnits.

Values of UnitLabl 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 NCGMP09 format, and for using NCGMP09 to digitize in single-attribute mode.

Many geologic map database schemas have characterized features--especially lines--with a single attribute such as "contact--approximate". NCGMP09 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

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_Toolbox\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.

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.

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)	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 • Define this element from scratch in ArcCatalog, or • 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.	Multiple Value
Number_of_cross_sections	An integer in the range 0 to 26.	Long
Enable_edit_tracking	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. This parameter is ignored if the installed version of ArcGIS is less than 10.1.	Boolean
Add_fields_for_cartographic_ representations	Default is unchecked. 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). 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 NCGMP09 toolbox directory.	Boolean
Add_LTYPE_and_PTTYPE	If checked, adds LTYPE field to ContactsAndFaults and GeologicLines and adds PTTYPE field to OrientationPoints. Useful for digitizing purposes, or for	Boolean

Add_standard_confidence_values	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. Default is checked. If checked: 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.	Boolean

Significant dependencies:

- *GeMS_definitions.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:

- ... bold
- ... italic

- <g> ... </g> FGDCGeoAge font
- ^{...} superscript
- _{...} subscript
-
  break (start new paragraph)

Parameter	Explanation	Data Type
Source_geodatabase	An existing NCGMP09-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 *MapUnit* or *Label* can be used for UnitLabl. Default (use *MapUnit*) is recommended for constructing and proofing the DMU table. It may be useful to use *Label* (box unchecked) for creating a final MSWord file of the DMU.	Boolean
List_of_Map_Units	If checked, creates a **List of Map Units** 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 an Microsoft Word document that uses the paragraph styles defined in USGS Pubs template *MapManuscript_v1-0_04-11.dotx*

FGDC CSDGM2 Metadata *GeMS_MetadataCSDGM2_Arc10.py*

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

Writing CSDGM2-style metadata for a geologic-map geodatabase presents several challenges:

1. Dataset citation, description, status, spatial domain, keywords, point of contact, data quality, spatial reference framework, and distribution information should be correct and useful.

2. Entity descriptions and entity-attribute information should be correct and useful.

3. Similar, but not identical, metadata records should be created for the geodatabase as a whole and each element (table, feature dataset, and feature class) within the geodatabase.

4. Metadata should somehow depict the composite nature of a geologic-map geodatabase which typically contains several nested datasets. CSDGM2 is most apt for a single dataset.

5. Metadata should be formally correct.

This script addresses challenges 2, 3, 4, and 5. Entity descriptions and entity-attribute information are supplied from the NCGMP09 documentation and the information contained within geodatabase tables DescriptionOfMapUnits, Glossary, and DataSources. You complete a single master metadata record (that for GeologicMap) and other metadata records are constructed by the script. The composite nature of the database is briefly described in a Supplemental Information statement that resides within the Description branch of Identifying Information.

Because construction of metadata is scripted AND final records are passed through ArcGIS, they should be formally correct, though you might check this with the mp data parser.

You are still responsible for meeting challenge 1. To use this tool,

- Run script Validate Database to ensure that the database is complete and there are no missing DMU, Glossary, or DataSources entries.
- In ArcCatalog, go to Customize>Options>Metadata and set Metadata Style to "FGDC CSDGM Metadata". OK and exit.
- In ArcCatalog, use the ArcGIS metadata editor to complete the record for the GeologicMap feature dataset. Save. NOTE THAT whatever errors or you create in this master metadata record will be faithfully propagated to metadata records for all parts of the geodatabase!
- Run this script from the Windows command line.
- Look at file <geodatabasename>-metadataLog.txt to see what parts of which metadata records need to be completed by hand using the ArcCatalog metadata editor. This will occur wherever you extend the database schema beyond the schema outlined in the NCGMP09 documentation. *If you expect to make metadata for many geodatabases that have the same extensions, modify file NCGMP09v11_Definition.py to include descriptions of your extensions.*
- Inspect metadata records in ArcCatalog (the Description tab) to see that they are complete.

You want ISO metadata? Change your Metadata Style and fix records using the ArcCatalog metadata editor. Export as ISO of your flavor, insofar as ArcCatalog allows. Let us know how this works.

Parameter	Explanation	Data Type
GeMS-style_geodatabase		Workspace
definitionExtensions (Optional)	Optional: file with extensions to definitions given in GeMS_Definitions.py. Useful for persistent entity and field descriptions, amongst other things. See file	File

my_GeMSDefinitions.py, in Scripts directory of GeMS toolbox, for an example and further directions.

Inclination Numbers *GeMS_InclinationNumbers_Arc10.py*

Creates annotation feature class OrientationPointLabels with dip and plunge numbers for appropriate features within OrientationPoints. Adds the new annotation 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 an 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. If checked, adds rules that involve MapUnitPolys feature class (no gaps, no overlaps, boundaries must be covered by ContactsAndFaults).	Boolean

	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.	

