

TEIS ENVIRONMENT

MANAGING TEI AND SOILS GIS DATA
VERSION 7.00 (MARCH 31, 2014)

AUTHORED BY:

MIKE NEAL (VERSIONS 1-3); JEFF KRUYTS (VERSIONS 4-7)

CLOVER POINT CARTOGRAPHICS LTD



Contents

1.0 Background	5
1.1 Vision	5
2.0 Datasets	5
2.1 Design Strategy	5
2.2 Structure Overview	5
2.3 Data Standards	6
2.3.1 Data Projection and Tolerances	6
2.3.2 Identifier Attributes	7
2.4 Loading Procedures	8
2.4.1 Soils Data	8
2.4.2 Non-Soils Data	8
2.5 Naming Conventions	8
2.5.1 Operational Data	8
2.5.2 Tools and Templates	8
2.6 Feature Classes	9
2.6.1 Operational_Data	9
2.6.2 Tools_and_Templates	9
3.0 Tools for TEIS Environment Workflows	13
3.1 Windows Command Prompt	13
3.1.1 Opening the Command Prompt Window	13
3.1.2 Command Prompt settings	13
3.1.3 Keyboard Functions	14
3.1.4 Executing Scripts	14
3.1.5 Script Log Files	15
3.2 Scripts for Maintenance and Export of Full TEIS Environment	15
3.2.1 Simple Data Loader	15

3.2.2 AddIndexes.py	16
3.2.3 RepairGeometry.py.....	16
3.2.4 CheckUniqueTEIS_ID.py.....	17
3.2.5 StandardizeNoData.py	17
3.2.6 ValidateData.py	17
3.2.7 CreateProjectBoundaries.py.....	18
3.2.8 CreateShortTable.py.....	19
3.2.9 ExportToBCGWStaging.py	19
3.2.10 ExportToTEIShare.py.....	20
3.2.11 CreateProjectBoundariesForWebsite.py	21
3.2.12 CreatePDFsForWebsite.py	21
3.2.13 Copy Updated Images and PDFs to MoE Webspace.....	23
3.2.14 ExportToContractorPackage.py	23
3.2.15 CreateAllTemplatesAndMetadataReports.py.....	24
3.2.16 CreateOneTemplateAndMetadataReport.py	24
3.2.17 CreateSubmissionStandardsDocTables.py	25
3.2.18 UpdateTEISMetadataFilesTable.py.....	25
3.2.19 CreateDistributionPackages.py.....	26
3.3 Scripts for Preparing Legacy Data for Incorporation into the TEIS Environment	26
3.3.1 CompareFields.py	26
3.3.2 CompareFieldsNoAlias.py	27
3.3.3 Simple Data Loader.....	27
3.3.4 AddIndexes.py	27
3.3.5 RepairGeometry.py.....	27
3.3.6 CheckUniqueTEIS_ID.py.....	28
3.3.7 StandardizeNoData.py	28
3.3.8 ValidateData.py	28

3.3.9 CreateProjectBoundaries.py (Optional for Legacy Data).....	29
3.3.10 CreateShortTable.py (Optional for Legacy Data).....	30
3.4 Scripts and Tools for Preparing New Mapping Project Data for Submission.....	30
3.4.1 Simple Data Loader.....	31
3.4.2 AddIndexes.py.....	31
3.4.3 Repair Geometry tool.....	31
3.4.4 Check Unique TEIS_ID tool.....	32
3.4.5 Standardize NoData tool.....	33
3.4.6 Validate Data tool.....	33
3.4.7 Validate User-Defined Data tool.....	34
3.4.8 Create Project Boundaries tool (Optional for Contractors).....	35
3.4.9 Create Short Table tool (Optional for Contractors).....	36
3.5 ArcGIS Desktop Tools.....	37
Appendix A: Document Version History.....	37

1.0 Background

1.1 Vision

To create a unified **TEIS_Environment** that is self-documented using metadata and internal tables. All tools necessary to do standard operations (create boundaries, create short tables, do quality control, etc.) on these data sets will be maintained together. All supplemental information will also be included.

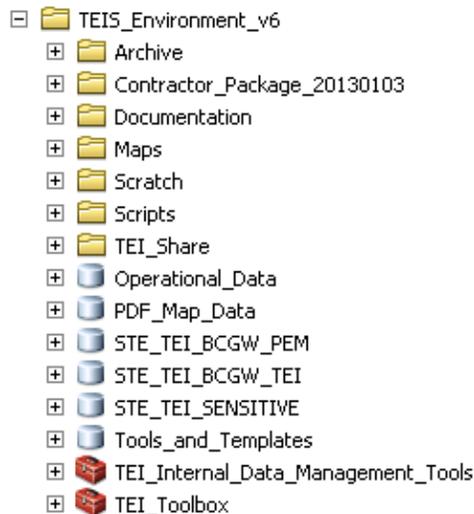
2.0 Datasets

2.1 Design Strategy

All data sets, tools, and resource files are located in a standardized data hierarchy. The principles used to do the design are:

- Ease of use for the end users (even across ArcGIS versions)
- Simple migration to ArcSDE or another data store in the future
- Ease of testing and relocation
- Ease of running complex tools with a minimum of input parameters
- Minimal changes to existing standard attributes

2.2 Structure Overview



The **TEIS_Environment** folder has a number of items in it:

- **Archive** folder – contains past versions of feature class and table templates. These past versions are automatically written here by the script that creates new templates (see section 3.2.15). Other outdated files and datasets may be moved here manually.
- **Contractor_Package_[date]** folder – contains a copy of the Contractor Package created on the date specified. The folder is created and populated automatically by a script (see section 3.2.14).

- **Documentation** folder – a repository for all documentation and metadata related to the TEIS Environment structure and workflows. This does NOT include project level metadata, ie. documents and/or reports submitted with mapping project data.
- **Maps** folder – contains ArcMap project (MXD) files (and supporting data) which are used to produce maps and map images for use on the MoE website (see section 3.2.11).
- **Scratch** folder – A location for all temporary data and tables. Anything left behind in this location (usually by automated processes) can be deleted, provided that no script or tool is currently executing. Normally, a **Scratch.gdb** file geodatabase will be the only content of this folder.
- **Scripts** folder – A home for all of the scripts, executables, log files, etc. used to manage the TEIS Environment data.
- **Operational_Data.gdb** file geodatabase – The primary location of all TEIS Environment geographic and attribute data.
- **PDF_Map_Data.gdb** file geodatabase – spatial data used by the ArcMap projects in the Maps folder. Some feature classes here are derived from operational data by automated processes (see section 3.2.10). Other feature classes were manually extracted from LRDW base map data.
- **STE_*.gdb** file geodatabases – these file geodatabases are created by a script (see section 3.2.8) and are used as staging areas for the transfer of data to the LRDW SDE geodatabase.
- **Tools_and_Templates.gdb** file geodatabase – Contains the template for Operational Data feature classes and tables, and system metadata tables, attribute domain lookup tables, and Quality Control parameters.
- **TEI_Internal_Data_Management** toolbox – this toolbox contains tools for MoE internal use. Currently it contains a tool for extracting TEI data within a specified area of interest, and a tool for checking the success of an SDE data load process.
- **TEI_Toolbox** toolbox – contains ArcGIS Desktop tools for executing commonly-used scripts. For contractors, this toolbox is a more user-friendly alternative to the command prompt interface for running the TEIS Environment’s automated processes.

2.3 Data Standards

2.3.1 Data Projection and Tolerances

All data sets are in the standard BC Albers projection. The projection name for all data sets (except the LRDW exports) has also been standardized to match the RISC standards as **NAD_1983_BC_Environment_Albers**. The template feature classes included in the Tools_and_Templates.gdb file geodatabase have already been set to this projection. When loading data (using ArcCatalog’s Load Data command) into one of the templates from a feature class that is stored in another defined map projection, the data will be reprojected automatically to match the template.

The XY tolerance selected for all feature classes in this repository is 0.001m, which is the default FGDB tolerance for ArcGIS versions 9.2 and higher. Incoming data from other formats (eg. shapefiles, Arc/Info coverages) is restructured to 0.001m tolerance automatically. Exported LRDW-ready feature classes are restructured to 0.002m which is the current standard for that ArcSDE repository.

More detailed information on TEIS Environment GIS data standards can be found in the **TEI Digital Submission Standards** document. Also consult the ESRI ArcGIS Desktop help documentation included with the software installation.

2.3.2 Identifier Attributes

The TEIS Environment table and feature classes use two unique identifiers: one at the project level that describes collections of data (**BAPID**), and one at the detailed polygon level that uniquely identifies each element (**TEIS_ID**). A foreign key for linking to user-defined fields or external reports is maintained in the **PROJPOLYID** field.

2.3.2.1 BAPID

The current definition of **BAPID** is a set of polygons that share project metadata, including Project Type, Author, Geographic Area, and Title. The project area does not have to be continuous (islands as separate polygons, for example); BAPID is assigned independent of geometry. The CreateProjectBoundaries tool is used to create project-level geometry from the detailed polygons when they become available. All BAPID level information is tracked in the BAPID Tracker database. BAPID is considered persistent; all versions of the TEIS_Environment will have the same BAPID for the same Project, and history will be maintained when a BAPID change is required.

The summarized criteria for assigning a BAPID is:

- A single spatial extent (one set of polygons). For example, an SEI project, themed on TEM but using the same polygons, would be considered the same project and use one BAPID. Multipart polygons are acceptable in the case of islands, etc.
- Mapping is completed in a 1-3 year window or the original time frame as determined at the onset of the project, with no interim deliverables.
- Data is delivered as a single contiguous product at one time (sub-areas submitted over multiple years will be treated as individual projects).
- Mapping is all done at a single scale level.
- Mapping is completed by the same mapper/company or group thereof.
- For an update (i.e. attribute changes/linework changes) where all of the original project data will be deleted and replaced, it will be considered the same project. (Keep the BAPID, update any project details.)
- For an update (i.e. model changes or updates at a later date by a different mapper, or to a different standard, etc.) where the intent is to maintain the original mapping as well as the updated (new) mapping, the updated mapping will be treated as a separate project (assigned its own BAPID).
- Where there are multiple levels of sampling detail/survey intensity within a project (ESIL or TSIL), a single BAPID will be assigned, and the differences captured in a comment field.

Metadata for BAPIDs is maintained in the **BAPID Tracker**, which is a Microsoft Access database and front end application for viewing and editing the metadata. The tables in this database can be accessed (read-only) through ArcCatalog by creating an OLE DB connection to the database. See the BAPID Tracker document for more details.

2.3.2.2 TEIS_ID

The **TEIS_ID** field contains a unique identifiers assigned to each detailed polygon in the **TEIS_Master_Long_Tbl**, and is maintained across all derived outputs (short table feature class, error tables, views, etc.). The **TEIS_ID** field should only be populated with unique values by running the **CheckUniqueTEIS_ID.py** script (see section 3.2.3) and the values should NOT be modified by manual editing. The **TEIS_ID** values are considered persistent; they will not change over time.

2.3.2.3 PROJPOLYID

PROJPOLYID takes the place of **ECP_TAG**, **TER_TAG** and other similarly-named fields that were used in older RISC standard datasets as unique identifiers within each mapping project. Values in **PROJPOLYID** may be unique only at project level, while the values in **TEIS_ID** are unique across the entire province.

2.4 Loading Procedures

2.4.1 Soils Data

At version 6 of the TEIS Environment, all existing soils data has been moved to a folder named SOILS_Master which is at the same folder level as the TEIS Environment folder. Adding new Soils data is a complicated (not automated) procedure involving normalized tables and file geodatabase relationships. The only time the TEIS Environment processes need to interact with soils data is during the building of the Master Project Boundaries feature class, as soils project boundaries are incorporated into it.

2.4.2 Non-Soils Data

Adding any non-soils data that matches the RISC standard is a fairly straight forward process. Legacy data, originally submitted to MoE in older GIS data formats, can be processed and transformed to match the current data standards and file geodatabase schema. New mapping project data should be submitted to MoE by contractors with the aid of the Contractor Package, and should already match the TEIS Environment schema. The steps currently recommended for data processing are depicted in the **TEIS Environment Overview** Visio diagram (“Legacy Data” and “New Project Data” pages) and described in detail in the **STE Data Loading Process** documents, all found in the **Documents** folder of the TEIS Environment.

