DIGITAL MAPPING TECHNIQUES 2024

The following was presented at DMT ‘24
May 13 - 16, 2024

The contents of this document are provisional

See Presentations and Proceedings
from the DMT Meetings (1997-2024)
http://ngmdb.usgs.gov/info/dmt/
Adapting the GeMS Toolbox Tools to Work With Enterprise Geodatabases

Christian Halsted
Director, Earth Resources Information
Maine Geological Survey
93 State House Station
Augusta, ME 04333-0093
Christian.H.Halsted@Maine.gov

Abstract:
In 2022, the Maine Geological Survey (MGS) entered into a two-year Cooperative Agreement with the National Geologic Map Database project to modify the GeMS ArcGIS Pro Toolbox to work with Esri enterprise geodatabases (EGDB) as well as file geodatabases. This new functionality is particularly important to MGS because all mapping and business workflows at the survey have utilized an EGDB on Microsoft SQL Server since 2013. Additionally, the single map – single file geodatabase GeMS approach would be an organizational step backwards for MGS because the current mapping system is multi-map. This means that features classes contain features for all maps at a particular scale. For example, all map unit polygons for each 1:24,000-scale bedrock map in the state are contained in one feature class called Bedrock_Units.

With the establishment of the GeMS submission requirements for STATEMAP funded projects, MGS developed a data translation process that allowed mapping geologists to continue working in the MGS database. A few GeMS fields, such as the confidence values, were added to the MGS schema to be filled out by MGS users. Database scripts developed by MGS then transferred the map data to a GeMS compliant file geodatabase for submission. This process was successful – over 70 GeMS packages have been created – but it could be more efficient particularly at the GeMS Validation point in the process. The Validate Database tool is run on the generated file geodatabase but if there were any errors the data had to be corrected in the MGS database, reexported to the GeMS file geodatabase, and then validated again. This loop could easily take 20-30 minutes and it was common to have to perform this error correction loop at least 2-3 times for a map.

MGS decided that it was important to keep the geologic mapping workflow in an EGDB but that it was time to migrate from the MGS schema to the GeMS schema in the EGDB. This would require that mappers learn to work directly with the GeMS schema. A number of map templates and downstream web data products would also need to be updated. However, these negative short-term impacts are outweighed by the benefits of natively mapping in GeMS so that there are no database translation scripts that must be run and maintained and by allowing mappers to use the GeMS tools, especially the Validate Database tool, to get ‘real time’ feedback on potential errors in the EGDB.

A key requirement of this strategy is that the GeMS ArcToolbox tools must function when run against an EGDB. Fortunately, the Esri tools and the arcpy Python module methods can execute on file or enterprise geodatabase objects interchangeably. A new fork of the USGS GeMS Tools GitHub repository was created. Many of the tools required only minor pathing changes and logic to handle
their tasks on EGDBs as well as file geodatabases. Most of the tools also had their interface updated along with some new validation code to allow for the addition of schema and map name dropdown boxes when working with EGDBs. These two data fields were necessary for doing selections on the multi-map feature classes to run the tool on only the data for the current map.

There were five tools that required or benefited from major code revisions. There were two primary reasons for this – limiting the usage of intermediate on-disk datasets and increasing performance. First, intermediate datasets are difficult to manage in an EGDB so intermediate output was created with in-memory objects, either Python lists or dictionaries or arcpy objects. Secondly, some processes could be run using database code rather than Python which dramatically increased performance because the I/O between the client and database is eliminated.

Three new tools were also developed to work with the multi-map GeMS schema and five tools were not updated because their functionality did not apply to an EGDB.

The goal for this project which ends in 2025, is to have all GeMS tools functioning with both enterprise and file geodatabases. Documentation and tool metadata will also be complete. And finally, a decision about managing the tools in GitHub, either as part of the current repository or as a long term fork, will be finalized and communicated.
Adapting the GeMS Toolbox Tools to Work With Enterprise Geodatabases

Christian Halsted
Director, Earth Resources Information
Maine Geological Survey
Christian.H.Halsted@Maine.gov
Project Scope

• MGS has a Coop Grant with NGMDB: Oct 2022 – Apr 2025
• Modify the GeMS ArcGIS Pro tools to work on File or Enterprise Geodatabases
  – Fork version of tools from GitHub
  – Update Python scripts
  – Determine if some tools can be replaced by RDBMS functions, stored procedures, and/or triggers.
• Possible GeMS schema enhancements
  – Tables for managing photos, citations, map text, photo captions, metadata
• Publish documentation for implementing GeMS in a RDBMS
## File geodatabase vs Enterprise Geodatabase