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 specifiedoutput 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 dateset of an NCGMP09-style geodatabase, a cross-section line (feature class, feature layer, or selection), and a DEM. The cross-section line need not be straight.

Final ouput 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_Csxx*InputFeatureClass***, where xx is Output_name_token. Existing feature classes with these names will be overwritten.

This script also creates empty NCGMP09 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
NCGMP09-style_geodatabase	An existing NCGMP09-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.	Boolean

	If unchecked, you should specify which feature classes should be projected with the Feature_classes_to_Project parameter	
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 *LocateFeaturesAlongRoutes*.	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 CrossSection*Output_name_token*. If this feature dataset does not exist it will be created. Output feature classes will be named ed_CS*Output_name_tokenInput_featureclass_name* 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_PTTYPE	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 NCGMP09-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	Feature Dataset

	can then be copied and pasted into the existing map geodatabase.	
Output_feature_class_name	Name of the output featureclass. Any existing featureclass with this name will be overwritten. 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	String
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

Set PlotAtScale values *GeMS_SetPlotAtScales_Arc10.py*

Sets values of item PlotAtScale so that a definition query

[PlotAtScale] >= 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 NCGMP09 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	The value of LocationConfidenceMeters, converted to mm on the map, at which lines change from "certain" (continuous) to "approximate" (dashed). If map scale denominator is 24,000 and the certain to approximate threshold is 1.0, lines with LocationConfidenceMeters <= 24 will have continous-	Double

	line symbols. If map scale denominator is 100,000 and the certain to approximate threshold is 2.5, lines with LocationConfidenceMeters <= 250 will have continuous-line symbols.	
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 standard (not directly observed), most geologic lines are inferred, and this is not common usage. Check this box to use "inferred" symbols for lines with LocationConfidenceMeters greater than a threshold value.	Boolean
Approximate_to_inferred_-threshold_mm_on_map (Optional)	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 > 98 will be drawn with short dashes.	Double
Use_approximate_strike-and-dip_symbols	Default is checked. Uncheck to avoid use of open-center symbols for approximately-oriented bedding.	Boolean
OrientationConfidenceDegrees_-threshold	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 > threshold value, the approximate symbol is assigned.	Double
Set_polygon_symbols_and_labels	If checked, *Symbol* and *Label* values will be calculated for MapUnitPolys, using values from table DescriptionOfMapUnits. Default is checked.	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_Toolbox/Scripts* directory

Topology Check *GeMS_TopologyCheck_Arc10.py*

Topology Check evaluates topological aspects of feature datasets in a GeMS-style geologic map geodatabase. It does not substantially alter the input geodatabase. Output is a new geodatabase *<inGeodatabase>_errors.gdb* and an HTML file *<inGeodatabase>_topologyReport.html*

For each feature dataset evaluated the output geodatabase will contain a feature dataset of the same name that contains feature classes which identify possible errors of various kinds.

Note that not all possible topological errors will be identified by this script. Some "errors" flagged by this script may be acceptable. Use your discretion!

By default only the GeologicMap feature dataset is evaluated.

The **Nodes** and **MapUnit adjacency** modules use functions *isFault* and *isContact*. 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
Geodatabase	Input Geodatabase. Note that code was developed with file geodatabases (.gdb) and you may encounter bugs with a personal geodatabase (.mdb).	Workspace
Validate_topology_of_all_-feature_datasets (Optional)	Default is unchecked (false). Check to evaluate topology of all feature datasets within input geodatabase. Evaluation of a feature dataset that lacks an identifiable *ContactsAndFaults* feature class will trigger an error.	Boolean
Validate_topology_of_these_-feature_datasets (Optional)	Select specific feature datasets to evaluate. Note that if no feature datasets are selected and Validate topology of all feature datasets is unchecked the GeologicMap feature dataset will be evaluated by default. Evaluation of a feature dataset that lacks an identifiable *ContactsAndFaults* feature class will trigger an error. Checking "Validate topology of all data sets" overrides any selection made here.	Multiple Value

Line_and_polygon_topology	Default is checked (true). Check to create and validate an ArcGIS topology class for each feature dataset. This class will include the following rules: • Must Not Overlap (Line) *xxxContactsAndFaults* • Must Not Self-Overlap (Line) *xxxContactsAndFaults, xxxGeologicLines* • Must Not Self-Intersect (Line) *xxxContactsAndFaults, xxxGeologicLines* • Must Be Single Part (Line*) xxxContactsAndFaults, xxxGeologicLInes* • Must Not Overlap (Area) *xxxMapUnitPolys* • Must Not Have Gaps (Area) *xxxMapUnitPolys* • Boundary Must Be Covered By (Area-Line) *xxxMapUnitPolys / xxxContactsAndFaults* Note the lack of a rule *Must Not Have Dangles (Line)*. This is because dangling fault lines (and dangling concealed fault or contact lines) are common in geologic maps and are not errors. Dangling contacts (which are errors) are identified by the **Nodes** option. Any identified errors are written to feature classes *errors_xxxTopology_line, errors_xxxTopology_point*, and *errors_xxxTopology_poly*.	Boolean
Nodes	Geologic map-logic imposes another set of topological constraints on nodes (line junctions) in the ContactsAndFaults feature class. • More than 4 lines never join at a node. • Four lines join at a node only if two opposite lines are contacts (not concealed) and the other lines are both of the same Type and one or both is concealed. *This explicitly disallows the possibility of faults that do not offset contacts.* • Where three lines join, either none are concealed or all are concealed. An exception is where two of the lines are the map boundary.	Boolean