When new or legacy data is received by the TEIS Environment data custodian, it is just a matter of appending it (using the ArcGIS Desktop **Append** tool) to the appropriate “Master” tables (see section 3.2.1). Afterwards, derived datasets (error tables, project boundaries, short table, BCGW staging geodatabase) must be regenerated, which requires significant server processing time (several days if the scripts are executed in the ArcGIS 10.1 framework, or two weeks or more under ArcGIS 10.0 or earlier). For this reason, it is recommended that new data be added to the Master tables in batches a few times per year at most.

2.5 Naming Conventions

2.5.1 Operational Data

- **TEI_*** – Datasets containing only data for a small number of mapping projects that are being processed prior to inclusion in the full TEIS Environment.
- **TEIS_Master_*** – Datasets containing data for all projects that have been incorporated into the TEIS Environment.
- **User_Defined_*** – Tables containing non-standard attributes for detailed mapping project polygons. Slated to be replaced by amalgamated tables in future versions of the TEIS Environment.

2.5.2 Tools and Templates

- **Domain_*** – Used by the Validate and Create Short Table script tools to identify valid values. They are also used as documentation of the various options and values permitted in the fields.
- **TEIS_Metadata_*** – Used to control the behavior of the QC scripts and produce documentation.

- **Template_*** – Used by the scripts and in manual processes as the source for all schema information.

2.6 Feature Classes

2.6.1 Operational_Data

The following are the primary datasets found in the **Operational_Data.gdb** file geodatabase. For descriptions of secondary feature classes and tables, see the **TEIS_Metadata_Tables** table in the **Tools_and_Templates.gdb** file geodatabase (particularly the **Feature_Class**, **Description** and **Comments** fields).

- **TEIS_Master_Long_Tbl** – The primary source (when combined with TEIS_Project_Details) for all detailed project data, and the source for all derived data sets.
- **TEIS_Master_Project_Boundaries** – Polygon feature class derived from the detailed geometries in both the TEIS_Master_Long_Tbl feature class and the various feature classes in the SOILS_Master folder, with attributes taken from the BAPID Tracker database. Each (possibly multipart) polygon depicts the spatial extent of an entire project.
- **TEIS_Master_Short_Tbl** – A feature class derived from TEIS_Master_Long_Tbl, with the same geometry, but summarized attributes (eg. map label strings concatenated from component attributes) that more closely match the attribute table schema found in the LRDW feature classes.
- **Misc_Hyperlinks** – A list of hyperlinks to the current web resources related to all TEIS data sets.

2.6.2 Tools_and_Templates

- **Domain_*** – Used by the Validate and Create Short Table script tools to identify valid values. They are also used as documentation of the various options and values permitted in the fields.
- **Template_DOMAIN** – A table template for manually creating new Domain_* tables. See section 2.6.2.5 for notes on maintaining and creating domain tables.
- **Template_ErrorSummary** and **Template_ErrorTable** – Table templates for the error report tables, which are created and populated by automated processes.
- **Template_Processing_Log** – A blank table with a structure appropriate for automated collecting of log information inside a file geodatabase. This is not recommended for most operations as logging to tables can negatively impact performance.
- **Template_FGDB_*** – Templates created from the Metadata for use in the Operational_Data.gdb file geodatabase and various scripts.
- **Template_SDE_*** – Templates created from the Metadata for use in testing and replicating the LRDW (ArcSDE) versions of the feature classes.
- **TEIS_Metadata_Files** – A list of each item (file, folder, or file geodatabase) in the TEIS Environment. Individual tables and feature classes within file geodatabases are not listed here. Used by automated processes to create a copy of the Contractor Package with up-to-date copies of the files, folders and file geodatabases needed (as specified in the **Cont_Pkg** attribute). See section 2.6.2.1 below.
- **TEIS_Metadata_Items** – A table that describes all fields in all Operational Data tables. Includes definitions, domains, ranges, and which feature classes or tables the field belongs to. Used by various automated processes. See section 2.6.2.2 below.
- **TEIS_Metadata_Report_Tables** – A list of each table found in the TEIS Digital Submission Standards document. This table is used by an automated process that dumps TEIS Environment table metadata into CSV (Microsoft Excel) files, whose contents can then be copied and pasted into the tables in the Microsoft Word document. See section 2.6.2.3 below.

- **TEIS_Metadata_Tables** – A table that describes all feature classes and tables in the environment. Used by various automated processes. See section 2.6.2.4 below.

2.6.2.1 Maintaining the TEIS_Metadata_Files table

This table should be updated anytime a folder, file or file geodatabase is added to the TEIS Environment structure. The structure of the TEIS_Metadata_Files table is simple enough to be understood at a glance, and its contents can be easily updated using the table editing functions in ArcMap.

A script named UpdateTEISMetadataFilesTable.py in the **Scripts** folder has been developed which reads the TEIS_Environment folder structure and updates the table automatically, although the user must still manually assign “Y” and “N” values in the Cont_Pkg, Info_Reqst, TEI_Share and TEI_Data_Zip fields for each record that the script has added to the table. The contents of these four fields indicate whether the item should be copied to the Contractor Package (see section 3.2.14), Information Request packages, the TEI_Share folder (see section 3.2.9) or the TEI_Data_Zip file respectively.

The table does not list individual feature classes and tables within a file geodatabase; only files and folders as they would be seen in Windows Explorer should be listed here. In the case of file geodatabases, only the parent folder (with the .gdb extension) should be listed in this table, not any of the files stored in the folder (such as a00000001.gdbindexes and so on).

2.6.2.2 Maintaining the TEIS_Metadata_Items table

This is a table describing every field in every operational data table in the TEIS Environment, including field types, text descriptions, and position order within the schemas of tables. As it is the primary source for field definitions, modifications to this table will affect the schemas of templates created by automated processes. The following is a description of sections (groups of fields) in this table, and what is affected by the information in them:

- Fields with names prefixed by FGDB_ are for defining the items’ field types within a file geodatabase environment (such as the TEIS Environment). Information in these fields is used by the CreateAllTemplatesAndMetadataReports.py script (see section 3.2.15) to create the Template_FGDB_* templates in the Tools_and_Templates.gdb file geodatabase.
- Fields with names prefixed by Oracle_, SDE_, E00_, and RISC_ contain information defining the item in other environments, such as the LRDW’s SDE geodatabase. Information here is used to create the Template_SDE_* templates in the Tools_and_Templates.gdb file geodatabase.
- The data in the Item_Indexed field is used by the AddIndexes.py script (see section 3.2.2) to determine which fields should have indexes built on them.
- The data in the Item_Mandatory, Item_Domain_Name, Item_Range_Minimum and Item_Range_Maximum fields is used by the ValidateData.py script (see section 3.2.5) to determine which fields in the input table should be considered mandatory, and if values in the operational data table are outside of their defined domains or ranges.
- The fields whose names correspond to table names (eg. STE_TEI_ATTRIBUTE_POLY_SP, TEI_Long_Tbl) contain values that define the ordinal positions of the items within that table. Values found within each of these fields should be unique and not miss any values, ie. if TEI_Long_Tbl has 281 fields, then each number 1 through 281 should occur once and only once in the TEI_Long_Tbl field of the TEIS_Metadata_Items table.

The table contents can be easily modified using the editing tools in ArcMap.

2.6.2.3 Maintaining the *TEIS_Metadata_Report_Tables* table

This table is particularly tailored to be used by the `CreateSubmissionStandardsDocTables.py` script (see section 3.2.17). Only if the structure of the `TEIS_Digital_Submissions_Standards` document (in the Documents folder) is changed will it be necessary to modify the contents (and possibly the structure) of this table.

To distinguish between “tables” in a file geodatabase and “tables” in a document, we will refer to the latter as “document tables”.

Three types of document tables are present in the March 31, 2011 version of the document: document tables 2 to 4 (each with three columns) which list and describe feature classes, document tables 5 to 17 (each with 8 columns) which list and describe the attributes of the feature classes, and document tables 18 to 73 (each with three columns) which list and describe the contents of the Domain tables. The `CreateSubmissionStandardsDocTables.py` script is able to construct. Currently in `TEIS_Metadata_Report_Tables`, each record represents one of the document tables in `TEIS_Digital_Submissions_Standards`. The fields in `TEIS_Metadata_Report_Tables` are as follows:

- **Table_Name:** the title of the document table as it should appear in the `TEIS_Digital_Submission_Standards` document.
- **Metadata_Base:** the metadata table where the information needed to populate each document table can be found.
- **IndexField:** the field of the table listed in **Metadata_Base** which contains a numeric ordering of fields. (In the case of document tables 2 to 4, which use `TEIS_Metadata_Tables` as the metadata base, the **IndexField** is **Subm_Type** which is a custom field created only for the purpose of specifying the order in which feature classes should be listed in the document tables.
- **Filter:** This is currently only needed for the document tables that use `TEIS_Metadata_Tables` as the metadata base. It contains search strings that are used by the `CreateSubmissionStandardsDocTables.py` script to determine which feature classes should be listed in which document table, based on the value found in the **Subm_Type** field in the `TEIS_Metadata_Tables` table for each feature class.
- **Header_n:** The *n*th column heading as it should appear in the document table.
- **Data_Field_n:** The field of the table listed in **Metadata_Base** that contains the text that should appear in the *n*th column in each row of the document table.
 - One special case is the use of a comma in the **Data_Field_n** fields (for example “`FGDB_Alias,FGDB_Name`”). The existence of a comma is detected by the `CreateSubmissionStandardsDocTables.py` script, which will then use the text from both fields in a single column of the document table. The script was written to combine these two text strings in the format “`FGDB_Alias (FGDB_Name)`” in the output.

Three types of document tables are present in the March 31, 2011 version of the `TEIS_Digital_Submissions_Standards` document:

1. Those that list and describe feature classes, using three columns (document tables 2 to 4);
2. those that list and describe the attributes of the feature classes, using eight columns (document tables 5 to 17);
3. and those that list and describe the contents of the Domain tables, using three columns (document tables 18 to 73).

The CreateSubmissionStandardsDocTables.py script creates one CSV table for each of the document tables 2 to 73. Only the first two types of document tables require any entries in the TEIS_Metadata_Report_Tables table; text for document tables of the third type is extracted by the script directly from the Domain tables in the Tools_and_Templates file geodatabase.

2.6.2.4 Maintaining the TEIS_Metadata_Tables table

This table houses information about the tables in the TEIS Environment, such as text description, dates of creation, modification and removal, feature class type, spatial reference, and environment in which the table is found and/or created. The table structure is simple enough to be understood at a glance, and its contents can be easily updated using the table editing functions in ArcMap. Information in this table is used by the automated processes for generating feature class and table templates (see section 3.2.15).

2.6.2.5 Maintaining the Domain tables

The Tools_and_Templates.gdb file geodatabase contains many tables with names prefixed by “Domain_”. These domain tables are read by the ValidateDomains.py script to validate values in the operational data. The TEIS_Metadata_Items table contains a field named Item_Domain_Name; this field maps each domain table to a field in the operational data tables.

If values need to be added to, deleted from, or changed in a domain table, it is a simple matter of editing the domain table using the table editing functions of ArcMap.

To create a new domain table, make a copy of the Template_DOMAIN table in the Tools_and_Templates.gdb file geodatabase, and rename this copy to Domain_*newdomain* (where *newdomain* is some name that refers to the type of values that the new table will contain).

