GDB_DataReview ArcMap Toolbox Tutorials

Air Force Civil Engineering Center (AFCEC) Geospatial Integration Office (GIO)

2018-05-04

Contents

Chapter 1

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 the current Air Force Data Model (GeoBase 3.1.0.1) developed under the AF GeoBase mission. This 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), while also providing some 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

Create Site Data

2.1 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.

2.2 Parameters

The tool has 3 parameters:

1. **Input_Feature_Dataset (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)** - This parameter is "HAVE THEIR CENTER IN" by default and provides the best results for straight forward situations. Other options are provided for other situations.

2.3 How to Use

2.3.1 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. 2.1).

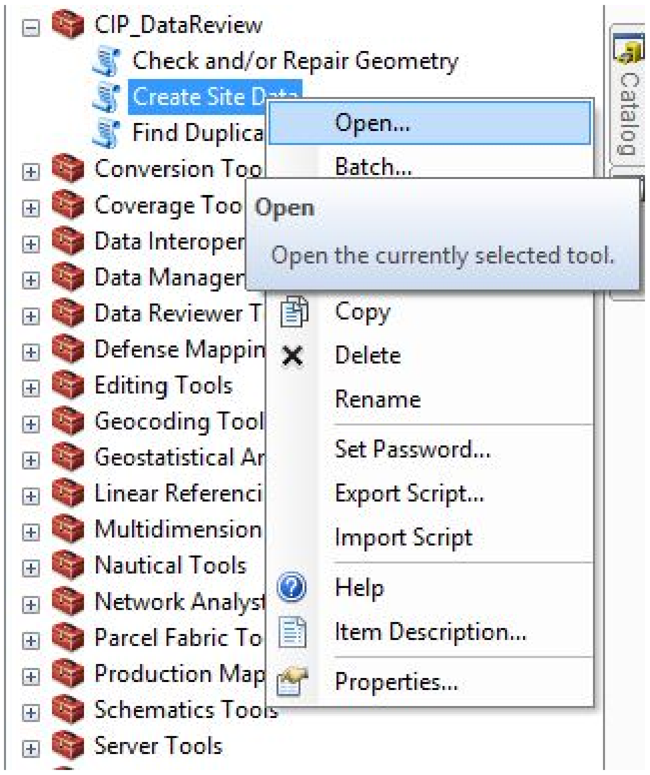
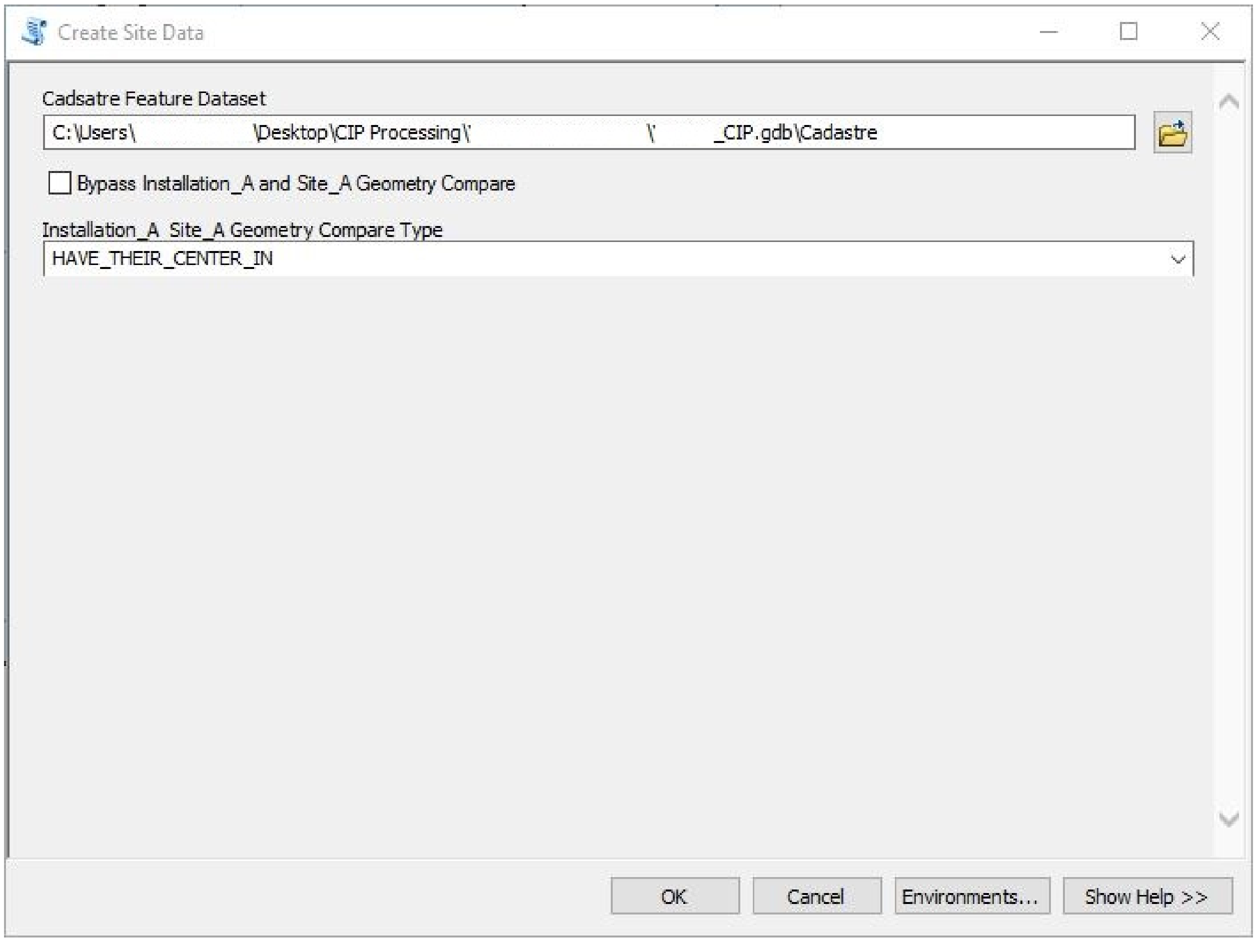


Figure 2.1: Opening the Create Site Data tool

2.3.2 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. 2.2). Last, we specify where we want to output the resulting tables, preferably in an Installation Review geodatabase specifically for holding CIP processing results.



