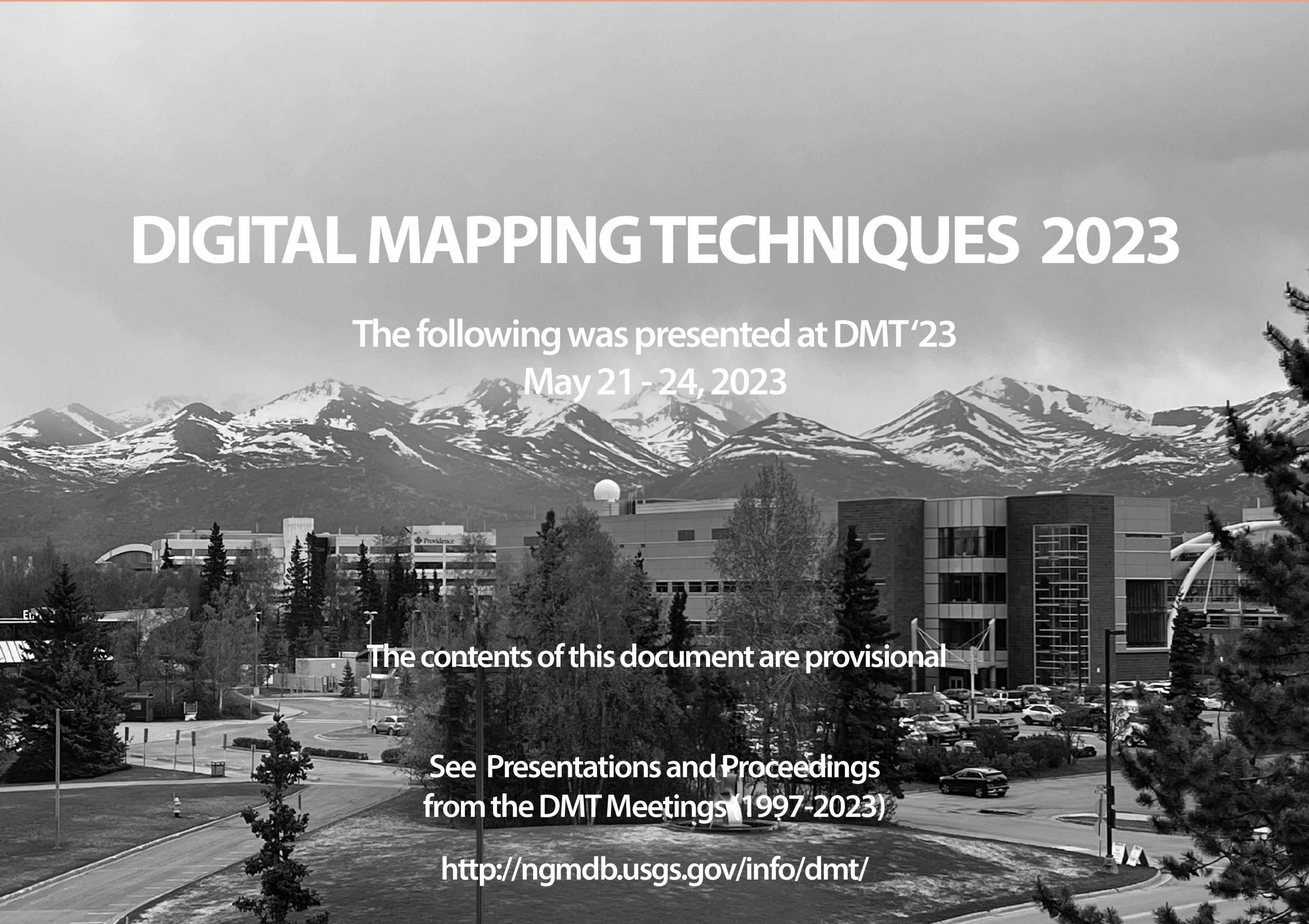


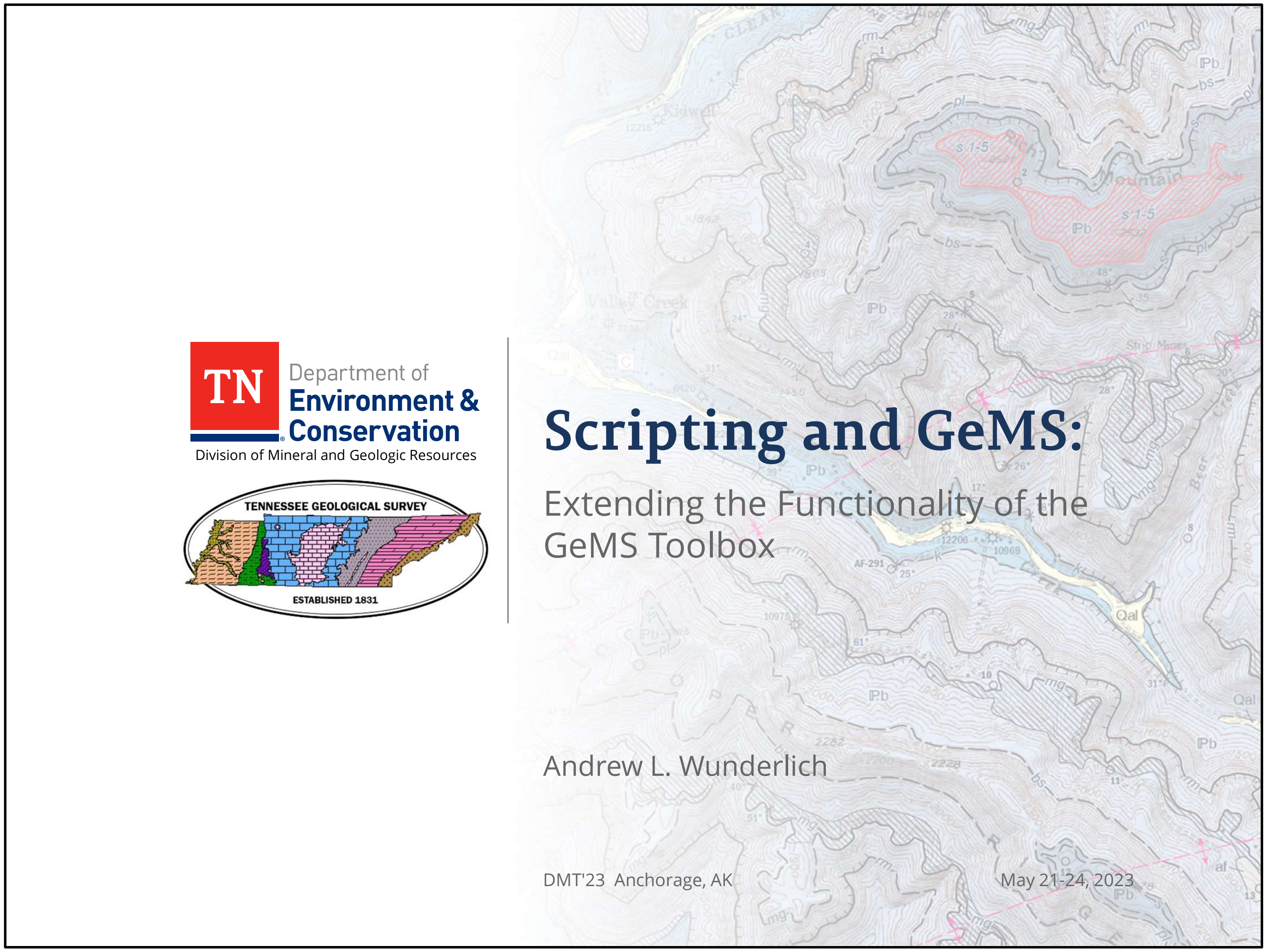
DIGITAL MAPPING TECHNIQUES 2023

The following was presented at DMT '23
May 21 - 24, 2023

The contents of this document are provisional

See Presentations and Proceedings
from the DMT Meetings (1997-2023)

http://ngmdb.usgs.gov/info/dmt/



TN Department of Environment & Conservation

Division of Mineral and Geologic Resources