	• There should be no pseudonodes (junctions between only two lines where both lines have same values for Type, LocationConfidenceMeters, ExistenceConfidence, IdentityConfidence, and DataSourceID). Nodes with two lines where the lines are identical—i.e., the line joins itself (forming a closed loop) are OK. • Nodes with one line are dangles. Dangles are permissible where the line is a fault or the line is concealed. Note that some fault dangles are errors. To evaluate conformance with these rules, the input ContactsAndFaults feature class is first planarized, that is, all lines are broken at intersections. Nodes that do not meet these constraints are written to output feature class *errors_xxxBadNodes*.	
Fault_directions	Fault lines that are contiguous should have the same direction. This, for example, allows thrust teeth to consistently be on same side. Note that this routine may not identify some direction errors where three fault lines meet. Fault lines are identified with the SQL query "TYPE" LIKE '%fault%' Nodes where fault direction changes are written to feature class *errors_xxxFaultDirNodes*.	Boolean
MapUnit_adjacency	Creates tables, in output HTML file, of adjacent map units for three classes of lines: • Concealed contacts and faults (*IsConcealed = 'Y')* • Contacts (not concealed) (*isContact* and *IsConcealed = 'N*') • Faults (not concealed) (*isFault* and *IsConcealed = 'N'*) These tables can be useful for identifying mis-tagged polygons and lines. Table **Faults (not concealed)** may be helpful in identifying fault lines with the wrong direction.	Boolean

	Identifying information for concealed lines that separate polygons with different MapUnit values and non-concealed contacts that separate polygons with identical MapUnit values is written to HTML tables **Bad concealed contacts and faults** and **Internal contacts**.	
Identify_duplicate_point_-features	Scans each point feature class and identifies those features that: • have the same location, and • have the same *Type* values, and • if attributes *Azimuth* and *Inclination* are present, have the same *Azimuth* and *Inclination* values Any duplicates are listed in tables *dups_XXXXXX.*	Boolean
Identify_short_lines_small_-polys_and_sliver_polys	Scans ContactsAndFaults and MapUnitPolys feature classes to identify lines that are shorter than a threshold value, polygons that are smaller than a threshold value, and polygons that have area-circumference ratios (polygon width) that are smaller than a threshold value. Map scale denominator (below) must be set correctly! Identified features are written to feature classes *errors_xxxShortArcs*, *errors_xxxSkinnyPolys*, and *errors_xxxSmallPolys*.	Boolean
Map_scale_denominator	Integer. Value used to identify too-short arcs, too-small polys, and too-skinny polys.	Long
minimum_line_length__mm	Threshold value, in millimeters at map scale, used to identify too-short arcs.	Double
minimum_poly_area__sq_mm	Threshold value, in square millimeters at map scale, used to identify too-small polygons.	Double
minimum_poly_width__mm	Threshold value, in millimeters at map scale, used to identify sliver polygons.	Double

force_exit_with_error	Default is unchecked (false). When checked, forces an error upon normal completion of script. Useful when debugging, as it returns focus to the script window while preserving all input values.	Boolean

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 DMName-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. Checks include:

- Schema errors:
  - Are required tables, feature datasets, and feature classes present?
  - Are required fields present in required and optional tables and feature classes?
  - Are required fields defined properly?
- Schema extensions: are there tables, feature datasets, feature classes or fields that are not defined by the standard?
- Is required content present?
- Are _ID values unique?
- Are all Source values defined in table DataSources?
- Are there entries in DataSources which are unused elsewhere in the database?
- Does table Glossary contain all required definitions?
- Are there terms in table Glossary which are unused elsewhere in the database?
- What map units are present on the map, in DescriptionOfMapUnits, in CorrelationOfMapUnits, and cross sections? Do they all match?
- Are there any extra units in DescriptionOfMapUnits?
- Are HierarchyKey values in DescriptionOfMapUnits properly formatted?
- Are there any pseudonulls (single-space values) or trailing spaces in text fields?

Validate Database also inventories the database and reports the number of rows, fields, and field definitions for all tables and feature classes. For more information, check tail of file *GeMS_ValidateDatabase_Arc10.py*.

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	Can be a file geodatase (.gdb) or a personal geodatabase (.mdb). .gdb or .mdb extension must be included.	Workspace
Output_workspace	A directory that must exist and be writable.	Folder

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.