**Attribute Rules and Dictionary Symbology in ArcGIS Pro Help Streamline Geologic Map Compilation in GeMS**

Suggested Citation:
Felger, T.J., 2023, Attribute rules and dictionary symbology in ArcGIS Pro help streamline geologic map compilation in GeMS — Digital Mapping Techniques 2020, June 8-10, 2020 – A Virtual Event, Pamphlet: 15 pages, at https://ngmdb.usgs.gov/Info/dmt/docs/DMT20_Felger_AttributeRules_Pamphlet.pdf.

Tracey J. Felger
U. S. Geological Survey
2255 N. Gemini Dr
Flagstaff, AZ 86001
tfelger@usgs.gov

Attribute rules and dictionary symbology are functionality available in ArcGIS Pro software by Environmental Systems Research Institute (Esri) that can dramatically streamline the compilation of geologic maps in the *Ge*ologic *M*ap *S*chema (GeMS). GeMS is a standard for the digital publication of geologic maps developed by the U.S. Geological Survey (USGS) National Cooperative Geologic Mapping Program (NCGMP) (USGSNCGMP, 2020). NCGMP recently mandated that Federal (FEDMAP) and State (STATEMAP) geologic mapping projects that are funded by NCGMP submit their deliverables in GeMS format.

GeMS was primarily designed to digitally encode the geologic and cartographic content presented on a single traditional geologic map. A secondary objective was to provide a stepping-stone toward the development of multimap databases, specifically the National Geologic Map Database (NGMDB; https://ngmdb.usgs.gov/ngmdb/ngmdb_home.html), which serves as a national archive of standardized geologic map content. The schema not only captures geologic attributes, but also includes attributes for cartography, and feature-level metadata. The purpose of feature-level metadata is to document the lineage, quality, and status of each feature. This information is important when visually comparing features to other data, evaluating the results of analytical operations, or combining features with other datasets to produce derivative data (such as multimap databases). The cartography and metadata attributes prescribed by GeMS are worthwhile; however, they also increase the data compilation workload.

Geologic map databases commonly include the following layers, regardless of the software platform or database used to compile the data: contacts and faults (lines), map units (polygons), and structural measurements (points). Comparing the GeMS schema for these layers with that of an ALACARTE database (Fitzgibbon, 1991; Fitzgibbon and Wentworth, 1991; Wentworth and Fitzgibbon, 1991), which was the schema used by many western region USGS FEDMAP projects during the 1990s, and still used by some today, shows that an ALACARTE contacts and faults layer had one attribute field versus GeMS, which has nine required fields. ALACARTE layers for map units had one field, and structural measurements used three fields, compared to GeMS, which has six and twelve required fields, respectively (Felger, 2020, slide 5).

The increased workload associated with populating and maintaining the additional fields required by GeMS has been one of the obstacles to users adopting the schema. Another obstacle is that GeMS also represents a paradigm shift in the way geologists think about geologic data. GeMS represents a 'granular', data-driven approach as opposed to the traditional cartographic approach employed by most geologic map schemas. Granular databases are ones in which the information is disaggregated into the smallest pieces possible and is stored in many fields. Attribute rules and dictionary symbology, available in Esri's ArcGIS Pro, can help lessen the workload and facilitate transitioning to the new paradigm that is inherent to GeMS.

Attribute rules are user-defined scripts embedded within a geodatabase that are either triggered automatically by editing operations (such as adding or modifying a feature) or they can be run on-demand. They are similar to formulas and macros in Microsoft Excel or triggers and stored procedures in SQL. The scripts are written in Arcade, which is a light-weight scripting language similar to JavaScript, but specific to the Esri platform (https://developers.arcgis.com/arcade/). Attribute rules can be used to automatically populate attributes (calculation rules), restrict invalid edits during edit operations (constraint rules), and perform quality assurance checks on existing features (validation rules).

Dictionary symbology creates symbols on-the-fly, based on one or more attributes of a feature, and optional logic statements. Dictionary symbology can be thought of as symbology 'Legos'. A typical Legos kit comes with an assortment of blocks of various sizes, shapes, and colors, and includes instructions for assembling the blocks into different objects, such as a house, helicopter, or dinosaur. Similarly, a symbol dictionary includes symbol components (such as a solid black line of a certain weight, and a question mark of a certain font size and style) and an embedded dictionary of rules (instructions) that control how the components are aggregated to visually represent the data. The symbol components and rules are stored in a mobile style file (.stylx), which is an SQLite database. The rules are user-defined scripts written in Arcade. When a symbol dictionary is applied to a feature layer, the feature attributes are evaluated by the rules that are embedded in the style file, and the appropriate symbol is 'constructed' and applied to the feature. For example, if a line feature has a Type value of 'fault', and IdentityConfidence and ExistenceConfidence are 'certain', then a solid black line with a width of .375 mm will be used to symbolize the line; however, if IdentityConfidence or ExistenceConfidence are 'questionable', then a question mark will be added to the line symbol. There are several advantages to this approach, compared to unique value symbology which is typically used to symbolize geologic map data (specifically contacts and faults). One is that there is no limit to the number of fields that can be used to symbolize a feature; in contrast, symbolizing on unique values in many fields in Esri's ArcMap or ArcGIS Pro is currently limited to three fields (https://pro.arcgis.com/en/pro-app/latest/help/mapping/layer-properties/unique-value.htm). Another advantage is that the symbolization is not restricted to unique attribute values but can also apply logic to the attributes to further refine the composite symbol. For example, dash length and gap can be varied as function of LocationConfidenceMeters, as described later in the document (Felger, 2020, slide 18).