TENNESSEE GEOLOGICAL SURVEY

ESTABLISHED 1831

Scripting and GeMS:

Extending the Functionality of the GeMS Toolbox

Andrew L. Wunderlich

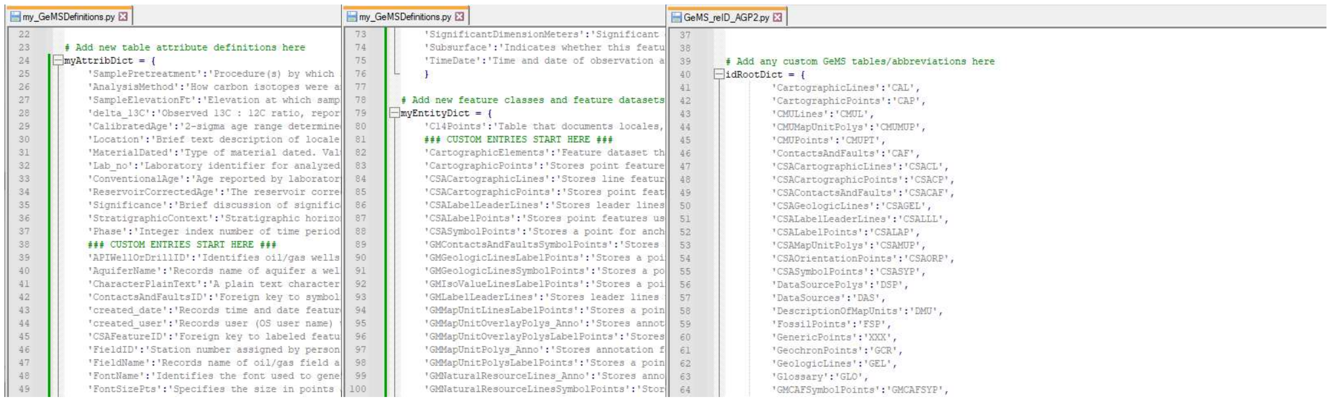
DMT'23 Anchorage, AK

May 21-24, 2023

This lightning talk is going to briefly cover how I have extended the existing GeMS tools to improve functionality, save time, and hopefully help others do the same.

