

VT DataRail Tools

What they Are and How To Use Them

Updated on 09/27/2019

What They Are

VT DataRail Tools are tools that support sharing and exchange of geospatial-data resources per the VT EGC (Enterprise GIS Consortium) Geospatial Data Exchange Protocol.

The tools are built with **arcpy**—via **ArcGIS Desktop 10.6.1** and **Python 2.7.14**.



Read the EGC Geospatial Data Exchange Protocol for important background information, protocol requirements, and definitions (hub, spoke, push, pull, etc.).

Each tool is either a script tool that is run via the toolbox VT DataRail Tools.tbx or a stand-alone script that should be run directly in Python.

Tool Descriptions and Instructions for Use

Inspect Metadata—ArcToolbox tool that calls vtDataRail_InspectMetadata.py

Purpose

For each of a given list of data items (feature classes, tables, rasters, etc.), reads the item's metadata and determines if the ISO-Core metadata-elements are populated—per the VT GIS Metadata Standard.

Writes a .txt report on findings. The report is appended if it already exists; this allows 1 report to be provided for multiple items.

ReadMe Notes

This tool only works when run in ArcGIS Desktop as a script tool. It doesn't work as a stand-alone script; this is because this script uses `arcpy.ExportMetadata_conversion()`, which doesn't work in stand-alone scripts. `arcpy.ExportMetadata_conversion()` might have a 32-bit-process/64-bit-process issue.

In this script's scope, the important thing is having key (core) metadata in the data item, regardless of the metadata style that is applied.

This script reads an item's metadata by exporting its metadata to 2 temporary files:

- One is an FGDC-CSDGM XML-file.
- The other is an ISO XML-file.

In case you are wondering why metadata is exported to 2 formats for reading:

- Some metadata content is more script-readable in 1 format vs. the other.
- Some FGDC CSDGM content isn't directly translatable to ISO, but can still be important.

In case you aren't very familiar with how metadata is stored and processed in ArcGIS... The ArcGIS metadata format is a container of other formats (e.g., ISO 19115 NAP, FGDC CSDGM, etc.). Metadata that is wired into an ArcGIS data-item (e.g., a feature class) is always stored in

the ArcGIS format; ArcGIS-format metadata contains elements that are translatable to any ArcGIS-supported metadata standard PLUS some elements (e.g., thumbnail, geoprocessing history, etc.) that are only in the ArcGIS metadata format.

A metadata style is an ArcGIS setting that determines how metadata is viewed and edited in the ArcGIS metadata editor. If a standard-related style (e.g., ISO 19115 NAP) is applied, the metadata editor shows all elements of that style PLUS ArcGIS elements that are extraneous to that style (e.g., thumbnail, geoprocessing history, etc.). An exception to style yielding a superset is the Item Description style, which only shows a subset of ArcGIS-metadata elements--even if the underlying metadata has been edited to include content beyond the Item Description style.

To select a metadata style in ArcGIS Desktop:

Customize | *Options | Metadata tab

How To Use

Run in ArcGIS Desktop as a script tool (in ArcToolbox).

Write Core-Metadata—ArcToolbox tool that calls vtDataRail_WriteCoreMetadata.py

Purpose

For a given data item (feature class, table, raster, etc.) OR a given HTML file that was previously produced by this tool, interacts w/ user via a GUI to produce 2 metadata files that are populated for ISO-Core metadata-elements per the VT GIS Metadata Standard.

One metadata file is an "ISO 19139" XML-file--which contains "ISO 19115 NAP"-format metadata--for importing metadata into a data item (e.g., geodatabase feature-class) in ArcCatalog.

The other metadata file is an HTML file that stores and presents the metadata.

Can be used to quickly produce NEW compliant metadata or update metadata that was PREVIOUSLY PRODUCED BY THIS TOOL.

By design, this script doesn't overwrite its output to the given data-item's metadata. Importing metadata overwrites pre-existing metadata. The "Import Metadata (Conversion)" tool can be used to import a metadata-XML file to a data item's metadata (again, pre-existing metadata is overwritten).

