GDB_DataReview ArcMap Toolbox Tutorials

Steven C. Gonzalez & Marie C. Cline
Air Force Civil Engineering Center (AFCEC)
Geospatial Integration Office (GIO)

2018-05-14

Contents

Chapter 1

Toolbox Overview

This document gives an overview of how to use the GDB_DataReview ArcMap Toolbox.

The GDB_DataReview ArcMap Toolbox provides numerous Python script tools to expedite the review of geodatabases in comparison with a template geodatabase model. This toolbox was developed to aid Air Force (AF) installations in maintaining geospatial data in compliance with the current Air Force Data Model (GeoBase 3.1.0.1) developed under the AF GeoBase mission.

The GeoBase 3.1.0.1 data model is based upon the Spatial Data Standards for Facilities, Infrastructure, and Environment (SDSFIE) SDSFIE-V 3.1 Gold model, which complies with the Department of Defense Instruction (DoDI) 8130.01, *Installation Geospatial Information and Service* (IGI&S), but allows some greater flexibility within the program to aid the AF mission. As part of the IGI&S program, this toolbox also aids in standardizing methods to adhere to the Fiscal Year 2017 CIP data call required by DoDI 8130.01.

The GDB_DataReview Toolbox provides methods to:

- Update feature class data using both tabular and spatial joins,
- Find duplicate geometries, delete duplicate features, and check/repair feature geometries,
- Standardize road prefixes, names and suffixes or building addresses,
- Search for and summarize indeterminant and missing data in a geodatabase when compared with a template geodatabase
- Batch exporting and importing geodatabase metadata

Chapter 2

Opening the Toolbox

In order to access this toolbox, open ArcCatalog (or the ArcCatalog window within ArcMap). Within the ‘Catalog Tree’ windowpane in ArcCatalog, click the Folder Connections folder and navigate to the location of the toolbox. You may need to add a folder connection by right clicking ‘Folder Connections’ and clicking ‘Connect to Folder.’ Then navigate to the folder with the toolbox within the newly connected folder. Then, you can click the plus sign by the toolbox to view the tools within it (Fig. 2.1).

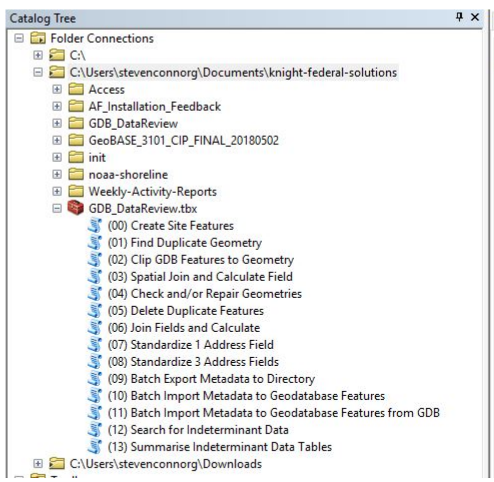


Figure 2.1: Opening the Create Site Data tool

Chapter 3

Create Site Data

3.1 Overview

The Create Site Data tool allows users to use the Cadastre dataset's Installation_A, Site_A, and Site_P feature classes to populate and update site data as needed.

This tool compares the geometry of Installation_A data (required to be populated) to Site_A data and populates features where needed. Upon the update of Site_A, points are created in Site_P for each feature in Site_A if they do not exist.

The user has the option to bypass the geometry compare between Installation_A and Site_A. When bypassed, no features are added to Site_A and site points are created using data that already exists in Site_A.

3.2 Parameters

The tool has 3 parameters:

1. **Input_Geodatabase (data type: Workspace)** - This parameter must be the path of the input geodatabase to search Feature Datasets' Feature Class features for duplicate features.

2. **Bypass Installation_A and Site_A Geometry Compare (data type: Boolean)** - This parameter is a check box that is unchecked by default. If the user checks the box, the geometry compare between Installation_A and Site_A will be bypassed.
3. **Installation_A & Site_A Geometry Compare Type (data type: String Value List)** - How do you want to limit the spatial join? By default, this parameter is set to "HAVE THEIR CENTER IN," in order to only update target features that have their center in the source features. This parameter may be changed to any of the following values, as specified in the SelectByLocation_management tool documentation:

- *INTERSECT* —The features in the input layer will be selected if they intersect a selecting feature. This is the default.
- *INTERSECT_3D* —The features in the input layer will be selected if they intersect a selecting feature in three-dimensional space (x, y, and z).
- *WITHIN_A_DISTANCE* —The features in the input layer will be selected if they are within a specified distance of a selecting feature. Specify a distance in the Search Distance parameter.
- *WITHIN_A_DISTANCE_3D* —The features in the input layer will be selected if they are within a specified distance of a selecting feature in three-dimensional space. Specify a distance in the Search Distance parameter.
- *WITHIN_A_DISTANCE_GEODESIC* —The features in the input layer will be selected if they are within a specified distance of a selecting feature. Distance between features will be calculated using a geodesic method which takes into account the curvature of the earth and correctly deals with data near and across the dateline and poles.
- *CONTAINS* —The features in the input layer will be selected if they contain a selecting feature.
- *COMPLETELY_CONTAINS* —The features in the input layer will be selected if they completely contain a selecting feature.
- *CONTAINS_CLEMENTINI* —This spatial relationship yields the same results as COMPLETELY_CONTAINS with the following exception: if the selecting feature is entirely on the boundary of the input feature (no part is properly inside or outside), the feature will not be selected. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.
- *WITHIN* —The features in the input layer will be selected if they are within a selecting feature.
- *COMPLETELY_WITHIN* — The features in the input layer will be selected if they are completely within or contained by a selecting feature.
- *WITHIN_CLEMENTINI* — The result will be identical to WITHIN with the exception that if the entirety of the feature in the input layer is on the boundary of the feature in the selecting layer, the feature will not be selected. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.
- *ARE_IDENTICAL_TO* — The features in the input layer will be selected if they are identical (in geometry) to a selecting feature.

3.3 How to Use

3.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Create Site Data' script tool to open (Fig. 3.1).

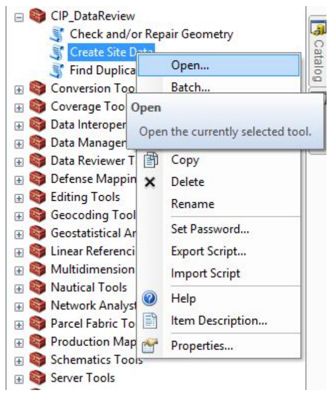


Figure 3.1: Opening the Create Site Data tool

3.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to select the Cadastre feature dataset for the geodatabase being processed. (Fig. 3.2). Last, we specify where we want to output the resulting tables, preferebly in an Installation Review geodatabase specifically for holding CIP processing results.

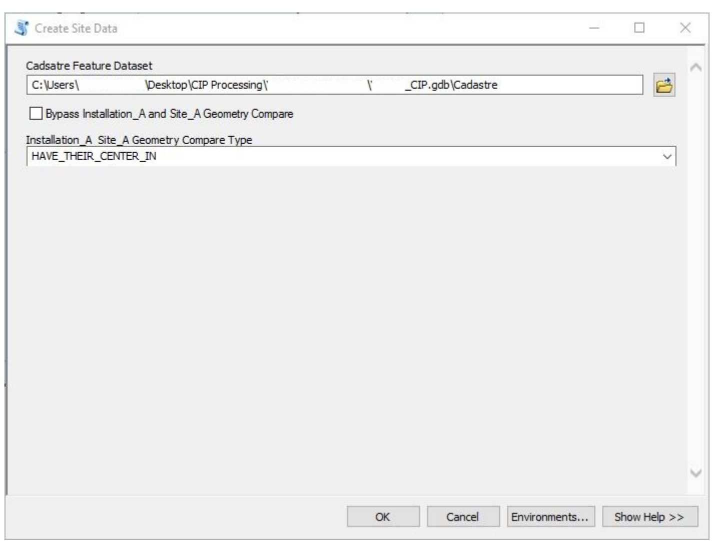


Figure 3.2: Create Site Data Tool parameters

3.3.3 Run the Tool and View Results

While the tool runs (with Background Processing disabled), we can see messages and warnings from the tool. Warnings are provided if required feature classes are missing or data verification from the user is suggested. Messages include the number of features added to Site_A and the number of points added to Site_P (Fig. 7.3). Here, we see that an Installation_A feature exists beyond the boundaries of the Site_A features so 1 feature is appended to Site_A. Then, 8 site points were created for the empty Site_P feature class.

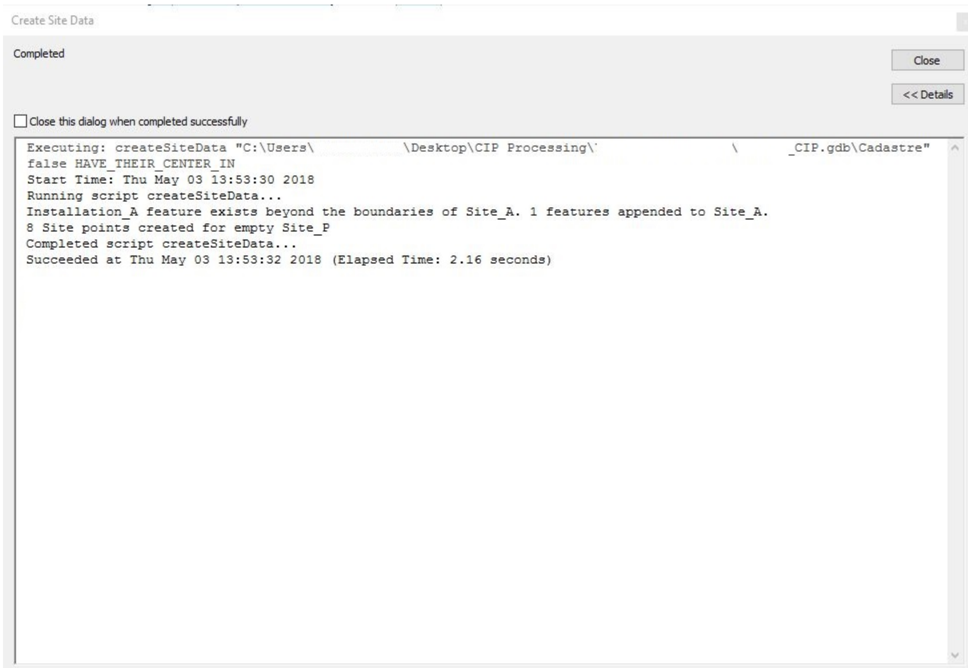


Figure 3.3: Create Site Data tool messages

After running the tool, we can see the Site_P and Site_A features are created properly, where previously missed (Fig. 3.4 & Fig. 3.5).

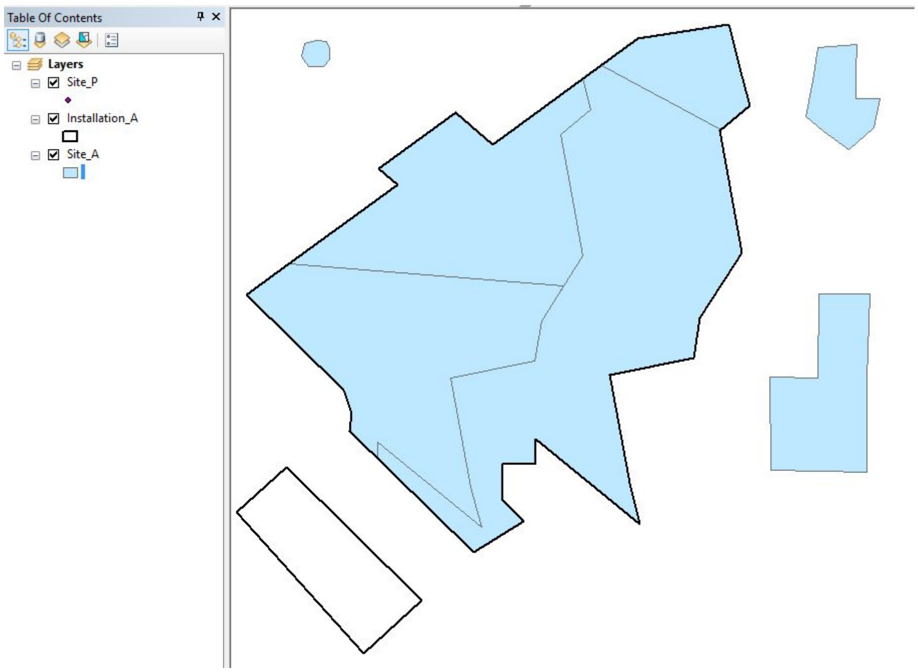


Figure 3.4: Before running the tool

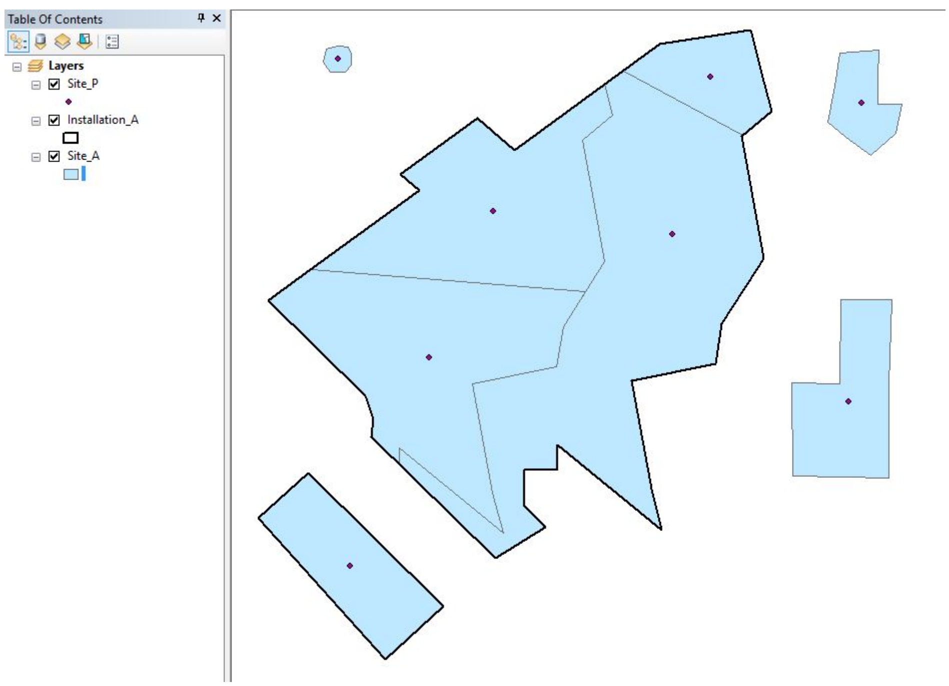


Figure 3.5: Newly created Site A and Site P features after running the tool

Chapter 4

Join Fields and Calculate

4.1 Overview

The ArcGIS Python Script Tool "Join Fields and Calculate" may be used to update the destination values in a target feature layer field with the values in another table's fields using a common key (join). This script will perform similarly as if you joined a table to a feature class to calculate a certain field based on another field in the joined table.