Cadsatre Feature Dataset
C:\Users\ \Desktop\CIP Processing\' \' _CIP.gdb\Cadastre

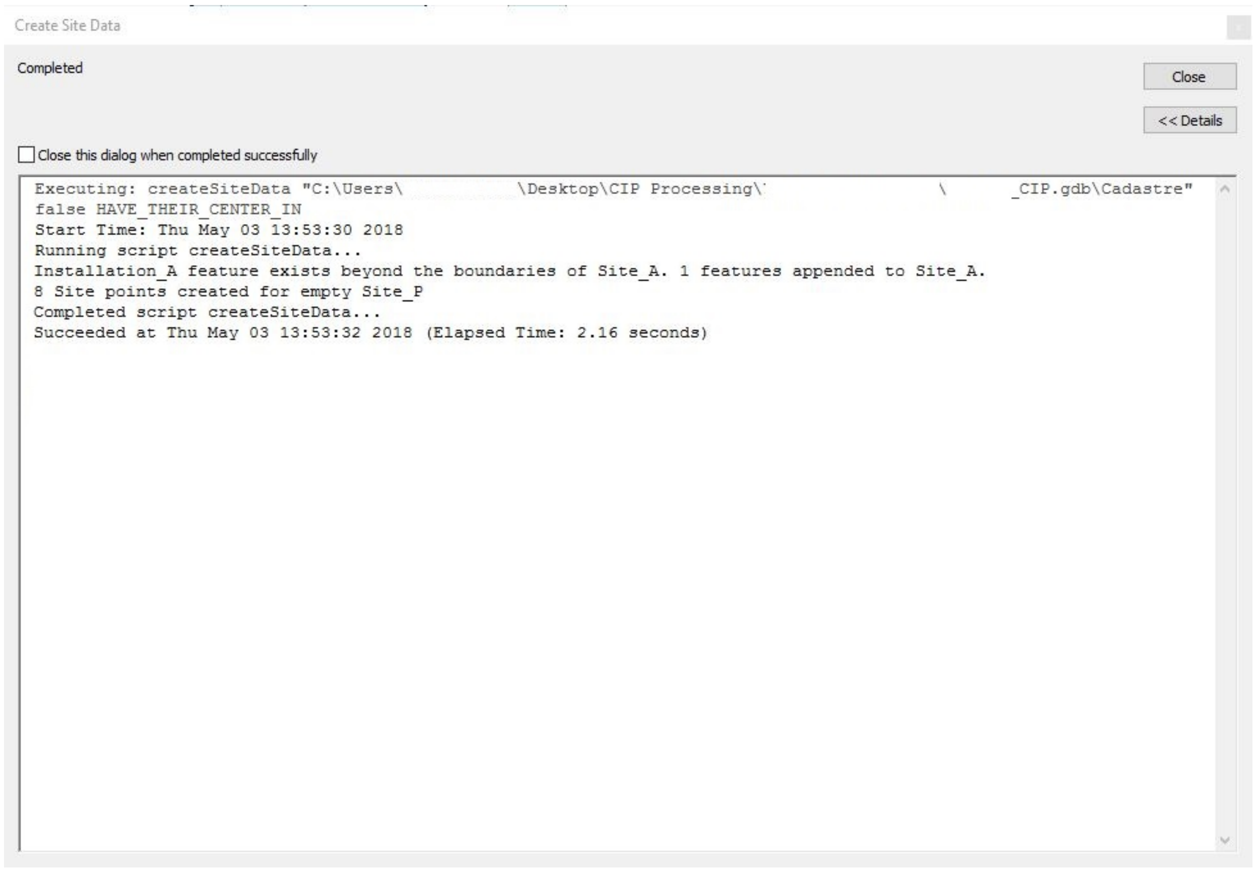
☐ Bypass Installation_A and Site_A Geometry Compare

Installation_A Site_A Geometry Compare Type
HAVE_THEIR_CENTER_IN

Figure 2.2: Create Site Data Tool parameters

2.3.3 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. 6.3). 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.



```
Executing: createSiteData "C:\Users\          \Desktop\CIP Processing\`          \      _CIP.gdb\Cadastre"
false HAVE_THEIR_CENTER_IN
Start Time: Thu May 03 13:53:30 2018
Running script createSiteData...
Installation_A feature exists beyond the boundaries of Site_A. 1 features appended to Site_A.
8 Site points created for empty Site_P
Completed script createSiteData...
Succeeded at Thu May 03 13:53:32 2018 (Elapsed Time: 2.16 seconds)
```

Figure 2.3: 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. 2.4 & Fig. 2.5).