Before importing this script's output to a data item, make sure that the data item is the data item to which the metadata should be imported (and overwritten).

ReadMe Notes

This tool's script requires an initialization file (vtDataRail_WriteCoreMetadata.INI.xml) in its script directory. The .INI.xml file is an XML file that is used for setting certain default metadata-element content. The XML file has a simple structure and specification for easy editing. Out of the box, this script comes with a vtDataRail_WriteCoreMetadata.INI.xml file in its directory.

The structure for vtDataRail_WriteCoreMetadata.INI.xml is:

```
<?xml version="1.0"?>
  <SETTINGS>
    <SETTING name="">
      ...setting's value goes here...
    </SETTING>
    <SETTING name="">
      ...setting's value goes here...
    </SETTING>
  </SETTINGS>
```

The SETTING-element 'name' attributes and contents are:

'name' Attribute	Contents
abstract	Brief summary of dataset's contents (1 or a few sentences). Can insert a standard disclaimer here.
poc_org	Dataset POC as an organization name, e.g., VT Agency of Natural Resources.
poc_individual	Dataset POC as an individual name.
poc_role	Dataset POC as a role, e.g., Database Administrator.
topics	Dataset's topic categories. Choose 1 or more from (farming, biota, boundaries, climatologyMeteorologyAtmosphere, economy, elevation, environment, geoscientificInformation, health, imageryBaseMapsEarthCover, intelligenceMilitary, inlandWaters, location, oceans, planningCadastre, society, structure, transportation, utilitiesCommunication). In .INI.xml file, separate topics with a ','.
extent_xmin	West boundary of dataset's extent, in decimal degrees.
extent_ymin	South boundary of dataset's extent, in decimal degrees.
extent_xmax	East boundary of dataset's extent, in decimal degrees.
extent_ymax	North boundary of dataset's extent, in decimal degrees.

NOTE: .INI.xml content is only in-play for attributes not already populated in input data-item or input HTML file.

When this script writes the output HTML-metadata file, it writes a Python dictionary object named named "MetaContainer" into a commented area. The MetaContainer stores metadata content. When an HTML metadata-file is given to this tool (file must have been previously created by this tool), this tool looks for and reads the MetaContainer.

Read an HTML file's comments in its source code to determine if it was generated by this tool.

This script has a seed XML-file in its directory (vtDataRail_WriteCoreMetadata.SEED.xml),

which is an internal XML-file that the script uses to start creating XML files. DON'T ALTER IT unless you are forking/branching the script's codebase; CONSIDER IT TO BE A PART OF THE CODEBASE.

In case you aren't very familiar with how metadata is stored and processed in ArcGIS... The ArcGIS metadata format is a container of other formats (e.g., ISO 19115 NAP, FGDC CSDGM, etc.). Metadata that is wired into an ArcGIS data-item (e.g., a feature class) is always stored in the ArcGIS format; ArcGIS-format metadata contains elements that are translatable to any ArcGIS-supported metadata standard PLUS some elements (e.g., thumbnail, geoprocessing history, etc.) that are only in the ArcGIS metadata format.

A metadata style is an ArcGIS setting that determines how metadata is viewed and edited in the ArcGIS metadata editor. If a standard-related style (e.g., ISO 19115 NAP) is applied, the metadata editor shows all elements of that style PLUS ArcGIS elements that are extraneous to that style (e.g., thumbnail, geoprocessing history, etc.). An exception to style yielding a superset is the Item Description style, which only shows a subset of ArcGIS-metadata elements--even if the underlying metadata has been edited to include content beyond the Item Description style.

To select a metadata style in ArcGIS Desktop:

Customize | *Options | Metadata tab

In ArcGIS Open Data, metadata that is imported to an item must be in the ArcGIS metadata-format. ArcGIS Online has a setting that determines the metadata style (e.g., ISO 19115 NAP, FGDC CSDGM) in which metadata is presented via ArcGIS Open Data.