4.2 Parameters

The tool has 8 parameters:

1. **Transfer_From (data type: Table View)** - Which table are do you want to transfer data from? This parameter must be the path to a table(e.g.: Comma-separated Values (.csv) file, Excel Workbook (.xlsx) Sheet, Esri geodatabase table, etc.). This table will act as 'source' data.
2. **Using_Join_Field (data type: Field)** - From the source table, which field should be used to joinwith another feature class' attributes? This will provide the 'key' to transfer data from the source table to the target table.
3. **Source_Field (data type: Field)** - From the source table, which field's data do you want to transfer to the target table? This field's data will be updated in the target feature class that have matching fields.
4. **Destination_Feature (data type: Feature Layer or Feature Class)** - Which feature class do you want to transfer data to? This parameter must be the path to a Esri Feature Class or Feature Layer. This table will act as 'target' data source.
5. **Destination_Join_Field (data type: Field)** - From the target table, which field should be used to joinwith another feature class' attributes? This will provide the 'key' to transfer data from the source table to the target table.

6. **Destination_Field (data type: Field)** - From the target table, which field's data do you want to transfer from the source table? This field's data will be updated from the source table that have matching fields using the join fields provided.

7. **Where_Clause (data type: String)** - How should the source values be filtered? Default is "IS NOT NULL", otherwise you will overwrite the target features will null values.

8. **Remove_Leading_Zeros (data type: Boolean)** - Do you want to remove leading zeros from the Source Join Field prior to 'joining' the tables?

9. **Source RPSUID Field (data type: Field)** - Which field in the source table holds the Real Property Site Unique ID values? This field acts as a second join "key" to ensure that the correct Real Property Unique IDs are joined for each unique Site (i.e.: RPSUID).

10. **Update RPSUID Field (data type: Field)** - Which field in the Destination Feature holds the Real Property Site Unique ID values? This field acts as a second join "key" to ensure that the correct Real Property Unique IDs are joined for each unique Site (i.e.: RPSUID).

4.3 How to Use

4.3.1 Begin by opening the toolbox

Navigate to the location of the script tool, then right-click the 'Join Fields and Calculate' script tool to open (Fig. 4.1).

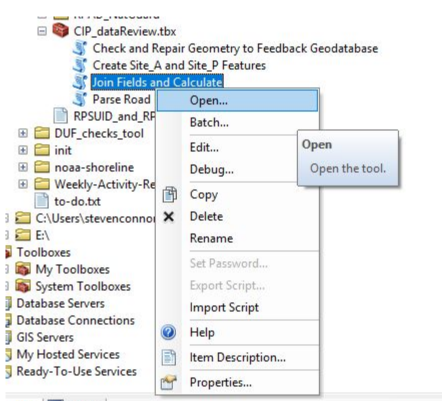


Figure 4.1: Opening the Tool

4.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to transfer the RPUID attributes (source field) from the 'RPSUID_and_RPUID.csv' table (transfer from) using the 'FacilityNumber' join field (Using_Join_Field) to the Building_A feature layer's (Destination Feature) 'realPropertyUniqueID' field (Destination_Field) using the 'buildingNumber' field (Destination_Join_Field) (Fig. 4.2).

We also keep the default value in the 'Where Clause' parameter of 'IS NOT NULL,' in order to transfer RPUID from the source table where RPUIDs are not null, **otherwise you may overwrite the target features will null values** (Fig. 4.2).

We noticed that the 'buildingNumber' field has some leading zeros that we want to remove the beginning of the values, so we click the "Remove Leading Zeros" toggle (Fig. 4.2). If you noticed that the 'Destination Join Field' values have leading spaces, you can also check the 'Remove_Leading_Zeros' parameter to remove these spaces.

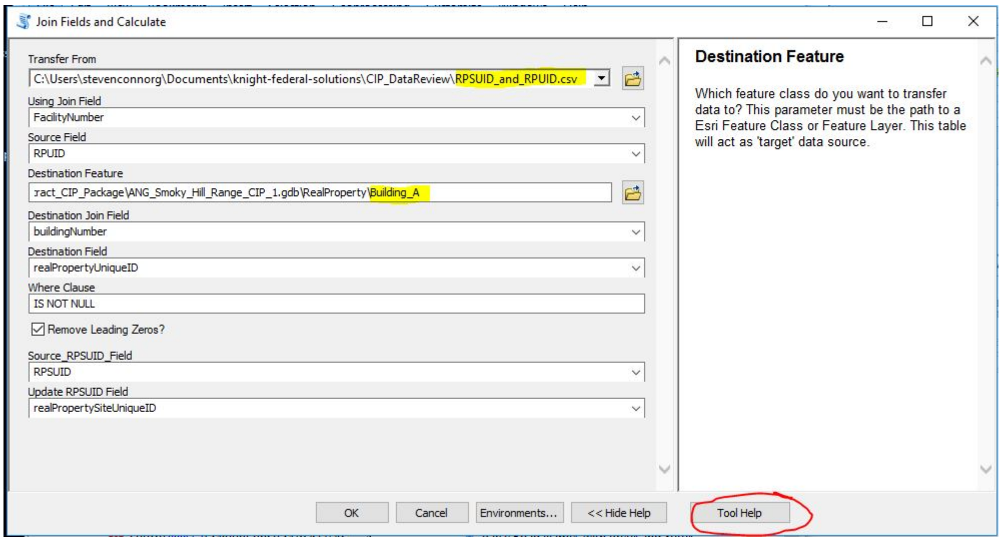


Figure 4.2: Tool parameters

Alternatively, you may also run this tool in 'batch' for multiple features in a geodatabase or geodatabases (Fig. 4.3).

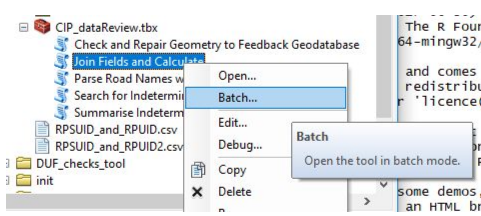


Figure 4.3: Running a tool in batch

You may also get more information for the tool and each tool parameter by clicking the 'Tool Help' button at the bottom of the tool dialog box.

4.3.3 Run the Tool and View Results

Open the destinate Feature Class and view the update destination field values (Fig. 4.4, Fig. 4.5).

Building_A

buildingNumber	realPropertyUniqueIdentifier	realPropertySiteUniqueID
<Null>		2059
1		1644
3		1644
5		1644
7		1644
8		1644
16		1644
1		2055
1		2055
1		2056
1		2059
1		2065
1		2056
1001		2055
1002		2064
1003		2055
1005		2055
1007		2055
1009		2055
101		2055
1011		2055
1012		2055

Figure 4.4: Attribues before running the Join Fields and Calculate tool

Building_A

buildingNumber	realPropertyUniqueIdentifier	realPropertySiteUniqueID
<Null>		2059
1	409435	1644
3	409457	1644
5	409459	1644
7	409462	1644
8	409463	1644
16	409453	1644
1	429546	2055
1	429546	2055
1	434992	2056
1	436407	2059
1	432253	2065
1	434992	2056
1001	429549	2055
1002	432237	2064
1003	429551	2055
1005	429553	2055
1007	429555	2055
1009	429556	2055
101	429557	2055
1011	429558	2055
1012	429559	2055

Figure 4.5: Attributes after running the tool, matching against both Building Number and RPSUID to update RPUID values

Chapter 5

Spatial Join and Calculate

5.1 Overview

This tool utilizes spatial joins to update field values in the target Feature Classes field to equal the source Feature Class fields in a source geodatabase. Using 'wildcard' fitlers, this tool allows users to update particular target Feature Datasets, Feature Classes, and Fields. For the purposes of this tool within the scope of the CIP Data Review task, target Fields are, by default, any fields that begin with "realPropertySiteUnique," in order to udpate RPSUID fields called either "realPropertySiteUniqueIdentifier" or "realPropertySiteUniqueID"; however, this tool could be extended to any number of source/target Feature Class/Field values.

5.2 Parameters

The tool has 8 parameters:

1. **Update Geodatabase (data type: Workspace/File Geodatabase)** - The path to the input geodatabase to update Feature Classes in.
2. **Source Feature (data type: Feature Class)** - The path to the source Feature Class, which will be used to update Feature Class fields in target Feature Classes.
3. **Source_Field (data type: Field)** - The field within the source Feature Class used to update values in target Feature Classes.
4. **Target Feature Dataset Wildcard (data type: String)** - Within the input geodatabase, do you want to update only certain Feature Datasets? Use this wildcard to filter input geodatabase Feature Datasets. The Default is '*' for 'All Feature Datasets,' but if you only wanted to update the Feature Classes in the 'Auditory' Feature Dataset, set this parameter to 'Auditory.' Similarly, if you only wanted to update Feature Classes within environmental Feature Datasets, set this parameter to 'environmental*', which will loop through all Feature Classes within Feature Datasets that start with 'environmental.'

5. **Target Feature Class Wildcard (data type: String) -** Within the input geodatabase, do you want to update only certain Feature Classes? Use this wildcard to filter input geodatabase Feature Classes to update. The Default is '*' for 'All Feature Classes,' but if you only wanted to update Feature Classes called "roadCenterline_L", set this parameter to 'roadCenterline_L.' Similarly, if you only wanted to update Feature Classes that begin with "road," set this parameter to 'road*', which will loop through all Feature Classes that start with 'road.'

6. **Target Field Wildcard (data type: String) -** This parameter is used to filter fields within the target Feature Classes that you want to update with the Source Feature Classes source Field. For the purposes of this tool within the scope of the CIP Data Review, this parameter is automatically set to "realPropertySiteUnique*" in order to 'catch' all RPSUID fields within the SDSFIE 3.101 data model, where certain fields are called "realPropertySiteUniqueIdentifier" and others are called "realPropertySiteUniqueID."

7. **Overlap Type (data type: String) -** How do you want to limit the spatial join? By default, this parameter is set to "within," in order to only update target features that are completely within the source features. This parameter may be changed to any of the following values, as specified in the SelectByLocation_management tool documentation:

    - *INTERSECT* —The features in the input layer will be selected if they intersect a selecting feature. This is the default.

    - *INTERSECT_3D* —The features in the input layer will be selected if they intersect a selecting feature in three-dimensional space (x, y, and z).

    - *WITHIN_A_DISTANCE* —The features in the input layer will be selected if they are within a specified distance of a selecting feature. Specify a distance in the Search Distance parameter.

    - *WITHIN_A_DISTANCE_3D* —The features in the input layer will be selected if they are within a specified distance of a selecting feature in three-dimensional space. Specify a distance in the Search Distance parameter.

    - *WITHIN_A_DISTANCE_GEODESIC* —The features in the input layer will be selected if they are within a specified distance of a selecting feature. Distance between features will be calculated using a geodesic method which takes into account the curvature of the earth and correctly deals with data near and across the dateline and poles.

    - *CONTAINS* —The features in the input layer will be selected if they contain a selecting feature.

    - *COMPLETELY_CONTAINS* —The features in the input layer will be selected if they completely contain a selecting feature.

    - *CONTAINS_CLEMENTINI* —This spatial relationship yields the same results as COMPLETELY_CONTAINS with the following exception: if the selecting feature is entirely on the boundary of the input feature (no part is properly inside or outside), the feature will not be selected. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.

- *WITHIN* —The features in the input layer will be selected if they are within a selecting feature.

- *COMPLETELY_WITHIN* — The features in the input layer will be selected if they are completely within or contained by a selecting feature.

- *WITHIN_CLEMENTINI* — The result will be identical to WITHIN with the exception that if the entirety of the feature in the input layer is on the boundary of the feature in the selecting layer, the feature will not be selected. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.

- *ARE_IDENTICAL_TO* — The features in the input layer will be selected if they are identical (in geometry) to a selecting feature.

- *BOUNDARY_TOUCHES* — The features in the input layer will be selected if they have a boundary that touches a selecting feature. When the inputs features are lines or polygons, the boundary of the input feature can only touch the boundary of the selecting feature, and no part of the input feature can cross the boundary of the selecting feature.

- *SHARE_A_LINE_SEGMENT_WITH* — The features in the input layer will be selected if they share a line segment with a selecting feature. The input and selecting features must be line or polygon.

- *CROSSED_BY_THE_OUTLINE_OF* — The features in the input layer will be selected if they are crossed by the outline of a selecting feature. The input and selecting features must be lines or polygons. If polygons are used for the input or selecting layer, the polygon's boundary (line) will be used. Lines that cross at a point will be selected, not lines that share a line segment.

- *HAVE_THEIR_CENTER_IN* — The features in the input layer will be selected if their center falls within a selecting feature. The center of the feature is calculated as follows: for polygon and multipoint, the geometry's centroid is used, and for line input, the geometry's midpoint is used.

8. **Search Distance (data type: String) -** If the overlap type parameter is equal to "WITHIN_A_DISTANCE_GEODE "WITHIN_A_DISTANCE", "WITHIN_A_DISTANCE_3D", "INTERSECT", "INTERSECT_3D", "HAVE_THEIR_CENTER_IN", "CONTAINS", or "WITHIN", you must provide a search distance.

5.3 How to Use

5.3.1 Begin by opening the toolbox

Navigate to the location of the script tool, then right-click the 'Calculate Feature RPSUIDs from Overlapping Polygon' script tool to open (Fig. 5.1).

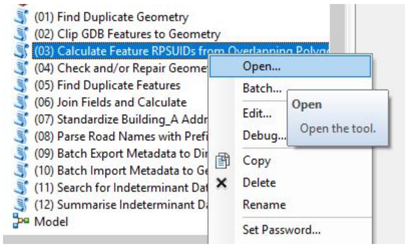


Figure 5.1: Opening the toolbox

5.3.2 Fill out the parameters

For this demostration, we want to update missing RPSUID values for 2 features in the Site_P Feature Class using RPSUID values from Site_A features that contain Site_P features 5.2).

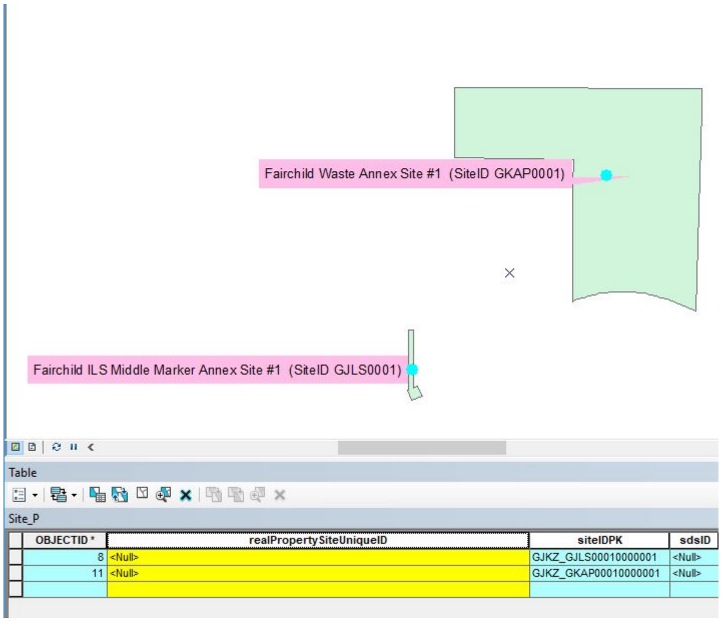


Figure 5.2: Missing RPSUID attributes for Site Point features

Next, fill out the parameters for the tool. Here, we want to transfer the RPSUID attributes (Source Field) from the Site_A Feature Class in the Cadastre Feature Dataset (Fig. 5.3).