Extension by Modification

- Necessary modifications (GeMS schema extensions):
  - Updates to “my_GeMSDefinitions.py” in the myAttribDict and myEntryDict sections to reflect GeMS schema additions
  - Update list of “_ID” value prefixes in “GeMS_reID_AGP2.py” to prevent use of default values for undefined feature classes.



my_GeMSDefinitions.py

```python

# Add new table attribute definitions here
myAttribDict = {
    'SamplePretreatment':'Procedure(s) by which
    'AnalysisMethod':'How carbon isotopes were a
    'SampleElevationFt':'Elevation at which samp
    'delta_13C':'Observed 13C : 12C ratio, repor
    'CalibratedAge':'2-sigma age range determine
    'Location':'Brief text description of locale
    'MaterialDated':'Type of material dated. Val
    'Lab_no':'Laboratory identifier for analyzed
    'ConventionalAge':'Age reported by laborator
    'ReservoirCorrectedAge':'The reservoir corre
    'Significance':'Brief discussion of signific
    'StratigraphicContext':'Stratigraphic horizo
    'Phase':'Integer index number of time period
    ### CUSTOM ENTRIES START HERE ###
    'APIWellOrDrillID':'Identifies oil/gas wells
    'AquiferName':'Records name of aquifer a wel
    'CharacterPlainText':'A plain text character
    'ContactsAndFaultsID':'Foreign key to symbol
    'created_date':'Records time and date featur
    'created_user':'Records user (OS user name)
    'CSAFeatureID':'Foreign key to labeled featu
    'FieldID':'Station number assigned by person
    'FieldName':'Records name of oil/gas field a
    'FontName':'Identifies the font used to gene
    'FontSizePts':'Specifies the size in points
```

my_GeMSDefinitions.py

```python
    'SignificantDimensionMeters':'Significant
    'Subsurface':'Indicates whether this featu
    'TimeDate':'Time and date of observation a
    }

# Add new feature classes and feature datasets
myEntityDict = {
    'C14Points':'Table that documents locales,
    ### CUSTOM ENTRIES START HERE ###
    'CartographicElements':'Feature dataset th
    'CartographicPoints':'Stores point feature
    'CSACartographicLines':'Stores line featur
    'CSACartographicPoints':'Stores point feat
    'CSALabelLeaderLines':'Stores leader lines
    'CSALabelPoints':'Stores point features us
    'CSASymbolPoints':'Stores a point for anch
    'GMContactsAndFaultsSymbolPoints':'Stores
    'GMGeologicLinesLabelPoints':'Stores a poi
    'GMGeologicLinesSymbolPoints':'Stores a po
    'GMIsoValueLinesLabelPoints':'Stores a poi
    'GMLabelLeaderLines':'Stores leader lines
    'GMMapUnitLinesLabelPoints':'Stores a poin
    'GMMapUnitOverlayPolys_Anno':'Stores annot
    'GMMapUnitOverlayPolysLabelPoints':'Stores
    'GMMapUnitPolys_Anno':'Stores annotation f
    'GMMapUnitPolysLabelPoints':'Stores a poin
    'GMNaturalResourceLines_Anno':'Stores anno
    'GMNaturalResourceLinesSymbolPoints':'Stor
```

GeMS_reID_AGP2.py

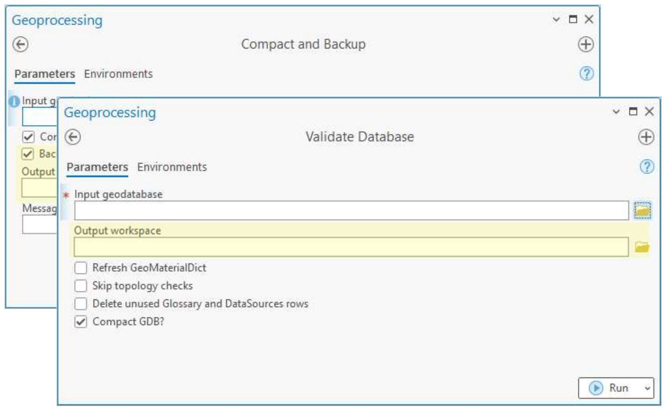
```python


# Add any custom GeMS tables/abbreviations here
idRootDict = {
        'CartographicLines':'CAL',
        'CartographicPoints':'CAP',
        'CMULines':'CMUL',
        'CMUMapUnitPolys':'CMUMUP',
        'CMUPoints':'CMUPT',
        'ContactsAndFaults':'CAF',
        'CSACartographicLines':'CSACL',
        'CSACartographicPoints':'CSACP',
        'CSAContactsAndFaults':'CSACAF',
        'CSAGeologicLines':'CSAGEL',
        'CSALabelLeaderLines':'CSALLL',
        'CSALabelPoints':'CSALAP',
        'CSAMapUnitPolys':'CSAMUP',
        'CSAOrientationPoints':'CSAORP',
        'CSASymbolPoints':'CSASYP',
        'DataSourcePolys':'DSP',
        'DataSources':'DAS',
        'DescriptionOfMapUnits':'DMU',
        'FossilPoints':'FSP',
        'GenericPoints':'XXX',
        'GeochronPoints':'GCR',
        'GeologicLines':'GEL',
        'Glossary':'GLO',
        'GMCAFSymbolPoints':'GMCAFSYP',
```