I developed and applied attribute rules and dictionary symbology to one of my active FEDMAP geologic mapping projects to test whether the functionality would help streamline compilation of a GeMS database, as well as facilitate a transition to editing in a 'native' GeMS environment. Testing was focused on the ContactsAndFaults and MapUnitPolys feature classes, which are the two feature classes required by the GeMS schema (USGSNCGMP, 2020, figure 1). I extended the GeMS schema for these feature classes by adding Global IDs, which are software-maintained unique identifiers, and are required for utilizing attribute rules (https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/an-overview-of-attribute-rules.htm). I further extended the GeMS schema for ContactsAndFaults by adding field GeMS_Ltype (text, 254). This field stores a string that is derived by concatenating the values in the Type, IsConcealed, IdentityConfidence, and LocationConfidenceMeters fields. For my purposes, I chose to treat IdentityConfidence equal to ExistenceConfidence. I developed the dictionary symbology for ContactsAndFaults by modifying the ArcMap style file developed by the Geological Survey of Canada (FGDC_GSC_20100414, available from the NGMDB GeMS Resources; https://ngmdb.usgs.gov/Info/standards/GeMS/). The attribute rules and dictionary symbology that I created currently support ~215 contact and fault types, including all the contacts in the Federal Geographic Data Committee (FGDC) Digital Cartographic Standard for Geologic Map Symbolization (FGDC, 2006) appendix A-1-1, and all the faults in appendix A-2-1 through A-2-10. Initial development and testing were done in Esri's ArcGIS Pro 2.4, and subsequently in ArcGIS Pro 2.8.1.

The attribute rules that I developed for ContactsAndFaults support two compilation workflows. The goal was to improve compilation efficiency and data quality, while at the same time providing users with the option to work in a familiar, cartography-driven environment (traditional), or the more data-driven (granular) approach prescribed by GeMS. I specifically wanted to eliminate the need to run the GeMS 'Reset ID Values' and 'Attribute By Key Values' scripts. In both workflows, I employ attribute domains, default values, and feature templates in conjunction with the attribute rules to streamline data entry and minimize attribution errors. I also symbolize ContactsAndFaults twice – once with the appropriate geologic line symbol, and a second time by buffering the line based on LocationConfidenceMeters – so that I can see a graphic representation of the spatial extent of the LocationConfidenceMeters value (Felger, 2020, slide 21).

The first workflow that I developed and tested for ContactsAndFaults uses the Symbol field as the 'parent' field from which the attributes for several other GeMS fields are derived. This method, which I refer to as the traditional workflow, provides an intuitive way for the person compiling the map to digitize features while minimizing the number of fields that must be populated by hand, or programmatically with a tool such as the GeMS 'Attribute By Key Values' script. When a line is digitized, the Symbol field is populated with a code from the FGDC Digital Cartographic Standard for Geologic Map Symbolization (FGDC, 2006) in zero padded format (e.g. the Symbol value for Contact—Identity and existence certain, location accurate is 01.01.01). Calculation rules are triggered upon creation or modification of a feature such that 'dependent' fields are populated or updated based on the value in the 'parent' field. In this case, the 'dependent' fields are Type, IsConcealed, ExistenceConfidence, and IdentityConfidence. So, if the map compiler populates Symbol with 01.01.01 (Contact—Identity and existence certain, location accurate) then Type is set to ‘Contact’, IsConcealed is set to ‘N’ (No), ExistenceConfidence is set to ‘certain’, and IdentityConfidence is set to ‘certain’. If the compiler changes the Symbol value to 01.01.08 (Contact—Identity or existence questionable, location concealed), the dependent values will automatically be updated so that Type is set to ‘Contact’, IsConcealed is set to ‘Y’ (Yes), ExistenceConfidence is set to ‘questionable’, and IdentityConfidence is set to ‘questionable’. This eliminates the need to run the GeMS ‘Attribute By Key Values’ script. The field GeMS_Ltype, which is populated by concatenating the values in the Type, IsConcealed, IdentityConfidence, and LocationConfidenceMeters fields, is similarly maintained by a calculation rule. A calculation rule also automatically populates ContactsAndFaults_ID, which eliminates the need to run the GeMS ‘Reset ID Values’ script. Finally, constraint rules are attached to LocationConfidenceMeters and DataSourceID requiring that those fields are populated (not null) before features can be saved (Felger, 2020, slides 11-13).