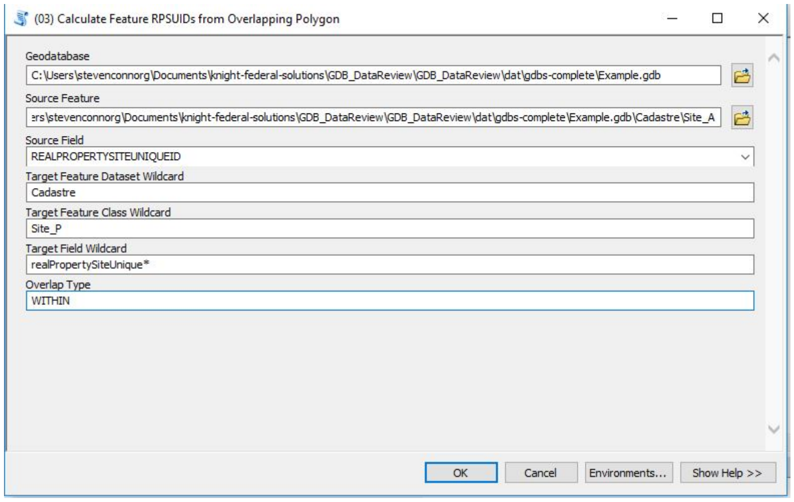


Figure 5.3: Tool parameters

Since we only want to update the Site_P features within the Cadastre Feature Dataset, we change the default value for the Target Feature Dataset Wildcard to "Cadastre," since we know that the Site_P Feature Class is only found within the Cadastre Feature Dataset. Further, we change the default value of the Target Feature Class Wildcard parameters to "Site_P" in order to only update Site_P features within the Cadastre Dataset. Since we know that the RPSUID field names within all Feature Classes in the data model begin with 'realPropertySiteUnique', we can keep the default value for the Target Field Wildcard parameter in order to update the realPropertySiteUniqueID field in Site_P features with with the Source Field in the Source Feature Class. For the purposes of this demostration, we keep the default value for the Overlap Type parameter to "WITHIN," in order to update the fields that begin with "realPropertySiteUnique" for features that are *within* each Source Feature Class feature. You may also get more information for the tool and each tool parameter by clicking the 'Tool Help' button at the bottom of the tool dialog box.

5.4 Run the Tool and View Results

If running the tool with Background Processessing disabled, we can see which RPSUIDs are being updated (Fig. 5.4).

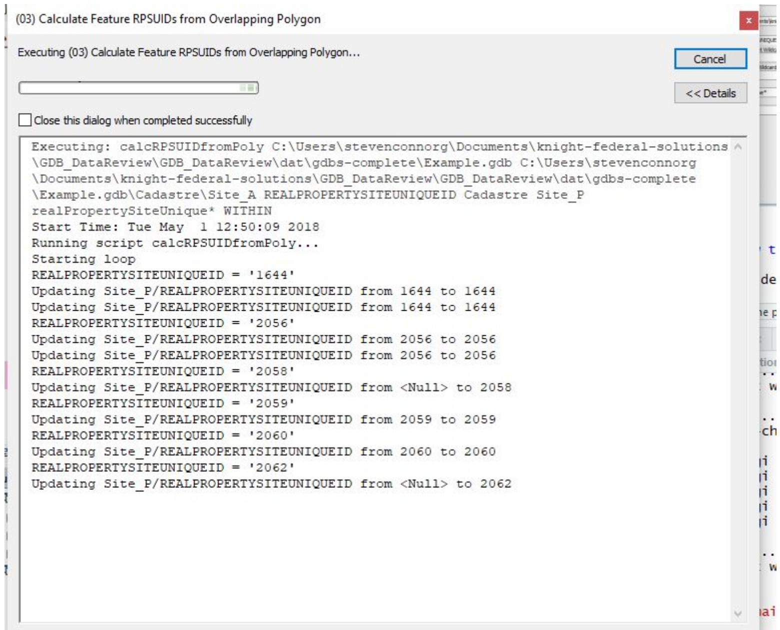


Figure 5.4: Tool parameters

After the tool has run, we see that the 2 Site_P features with missing RPSUID values are updated accordingly 5.5).

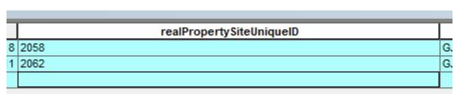


Figure 5.5: AFter running the tool, new Site A feaures and new Site P features are created.

Chapter 6

Find Duplicate Geometry

6.1 Overview

The Find Duplicate Geometry tool allows users to search an entire geodatabase's Feature Classes within Feature Datasets for features with duplicate geometries. This tool loops through each Feature Dataset's Feature Class features and searches for duplicate geometries.

All features with duplicate geometries are written to the output .csv file, as specified, and describes the Feature Dataset and Feature Class with duplicate geometries, the OBJECTIDs of the duplicate geometries, and a summary, which gives the count of duplicate geometries spread over unique geometries. Further, this tool creates layer files for each Feature Class' duplicate features, *allowing users to edit their geodatabase directory from a temporary, filtered layer of only duplicate features to be evaluated further, instead of arbitrarily deleting duplicated features without further consideration.*

6.2 Parameters

The tool has 5 parameters:

1. **Input_Geodatabase (data type: Workspace) -** This parameter must be the path of the input geodatabase to search Feature Datasets' Feature Class features for duplicate geometries.

2. **XY_Tolerance (data type: String) -** The XY_Tolerance parameter will be applied to each vertex when evaluating if there is an identical vertex in another entity, and must be input in the same units as the the source geodatabase's coordinate reference system (CRS).
3. **Z_Tolerance (data type: String) -** The Z_Tolerance parameter will be applied to each vertex when evaluating if there is an identical vertex in another entity with regard to elevation, and must be input in the same units as the the source geodatabase's coordinate reference system (CRS).
4. **Output_CSV (data type: File) -** The path to the output Duplicate_Geometry_Summary .xlsx/.csv file.

5. **Output_Layers_Directory (data type: Folder) -** The path to the directory/folder to store layer files with duplicate geometries.

6.3 How to Use

6.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Find Duplicate Geometry' script tool to open (Fig. 6.1).

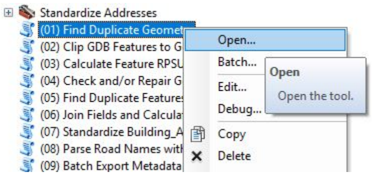


Figure 6.1: Opening the Find Duplicate Geometries tool

6.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to search all Feature Classes within Feature Datasets in the Example.gdb for duplicate geometries using the default XY Tolerance and Z Tolerance values of '0' (Fig. 6.2). We specify that we want the Duplicate Geometry Summary to be written to a Comma-separated Values (.csv) file called 'test.csv.' Further, we specify that we want all the duplicate Feature Class feature layers to be saved to the Output Layers Directory 'layer.'

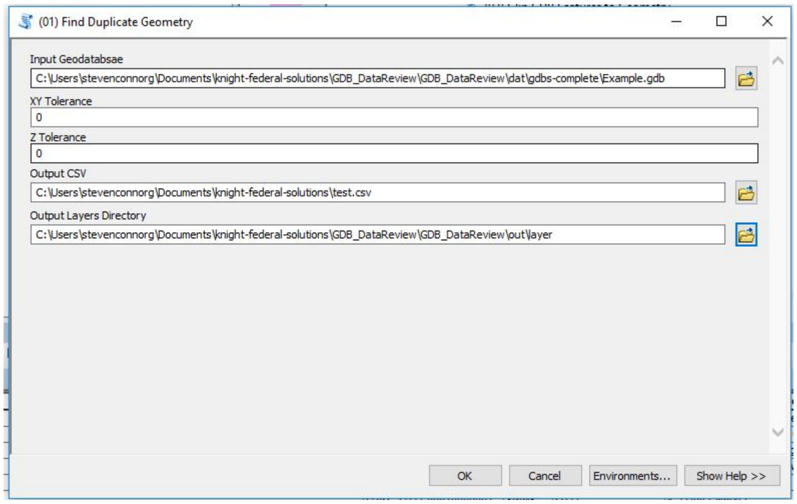


Figure 6.2: Find Duplicate Geometries parameters

6.4 Run the Tool and View Results

While the tool runs (with Background Processing disabled), we can see the messages from the tool, showing how many duplicate features are found for each Feature Class (Fig. 6.3).

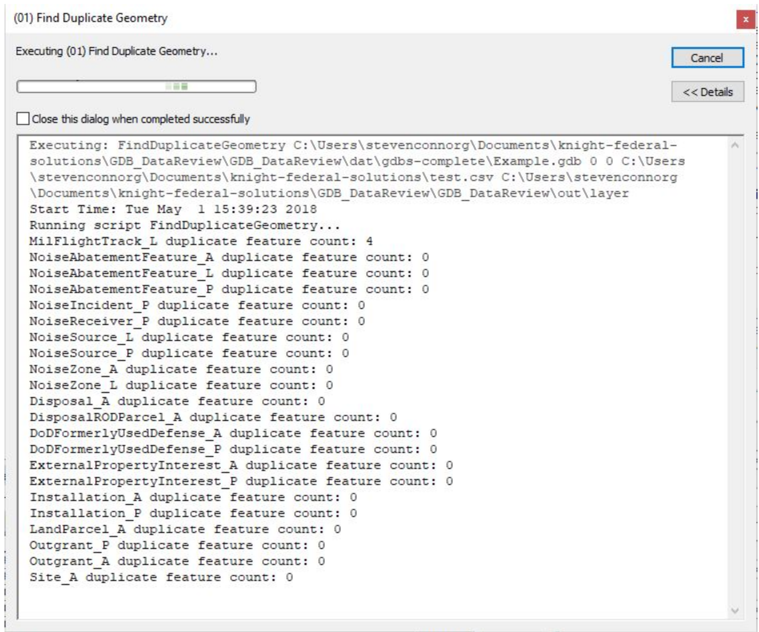
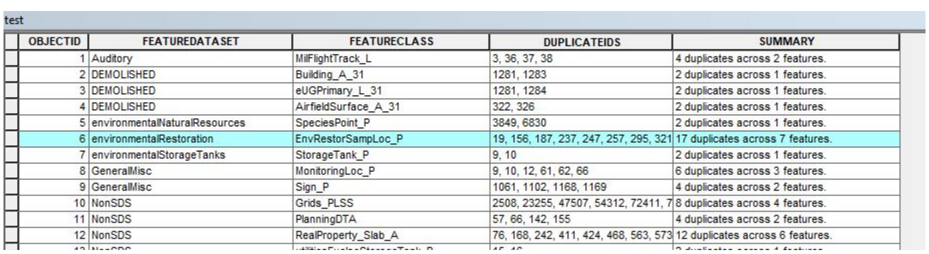


Figure 6.3: Messages from the Find Duplicate Geometries tool

After the tool has run, we can open the output .csv we specified in the tool parameters to examine which Feature Classes have duplicated geometries . For example, we find that the EnvRestorSampLoc_P Feature Class within the environmentalRestoration Feature Dataset has 17 total duplicates spread across 7 unique geometries (Fig. 6.4).


test

OBJECTID	FEATUREDATASET	FEATURECLASS	DUPLICATEIDS	SUMMARY
1	Auditory	MilFlightTrack_L	3, 36, 37, 38	4 duplicates across 2 features.
2	DEMOLISHED	Building_A_31	1281, 1283	2 duplicates across 1 features.
3	DEMOLISHED	eUGPrimary_L_31	1281, 1284	2 duplicates across 1 features.
4	DEMOLISHED	AirfieldSurface_A_31	322, 326	2 duplicates across 1 features.
5	environmentalNaturalResources	SpeciesPoint_P	3849, 6830	2 duplicates across 1 features.
6	environmentalRestoration	EnvRestorSampLoc_P	19, 156, 187, 237, 247, 257, 295, 321	17 duplicates across 7 features.
7	environmentalStorageTanks	StorageTank_P	9, 10	2 duplicates across 1 features.
8	GeneralMisc	MonitoringLoc_P	9, 10, 12, 61, 62, 66	6 duplicates across 3 features.
9	GeneralMisc	Sign_P	1061, 1102, 1168, 1169	4 duplicates across 2 features.
10	NonSDS	Grids_PLSS	2508, 23255, 47507, 54312, 72411, 7	8 duplicates across 4 features.
11	NonSDS	PlanningDTA	57, 66, 142, 155	4 duplicates across 2 features.
12	NonSDS	RealProperty_Slab_A	76, 168, 242, 411, 424, 468, 563, 573	12 duplicates across 6 features.

Figure 6.4: Find Duplicate Geometries parameters

Navigating to the output layer directory we specified in the tool, we find layer files with duplicate features for each Feature Class (Fig. 6.5).

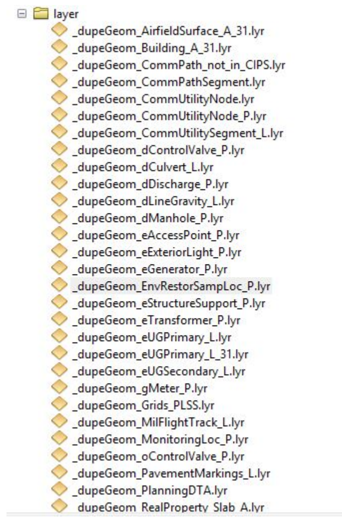


Figure 6.5: Find Duplicate Geometries parameters

After pulling in the _dupeGeom_EnvRestorSampLoc_P layer file, we can zoom to a feature and select the features at that location to examine the duplicate features at that location (Fig. 6.6).

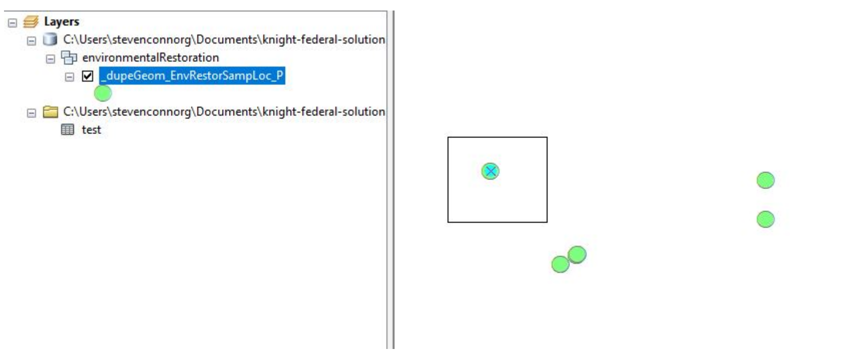
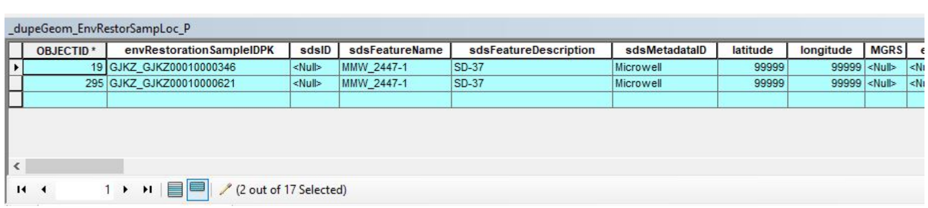


Figure 6.6: The features with duplicated geometries

Then, we can view the Attribute Table for the selected features to examine which feature we should amend or delete (Fig. 6.7). Here, we find that the attributes are exactly the same for the first duplicated geometry, and so we should probably delete one of these features. Editing the layer files directly will update the associated Feature Classes in the original geodatabase.



_dupeGeom_EnvRestorSampLoc_P

OBJECTID *	envRestorationSampleIDPK	sdsID	sdsFeatureName	sdsFeatureDescription	sdsMetadataID	latitude	longitude	MGRS
19	GJKZ_GJKZ00010000346	<Null>	MMW_2447-1	SD-37	Microwell	99999	99999	<Null>
295	GJKZ_GJKZ00010000621	<Null>	MMW_2447-1	SD-37	Microwell	99999	99999	<Null>

1 (2 out of 17 Selected)

Figure 6.7: Examining the attributes of duplicated features from layer files

Chapter 7

Delete Duplicate Features

7.1 Overview

The Delete Duplicate Features tool allows users to search an entire geodatabase's Feature Classes for duplicated features This tool loops through each Feature Dataset's Feature Class features and searches for duplicate features, **not including geometry.** That is, this tool searches for features within each feature class that have the same attributes across all fields in the attribute table, not including the fields listed below in the 'Note' section.