<table>
<thead>
<tr>
<th></th>
<th>Enterprise Geodatabase</th>
<th>File Geodatabase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage</strong></td>
<td>Relational database</td>
<td>File system folder</td>
</tr>
<tr>
<td><strong>Storage format</strong></td>
<td>Microsoft SQL Server</td>
<td>Each dataset is a separate file on disk. All the datasets that belong to one geodatabase are contained in a single folder.</td>
</tr>
<tr>
<td></td>
<td>PostgreSQL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oracle</td>
<td></td>
</tr>
<tr>
<td><strong>Users</strong></td>
<td>Multiple editors and multiple readers</td>
<td>Single editor and multiple readers</td>
</tr>
<tr>
<td><strong>Size limits</strong></td>
<td>Size is controlled by the DBMS – virtually unlimited</td>
<td>1 TB (can be raised)</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Managed through the DBMS</td>
<td>Managed through the operating system</td>
</tr>
<tr>
<td><strong>SQL programming support</strong></td>
<td>Yes</td>
<td>Limited</td>
</tr>
</tbody>
</table>

Source: Esri
Creating an Enterprise Geodatabase

- Choose RDBMS software
- Install RDBMS on premises or in the cloud (or localhost for development)
- Use Esri ArcToolbox tools to do one of the following:
  - Create a new enterprise geodatabase instance
  - Enable an existing database instance as an enterprise geodatabase

- Esri license and keycodes for ArcGIS Enterprise are required
Why Does MGS Use Enterprise Geodatabases?

- **Single source** – all spatial and non-spatial data in one location and highly integrated
- **Collaboration** – concurrent multi-user editing, no database locking, all edits shared immediately with all users
- **Data Integrity** – uniform data structure, domains, relationships that is not editable by users
- **Performance** – data processing happens on database server, not locally or over a network, so is much faster
- **Maintenance** – centrally managed updates, patching, backups
- **Multi-level security** – full control, edit only, view only for different users on same data
Why Didn’t MGS Transition to GeMS File Geodatabases?

Because...

- File geodatabases would have been a major technological step backward
- Existing deep integration in a single RDBMS used by all staff, not just mappers
- 10+ years using SQL Server
- 30+ years using our schema
- Multi-map feature classes
MGS developed a custom ArcToolbox tool and a stored procedure (4,000 lines of code!)
GeMS Compliance Challenge

MGS to GeMS workflow is time consuming, not done by mapping staff, and requires maintaining a significant amount of code.

Done by Mappers
Repeat 2-5 times

Done by Data Manager

MGS schema
EGDB

(7-14 minutes)

GeMS schema
FGDB

(6-15 minutes)

Pass

Fail

ngmdb connect
New MGS GeMS Workflow

Enable mapping staff to validate data quickly and repeatedly as they are mapping so data is exported to GeMS file geodatabase only once for final validation and submission.

How to achieve this?
1) Migrate ~2 million records in the MGS EGDB from MGS to GeMS schema
2) Update GeMS tools scripts to work on enterprise geodatabases
3) New GeMS tool to export to file geodatabase
GeMS Multi-Map Schema in Enterprise Geodatabase

- Use the Esri ArcToolbox tool to create database authenticated users, and thus schemas, for each map series and scale combination.

- Create ArcCatalog connection files to administer the schemas.
Create GeMS Objects in Enterprise Geodatabase

- Use newly updated GeMS Create New Database tool to add GeMS tables, feature classes and feature datasets to selected db schema
- Should be connected as the user/schema
- Name of geodatabase isn’t used when creating EGDBs
Create GeMS Objects in Enterprise Geodatabase

- All GeMS objects will have the same base name as they would in a file geodatabase
- The database instance name and schema/username are appended with dot(.) notation allowing the same base name to be reused
  
  `MGS_Data.GEMS_BDRK24K.Stations` becomes `GEMS_BDRK24K.Stations`

- Note: Esri is removing the database name in EGDBs created from ArcGIS Pro 3.x or upgraded to enterprise database version 11.x

  `MGS_Data.GEMS_BDRK24K.Stations` becomes `GEMS_BDRK24K.Stations`
Create GeMS Objects in Enterprise Geodatabase

• New extension tables added to Create New Database and A La Carte tools
• GeMS_Definition.py file dictionaries updated – startDict, shape_dict, entityDict
• Applies to both file and enterprise geodatabases
Create GeMS Objects in Enterprise Geodatabase

- GeMS objects in EGDB will be multi-map enabled with new MapName field by default
- Usually, the MapName value is the quadrangle name
- A domain called “MapNameValues” will be created in the EGDB
- New GeMS MapName Definition Query tool will automatically set a definition query on every layer in a map window
Create GeMS Objects in Enterprise Geodatabase