I refer to the second workflow that I developed for digitizing ContactsAndFaults as the granular workflow. This workflow is data-centric and requires the compiler to populate five primary ‘component’ fields. Attribute rules are used to calculate four ancillary field values and to ensure that features are not created or cannot be saved if certain fields have null values. In this case, the compiler populates Type, IsConcealed, LocationConfidenceMeters, ExistenceConfidence, and DataSourceID. Constraint rules ensure that LocationConfidenceMeters and DataSourceID are not null when a feature is created or modified. Calculation rules automatically populate ContactsAndFaults_ID, IdentityConfidence, Symbol, and GeMS_Ltype (Felger, 2020, slide 21). This workflow is used in conjunction with dictionary symbology.

For MapUnitPolys, I developed attribute rules to support two different methods of creating polygons. In the first method, polygons are created interactively while digitizing. The compiler uses the ‘Construct Polygon’ tool in Esri’s ArcGIS Pro to select the lines in the ContactsAndFaults layer that will act as the boundaries of the polygon to be created, and then chooses the appropriate MapUnitPolys feature template to apply to the polygon. In this method, there must be at least one polygon feature template that has values for MapUnit, IdentityConfidence, and DataSourceID, otherwise the tool will produce an error, and the polygon will not be created. The second method uses either the ‘Feature to Polygon’ geoprocessing tool or the GeMS ‘Make Polygons’ tool to create polygons in bulk from ContactsAndFaults and an optional point feature class that has a MapUnit field, and that serves as the ‘label’ point for the polygon. This method, which mimics the way in which polygons were created in coverages in Esri’s workstation ArcInfo, is still widely used by compilers. In the first method, attribute rules are attached to the MapUnitPolys feature class, whereas in the second method they are attached to a ‘label’ point feature class that has a MapUnit field. The GeMS ‘Create New Database’ script includes the option to create a MapUnitPoints feature class; however, the purpose of this feature class is to capture the location of very small map-unit extents that are too small to show as polygons – not to serve as ‘label’ points that are used to build MapUnitPolys. If the compiler wishes to build polygons using the second method, they must create a MapUnitLabelPoints feature class by hand, and that feature class should have the same fields as MapUnitPolys, including a MapUnitPolys_ID field. Either method of creating polygons can be used regardless of whether the compiler chooses the traditional or granular workflows for digitizing ContactsAndFaults.

The attribute rules that are applied to the MapUnitPolys/LabelPoints feature class include constraint rules and calculation rules. The constraint rules are that MapUnit, IdentityConfidence, and DataSourceID cannot be null. The calculation rules automatically populate MapUnitPolys_ID (eliminating the need to run the 'Reset ID Values' script), and Label and Symbol are automatically populated based on the values in MapUnit and IdentityConfidence (Felger 2020; slide 14). For example, if MapUnit is 'Tv', and IdentityConfidence is 'certain', Label and Symbol will be set to 'Tv'; however, if MapUnit is 'Tv', and IdentityConfidence is 'questionable', Label and Symbol will be set to 'Tv?'. These calculation rules could be modified to accommodate more complex scenarios, such as incorporating the FGDCGeoAge font symbols into Label (e.g. '^' for Triassic), or populating Symbol with a user-assigned FGDC code from the CMYK color chart (e.g. Symbol is set to 9 for 'Tv').

The functionality described above accomplishes the same tasks as the GeMS 'Attribute By Key Values' and 'Reset ID Values' scripts; however, the attribute rules are automatically triggered as part of the editing process, and do not require the user to periodically run any stand-alone scripts. Thus, the required fields in the attribute table are always fully populated, and the dependent values are always synchronized with the 'parent' value. This not only improves compilation efficiency and data quality, but it means that when the GeMS 'Validate Database' script is run, the report contains fewer extraneous errors, making it is easier for the compiler to identify and fix geologic content errors more efficiently.