Note

> By default, this tool does not consider compare attributes in across any fields that are 'OID', 'Guid', 'GlobalID', 'Blob', or 'Raster' field types. Furthmore, the following fields are ignored in searching for duplicate features, by default (not case sensitive): 'LAST_EDITED_DATE', 'LAST_EDITED_USER', 'CREATED_USER', 'CREATED_DATE'. As stated previously, this tool does not include geometries in the feature comparisons by default.

7.2 Parameters

The tool has 3 parameters:

1. **Input_Geodatabase (data type: Workspace)** - This parameter must be the path of the input geodatabase to search Feature Datasets' Feature Class features for duplicate features.
2. **XY_Tolerance (data type: String)** - The XY_Tolerance parameter will be applied to each vertex when evaluating if there is an identical vertex in another entity, and must be input in the same units as the the source geodatabase's coordinate reference system (CRS).

3. **Z_Tolerance (data type: String)** - The Z_Tolerance parameter will be applied to each vertex when evaluating if there is an identical vertex in another entity with regard to elevation, and must be input in the same units as the the source geodatabase's coordinate reference system (CRS).

7.3 How to Use

7.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Delete Duplicate Features' script tool to open (Fig. 7.1).

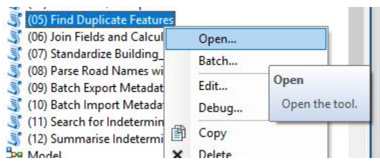


Figure 7.1: Opening the Delete Duplicate Features tool

7.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to search all Feature Classes within Feature Datasets in the Example.gdb for duplicate features (Fig. 7.2). We specify that we want to keep the default XY Tolerance and Z Tolerance parameters to zero, though this could be increased to allow duplicate geometry checks to be more lenient.

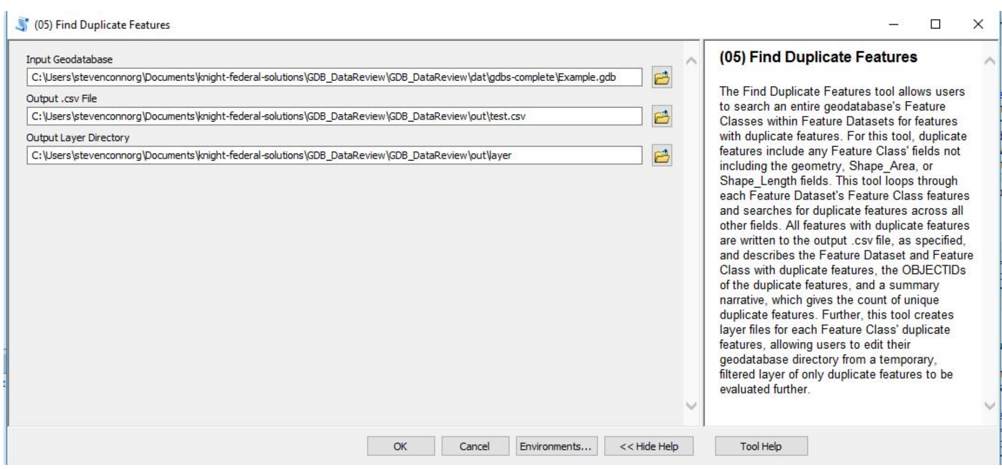


Figure 7.2: Delete Duplicate Features parameters

7.3.3 Run the Tool and View Results

While the tool runs (with Background Processing disabled), we can see the messages from the tool, which displays how many duplicate features will be deleted across each feature class, if applicable (Fig. 7.3). Here, we see that 102 duplicates were found in the MilFlightTrack_L feature class across 51 unique features, indicating that each of the 51 features may have been duplicated once (51 x 2 = 102).

Figure 7.3: Delete Duplicate Features messages

Chapter 8

Check and/or Repair Geometries

8.1 Overview

The Check and/or Repair Geometries tool allows users to search an entire geodatabase's Feature Classes for geometry problems. This tool loops through each Feature Dataset's Feature Class features and searches for geometry problems, including null geometry, self intersections, duplicate vertexes, and more.

If geometry problems exists, an output table is created containing the following fields: CLASS, FEATURE_ID, and PROBLEM. The feature classes which contain geometry problems are then repaired.

After the repair is conducted, the subset of feature classes with repaired geometry problems are checked again for geometry problems to confirm their repair. Another output table is generated for the subset of feature classes. An empty output table confirms the geometry problems were correctly repaired.

8.2 Parameters

The tool has 2 parameters:

1. **Input_Geodatabase (data type: Workspace)** - This parameter must be the path of the input geodatabase to search Feature Datasets' Feature Class features for duplicate features.
2. **Output_Geodatabase (data type: Workspace)** - This parameter should be the path of the Installation Review Geodatabase to compile all CIP processing outputs in a single location.
3. **Repair Geometries? (data type: Boolean)** - Do you want to automatically repair geometries, where possible?
4. **Delete Null Geometries? (data type: Boolean)** - Do you want to automatically delete features that have null/empty geometries?

8.3 How to Use

8.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Check and/or Repair Geometry' script tool to open (Fig. 8.1).

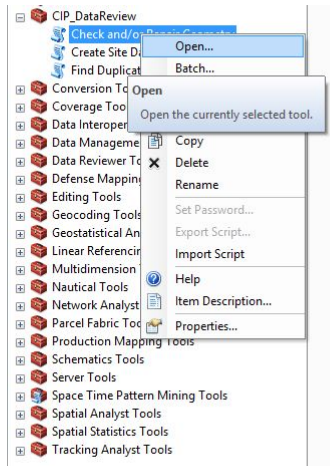


Figure 8.1: Opening the Check and/or Repair Geometries tool

8.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to search all Feature Classes within Feature Datasets in the Example.gdb for duplicate features (Fig. 8.2). Lastly, we specify where we want to output the resulting tables, preferably in an Installation Review geodatabase specifically for holding CIP processing results.

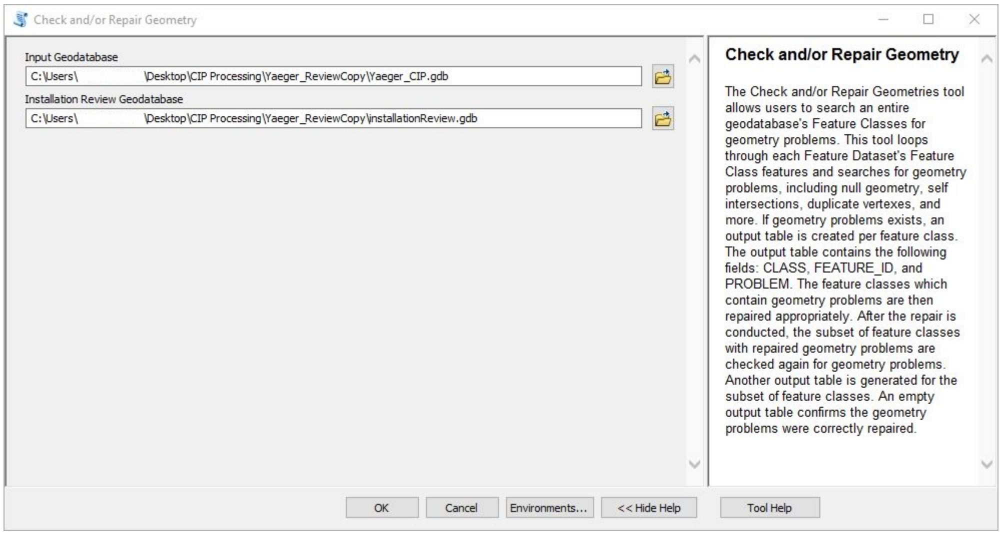


Figure 8.2: Check and/or Repair Geometries tool parameters

8.3.3 Run the Tool and View Results

While the tool runs (with Background Processing disabled), we can see the messages from the tool, which displays the following: how many feature classes are being checked for geometry problems, how many geometry problems that were found, where the output results are found, how many and which feature classes will be processed to repair geometry problems, how many feature classes are being re-checked for geometry errors, and how many geometry problems were found after the re-check (Fig. 8.3).

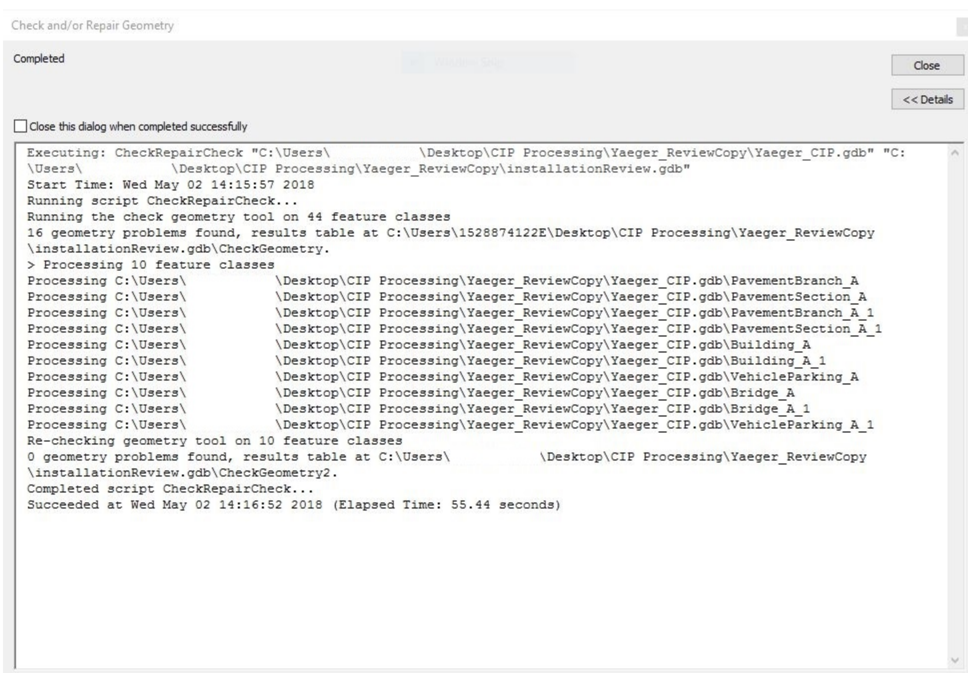


Figure 8.3: Output messages from the Check and/or Repair Geometries tool

Here, we see that 16 geometry problems were found in the Yaeger CIP geodatabase across 10 different feature classes, with each listed. A re-check was conducted on the 10 feature classes and then 0 geometry problems were found (Fig. 8.4).

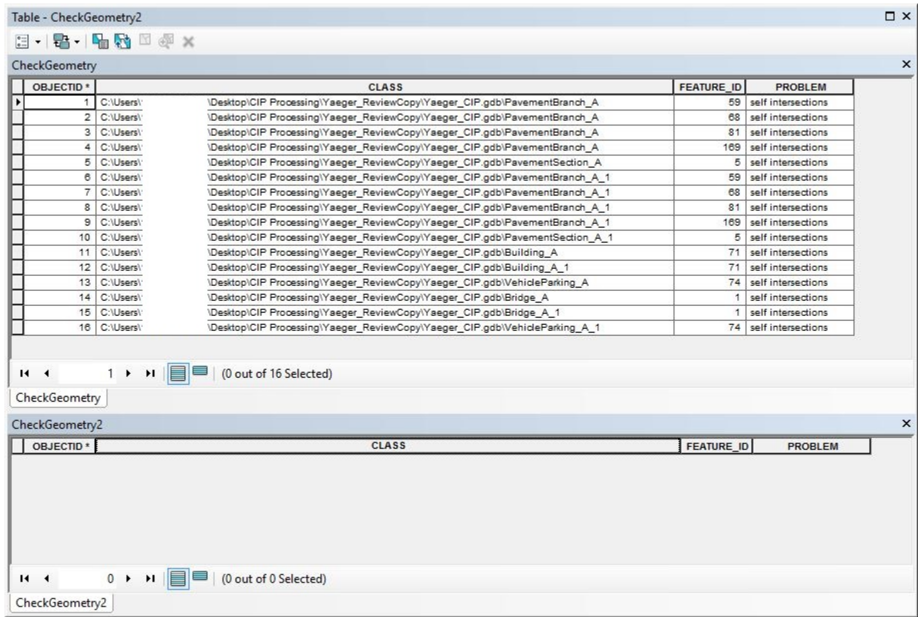


Table - CheckGeometry2

CheckGeometry

OBJECTID *	CLASS	FEATURE_ID	PROBLEM
1	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A	59	self intersections
2	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A	68	self intersections
3	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A	81	self intersections
4	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A	169	self intersections
5	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementSection_A	5	self intersections
6	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A_1	59	self intersections
7	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A_1	68	self intersections
8	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A_1	81	self intersections
9	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A_1	169	self intersections
10	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementSection_A_1	5	self intersections
11	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Building_A	71	self intersections
12	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Building_A_1	71	self intersections
13	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\VehicleParking_A	74	self intersections
14	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Bridge_A	1	self intersections
15	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Bridge_A_1	1	self intersections
16	C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\VehicleParking_A_1	74	self intersections

1 (0 out of 16 Selected)

CheckGeometry

CheckGeometry2

OBJECTID *	CLASS	FEATURE_ID	PROBLEM

0 (0 out of 0 Selected)

CheckGeometry2

Figure 8.4: Output table of duplicate geometries found for each Feature Class, with the re-run showing that all geometries have been fixed.

Chapter 9

Standardize 1 Address Field

9.1 Overview

The Standardize Address Field tool allows users to standardize 1 address field in a feature class. This tool works by searching the address field within the input feature class, then replaces any street prefixes (e.g.: North, north, East, West) with a standard prefix abbreviation (i.e.: "N", "S", "E", and "W"), while all suffixes (e.g.: AVE, Avenue, Street) are reformatted to standard USPS suffixes.

9.2 Parameters

The tool has 2 parameters:

1. **Feature Class (data type: Feature Class)** - The path to the Feature Class with the address field to standardize.
2. **Field (data type: Field)** - The address field in the Feature Class to be standardized.

9.3 How to Use

9.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Standardize 1 Address Field' script tool to open (Fig. 9.1).

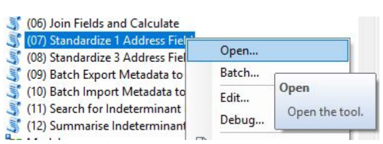


Figure 9.1: Opening the Standardize Address Field tool

9.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to update the buildingAddress field in the Building_A Feature Class in the Example.gdb (Fig. 9.2) because we notice none standardized addresses in the field (Fig. 9.3).

