Using the CLI Toolbox

Introduction, Guidelines, and Suggested Workflows

Cultural Resource GIS Facility
Olmsted Center for Landscape Preservation
National Center for Preservation Technology and Training
November 2014
Adam Cox, GIS Specialist, NPS

PREFACE

The CLI Toolbox is tailored to the specific needs of NPS Cultural Landscapes Program staff working with GIS data. The tools are designed to facilitate the creation of GIS data for new cultural landscapes, as well as the analysis and improvement of existing landscape GIS data that resides in the CR Enterprise spatial database, which is served throughout the NPS network. Because direct edits to the NPS CR Enterprise database can only be made by the designated regional data editors, this toolbox is intended as a way for Cultural Landscapes Program staff to create or improve data that will be easy for the data editors to improve the CR Enterprise.

Beyond the Cultural Landscapes Program, the CLI Toolbox also aims to facilitate broader adoption of the NPS CR Spatial Data Transfer Standards within the NPS.

TECHNICAL NOTE: Throughout this document, any mention of a "geodatabase" refers to an ESRI file geodatabase (.gdb), not an MS Access-based personal geodatabase (*.mdb).*

ACKNOWLEDGEMENTS

The creation of the CLI Toolbox was greatly aided by the following NPS staff:
Deidre McCarthy and Matt Stutts (CR-GIS), Tim Layton (OCLP), Allen McCoy, Susan Dolan, and Julia Yu (PWRO-Seattle), Vida Germano (PWRO-San Francisco), Angie Southwould (AKRO), Cassidy Van Beek (MWRO), Tom Gwaltney, Janice Vogel, and Mike Commisso (NCRO), and Andras Nagy (NCPTT)

--Contents--

1. Background Information

Project Background -- 1

About the Cultural Landscape Inventory -- 2

Features -- 2

Boundaries -- 3

About the NPS CR Spatial Data Transfer Standards -------------------------- 3

About the CLI Standards Geodatabase -- 4

About the CR Link and CR Catalog Tables ------------------------------------- 5

CLI Toolbox Key Concepts -- 6

Directory Structure -- 6

Scratch Geodatabase -- 6

CLI Lookup Tables --- 7

Organization of the CLI Toolbox -- 8

2. Using the Tools to Create Data

Before You Start… --- 10

Creating Project Folders -- 10

Importing Existing Spatial Data -- 11

Creating Data with Scratch Feature Classes ----------------------------------- 12

Creating Geometry -- 12

Updating Fields --- 14

Migrating Features from Scratch to Standards ------------------------------- 16

Completing the CLI Standards Geodatabase ---------------------------------- 17

Field Values QA/QC -- 18

Creating GUIDs --- 19

Synchronizing the CR Link and Catalog Tables ------------------------ 20

Adding Program IDs to the CR Link Table ------------------------------------ 21

3. Using the Tools to Display and Review Data

Displaying the Contents of a Geodatabase ------------------------------------ 23

Creating Google Earth Files -- 25

Creating Spreadsheet Summaries --- 25

Single Landscape Summary Spreadsheet ---------------------------------- 25
Multiple Landscape Summary Spreadsheet-------------------------------- 27

4. Accessing Data in the CR Enterprise Database

About the CR Enterprise Database--- 28
Summarizing CLI Data in the Enterprise Database ----------------------------- 28
Acquiring Local Copies of Data from the Enterprise Database --------------- 29

Appendices

Appendix A: Glossary of Selected GIS Terminology --------------------------- 31
Appendix B: Full List of CLI Tools (with page references) ------------------- 33
Appendix C: Using CLI Tools Outside of CLI Data ---------------------------- 36
Appendix D: Python Installation Notes -- 38

PART 1: BACKGROUND INFORMATION

Project Background

These tools and guidelines are the result of the CLI Legacy GIS Conversion project that was conducted from October 2012 to October 2014. The project was funded by the Cultural Resource GIS Facility in WASO, and carried out by GIS staff Adam Cox and Andras Nagy while based at the National Center for Preservation Technology and Training in Natchitoches, Louisiana.

The objective of the project was to create an NPS CR Spatial Data Transfer Standards compliant geodatabase of all the features in the Cultural Landscape Inventory (CLI) that had been listed in completed CLI units (herein generally referred to as "landscapes") by November, 2012. The ultimate goal was the inclusion of spatial data for CLI features in the NPS CR Enterprise database.

From the outset, the following rules were used to define the scope of work for the project:

- The Alaska Region CLI features would not be included as the AKRO GIS staff was tackling the same project on their own.
- Landscapes would only be included if their Inventory Status in the CLI had been marked as "Complete" by November 2012.
- Officially, only Contributing features would be included, though if convenient, Non-Contributing or Undetermined features may be included as well. (At first, *all* features were expected to be included, until this clearly became an unrealistic goal.)
- Though there are many features *described* in each landscape, only features that have been listed *individually* in the CLI would be included (for more information on what this means, see "About the Cultural Landscape Inventory" on page 2).

Using these guidelines, the scope of work for the legacy project included approximately 22,000 features in approximately 650 landscapes. While creating data for all of these features and landscapes, many Python scripts and ArcGIS tools were created in order to:

- Accommodate the complex nature of the CLI database
- Create data that conforms to the NPS CR Spatial Data Transfer Standards

The CLI Toolbox is an aggregation of these tools and scripts into a well-ordered and well-documented ArcGIS toolbox, written in Python 2.7 for ArcGIS 10.1/10.2. It was compiled at the end of the legacy conversion project thanks to additional funding from the Olmsted Center for Landscape Preservation.

In creating this toolbox and the accompanying documentation, we hope to establish a protocol for the future creation and use of GIS data within the Cultural Landscapes Program. Though this guidance document can act as a strict procedural guide, it's also important to keep in mind that these tools are designed for maximum flexibility; they function outside of the workflows described here, as well as outside of the CLI altogether.

About the Cultural Landscape Inventory

Very central to understanding how GIS will be used in the CLI, and ultimately incorporated into the NPS Cultural Resource Enterprise database, is the construction of the CLI database itself. At the most basic level, there are two types of objects from the CLI that are included in the GIS, features and boundaries.

--FEATURES--

Each landscape listed in the CLI has a 6-digit identifying number that identifies it as a unit. For example, the Port Oneida Historic District landscape has the number 500003. Within each landscape, features are divided into various categories of Landscape Characteristics, of which there is a potential universe of thirteen. Not all landscapes have significance in each of the potential Landscape Characteristics—for example, the inventory for Port Oneida Historic District only uses nine of the thirteen categories, as the cultural landscape of the Port Oneida Historic District retains integrity through nine Landscape Characteristics. Each Landscape Characteristic has a narrative description of features, as well as a table listing all of the individual features that fit within that category. Only when a feature is listed in one of these tables does it receive an Analysis Evaluation Features ID. This is also a 6-digit identifying number, and it is the most fine-grained level of catalog number for features in the CLI. Also, the features with an Analysis Evaluation Features ID are the only ones that can be reliably retrieved from the CLI database using a query.

What this means for GIS: only features that have been listed in the table for a given Landscape Characteristic have been given an Analysis Evaluation Features ID, and only features with an Analysis Evaluation Features ID can be incorporated into the NPS CR Enterprise database. Therefore, each feature included in this project has a three-tier level of characterization that must be preserved in the GIS. To accommodate this, the following fields are included in all CLI GIS data:

CLI_NUM	The CLI number for the landscape in which the feature is listed
LAND_CHAR	The Landscape Characteristic category in which the feature is listed
CLI_ID	The Analysis Evaluation Features ID given to the feature itself

As an example, the feature "Maple Tree Rows" in the Port Oneida Historic District has the following attributes: CLI_NUM = 500003, LAND_CHAR = Vegetation, CLI_ID = 105248.

--BOUNDARIES--

In the CLI, the boundaries of landscapes are not listed as "features" in the way described above, and therefore, they do not have Analysis Evaluation Features IDs. To deal with this in GIS, their CLI_ID will be set to equal their CLI_NUM, and their LAND_CHAR will be Boundary. Thus, the boundary polygon for Port Oneida Historic District would have the following attributes:
CLI_NUM = 500003, LAND_CHAR = Boundary, CLI_ID = 500003.

One can also think of landscape units and their boundaries as sitting in a three-tiered hierarchy: There is the landscape, the park that contains the landscape, and the region that contains the park. For example, Port Oneida Historic District is positioned thusly: Midwest Region>SLBE >500003. To combine this with the feature example above, the full lineage of the "Maple Tree Rows" could be shown as:
Midwest Region>SLBE>500003>Vegetation>105248.

It is important to note that in some cases, a given landscape is designated in the CLI as a component landscape that fits within a larger parent landscape. Because this is not a relationship that applies to all landscapes, and because parent landscapes and component landscapes both have their own CLI numbers and their own sets of features, this parent/component relationship is not built into the three-tiered hierarchy that is found in the toolbox.

About the NPS CR Spatial Data Transfer Standards

The CLI Legacy Conversion project was based on the use of the NPS CR Spatial Data Transfer Standards, as defined by the v2 documentation released by CR-GIS in January, 2014. These standards are a set of attributes designed to track the creation and lineage of cultural resource spatial data in the NPS. In addition to documenting the transfer standards, CR-GIS also designed a geodatabase schema that is based on them, which is intended as an implementation model for the transfer standards. The January 2014 documentation describes both the transfer standards themselves, as well as the implementation model. This documentation is included with the CLI Toolbox (document named "CulturalResourceStandards_Guidelines_v2.pdf"), and can be found on IRMA as well (https://irma.nps.gov/App/Portal/Topics/Cultural, follow the "Cultural Resource Digital Layer Standards" link).

The implementation model is also the foundation of the schema of the NPS CR Enterprise database that is served through the NPS network. Because the ultimate goal of the CLI Legacy Conversion project was to upload the CLI spatial data to the CR Enterprise database, using the geodatabase implementation model was imperative to making this process easy.

However, there are a few additions that were made to the implementation schema to accommodate the unique nature of the CLI database, which result in a geodatabase template called the CLI Standards. This format is the basis for most operations in the CLI toolbox, and is described in more detail in the following pages.

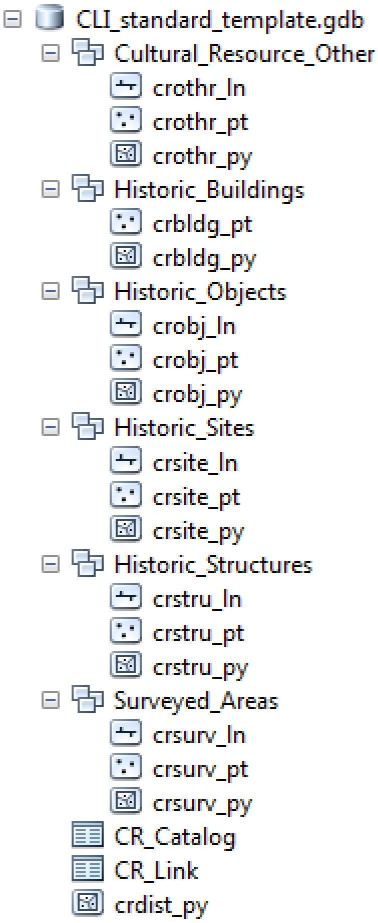
About the CLI Standards Geodatabase

The CLI Standards Geodatabase is based very closely on the v2 Spatial Data Transfer Standards Geodatabase Implementation model that is described in the v2 transfer standards documentation released by CR-GIS in January 2014. The geodatabase comprises a set of feature classes (inside of feature class datasets) that are based on the National Register features types. Each feature class has all of the spatial data transfer standards fields, and the only significant addition to in the CLI Standards model are the CLI_ID, LAND_CHAR, and CLI_NUM fields which are described in the "About the Cultural Landscape Inventory" section (see p. 2).