This table schema found in Template_DOMAIN is not set in stone; for example, the CODE field in the template is set to Text (length 2), but this may be altered as required. In this case, you cannot use the template to create the new domain table, but instead you must create the new domain table from scratch. To do this, start ArcCatalog, right-click on the Tools_and_Templates.gdb file geodatabase, select New, and select Table. For both the name and alias of the new table, enter Domain_*newdomain* (where *newdomain* is some name that refers to the type of values that the new table will contain). Select Default for the Configuration Keyword. In the list of field names, which currently only consists of OBJECTID, add the following four fields:

Field Name	Data Type	Alias	Allow NULL values	Default Value	Length
Code	Text	Code	Yes	(blank)	(as needed)
Short_Name	Text	Short_Name	Yes	(blank)	100
Description	Text	Description	Yes	(blank)	500
Notes	Text	Notes	Yes	(blank)	250

Use the table editing tools in ArcMap to populate the new domain table with valid values and text definitions (following the pattern of the existing domain tables). Then, use the table editing tools in ArcMap to edit the contents of the TEIS_Metadata_Items table. Find the records in this table that reference the items (ie. the fields in the operational data tables) to which the new domain table should apply. In these records in TEIS_Metadata_Items, update the value in the Item_Domain_Name field with the name of the new domain table (minus the Domain_ prefix).

3.0 Tools for TEIS Environment Workflows

3.1 Windows Command Prompt

TEIS Environment data processing tools take the form of scripts written in Python, which is the native scripting language in ArcGIS 9.3 and up. Script files have a **.py** extension and reside in the TEIS Environment’s **Scripts** folder. The scripts are best executed at the Windows Command Prompt.

3.1.1 Opening the Command Prompt Window

The Windows Command Prompt can be invoked several ways. A shortcut to the executable (which is normally C:\Windows\System32\cmd.exe) is usually found in the **Start** menu under **Accessories**. You can also find it by clicking the **Start** button and typing “cmd” in the “Search programs and files” textbox that appears.

3.1.2 Command Prompt settings

It is a good idea to modify some of the default settings for the window that the Command Prompt appears in. The settings are accessible by clicking the small “C:\” icon to the left of the words “Command Prompt” in the title bar of the Command Prompt window. A menu should appear. Select the last option, “Properties”. See screen capture below.

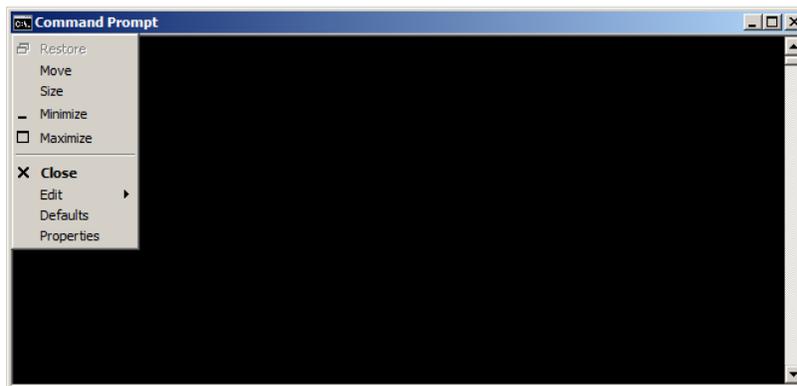
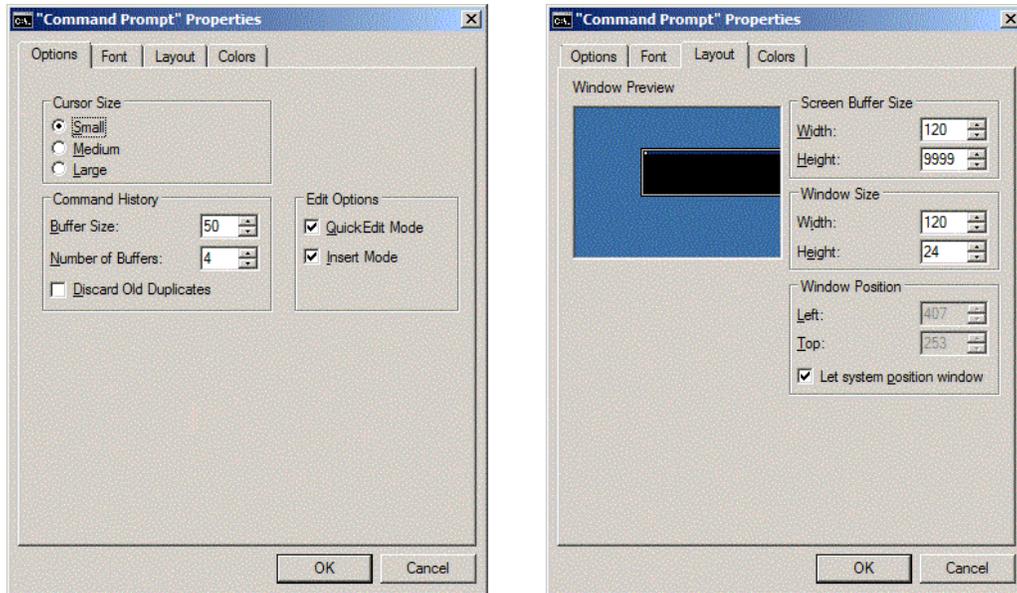


Figure 1: Entering the Properties dialog for the Command Prompt window.

In the Properties dialog box that appears, with the Options tab selected, ensure that there are check marks in both the “QuickEdit Mode” and “Insert Mode” checkboxes. With the Layout tab selected, change the Screen Buffer Size Height to 9999. Change the Screen Buffer Size Width to 120, and the Window Size Width to 120. Click OK. (If the window is now too wide for your monitor display, you can go into Properties again, and change both the Screen Buffer Size Width and the Window Size Width back to 80 or to another value.) See screen capture below.



3.1.3 Keyboard Functions

To enter a command at the command prompt, the command can be typed directly from the keyboard. You may also use the up-arrow and down-arrow keys on the keyboard to scroll through commands that have been run previously.

The Ctrl-C (Copy) and Ctrl-V (Paste) functions that are used in most other Windows applications are both performed by a right-click in the Command Prompt window. Text in the Command Prompt window can be selected by dragging the mouse over a line or block of text. To Copy the selected text to the clipboard, simply right-click anywhere in the Command Prompt window. That selected text will now be available for pasting elsewhere. If no text is currently selected in the Command Prompt window, then a right-click will Paste the current text contents of the clipboard at the current text cursor location.

In the Command Prompt window, the Ctrl-C key sequence does not perform a Copy; it is actually used to cancel a script that is currently in progress.

The Pause key (on the keyboard) can be used to pause a script or other process in progress. To resume after pausing, press any other keyboard key, such as the Enter key or the space bar. Dragging the mouse cursor to select a block of text will also pause a running script; to resume from this kind of pause, right-click on the mouse somewhere inside the Command Prompt window (which will Copy the selected block of text) or press the Esc key.

3.1.4 Executing Scripts

Each script tool described in the following sections can be executed by typing the script's filename, including the **.py** extension, at the Command Prompt. If parameters are required, they must be entered on the same command line after the script name and separated by a single space. For this to work properly, Windows Explorer needs to be set to associate the **.py** file extension with the Python.exe application which should have been installed with ArcGIS Desktop. To ensure that this file association is set properly, start Windows Explorer and navigate to the Scripts folder. Right-click on one of the **.py** script files, select **Open with**, and select **Choose default program**. The Python application should appear in the list of recommended programs. Select Python, put a tick mark in the "Always use the selected program to open this kind of file" checkbox, and click OK.

It is a good idea to include full path names anytime you enter the name of a script or a parameter at the Command Prompt. It is also a good idea not to use spaces in the names of any folders where TEIS Environment data or scripts are stored.

The TEI_Toolbox is included in the TEIS Environment and in the Contractor Package as a more user-friendly method of running some of the more commonly-used scripts from within ArcCatalog or ArcMap rather than at the Command Prompt.

3.1.5 Script Log Files

All TEIS Environment scripts create log files in the Scripts\log folder. These log files, which are simple text files, can be a good source of information for troubleshooting and debugging.

The filenames of the logs for each script are a combination of the filename of the script, plus the name of the computer on which it was executed, plus a .LOG extension. For example: VALIDATE_DOMAINS.PY.GISSERV1.LOG is the log of the ValidateDomains tool run on a computer called GISSERV1. All log files are cumulative, ie. they contain information from every time the script was ever executed. The first entry on every log file line is the initial date and time that the script was run, followed by the time of the logged event in brackets. Log file entries are essentially the same as what is reported in the Command Prompt window during script execution.

3.2 Scripts for Maintenance and Export of Full TEIS Environment

Subsection numbers here refer to tag numbers indicated on individual processes in the workflow diagrams of the TEIS_Environment_Overview.pdf document.

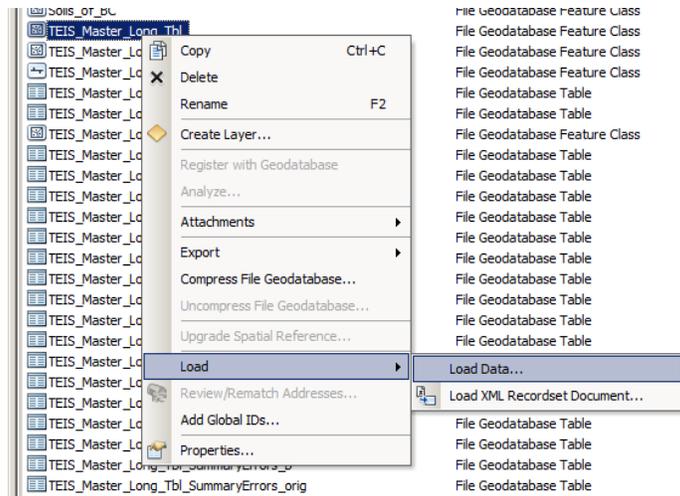
3.2.1 Simple Data Loader

This is not a script, but a function built into ArcCatalog application.

When new mapping project data is received, or when legacy data has been newly processed to match the TEIS Environment schema, use ArcCatalog's Simple Data Loader function to add all of the records (features) from the newly-arrived data to the ends of the appropriate Master tables and feature classes of the full TEIS Environment.

After adding new data to the Master tables and feature classes, derived datasets (error tables, project boundaries, short table) must be regenerated, which requires significant server processing time (three weeks at the very least for the full dataset with approximately 24,000,000 records). For this reason, it is recommended that new data be added to the Master tables in batches a few times per year at most.

In the ArcCatalog application, right-click the Master table or feature class to which the new data will be added. Select the **Load** item, and select **Load Data**. See the screen capture below.



In the first step, select the data source(s) that are to be loaded into the target feature class or table. After clicking Next twice more (accepting the default options), use the following step to specify which field in the source table should be used to populate each field in the target table. In the steps that follow, the options can safely be left as they are. After the tool is completed, check the number of records in your target table to ensure that the records have been added as expected.

3.2.2 AddIndexes.py

Purpose: To create any missing field (attribute) indexes. Indexes can improve overall file geodatabase performance.

Description: This tool creates any missing indexes on attributes or fields in feature classes and tables within the TEIS Environment. Spatial indexes are not modified. The file geodatabases are then compacted. This tool should be run periodically after major changes are made, such as the loading of additional data.

Parameters: None.

Usage example: X:\fullpath\AddIndexes.py

Output: Indexes are created on all existing feature classes and tables found in the folder structure in which the script is located. No new outputs are created.

3.2.3 RepairGeometry.py

Purpose: To run standard geometry repairs, as well as fix inverted polygons that cause errors when transferring data between FGDB and ArcSDE formats using FME.

Description: This script checks every polygon feature in the input feature class for “inversion”, where the polygon contains an interior ring that shares a vertex with the outer shell. Such a polygon is valid when stored this way in a file geodatabase, but it needs to be stored a different way, with the shape of the “hole” represented as part of the outer shell, in order to be considered valid in an ArcSDE geodatabase (such as the BCGW). This script exports each problem polygon to a temporary feature class, performs a Union tool on it which makes the geometry SDE-valid, and overwrites the geometry of the original polygon with the fixed geometry from the output of the Union tool.

The script also runs the ArcGIS RepairGeometry tool with the DELETE_NULL option, which fixes other common types of geometry errors and results in the deletion of any records with null geometry.

Parameters: Long Table feature class.