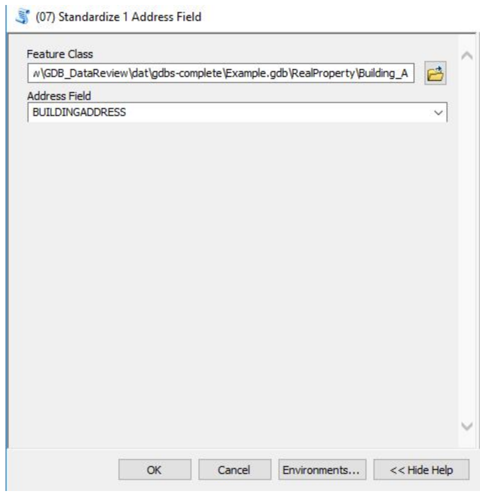
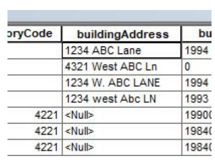


Figure 9.2: Tool parameters to run

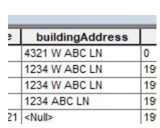


oryCode	buildingAddress	bu
	1234 ABC Lane	1994
	4321 West ABC Ln	0
	1234 W. ABC LANE	1994
	1234 west Abc LN	1993
4221	<Null>	1990(
4221	<Null>	1984(
4221	<Null>	1984(

Figure 9.3: The unstandardized address field attributes before running the tool.

9.3.3 Run the Tool and View Results

After the tool has run, we can see that the building address values have been appropriately standardized (Fig. 9.4).



e	buildingAddress	
	4321 W ABC LN	0
	1234 W ABC LN	19
	1234 W ABC LN	19
	1234 ABC LN	19
21	<Null>	19

Figure 9.4: The standardized address field attributes after running the tool.

Chapter 10

Standardized Road Prefix, Name, and Suffix

10.1 Overview

The purpose of this tool is to standardize the 3 field (road prefix, road name, and road suffix) values within a feature class. This tool works by first searching the ROADNAME field within that feature class, then removes any prefixes or suffixes within the field and moves them to the appropriate field. For all prefixes and suffixes found, the prefixes are reformatted to "N", "S", "E", and "W." For all suffixes found, the suffixes are reformatted to standard USPS suffixes.

10.2 Parameters

The tool has 4 parameters:

1. **Road Feature Class (data type: Feature Class) -** This parameter must be the path to the Feature Class that has the 3 road fields to be standardized.
2. **Prefix Field (data type: Field) -** The field within the feature class that has or should have road prefixes.
3. **Name Field (data type: Field) -** The field within the feature class that has road names.
4. **Suffix Field (data type: Field) -** The field within the feature class that has or should have road suffixes.

10.3 How to Use

10.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Standardize 3 Address Fields' script tool to open (Fig. 10.1).

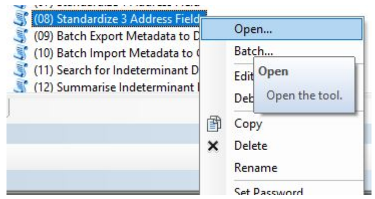


Figure 10.1: Opening Standardize Road Prefix, Name, and Suffix tool

10.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to update the road prefix, road name, and road suffix fields in the RoadCenterline_L feature class (Fig. 10.2). The fields can be derived directly from the Feature Class by using the drop-down menu.

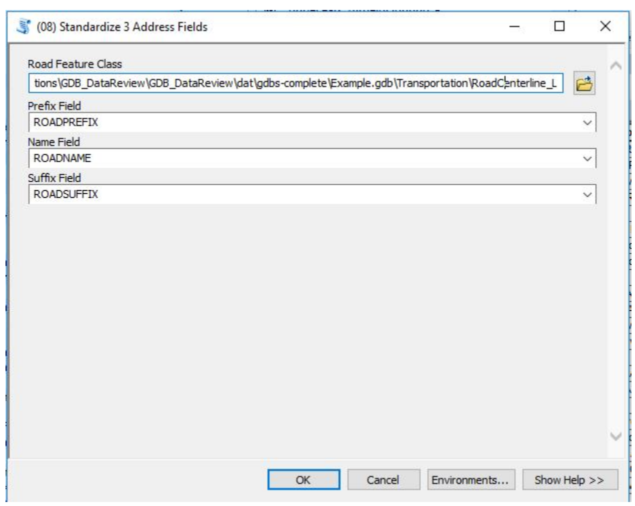


Figure 10.2: Standardize Road Prefix, Name, and Suffix tool parameters

10.4 Run the Tool and View Results

Before running the tool, we see that, indeed, the road prefixes and road suffixes are incorrectly populated inside the road name field Open the destinate Feature Class and view the update destination field values (Fig. 10.3).

roadPrefix	roadName	roadSuffix	
TBD	Sikorsky Rd	TBD	y
TBD	Perimeter Rd	TBD	y
TBD	Hansell Ave	TBD	y
TBD	Obstacle Rd	TBD	y
TBD	Cuba Rd	TBD	y
TBD	Richmond Rd	TBD	y
TBD	Graham Rd	TBD	y
TBD	Gate 35 Rd	TBD	y
TBD	Bong St	TBD	y
TBD	Vermont Ave	TBD	y
TBD	Colorado Ave	TBD	y
TBD	El Paso Ave	TBD	y
TBD	Twining Ave	TBD	y
TBD	Marsh Rd	TBD	y
TBD	Nebraska Ave	TBD	y
TBD	O'Malley Ave	TBD	y
TBD	Mitchell Dr	TBD	y
TBD	Seattle Ave	TBD	y
TBD	Graham Rd	TBD	n
TBD	Delaware Ave	TBD	y
TBD	Ft. Wright Oval	TBD	y

Figure 10.3: Unstandardized road prefixes, names, and suffixes before running the tool.

After running the tool, we can see that the prefixes and suffixes have been populated in the correct fields, and have also been standardized to match USPS standards (Fig. 10.4).

roadPrefix	roadName	roadSuffix
TBD	Wyoming	AVE
TBD	Wisconsin	AVE
TBD	Wilton	RD
TBD	Wildlife	RD
TBD	Wildlife	RD
TBD	Wildlife	RD
TBD	Westover	ST
TBD	Westover	ST
TBD	Westover	ST
TBD	Westover	ST
TBD	Westover	ST
TBD	Weston	RD
TBD	Washington	AVE
TBD	Washington	AVE
TBD	Walnut	ST
TBD	Wainwright	BLVD
TBD	Wainwright	BLVD
TBD	Wainwright	BLVD
TBD	Wainwright	BLVD
TBD	Wainwright	BLVD
TBD	Wainwright	BLVD
TBD	Virginia	AVE
TBD	Vet	RD
TBD	Vet	RD
TBD	Vet	RD

Figure 10.4: Standardized road prefixes, names, and suffixes after running the tool.

Chapter 11

Search for Missing and Indeterminant Data

11.1 Overview

Search a 'source' geodatabase for indeterminate data from feature dataset/feature class combinations in a target geodatabase. First, searches for missing feature datasets in target geodatabase not in source geodatabase. Then, searches for feature classes in 'x' feature dataset. Then, for each feature class in the source geodatabase, this tool searches for 'indeterminate' values in each field. Indeterminate values, here, means any null, to be determined (TBD), or 'other' values.

This tool creates 4 output tables, each prepended with the name of the Model_Geodatabase (e.g.: If your 'model' geodatabase called 'CIP', the tables will be called (CIP_MissingFDS, CIP_Missing_FCs, CIP_MissingFields, and CIP_MissingData). These tables include:

- [modelGeodatabaseName]_MissingFDS - Gives a list of Feature Datasets within the target geodatabase that are not included in the source geodatabase.
- [modelGeodatabaseName]_MissingFCs - Gives a list of Feature Classes for each Feature Dataset within the target geodatabase that are not included in the source geodatabase.
- [modelGeodatabaseName]_MissingFields - Gives a list of Fields for each Feature Dataset/Feature Class combination within the target geodatabase that are not included in the source geodatabase.
- [modelGeodatabaseName]_MissingData - For each Feature Dataset/Feature Class combination in both the target and source geodatabase, this table gives an overview of missing attributes for each field in the source geodatabase's Feature Class.
  - For Fields in each of the source geodatabase's Feature Classes, this table highlights fields not included in the target geodatabase's Feature Class under the 'FIELD_NONSDS' column (e.g.: 'FIELD_NONSDS' = F when fields are included in both geodatabases, and 'FIELD_NONSDS'

= T when the field exists in the source geodatabase for said Feature Class, but not the target geodatabase's Feature Class).
  - This table then lists whether or not the feature class is empty (i.e.: EMPTY_FC = T or F).
  - Then, for each field, the MissingData table gives a count of Null[1], 'TBD'[2], and 'Other'[3] features, further giving the counts of each value in 'NULL_VALUE_COUNTS', 'TBD_VALUE_COUNTS', and 'OTHER_VALUE_COUNTS' fields.
  - The sum of the Null, TBD, and Other features are populated in the 'TOTAL_INDT_COUNT' (i.e.: Total indeterminant feature count), with the 'TOTAL_DET_COUNT' column giving the total number of features with 'determinated' values (i.e.: not indeterminant values).
  - The POP_VALS column lists the count of all unique populated values for each field, while the INC_POP_VALS column lists any field values that are not included in the field's domain.

11.2 Parameters

The tool has 2 parameters:

1. **Source Geodatabase (data type: Workspace/File Geodatabase)** - The path to the file geodatabase to be searched for indeterminant/missing data.
2. **Target Geodatabase (data type: Workspace/File Geodatabase)** - The path to the file geodatabase with which the source geodatabase will be compared against.

Disclaimer!

This script tool currently requires an Advanced ArcGIS License!

11.3 How to Use

11.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Search for Indeterminant Data' script tool to open (Fig. 11.1).

[1] Null values include :None, “None”, “none”, “NONE”, “”,“-99999”,“77777”,77777, ” “,”NA“,”na“,”N/A“,”n/a“,”NULL“,”Null“,”“,”null“,”“”“,” “,” “,” “,” “

[2] TBD values include : “tbd”,“TBD”,“To be determined”,“Tbd”,99999,“99999”

[3] Other values include : “Other”, “other”, “OTHER”,“88888”,88888

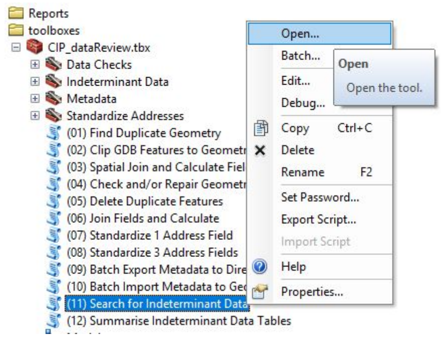


Figure 11.1: Opening the Search for Missing and Indeterminant Data tool

11.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to compare the 'Example.gdb' against the 'CIP.gdb' (Fig. 11.2). The fields can be derived directly from the Feature Class by using the drop-down menu.

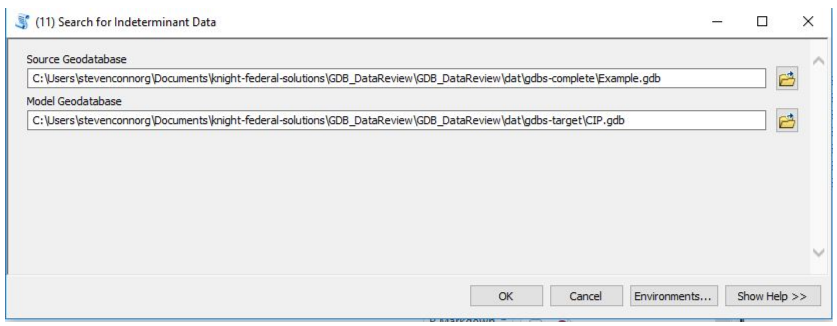


Figure 11.2: Opening the Search for Missing and Indeterminant Data tool

11.4 Run the Tool and View Results

While we run the tool, we can see view the messages of the tool, giving a listing of the fields being searched for indeterminant data with the counts of indeterminant values (Fig. 11.3).

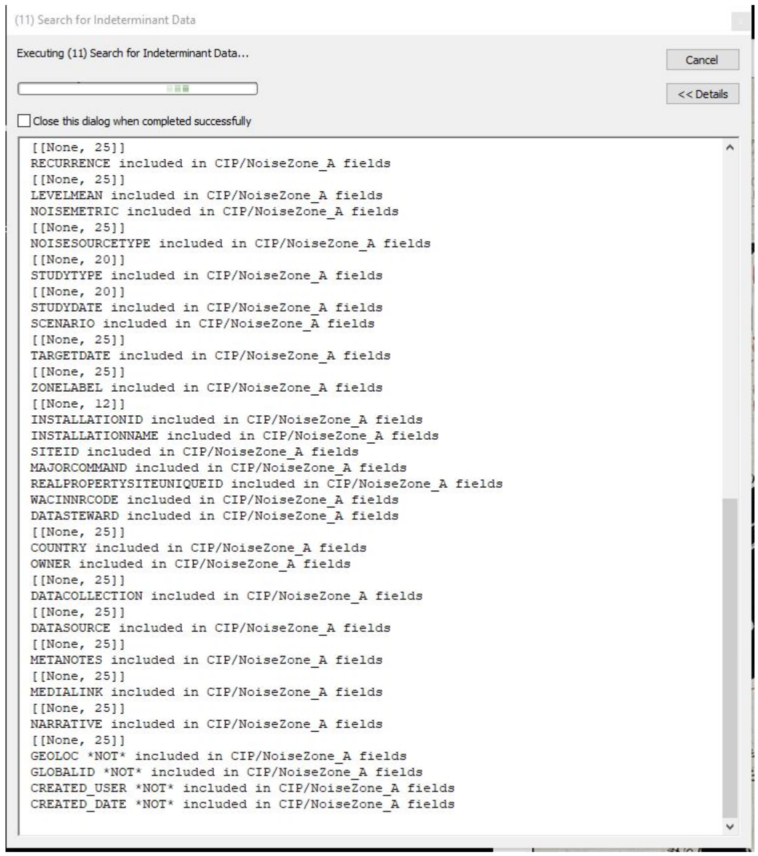


Figure 11.3: Search for Missing and Indeterminant Data tool messages

After the tool has run, we can inspect the output tables within the 'Example.gdb' geodatabase (Fig. 11.4). Opening the CIP_MissingFDS table, we see that the Example geodatabase have no missing Feature Datasets 11.5).

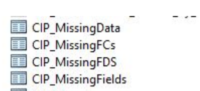


Figure 11.4: Search for Missing and Indeterminant Data output tables

Figure 11.5: The output Missing Feature Datasets table

Examining the MissingFCs table, we see that the Example geodatabase has one Feature Class, RoadSeg_L from the Transportation Feature Dataset, missing when compared with the CIP geodatabase 11.6).

Contents | Preview | Description

OBJECTID *	FC_MISSING	FDS	INSTALLATION
1	RoadSeg_L	Transportation	Example

Figure 11.6: The output Missing Feature Classes table

We can look at the MissingFLD table to see which fields are missing from each Feature Class from the target geodatabase that are included in the source geodatabase 11.7).