How To Use

Run in ArcGIS Desktop as a script tool (in ArcToolbox).

vtDataRail_SendFreight.py— A stand-alone script that should be run directly in Python

Purpose

Pushes or pulls data from a source geodatabase to a target geodatabase—supporting the Vermont Enterprise GIS Consortium (EGC) Geospatial Data Exchange Protocol.

ReadMe Notes

Data-object types that can be pushed/pulled with this script are:

- feature class
- non-spatial table (must be discoverable by browsing geodatabase)
- view-based data, vector or non-spatial (must be discoverable by browsing geodatabase)
- raster dataset

Integrity-assertion objects--such as a topologies and network datasets--aren't pushed/pulled. Not pushing/pulling them is prudent, as they might not function in the target-location's geodatabase release.

Most data can be exchanged without a lot of consideration around disk-space capacity in the target location. Exceptionally large datasets require particularly careful disk-space planning before being pushed/pulled. Check target-location capacity before pushing/pulling a large new




















dataset.

BOTH geodatabases involved with the push/pull must have a populated non-spatial table named A_README as articulated in the EGC Geospatial Data Exchange Protocol.

Table Name: A_README	
Purpose: High-level description of the geodatabase.	
Field Name	Description
PROTOCOL	Text, Length = 40. Enter EGC Geospatial Data Exchange Protocol .
DB_TYPE	Text, Length = 5. If geodatabase is a hub geodatabase, enter Hub . Otherwise, enter Spoke .
CONSTRAINTS	Text, Length = 254. If the geodatabase contains secure and confidential data, describe constraints and reference the corresponding MOU.
NOTE	Text, Length = 254. Free-form field for other helpful information.

If a geodatabase is a SPOKE geodatabase, the SPOKE geodatabase must have a non-spatial table named A_XCHANGE_PARAMETERS as articulated in the EGC Geospatial Data Exchange Protocol.

Table Name: A_XCHANGE_PARAMETERS	
Purpose: For spoke geodatabases only. Used when pushing/pulling data using VT DataRail Tools. Provides optional special directives for controlling which data objects get pushed/pulled to the geodatabase's hub-counterpart or to another publisher or subscriber. When this table isn't empty, only data objects entered in this table are processed; all other data objects are ignored.	
Field Name	Description
OBJECT_NAME	<p>Text, Length = 150. Name of data object—a feature dataset, feature class, table, or raster—to which the special directive applies. Not case-sensitive.</p> <p>Just enter the data-object's name. If the data object is within a feature dataset, just enter the feature class's name.</p> <p>If the data object has a schema prefix in the spoke geodatabase—e.g., GDB_ORG.ORG_Admin., that prefix must be included in the name. For examples: GDB_ORG.ORG_Admin.Boundary_PROJECTS_poly GDB_ORG.ORG_Admin.Boundary_TAXDISTRICTS_poly</p>
IS_FDATASET	<p>A short-integer field. Set default to 0.</p> <p>To apply a special directive to a feature dataset and all of its participating feature classes, set to 1 to indicate that the OBJECT_NAME entry is a feature dataset.</p> <p>Note: It is possible for a feature dataset and a feature class to have the same name in a geodatabase; this field differentiates between the two.</p>