- Grant view and/or edit privileges on objects to database users

- Create ArcCatalog connection files for user access to database
Changes Behind the GeMS Tools for Enterprise Geodatabase Use

- Fortunately, Esri tools and arcpy module methods can execute on file or enterprise geodatabase objects interchangeably.
- Many GeMS tool scripts required only minor path logic changes.

```python
outputDir = sys.argv[1]
if outputDir == "#":
    outputDir = os.getcwd()
outputDir = outputDir.replace("\\", "/")
thisDB = sys.argv[2]
thisDB = thisDB + ".gdb"

if getGDBType(outputDir) == 'EGDB':
    addMsgAndPrint('Creating GeMS database in enterprise geodatabase')
    desc = arcpy.Describe(outputDir)
    cp = desc.connectionProperties
    dbUser = cp.user
    dbName = cp.database
    dbNameUserPrefix = dbName + '.' + dbUser + '.'

thisDB = sys.argv[2]
if getGDBType(outputDir) == 'FileGDB':
    thisDB = thisDB + ".gdb"
```

- New `getGDBType` function added to GeMS_utilityFunctions.py for global use.
Changes Behind the GeMS Tools for Enterprise Geodatabase Use

- Many tools have updated validation scripts to conditionally show the DB.Schema and MapName fields when the Geodatabase input is an enterprise geodatabase rather than a file geodatabase.
Changes Behind the GeMS Tools for Enterprise Geodatabase Use

• Some tools had major rewrites to:
  – Convert intermediate file outputs to in-memory objects

```python
## MAP OUTLINE
- # make XY file for map outline
  - addHMsgandPrint(“ writing map outline file”)  
  - genx = open(os.path.join(“scratch”, “xxxbox.csv”), “w”)
  - genx.write(“LONGITUDE, LATTITUDE
”)
  - genx.write(“”, “ format(“%f”, str(minxlong, str(maxxlat)))
  - genx.write(“”, “ format(“%f”, str(maxxlong, str(minxlat)))
  - genx.write(“”, “ format(“%f”, str(maxylong, str(minylat)))
  - genx.write(“”, “ format(“%f”, str(maxylong, str(minylat)))
  - genx.close()
  -
- # convert XY file to .dbf table
  - boxdbf = arcpy.CreateScratchName(“XXX”, “.dbf”, “”, “scratch”)
  - boxdbf = os.path.basename(boxdbf)
  - arcpy.TableToTable_conversion(os.path.join(“scratch”, “xxxbox.csv”), “scratch”, boxdbf)
  -
- # make XY event layer from .dbf table
  - arcpy.MakeXYEventLayer_management(  
  - )
  -
  - # convert event layer to preliminary line feature class with PointsToLine_management
  - arcpy.PointsToLine_management(“boxlayer”, “xxxlineOut”)  
  -
  - # densify MapOutline
  - arcpy.Densify edits(“xxxMapOutline”, “DISTANCE”, 0.0001)
  - arcpy.Project_management(  
    - “xxxMapOutline”, “MapOutline”, outSpRef, geotransformation, xycs
  - )
  
```
Changes Behind the GeMS Tools for Enterprise Geodatabase Use

- Some tools had major rewrites to:
  - Use database stored procedures instead of arcpy, for example Validate Database
Status of the GeMS Tools for Enterprise Geodatabase Use

- Public fork at GitHub https://github.com/chhalsted/gems-tools-pro
- Late alpha or early beta stage as of May 2024
- Fork merging to parent still to be determined

- Minor, mostly pathing updates
  - Inclination Numbers
  - Set PlotAtScale values
  - A La Carte GeMS
  - Create New Database
  - DMU to .docx
  - Export GDB to GPKG
  - Make Topology
  - Topology check
  - Fix string values
  - Geologic Names Check
  - Symbol to RGB
  - Translate to Shapefiles

- Major updates or functionality changes
  - Make Polygons
  - MapOutline
  - Project Map Data to Cross Section
  - Validate Database

- Not yet upgraded
  - Set Symbol Values
  - .docx to DMU
  - Attribute by Key Values
  - Deplanarize CAF
  - Reset ID values

- Upgrade not needed
  - Compact and Backup
  - Rename ID fields
  - Submission Directory Tree
  - Build Metadata
  - Relationship Classes

- New tools
  - Export EGDB to FileGDB
  - MapName Definition Query
  - MapName Domain Update
Next Steps

- MGS schema to GeMS schema production migration – summer 2024
- Possible GeMS schema enhancements
  - Tables for managing photos, citations, map text, photo captions, metadata
- Update GeMS Toolbox tool metadata for EGDB usage
- Write documentation for implementing GeMS in a RDBMS
- Decision on GitHub merge commit

Questions?

THANKS!