Contents | Preview | Description

OBJECTID *	FDS	FC	FIELD_MISSING	INSTALLATION
1	Auditory	NoiseZone_A	SHAPE	Example
2	Auditory	NoiseZone_A	CREATEDATE	Example
3	Auditory	NoiseZone_A	CREATOR	Example
4	Auditory	NoiseZone_A	EDITOR	Example
5	Auditory	NoiseZone_A	DATEEDITED	Example
6	Cadastre	Installation_A	SHAPE	Example
7	Cadastre	Installation_A	CREATEDATE	Example
8	Cadastre	Installation_A	CREATOR	Example
9	Cadastre	Installation_A	EDITOR	Example
10	Cadastre	Installation_A	DATEEDITED	Example
11	Cadastre	LandParcel_A	SHAPE	Example
12	Cadastre	LandParcel_A	CREATEDATE	Example
13	Cadastre	LandParcel_A	CREATOR	Example
14	Cadastre	LandParcel_A	EDITOR	Example
15	Cadastre	LandParcel_A	DATEEDITED	Example
16	Cadastre	Outgrant_A	SHAPE	Example
17	Cadastre	Outgrant_A	CREATEDATE	Example
18	Cadastre	Outgrant_A	CREATOR	Example
19	Cadastre	Outgrant_A	EDITOR	Example
20	Cadastre	Outgrant_A	DATEEDITED	Example
21	Cadastre	Site_A	SHAPE	Example
22	Cadastre	Site_A	CREATEDATE	Example
23	Cadastre	Site_A	CREATOR	Example
24	Cadastre	Site_A	EDITOR	Example
25	Cadastre	Site_A	DATEEDITED	Example
26	Cadastre	Site_P	SHAPE	Example
27	Cadastre	Site_P	CREATEDATE	Example
28	Cadastre	Site_P	CREATOR	Example
29	Cadastre	Site_P	EDITOR	Example
30	Cadastre	Site_P	DATEEDITED	Example
31	environmentalCulturalResources	HistoricDistrict_A	SHAPE	Example
32	environmentalCulturalResources	HistoricDistrict_A	CREATEDATE	Example
33	environmentalCulturalResources	HistoricDistrict_A	CREATOR	Example
34	environmentalCulturalResources	HistoricDistrict_A	EDITOR	Example
35	environmentalCulturalResources	HistoricDistrict_A	DATEEDITED	Example
36	environmentalNaturalResources	Wetland_A	SHAPE	Example
37	environmentalNaturalResources	Wetland_A	CREATEDATE	Example
38	environmentalNaturalResources	Wetland_A	CREATOR	Example
39	environmentalNaturalResources	Wetland_A	EDITOR	Example
40	environmentalNaturalResources	Wetland_A	DATEEDITED	Example
41	environmentalRestoration	EnvRemediationSite_A	SHAPE	Example
42	environmentalRestoration	EnvRemediationSite_A	CREATEDATE	Example
43	environmentalRestoration	EnvRemediationSite_A	CREATOR	Example
44	environmentalRestoration	EnvRemediationSite_A	EDITOR	Example
45	environmentalRestoration	EnvRemediationSite_A	DATEEDITED	Example
46	RealProperty	Building_A	SHAPE	Example
47	RealProperty	Building_A	REALPROPERTYUNIQUEID	Example
48	RealProperty	Building_A	CREATEDATE	Example
49	RealProperty	Building_A	CREATOR	Example

0 (of 162)

Figure 11.7: The output Missing Feature Fields table

To examine indeterminant field attribution, we can examine the MissingData table 11.8).

Contents | Preview | Description

OBJECTID *	INSTALLATI	FDS	FC	FIELD	FIELD_NONSDS	EMPTY_FC	NULL_FC_COUNT	TBD_FC_COUNT	OTHER_FC_COUNT	
1	Example	Auditory	NoiseZone_A	NOISEZONEID	F	F	0	0	0	
2	Example	Auditory	NoiseZone_A	SDSID	F	F	25	0	0	25 features are 'NULL'.
3	Example	Auditory	NoiseZone_A	SDSFEATURE	F	F	25	0	0	25 features are 'NULL'.
4	Example	Auditory	NoiseZone_A	SDSFEATURE	F	F	23	0	0	23 features are 'NULL'.
5	Example	Auditory	NoiseZone_A	SDSMETADA	F	F	25	0	0	25 features are 'NULL'.
6	Example	Auditory	NoiseZone_A	AREASIZE	F	F	25	0	0	25 features are 'NULL'.
7	Example	Auditory	NoiseZone_A	AREASIZEUO	F	F	25	0	0	25 features are 'NULL'.
8	Example	Auditory	NoiseZone_A	PERIMETERSI	F	F	25	0	0	25 features are 'NULL'.
9	Example	Auditory	NoiseZone_A	PERIMETERSI	F	F	25	0	0	25 features are 'NULL'.
10	Example	Auditory	NoiseZone_A	LATITUDE	F	F	25	0	0	25 features are 'NULL'.
11	Example	Auditory	NoiseZone_A	LONGITUDE	F	F	25	0	0	25 features are 'NULL'.
12	Example	Auditory	NoiseZone_A	MGRSCENTR	F	F	25	0	0	25 features are 'NULL'.
13	Example	Auditory	NoiseZone_A	STARTTIME	F	F	25	0	0	25 features are 'NULL'.
14	Example	Auditory	NoiseZone_A	ENDTIME	F	F	25	0	0	25 features are 'NULL'.
15	Example	Auditory	NoiseZone_A	DURATION	F	F	25	0	0	25 features are 'NULL'.
16	Example	Auditory	NoiseZone_A	RECURRENCE	F	F	25	0	0	25 features are 'NULL'.
17	Example	Auditory	NoiseZone_A	LEVELMEAN	F	F	0	0	0	
18	Example	Auditory	NoiseZone_A	NOISEMETRIC	F	F	25	0	0	25 features are 'NULL'.
19	Example	Auditory	NoiseZone_A	NOISESOURC	F	F	20	0	0	20 features are 'NULL'.
20	Example	Auditory	NoiseZone_A	STUDYTYPE	F	F	20	0	0	20 features are 'NULL'.
21	Example	Auditory	NoiseZone_A	STUDYDATE	F	F	0	0	0	
22	Example	Auditory	NoiseZone_A	SCENARIO	F	F	25	0	0	25 features are 'NULL'.
23	Example	Auditory	NoiseZone_A	TARGETDAT	F	F	25	0	0	25 features are 'NULL'.
24	Example	Auditory	NoiseZone_A	ZONELABEL	F	F	12	0	0	12 features are 'NULL'.
25	Example	Auditory	NoiseZone_A	INSTALLATIO	F	F	0	0	0	
26	Example	Auditory	NoiseZone_A	INSTALLATIO	F	F	0	0	0	
27	Example	Auditory	NoiseZone_A	SITEID	F	F	0	0	0	
28	Example	Auditory	NoiseZone_A	MAJORCOMM	F	F	0	0	0	
29	Example	Auditory	NoiseZone_A	REALPROPER	F	F	0	0	0	
30	Example	Auditory	NoiseZone_A	WACINNRCO	F	F	0	0	0	
31	Example	Auditory	NoiseZone_A	DATASTEWA	F	F	25	0	0	25 features are 'NULL'.
32	Example	Auditory	NoiseZone_A	COUNTRY	F	F	0	0	0	
33	Example	Auditory	NoiseZone_A	OWNER	F	F	25	0	0	25 features are 'NULL'.
34	Example	Auditory	NoiseZone_A	DATACOLLE	F	F	25	0	0	25 features are 'NULL'.
35	Example	Auditory	NoiseZone_A	DATASOURC	F	F	25	0	0	25 features are 'NULL'.
36	Example	Auditory	NoiseZone_A	METANOTES	F	F	25	0	0	25 features are 'NULL'.
37	Example	Auditory	NoiseZone_A	MEDIALINK	F	F	25	0	0	25 features are 'NULL'.
38	Example	Auditory	NoiseZone_A	NARRATIVE	F	F	25	0	0	25 features are 'NULL'.
39	Example	Auditory	NoiseZone_A	GEOLOC	T	F	0	0	0	
40	Example	Auditory	NoiseZone_A	GLOBALID	T	F	0	0	0	
41	Example	Auditory	NoiseZone_A	CREATED_US	T	F	0	0	0	
42	Example	Auditory	NoiseZone_A	CREATED_D	T	F	0	0	0	
43	Example	Auditory	NoiseZone_A	LAST_EDITED	T	F	0	0	0	
44	Example	Auditory	NoiseZone_A	LAST_EDITED	T	F	0	0	0	
45	Example	Cadastre	Installation_A	INSTALLATIO	F	F	0	0	0	
46	Example	Cadastre	Installation_A	SDSID	F	F	1	0	0	1 feature is 'NULL'.
47	Example	Cadastre	Installation_A	SDSFEATURE	F	F	0	0	0	
48	Example	Cadastre	Installation_A	SDSFEATURE	F	F	0	0	0	

1 (of 1310)

Figure 11.8: The output Missing Data table

Chapter 12

Summarise Indeterminant/Missing Data Tables

12.1 Overview

This tool takes the 4 tables created with the Search for Missing and Indeterminant Data tool and creates an outbook Excel Workbook which includes the following sheets:

1. **Summary_by_FC** - gives the counts and percentages of 'Other', 'Null', and 'TBD' cells by Feature Class, as well as the total counts and percentages of indeterminate (Other + Null + TBD) and determinate cells (not Other, Null, or TBD),
2. **Summary_by_Field** - gives the same statistics as the Summary_by_FC sheet, but broken down further by Feature Class Fields,
3. **Empty Feature Classes** - gives the standard Feature Classes in the comparison geodatabase not included in the input geodatabase(i.e.: Feature Classes included in comparison geodatabases)
4. **Indeterminate_Overview**, gives :
    - The total count of feature classes that are empty
    - The total number of standard feature classes that are empty
    - The source geodatabase installation name
    - The total number of missing feature classes
    - The total number of missing feature datasets
    - The total number of empty fields from empty feature classes
    - The total number of empty fields from non-empty feature classes.

12.2 Parameters

The inputs required for this tool to work are the 4 output tables created with the "Search for Indeterminate Data" script tool from one comparison geodatabase (**repeat:** ensure these are all from the same comparison geodatabase [i.e.: [comparison GDB] is the same across all four input tables]):

1. **comparisonGDBname_MissingFDS Table (data type: GDB Table)** - The path to the MissingFDS table created with the Search for Missing and Indeterminant Data tool for one 'target' geodatabase.
2. **comparisonGDBname_MissingFCs (data type: GDB Table)** - The path to the MissingFCs table created with the Search for Missing and Indeterminant Data tool for one 'target' geodatabase.
3. **comparisonGDBname_MissingFields (data type: GDB Table)** - The path to the MissingFields table created with the Search for Missing and Indeterminant Data tool for one 'target' geodatabase.
4. **comparisonGDBname_MissingData (data type: GDB Table)** - The path to the MissingData table created with the Search for Missing and Indeterminant Data tool for one 'target' geodatabase.
5. **Output Excel File' (data type: .xlsx file)** - The path to the output Excel Workbook to save the summary sheets to.

Disclaimer!

This script tool requires a few non-standard ArcGIS 10.x Python modules: numpy and pandas. To install these modules for use in ArcGIS, you can download and install the modules using the commands "pip install pandas" and "pip install numpy."

To do this, follow these instructions:

1. Press the windows key on your keyboard
2. Type "cmd" to open the command prompt window
3. Set your working directory to the ArcGIS Python directory with the 'pip' command (e.g.: if you are running ArcMap10.6, input: "C:/Python27/ArcGIS10.6/Scripts")
4. Type 'pip install numpy' and press enter, then type 'pip install pandas' and press enter. If all goes well, you will have these modules successfully installed for use in ArcGIS' Python distribution

12.3 How to Use

12.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Summarise Indeterminant Data Tables' script tool to open (Fig. 12.1).

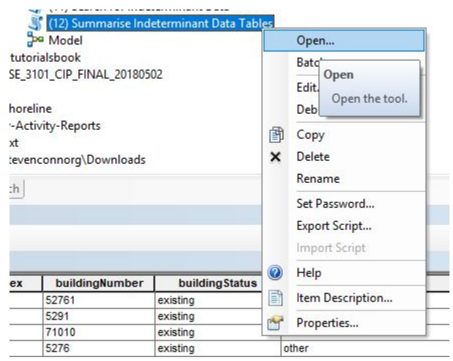


Figure 12.1: Opening the Summarise Indeterminant Data Tables tool

12.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to summarise the Indeterminant Data Tables created in the Example.gdb that was created when comparing against the CIP geodatabase. Again, be sure that these input tables all derive from the same target geodatabase!

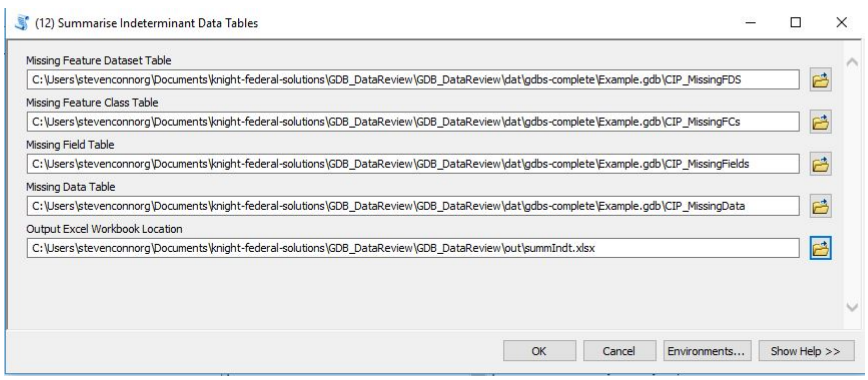


Figure 12.2: Setting the parameters for the Summarise Indeterminant Data Tables tool

12.4 Run the Tool and View Results

While we run the tool, we can see view the messages of the tool (Fig. 12.3).

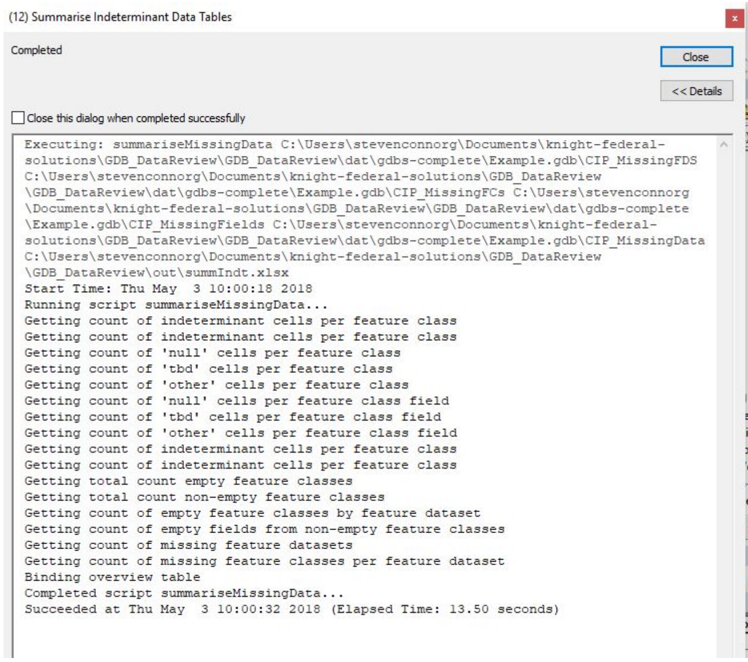


Figure 12.3: Opening the Delete Duplicate Features tool

After the tool has run, we can open the output Excel Workbook we specified to see the 4 output sheets : Summary_by_FC, Summary_by_Field, Empty Feature Classes, and Indeterminate_Overview (Fig. 12.4).

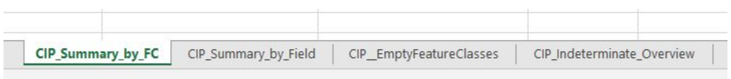


Figure 12.4: The output Excel Workbook sheets Summary by FC, Summary by Field, Empty Feature Classes, and Indeterminate Overview

Viewing the Summary_by_FC sheets gives us a comprehensive overview of the counts and percentages of indeterminant Attribute Table cells by indeterminant data type (i.e.: Null, TBD, and Other) (Fig. 12.5).