Feature symbology is an important component of the digitizing process. In the case of ContactsAndFaults, features can be (and typically are) symbolized by matching the value in the Symbol field to the FGDC style. Dictionary symbology, however, provides a powerful alternative because it can symbolize features based on values in multiple fields, and is also able to employ logic, and so is not limited to unique values. This makes dictionary symbology well-suited for symbolizing lines in the GeMS schema, and it accomplishes some of what the GeMS 'Set Symbol Values' script does, but in a dynamic fashion. The dictionary symbology that I created symbolizes features based on values in the Type, IsConcealed, LocationConfidenceMeters, and ExistenceConfidence fields (Felger, 2020, slide 18). In the case of LocationConfidenceMeters, a line is symbolized as solid, or with dashes of different lengths, based on ranges of values. For example, if LocationConfidenceMeters is less than or equal to 20 m, the line is solid, if LocationConfidenceMeters is greater than 20 m and less than 50 m, it is symbolized with short dashes, and if LocationConfidenceMeters is greater than or equal to 50 m, it is symbolized with long dashes. These ranges can be customized to accommodate different user preferences or different map scales. Furthermore, the user can interactively toggle the display of the LocationConfidenceMeters and ExistenceConfidence symbology on or off as desired. For example, if a feature has attributes of Type = 'Contact', IsConcealed = 'N' (No), LocationConfidenceMeters = '30', and ExistenceConfidence = 'questionable', the line will be symbolized as a short-dashed, contact-weight line with question marks. If the user turns off the display of the LocationConfidenceMeters field, the line will be symbolized as a solid contact-weight line with question marks. If the ExistenceConfidence field is turned off, the line will be symbolized as a solid contact-weight line without question marks. This functionality allows the user to easily produce a more detailed or simplified depiction of the symbology on the fly as needed.

As mentioned previously, both attribute rules and dictionary symbology require scripting in Arcade. This was a steep learning curve for me and may be a deterrent for many users; however, attribute rules can be exported and imported between geodatabases (as .csv files), and dictionary symbology can be shared via mobile style files (.stylx), so that interested users do not need to start from scratch. The attribute rules and dictionary symbology that I have developed thus far, as well as other files to facilitate implementation, are available as supplemental files at https://ngmdb.usgs.gov/Info/dmt/docs/DMT20_Felger_SupplementalFiles.zip. The files include two ArcGIS Pro project packages (one for each workflow) that are pre-configured so that users can experiment with the functionality without having to start from scratch. I recommend starting with the traditional workflow, since it is more intuitive, and probably most similar to workflows that compilers are currently using. Note that the instructions that follow assume that the user is familiar with editing vector data in ArcGIS Pro and the GeMS schema. Double-click the GeMS_attrules_traditional_demo.ppkx to open the project, and modify the attributes of the existing features to see how the attribute rules work. Try the following tests:

1) change the Symbol value for the existing line from 01.01.01 to 02.02.01 and notice how the symbol changes from a contact to a normal fault in the map view, and how the values in the Type and GeMS_Ltype fields change
2) change the Symbol value from 02.02.01 to 02.02.07 and notice how the symbol changes from a solid to a dotted line, and how the values in the IsConcealed and GeMS_Ltype fields change
3) change the Symbol value from 02.02.07 to 02.02.04 and notice how the symbol changes from a dotted line to a dashed line with queries, and how the values in the IsConcealed, ExistenceConfidence, IdentityConfidence and GeMS_Ltype fields change
4) change the LocationConfidenceMeters value from 100 to 50 and note how the green buffer around the line changes
5) for the existing polygon in MapUnitPolys, change the IdentityConfidence value from ‘questionable’ to ‘certain’ and notice how the polygon label changes, and how the values in Symbol and Label change
6) change the value in MapUnit from Tv to Tr and notice how the polygon label changes, and how the values in Symbol and Label change
7) for either the line or the polygon, try changing values in the dependent fields and notice that you cannot apply the change, or that the values revert to the original values
8) try deleting values in required fields, and notice that you get an error message saying the field must be populated

Once you have a feel for modifying attributes for existing features, try digitizing some features from scratch:

1) digitize features in ContactsAndFaults and MapUnitPolys/LabelPoints and notice how fields are automatically populated, and that you get error messages if certain fields are not populated as required
2) digitize a point in the MapUnitLabelPoints_from_Polys feature inside the existing polygon and notice how it inherits the attributes of the polygon that contains it
3) experiment with using the 'Construct Polygon' tool to create polygons interactively from ContactsAndFaults and the MapUnitPolys feature template

To examine the attribute rules associated with each feature class, click on the Catalog view tab, go to the Databases folder, and expand the geodatabase (GeMs_AttRules_U11N83_Traditional.gdb) to the feature class level. Right-click on a feature class, then choose Design>Attribute Rules to access the Attribute Rules interface. If you want to apply the rules to your own geodatabase, follow the instructions outlined in the 'Traditional workflow' section in the appendix.