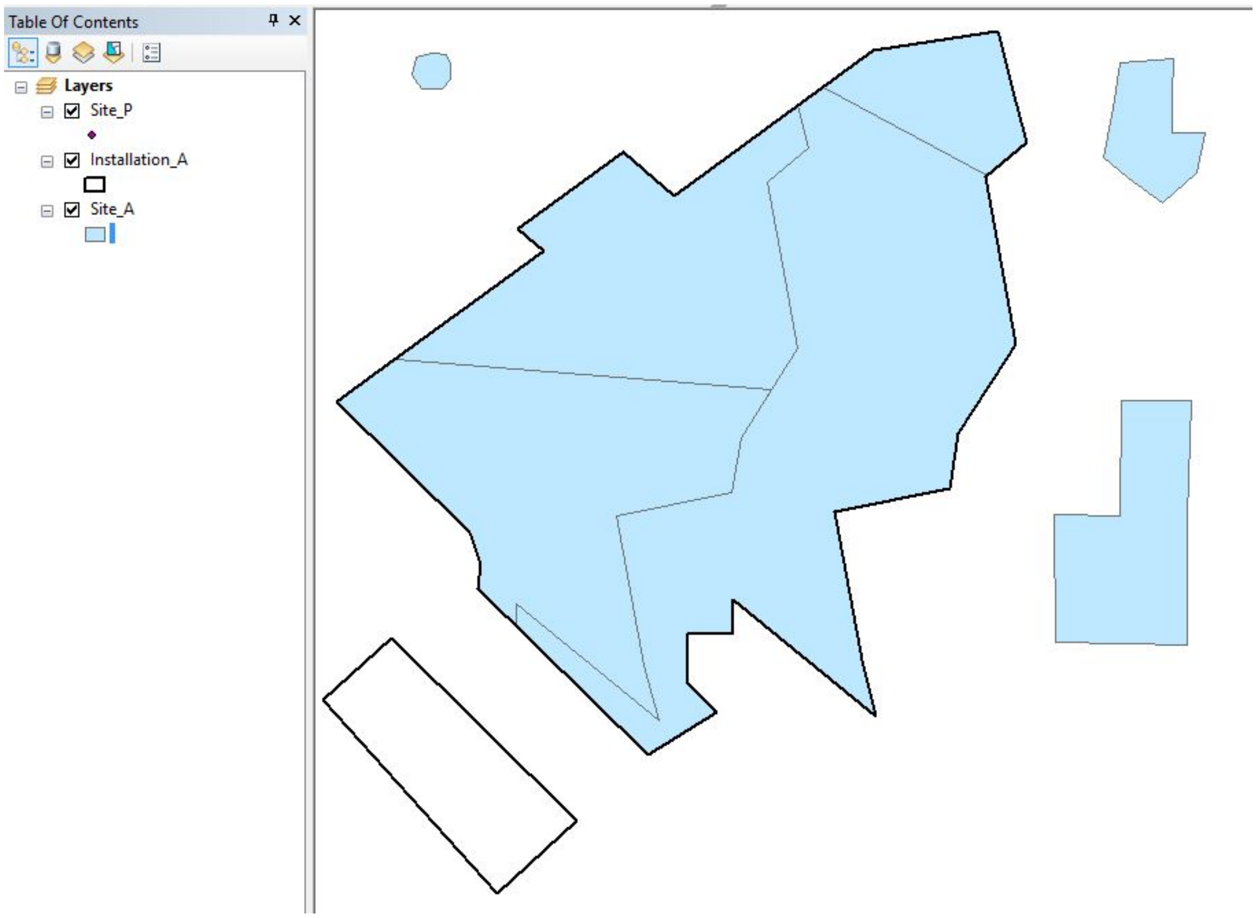


Figure 2.4: Before running the tool

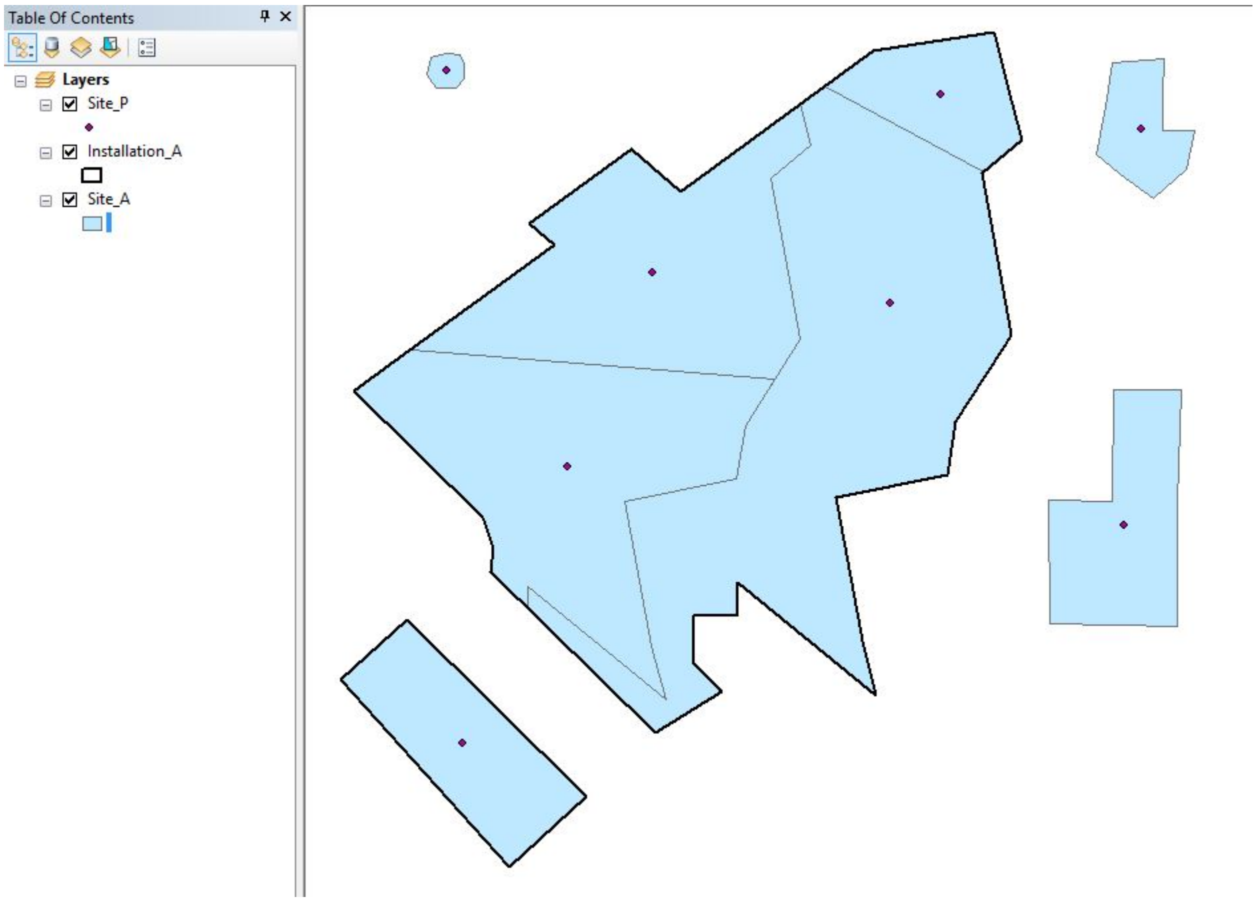


Figure 2.5: Newly created Site A and Site P features after running the tool

Chapter 3

Join Fields and Calculate

3.1 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.

3.2 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?

3.3 How to Use

3.3.1 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. 3.1).