B	C	D	E	F	G	H	I	J	K	L
INSTALLATIOI	FDS	FC	OTHER_PCT	TBD_PCT	NULL_PCT	OTHER_CNT	TBD_CNT	NULL_CNT	DETERMINED_PCT	UNDETERMINED_P
Example	Auditory	NoiseZone_A	0	0	61.36363636	0	0	675	38.63636364	61.363
Example	Cadastre	Installation_A	0	9.302325581	20.93023256	0	4	9	69.76744186	30.232
Example	Cadastre	LandParcel_A	0.218000623	0	39.45811274	14	0	2534	60.32388664	39.676
Example	Cadastre	Outgrant_A	0	0	39.64646465	0	0	157	60.35353535	39.646
Example	Cadastre	Site_A	0	0	30.55555556	0	0	121	69.44444444	30.555
Example	Cadastre	Site_P	0	0	37.14285714	0	0	143	62.85714286	37.142
Example	MilitaryRangeTraining	ImpactArea_A				0	0	0		
Example	MilitaryRangeTraining	MilQuantityDistCombinedArc_A				0	0	0		
Example	MilitaryRangeTraining	MilRange_A	0	0	47.43589744	0	0	222	52.56410256	47.435
Example	MilitaryRangeTraining	MilTrainingLoc_A	0	1.203208556	42.9144385	0	9	321	55.88235294	44.117
Example	Pavements	PavementBranch_A	2.050919378	0	51.71751869	406	0	10238	46.23156193	53.768
Example	Pavements	PavementSection_A	0.042225168	0.003927923	60.80620612	43	4	61922	39.14764079	60.852
Example	Planning	AirAccidentZone_A	0	11.11111111	44.44444444	0	32	128	44.44444444	55.555
Example	Planning	LandUse_A	0.091168091	0	56.17094017	8	0	4929	43.73789174	56.262
Example	RealProperty	Building_A	4.396282102	5.050438307	23.94828422	1996	2293	10873	66.60499537	33.395
Example	RealProperty	Tower_P	0	23.80952381	33.33333333	0	10	14	42.85714286	57.142
Example	Recreation	GolfCourse_A				0	0	0		
Example	Recreation	RecArea_A	2.203856749	3.03030303	38.29201102	8	11	139	56.4738292	43.52
Example	Security	AccessControl_L	0	5.870674986	33.39478162	0	414	2355	60.73454339	39.265
Example	Security	AccessControl_P	0	6.936316695	30.36144578	0	403	1764	62.70223752	37.297
Example	Security	Fence_L	0	3.417366947	51.75070028	0	1464	22170	44.83193277	55.168
Example	Transportation	Bridge_A	2	26	44	2	26	44	28	
Example	Transportation	Bridge_L				0	0	0		
Example	Transportation	RailSegment_L	0	16.32653061	55.10204082	0	104	351	28.57142857	71.428
Example	Transportation	RailTrack_L	0	2.076606521	56.97689142	0	656	17999	40.94650206	59.053
Example	Transportation	RoadCenterline_L	0	10.1017112	46.9259724	0	3784	17578	42.9723164	57.02
Example	Transportation	RoadPath_L				0	0	0		
Example	Transportation	VehicleParking_A	0.008324546	7.045341029	42.87973805	3	2539	15453	50.06659637	49.933
Example	WaterWays	DocksAndWharfs_A	0	5.405405405	45.94594595	0	8	68	48.64864865	51.351
Example	environmentalCulturalResources	HistoricDistrict_A				0	0	0		
Example	environmentalNaturalResources	Wetland_A	0	8.676067372	52.31100666	0	443	2671	39.01292597	60.987
Example	environmentalRestoration	EnvRemediationSite_A	0	0	60	0	0	2040	40	

Figure 12.5: The output Summary by Feature Class table

The Summary_by_Field provides provides a breaks down of the Summary_by_FC table by field (Fig. 12.6).

B	C	D	E	F	G	H	I	J	K	L	M	N
INSTALLATIOI	FDS	FC	FIELD	OTHER_PC	TBD_PC	NULL_PC	OTHER_CN	TBD_CN	NULL_CN	DETERMINED_PC	UNDETERMINED_PC	DETERMINED_CN
Example	Auditory	NoiseZone_A	AREASIZE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	AREASIZEUOM	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	DATACOLLECTION	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	DATASOURCE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	DATASTEWARD	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	DURATION	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	ENDTIME	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	LATITUDE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	LONGITUDE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	MEDIALINK	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	METANOTES	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	MGRSCENTROID	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	NARRATIVE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	NOISEMETRIC	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	NOISESOURCETYPE	0	0	80	0	0	20	20	80	5
Example	Auditory	NoiseZone_A	OWNER	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	PERIMETERSIZE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	PERIMETERSIZEUOM	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	RECURRENCE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	SCENARIO	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	SDSFEATUREDESCRIPTION	0	0	92	0	0	23	8	92	2
Example	Auditory	NoiseZone_A	SDSFEATURENAME	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	SDSID	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	SDSMETADATAID	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	STARTTIME	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	STUDYTYPE	0	0	80	0	0	20	20	80	5
Example	Auditory	NoiseZone_A	TARGETDATE	0	0	100	0	0	25	0	100	0
Example	Auditory	NoiseZone_A	ZONELABEL	0	0	48	0	0	12	52	48	13

Figure 12.6: The output Summary by Field table

The Empty Feature Classes sheet provides a listing of the Feature Classes included in the Example.gdb that are empty, as well as the empty fields from those empty feature classes (Fig. 12.7).

FDS	FC	INSTALLATION	TOTAL_EMPTY_FIELD
MilitaryRangeTraining	ImpactArea_A	Example	35
MilitaryRangeTraining	MilQuantityDistCombinedArc_A	Example	36
Recreation	GolfCourse_A	Example	35
Transportation	Bridge_L	Example	52
Transportation	RoadPath_L	Example	36
environmentalCulturalResources	HistoricDistrict_A	Example	39

Figure 12.7: The output Empty Feature Classes table

Lastly, the Indeterminate_Overview sheet provides a general overview of missing and indeterminant data at a geodatabase level (Fig. 12.8).

InclFeatsEmpty	InclFeatsNonEmpty	Installation	MissingFCcount	MissingFDScount	TotalEmptyFields	TotalEmptyFieldsfromEmptyFC
6	26	Example	1	0	0	233

Figure 12.8: The output Indeterminate Overview table

Chapter 13

Batch Export Metadata

13.1 Overview

This tool provides an automated method to export metadata for each Feature Dataset and Feature Class (within Feature Datasets) in the input geodatabase, by exporting each item's metadata to an .xml file to an output directory, as specified. This tool allows you to specify a metadata translator. Within the scope of the Department of Defense Instruction (DoDI) 8130.01, it is recommended that you download and install the SDSFIE-M Metadata Style for ArcGIS from The Spatial Data Standards for Facilities, Infrastructure, and Environment (SDSFIE) Metadata standard. In this case, you would change the input the metadata translator to the ARCGIS2SDSFIE-M.xml[1] provided with the SDSFIE-M Metadata Style for ArcGIS software.^[Be sure to install the software to the software path for your ArcGIS distribution currently installed to view this metadata style within ArcCatalog, for example: "C:/Program Files (x86)/ArcGIS/Desktop10.X/"

If the source metadata is a Feature Dataset, the output .xml file is named after the Feature Dataset. Alternatively, the output .xml metadata for Feature Classes are exported with the Feature Dataset name prepended before the Feature Class name.

These output .xml files can more easily be edited in batch using the Batch Metadata Modifier Tool developed out of the University of Idaho's Interactive Numeric & Spatial Information Data Engine (INSIDE) geospatial data clearinghouse.

13.2 Parameters

The tool has 3 parameters:

1. **Input Geodatabase (data type: Workspace/File Geodatabase) -** The input geodatabase to export Feature Dataset/Feature Class metadata from.

[1] Under standard Windows installations, this should be located at "C:/Program Files (x86)/ArcGIS/Desktop10.6/Metadata/Translator/ARCGIS2SDSFIE-M.xml".

2. **Metadata Translator (data type: .XML file) -** The metadata translator to be used to create output .xml metadata files. These files are typically located at "C:/Program Files (x86)/ArcGIS/[Desktop10.x]/Metadata/Translator/. Please change according to your ArcGIS installation file path.
3. **Output Directory (data type: Folder) -** The folder within which to write output .xml metadata files.

13.3 How to Use

13.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Batch Export Metadata to Directory' script tool to open (Fig. 13.1).

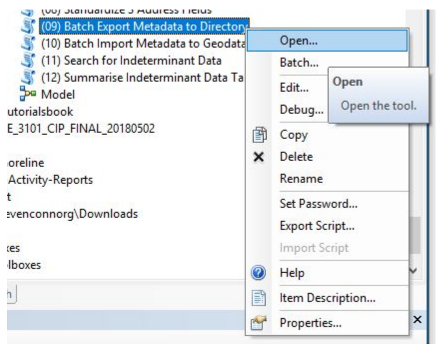


Figure 13.1: Opening the Batch Export Metadata tool

13.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to export metadata for all Feature Datasets and all Feature Classes (within those Feature Datasets) within the Example.gdb geodatabase using the default ARCGIS2FGDC metadata translator that comes with ArcGIS to a new directory called 'metadata' (Fig. 13.2).

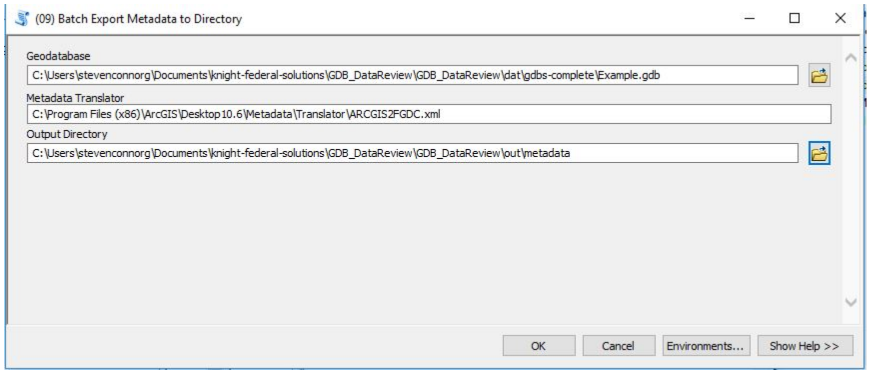


Figure 13.2: Batch Export Metadata tool parameters

13.4 Run the Tool and View Results

After running the tool, we can view the output metadata files inside the output directory specified (Fig. 13.3). For Feature Classes, the output .xml files has the associated Feature Dataset name prepended to the filename, while Feature Dataset metadata file is simply the name of the Feature Dataset.

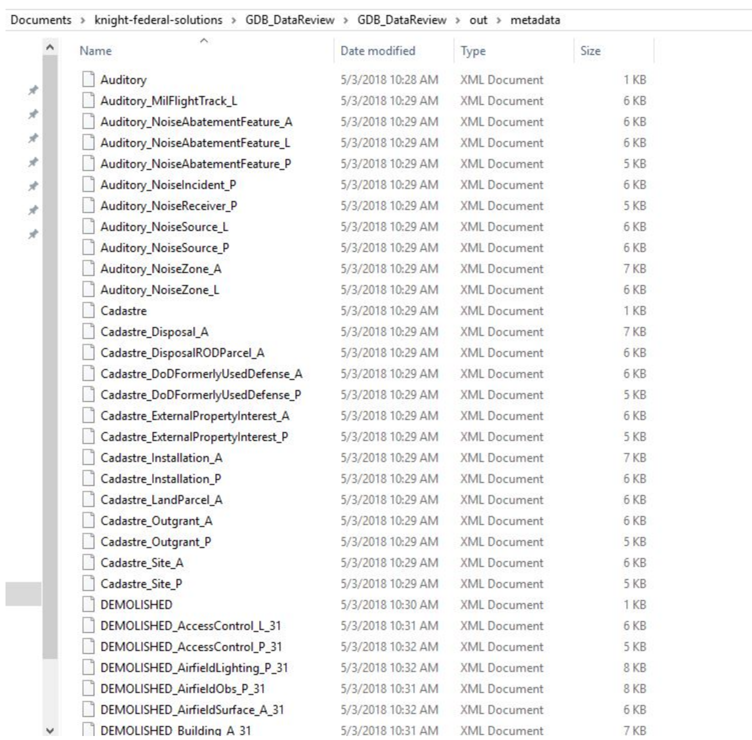


Figure 13.3: Batch Export Metadata output .xml files for each Feature Dataset/Feature Class

Chapter 14

Batch Import Metadata

14.1 Overview

This tool provides an automated method to import metadata for each Feature Dataset and Feature Class (within Feature Datasets) in the input geodatabase, following the output .xml file naming convention created with the Batch Export Metadata tool, and potentially updated using the Batch Metadata Modifier Tool developed out of the University of Idaho's Interactive Numeric & Spatial Information Data Engine (INSIDE) geospatial data clearinghouse. As per SDSFIE standards, it is recommended that you view this metadata using the SDSFIE-M Metadata Style for ArcGIS[1].

Alternatively, if you wish to import metadata from Feature Classes in a Esri geodatabase, you should use the Batch Import Metadata from GDB tool.

14.2 Parameters

The tool has 2 parameters[2]:

1. **Input Geodatabase (data type: Workspace/File Geodatabase) -** The input geodatabase to export Feature Dataset/Feature Class metadata from.
2. **Input Metadata Directory (data type: Folder) -** The folder with the .xml files to import into the geodatabase features.

[1] When installing this tool, be sure to install it to the location of your current ArcGIS distribution on your computer, typically located at C:/Program Files (x86)/ArcGIS/Desktop10.6/, replacing with the approproate ArcGIS version

[2] More information may be found on the help page for ArcMap's Metadata Importer.

14.3 How to Use

14.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the ‘Batch Import Metadata’ script tool to open (Fig. 14.1).

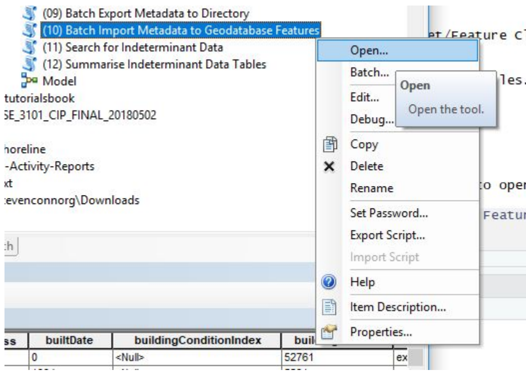


Figure 14.1: Opening the Batch Import Metadata tool

14.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to export metadata for all Feature Datasets and all Feature Classes (within those Feature Datasets) within the Example.gdb geodatabase using the default ARCGIS2FGDC metadata translator that comes with ArcGIS to a new directory called ‘metadata’ (Fig. 14.2).

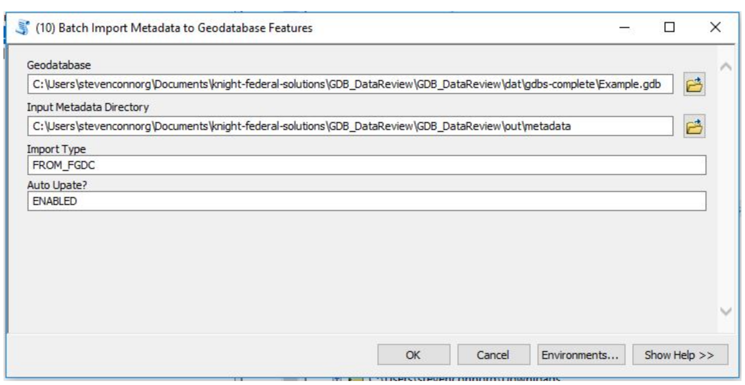


Figure 14.2: Batch Import Metadata tool parameters

14.4 Run the Tool and View Results

While the tool runs, we can see which .xml files are being imported to each Feature Dataset’s/Feature Class’ metadata, as well as which Feature Datasets/Feature Classes do not have matching .xml files in the output directory (Fig. 14.3).