To experiment with the granular workflow and dictionary symbology, double-click GeMS_attrules_granular_demo.ppkx to open the pre-configured project. This project comes with 30 contact and fault template lines that are stored in the ContactsAndFaultsTemplate feature class and symbolized using dictionary symbology. Toggle the display of the template lines off after you have browsed them and zoom into the existing line and polygon to the north. Experiment with the granular workflow and dictionary symbology by first modifying the attributes of the existing features. Try the following tests:

1) for the existing contact, change ExistenceConfidence to 'questionable', and notice how the line symbol changes in the display, and the values for IdentityConfidence, Symbol, and GeMS_Ltype are automatically updated
2) change the value in Type to 'Fault' and notice the changes in the Symbol and GeMS_Ltype fields
3) change the LocationConfidenceMeters value from 20 to 30 and notice how the solid line changes to a short-dashed line, the green buffer changes width, and the values in the Symbol and GeMS_Ltype fields change
4) change the LocationConfidenceMeters value to 50 and note that the symbol changes to a long-dashed line, the buffer changes width, and the values in the Symbol and GeMS_Ltype fields change
5) change the value in IsConcealed to 'Yes' and note that the line symbol changes to a dotted line
6) change the value in IsConcealed back to 'No'

Keep in mind that even though the value in Symbol is being populated and updated as changes are made to the Type, IsConcealed, LocationConfidenceMeters, and ExistenceConfidence values, the value in Symbol is NOT the value that is being used to symbolize the feature – the symbology is being adjusted and applied dynamically via the dictionary symbology. To look at this more closely, right-click on ContactsAndFaults-SYMBOL in the contents pane and choose Symbology from the drop-down menu. Note that the lines are being symbolized using a symbol dictionary that is being applied to four fields. To experiment with the dictionary symbology, do the following:

1) click on the drop-down arrow next to LocationConfidenceMeters in the Symbology field list, choose 'None' from the list of values, and notice how the line symbol changes from short dashes to a solid line
2) click on the drop-down arrow next to ExistenceConfidence in the Symbology field list, choose 'None' from the list of values, and notice how the question marks are removed from the line
3) do the same thing for the IsConcealed and Type fields and notice how the symbol changes to a solid magenta line – this is the symbol for a line that cannot be rendered by the dictionary. In this case, the Type and IsConcealed fields are primary symbol components, so the fields must be turned on in the dictionary and populated with values in order for the symbol to render. The ExistenceConfidence and LocationConfidenceMeters fields, on the other hand are secondary, and can be toggled on or off if needed. This provides the compiler with an easy way to simplify the line symbology. This can be handy when editing closely spaced lines, or for producing a simplified map or figure.

Once you feel comfortable modifying the attributes of existing features, try digitizing some features from scratch. To examine the attribute rules associated with each feature class, click on the Catalog view tab, go to the Databases folder, and expand the geodatabase (GeMs_AttRules_U11N83_Granular.gdb) to the feature class level. Right-click on a feature class, then choose Design>Attribute Rules to access the Attribute Rules interface. The Arcade script that is embedded in the symbol dictionary is not accessible from within ArcGIS Pro, and instead must be accessed by opening the .stylx file with an SQLite browser. To apply this workflow to your own geodatabase, follow the instructions outlined in the 'Granular workflow' section in the appendix.

The workflows that I have developed and tested were successful at streamlining the compilation of a geologic map in GeMS. These workflows give map compilers the flexibility to choose the methods for digitizing lines and creating polygons that best suit their personal preference, or that best suit the compilation project. Initial testing indicates that the traditional workflow for digitizing ContactsAndFaults is well suited for compiling existing analog maps, whereas the granular workflow, combined with dictionary symbology is ideal for compiling new field mapping. In the case of compiling an existing analog map, the digitizing is likely being done by someone other than the mapper, and attributes such as LocationConfidenceMeters values can only be estimated and applied based on line type (e.g. all lines that are described as 'contact, location accurate' on the source map are assigned a LocationConfidenceMeters value of 20 m). In contrast, the geologist who is compiling new field mapping can use a combination of field observations, GPS data, and high-resolution imagery to customize attributes such as LocationConfidenceMeters for each feature. In that scenario, attribute rules combined with dictionary symbology provide an optimal workflow. The attribute rules and dictionary symbology I developed should prove useful to the wider geologic mapping community; however, they are customized to suit my specific compilation needs so may need to be modified in order to meet the needs of other projects and agencies.

**Acknowledgments**

I would like to thank Dave Soller (USGS) and Mark Yacucci (Illinois State Geological Survey) for their reviews of this manuscript, which greatly improved its readability and detail. Geoff Phelps and Ryan Crow (both USGS) tested the workflows in the appendix and provided feedback that was invaluable in refining the instructions and supplemental files. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

**References**

Federal Geographic Data Committee [prepared for the Federal Geographic Data Committee by the U.S. Geological Survey], 2006, FGDC Digital Cartographic Standard for Geologic Map Symbolization: Reston, Va., Federal Geographic Data Committee Document Number FGDC-STD-013-2006, 290 p., 2 plates, at https://ngmdb.usgs.gov/fgdc_gds/geolsymstd.php.

Felger, T.J., 2020, Attribute rules and dictionary symbology in ArcGIS Pro help streamline geologic map compilation in GeMS—Digital Mapping Techniques 2020, June 8-10, 2020 – A Virtual Event, Presentation: 23 slides, at https://ngmdb.usgs.gov/Info/dmt/docs/DMT20_Felger.pdf.