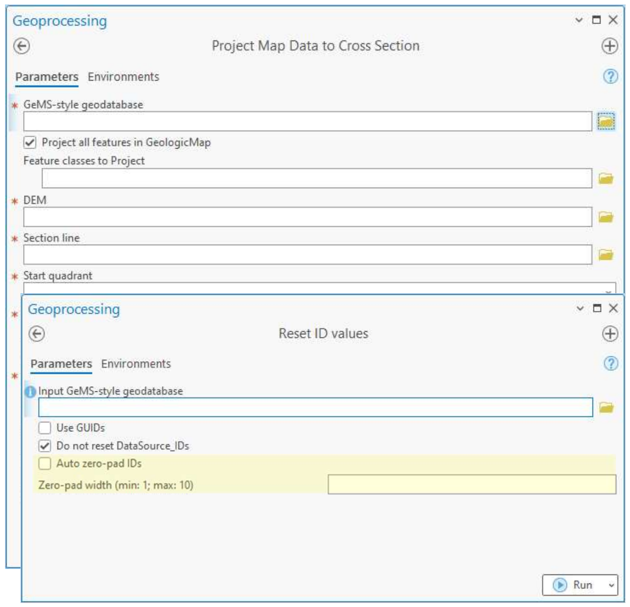
When discussing “extension” or “extending” the tools, there are two ways that this is accomplished: (1) modifying existing tools and (2) creating new tools that “dovetail” into the existing toolbox (discussed later). Within modification, there are two categories of modifications to briefly go over here: the first being the most basic, necessary mods to use the tools (editing lists of constants within scripts). Modifications like this are encouraged and often required, as several of the tools don’t behave very well if you do not maintain these ancillary files.

Extension by Modification

- Convenience modifications:
  - Allow user to specify location of script output(s)
  - Allow user to set adjustable inputs in toolbox interface


Geoprocessing
Compact and Backup
Parameters Environments
Input g
☑ Cor
☑ Bac
Output
Messag

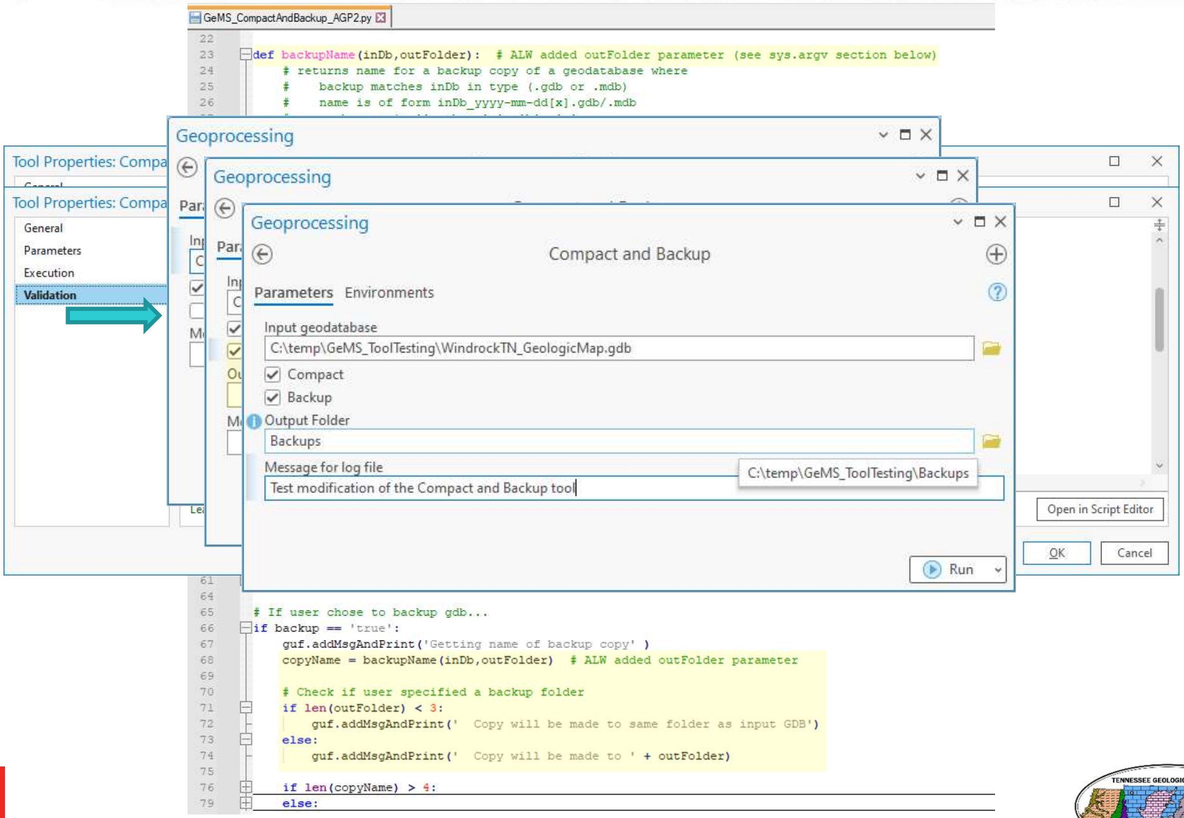
Geoprocessing
Validate Database
Parameters Environments
* Input geodatabase
Output workspace
☐ Refresh GeoMaterialDict
☐ Skip topology checks
☐ Delete unused Glossary and DataSources rows
☑ Compact GDB?
Run


Geoprocessing
Project Map Data to Cross Section
Parameters Environments
* GeMS-style geodatabase
☑ Project all features in GeologicMap
Feature classes to Project
* DEM
* Section line
* Start quadrant

Geoprocessing
Reset ID values
Parameters Environments
Input GeMS-style geodatabase
☐ Use GUIDs
☑ Do not reset DataSource_IDs
☐ Auto zero-pad IDs
Zero-pad width (min: 1; max: 10)
Run

The second type of modifications are referred to as "convenience" modifications, and these are slightly more advanced. These mods go a little deeper in that you must modify the code as well as the toolbox user interface through ArcGIS. These mods really help with workflow; e.g., being able to specify the location that a tool stores its output(s) is a real time saver and assists with maintaining organization of files and folders within a project. Similarly, adding variables to the tool interface that are otherwise only modifiable constants in the script save the steps of opening, finding code, and editing it. This is also not a good practice as errors could be unwittingly introduced into the script.