Usage example: X:\fullpath\StandardizeNoData.py X:\fullpath\TEIS_Master_Long_Tbl

Output: The geometry attributes of the input feature class or table are modified. No new outputs are created.

3.2.4 CheckUniqueTEIS_ID.py

Purpose: To apply uniqueness to the primary key (the TEIS_ID field) of the detailed polygon dataset (TEIS_Master_Long_Tbl or TEI_Long_Tbl).

Description: This tool reads the values in the TEIS_ID field and reports whether or not any duplicate values exist. Optionally, it writes a new unique value to the TEIS_ID field for any record where the TEIS_ID value is currently found to be duplicate or null.

Parameters: Long Table feature class with a TEIS_ID field, and the string “Renumber” (if the user wishes any nonunique or null TEIS_ID values to be assigned a new unique value) or “NoRenumber” (if the user only wishes an on-screen report of whether or not all TEIS_ID values are found to be unique).

Usage example 1: X:\fullpath\CheckUniqueTEIS_ID.py X:\fullpath\TEIS_Master_Long_Tbl
Renumber

Usage example 2: X:\fullpath\CheckUniqueTEIS_ID.py X:\fullpath\TEIS_Master_Long_Tbl
NoRenumber

Output: The input feature class is modified. No new outputs are created.

3.2.5 StandardizeNoData.py

Purpose: To standardize the representation of null values throughout an input table.

Description: This script modifies the input table by setting all NULL and single space values in text fields to an empty string, and all invalid zero values in numeric fields to NULL. This standardization of NULL values is necessary due to the different ways they are represented in source data of varying formats (coverages, shapefiles etc.).

Parameters: Long Table feature class or Project Details table.

Usage example: X:\fullpath\StandardizeNoData.py X:\fullpath\TEIS_Master_Long_Tbl

Output: The attributes of the input feature class or table are modified. No new outputs are created.

3.2.6 ValidateData.py

Purpose: To validate the attributes and geometry in the Long Table feature class and produce tables of errors for use in the QA process and/or for population of the POLY_COM (comments) field in the Long Table and the ERROR_SUM field in the Short Table.

Description: This script scans a Long Table feature class and reports data errors of various types. It will check that the values found in each field conform to the domain, range and critical range rules set for the field in the

TEIS_Metadata_Items table. It will also check that all mandatory fields (determined by the project type) are populated. And it will run the ArcGIS system tool CheckGeometry to identify any geometry errors. These errors are report in the output tables listed below, along with the TEIS_ID value of the record in which each error occurs.

This script will detect whether the local machine has ArcGIS 10.1 installed, and if so, it will use the new Data Access Module cursor object which operates significantly more quickly than the cursor object available in ArcGIS 10.0 and earlier.

Parameters: Three strings: the feature class to be validated, MandatoryFields or NoMandatoryFields, and CheckGeometry or NoCheckGeometry. The user may not be interested in checking that all Mandatory Fields are populated; for many legacy datasets, these fields are not and cannot be populated, so reporting these errors may not be useful. It is therefore presented as an option. Running the CheckGeometry tool on very large datasets can require several hours, and the user may already be certain that there are no geometry errors, so it is presented as an option as well.

Usage example 1: X:\fullpath\ValidateData.py X:\fullpath\TEIS_Master_Long_Tbl
NoMandatoryFields NoCheckGeometry

Usage example 2: X:\fullpath\ValidateData.py X:\fullpath\TEIS_Master_Long_Tbl
MandatoryFields CheckGeometry

Output: Seven tables in the same file geodatabase as the input feature class:

- TEIS_Master_Long_Tbl_CriticalRangeErrors
- TEIS_Master_Long_Tbl_DomainErrors
- TEIS_Master_Long_Tbl_GeometryErrors (created only if CheckGeometry parameter specified)
- TEIS_Master_Long_Tbl_MandatoryFieldErrors (created only if MandatoryFields parameter specified)
- TEIS_Master_Long_Tbl_RangeErrors
- TEIS_Master_Long_Tbl_RowErrors
- TEIS_Master_Long_Tbl_SummaryErrors

3.2.7 CreateProjectBoundaries.py

Purpose: To create the polygon feature class TEIS_Master_Project_Boundaries which illustrates the overall boundaries of each mapping project (ie. one polygon feature per BAPID). This feature class is later used by CreateProjectBoundariesForWebsite.py (see section 3.2.10) to create feature classes displayed on project maps on the MoE website, and by ExportToBCGWStaging.py (see section 3.2.8) to create an updated version of the BCGW TEI layers.

Description: This script makes repeated runs of the Dissolve tool on subsets (“slices”) of 100,000 records. The outputs of each of these Dissolve operations is saved in the Scratch.gdb file geodatabase. After all of the slices have been dissolved, these outputs are merged together and another, final Dissolve is run to create the TEIS_Master_Project_Boundaries feature class in Operational_Data.gdb.

Parameters: The Long Table feature class and the BAPID Tracker “tblProjects” table.

Usage example: X:\fullpath\CreateProjectBoundaries.py X:\fullpath\TEIS_Master_Long_Tbl
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: TEIS_Master_Project_Boundaries feature class in the Operational_Data.gdb file geodatabase.

3.2.8 CreateShortTable.py

Purpose: To create the detailed polygons feature class with geometry identical to the Long Table feature class, but with a summarized attribute table with individual codes concatenated together into map label strings that ecologists are accustomed to reading. This Short Table feature class is later read by the ExportToBCGWStaging.py script (see section 3.2.8) to create data in a format that can be loaded into the BCGW SDE geodatabase.

Description: This script reads the Long Table feature class record by record, reformats the attribute values to match the Short Table attribute table schema (according to logic hard-coded in the script), and inserts each reformatted record into the new Short Table feature class. Some attributes are read from the Project Details and the Project Tracker tables and written to the Short Table records according to their BAPID values. The script requires the existence of a Project Details table but not the existence of the Project Tracker table (which only exists in the master TEIS Environment).

This script will detect whether the local machine has ArcGIS 10.1 installed, and if so, it will use the new Data Access Module cursor object which operates significantly more quickly than the cursor object available in ArcGIS 10.0 and earlier.

Parameters: The Long Table feature class and the BAPID Tracker “tblProjects” table.

Usage example: X:\fullpath\CreateShortTable.py X:\fullpath\TEIS_Master_Long_Tbl
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: TEIS_Master_Short_Table feature class in the Operational_Data.gdb file geodatabase.

3.2.9 ExportToBCGWStaging.py

Purpose: To create feature classes based on TEIS_Master_Project_Boundaries (created in step 3.2.6) and TEIS_Master_Short_Table (created in step 3.2.7) that can be loaded into the BCGW ArcSDE geodatabase as an update to the following BCGW feature classes:

- STE_PEM_20K_POLYS_SP
- STE_PEM_50K_POLYS_SP
- STE_PEM_ATTRIBUTE_POLYS_SP
- STE_SEI_ATTRIBUTE_POLYS_SP
- STE_TEI_PROJECT_BOUNDARIES_SP
- STE_TEM_20K_POLYS_SP
- STE_TEM_50K_POLYS_SP
- STE_TEM_ATTRIBUTE_POLYS_SP
- STE_TEM_GT_20K_POLYS_SP
- STE_TER_ATTRIBUTE_POLYS_SP
- STE_TER_BIOTERRAIN_POLYS_SP
- STE_TER_INVENTORY_POLYS_SP
- STE_TER_STABILITY_POLYS_SP

Description: Templates for the feature classes listed above are copied from the Tools_and_Templates.gdb file geodatabase into the STE_TEI_BCGW_PEM.gdb and STE_TEI_BCGW_TEI.gdb file geodatabases as appropriate. Data is appended to the empty templates from the two source feature classes, TEIS_Master_Short_Tbl and

TEIS_Master_Project_Boundaries, excluding records belonging to sensitive BAPIDs. Records that belong to sensitive BAPIDs are saved in a third file geodatabase, STE_TEI_SENSITIVE.gdb.

Geometry issues are then addressed. The SimplifyPolygon tool is run on the STE_TEI_PROJECT_BOUNDARIES_SP feature class to reduce the number of vertices, as the high complexity of the geometry in this layer has proven impossible to transfer into the BCGW SDE geodatabase without errors. Also, the RepairGeometry tool is run on each feature class.

Parameters: The BAPID Tracker “tblProjects” table.

Usage example: X:\fullpath\ExportToBCGWStaging.py
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: STE_TEI_BCGW_PEM.gdb and STE_TEI_BCGW_TEI.gdb file geodatabases containing the feature classes listed above, and STE_TEI_SENSITIVE.gdb file geodatabase containing the feature classes TEI_ATTRIBUTE_POLYS_SENSITIVE and TEI_PROJECT_BOUNDARIES_SENSITIVE.

3.2.10 ExportToTEIShare.py

Purpose: To create the TEI_Share folder which contains a copy of much of the TEIS Environment data as well as files and folders as specified in the TEIS_Metadata_Files table in the TEI_Share field. (You may wish to run the UpdateTEISMetadataFiles.py script (see section 3.2.18) to update the table before running this script.)

Description: A new empty TEI_Share folder is created (after deleting the old one if one exists, along with all of its contents). Then the script creates a new file geodatabase named TEI_Data and creates the following datasets, which generally are copies of the corresponding feature classes in Operational_Data.gdb, but excluding records belonging to sensitive BAPIDs (as indicated by the SENSITIVE field in the TEIS_Project_Tracker table):

- TEI_Long_Tbl
- TEI_Long_Tbl_SummaryErrors
- TEI_Project_Boundaries
- TEI_Project_Boundaries_Sensitive
- TEI_Sample_Site_Lines
- TEI_Sample_Site_Points
- TEI_Short_Tbl
- TEI_Symbols_Lines
- TEI_Symbols_Points
- TEI_Symbols_Polys
- TEI_Symbols_Polys_Arcs
- TEIS_Master_Project_Details
- TEIS_Project_Tracker
- TEIS_Related_BAPIDs
- User_Defined_Fields_*

The script also copies the Tools_and_Templates.gdb file geodatabase into the TEI_Share folder, renaming it Metadata_and_Templates.gdb. Finally, it copies files and folder from the TEIS_Environment folder structure into the TEI_Share folder according to the values in the TEI_Share field of the TEIS_Metadata_Files table.

Parameters: The BAPID Tracker “tblProjects” table.

Usage example: X:\fullpath\ExportToTEIShare.py
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: TEI_Share folder at the same folder level as the Operational.gdb file geodatabase.

3.2.11 CreateProjectBoundariesForWebsite.py

Purpose: To update the feature classes that are used to create map images of project boundaries for the MoE website.

Description: This script extracts subsets of the current TEIS_Master_Project_Boundaries feature class, one per broad project type found in the LUT_Broad_Project_Types table (currently PEM, SEI, SL, TEM, TER, and WLD). For each project type, it creates two boundary layers in the PDF_Map_Data.gdb file geodatabase: one whose records are sorted by project boundary size (so that smaller projects are drawn after, and overlaid upon, larger projects), and another sorted by BAPID number (for sorted attribute display in the resulting PDFs' Model Tree view). These 14 feature classes are referenced in the MXD files in the TEIS Environment's Maps folder.

Parameters: The BAPID Tracker "tblProjects" table.

Usage example: X:\fullpath\CreateBoundariesForWebsite.py
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: 12 polygon feature classes in the PDF_Map_Data.gdb file geodatabase:

- TEIS_Project_Boundaries_PDF_PEM_Area_Sort
- TEIS_Project_Boundaries_PDF_PEM_BAPID_Sort
- TEIS_Project_Boundaries_PDF_SEI_Area_Sort
- TEIS_Project_Boundaries_PDF_SEI_BAPID_Sort
- TEIS_Project_Boundaries_PDF_SL_Area_Sort
- TEIS_Project_Boundaries_PDF_SL_BAPID_Sort
- TEIS_Project_Boundaries_PDF_TEM_Area_Sort
- TEIS_Project_Boundaries_PDF_TEM_BAPID_Sort
- TEIS_Project_Boundaries_PDF_TER_Area_Sort
- TEIS_Project_Boundaries_PDF_TER_BAPID_Sort
- TEIS_Project_Boundaries_PDF_WLD_Area_Sort
- TEIS_Project_Boundaries_PDF_WLD_BAPID_Sort

3.2.12 CreatePDFsForWebsite.py

Purpose: To automatically generate map documents in PDF format and map images in PNG format for use on the MoE website.