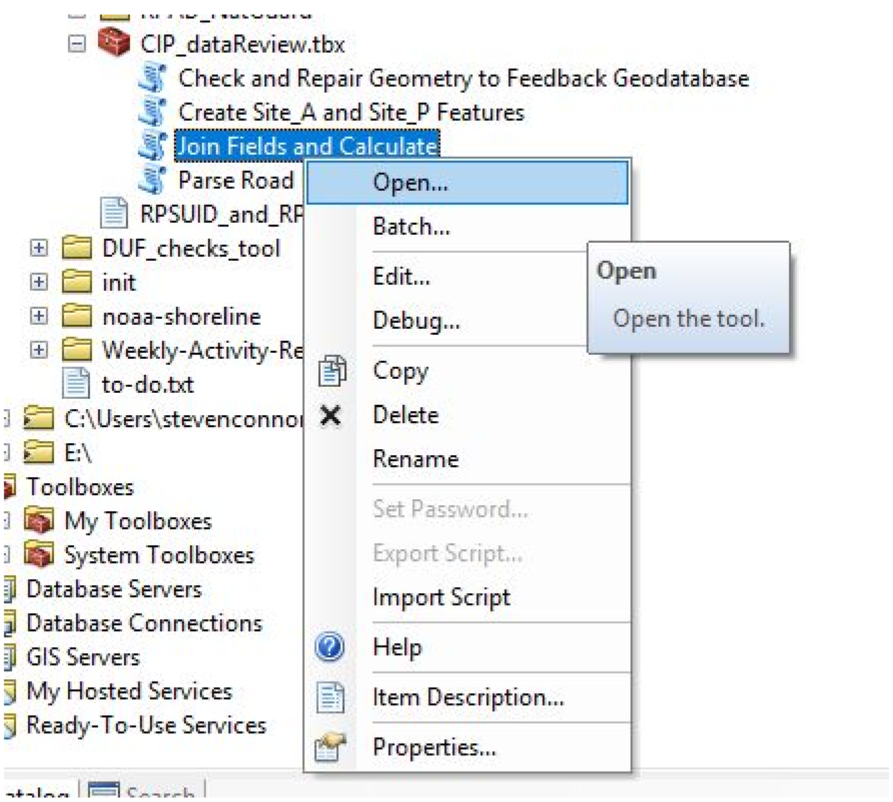


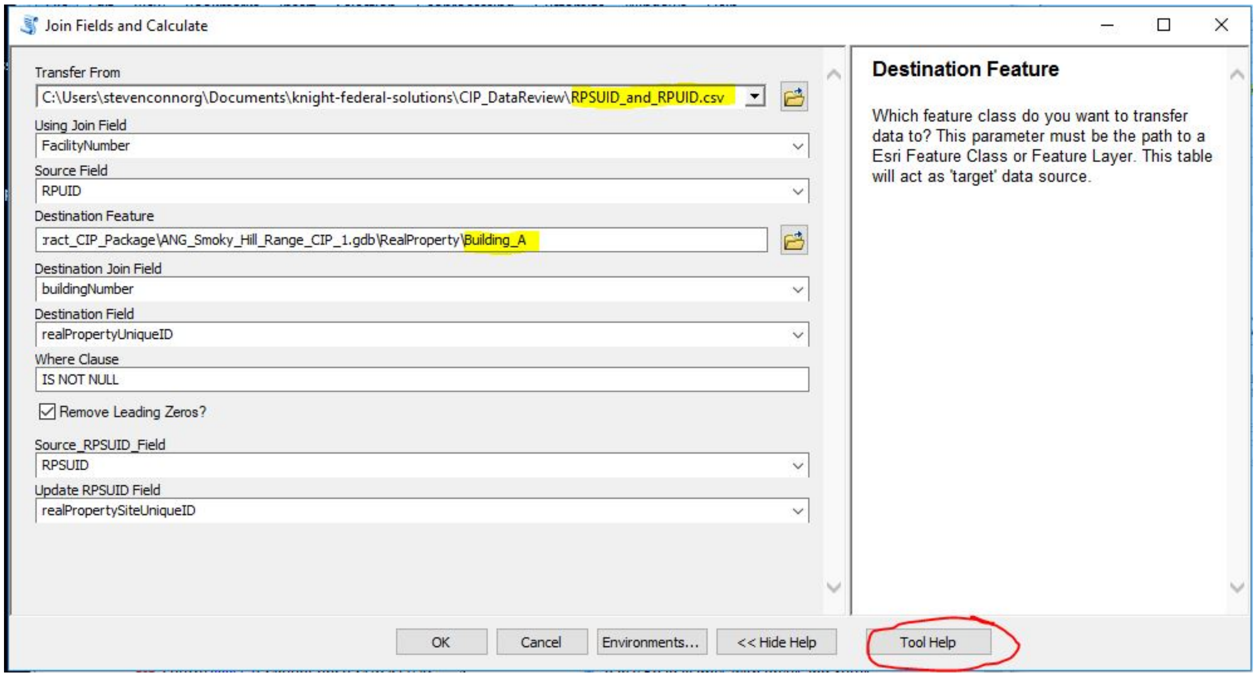
Figure 3.1: Opening the Tool

3.3.2 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 (Destinate Feature) ‘realPropertyUniqueID’ field (Destination_Field) using the ‘buildingNumber’ field (Destination_Join_Field) (Fig. 3.2).

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. 3.2).

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. 3.2).


Join Fields and Calculate

Transfer From
C:\Users\stevenconnorg\Documents\knight-federal-solutions\CIP_DataReview\RPSUID_and_RPUID.csv

Using Join Field
FacilityNumber

Source Field
RPUID

Destination Feature