The feature types are:

Historic Sites (archeological sites…)
Historic Buildings (houses…)
Historic Structures (bridges…)
Historic Objects (statues…)
Other Cultural Resources (vegetation…)
Historic Districts (national register districts…)
Surveyed Areas (surveyed areas…)

The CLI Standards geodatabase contains two tables called "CR_Link" and "CR_Catalog." These are non-spatial tables that serve to make connections between the spatial data and all of the various NPS CR programs that have documented them. For example, a house may be listed as a feature in the CLI, an entry in the National Register, as well as in the List of Classified Structures, and it will most certainly have a listing in FMSS, the NPS facilities management system. In such a case, the row in the CR Link table that holds the CR_ID for that house will list its CLI Feature ID, National Register Reference Number, List of Classified Structures Number, and FMSS Asset Code. This allows for a linkage between various cultural resource databases. See the next section for a more detailed description of how these two tables interact.



```
⊟ CLI_standard_template.gdb
  ⊟ Cultural_Resource_Other
      crothr_ln
      crothr_pt
      crothr_py
  ⊟ Historic_Buildings
      crbldg_pt
      crbldg_py
  ⊟ Historic_Objects
      crobj_ln
      crobj_pt
      crobj_py
  ⊟ Historic_Sites
      crsite_ln
      crsite_pt
      crsite_py
  ⊟ Historic_Structures
      crstru_ln
      crstru_pt
      crstru_py
  ⊟ Surveyed_Areas
      crsurv_ln
      crsurv_pt
      crsurv_py
    CR_Catalog
    CR_Link
    crdist_py
```

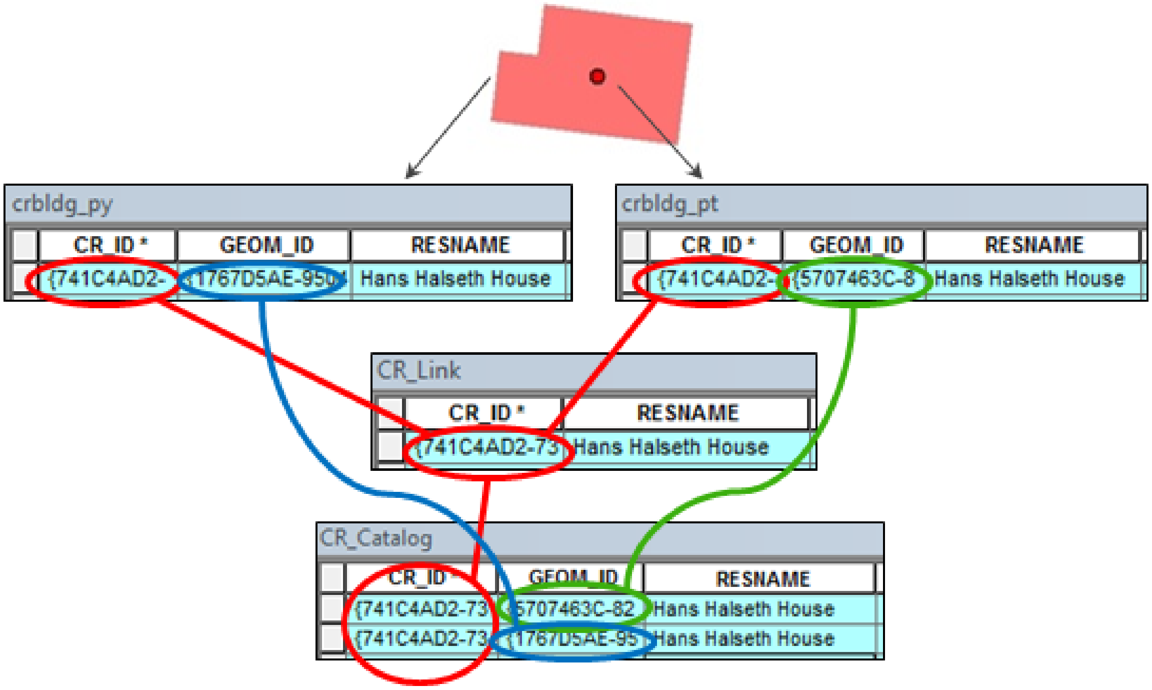
Figure 1. Contents of the CLI Standards geodatabase, not shown are the relationship classes between each feature class and the CR_Link table.

When landscape GIS data is in this geodatabase schema, many useful operations can be carried out on it. For example, one may display all of the features in a landscape based on their contributing status in the CLI, or one may export a Google Earth file that has all of the landscape features grouped and colored by their respective landscape characteristic. Most importantly, this schema is very close to that of the NPS CR Enterprise database, so this will be an easy format for data editors to deal with when uploading new landscape data to the Enterprise (see p. 28).

About the CR Link and CR Catalog Tables

Near the end of the CLI Legacy Conversion project, the addition of the CR Catalog table was made to the CLI Standards schema. The CR Catalog concept was developed by the Alaska Region, and is implemented in the NPS CR Enterprise database. This reconfiguration increases the ease with which a user can update the program IDs that are stored in the CR Link table (NRIS reference numbers, LCS numbers, FMSS numbers, etc.).

The way that the CR Link and CR Catalog tables work is very straight-forward, and is all based on the CR_ID and GEOM_ID that is stored for each spatial feature. You may want to review the document titled "Graphical Explanation of GUIDs" (in this documentation folder) before continuing.
Figure 2 shows how the CR_ID and GEOM_IDs for two spatial features that represent a single cultural



crbldg_py

CR_ID *	GEOM_ID	RESNAME
{741C4AD2-	{1767D5AE-950	Hans Halseth House

crbldg_pt

CR_ID *	GEOM_ID	RESNAME
{741C4AD2-	{5707463C-8	Hans Halseth House

CR_Link

CR_ID *	RESNAME
{741C4AD2-73	Hans Halseth House

CR_Catalog

CR_ID	GEOM_ID	RESNAME
{741C4AD2-73	{5707463C-82	Hans Halseth House
{741C4AD2-73	{1767D5AE-95	Hans Halseth House

Figure 2. Illustration showing how Cultural Resource GUIDs (CR_IDs) and Geometry GUIDs (GEOM_IDs) for each feature are reflected in the CR Link and CR Catalog tables.

feature (a house) are carried through from the spatial features to the CR Link table to the CR Catalog. When the CR Link and CR Catalog work in conjunction like this, there is one line per CR_ID in the CR Link table, and one line per GEOM_ID in the CR Catalog. That means that if a new program ID needs to be assigned to this house (say, an LCS ID), it need only be added to the single corresponding line in the CR Link table.

Put another way: the CR Link table lists all of the CR_IDs and their associated program IDs, while the CR Catalog lists all of the GEOM_IDs and their associated CR_IDs.

For more information on the way the CR Link tables and CR Catalog tables are constructed, see pp. 54, and 73-74 of the Cultural Resource Spatial Data Transfer Standards v2 documentation.

CLI Toolbox Key Concepts

The following are elaborations on a few important concepts that are integral to the design and implementation of the tools in the CLI Toolbox.

--DIRECTORY STRUCTURE--

Based on the way that landscapes are inventoried in the CLI program, a very simple directory structure is recommended when creating and maintaining landscape GIS data. Below, A) is a specific example, and B) is the generic form:

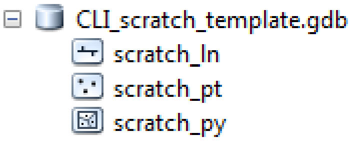
A) C:\CLI_GIS\Midwest Region\SLBE\500003
B) <drive letter>:\CLI_GIS\<region name>\<park alpha>\<CLI number>

One advantage to using this structure is that it keeps all data related to a single landscape in the same folder. Another is that it is entirely scalable. If you begin with one landscape in one region, and a year later begin work on another landscape in a different park or region, you will not have to move files or rename any folders. Just use the ***Create New Project Folder*** tool for every landscape, and a network of landscape folders will begin to form. Keeping a set directory structure and standardized file names makes maintenance and tracking a much simpler task. However, with the understanding that various offices will have their own data storage protocols, the tools in this toolbox are designed for flexible applications, and do not require that your landscape directories are structured this way. They only encourage it.

--SCRATCH GEODATABASE--

The scratch geodatabase is designed to aid in the importation and creation of standards compliant spatial data. The intended use for the scratch geodatabase is to have one in each landscape directory, named with the landscape number. (This is automatically created with the ***Create New Project Folder*** tool.) However, a scratch geodatabase could be placed anywhere, and data imports and exports will work just the same.

By default, the scratch geodatabase contains a blank point, line, and polygon feature class, each of which already has all of the standards fields with the correct domains assigned. The scratch geodatabase serves two purposes:



CLI_scratch_template.gdb
scratch_ln
scratch_pt
scratch_py

Figure 3. Contents of new scratch geodatabase.

A. As a place to which existing datasets can be imported and modified before sorting their features into the CLI Standards geodatabase.
B. As a place to draft new features in the basic point, line, and polygon scratch feature classes before they are sorted to the various feature classes in the CLI Standards geodatabase.

For more information on the scratch geodatabase and its recommended uses, see the guidelines on creating new and importing existing data. Having a separate scratch geodatabase for each landscape was immediately necessary during the legacy conversion project, as work was always in an "in-progress" state on many, many landscapes.

--CLI LOOKUP TABLES--

Behind many of the functions used in these tools are two master tables, which are located inside of this tool package (CLI Toolbox\scripts\clitools\bin\CLI_InfoTables.gdb\<tables>). These tables serve as reference lists for 1) all of the features in all of the completed cultural landscapes and 2) all of the completed cultural landscapes units (as of this writing, these tables have approximately 33,000 and 750 rows, respectively). From these tables all sorts of information can be gathered: the list of expected features for a landscape, the list of completed landscapes in a park, the landscape characteristic and contributing status of any given feature, the park in which a landscape is located, etc.

These tables can only contain a snapshot of the CLI database (which continues to grow with the landscape program), so strategies have been devised to make their maintenance as easy as possible. A version of the tables, in the form of one, consolidated table that holds all information, will be served through the NPS CR Enterprise database. This table is called CLI_Feature_Table, and the ***Update Local CLI Tables*** tool, when run from the CR Enterprise Access map document, will pull all information from this table and update the user's local versions of the tables. The majority of users will only ever need to perform this operation periodically to keep their tables up-to-date.

In order to keep the CLI_Feature_Table on the CR Enterprise as up-to-date as possible, the document "Updating the CLI Feature Table" explains how to query the CLI database and prepare a new

CLI_Feature_Table that a designated data editor can use to update the master table in the Enterprise. This is an operation that only one person should do every few months, but it will need to be coordinated with the data editor. For the most industrious users, an alternative to using the ***Update Local CLI Tables*** tool would be to follow the "Updating the CLI Feature Table" guide, run the ***Prepare CLI Table for Data Editor*** tool as directed, and also check the "Update Local Tables?" box. This will transfer your newly prepared queries from the CLI into your local toolbox tables, bypassing the Enterprise table altogether. This would be necessary if one has added a new completed landscape to the CLI, and would like to have it (and its features) show up in the local tables that the toolbox uses.

Ultimately, the regular user of this toolbox need not touch these tables at all. However, it is important to understand that all landscape feature counts, resource names, CLI numbers, contributing statuses, landscape characteristic values, and park information come from these tables, which in turn come as straight exports from the CLI. Thus, if you find that a feature seems to be in the wrong landscape characteristic in the GIS, or it has an incorrect resource name, it's because of the way it is entered in the CLI.

--ORGANIZATION OF THE CLI TOOLBOX--

The tools in the CLI Toolbox are split into 5 toolsets. Each set of tools is geared toward a certain type of usage. However, many workflows may include tools from various toolsets. The following is a quick description of each toolset:

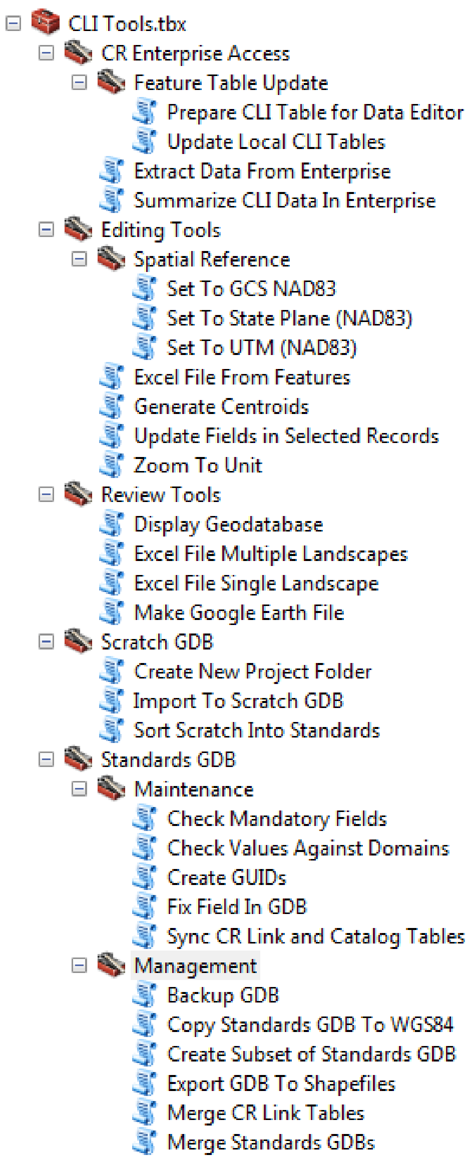
- *CR Enterprise Access*
  These are tools designed to interact with the NPS CR Enterprise Database. This is a database that acts as a repository for all NPS cultural resource spatial data. The Feature Table Update toolset has two tools to assist with the maintenance of the CLI Lookup tables (see p. 7). These tools must be run from ArcMap, using the CR Enterprise Access map document that is included in this installation.
- *Editing Tools*
  These tools are designed to assist with various data creation and editing tasks that will be common throughout the spatial data creation process. They must be run from ArcMap, not ArcCatalog.
- *Review Tools*
  This toolset contains a few tools that were created to assist in summarizing and reviewing the GIS data. Tools will display a geodatabase using preset layer and symbology schemes, create spreadsheet summaries of the progress of landscapes, or make Google Earth format versions of landscape data. These tools can be run from either ArcMap or ArcCatalog, besides Display Geodatabase, which must be used in ArcMap.

- *Scratch GDB*

  These tools provide a set of operations that are geared toward the use of the intermediate scratch geodatabase, which is designed to help with the conversion to and creation of standards-compliant spatial data. These tools can be run from either ArcMap or ArcCatalog.

- *Standards GDB*

  These tools assist in the final steps of creating and maintaining a CLI Standards geodatabase, as well as operations that are used to manage these geodatabases as a whole. These tools can be run from either ArcMap or ArcCatalog.



- CLI Tools.tbx
  - CR Enterprise Access
    - Feature Table Update
      - Prepare CLI Table for Data Editor
      - Update Local CLI Tables
    - Extract Data From Enterprise
    - Summarize CLI Data In Enterprise
  - Editing Tools
    - Spatial Reference
      - Set To GCS NAD83
      - Set To State Plane (NAD83)
      - Set To UTM (NAD83)
    - Excel File From Features
    - Generate Centroids
    - Update Fields in Selected Records
    - Zoom To Unit
  - Review Tools
    - Display Geodatabase
    - Excel File Multiple Landscapes
    - Excel File Single Landscape
    - Make Google Earth File
  - Scratch GDB
    - Create New Project Folder
    - Import To Scratch GDB
    - Sort Scratch Into Standards
  - Standards GDB
    - Maintenance
      - Check Mandatory Fields
      - Check Values Against Domains
      - Create GUIDs
      - Fix Field In GDB
      - Sync CR Link and Catalog Tables
    - Management
      - Backup GDB
      - Copy Standards GDB To WGS84
      - Create Subset of Standards GDB
      - Export GDB To Shapefiles
      - Merge CR Link Tables
      - Merge Standards GDBs

Figure 4. Full contents of CLI Toolbox. For an explanation of each tool, see Appendix B or each tool's help documentation.