Description: Runs a series of commands to export the predefined ArcMap MXD project files (found in the Maps folder of the operational TEIS Environment) to PDF files (in the folder Maps\Output_PDF) and PNG files (in the folder Maps\Output_Images) which can then be copied to the appropriate location on the MoE webspace. The script will loop through each of the broad project types defined in the LUT_Broad_Project_Types table, and it will automatically modify the map title text and turn layers on and off as appropriate.

Note: This script can **only** be run on a machine with **ArcGIS 10.1 or later** installed. It uses a function to programmatically "Add All Values" to the layer symbology definition which not available in ArcGIS 10.0 or earlier.

Parameters: None.

Usage example: X:\fullpath\CreatePDFsForWebsite.py

Output: 20 PDF files for each broad project type (120 PDF files in total) in the Maps\Output_PDF folder, and one PNG file for each broad project type (6 PNG files in total) in the Maps\Output_Images folder.

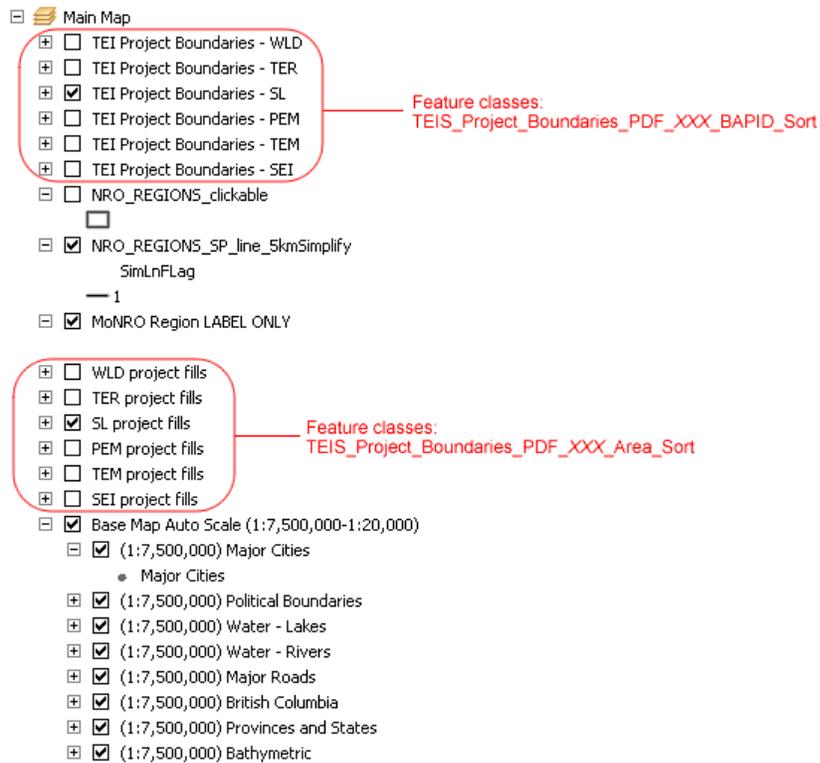
3.2.12.1 Maintaining the MXD files

There are three MXD files, all found in the **Maps** folder, which are used by the CreatePDFsForWebsite.py script:

- TEI_all_PDF_ArchE.mxd
- TEI_all_PDF_Letter.mxd
- TEI_all_website_front_page_images.mxd

The most likely reason for changing these MXD files would be a change in the classification of Broad Project Types, which are currently PEM, SEI, SL, TEM, TER and WLD. If these codes are changed, then modifications would have been made to the LUT_Broad_Project_Types table (found in the Tools_and_Templates.gdb file geodatabase), and new Project Boundary feature classes will need to be created in the PDF_Maps.gdb file geodatabase by running the CreateProjectBoundariesForWebsite.py script (see section 3.2.11). Once these feature classes have been created, the layers in the MXD files will need to be changed to match these new Broad Project Types.

For each broad project type 'XXX', there are two feature classes required, named **TEIS_Project_Boundaries_PDF_XXX_BAPID_Sort** and **TEIS_Project_Boundaries_PDF_XXX_Area_Sort**. The former are listed near the top of the MXD table of contents and named "TEI Project Boundaries - XXX", and the latter are listed near the middle of the MXD table of contents and named "XXX project fills". Insert any new broad project type layers into the MXD table of contents following the same pattern. It doesn't matter which layers are turned on or off when the MXD is saved, as the script that creates the output maps will turn layers on and off programmatically as appropriate. See the image of the MXD table of contents below.



3.2.13 Copy Updated Images and PDFs to MoE Webpace

After updating the map data and map outputs (PDF files in the Maps\Output_PDF folder, and PNG files in the Maps\Output_Images folder) in stages 3.2.11 to 3.2.12, the outputs can be copied into place on the MoE web server to replace the newly outdated versions. This is a simple process of copying files. Consult with the MoE website administrator and/or the website developers if you do not know the correct directory paths.

3.2.14 ExportToContractorPackage.py

Purpose: To build a new contractor package based on the current scripts, templates, and resources in the TEIS_Environment.

Description: Copies the necessary directories, documentation, templates, scripts, and other resources into a dated stamped Contractor Package directory. The **Cont_Pkg** field in the TEIS_Metadata_Files table (see section 2.6.2.2) in the Tools_and_Templates geodatabase defines which files, folders and geodatabases are copied from the operational TEIS Environment to the Contractor Package. (You may wish to run the UpdateTEISMetadataFiles.py script (see section 3.2.18) to update the table before running this script.)

A special case is made for the Operational_Data.gdb file geodatabase. It is not copied in its entirety into the Contractor Package, but rather a Contractors' version of it is built, containing only templates for the Contractor versions of feature class templates.

Another special case is made for the TEIS_Metadata_Items table in the Tools_and_Templates.gdb file geodatabase; the Contractors' version of this table will be stripped of extra fields that are not required by the Contractor Package tools. Specifically, all fields are removed whose names begin with 'SDE', 'E00', 'RISC', 'STE', 'Soils', 'TEI', or 'TEIS', as well as most of the fields whose names begin with 'Oracle'.

Parameters: None.

Usage example: `X:\fullpath\ExportToContractorPackage.py`

Output: A new folder named `Contractor_Package_YYYYMMDD` (where `YYYYMMDD` describes the current date) under the TEIS Environment main folder. If such a folder named with the current date already exists, another new folder is created, named `Contractor_Package_YYYYMMDD_1`, and if that already exists, another new folder is created, named `Contractor_Package_YYYYMMDD_2`, and so on.

3.2.15 CreateAllTemplatesAndMetadataReports.py

Purpose: To create new feature class and table templates based on the current contents of the Metadata tables in the `Tools_and_Templates.gdb` file geodatabase.

Description: This tool loads information from the `TEIS_Metadata_Tables` and `TEIS_Metadata_Items` tables in the `Tools_and_Templates.gdb` file geodatabase, and creates new templates and HTML documentation. Each existing template is first copied into the `Past_Templates.gdb` file geodatabase in the Archives folder before the template in `Tools_and_Templates.gdb` file geodatabase is overwritten. The HTML documents (in the `Documentation\Metadata` folder) are overwritten without archiving the existing documents.

Note that the TEIS Environment's table schemas, as defined in the `TEIS_Metadata_Tables` and `TEIS_Metadata_Items` tables, are generally fairly static. This script is generally only used after making modifications to these metadata tables. See sections 2.6.2.2 and 2.6.2.4 for notes about the maintenance of the metadata tables.

Parameters: None.

Usage example: `X:\fullpath\CreateAllTemplatesAndMetadataReports.py`

Output: Newly outdated templates moved to the `Past_Templates.gdb` file geodatabase in the Archives folder. New templates created in the `Tools_and_Templates.gdb` file geodatabase. New HTML metadata documents created in the `Documentation\Metadata` folder (overwriting the existing documents without archiving them).

3.2.16 CreateOneTemplateAndMetadataReport.py

Purpose: To create a specific new feature class or table templates based on the current contents of the Metadata tables in the `Tools_and_Templates.gdb` file geodatabase.

Description: This tool loads information from the `TEIS_Metadata_Tables` and `TEIS_Metadata_Items` tables in the `Tools_and_Templates.gdb` file geodatabase, and creates the new template and HTML document specified by the user. The existing template is first copied into the `Past_Templates.gdb` file geodatabase in the Archives folder before the template in `Tools_and_Templates.gdb` file geodatabase is overwritten. The HTML document (in the `Documentation\Metadata` folder) is overwritten without archiving the existing document.

Note that the TEIS Environment's table schemas, as defined in the `TEIS_Metadata_Tables` and `TEIS_Metadata_Items` tables, are generally fairly static. This script is generally only used after making modifications to these metadata tables. See sections 2.6.2.2 and 2.6.2.4 for notes about the maintenance of the metadata tables.

Parameters: Name of the template to be created. Should match with one of the values found in the Feature_Class field of the TEIS_Metadata_Tables table in the Tools_and_Templates.gdb file geodatabase.

Usage example: X:\fullpath\CreateOneTemplatesAndMetadataReports.py TEIS_Master_Long_Tbl

Output: Newly outdated template moved to the Past_Templates.gdb file geodatabase in the Archives folder. New template created in the Tools_and_Templates.gdb file geodatabase. New HTML metadata document created in the Documentation\Metadata folder.

3.2.17 CreateSubmissionStandardsDocTables.py

Purpose: To partially automate the updating of table contents in the TEIS_Digital_Submission_Standards document in the Documentation folder.

Description: Reads the TEIS_Metadata_Report_Tables, TEIS_Metadata_Tables and TEIS_Metadata_Items tables in the Tools_and_Templates geodatabase in the operational TEIS Environment, then outputs a series of CSV (comma-delimited text) files to the \Documentation\TEI_Digital_Submission_Standards_tables folder. These CSV files can each be opened in Microsoft Excel and their contents copied (as a block of cells) and pasted into the appropriate table of the document, using Microsoft Word.

Note that the ordering, numbering and content of the output CSV files is according to the data in the TEIS_Metadata_Report_Tables; if changes are later made to the structure of the TEIS_Digital_Submissions_Standards document, then changes will be needed in the TEIS_Metadata_Report_Tables table as well.

Parameters: None.

Usage example: X:\fullpath\CreateSubmissionStandardsDocTables.py

Output: CSV files in the Documentation\TEI_Digital_Submission_Standards_tables folder.

3.2.18 UpdateTEISMetadataFilesTable.py

Purpose: To partially automate the updating of table contents in the TEIS_Metadata_Files table in the Tools_and_Templates.gdb file geodatabase.

Description: Reads the folder structure within the TEIS_Environment folder and compares it to the current contents of the TEIS_Metadata_Files table. For any files or folders found that are not entered in the table, a record is added to the table, marked with the current date in the Date_Added field. Any files listed in the table that are not found in the folders are marked with the current date in the Date_Missing field in the table. If any files are found in the folders that were previously marked as missing in the table, the value in the Date_Missing field is removed.

After the script is completed, the user will need to start ArcMap and edit the TEIS_Metadata_Files table to add “Y” and “N” values to the Cont_Pkg, Info_Reqst, TEI_Share and TEI_Data_Zip fields as appropriate for any new records that were added.

Parameters: None.

Usage example: X:\fullpath\UpdateTEISMetadataFiles.py

Output: The TEIS_Metadata_Files table is modified.

3.2.19 CreateDistributionPackages.py

Purpose: To export the TEIS data into smaller subsets in file geodatabase format, for distribution in response to ad hoc data requests.

Description: The script reads the LUT_Distribution_Packages table in the Tools_and_Templates.gdb file geodatabase to determine which BAPIDs will be copied to which distribution package. It creates a new folder named "Distribution_Packages" under the main TEIS Environment folder, and a subfolder for each package. Within each subfolder, it writes shapefiles extracted from the data in the Operational_Data.gdb file geodatabase.