ract_CIP_Package\ANG_Smoky_Hill_Range_CIP_1.gdb\RealProperty\Building_A

Destination Join Field
buildingNumber

Destination Field
realPropertyUniqueID

Where Clause
IS NOT NULL

☑ Remove Leading Zeros?

Source_RPSUID_Field
RPSUID

Update RPSUID Field
realPropertySiteUniqueID

Destination Feature

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.

Figure 3.2: Tool parameters

Alternatively, you may also run this tool in 'batch' for multiple features in a geodatabase or geodatabases (Fig. 3.3).

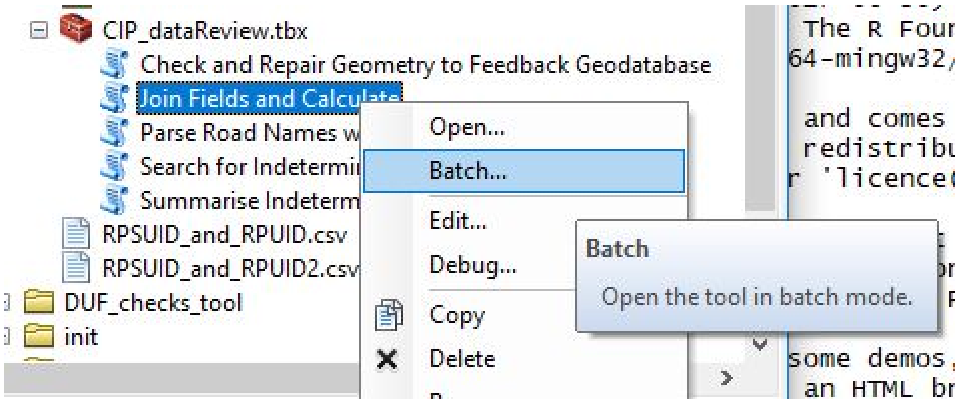
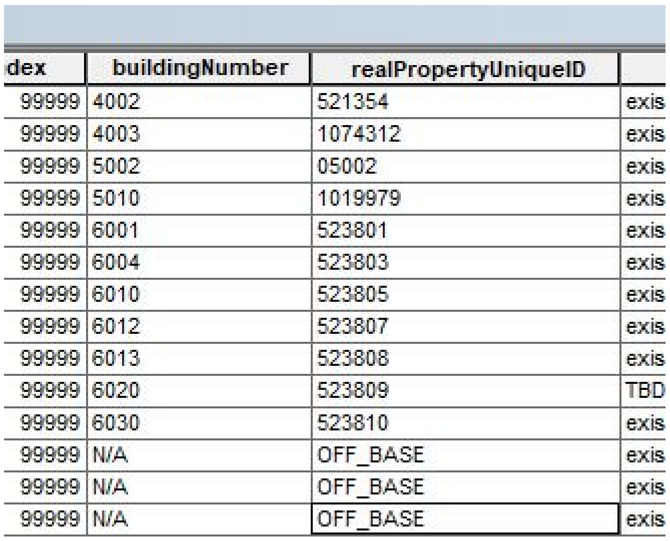


Figure 3.3: Running a tool in batch

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.