Mod Example: Compact and Backup



GeMS_CompactAndBackup_AGP2.py

```
def backupName(inDb,outFolder):  # ALW added outFolder parameter (see sys.argv section below)
    # returns name for a backup copy of a geodatabase where
    #    backup matches inDb in type (.gdb or .mdb)
    #    name is of form inDb_yyyy-mm-dd[x].gdb/.mdb
```

Geoprocessing — Compact and Backup

Parameters | Environments

Input geodatabase: C:\temp\GeMS_ToolTesting\WindrockTN_GeologicMap.gdb
✓ Compact
✓ Backup
Output Folder: Backups
Message for log file: Test modification of the Compact and Backup tool

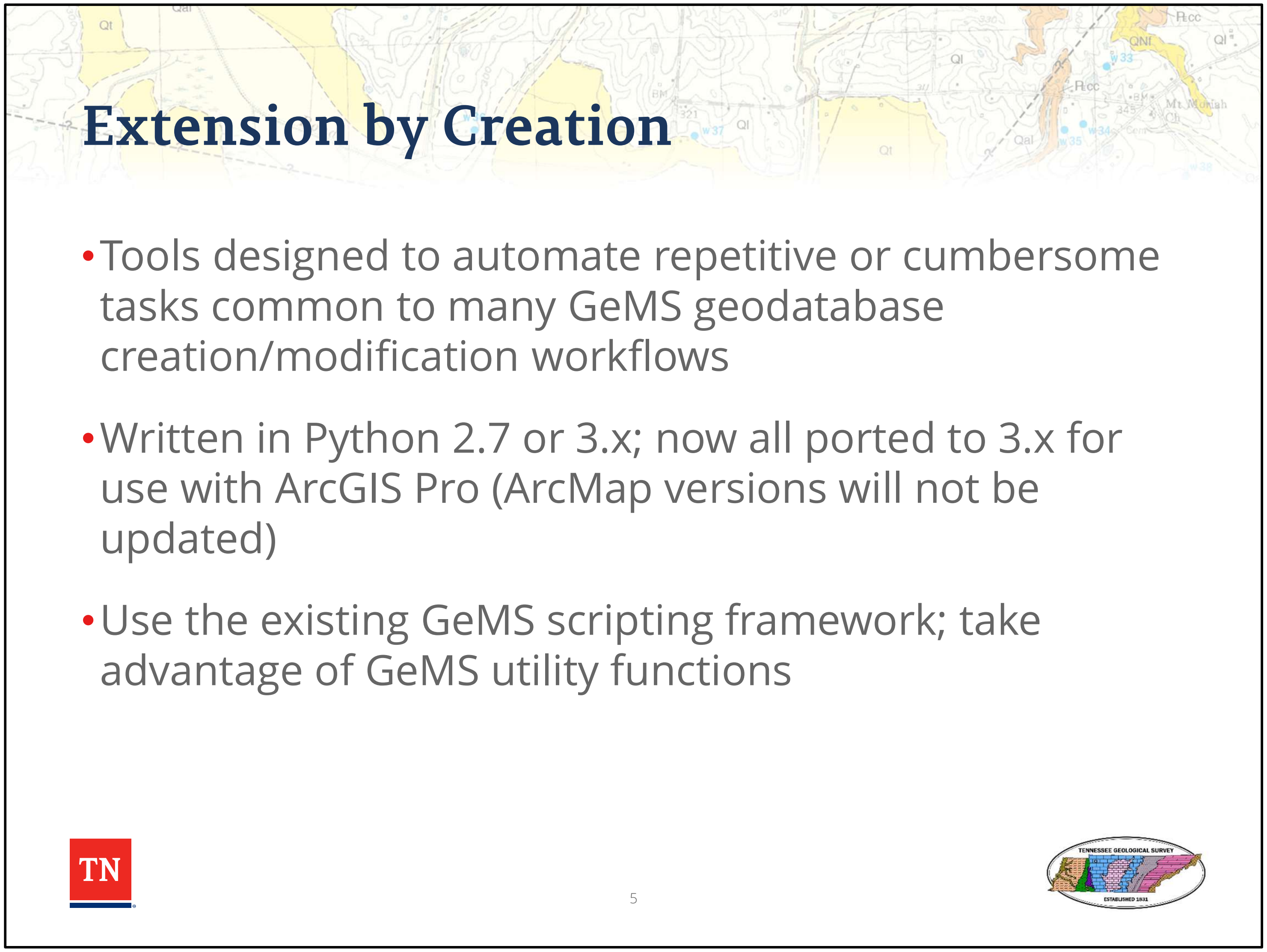
C:\temp\GeMS_ToolTesting\Backups

```
# If user chose to backup gdb...
if backup == 'true':
    guf.addMsgAndPrint('Getting name of backup copy' )
    copyName = backupName(inDb,outFolder)  # ALW added outFolder parameter

    # Check if user specified a backup folder
    if len(outFolder) < 3:
        guf.addMsgAndPrint('  Copy will be made to same folder as input GDB')
    else:
        guf.addMsgAndPrint('  Copy will be made to ' + outFolder)

    if len(copyName) > 4:
    else:
```

This slide demos the modified Compact and Backup script highlighting the addition of an output folder variable (code available on the GeMS GitHub site).