The script will attempt to use the ArcGIS "Compress File Geodatabase Data" tool on each non-empty feature class and table that is created, in order to reduce the amount of disk space required to store the distribution packages. When executed within a Python script, this Compress tool has proved problematic for unknown reasons, so it may be necessary to compress each feature class or table manually in ArcGIS after the script has completed.

Parameters: The tblProjects table (in the BAPID Tracker database).

Usage example: X:\fullpath\CreateDistributionPackages.py
X:\fullpath\BAPID_Tracker_BE.mdb\tblProjects

Output: The Distribution_Packages folder and its subfolders and contents.

3.3 Scripts for Preparing Legacy Data for Incorporation into the TEIS Environment

Preparation of legacy data for inclusion in the TEIS Environment is normally performed by the Data Custodian. It is a good idea to make a copy of the Contractor Package and perform the data processing in this folder for a "batch" of legacy project data in isolation from other datasets. The Scripts folder in the Contractor Package does not contain all the scripts referenced in this section; simply copy all files from the main TEIS Environment's Scripts folder into the Contractor Package's Scripts folder.

3.3.1 CompareFields.py

Purpose: To provide a quick check of the schema compatibility between an input table and one of the tables or templates in the TEIS Environment.

Description: This script compares the attribute definitions of two feature classes and provides details of any differences.

Parameters: The input table, and the template to which it is to be compared.

Usage example: X:\fullpath\CompareFields.py X:\fullpath\input_table X:\fullpath\template

Output: An on-screen report indicating fields which may be incompatible (when using the ArcCatalog Simple Data Loader; see section 3.2.1) due to differences in alias name, field type, length, scale or precision.

3.3.2 CompareFieldsNoAlias.py

Purpose: To provide a quick check of the schema compatibility between an input table and one of the tables or templates in the TEIS Environment. Similar to CompareFields.py (see section 3.3.1) except that it does not report differences in alias names.

Description: This script compares the attribute definitions of two feature classes and provides details of any differences.

Parameters: The input table, and the template to which it is to be compared.

Usage example: X:\fullpath\CompareFieldsNoAlias.py X:\fullpath\input_table
X:\fullpath\template

Output: An on-screen report indicating fields which may be incompatible (when using the ArcCatalog Simple Data Loader; see section 3.2.1) due to differences in field type, length, scale or precision.

3.3.3 Simple Data Loader

Refer to section 3.2.1 for a description of ArcCatalog’s Simple Data Loader tool.

3.3.4 AddIndexes.py

Purpose: To create any missing field indexes, which can improve query times and overall file geodatabase performance.

Description: This tool creates any missing indexes on items in feature classes and tables within the TEIS Environment. Spatial indexes are not modified. The file geodatabases are then compacted. This tool should be run periodically after major changes are made (such as a data load).

Parameters: None.

Usage example: X:\fullpath\AddIndexes.py

Output: Indexes are created on existing feature classes and tables. No new outputs are created.

3.3.5 RepairGeometry.py

Purpose: To run standard geometry repairs, as well as fix inverted polygons that cause errors when transferring data between FGDB and ArcSDE formats using FME.

Description: This script checks every polygon feature in the input feature class for “inversion”, where the polygon contains an interior ring that shares a vertex with the outer shell. Such a polygon is valid when stored this way in a file geodatabase, but it needs to be stored a different way, with the shape of the “hole” represented as part of the outer shell, in order to be considered valid in an ArcSDE geodatabase (such as the BCGW). This script exports each problem polygon to a temporary feature class, performs a Union tool on it which makes the geometry SDE-valid, and overwrites the geometry of the original polygon with the fixed geometry from the output of the Union tool.

The script also runs the ArcGIS RepairGeometry tool with the DELETE_NULL option, which fixes other types of geometry errors and results in the deletion of any records with null geometry.

Parameters: Long Table feature class.

Usage example: X:\fullpath\StandardizeNoData.py X:\fullpath\TEIS_Master_Long_Tbl

Output: The geometry attributes of the input feature class or table are modified. No new outputs are created.

3.3.6 CheckUniqueTEIS_ID.py

Purpose: To apply uniqueness to the primary key (the TEIS_ID field) of the detailed polygon dataset (TEIS_Master_Long_Tbl or TEI_Long_Tbl).

Description: This tool reads the values in the TEIS_ID field and reports whether or not any duplicate values exist. Optionally, it writes a new unique value to the TEIS_ID field for any record where the TEIS_ID value is currently found to be duplicate or null.

Parameters: Long Table feature class with a TEIS_ID field, and the string “Renumber” (if the user wishes any nonunique or null TEIS_ID values to be assigned a new unique value) or “NoRenumber” (if the user only wishes an on-screen report of whether or not all TEIS_ID values are found to be unique).

Usage example 1: X:\fullpath\CheckUniqueTEIS_ID.py X:\fullpath\TEIS_Master_Long_Tbl
Renumber

Usage example 2: X:\fullpath\CheckUniqueTEIS_ID.py X:\fullpath\TEIS_Master_Long_Tbl
NoRenumber

Output: The input feature class is modified. No new outputs are created.

3.3.7 StandardizeNoData.py

Purpose: To standardize the representation of null values throughout an input table.

Description: This script modifies the input table by setting all NULL and single space values in text fields to an empty string, and all invalid zero values in numeric fields to NULL.

Parameters: Long Table feature class or Project Details table.

Usage example: X:\fullpath\StandardizeNoData.py X:\fullpath\TEI_Long_Tbl

Output: The input feature class or table is modified. No new outputs are created.

3.3.8 ValidateData.py

Purpose: To validate the attributes and geometry in the Long Table feature class and produce tables of errors for use in the QA process and/or for population of the POLY_COM (comments) field in the Long Table and the ERROR_SUM field in the Short Table.

Description: This script scans a Long Table feature class and reports data errors of various types. It will check that the values found in each field conform to the domain, range and critical range rules set for the field in the TEIS_Metadata_Items table. It will also (optionally) check that all mandatory fields (determined by the project type) are populated. And it will (optionally) run the ArcGIS system tool CheckGeometry to identify any geometry errors. These errors are report in the output tables listed below, along with the TEIS_ID value of the record in which each error occurs.

This script will detect whether the local machine has ArcGIS 10.1 installed, and if so, it will use the new Data Access Module cursor object which operates significantly more quickly than the cursor object available in ArcGIS 10.0 and earlier.

Parameters: Three strings: the feature class to be validated, `MandatoryFields` or `NoMandatoryFields`, and `CheckGeometry` or `NoCheckGeometry`. The user may not be interested in checking that all Mandatory Fields are populated; for many legacy datasets, these fields are not and cannot be populated, so reporting these errors may not be useful. It is therefore presented as an option. Running the `CheckGeometry` tool on very large datasets can require several hours, and the user may already be certain that there are no geometry errors, so it is presented as an option as well.

Usage example 1: `X:\fullpath\ValidateData.py X:\fullpath\TEIS_Master_Long_Tbl NoMandatoryFields NoCheckGeometry`

Usage example 2: `X:\fullpath\ValidateData.py X:\fullpath\TEIS_Master_Long_Tbl MandatoryFields CheckGeometry`

Output: Seven tables in the same file geodatabase as the input feature class:

- `TEIS_Master_Long_Tbl_CriticalRangeErrors`
- `TEIS_Master_Long_Tbl_DomainErrors`
- `TEIS_Master_Long_Tbl_GeometryErrors` (created only if `CheckGeometry` parameter specified)
- `TEIS_Master_Long_Tbl_MandatoryFieldErrors` (created only if `MandatoryFields` parameter specified)
- `TEIS_Master_Long_Tbl_RangeErrors`
- `TEIS_Master_Long_Tbl_RowErrors`
- `TEIS_Master_Long_Tbl_SummaryErrors`

3.3.9 CreateProjectBoundaries.py (Optional for Legacy Data)

Purpose: To create the polygon feature class `TEIS_Master_Project_Boundaries` which illustrates the overall boundaries of each mapping project (BAPID).

Description: This script dissolves boundaries between all pairs of adjacent polygons that share the same BAPID number, resulting in a polygon feature class that depicts the overall boundaries of each mapping project. During this process, all attributes other than BAPID are eliminated. Attributes are then added to the resulting feature class from the Project Details table.

The overall workflow does not require the running of this script at this stage (it is only required in the main operational TEIS Environment at step 3.2.6); however, the output feature class, when viewed in ArcMap, may help to identify geometry and topology issues in the input Long Table feature class, such as gaps between detailed polygons.

Parameters: Long Table feature class AND Project Details table, separated by a space.

Usage example 1: `X:\fullpath\CreateProjectBoundaries.py X:\fullpath\TEI_Long_Tbl X:\fullpath\TEI_Project_Details`

Output: `TEI_Project_Boundaries` feature class in the `Operational_Data.gdb` file geodatabase.

3.3.10 CreateShortTable.py (Optional for Legacy Data)

Purpose: To create the detailed polygons feature class with geometry identical to the Long Table feature class, but with a summarized attribute table with individual codes concatenated together into standardized map label strings.

Description: Using logic embedded in the code (ie. not defined in an external table), this script concatenates the Long Table’s mostly one- and two-character detailed attributes into longer standard map label strings, which ecologists are most accustomed to reading.

The overall workflow does not require the running of this script at this stage (it is only required in the main operational TEIS Environment at step 3.2.5.3); however, having the attributes in the form of map labels may help an ecologist to identify errors or mistakes in the attributes.

Parameters: Long Table feature class AND Project Details table, separated by a space. The *long_tbl_SummaryError* table need not be specified; if it exists (having been created at step 3.3.7), then the script will find and incorporate it.

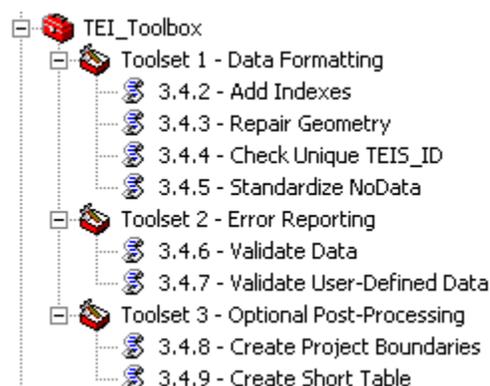
Usage example 1: X:\fullpath\CreateShortTable.py X:\fullpath\TEI_Long_Tbl
X:\fullpath\TEI_Project_Details

Output: TEI_Short_Tbl feature class in the Operational_Data.gdb file geodatabase.

3.4 Scripts and Tools for Preparing New Mapping Project Data for Submission

Preparation of new mapping project data for submission and eventual inclusion in the TEIS Environment is normally performed by the Contractors. Contractors will have only a copy of the Contractor Package folder (created at step 3.2.14), which contains in its root folder the TEI_Toolbox (represented by a file, TEI_Toolbox.tbx). This toolbox provides a more user-friendly method of running the TEIS Environment’s scripts for this workflow from an interface within ArcCatalog or ArcMap (version 9.3.1 and up).

The original TEIS Environment scripts required adaptations to run as “script tools” in the TEI Toolbox. Thus, the script tools in the TEI Toolbox refer to script files in the Scripts folder whose names are prefixed with Tool_. These are the only script files that are included in the Scripts folder of the Contractor Package. They can also be run at the Windows Command Prompt window if desired. (Right-click any tool and select Properties to find the location and name of the script that the tool runs.)



3.4.1 Simple Data Loader

This is not a script or tool, but a function built into ArcCatalog application. This is useful for migrating data from other GIS data formats into standard TEIS Environment FGDB feature class templates, which can be copied from the Tools_and_Templates.gdb file geodatabase. Refer to section 3.2.1 and to the ArcGIS Help documentation for a description of the Simple Data Loader.