PART 2: USING THE TOOLS TO CREATE DATA

Before You Start…

To install and use the toolbox, simply move this "CLI Toolbox" folder and its contents to an easy-to-locate directory on your computer (suggestion: C:\CLI_GIS\CLI Toolbox). DO NOT rename, remove, or reorganize any of the files within this folder, as all of the operations are referential. Next, in ArcMap or ArcCatalog, open the ArcToolbox window, right-click on the ArcToolbox, choose Add Toolbox… and navigate to this directory to locate the CLI Toolbox. You may also run tools by finding the CLI Toolbox in the ArcCatalog window in ArcMap.

Here are a few important notes regarding the general use of tools in ArcGIS.

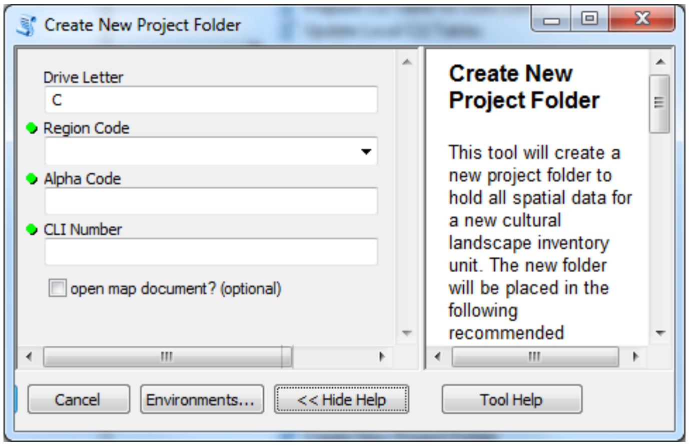
- You can run tools from either ArcMap or ArcCatalog, though some of the CLI tools that deal specifically with editing or displaying features must be run from ArcMap, and the tools that access and summarize the CR Enterprise database must be run from the CR Enterprise Access map document.
- By right-clicking on a tool in the toolbox, you can run it in batch mode. This could be useful for importing multiple datasets to a scratch geodatabase (p. 11), making multiple spreadsheet summaries (p. 25-27), or making Google Earth versions of more than one geodatabase (p. 25).
- When viewing a tool dialog box, always use the Show Help >> button to get the extra help that is written into the interface. Information about a specific parameter will be shown when the cursor is in that parameter. A lot of usage questions can be answered by reading this information. You can also right-click on the tool to view its Item Description or use the Tool Help button to see all of the help documentation at once.
- The Results window (found under the Geoprocessing menu) logs all the information generated from each attempt to run a tool. If you'd like to rerun an operation with all or most of the same parameter values as were previously entered, just double click the instance log of the tool in the Results window, and the dialog box will reopen with all the parameters filled as they were at the time that the tool was run. This is *extremely* useful. Further note: As the Results window fills up with tool logs, the map document will take longer and longer to save. Remove these logs to decrease the save time of a map document.
- The ESRI help menu is extremely useful, and will answer a lot of general questions. It can be opened from the main Help menu in ArcMap or ArcCatalog.

Creating Project Folders

The easiest way to begin creating data for a new landscape is to use the ***Create New Project Folder*** tool.

1. Open the ***Create New Project Folder*** tool from ArcMap or ArcCatalog

2. Enter the Region Code, Alpha Code and CLI Number for the landscape (see Figure 5).
3. Be sure to enter the Alpha Code and CLI Number correctly, as they will not be checked against any existing lists of parks or landscapes.
4. Click OK to run the process.



Create New Project Folder

Drive Letter: C

Region Code:

Alpha Code:

CLI Number:

☐ open map document? (optional)

Create New Project Folder

This tool will create a new project folder to hold all spatial data for a new cultural landscape inventory unit. The new folder will be placed in the following recommended

Figure 5. Dialog box for *Create New Project Folder*

This will create a new folder named by the landscape number and embedded in the preferred directory structure. This folder will contain a new scratch geodatabase and map document, both named with the CLI number.

The map document and scratch geodatabase templates that are used in this process are located in the bin folder (CLI Toolbox\scripts\clitools\bin). If desired, modifications can be made to the map document template and they will be present in all future project folders. For example, one may want to have a layer of counties or an imagery basemap present in each new map document. Just locate the "mxd_template.mxd" in the bin folder, add the desired layers to it, and save it. At present, the NPS boundary layer is in the map document template, as well as each of the scratch feature classes from the accompanying scratch gdb. (Warning: Do not remove the NPS boundary layer from the template map document. This will cause an error when the tool tries to zoom to the Alpha Code that has been entered.)

Importing Existing Spatial Data

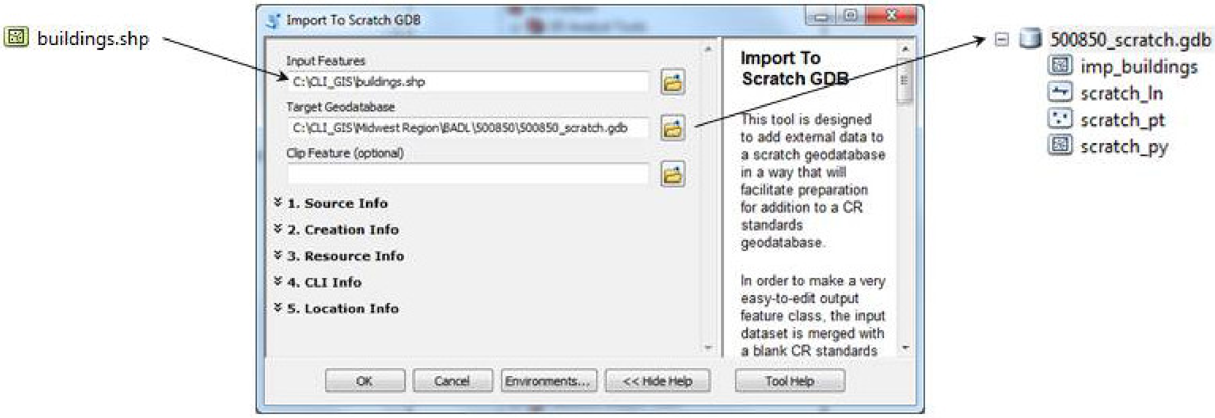
The ***Import to Scratch GDB*** tool is the best way to prepare an existing dataset for inclusion in a CLI Standards geodatabase. Or, more generally, it's a good way to add and populate all of the standards-compliant fields.

1. Open the ***Import to Scratch GDB*** tool from ArcMap or ArcCatalog
2. Locate the dataset (e.g. a feature class or shapefile) to be imported to a scratch geodatabase. In fact, any geodatabase can work as the target; a scratch geodatabase is not required.
3. Enter values for any fields that you'd like to batch-populate during the process.
4. See the tool dialog help for more information on each parameter.
5. Click OK to run the process.

When a feature class or shapefile is imported with this tool, a copy of the dataset is made in the target geodatabase and all of the CLI standards fields are added to it, while all of its original fields and attributes are retained. The original dataset itself is unaltered.

In addition to the CLI standards fields, the "fclass" field is also added. As explained in more detail later (see p. 15), if features are to be taken from this imported feature class to a CLI Standards geodatabase using the ***Sort Scratch to Standards*** tool, they must have a value in the "fclass" field.

See Figure 6 for an illustration of how a dataset is processed through this tool.



Input Features: C:\CLI_GIS\buildings.shp

Target Geodatabase: C:\CLI_GIS\Midwest Region\BADL\500850\500850_scratch.gdb

Clip Feature (optional):

Import To Scratch GDB

This tool is designed to add external data to a scratch geodatabase in a way that will facilitate preparation for addition to a CR standards geodatabase.

In order to make a very easy-to-edit output feature class, the input dataset is merged with a blank CR standards

Figure 6. At left is the original dataset, in the center is the *Import To Scratch GDB* tool dialog, and at right are the resulting contents of the target geodatabase.

Creating Data with Scratch Feature Classes

Creating new data can be thought of as a two-step process: making geometry, and updating fields. The scratch feature classes are designed to make these processes very easy.

--CREATING GEOMETRY--

While creating geometry for features, only two fields need to be filled out each time a feature is created: the CLI_ID and "fclass" fields. All other fields can be batch-populated later. To digitize features from an aerial, add an imagery basemap, begin an edit session in the scratch geodatabase, and draft features in the point, line, or polygon feature class.

Here are a few tools from the editing toolset that are useful for this process:

- In the ***Editing Tools*** > ***Spatial Reference*** toolset, you can quickly set the coordinate system of the data frame. It is recommended to use a projected coordinate system (UTM or State Plane) when digitizing features that have right angles, like buildings. You can then use the Rectangle in the Create Features window, or the ctrl + E constrain perpendicular shortcut (use the ArcMap help for an extensive list of editing keyboard shortcuts).

- The ***Generate Centroids*** tool will take input features from a line or polygon feature class and place centroids for them in the corresponding point feature class. The tool should be run after the "fclass" and CLI_ID have been entered, to eliminate the need to reenter those values in the new centroid features. Certain fields will be filled in during this process, e.g. the BND_TYPE of the new centroid is automatically set to "Derived point".
- If you are editing a CLI Standards geodatabase that contains multiple landscapes, use the ***Zoom To Unit*** tool to move from one landscape to the next. However, the zoom won't work correctly if there are features selected when it is run.

In addition to digitizing (drawing) new data, feature geometry can also be copied and pasted from existing datasets into the scratch feature classes. CAUTION: copying features from a feature class that has an *undefined coordinate system* will lead you down a dark and mirthless road. Be sure to define the correct coordinate system for *any* dataset you are using.

Use these steps to copy and paste features:

1. In ArcMap, add the source and target datasets to the table of contents.
2. Start an Edit session in the scratch geodatabase.
3. Select *only* the features that you wish to copy from the source dataset.
4. Right-click in the data frame (you must be in Data View) and select Copy.
5. Right-click in the data frame again and select Paste.
6. Select the scratch feature class into which you'd like to paste the features.

This can be a very slick process, because it is an Append operation with the schema type set to "NO_TEST". This means that any fields that are common between the original and the target will have their values transferred.

To take this concept further, consider the following example of how it can be used to get geometry from a single imported feature class into many different scratch geodatabases, i.e. many different landscapes.

1. Create a blank geodatabase in the alpha code folder, for example, C:\CLI_GIS\Midwest Region\SLBE\empty.gdb
2. If you have a dataset that will be used in all of the landscapes in SBLE, say trails.shp use ***Import to Scratch GDB*** to import this dataset into the new empty.gdb, instead of importing it to an individual scratch geodatabase. The result will be a feature class called imp_trails.
3. Add imp_trails as a layer in all of the landscape map documents for SLBE, and, using the process described above, copy and paste features from imp_trails into the scratch feature classes for that landscape.

4. Only update the CLI_ID and "fclass" values once the feature has been pasted into the scratch feature class.
5. Now, your scratch feature classes will have a combination of newly digitized features (from you) and imported features (copied from imp_buildings), while the original imp_buildings feature class remains unaltered.

--UPDATING FIELDS--

Once you have geometry for all of your features, and they all have a CLI_ID and an "fclass" value, you can use ***Update Fields in Selected Records*** to batch populate (almost!) all of the remaining fields. Simply select all of the features you wish to update, enter information in the tool dialog for whatever fields you'd like to fill out, and run the tool. There are a number of special methods within the dialog box that allow for quick ways to fill the parameters, all of which are documented within the tool dialog itself. One of these features that deserves explanation here is the "Get CLI Info from Table" checkbox. If this box is checked, all of the CLI-related fields will be updated for each feature based on its CLI_ID. The fields that are updated are:

- CLI_NUM (number for landscape that the feature is in)
- LAND_CHAR (the feature's landscape characteristic category)
- RESNAME (feature name)
- CONTRIBRES (contributing status)
- ALPHA_CODE (alpha code for park location of feature)
- UNIT (name of park)
- REGION_CODE (name of region)
- UNIT_TYPE (type of park)

All of this information is pulled from the CLI Lookup Tables that are described on page 7. Filling these fields with this method ensures that the GIS data is a direct reflection of the CLI.