Extension by Creation

- Tools designed to automate repetitive or cumbersome tasks common to many GeMS geodatabase creation/modification workflows
- Written in Python 2.7 or 3.x; now all ported to 3.x for use with ArcGIS Pro (ArcMap versions will not be updated)
- Use the existing GeMS scripting framework; take advantage of GeMS utility functions

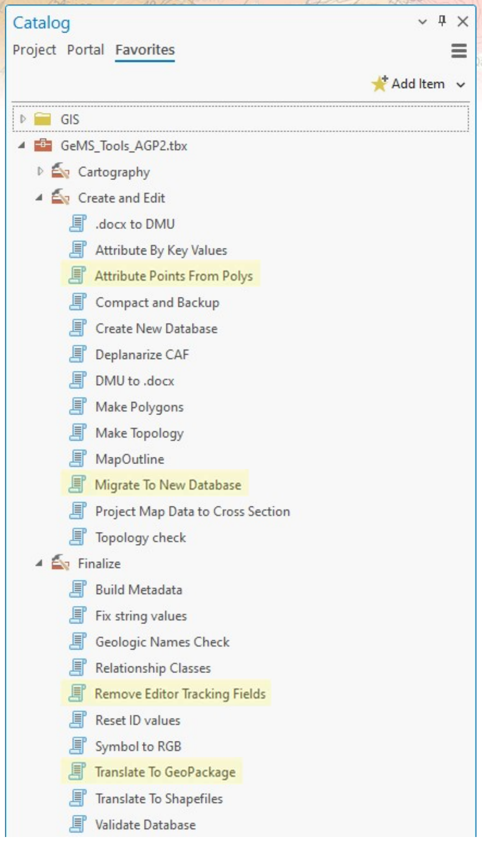
TN

TENNESSEE GEOLOGICAL SURVEY ESTABLISHED 1831

Extension by creation is just that: writing new scripts. In many cases these scripts piggy-back on others or rely on functions already included within the GeMS utility functions script. Code can be cobbled together from examples (mostly I take snippets from Esri's ArcPy examples) as well as from the many online forums where people ask questions and others respond with code samples that may help solve a particular problem.

Extension by Creation

- Create and Edit:
  - Attribute Points From Polys
  - Migrate To New Database
- Finalize:
  - Remove Editor Tracking Fields
  - Translate to GeoPackage


Catalog

Project Portal Favorites

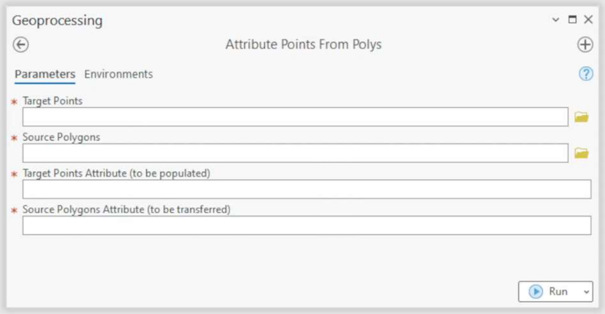
Add Item

- GIS
- GeMS_Tools_AGP2.tbx
  - Cartography
  - Create and Edit
    - .docx to DMU
    - Attribute By Key Values
    - Attribute Points From Polys
    - Compact and Backup
    - Create New Database
    - Deplanarize CAF
    - DMU to .docx
    - Make Polygons
    - Make Topology
    - MapOutline
    - Migrate To New Database
    - Project Map Data to Cross Section
    - Topology check
  - Finalize
    - Build Metadata
    - Fix string values
    - Geologic Names Check
    - Relationship Classes
    - Remove Editor Tracking Fields
    - Reset ID values
    - Symbol to RGB
    - Translate To GeoPackage
    - Translate To Shapefiles
    - Validate Database

These are four scripts that I have created. Attribute Points From Polys, Remove Editor Tracking Fields, and Translate to GeoPackage are all available now on the GeMS Github site’s Issues page (https://github.com/DOI-USGS/gems-tools-pro/issues)

Attribute Points From Polys

- Script uses an “in-memory” spatial join to transfer an attribute from a polygon class to an attribute of a point class which intersects it.
- Target points and source polygons are filtered by feature type by the ArcGIS Pro “Tool Properties > Parameters”
- Target and source attributes are dependent on the target and source point feature attribute tables