DIRECTIVE	<p>Text, Length = 20. The directive.</p> <p>Valid entries are DETECT_CHANGES, empty quote, and <Null>.</p> <p>If the data object is a feature class or table to be pushed/pulled only if changes (geometry and/or attribute) are detected between it and a pre-existing target counterpart, enter DETECT_CHANGES. Otherwise, leave as Null or enter an empty quote.</p> <div><div></div><div><p>A sort field is required for using DETECT_CHANGES. The sort field must exist in both the source data and pre-existing target data. It must contain unique values. It's sorted in ascending order and is used to compare source features to target features.</p><p>The Feature Compare (Data Management) GP tool is used to conduct change detection. A limitation to change detection on multi-part features is known. Multi-part features that are the same (including attributes) in both data objects have been reported as different by the GP tool.</p></div></div> <div><table><tr><th>SHAPE</th><th>DISTRICT_NUM</th><th></th><th>SHAPE</th><th>DISTRICT_NUM</th></tr><tr><td></td><td>88</td><td></td><td></td><td>88</td></tr><tr><td></td><td>89</td><td>←CHANGE→</td><td></td><td>89</td></tr><tr><td></td><td>90</td><td>←CHANGE→</td><td></td><td>91</td></tr></table></div>	SHAPE	DISTRICT_NUM		SHAPE	DISTRICT_NUM		88			88		89	←CHANGE→		89		90	←CHANGE→		91
SHAPE	DISTRICT_NUM		SHAPE	DISTRICT_NUM																	
	88			88																	
	89	←CHANGE→		89																	
	90	←CHANGE→		91																	
SORT_FIELD	Only in-use if DIRECTIVE is DETECT_CHANGES . Enter the sort field's name.																				
NOTE	Text, Length = 254. Free-form field for other helpful information.																				

If a geodatabase is a HUB geodatabase, the HUB geodatabase must have a non-spatial table named A_XCHANGE_LOG as articulated in the EGC Geospatial Data Exchange Protocol.

Table Name: A_XCHANGE_LOG	
Purpose: Records info on data-exchange activities. VT DataRail Tools write info into this table. Don't register this table as versioned.	
Field Name	Description
DATE	A Date-type field. Date on which record is entered.
NOTE	<p>Text, Length = 254. Free-form field for helpful information.</p> <p>For examples:</p> <p>Refreshed feature-class Boundary_TAXDISTRICTS_poly Added new raster dataset EcologicHabitat_POLLIWOGPOND</p>

Stand-alone feature-classes are pushed/pulled as stand-alone feature-classes to the target.

If a source feature-class is within a feature dataset, the script searches for a feature dataset that has the same name in the target. If the feature dataset is found in the target, the script pushes/pulls data to a feature class within the target feature-dataset. If the target feature-dataset isn't found in the target, the script creates the feature dataset in the target and pushes/pulls data to a feature class within the target feature-dataset.

Source-geodatabase views (when discoverable by browsing the geodatabase) are pushed/pulled as non-view counterparts (e.g., regular feature-class) in the target.

When a feature class or table (or view) is pushed/pulled and it already exists in the target geodatabase, fields of the target geodatabase that aren't reflected in the source geodatabase are populated with Null values, AND fields of the source geodatabase that aren't reflected in the target geodatabase don't carry their values to the target.

When a raster dataset is pushed/pulled and it already exists in the target geodatabase, an exclusive lock is required for the target-geodatabase connection because the script deletes the target raster-dataset and then replaces it by copying from source geodatabase. An "exclusive lock" means that no other users can be connected to the data object. Consult with a DBA for more info.

Source-geodatabase multi-versioned data-objects that don't already exist in the target geodatabase are pushed/pulled as non-versioned data-objects in the target. AVOID PUSHING/PULLING DATA TO MULTI-VERSIONED TARGET-GEODATABASE DATA-OBJECTS; DOING SO MIGHT CREATE ENORMOUS DELTA TABLES AND CAUSE POOR PERFORMANCE IN THE TARGET GEODATABASE.

Metadata is only carried along through the push/pull if the data object doesn't already exist in the target location (because in that case, the data object is copied and pasted as opposed to loaded by row). Make sure that metadata is kept current in both the source data-object and its target counterpart.

This script logs its activity to a log file (vtDataRail_SendFreight.log) which is written into the script's directory at execution time. Periodically truncate or delete the log file to preserve disk space.

Run this script directly in Python--not in ArcGIS Desktop (ArcToolbox). In ArcToolbox (for an unknown reason), the script sometimes has a problem with getting a correct result from `arcpy.ListTables()`.

How To Use

Run in Python (not in ArcGIS Desktop); set major variables in script's section that is commented w/:

```
# ***** SET MAJOR VARIABLES HERE *****.
```