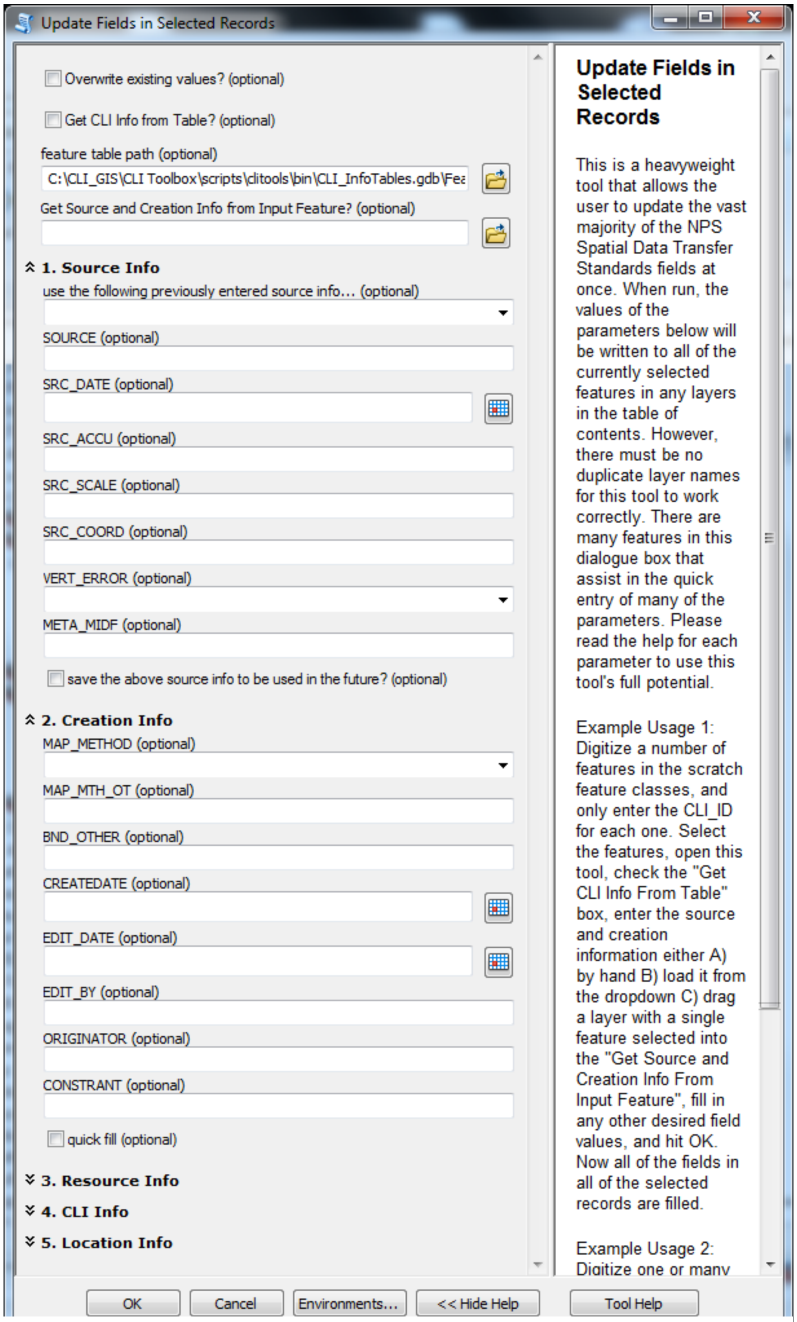
Recommended use for ***Update Fields in Selected Records***:

1. Draft (or copy/paste) all feature geometry
2. Be sure a CLI_ID and "fclass" value has been entered for each feature
3. Select ALL features and run tool with *only* the "Get CLI Info from Table?" box checked
4. Unselect all features and reselect any that were digitized using the same source data (aerial imagery, e.g.) and rerun tool, filling out all source/creation info parameters that apply to each selected feature.
5. Take care that the "Overwrite existing values?" box is used appropriately.

Tip: Start an Edit session for the feature classes whose values you are updating, and if a mistake is made, use the Undo function from the main Edit menu to reset the field values (you will not be able to undo operations if there is no current edit session).

Further notes on the ***Update Fields in Selected Records*** tool:

- ***There cannot be layers present in the table of contents that have the same name.*** This is a shortcoming, but simply renaming the offending layers in the table of contents will fix it.
- The tool can be used on any feature class in any geodatabase. If any fields from the input parameters don't exist in the feature class, they will be skipped without issue.
- To get the most out of this tool and the advanced functions available in the tool dialog, read the help for each parameter. There are some very significant shortcuts written into the tool dialog.



Update Fields in Selected Records

□ Overwrite existing values? (optional)

□ Get CLI Info from Table? (optional)

feature table path (optional)
C:\CLI_GIS\CLI Toolbox\scripts\clitools\bin\CLI_InfoTables.gdb\Fea

Get Source and Creation Info from Input Feature? (optional)

1. Source Info

use the following previously entered source info... (optional)

SOURCE (optional)

SRC_DATE (optional)

SRC_ACCU (optional)

SRC_SCALE (optional)

SRC_COORD (optional)

VERT_ERROR (optional)

META_MIDF (optional)

□ save the above source info to be used in the future? (optional)

2. Creation Info

MAP_METHOD (optional)

MAP_MTH_OT (optional)

BND_OTHER (optional)

CREATEDATE (optional)

EDIT_DATE (optional)

EDIT_BY (optional)

ORIGINATOR (optional)

CONSTRANT (optional)

□ quick fill (optional)

3. Resource Info

4. CLI Info

5. Location Info

Update Fields in Selected Records

This is a heavyweight tool that allows the user to update the vast majority of the NPS Spatial Data Transfer Standards fields at once. When run, the values of the parameters below will be written to all of the currently selected features in any layers in the table of contents. However, there must be no duplicate layer names for this tool to work correctly. There are many features in this dialogue box that assist in the quick entry of many of the parameters. Please read the help for each parameter to use this tool's full potential.

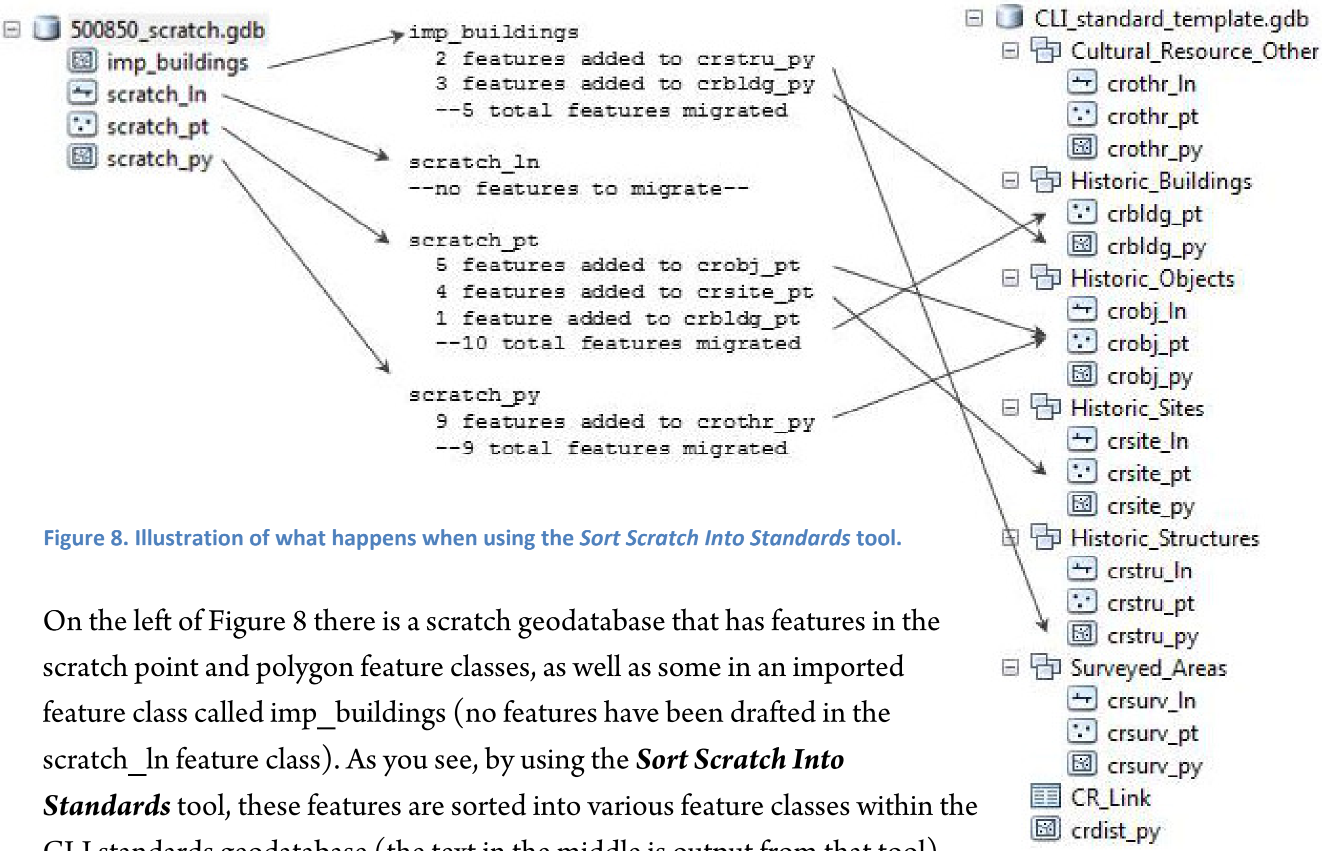
Example Usage 1: Digitize a number of features in the scratch feature classes, and only enter the CLI_ID for each one. Select the features, open this tool, check the "Get CLI Info From Table" box, enter the source and creation information either A) by hand B) load it from the dropdown C) drag a layer with a single feature selected into the "Get Source and Creation Info From Input Feature", fill in any other desired field values, and hit OK. Now all of the fields in all of the selected records are filled.

Example Usage 2: Digitize one or many

Figure 7. Dialog box for Update Fields in Selected Records. Parameters under the Resource Info, CLI Info and Location Info categories are not shown here.

Migrating Features from Scratch to Standards

All of the above steps are geared toward creating a set of features in a scratch geodatabase—either in the scratch feature classes or in imported feature classes—that have AT LEAST the CLI_ID and "fclass" field filled out. Ideally, they will have all fields filled out at this point. The next step is to sort these features into a CLI Standards geodatabase based on the "fclass" value. This process is illustrated below, and it should further demonstrate the intention behind the implementation of the scratch geodatabase.



```
imp_buildings
  2 features added to crstru_py
  3 features added to crbldg_py
  --5 total features migrated

scratch_ln
--no features to migrate--

scratch_pt
  5 features added to crobj_pt
  4 features added to crsite_pt
  1 feature added to crbldg_pt
  --10 total features migrated

scratch_py
  9 features added to crothr_py
  --9 total features migrated
```

Figure 8. Illustration of what happens when using the *Sort Scratch Into Standards* tool.

On the left of Figure 8 there is a scratch geodatabase that has features in the scratch point and polygon feature classes, as well as some in an imported feature class called imp_buildings (no features have been drafted in the scratch_ln feature class). As you see, by using the ***Sort Scratch Into Standards*** tool, these features are sorted into various feature classes within the CLI standards geodatabase (the text in the middle is output from that tool). The operation is all based on each feature's "fclass" value.

If the ***Sort Scratch Into Standards*** tool is run and a feature does *not* have a value in the "fclass" field, that feature will be ignored during the operation. This is essential because, referring to the example above, there may be hundreds of buildings in the imported buildings feature class (imp_buildings), but only five of them were buildings in this landscape, so only five were assigned "fclass" values (three values were "crbldg_py" and two were "crstru_py"). This is a good example of the ease with which the user is able to split features in the same imported feature class into multiple standards feature classes. This may not be necessary if you have a single GPS file for one feature, but it is in cases where a dataset covers a large geographic area or large variety of feature types.

If the data import, creation, and field population steps are all followed as described above, the result of this sorting operation will be a CLI Standards geodatabase with the features from the scratch geodatabase sorted into the correct feature classes. However, there are a few more steps necessary to complete the CLI Standards geodatabase.

Steps for using ***Sort Scratch Into Standards***

1. Open the ***Sort Scratch Into Standards*** tool from ArcMap or ArcCatalog.
2. Locate the scratch geodatabase (e.g. a feature class or shapefile) that contains feature classes with features to sort.
3. Select which feature classes have features that should be sorted to the new CLI Standard geodatabase.
4. If the features should be added to an existing CLI Standard geodatabase (for example, a regional geodatabase), enter this geodatabase as the target.
5. If no target is provided, a new CLI Standard geodatabase will be created which contains only the newly sorted data. (This can later be incorporated into a regional or park geodatabase using the ***Merge Standards GDBs*** tool.)
6. Click OK to run the process.

Completing the CLI Standards Geodatabase

Once features have been sorted into a CLI Standards geodatabase, there are a few final steps before that geodatabase is fully ready for analysis and upload to the CR Enterprise database, or for aggregation with other CLI Standards geodatabases. Follow the numbered steps below, and read past them for a more detailed description of the tools and processes involved.

1. Finalize all geometry:
    a. Use ***Zoom To Unit*** to move between various landscapes
    b. Use ***Generate Centroids*** for any polygons representing buildings or structures
    c. Make any necessary modifications to geometry of existing features
2. Finalize all fields:
    a. Use the ***Check Mandatory Fields*** to make sure there are no Null values in the required fields (disregard GUIDs)
    b. Use ***Update Fields in Selected Records*** to batch update data creation or source information if missing
    c. Use ***Check Values Against Domains*** to make sure all attributes meet the standards domain values
    d. Use the ***Fix Field in GDB*** tool to standardize attributes across all feature classes
3. Use ***Create GUIDs*** to create GUIDs for all features in the geodatabase (do not overwrite GUIDs)
4. Use ***Sync CR Link and Catalog Tables*** to populate and consolidate the CR Link and CR Catalog tables

a. During ***Sync CR Link and Catalog Tables***, an error log will be created with if a CR_ID has conflicting program IDs. If this is the case, rerun the tool once. If the errors persist, follow these steps:
   i. Make sure entire geodatabase is visible in data frame
   ii. Use the Find function (look for in the Tools toolbar) with problematic CR_IDs from log to locate the conflicting spatial data
   iii. Use ctrl + F in the CR Link table to find problematic CR_IDs and associated program IDs
   iv. Make sure that the CR_ID in the CR Link table has the correct program ID.
   v. As necessary, calculate new CR_IDs for specific spatial features (see below)

b. If any changes are made, rerun ***Create GUIDs*** (to ensure correctly transferred CR_IDs for multiple geometry features) and then rerun ***Sync CR Link and Catalog Tables***.