* Target Points

* Source Polygons

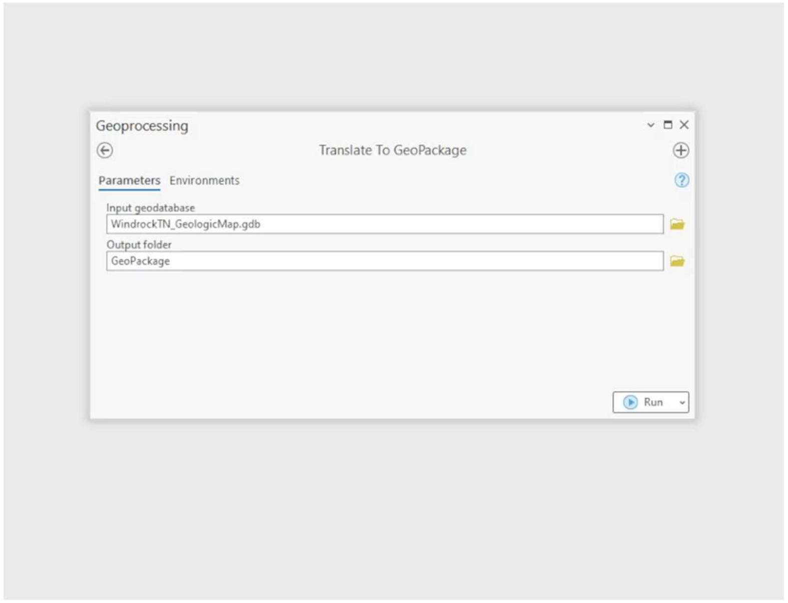
* Target Points Attribute (to be populated)

* Source Polygons Attribute (to be transferred)

Attribute Points From Polys is a Python script of about 100 lines. It stores the polygon attribute values from the spatial join that correspond to the target point feature OIDs and transfers those values to the target attribute in the point class, all without creating an intermediate output or requiring any additional user action.

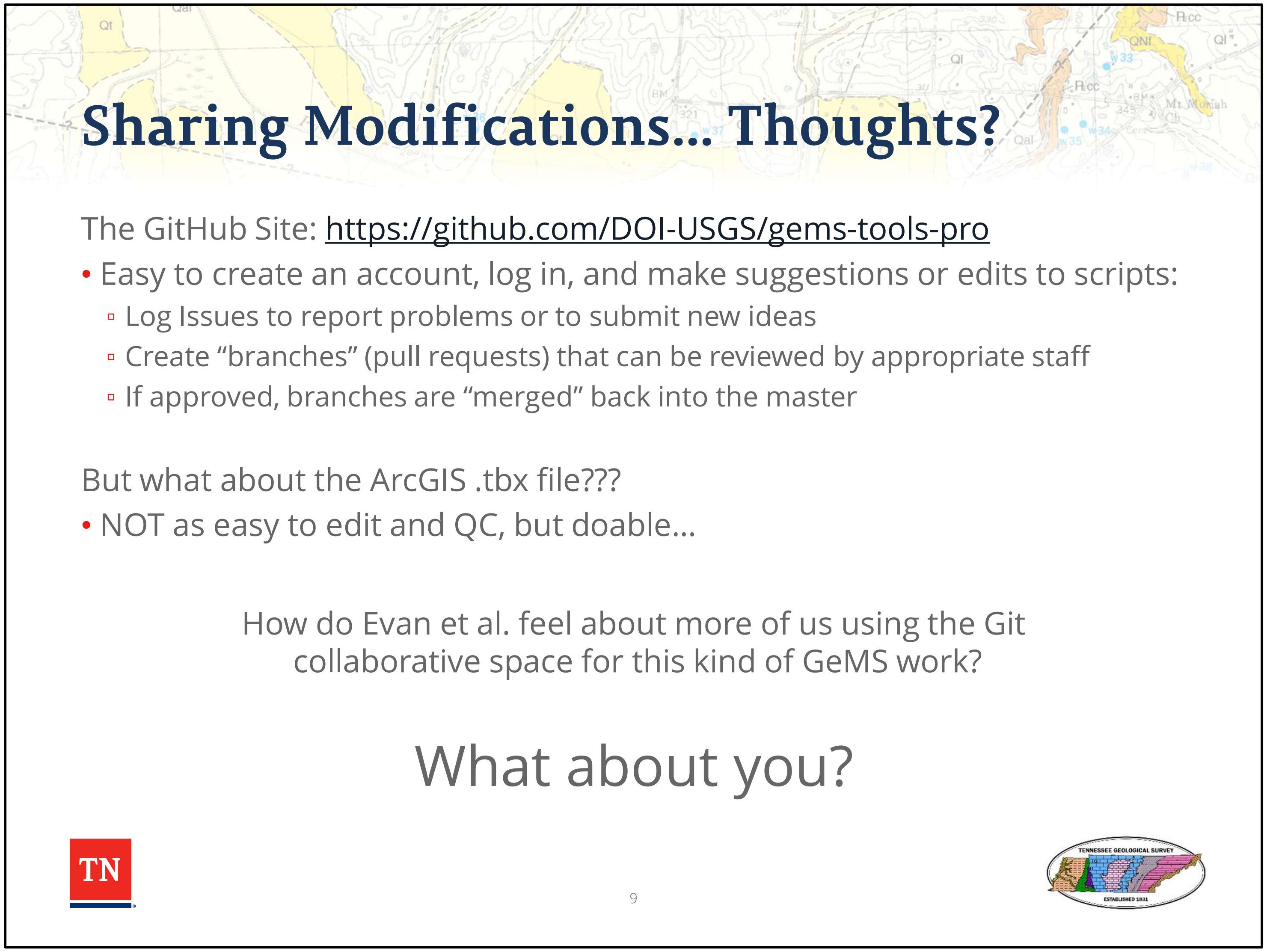
Translate to GeoPackage

- Translates feature classes, tables, and annotation (as points) from a file geodatabase (gdb) to an OGC GeoPackage.
- Geopackages do not support Feature Datasets, so prefixes are added to the output feature class names
- Relationship and Topology classes are not translated by the tool (incompatible with GeoPackages)