Fitzgibbon, T.T., and Wentworth, C.M., 1991, revised 2011, ALACARTE user interface - AML code and demonstration maps, version 3.5: U.S. Geological Survey Open-File Report 91-587A, software package, at https://pubs.usgs.gov/of/1991/0587.

Fitzgibbon, T.T., 1991, revised 2011, ALACARTE Installation and System Manual: U.S. Geological Survey Open-File Report 91-587B, 65 p., at https://pubs.usgs.gov/of/1991/0587.

U.S. Geological Survey National Cooperative Geologic Mapping Program (USGSNCGMP), 2020, GeMS (Geologic Map Schema)—A standard format for the digital publication of geologic maps: U.S. Geological Survey Techniques and Methods, book 11, chap. B10, 74 p., at https://doi.org//10.3133/tm11B10.

Wentworth, C.M., and Fitzgibbon, T.T., 1991, ALACARTE User Manual: U.S. Geological Survey Open-File Report 91-587C, 267 p., at https://pubs.usgs.gov/of/1991/0587.

**Appendix. Instructions for configuring a GeMS geodatabase for use with attribute rules and dictionary symbology**

General notes:

1) Attribute rules and dictionary symbology are only available in ArcGIS Pro, not ArcMap. A feature class with attribute rules is not accessible in ArcCatalog or ArcMap
2) You must have ArcGIS Pro 2.4 or later and the GeMS toolbox installed on your computer
3) All attribute rules are enabled by default; however, you can disable or delete rules as desired. Keep in mind that disabling or deleting rules may cause other rules to not work properly
4) All fields that have associated attribute rules are designated as 'Editable', and all attribute rules are set to run on 'Insert' and 'Update'. This means that you can manually change the value in a field that has an associated calculation rule; however, any change you make will be overwritten because the calculation rule will run again after you make the change. If you want to be able to manually change values in fields that have an associated calculation rule, change the rule so that it is only triggered during an 'Insert' operation. Keep in mind that this may produce undesirable results, especially in the Traditional workflow, because 'dependent' values may become out of sync with the 'parent' value
5) Attribute domains are attached to the Symbol, IsConcealed, and Existence/IdentityConfidence fields as part of the traditional workflow, and to the Type, IsConcealed, and Existence/IdentityConfidence fields as part of the granular workflow. The attribute domains that are attached to the Existence/IdentityConfidence fields are automatically added by the GeMS 'Create New Database' script unless the user unchecks the option to 'Add standard confidence values' when creating the geodatabase

Follow the instructions below to import and start using the attribute rules and dictionary symbology on your own geodatabase. The instructions are grouped by workflow (traditional vs. granular). They assume the user is familiar with using ArcGIS Pro and GeMS, so the steps are not described in detail. Refer to the ArcGIS Pro and/or GeMS documentation for additional information.

Traditional workflow (digitizing lines by choosing a line symbol):

1. Download the supplemental files
2. Download and install the GeMS tools for ArcGIS Pro ([https://github.com/usgs/gems-tools-pro](https://github.com/usgs/gems-tools-pro))
3. Open an existing ArcGIS Pro project, or create a new one as needed
4. Create a new GeMS database if you do not already have one
    a. If you want to build MapUnitPolys from ContactsAndFaults and label points, you will need to create a MapUnitLabelPoints feature class by hand. The feature class should have the same fields as MapUnitPolys, including a MapUnitPolys_ID field
5. Add your geodatabase to the Project Databases folder
6. Add Global IDs to ContactsAndFaults and MapUnitPolys/LabelPoints
7. Add field GeMS_Ltype (text, 254) to ContactsAndFaults
8. Attach attribute domains to the Symbol and IsConcealed fields in ContactsAndFaults