5. Check that the CR Catalog and CR Link are in conchordance
   a. Add the two tables to the table of contents, and open them.
   b. Select all rows in the CR Catalog
   c. Use the relationship class with the CR Link table to select corresponding CR_IDs in the CR Link.
   d. If all rows are not selected in the CR Link table, redo steps 3 and 4 above until they are.

To create new GUIDs using Field Calculator, use the code below (see screenshot also). Make sure the Parser is set to VB Script.

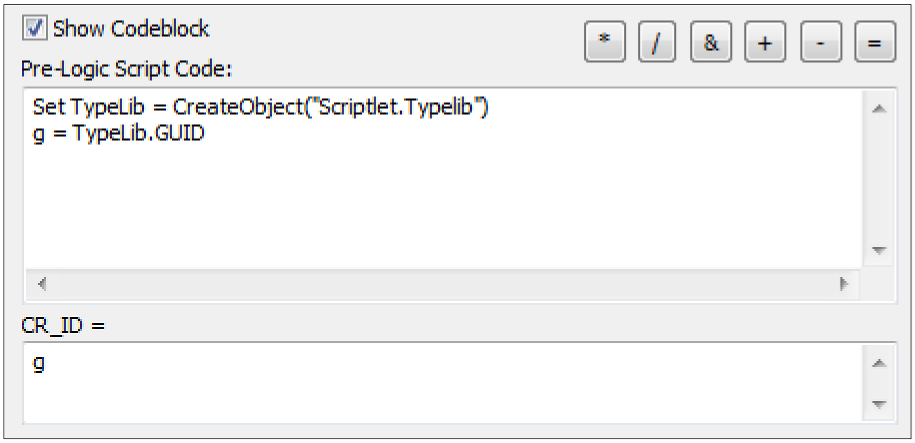
Code Block:

Set TypeLib = CreateObject("Scriptlet.Typelib")

g = TypeLib.GUID

Expression:

g



☑ Show Codeblock

Pre-Logic Script Code:

```
Set TypeLib = CreateObject("Scriptlet.Typelib")
g = TypeLib.GUID
```

CR_ID =

```
g
```

Figure 9. Code for Field Calculator to create a new GUID. Make sure the Parser (at the top of the Field Calculator window) is set to VB Script.

By following all of these steps, the user should produce a cohesive CLI Standard geodatabase. Read on for more background and information on how the tools and processes described above are carried out.

--FIELD VALUES QA/QC--

There are three tools in the Standards GDB > Maintenance toolset that will assist with overall analysis and correction of the attribute values that have been entered (or left out) thus far. Their use is fairly straight-forward, so here is a quick description of each one:

- ***Check Mandatory Fields*** – Analyzes all of the CR Spatial Data Transfer Standards required fields and return a list of all the feature classes in the geodatabase and the fields in each feature class that have empty or Null values. Can be used on any file geodatabase. You can use this list to inspect the attribute table of each feature class to make the necessary corrections. Tip: When you have opened an attribute table and need to know where the Null or empty values are in a given field, just double-click on the field header and the table will be sorted based on that field, with Null or empty values on top.

- ***Check Values Against Domains*** – This tool can be used on any geodatabase; it is not restricted to CLI Standards or Scratch geodatabases. It makes lists of all the domains that are present in the geodatabase and then iterates through every field that has been assigned to a domain. All of the field's values are checked against those in the appropriate domain. For example, in the MAP_METHOD field, there could be the "HDIG" value (for "heads-up digitized"), but this will be flagged because "HDIG" is not in MAP_METHOD domain. This tool will return a list of all feature classes, and any values within those feature classes that don't match the domain for the field they are in. Findings from this tool may be easily corrected with the following tool.
    - Note 1: Not all fields that have domains are required by the Spatial Data Transfer Standards.
    - Note 2: The domain for the LAND_CHAR field does not carry variations on the normal values with different cases. For example, it contains "Buildings and Structures" but not "Buildings And Structures". However, the CLI itself *does* contain both, which means that both will be found in the spatial data. Therefore you may find instances where the LAND_CHAR has values that are not in the domain. All tools in this toolbox are written to disregard the case of LAND_CHAR values, so "Buildings And Structures" is not a problem at all and need not be corrected.
    - Note 3: Similar to the LAND_CHAR domain, any domain that concerns park type or alpha code should not be considered an authoritative list of all possible park types or alpha codes. So there may be values for those fields that are completely acceptable, but not in the domain.

- ***Fix Field in GDB*** – This tool can be used on any geodatabase; it is not restricted to CLI Standards or Scratch geodatabases. When a geodatabase is entered into the first parameter, it will create a list of all the fields that exist anywhere in that geodatabase. When one of these fields is selected, it will generate a list of all the unique values that exist in that field. You then have the opportunity to select any number of the existing values, and enter a new value that should replace all of these old values. For example, you would use this tool to change all occurrences of "HDIG", "digitized", or "Digitised" to the standards-compliant value "Digitized".

--CREATING GUIDS--

Use the ***Create GUIDs*** tool to create the CR_IDs and GEOM_IDs for all features in the CLI Standards geodatabase. You may wish to only create GUIDs for features in a certain landscape or park that exist within the geodatabase. To do this, enter a CLI Number or Alpha Code in the tool dialog. You can create only CR_IDs or only GEOM_IDs with the tool, but it generally makes sense to always create both. It is not advisable to overwrite existing GUIDs, but there may be circumstances in which this is necessary.

When this tool is run, the first half of its operation is to simply calculate new GUIDs. After this is done, and if CR_IDs have been created, it will continue to transfer CR_IDs from one feature to another wherever there is multiple geometry representing a single physical feature (refer to the document "Graphical Explanation of GUIDs" for a clear explanation of how overlapping CR_IDs work). During this process, CR_IDs are transferred for any features that:

1. are within the same feature class dataset (i.e. are both considered Historic Structures)
2. overlap each other
3. have matching CLI_IDs (meaning they represent the same feature)

CR_IDs are transferred from polygons to lines and polygons to points where polygon features are present, and from lines to points where no polygons are present.

Running the ***Create GUIDs*** tool can take up to an hour for an entire region if CR_IDs are created (which causes the transfer process to occur), depending on how many multiple geometry features there are.

--SYNCHRONIZING THE CR LINK AND CR CATALOG TABLES--

Pulling information from the spatial data to create and populate the CR Link and CR Catalog tables is done with the ***Sync CR Link and CR Catalog Tables*** tool. It is important to run this tool last, though it can be run as many times as necessary—to ensure that these tables reflect not only the current spatial records and GUIDs—but also the current qualitative information attached to each feature (LAND_CHAR, ALPHA_CODE, etc.).

When the tool is run, first, the CR Catalog is created/updated and all relevant information from all of the features in the feature classes is written to it. The CR Catalog has one row per GEOM_ID, which means that because every spatial feature has a unique GEOM_ID, there will be a row in the CR Catalog for every single spatial feature in the geodatabase.

Second, the CR Link table is created/updated and all of the unique CR_IDs are written to it. The CR Link table has one row per CR_ID, which means that there is a row for each *physically distinct* feature in the geodatabase. The other characteristic of the CR Link table is that it contains program IDs (NRIS_ID, LCS_ID, ASMIS_ID, CLI_ID, FMSS_ID etc.) for each physical feature that is represented in the geodatabase. Apart from the CLI_ID, this information is *not* stored in the spatial features, which means that unlike the CR Catalog, the CR Link table cannot be recreated from scratch every time the tool is run, because if it were, any program IDs that were already in it would be lost.

To deal with this, the synchronization of the CR Link table is really a 3-step process:

1. All of the features from all of the feature classes are appended to the existing CR Link table. That way, if you have entered numbers in the FMSS_ID field of, say, the Historic Buildings point feature class, all of these FMSS_IDs will be transferred along in to the CR Link table.
2. The tool runs through every row in the now-inflated CR Link table, and begins an in-memory list of all unique CR_IDs. During this iteration, if a CR_ID is encountered that already exists in the rapidly growing list, each of the program IDs that are attached to these identical CR_IDs are combined. This means that if one CR_ID had only been assigned a CLI_ID, and another instance of the same CR_ID only has an FMSS_ID, these values will be combined into a single line of the CR Link table. *IMPORTANT NOTE:* It may happen that two conflicting database IDs have been entered for the same CR ID, say, two CLI_IDs for the same house. The tool will pick one of these IDs, and record the error in a log. You must adjust the table or spatial data manually at this point to correct the error, and then rerun the tool. See the tool messages/documentation and the steps on page 18 for a more thorough explanation of this process.
3. Finally, the CR Link table is rewritten with all of the unique CR_IDs and their respective combined program IDs.

Running the ***Sync CR Link and Catalog Tables*** tool can take up to 15 minutes for an entire region.

Adding Program IDs to the CR Link Table

Chronologically, it doesn't matter when the user chooses to add program IDs to the CR Link table, however there is no CR Link table without CR_IDs, so if brand new data is being created, the best practice would be to follow the steps above to create GUIDs and synchronize the tables before proceeding.

Program IDs can be added to the CR Link table it two basic ways:

1. By directly entering the ID in the correct row of the CR Link table
    a. Select the feature to which you'd like to add an ID.
    b. Open the attribute table for the feature class (right-click in the table of contents, choose Open Attribute Table)
    c. Use the relationship class with the CR Link table to select the corresponding row in the CR Link table for that selected feature. This relationship class is based on the CR_ID.

    d. You should now be looking at a single selected row in the CR Link table. Find the appropriate field, and enter the program ID. (Use ctrl + shift + E to start an Edit session if desired, or use Field Calculator.)

2. By entering the ID into a matching field in the feature class attribute table and then running the ***Sync CR Link and Catalog Tables*** tool to pull the new ID into the CR Link table.
    a. Select the feature to which you'd like to add an ID.
    b. Open the attribute table for the feature class (right-click in the table of contents, choose Open Attribute Table)
    c. Make sure there is a field to hold the new ID that you want to add (LCS_ID and FMSS_ID fields are already present). The name of this field must be identical to the corresponding field in the CR Link table.
    d. Enter the program ID. (Use ctrl + shift + E to start an Edit session if necessary, or use Field Calculator.)
    e. Save edits and close Edit session if necessary, and run the ***Sync CR Link and Catalog Tables*** tool on the geodatabase, and the CR Link table will be updated accordingly.

A third, more powerful way to batch update program IDs would use a table that can be temporarily joined to the CR Link table and have its values transferred to the CR Link table. An example of this use would be to have a table (*.dbf, *.csv, *.xls, geodatabase table, etc.) that lists FMSS_IDs in one column, and corresponding LCS_IDs in another. The user would join this table to the CR Link table by linking the LCS_ID fields, and then use field calculator on the CR Link FMSS_ID field to transfer all of the FMSS_IDs from the other table.

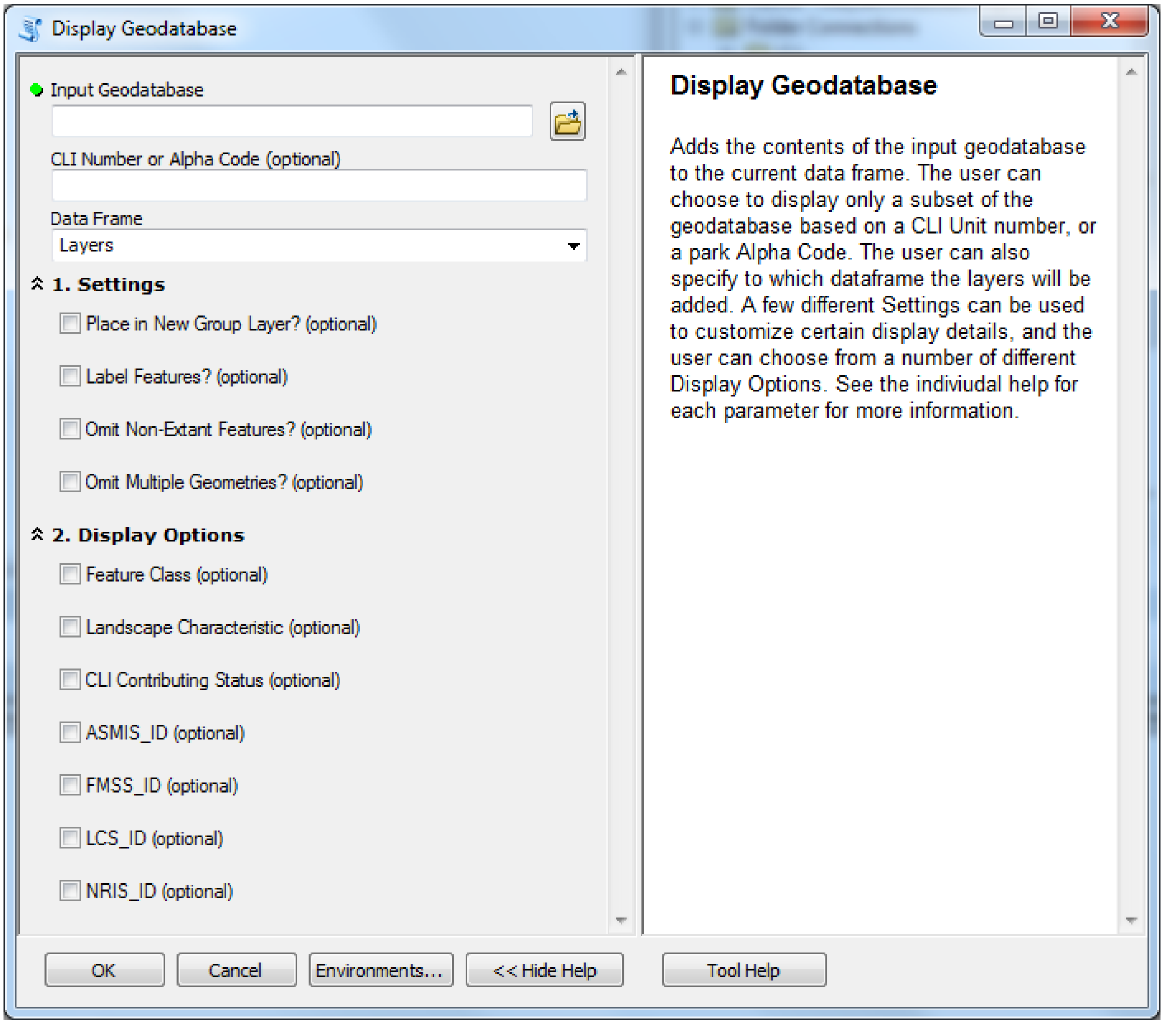
It's important to realize that if the Display Geodatabase tool is used to show features that have been assigned an FMSS_ID, for example, the feature must have the FMSS_ID in the CR Link table for it to register. Program IDs entered directly into the feature class attribute table will be ignored by that tool, unless the ***Sync CR Link and Catalog Tables*** tool has been run to pull the IDs into CR Link table.