Input geodatabase: WindrockTN_GeologicMap.gdb
Output folder: GeoPackage

Translate to GeoPackage is a Python script of about 140 lines. GeoPackages do not support Esri file geodatabase Feature Datasets, so prefixes are added to the feature class names so that they sort together in the output. If the input database is a GeMS-style database, the prefixes are defined (e.g., "GeologicMap" = "GM", "CrossSectionA = "CSA", etc.). Custom names can be specified, or the tool will generate its own prefix. Translations are accomplished through ArcPy conversion functions FeatureClassToFeatureClass, AddRasterToGeoPackage, and TableToTable, along with management function FeatureToPoint which handles annotation. The script can only convert feature classes, tables, and raster datasets.



Sharing Modifications… Thoughts?

The GitHub Site: https://github.com/DOI-USGS/gems-tools-pro

- Easy to create an account, log in, and make suggestions or edits to scripts:
  - Log Issues to report problems or to submit new ideas
  - Create "branches" (pull requests) that can be reviewed by appropriate staff
  - If approved, branches are "merged" back into the master

But what about the ArcGIS .tbx file???

- NOT as easy to edit and QC, but doable…

How do Evan et al. feel about more of us using the Git collaborative space for this kind of GeMS work?

What about you?

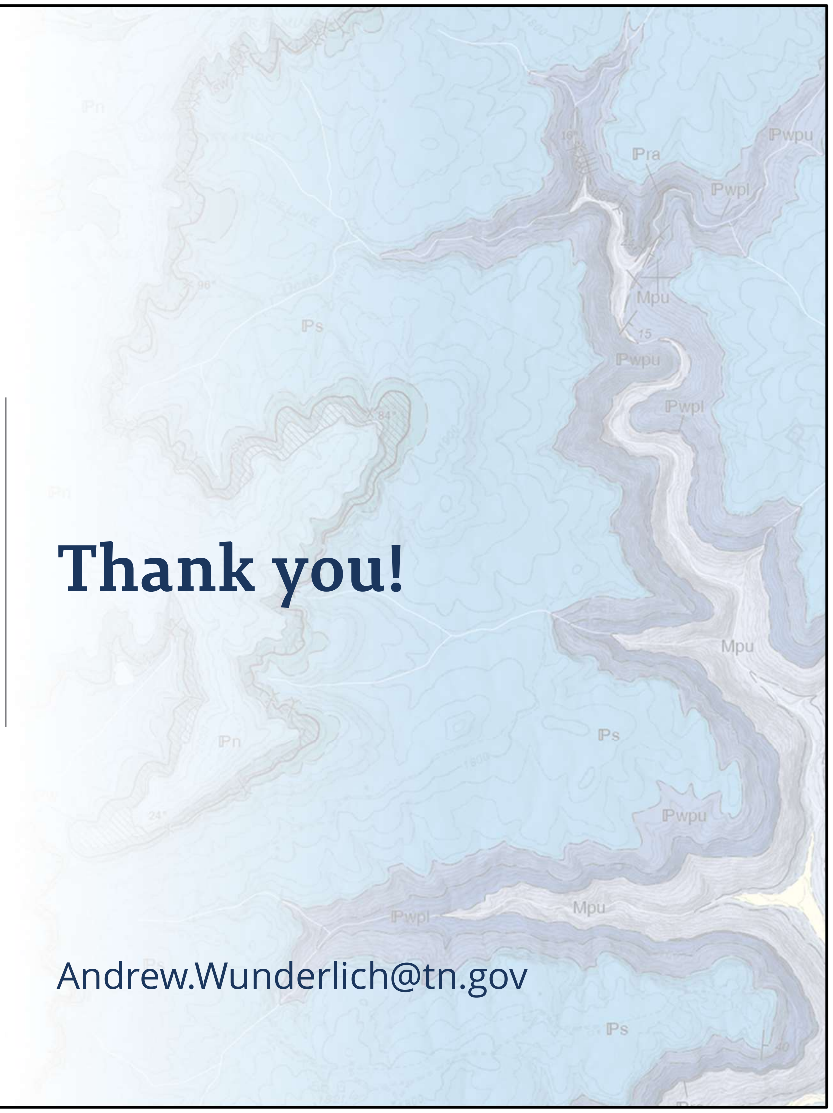
The reason for presenting the extension idea in the two forks is to try and encourage people who may be intimidated by the idea of programming to classify their idea to make it more accessible conceptually. We want there to be a place to make suggestions for modifications to existing tools, suggest new tools, and be able to share experiences using the tools. Where is that space? Github is an obvious but underutilized resource for sharing and modifying collaboratively. After the presentation, I had conversations with several folks and the other option that came up was an ArcGIS Hub site. This is a developing issue and will be discussed further after the meeting with more info on collaboration likely coming through the Listserv.

TN Department of Environment & Conservation

Division of Mineral and Geologic Resources

TENNESSEE GEOLOGICAL SURVEY

ESTABLISHED 1831



Thank you!

Andrew.Wunderlich@tn.gov