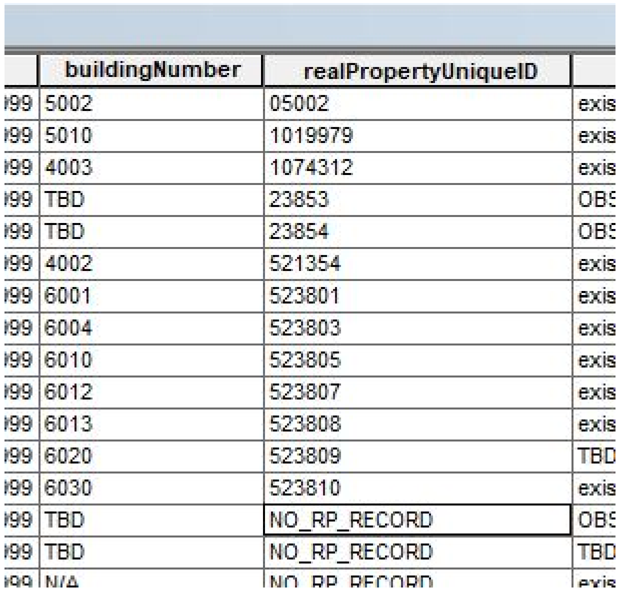
3.3.3 Run the Tool and View Results

Open the destinate Feature Class and view the update destination field values (Fig. 3.4, Fig. 3.5).



dex	buildingNumber	realPropertyUniqueID	
99999	4002	521354	exis
99999	4003	1074312	exis
99999	5002	05002	exis
99999	5010	1019979	exis
99999	6001	523801	exis
99999	6004	523803	exis
99999	6010	523805	exis
99999	6012	523807	exis
99999	6013	523808	exis
99999	6020	523809	TBD
99999	6030	523810	exis
99999	N/A	OFF_BASE	exis
99999	N/A	OFF_BASE	exis
99999	N/A	OFF_BASE	exis

Figure 3.4: Attribues before running the tool



	buildingNumber	realPropertyUniqueID	
99	5002	05002	exis
99	5010	1019979	exis
99	4003	1074312	exis
99	TBD	23853	OBS
99	TBD	23854	OBS
99	4002	521354	exis
99	6001	523801	exis
99	6004	523803	exis
99	6010	523805	exis
99	6012	523807	exis
99	6013	523808	exis
99	6020	523809	TBD
99	6030	523810	exis
99	TBD	NO_RP_RECORD	OBS
99	TBD	NO_RP_RECORD	TBD
99	N/A	NO_RP_RECORD	exis

Figure 3.5: Attributes after running the tool

Chapter 4

Calculate Feature RPSUIDs from Overlapping Polygons

4.1 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.

4.2 Parameters

The tool has 8 parameters:

1. **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.

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 'Calculate Feature RPSUIDs from Overlapping Polygon' script tool to open (Fig. 4.1).

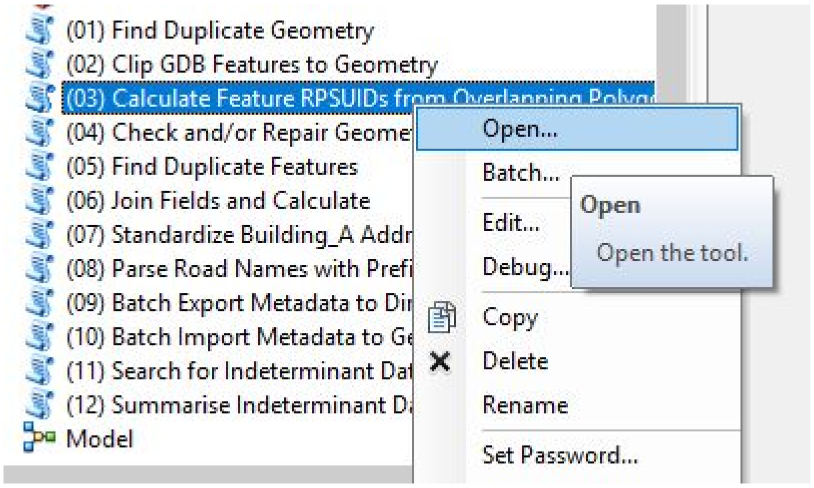
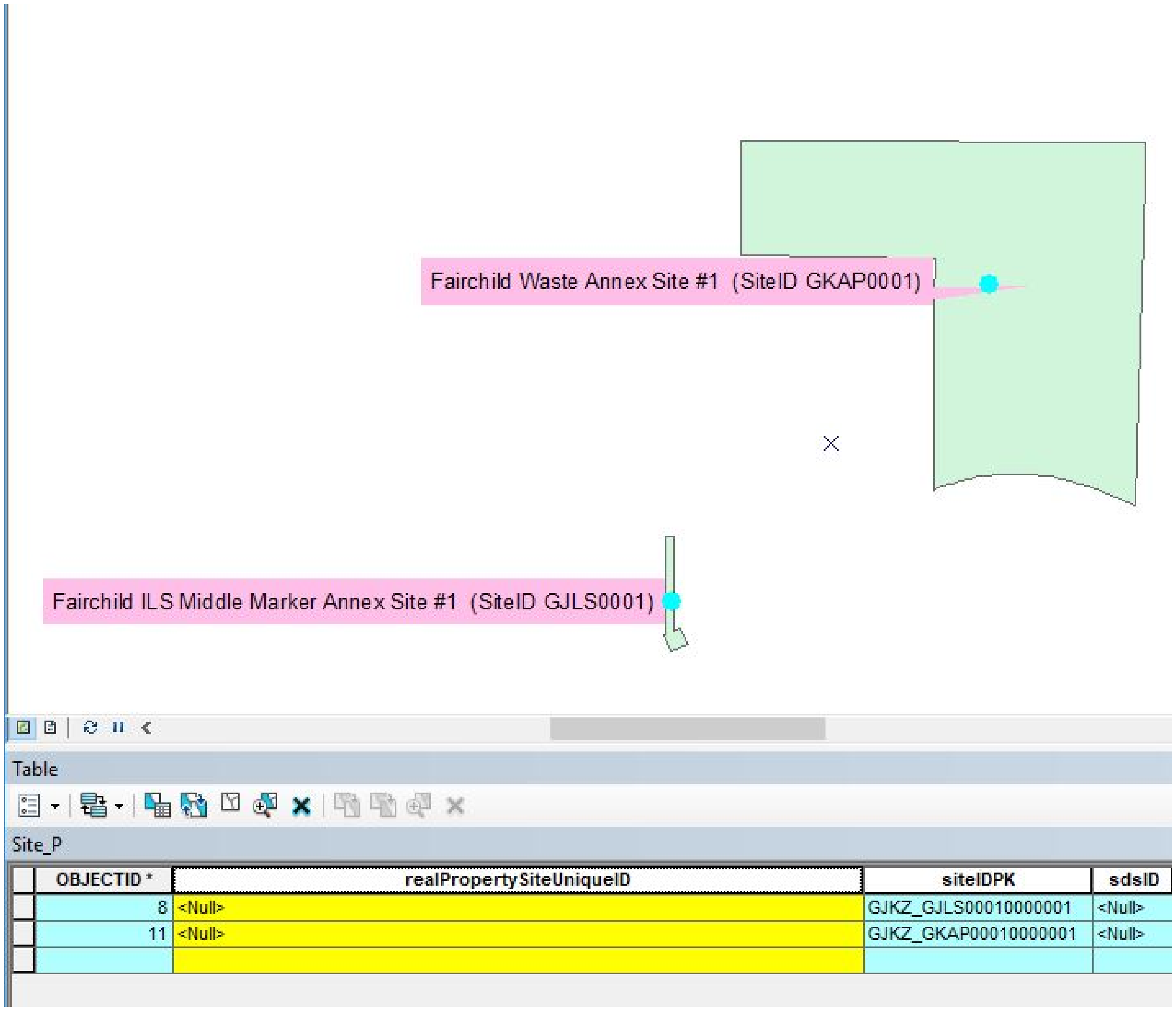