PART 3: USING THE TOOLS TO DISPLAY AND REVIEW DATA

Displaying the Contents of a Geodatabase

One of the most useful ways that this data can be used is to display it with different symbology and layering schemes. Use the ***Display Geodatabase*** tool to do this. Either a CLI Standards geodatabase or a Scratch geodatabase can be displayed, though with a scratch geodatabase, the display options are significantly reduced (due to the lack of CR Link table). The ***Display Geodatabase*** tool must be run from ArcMap.

1. Open the ***Display Geodatabase*** tool from ArcMap.
2. Locate the geodatabase you'd like to display
3. Enter a CLI Number or Alpha Code if you'd like to only display a subset of the geodatabase.
4. Select any of the Settings that should be applied to the displayed data.
5. Select one or more Display Options to determine what symbology and layering scheme will be



Display Geodatabase

● Input Geodatabase

CLI Number or Alpha Code (optional)

Data Frame

Layers

1. Settings

□ Place in New Group Layer? (optional)

□ Label Features? (optional)

□ Omit Non-Extant Features? (optional)

□ Omit Multiple Geometries? (optional)

2. Display Options

□ Feature Class (optional)

□ Landscape Characteristic (optional)

□ CLI Contributing Status (optional)

□ ASMIS_ID (optional)

□ FMSS_ID (optional)

□ LCS_ID (optional)

□ NRIS_ID (optional)

Display Geodatabase

Adds the contents of the input geodatabase to the current data frame. The user can choose to display only a subset of the geodatabase based on a CLI Unit number, or a park Alpha Code. The user can also specify to which dataframe the layers will be added. A few different Settings can be used to customize certain display details, and the user can choose from a number of different Display Options. See the indiviudal help for each parameter for more information.

Figure 10. Dialog box for Display Geodatabase.

used to display the contents of the geodatabase.

6. Click OK to run the process.

Once an input geodatabase has been entered, the user may apply any of the settings desired, and may choose any number of Display Options. If more than one Display Options is chosen, they will each be placed in a separate group layer. The settings are fairly straight-forward, but the various display options deserve some explanation.

- **Feature Class:** This is the most basic display option: the tool will add a layer for each feature class in the geodatabase and apply a standardized color scheme. If the user plans to edit the geometry or add new features to the geodatabase, this option should be chosen alone. The reason for this is that all the other display options operate by adding multiple layers of the same feature class using different definition queries. This is awful for editing because 1. Creating new features in layers with existing definition queries is very cumbersome, and 2. The Update Records in Selected Features tool will not work if multiple layers in the table of contents have the same name.
- **Landscape Characteristic:** The features from the geodatabase are divided by Landscape Characteristic (using the LAND_CHAR field), and are added to separate group layers. This makes review of features very easy as the groups for each landscape characteristic can be turned on or off.
- **CLI Contributing Status:** New group layers are added to the table of contents for the following CLI contributing status categories:
  - Contributing
  - Non-Contributing
  - Undetermined
  - Unknown

  The features are then sorted into the appropriate category. Here are a couple of important things to keep in mind:

  1. In the spatial data, the CR Spatial Data Transfer Standards do not quite support these same values. Resources are either flagged as "Yes", "No", or "Unknown". However, this Display Option uses each feature's *CLI* contributing status, as the name indicates.
  2. There are some values in the CLI that are not accommodated here, and will be sorted to the "Unknown" category. For example: "Non-contributing – compatible" which occurs in a few situations. However, "Non-contributing" and "Non contributing" are both placed into the "Non-Contributing" category.
- **Database ID Presence:** All of the final Display Options work the same way: "Yes ID" and "No ID" group layers are added and features are sorted into them based on whether that CR_ID has an associated program ID in the CR Link table. For example, you may display a CLI Standards

geodatabase for an entire park, and use this option to see which features are linked to an FMSS ID in the CR Link table. You could then find all those that don't have an FMSS ID, add the FMSS ID to the CR Link table (see p. 21), and then redisplay the geodatabase to see the change. These Database ID Presence options are not available if a scratch geodatabase is the input, due to a lack of a CR Link table.

Creating Google Earth Files

The ***Create Google Earth File*** tool will convert a set of landscape GIS data into a file that is readable by Google Earth, a KMZ file (which is a compressed KML file). A KMZ file would not be used for editing data or adding features, but it is especially useful for distributing data to interested parties that may not have GIS capabilities.

KMZ files can best be viewed with either Google Earth or ArcGIS Explorer. Both of these applications are free and easy to use, and both come fully equipped with basemap aerial imagery and 3D terrain, which makes them a great way to view landscapes in their larger physiographic contexts. For help on how to use Google Earth, see the document titled "Tips for Using CLI Google Earth Files".

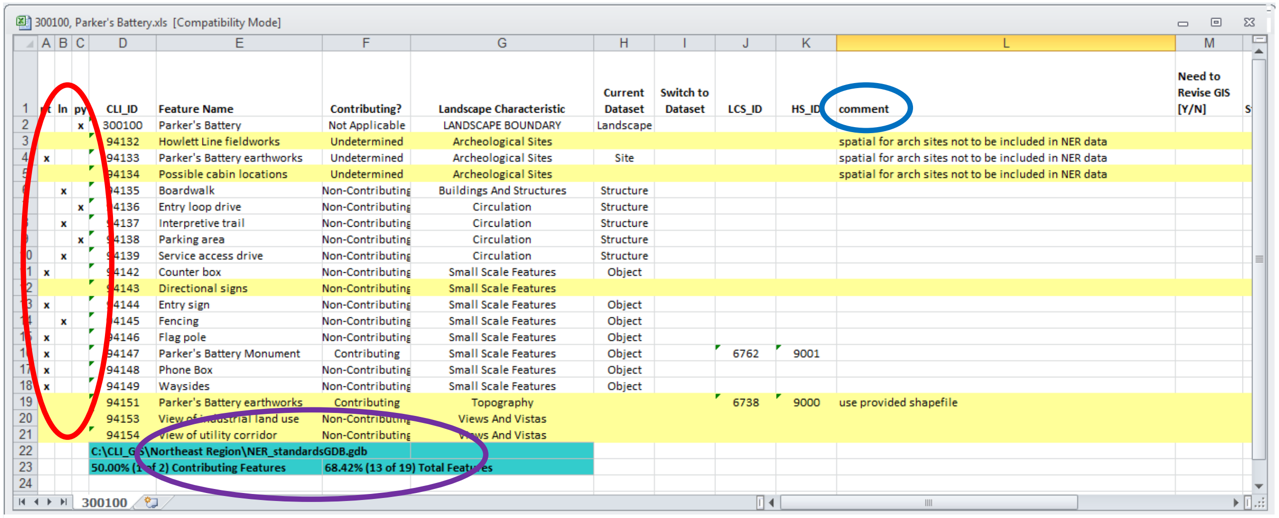
All Display Options and most of the Settings that are available in the ***Display Geodatabase*** tool are also available in the Make Google Earth File tool. The tool works by using the ***Display Geodatabase*** operation on a blank map document, and then converting that map document to a KMZ using ESRI's "Map to KML" tool (in the Conversion toolbox).

Creating Spreadsheet Summaries

There are two tools in the CLI Toolbox that will facilitate the creation of MS Excel spreadsheet summaries, which can be used to review the progress that has been made on a given landscape, park, or region. What should become very clear is that these spreadsheet operations are entirely based on the feature lookup tables that have been described above in the CLI Toolbox Concepts section (p. 7). It's from these tables that lists of all expected features in a landscape, or all expected landscapes in a region, are made.

--SINGLE LANDSCAPE SUMMARY SPREADSHEET--

The ***Excel File Single Landscape*** tool will create a spreadsheet that lists every feature in a given landscape, and highlights any that don't yet have spatial data in the geodatabase that is being analyzed. Example usage: You are working in a scratch geodatabase, and use this tool periodically to check your progress on getting spatial data for all of the features.



300100, Parker's Battery.xls [Compatibility Mode]

pt	ln	py	CLI_ID	Feature Name	Contributing?	Landscape Characteristic	Current Dataset	Switch to Dataset	LCS_ID	HS_ID	comment	Need to Revise GIS [Y/N]	S
		x	300100	Parker's Battery	Not Applicable	LANDSCAPE BOUNDARY	Landscape						
			94132	Howlett Line fieldworks	Undetermined	Archeological Sites					spatial for arch sites not to be included in NER data		
x			94133	Parker's Battery earthworks	Undetermined	Archeological Sites	Site				spatial for arch sites not to be included in NER data		
			94134	Possible cabin locations	Undetermined	Archeological Sites					spatial for arch sites not to be included in NER data		
	x		94135	Boardwalk	Non-Contributing	Buildings And Structures	Structure						
		x	94136	Entry loop drive	Non-Contributing	Circulation	Structure						
	x		94137	Interpretive trail	Non-Contributing	Circulation	Structure						
		x	94138	Parking area	Non-Contributing	Circulation	Structure						
	x		94139	Service access drive	Non-Contributing	Circulation	Structure						
x			94142	Counter box	Non-Contributing	Small Scale Features	Object						
			94143	Directional signs	Non-Contributing	Small Scale Features							
x			94144	Entry sign	Non-Contributing	Small Scale Features	Object						
	x		94145	Fencing	Non-Contributing	Small Scale Features	Object						
x			94146	Flag pole	Non-Contributing	Small Scale Features	Object						
x			94147	Parker's Battery Monument	Contributing	Small Scale Features	Object		6762	9001			
x			94148	Phone Box	Non-Contributing	Small Scale Features	Object						
x			94149	Waysides	Non-Contributing	Small Scale Features	Object						
			94151	Parker's Battery earthworks	Contributing	Topography			6738	9000	use provided shapefile		
			94153	View of industrial land use	Non-Contributing	Views And Vistas							
			94154	View of utility corridor	Non-Contributing	Views And Vistas							
			C:\CLI_GIS\Northeast Region\NER_standardsGDB.gdb										
			50.00% (1 of 2) Contributing Features		68.42% (13 of 19) Total Features								

300100

Figure 11. Example of single landscape summary

Above is an example spreadsheet summary, and despite the illegibility, one can still see that a few rows are highlighted. These are features that are listed in the CLI for this landscape, but do not yet have GIS data to represent them.

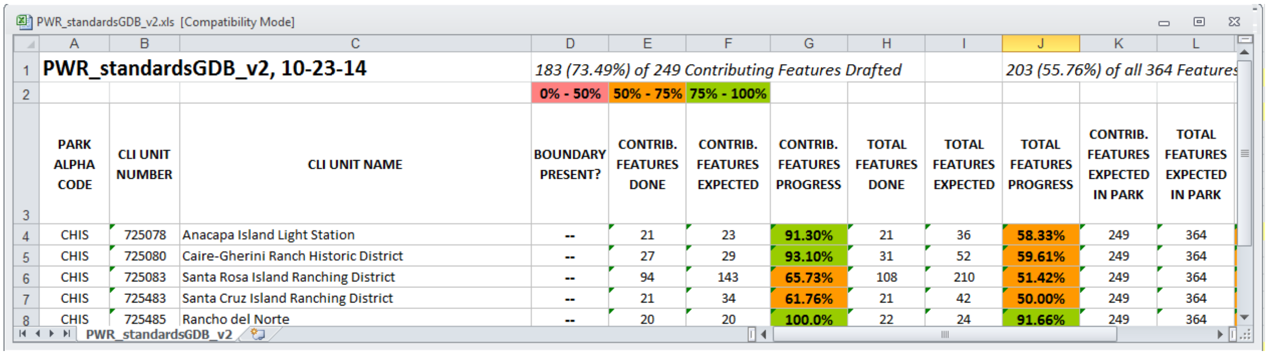
There are a few other aspects of the spreadsheet worth pointing out. Circled in red (on the left) is a set of hash-marks that indicate what type of geometry—point, line, or polygon—is currently being used to represent each feature. Circled in purple (bottom) is the path of the geodatabase whose spatial data has been summarized, as well as percentages of the expected contributing features and total features that have been represented with spatial data.

Circled in blue (top right) is the "comment" column. This column provides a storage place for comments about specific features, and is especially useful to hold explanations for a feature's absence from the spatial data, or suggestions on where to find it. When creating a new spreadsheet summary, the user can select an existing spreadsheet from which the comments on each feature should be pulled and incorporated into the new one.