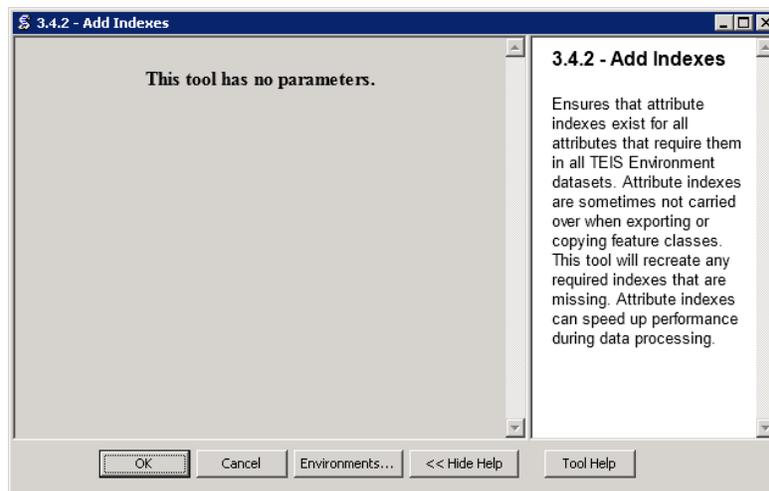
3.4.2 AddIndexes.py

Purpose: To create any missing field (attribute) indexes. Indexes can improve overall file geodatabase performance.

Description: This tool creates any missing indexes on attributes or fields in feature classes and tables within the TEIS Environment. Spatial indexes are not modified. The file geodatabases are then compacted. This tool should be run periodically after major changes are made, such as the loading of additional data.

Parameters: None.

Usage example:



Output: Indexes are created on all existing feature classes and tables found in the folder structure in which the script is located. No new outputs are created.

3.4.3 Repair Geometry tool

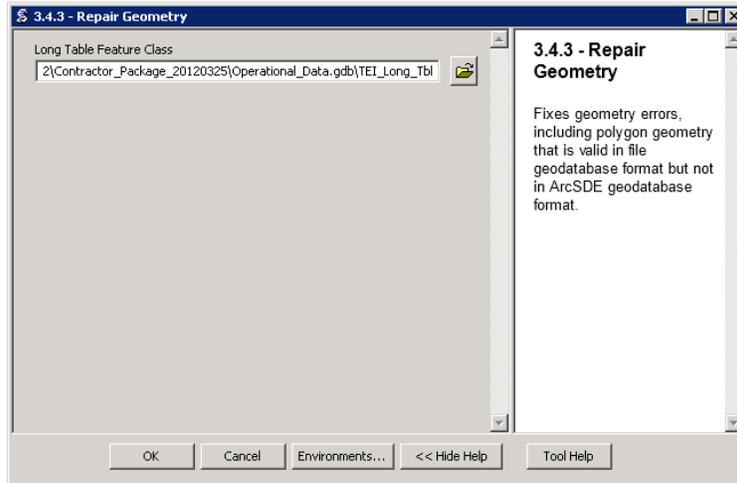
Purpose: To run standard geometry repairs, as well as fix inverted polygons that cause errors when transferring data between FGDB and ArcSDE formats using FME.

Description: This script checks every polygon feature in the input feature class for “inversion”, where the polygon contains an interior ring that shares a vertex with the outer shell. Such a polygon is valid when stored this way in a file geodatabase, but it needs to be stored a different way, with the shape of the “hole” represented as part of the outer shell, in order to be considered valid in an ArcSDE geodatabase (such as the BCGW). This script exports each problem polygon to a temporary feature class, performs a Union tool on it which makes the geometry SDE-valid, and overwrites the geometry of the original polygon with the fixed geometry from the output of the Union tool.

The script also runs the ArcGIS RepairGeometry tool with the DELETE_NULL option, which fixes other types of geometry errors and results in the deletion of any records with null geometry.

Parameters: Long Table feature class.

Usage example:



Output: The geometry attributes of the input feature class or table are modified. No new outputs are created.

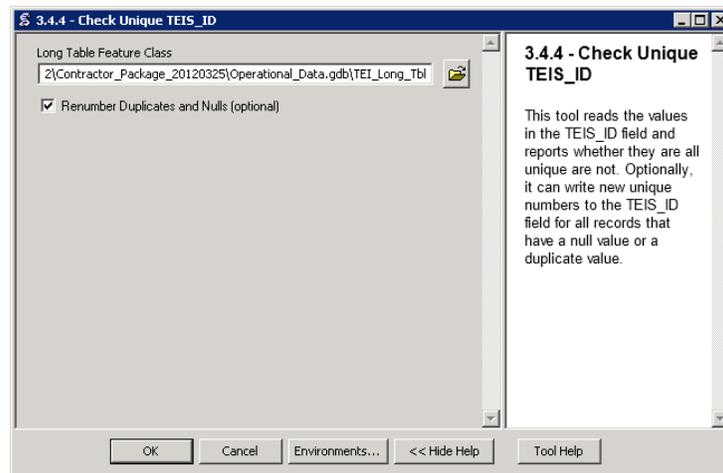
3.4.4 Check Unique TEIS_ID tool

Purpose: To maintain the uniqueness of the values in the TEIS_ID field.

Description: This tool reads the values in the TEIS_ID field and reports whether they are unique or not. Optionally, it can write a unique number to the TEIS_ID field for any record that has a null value or a duplicate value.

Parameters: Long Table feature class with a TEIS_ID field, and a checkbox to indicate whether duplicate and null values should be overwritten with new unique values.

Usage example:



Output: The input feature class is modified. No new outputs are created.

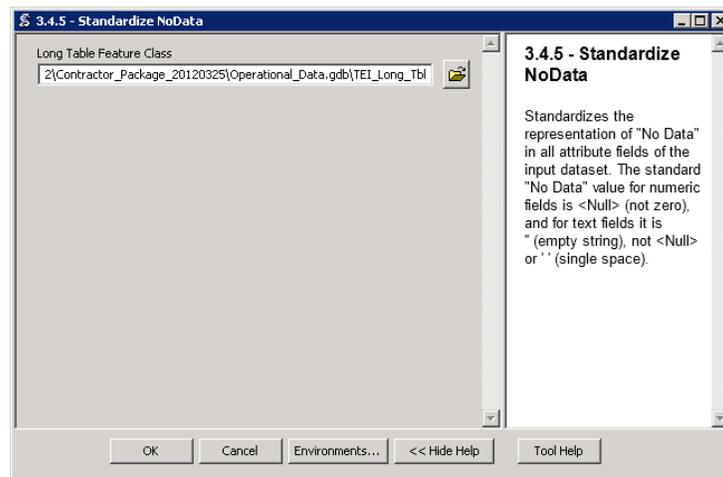
3.4.5 Standardize NoData tool

Purpose: To standardize the representation of null values throughout an input table.

Description: This tool modifies the input table by setting all NULL and single space values in text fields to an empty string, and all invalid zero values in numeric fields to NULL.

Parameters: Long Table feature class or Project Details table.

Usage example:



Output: The input feature class or table is modified. No new outputs are created.

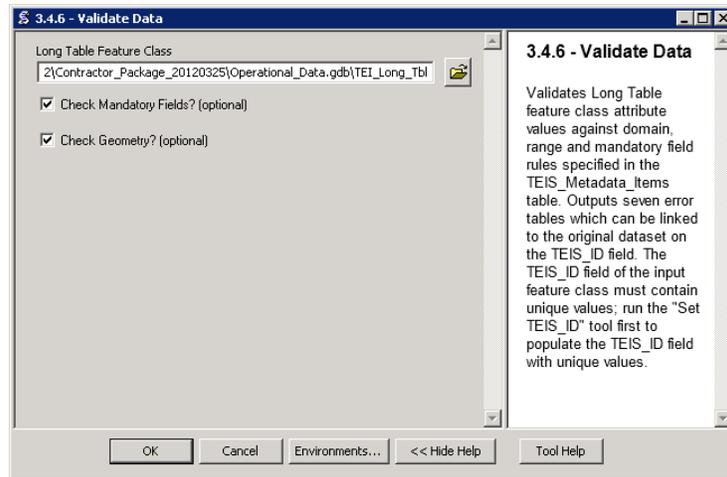
3.4.6 Validate Data tool

Purpose: To list all values in an input table that are in violation of the attribute domains and ranges and cross-field validation rules defined in the TEIS Environment.

Description: The Validate Data tool uses the TEIS_Metadata_* and Domain_* tables in the Tools_and_Templates.gdb file geodatabase to validate an input feature class/table against defined lists and ranges of possible valid attribute values. There is also logic embedded in the script's code for applying cross-field validation rules.

Parameters: Long Table feature class or Project Details table, and checkbox options for validating mandatory fields and geometry (which can be slow for very large datasets).

Usage example:



Output: Seven new tables in the Operational_Data.gdb file geodatabase, listing the errors found in the input table:

- *input_table_name_CriticalRangeErrors*
- *input_table_name_DomainErrors*
- *input_table_name_GeometryErrors* (only if the “Check Geometry?” checkbox was ticked)
- *input_table_name_MandatoryFieldErrors* (only if the “Check Mandatory Fields?” checkbox was ticked)
- *input_table_name_RangeErrors*
- *input_table_name_RowErrors*
- *input_table_name_SummaryErrors*

In each output table, the FEATURE_ID field should be used to link to the TEIS_ID field of the of the Long Table feature class input, or to the OBJECTID field of a Project Details table input. If geometry errors are found in individual features, view each of these features in ArcMap to determine whether it should be repaired manually using the editing tools. Use the Repair Geometry tool in the ArcToolbox if necessary. See the ArcGIS Desktop Help for the Repair Geometry tool to see a list of all possible geometry problems and fixes.

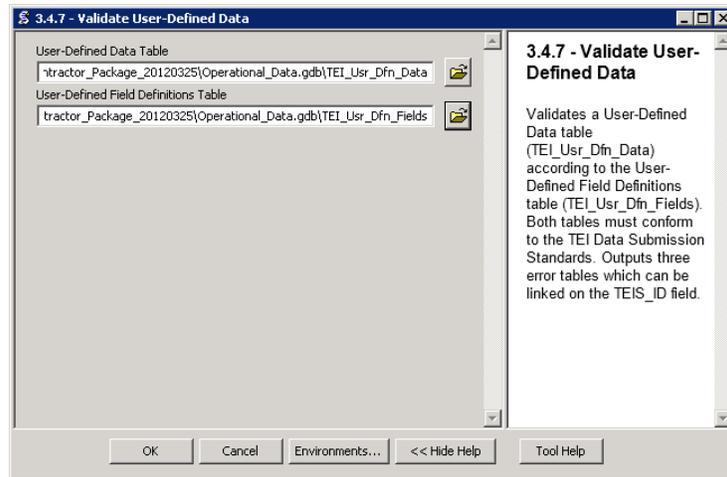
3.4.7 Validate User-Defined Data tool

Purpose: To validate user-defined data according to the user-defined domains and ranges.

Description: This tool can be used to validate the values in the TEI_Usr_Dfn_Data table against the domains and ranges as defined in TEI_Usr_Dfn_Fields table. The user will have populated both tables before executing this tool. This structure for storing and submitting user-defined data is new in this version of the TEIS Environment, and this tool for validating user-defined data is still in a testing phase.

Parameters: Long Table feature class or Project Details table.

Usage example:



Output: Three new tables: TEI_Usr_Dfn_Data_DomainErrors, TEI_Usr_Dfn_Data_MandatoryFieldErrors, and TEI_Usr_Dfn_Data_RangeErrors, in the Operational_Data.gdb file geodatabase, listing the errors found in the TEI_Usr_Dfn_Data input table. In these output tables, the FEATURE_ID field should be used to link to the TEIS_ID field of the of the TEI_Usr_Dfn_Data table.