    a. Use the ‘Import table to Domain’ geoprocessing tool, navigate to the supplemental files, and attach FGDC_CAF_Symbol.dbf to the Symbol field, and GeMS_IsCon.dbf to the IsConcealed field
    b. Change the merge and split policies from ‘Default’ to ‘Duplicate’, otherwise the attributes will be deleted for lines that you split or merge
9. Add FGDC_ContactsAndFaults_SymbolDict.stylx to the Project Style folder
10. Import attribute rules into ContactsAndFaults and MapUnitPolys/LabelPoints feature classes
    a. In Catalog View, expand the GeologicMap feature dataset so you can see a list of the feature classes
    b. Import Attribute Rules for ContactsAndFaults by doing the following:
        i. Right click on ContactsAndFaults>Design>AttributeRules to access the Attribute Rules interface
        ii. Click on the Import Rules button on the ribbon, navigate to the location of the supplemental files, and choose AttributeRules_ContactsAndFaults_Traditional.csv
        iii. If the import is successful, you will see six Calculation Rules and two Constraint Rules
    c. Import Attribute Rules for MapUnitPolys (or MapUnitLabelPoints) by doing the following:
        i. Right click on MapUnitPolys>Design>AttributeRules to access the Attribute Rules interface
        ii. Click on the Import Rules button on the ribbon, navigate to the location of the supplemental files, and choose AttributeRules_MUPolys_or_Pts.csv
            1. If you build MapUnitPolys interactively as you digitize, import the rules into the MapUnitPolys
            2. If you build MapUnitPolys from ContactsAndFaults and MapUnitLabelPoints, you will import the attribute rules into the MapUnitLabelPoints feature class
        iii. Optional – if you create MapUnitPolys interactively, but also want to maintain a MapUnitLabelPoints feature class, import AttributeRules_MUPtsFromMUPolys.csv into your MapUnitLabelPoints feature class – these rules automatically assign the attributes of the underlying polygon to the point when you digitize the point
    d. Add ContactsAndFaults, MapUnitLabelPoints and/or MapUnitPolys to a Map View
    e. Symbolize ContactsAndFaults on Symbol (*recommendation: import symbology from ContactsAndFaults_FGDC_Symbol_Traditional.lyrx)*
    f. Symbolize ContactsAndFaults a second time on LocationConfidenceMeters (Felger 2020; slide 21). *Note – you must have at least one line in your feature class to be able to symbolize using ‘proportional symbols’ as described below*

i. Copy and paste your ContactsAndFaults layer in the Contents pane and add -LCM to the name (i.e. ContactsAndFaults-LCM); make sure it is below ContactsAndFaults in the Contents pane
ii. Right-click on ContactsAndFaults-LCM and choose Symbology from the pulldown. This will open the symbology window
iii. In the pulldown under 'Primary Symbology' in the new window, choose 'proportional symbology', and set Field = LocationConfidenceMeters, Normalization = none, Unit = meters, Data represents = Distance from center, Template > click on this to set the desired line symbol and color (note: choose a simple line, such as 01.01.01 to ensure the buffer accurately reflects the value in LocationConfidenceMeters; more complex symbols, such as hatched lines, might not scale properly)
iv. Adjust the transparency of the layer to a desired value so that you can see through the buffer to the underlying basemap

g. Configure default values and Feature Templates so that required values are populated appropriately. Keep in mind that if you try to digitize a feature, or modify an existing feature, and fields that participate in Constraint Rules are null, you will receive an error message, and will not be able to complete the feature or attribute modifications until you populate the required fields

*h.* Digitize features

*i.* For ContactsAndFaults, choose the line type you want from the Feature Template list (e.g. Contact—Identity and existence certain, location accurate), populate LocationConfidenceMeters and DataSourceID with appropriate values for the feature, and digitize the line
*ii.* For MapUnitPolys/LabelPoints, choose the appropriate Feature Template, enter values for MapUnit and IdentityConfidence, and digitize the point or polygon
*iii.* *Tips: 1) have the attribute table and/or attributes window open as you edit so that you can confirm that the calculation rules are populating fields as desired; 2) for MapUnitPolys/LabelPoints, turn on labelling for the Label field so that you get visual confirmation that the input values for MapUnit and IdentityConfidence are correct (e.g. if MapUnit is 'Tv, and IdentityConfidence is 'questionable', Label should display as 'Tv?')*

Granular workflow combined with dictionary symbology:

1. Download the supplemental files
1. Download and install the GeMS tools for ArcGIS Pro ([https://github.com/usgs/gems-tools-pro](https://github.com/usgs/gems-tools-pro))
2. Open an existing ArcGIS Pro project, or create a new one as needed
3. Create a new GeMS database if you do not already have one

   a. If you want to build MapUnitPolys from ContactsAndFaults and a label points, you will need to create a MapUnitLabelPoints feature class by hand. The feature class should have the same fields as MapUnitPolys, including a MapUnitPolys_ID field