The intention is for ONE spreadsheet to exist for each landscape, within the landscape folder in the recommended directory structure. The tool dialog is set up to facilitate this use: When a CLI number is entered, the tool dialog will look for an existing spreadsheet for that landscape and enter its path in the Get Comments From… parameter. It will also select to delete this old spreadsheet in order to seamlessly replace it with the new one. Thus: The new spreadsheet will have the updated geometry progress (i.e. less highlighted rows) but the same set of comments as the previous one. *However*, none of this is required to use the tool; spreadsheets can be made outside of the recommended directory structure, comments can be left untouched, and old spreadsheets need not be deleted.

--MULTIPLE LANDSCAPE SUMMARY SPREADSHEET--

Where the ***Excel File Single Landscape*** makes an annotated list of all features in a given landscape, the ***Excel File Multiple Landscapes*** tool will summarize the progress that has been made on many different landscapes. The result will look something like this:



PWR_standardsGDB_v2.xls [Compatibility Mode]

PWR_standardsGDB_v2, 10-23-14

183 (73.49%) of 249 Contributing Features Drafted

203 (55.76%) of all 364 Features

0% - 50% | 50% - 75% | 75% - 100%

PARK ALPHA CODE	CLI UNIT NUMBER	CLI UNIT NAME	BOUNDARY PRESENT?	CONTRIB. FEATURES DONE	CONTRIB. FEATURES EXPECTED	CONTRIB. FEATURES PROGRESS	TOTAL FEATURES DONE	TOTAL FEATURES EXPECTED	TOTAL FEATURES PROGRESS	CONTRIB. FEATURES EXPECTED IN PARK	TOTAL FEATURES EXPECTED IN PARK
CHIS	725078	Anacapa Island Light Station	--	21	23	91.30%	21	36	58.33%	249	364
CHIS	725080	Caire-Gherini Ranch Historic District	--	27	29	93.10%	31	52	59.61%	249	364
CHIS	725083	Santa Rosa Island Ranching District	--	94	143	65.73%	108	210	51.42%	249	364
CHIS	725483	Santa Cruz Island Ranching District	--	21	34	61.76%	21	42	50.00%	249	364
CHIS	725485	Rancho del Norte	--	20	20	100.0%	22	24	91.66%	249	364

Figure 12. Example of multiple landscape summary, showing progress of landscapes in Channel Islands National Park.

Instead of one row per feature, the multiple landscape spreadsheet has one row per landscape. For each landscape, the number of contributing features and number of total expected features are recorded, and percentages for these two categories are listed. Also, there is a marker for whether a landscape has a boundary or not.

When using the tool dialog, the user will enter a geodatabase, presumably one that has data for more than one landscape in it. Once entered, the tool dialog will make a list of all landscapes that have data in the geodatabase. From that list the user can choose which landscapes should be included in the summary.

Alternatively, if the object is to get a summary of all the landscapes in an entire region, even if there is no data in the geodatabase for some of the landscapes, the user can select a region to use in the dialog box. With this choice the spreadsheet will have a row for every landscape that is expected in the region.

PART 4: ACCESSING THE CULTURAL RESOURCE ENTERPRISE DATABASE

About the CR Enterprise Database

The CR Enterprise is a spatial database that is dedicated to storing cultural resource spatial data and serving it throughout the NPS network. Because the database functions as a repository, the CLI Toolbox is equipped with two tools that allow the user to analyze and download data from it. For the purposes of the CLI Toolbox, the only access to the CR Enterprise is through the layers in the CR Enterprise Access map document that is included in the toolbox. These layers are "views" of the corresponding feature classes in the enterprise database, which show only data that is marked as "Unrestricted". Thus, it is important to note that when viewing the data in the CR Enterprise Access map document, or when downloading/making spreadsheet summaries of that data using the map document (see below), any data that is marked as restricted (at any level) will be omitted.

Only the designated regional data editors will have edit-access to the CR Enterprise. The intention with the tools in the CR Enterprise Access toolset is to allow for Cultural Landscapes Program staff (or any NPS staff) to get local copies of the data, which can be analyzed, updated, used for fieldwork, etc. The protocol for getting updated data from landscape or park staff to the regional data editors will vary significantly from region to region, but by using the tools in the CLI Toolbox, the new or updated data should be presented to the data editor in a complete and standardized format.

Summarizing CLI Data in the Enterprise Database

The presence or absence of CLI features in the CR Enterprise spatial data can be summarized using the ***Summarize CLI Data in Enterprise*** tool. The tool produces MS Excel spreadsheets that are similar to the output of the other spreadsheet summary tools in the Review Tools toolset (see pp. 25-27). Also similar to the other tools, the user may choose between creating a summary of a single landscape or of multiple landscapes.

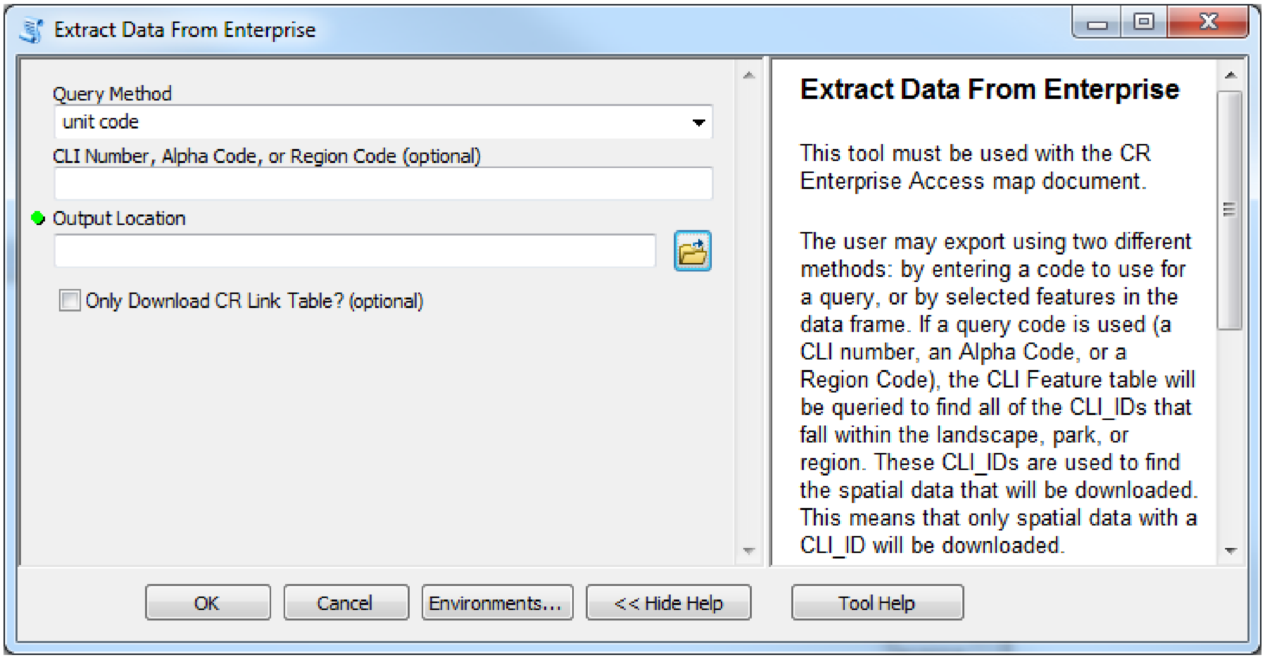
1. Open the CR Enterprise Access map document (found in the **CLI Tools** folder)
    a. You must be connected to the NPS network for the database connections to work. If you are on the network but still have connection issues, refer to the "readme" file in the **CLI Tools** folder.
2. Open the ***Summarize CLI Data in Enterprise*** tool, inside the CR Enterprise Access toolset. The CLI Toolbox should already be added to the map document. If this is not the case, refer to the directions at the top of page 10.
3. Choose whether you want to create a summary for a single landscape (which will list each feature), or a whole park or region (which will list each landscape). Enter a corresponding code.
4. Select an output location for the spreadsheet.
5. Click OK to run the process.

These are light-weight versions of the other spreadsheet tool outputs: In the single landscape summary, the type of geometry that represents each feature will not be shown, and there are no columns for tracking comments or editing operations. In the multiple landscape summary, the user does not have the option of omitting CLI features that are listed in the Archeological Site landscape characteristic, and must choose to list all of the landscapes in a park or region, not an arbitrary assortment of landscapes (e.g. a few different landscapes from a few different parks).

Creating these spreadsheet summaries is a good way to check on how many CLI features have spatial data in a given landscape, park, or region. This information will be useful in planning for future landscape work and on-the-ground data collection.

Acquiring Local Copies of Data from the Enterprise Database

It is therefore recommended that to edit, analyze, or distribute CLI data from the enterprise (all of which is, as mentioned above, unrestricted by nature of the connections to the database), a local copy should be made and used. To that end, the ***Extract Data From Enterprise*** tool was developed, which will create a local CLI Standards geodatabase from data in the enterprise. This tool gives the user two options for how the CR Enterprise should be queried: By unit code, or by using the currently selected features in the data frame. The latter option is provided in case the user would like to download features that do not yet have a CLI_ID. Alternatively, if the spatial data is not needed, the user can download just the CR Link table rows that correspond to the features that match the query method.


Extract Data From Enterprise

Query Method
unit code

CLI Number, Alpha Code, or Region Code (optional)

Output Location

□ Only Download CR Link Table? (optional)

Extract Data From Enterprise

This tool must be used with the CR Enterprise Access map document.

The user may export using two different methods: by entering a code to use for a query, or by selected features in the data frame. If a query code is used (a CLI number, an Alpha Code, or a Region Code), the CLI Feature table will be queried to find all of the CLI_IDs that fall within the landscape, park, or region. These CLI_IDs are used to find the spatial data that will be downloaded. This means that only spatial data with a CLI_ID will be downloaded.

Figure 13. Tool dialog for Extract Data From Enterprise

1. Open the CR Enterprise Access map document (found in the **CLI Tools** folder)
    a. You must be connected to the NPS network for the database connections to work. If you are on the network but still have connection issues, refer to the "readme" file in the **CLI Tools** folder.
2. Open the ***Extract Data From Enterprise*** tool, inside the CR Enterprise Access toolset. The CLI Toolbox should already be added to the map document. If this is not the case, refer to the directions at the top of page 10 to do so.
3. Choose whether you want to query the database by using an input code or by selecting features.
    a. Enter a unit code if necessary, otherwise be sure to have selected all relevant features using any of the normal ArcMap selection methods.
4. Select an output location for the resulting geodatabase.
5. Click OK to run the process.

When the process is finished, the user will have a CLI Standards geodatabase with all of the data that matched the query. This geodatabase can be displayed with various display options (p. 23), summarized (p. 25-27), exported to a Google Earth file (p. 25), have program IDs added to its CR Link table (p. 21), etc. The process of rolling up edited (or new) CLI GIS data from a park or regional office into the CR Enterprise database is one that will vary region by region, and is ultimately determined by the designated regional data editors. Hopefully, regardless of the specifics, some tools from the CLI Toolbox will assist the process along the way.

APPENDIX A: Glossary of Selected GIS Terminology

attribute table: The table of attributes, or field values, that are attached to the features in a spatial dataset.

ArcCatalog: The ArcGIS software used to browse and manage spatial datasets.

ArcGIS: A suite of software developed by Environmental Systems Research Institute (ESRI) for creating, analyzing, and managing spatial data.

ArcMap: The ArcGIS software used to create and edit map documents.

data frame: A container within a map document that can contain many layers, which represent spatial datasets.

dataset/spatial dataset: General term for a collection of spatial data, like a feature class or a shapefile. In ArcGIS, a spatial dataset contains a collection of spatial features that all have the same geometry type. A spatial dataset is a combination of geometry (the coordinates for each feature) and attributes which are stored in the attribute table and associated with each geometry.

definition query: A definition query may be applied to a layer in order to reduce the layer to only a subset of its original features. For example, the definition query "LAND_CHAR" = 'Vegetation' will show only features whose LAND_CHAR field value is 'Vegetation'.

geometry: The set of coordinates attached to each feature in a spatial dataset. In ArcGIS, there are many different types of geometry, but in the NPS Spatial Data Transfer Standards, only point, line, and polygon geometry types are used.

feature: Generally, "feature" refers to a single piece of spatial data—a point or polygon for example. However, throughout the documentation, cultural resources that are listed in the CLI are also referred to as "features", or "CLI features".

feature class: In more general GIS contexts, any collection of spatial data (or features) is considered a feature class. However, throughout this documentation, feature class refers *specifically* to a spatial dataset that is stored in a file geodatabase.

feature class dataset: This is a somewhat confusing term for a grouping of feature classes inside of a geodatabase. The CLI Standard geodatabase contains a feature class dataset for each cultural resource type, inside of which is a feature class for each acceptable geometry type (point, line, or polygon).

field: A column in any table, either an attribute table for a spatial dataset, or a non-spatial table such as the CR Link.

field value: Used to refer to a value for a feature that is stored in the attribute table, synonymous with "attribute".

geodatabase: There are two types of geodatabase, file geodatabases (*.gdb) and personal (*.mdb). Throughout the entire documentation and CLI Toolbox, only file geodatabases are used, so any reference to a geodatabase should be interpreted thusly. Viewed in Windows, a file geodatabase is a folder whose name ends with ".gdb" (inside of which are many tiny, nonsense files), and a personal geodatabase is an MS Access database.

GUID: A Globally Unique Identifier (GUID) is a long string of characters and numbers that is unique within a given database. In the CR Spatial Data Transfer Standards, GUIDs are used as cultural resource IDs (CR_ID) and geometry IDs (GEOM_ID).

layer: In this document, the term layer is reserved to mean a visual representation of a spatial dataset, such as a feature class or a shapefile. Layers exist in data frames within a map document, or can be saved to a *.lyr file.

map document: A document that is used to display, edit, and analyze spatial data.

parameter: A parameter is a value that is passed to a function. In the CLI Toolbox, each tool dialog is a list of parameters that the user enters to guide the process that the tool is about to carry out.

program ID: All other IDs for various cultural resource database, like the List of Classified Structures and the National Register, are referred to generally as program IDs through this documentation. Also, even though the facilities management system is not technically a cultural resource database, FMSS IDs are considered program IDs as well.