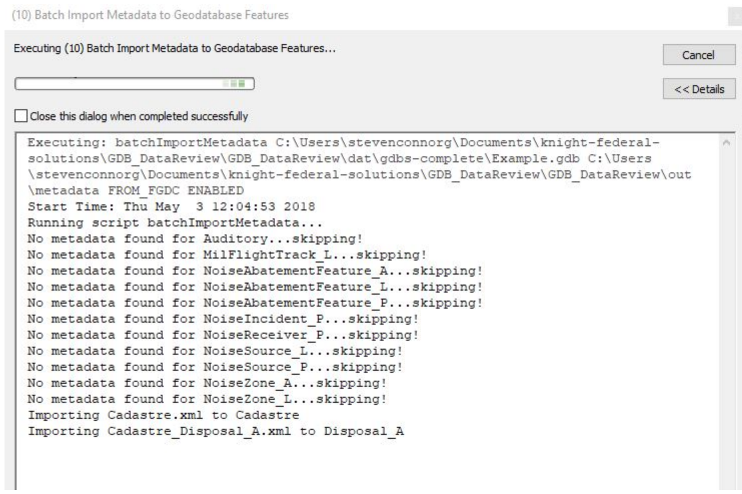


Figure 14.3: Batch Import Metadata tool messages

After running the tool, we can view the update Item Descriptions for the Feature Classes and Feature Datasets imported with the .xml files (Fig. 14.4). You can also change the way ArcMap displays the metadata by going to Customize > ArcMap Options in ArcMap, then clicking the Metadata tab and changing the Metadata Style in the drop-down menu. You can find more information on Item Descriptions on Esri's Item Desecription Help Page.

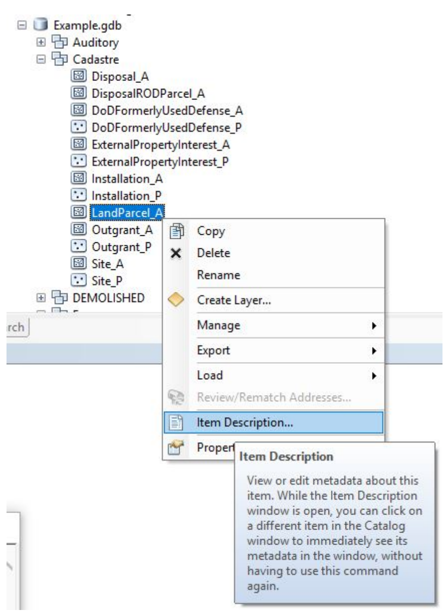


Figure 14.4: Opening the Item Description to view updated Metadata

Chapter 15

Batch Import Metadata from GDB

15.1 Overview

This tool provides an automated method to import metadata for each Feature Dataset and Feature Class (within Feature Datasets) in the input geodatabase from a source geodatabase. This tool different from the Batch Import Metadata tool because it imports metadata from a source geodatabase and not from a directory of metadata .xml files.

15.2 Parameters

The tool has 3 parameters[1]:

1. **Source Geodatabase (data type: Workspace/File Geodatabase) -** The input geodatabase to export Feature Dataset/Feature Class metadata from.
2. **Target Geodatabase (data type: Workspace/File Geodatabase) -** The geodatabase to input metadata from the Source Geodatabase from.
3. **Sync Type (data type: String) -** How would you like the metadata items to be Syncronized upon import?[2]
   - *ALWAYS* — Properties of the source item are always added to or updated in its metadata. Metadata will be created if it doesn't already exist. This is the deault.
   - *ACCESSED* — Properties of the source item are added to or updated in its metadata when it is accessed. Metadata will be created if it doesn't already exist.
   - *CREATED* — Metadata will be created and properties of the source item will be added to it if the item doesn't already have metadata.
   - *NOT_CREATED* — Properties of the source item are added to or updated in existing metadata.

[1]More information of the Import Type and Auto Update parameters may be found at the help page for ArcMap's Import Metadata Tool.

[2]More information on Sync Type paramters may be found at the help file for the Synchronize Metadata function

- *OVERWRITE* — The same as "ALWAYS" except all information that can be recorded automatically in the metadata will be recorded. Any properties typed in by a person will be replaced with the item's actual properties.
- *SELECTIVE* — The same as "OVERWRITE" except the title and the content type will not be overwritten with default values for the item. Used when metadata is upgraded to the ArcGIS 10.x metadata format.

15.3 How to Use

15.3.1 Begin by opening the toolbox

Navigate to the location of the script toolbox, then right-click the 'Batch Import Metadata to Geodatabase from GDB' script tool to open (Fig. 14.1).

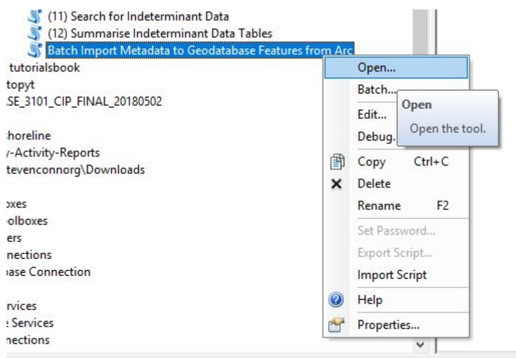


Figure 15.1: Opening the Batch Import Metadata tool

15.3.2 Fill out the parameters

Next, fill out the parameters for the tool. Here, we want to import metadata from the geodatabas with metadata to the geodatabase without metadata.

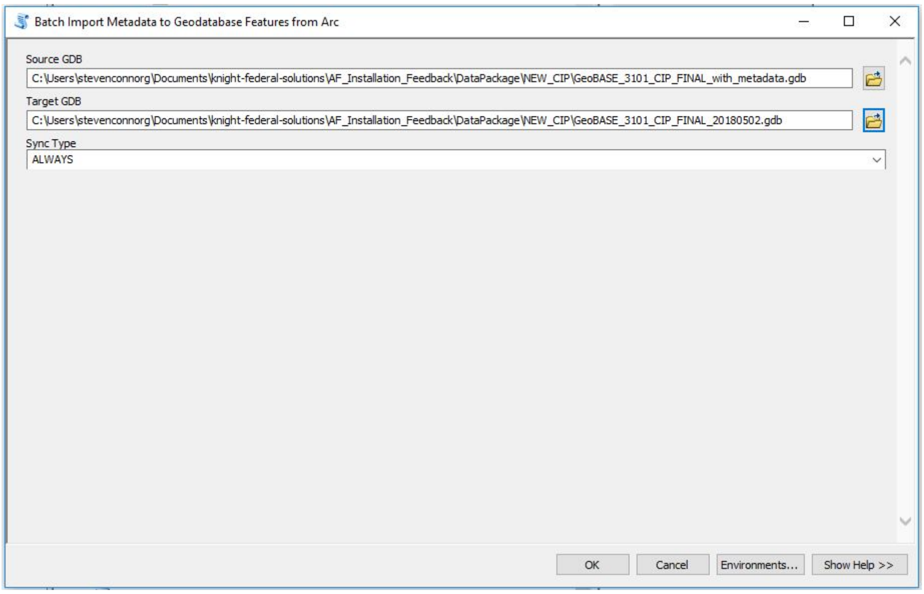


Figure 15.2: Batch Import Metadata tool parameters

15.4 Run the Tool and View Results

While the tool runs, we can see which geodatabase Feature Datasets/Feature Classes are being imported to each Feature Dataset's/Feature Class' metadata, as well as which Feature Datasets/Feature Classes do not have matching .xml files in the source geodatabase (Fig. 15.3).

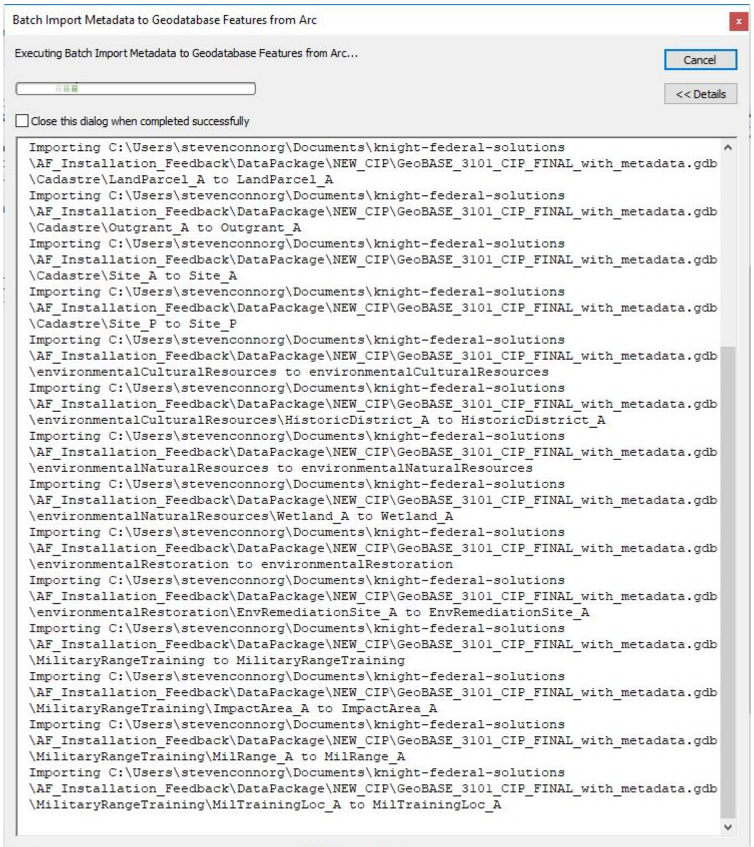


Figure 15.3: Batch Import Metadata tool messages

After running the tool, we can view the update Item Descriptions for the Feature Classes and Feature Datasets imported from the source geodatabase (Fig. 15.4). If importing SDSFIE-M metadata, it is recommended that you alter the way ArcMap displays the metadata by going to Customize > ArcMap Options in ArcMap, then clicking the Metadata tab and changing the Metadata Style in the drop-down menu. To do this, you must view this metadata using the SDSFIE-M Metadata Style for ArcGIS[3]. After doing this, we can view the SDSFIE-M metadata style within ArcCatalog.

You can find more information on Item Descriptions on Esri's Item Desecription Help Page.

[3]When installing this tool, be sure to install it to the location of your current ArcGIS distribution on your computer, typically located at C:/Program Files (x86)/ArcGIS/Desktop10.6/, replacing with the approproate ArcGIS version

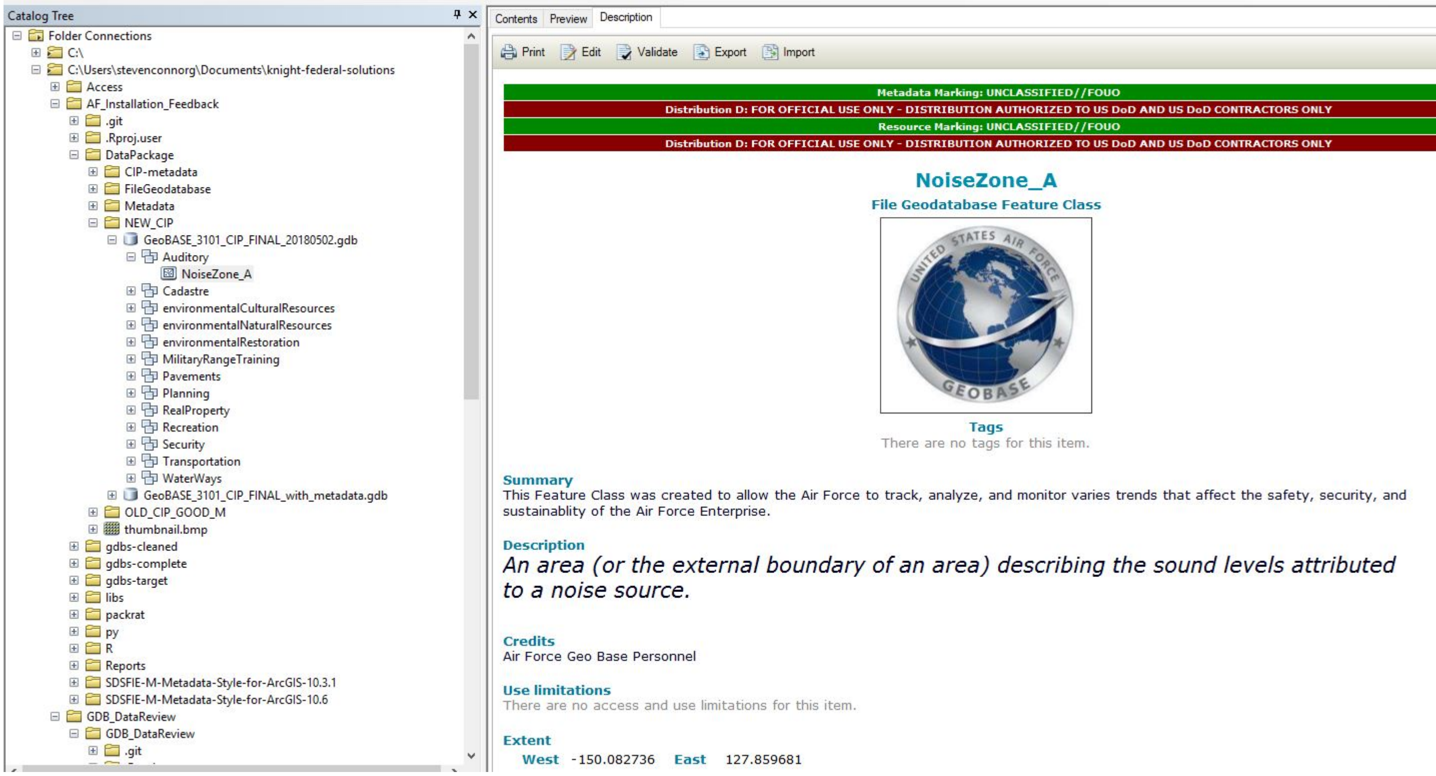


Figure 15.4: Opening the Item Description to view updated Metadata

Chapter 16

Production Information

This report was created using R (R Core Team, 2017) with the following packages: bookdown (Xie, 2018a), knitr (Xie, 2018b), rmarkdown (Allaire et al., 2017), and tinytex (Xie, 2018c). For posterity, package information is listed below:

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 16299)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] tinytex_0.5    knitr_1.20     rmarkdown_1.8  bookdown_0.7.8
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.16    digest_0.6.15   rprojroot_1.3-2 backports_1.1.2
##  [5] magrittr_1.5    evaluate_0.10.1 stringi_1.1.7   rstudioapi_0.7
##  [9] tools_3.4.1     stringr_1.3.0   jpeg_0.1-8      xfun_0.1
## [13] yaml_2.1.18     compiler_3.4.1  htmltools_0.3.6
```

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2017). *rmarkdown: Dynamic Documents for R.* R package version 1.8.

R Core Team (2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2018a). *bookdown: Authoring Books and Technical Documents with R Markdown.* R package version 0.7.8.

Xie, Y. (2018b). *knitr: A General-Purpose Package for Dynamic Report Generation in R.* R package version 1.20.

Xie, Y. (2018c). *tinytex: Helper Functions to Install and Maintain 'TeX Live', and Compile 'LaTeX' Documents.* R package version 0.5.