(01) Find Duplicate Geometry
(02) Clip GDB Features to Geometry
(03) Calculate Feature RPSUIDs from Overlapping Polyg
(04) Check and/or Repair Geome
(05) Find Duplicate Features
(06) Join Fields and Calculate
(07) Standardize Building_A Addr
(08) Parse Road Names with Prefi
(09) Batch Export Metadata to Dir
(10) Batch Import Metadata to Ge
(11) Search for Indeterminant Dat
(12) Summarise Indeterminant D
Model
Open...
Batch...
Edit...
Debug...
Copy
Delete
Rename
Set Password...
Open
Open the tool.

Figure 4.1: Opening the toolbox

4.3.2 Fill out the parameters

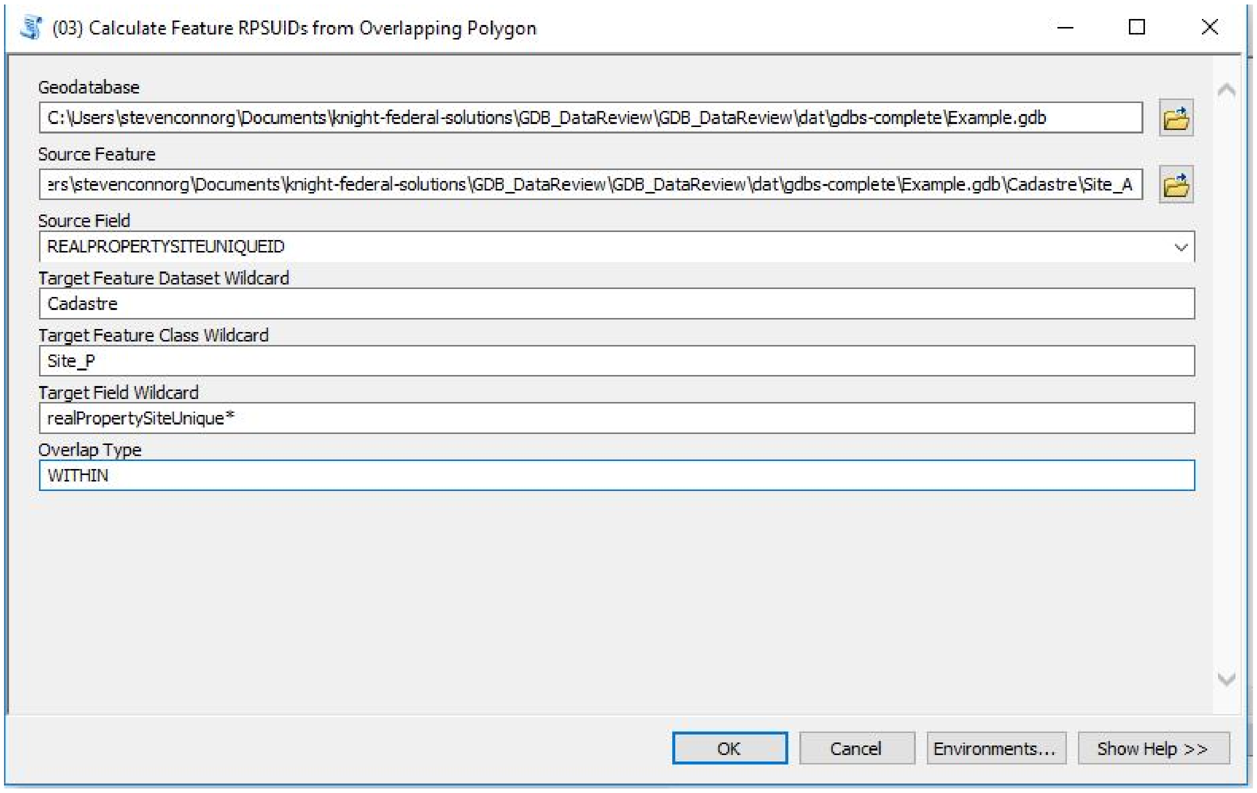
For this demostration, we want to update missing RPSUID values for 2 features in the Site_P Feature Class using RPSUID values from Site_A features that contain Site_P features 4.2).