Python: An open-source programming language that is well-integrated into ArcGIS software, and can carry out tasks beyond ArcGIS as well. All of the code for the tools is written in Python, version 2.7.

shapefile: A shapefile is a common spatial data format. It is analogous to a feature class, but a feature class exists only within a geodatabase. Technically, a shapefile is a collection of three to seven files that work together. In ArcMap or ArcCatalog, a shapefile will have the extension ".shp".

spatial data: This is a general term for data that has a spatial component, e.g. coordinates or relational geometry.

tool: A tool is really two components: a dialog box into which the user will enter parameters, and the script behind it which takes the input parameters to carry out various operations.

toolbox: A collection of tools and/or toolsets.

toolset: A subset of tools within a toolbox, a purely organizational grouping.

APPENDIX B: Full List of CLI tools (with page references)

The following is a full list of all the tools in the CLI Toolbox and short description of each. Also, page number references are given for the tools that are described elsewhere in this document. See each tool's Item Description and embedded help for more detailed information.

CR Enterprise Access:

- *Feature Table Update* > ***Prepare CLI Table for Data Editor*** **(p.7)**
  - After following the instructions in 'Updating the CLI Feature Table', use this tool to convert the resulting MS Excel workbook into a geodatabase table whose rows can be used to replace the CLI Feature Table on the CR Enterprise.
- *Feature Table Update* > ***Update Local CLI Tables*** **(p. 7)**
  - Updates local toolbox tables using the CLI Feature Table that is stored in the CR Enterprise Database. Must be used from the CR Enterprise Access map document.
- ***Extract Data From CR Enterprise*** **(pp. 29-30)**
  - Export data from the CR Enterprise database. Must be used from the CR Enterprise Access map document.
- ***Summarize CLI Data In Enterprise*** **(p. 28)**
  - Creates a spreadsheet summary of the features that have GIS data for a given landscape, park, or region. Must be used from the CR Enterprise Access map document.

Editing Tools

- *Spatial Reference* > ***Set To GCS NAD83*** **(p. 12)**
  - Sets the active data frame coordinate system to a geographic coordinate system using North American Datum 1983.
- *Spatial Reference* > ***Set To State Plane (NAD83)*** **(p. 12)**
  - Uses the current extent of the active data frame to find the appropriate State Plane zone, and sets the data frame coordinate system to that projected coordinate system, using North American Datum 1983 (NAD83).
- *Spatial Reference* > ***Set To UTM (NAD83)*** **(p. 12)**
  - Uses the current extent of the active data frame to find the appropriate UTM (Universal Transverse Mercator) zone, and sets the data frame coordinate system to that projected coordinate system, using North American Datum 1983 (NAD83).
- ***Excel File From Features***
  - Converts rows in the attribute table for currently selected features in a given layer into an MS Excel workbook (*.xls).

- ***Generate Centroids* (p. 13, 17)**
  - Takes the input polygon or polyline features, creates a centroid for each one, and appends the centroid to the appropriate point feature class.
- ***Update Fields in Selected Records* (pp. 14-15, 17)**
  - Batch populate fields in all currently selected features
- ***Zoom To Unit* (p. 13, 17)**
  - Enter a CLI number, Alpha Code, or Region Code and zoom to features that match the code.

Review Tools

- ***Display Geodatabase* (p. 23)**
  - Choose from a number of predefined symbology schemes to display a CLI Standards or Scratch geodatabase.
- ***Excel File Multiple Landscapes* (p. 27)**
  - Create a summary spreadsheet showing feature progress percentages for multiple landscapes.
- ***Excel File Single Landscape* (p. 25)**
  - Create a summary spreadsheet showing progress for all features in a single landscape.
- ***Make Google Earth File* (p. 25)**
  - Convert the spatial data in a geodatabase into a Google Earth file (*.kmz) using the symbology schemes available in the Display Geodatabase tool.

Scratch GDB

- ***Create New Project Folder* (pp. 6, 10)**
  - Create a new project folder for a landscape. New folder contains scratch geodatabase and blank map document setup for data import, creation, and editing.
- ***Import To Scratch GDB* (p. 11)**
  - Import an existing spatial dataset (shapefile, feature class, etc.) into a geodatabase, and prepare it for NPS Spatial Data Transfer Standards fields.
- ***Sort Scratch Into Standards* (p. 12, 16-17)**
  - Sort multiple feature classes from a Scratch geodatabase into a CLI Standards geodatabase.

Standards GDB

- *Maintenance* > ***Check Mandatory Fields* (p. 17-18)**
  - Checks for Null or empty ("") values in all of the NPS Spatial Data Transfer Standards fields in a geodatabase.

- *Maintenance* > ***Check Values Against Domains*** **(pp. 17, 19)**
    - For any field with a domain assigned, checks all values in that field against the values in its assigned domain.
- *Maintenance* > ***Create GUIDs*** **(pp. 17, 19, 21)**
    - Create Globally Unique Identifiers (GUIDs) for features in a CLI Standards geodatabase.
- *Maintenance* > ***Fix Field in GDB*** **(pp. 17, 20-22)**
    - Change all occurrences of a given value in a given field across all feature classes in a geodatabase.
- *Maintenance* > ***Sync CR Link and Catalog Tables*** **(pp. 18-19)**
    - Synchronizes the CR Link and CR Catalog tables in a CLI Standards geodatabase.
- *Management* > ***Backup GDB***
    - Quickly copy a geodatabase to a backup location.
- *Management* > ***Copy Standards GDB To WGS84***
    - Make a copy of a CLI Standards geodatabase with all feature classes projected to a geographic coordinate system using datum World Geodetic System 1984 (WGS84).
- *Management* > ***Create Subset of Standards GDB***
    - Export a subset geodatabase from a larger geodatabase.
- *Management* > ***Export GDB To Shapefiles***
    - Converts all feature classes in a geodatabase into shapefiles.
- *Management* > ***Merge CR Link Tables***
    - Merge one CR Link table into another.
- *Management* > ***Merge Standards GDBs*** **(p. 17)**
    - Merge one CLI Standards geodatabase into another.

APPENDIX C: Using CLI Tools Outside of CLI Data

A number of the tools in this toolbox can be used outside of any CLI data/functions that the toolbox was initially designed for. Below are examples of some such uses. Please note that here, as in the rest of this document, "geodatabase" refers to an ESRI file geodatabase (*.gdb), not a personal geodatabase (*.mdb). Many of the tools below *may* work on personal geodatabases, but that has not been tested at all.

Accessing the CR Enterprise Database

Extract Data From Enterprise:

Though this tool is designed to support the download of CLI-specific data from the CR Enterprise database, the user can also download non-CLI-specific data by choosing to download selected features and selecting arbitrary features from anywhere in the CR Enterprise. A user could download all features that fall within a given state or city boundary, for example.

Specifically Supporting the NPS CR Spatial Data Transfer Standards

Check Mandatory Fields:

Creates a summary of all Null or empty ("") values that occur in any of the NPS CR Spatial Data Transfer Standards required fields. Analyzes all feature classes in a geodatabase.

Import to Scratch GDB:

The input dataset can be placed into any geodatabase, and its new copy will be augmented with all of the NPS CR Spatial Data Transfer Standards fields, with the option to prepopulate each one. Use this tool for general conversion of a feature class or shapefile to a feature class version that is compliant with the NPS CR Spatial Data Transfer Standards.

Supporting General Operations in Any Environment

Backup GDB:

Backs up any geodatabase, even during an open edit session (unsaved edits will not be reflected in the backup copy). Best used with a default backup location set in the tool parameter properties. (Definitely won't work on a personal geodatabase.)

Check Values Against Domains:

Checks the values of all fields that have a domain assigned against the values in that domain. Works on any geodatabase.

Excel File From Features:

Used in ArcMap, will create an MS Excel workbook out of the selected features in any input layer. Should work anywhere there is a layer with a selection applied to it.

Export GDB to Shapefiles:

Exports all feature classes from an input geodatabase to separate shapefiles in the output folder. Works on any geodatabase.

Fix Field in GDB:

Allows user to change a set of values that occur in a specific field in all feature classes to a new value. Works on any geodatabase.

Spatial Reference Tools:

All of the tools in the Editing > Spatial Reference toolset will work for any map document. They must be used in data view.

APPENDIX D: Python Installation Notes

The code used for all of the tools is written in Python 2.7. The structure of the CLI Toolbox was designed to meet four criteria:

1. Create a package that allows various tools and functions to reuse the same code
2. Include the xlrd and xlwt packages to allow for the creation of MS Excel workbooks
3. *Do not* require the user to install third-party Python packages or know what that means
4. Produce a structure that will facilitate future development and extended functionality for advanced GIS users.

In order to meet the first goal, a new Python package was developed called "clitools" to hold all of the new classes and functions that are used by the tools in the toolbox. The second goal was met by embedding these two packages in the new clitools package. The xlrd and xlwt modules were developed by David Giffin <david@giffin.org>, and they are freely distributable under a 4-clause BSD-style license (thanks Mr. Giffin!). The third goal was achieved by packaging all of this code in a way that it is easily accessible by all of the source scripts for the tools, and by setting all of the tools in the toolbox to look for relative paths to their source scripts. That way, wherever the **CLI Toolbox** folder is moved to, all of the tools will find the correct source scripts, and all of those source scripts will successfully import the necessary functions from within the clitools package.

Finally, the fourth goal was achieved by setting everything up to allow a future developer to move the clitools package into the local Python installation, without having to modify any code. When this is done, all of the functions and classes in the clitools package are available to any new Python application, as well as the Python console that is available in ArcMap and ArcCatalog.

See the following page for a more detailed explanation of the file structure, and an explanation of how to move the clitools package to the local Python installation.

CLI Toolbox/

- CLIToolbox.tbx *(toolbox added to ArcGIS; source scripts use relative paths)*
- **scripts**/
  - CalculateGUIDs_v1.6.py *(and all other source scripts for tools in CLIToolbox.tbx)*
  - **clitools**/ *(the python package that the source scripts use)*
    - **bin**/ *(folder holding all auxiliary data used by modules in package)*
    - __init__.py
    - classes.py
    - management.py
    - general.py
    - mxdops.py
    - summarize.py
    - enterprise.py

      custom-built Python modules within package
    - **xlrd**/
    - **xlrd**/

      embedded packages for reading/writing Excel files

All of the *.py files in the scripts folder import standard python libraries, or specific functions from the adjacent clitools package. For example, the CalculateGUIDs_v1.6.py script uses this statement:

```
import os
from clitools.management import CreateGUIDs
```

All of the actual functions for all of the tool operations are defined in the modules in the clitools package (classes.py, management.py, etc.). A few of these functions import the xlrd or xlwt modules using this standard import statement:

```
import xlrd
import xlwt
```

By embedding all of these packages within the toolbox folder, the user need not modify their local Python installation to use the new clitools module or the third-party xlrd and xlwt modules.

This means that all of the functionality in the clitools package is available with normal import statements for any *.py file that is located in the **scripts** directory. However, for any user interested in adding the clitools package (or xlrd and xlwt packages) to their local Python installation so the functions are available to any script anywhere, the conversion is very simple:

1. *Move* the entire clitools package to the local Python site-packages directory
2. *Move* the xlrd and xlwt packages from within the **clitools** folder to the local Python site-packages directory

(best practice would be to copy the packages, paste them to the site-packages directory, and after they have imported correctly (see below) delete the entire clitools package in the scripts directory)

The resulting structure of the site-packages directory will look like this:

```
...\site-packages\
        clitools\ (the clitools package moved from the scripts directory)
        xlrd\ (the xlrd package moved from within the clitools package)
        xlwt\ (the xlwt package moved from within the clitools package)
```

The site-packages directory is found here in standard Python installations:
C:\Python27\Lib\site-packages

or here when ArcGIS is installed:
C:\Python27\ArcGIS10.x\Lib\site-packages

To test whether the packages have been correctly moved to the local Python installation, open the start menu and type “idle” to start IDLE. In the console, try to import each new module by typing:

```
>>> import clitools, xlrd, xlwt
```

If any import errors occur, remove all packages that have been copied to the site-packages directory and the start process over. When all three modules import without error, delete the clitools package from its original location in the **scripts** directory (if you are skiddish about deleting things, you could just rename the clitools directory to “clitools1”). Because of the way the imports are constructed in each *.py file, all tools and modules will continue to function as if nothing has changed. However, now all of the modules in the clitools package—classes.py, mxdops.py, general.py, management.py, enterprise, and summarize.py—can be imported to any Python applications or the ArcGIS console.

The following is an example of how to use the clitools MakeUnit function in IDLE to list the landscapes in a park (Sleeping Bear Dunes National Lakeshore):

```
>>> from clitools.classes import MakeUnit
>>> unit = MakeUnit("SLBE")
>>> for landscape in unit.landscapes:
        print landscape

(u'450004', 'South Manitou Island Light Station')
(u'500003', 'Port Oneida Rural Historic District')
(u'500330', 'North Manitou Island Life-Saving Station')
```

```
(u'975462', 'Glen Haven')
(u'975538', 'Bufka, Kropp, Eitzen Agricultural District')
(u'975586', 'North Manitou Island - Cottage Row')
```

For more help on any function or module, try the following lines after the import statement in the example above has been used:

```
>>> help(MakeUnit)
>>> help(clitools.classes)
```

Alternatively, all help from the documentation for the clitools modules and functions can be found in the **python documentation** folder.