4. Add your geodatabase to the Project Databases folder
5. Add Global IDs to ContactsAndFaults and MapUnitPolys/LabelPoints
6. Add field GeMS_Ltype (text, 254) to ContactsAndFaults
7. Attach attribute domains to the Type and IsConcealed fields in ContactsAndFaults
   a. Use 'Import table to Domain' geoprocessing tool, navigate to the supplemental files, and attach GeMS_Type_CAF.dbf to the Type field, and GeMS_IsCon.dbf to the IsConcealed field
   b. Change the merge and split policies from 'Default' to 'Duplicate', otherwise the attributes will be deleted for lines that you split or merge
8. Add FGDC_ContactsAndFaults_SymbolDict.stylx to the Project Style folder
9. Import attribute rules into ContactsAndFaults and MapUnitPolys/LabelPoints feature classes
   a. In Catalog View, expand the GeologicMap feature dataset so you can see a list of the feature classes
   b. Import Attribute Rules for ContactsAndFaults by doing the following:
      i. Right click on ContactsAndFaults>Design>AttributeRules to access the Attribute Rules interface
      ii. Click on the Import Rules button on the ribbon, navigate to the location of the supplemental files, and choose AttributeRules_ContactsAndFaults_Granular.csv
      iii. If the import is successful, you will see four Calculation Rules and two Constraint Rules
   c. Import Attribute Rules for MapUnitPolys (or MapUnitLabelPoints) by doing the following:
      i. Right click on MapUnitPolys>Design>AttributeRules to access the Attribute Rules interface
      ii. Click on the Import Rules button on the ribbon, navigate to the location of the supplemental files, and choose AttributeRules_MUPolys_or_Pts.csv
         1. If you build MapUnitPolys interactively as you digitize, import the rules into the MapUnitPolys
         2. If you build MapUnitPolys from ContactsAndFaults and MapUnitLabelPoints, you will import the attribute rules into the MapUnitLabelPoints feature class
      iii. Optional – if you create MapUnitPolys interactively, but also want to maintain a MapUnitLabelPoints feature class, import AttributeRules_MUPtsFromMUPolys.csv into your MapUnitLabelPoints feature class – these rules automatically assign the attributes of the underlying polygon to the point when you digitize it

d. Add ContactsAndFaults_FGDC_SymbolDictionary_Granular.lyrx to a Map View. This layer file references the ContactsAndFaultsGranular_Template feature class in the included GeMs_AttRules_U11N83_Granular.gdb (you may have to set the data source for the layer file if the link to the source feature class is broken). It will show up in your contents pane as ContactsAndFaultsTemplate. The feature class contains 30 contact and fault 'template' lines that are symbolized using the dictionary symbology in FGDC_ContactsAndFaults_SymbolDict.stylx, and the values in the Type, IsConcealed, ExistenceConfidence, and LocationConfidenceMeters fields. This will establish default Feature Templates that can be connected to the ContactsAndFaults feature class in your geodatabase (see next step), and subsequently modified to suit your needs

*e.* Change the data source for ContactsAndFaultsTemplate so that it points to the ContactsAndFaults in your geodatabase

   *i.* right click on ContactsAndFaultsTemplate in the contents pane, choose Properties>Source>SetDataSource, and navigate to the location of to the ContactsAndFaults feature class in your geodatabase

   *ii.* Change the name ContactsAndFaultsTemplate to ContactsAndFaults-Symbol in the contents pane

f. Symbolize ContactsAndFaults a second time on LocationConfidenceMeters (Felger 2020; slide 21). *Note – you must have at least one line in your feature class to be able to symbolize using 'proportional symbols' as described below*

   i. Copy and paste your ContactsAndFaults-Symbol layer in the Contents pane and change -Symbol to -LCM (i.e. ContactsAndFaults-LCM); make sure it is below ContactsAndFaults in the Contents pane

   ii. Right-click on ContactsAndFaults-LCM and choose Symbology from the pulldown. This will open the symbology window

   iii. In the pulldown under 'Primary Symbology' in the new window, choose 'proportional symbology', and set Field = LocationConfidenceMeters, Normalization = none, Unit = meters, Data represents = Distance from center, Template > click on this to set the desired line symbol and color (note: choose a simple line, such as 01.01.01 to ensure the buffer accurately reflects the value in LocationConfidenceMeters; more complex symbols, such as hatched lines, might not scale properly)

   iv. Adjust the transparency of the layer to a desired value so that you can see thru the buffer to the underlying basemap

g. Configure default values and Feature Templates so that required values are populated appropriately. Keep in mind that if you try to digitize a feature, or modify an existing feature, and fields that participate in Constraint Rules are null, you will receive an error message, and will not be able to complete the feature or attribute modifications until you populate the fields

h. Digitize features

i. For ContactsAndFaults, choose the line type you want from the Feature Template list (e.g. Contact), populate LocationConfidenceMeters and DataSourceID with appropriate values for the feature, and digitize the line. *(Notes: The default values for IsConcealed and ExistenceConfidence for all of the template lines are 'N' and 'certain', respectively. If these values are not correct for the feature you are digitizing, you can either change them in the Feature Template before you digitize the line, or modify the values in the attribute window after you have digitized the line. There are attribute domains attached to the Type, IsConcealed, and Existence/IdentityConfidence fields.)*

ii. For MapUnitPolys/LabelPoints, choose the appropriate Feature Template, enter values for MapUnit and IdentityConfidence, and digitize the point or polygon

iii. *Tips: 1) have the attribute table and/or attributes window open as you edit so that you can confirm that the calculation rules are populating fields as desired; 2) for MapUnitLabelPoints/Polys, turn on labelling for the Label field so that you get visual confirmation that the input values for MapUnit and IdentityConfidence are correct (e.g. if MapUnit is 'Tv, and IdentityConfidence is 'questionable', Label should display as 'Tv?')*