3.4.8 Create Project Boundaries tool (Optional for Contractors)

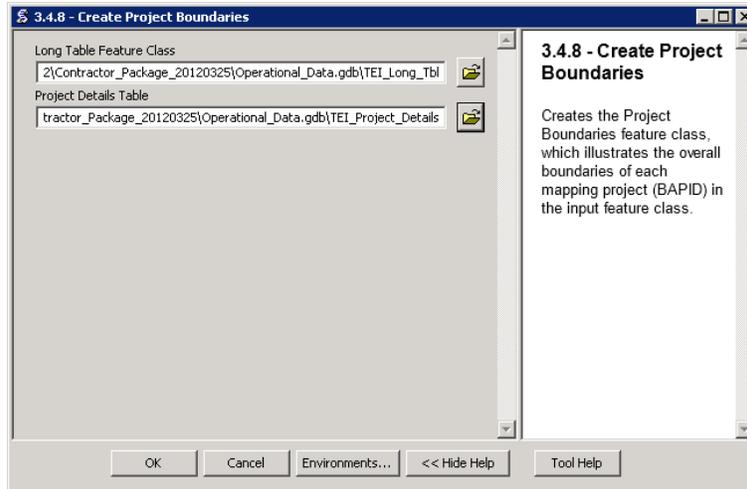
Purpose: To create the polygon feature class TEIS_Master_Project_Boundaries which illustrates the overall boundaries of each mapping project (BAPID).

Description: This tool dissolves boundaries between all pairs of adjacent polygons that share the same BAPID number, resulting in a new polygon feature class that depicts the overall boundaries of each mapping project. During this dissolve process, all attributes other than BAPID are eliminated. Attributes are then added to the resulting feature class from the Project Details table.

The overall workflow does not require the running of this script by Contractors; however, the output feature class, when viewed in ArcMap, may be useful to help to identify geometry and topology issues in the input Long Table feature class, such as gaps between detailed polygons.

Parameters: Long Table feature class AND Project Details table.

Usage example:



Output: TEI_Project_Boundaries feature class in the Operational_Data.gdb file geodatabase.

3.4.9 Create Short Table tool (Optional for Contractors)

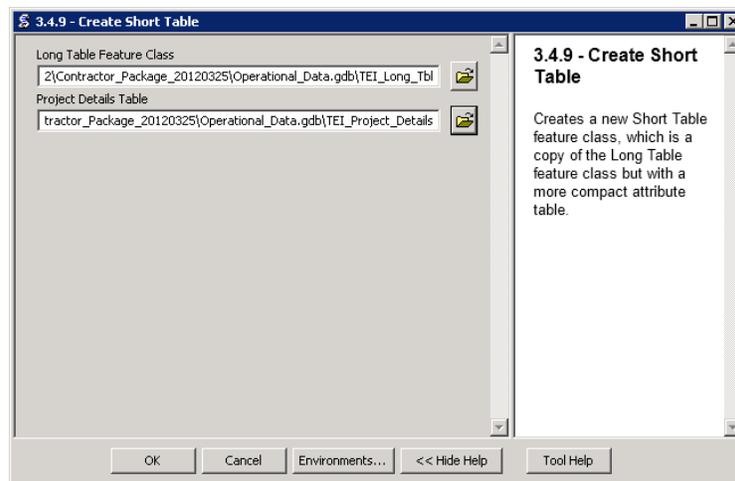
Purpose: To create the detailed polygons feature class with geometry identical to the Long Table feature class, but with a summarized attribute table with individual codes concatenated together into standardized map label strings.

Description: Using logic embedded in the tool's script code (ie. the rules are not defined in any external metadata table), this tool concatenates the Long Table's mostly one- and two-character detailed attributes into longer standard map label strings, which ecologists are most accustomed to reading.

The overall workflow does not require the running of this tool by Contractors; however, having the attributes in the form of map labels may help an ecologist to identify errors or mistakes in the attributes.

Parameters: Long Table feature class AND Project Details table, separated by a space. The *long_tbl_SummaryError* table need not be specified; if it exists (having been created at step 3.4.5), then the script will find and incorporate it.

Usage example:



Output: TEI_Short_Tbl feature class in the Operational_Data.gdb file geodatabase.

3.5 ArcGIS Desktop Tools

The following is a list of tools available in a standard installation of ArcGIS Desktop 10. This is not a comprehensive list of ArcGIS tools, but a short list of tools that are referred to in the TEIS Environment Workflow diagrams.

Tool Name (Toolbox / Toolset)	Parameters	Description
Import from E00 (Conversion / To Coverage)	E00 file, output folder, output coverage name	Converts an E00 interchange file to an Arc/Info coverage (which can subsequently be converted to an FGDB feature class).
Feature Class to Feature Class (Conversion / To Geodatabase)	Input GIS data, output location (file geodatabase), output feature class name	Converts from another GIS data format (such as a coverage) to an FGDB feature class.
Table to Table (Conversion / To Geodatabase)	Input table data, output location (file geodatabase), output table name	Converts from another table format (such as CSV or DBF) to a table in a file geodatabase.
Copy (Data Management / General)	Input data, output data	Copies input data (feature class, table, or other type) and pastes it in the same or another location.
Join Field (Data Management Tools / Joins)	Input table, input table join field, join table, join table join field, join table fields to be joined to input table	Copies fields of a table to another table based on a common attribute field.
Append (Data Management / General)	Input dataset(s), target dataset	Copies records (features, or rows of a table) of an input dataset into an existing target dataset.

Appendix A: Document Version History

Version 7.00 (2014-03-31):

- Added section 3.2.19 *CreateDistributionPackages.py* to describe this scripted process.

Version 6.01 (2013-05-23):

- Updated section 3.2.7 *CreateProjectBoundaries.py* to specify the argument and a command-line example.
- Updated section 3.2.8 *CreateShortTable.py* to specify the argument and a command-line example.
- Updated section 3.2.9 *ExportToBCGWStaging.py* to specify the argument and a command-line example.
- Updated section 3.2.10 *ExportToTEIShare.py* to specify the argument and a command-line example.
- Updated section 3.2.11 *CreateProjectBoundariesForWebsite.py* to specify the argument and a command-line example.

Version 6.00 (2013-03-31):

- Updated section 2.3.2.2 *TEIS_ID* to change description of TEIS_ID field from not persistent to persistent.

- Updated section 2.4.1 *Soils data* to describe the new SOILS_Master folder.
- Updated section 2.5.1 *Operational Data* to remove reference to soils feature classes.
- Updated section 2.6.1 *Operational_Data* remove descriptions of TEIS_Master_Project_Details and TEIS_Project_Tracker, which have been replaced by the BAPID Tracker database.
- Removed section 2.6.1.1 *Re-creating TEIS Relationship Classes*, as the entities described here have been replaced by the BAPID Tracker database.
- Updated section 2.6.2.1 *Maintaining the TEIS_Metadata_Files table* to add description of the new UpdateTEISMetadataFilesTable.py script.
- Updated section 2.6.2.3 *Maintaining the TEIS_Metadata_Report_Tables table* to replace SS_Doc_Idx field with Subm_Type field, as that field name has changed in the TEIS_Metadata_Tables table.
- Updated section 3.2.12 *CreatePDFsForWebsite.py* to specify that it can only be run in ArcGIS 10.1 or later.
- Removed section 3.2.13 *Create map images for website front pages*, as this manual function is now performed by the CreatePDFsForWebsite.py script (see section 3.2.12). Renumbered the subsequent subsections of 3.2.
- Removed section 3.2.19 *PopulateProjectTracker.py* as the Project Tracker table and this script have been replaced by the BAPID Tracker database application.
- Added section 3.2.18 *UpdateTEISMetadataFilesTable.py*.

Version 5.32 (2012-10-26):

- Added section 3.2.3 *RepairGeometry.py* with a description of this new script which repairs the geometry of polygons that are valid in FGDB format but not in ArcSDE geodatabases. Renumbered subsequent sections.
- Updated section 3.2.4 *CheckUniqueTEIS_ID.py* with a description of this new script which replaces SetTEIS_ID.py for maintaining unique values in the TEIS_ID field.
- Updated section 3.2.6 *StandardizeNoData.py* to note the removal of the RepairGeometry function from this script. It now takes place in the new RepairGeometry.py script.
- Added section 3.3.5 *RepairGeometry.py* with a description of this new script which repairs the geometry of polygons that are valid in FGDB format but not in ArcSDE geodatabases. Renumbered subsequent sections.
- Updated section 3.3.7 *StandardizeNoData.py* to note the removal of the RepairGeometry function from this script. It now takes place in the new RepairGeometry.py script.
- Updated section 3.3.6 *CheckUniqueTEIS_ID.py* with a description of this new script which replaces SetTEIS_ID.py for maintaining unique values in the TEIS_ID field.
- Added section 3.4.3 *Repair Geometry tool* with a description of this new tool. Renumbered subsequent sections and tools.
- Updated section 3.4.4 *Check Unique TEIS_ID tool* with a description of this new tool which replaces the Set TEIS_ID tool for maintaining unique values in the TEIS_ID field.
- Updated section 3.4.5 *Standardize NoData tool* to note the removal of the RepairGeometry function from this tool. It now takes place in the new Repair Geometry tool.

Version 5.31 (2012-10-24):

- Updated section 3.2.11 *CreatePDFsForWebsite.py* with a description of some manual edits necessary in the MXD files before running the script in ArcGIS 10.0.

Version 5.3 (2012-09-28):

- Updated section 2.2 *Structure Overview* with descriptions of new file BCGW staging file geodatabases and toolbox.
- Updated section 2.6.2.1 *Maintaining the TEIS_Metadata_Files table* to describe fields that were recently added to the TEIS_Metadata_Files table.
- Replaced section 3.2.5 *SliceProcessing.py* with section 3.2.5 *ValidateData.py*, section 3.2.6 *CreateProjectBoundaries.py*, and section 3.2.7 *CreateShortTable.py*. Renumbered subsequent sections.
- Renamed section 3.2.8 *ExportToLRDWStaging.py* to 3.2.8 *ExportToBCGWStaging.py* and updated the description.
- Added section 3.2.9 *ExportToTEIShare.py*.
- Updated section 3.2.10 *CreateProjectBoundariesForWebsite.py* to reflect changes made in the script.
- Updated section 3.2.11 *CreatePDFsForWebsite.py* to reflect changes made in the script.
- Updated section 3.2.11.1 *Maintaining the MXD files* to reflect changes made to the MXD files.
- Updated section 3.2.12 *Create map images for website front pages* to reflect changes made to the MXD file and to the broad project types.
- Replaced sections 3.3.7 *ValidateDomains.py*, 3.3.8 *ValidateGeometry.py*, 3.3.9 *ValidateRows.py*, 3.3.10 *CreateErrorSummary.py* and 3.3.11 *ValidateMandatoryFields.py* with section 3.3.7 *ValidateData.py*. Renumbered subsequent sections.
- Updated section 3.4 *Scripts and Tools for Preparing New Mapping Project Data for Submission* with description of new structure of the TEI Toolbox; renumbered all subsections and updated their descriptions.

Version 5.24 (2012-02-06):

- Updated section 3.2.6 *ExportToLRDWStaging.py* to describe the RepairGeometry function, and to note that the step of testing for unique TEIS_ID and/or BAPID values now takes place elsewhere.
- Updated section 3.2.11 *ExportToContractorPackage.py* to describe a new function that strips the Contractors' version of the TEIS_Metadata_Items table of unnecessary fields.
- Updated section 3.3.8 *ValidateGeometry.py* to describe new function (checking that polygons do not fall outside of BC's coordinate range)

Version 5.23 (2011-11-02):

- Added new section *Appendix A: Document Version History*

Version 5.22 (2011-10-29):

- Added new section 3.2.15 *PopulateProjectTracker.py*

Version 5.21 (2011-09-29):

- Added new content (descriptions of TEIS_Project_Tracker and TEIS_Related_BAPIDs tables) to section 2.6.1 *Operational Data*
- Added new section 2.6.1.1 *Re-creating TEIS Relationship Classes*

Version 5.2 (2011-09-12):

- Added new content to section 2.6.2.3 *Maintaining the TEIS_Metadata_Report_Tables table*
- Added new content to section 3.2.7.1 *Maintaining CreateBoundariesForWebsite.py and TEI_*_PDF_*.mxd*
- Added new content (just the final paragraph) to section 3.2.9 *Create map images for website front pages*

Version 5.1 (2011-06-16): Document completely revised and restructured.