Fairchild Waste Annex Site #1 (SiteID GKAP0001)
Fairchild ILS Middle Marker Annex Site #1 (SiteID GJLS0001)
Table
Site_P

OBJECTID *	realPropertySiteUniqueID	siteIDPK	sdsID
8	<Null>	GJKZ_GJLS00010000001	<Null>
11	<Null>	GJKZ_GKAP00010000001	<Null>

Figure 4.2: Missing RPSUID attributes for Site Point featuresl

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. 4.3).


(03) Calculate Feature RPSUIDs from Overlapping Polygon

Geodatabase
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb

Source Feature
ers\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb\Cadastre\Site_A

Source Field
REALPROPERTYSITEUNIQUEID

Target Feature Dataset Wildcard
Cadastre

Target Feature Class Wildcard
Site_P

Target Field Wildcard
realPropertySiteUnique*

Overlap Type
WITHIN

Figure 4.3: Tool parameters

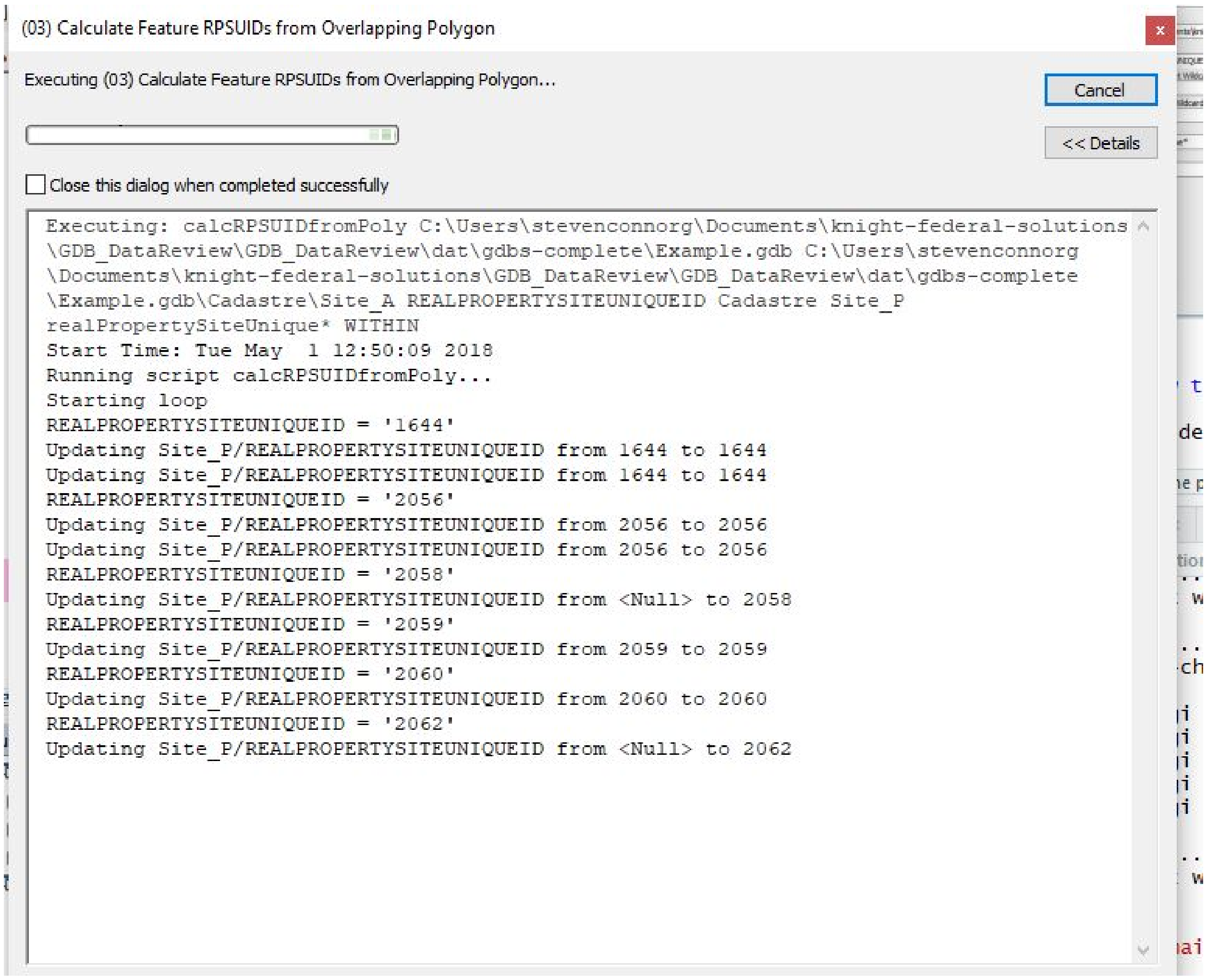
Since we only want to update the Site_P features within the Cadastre Feature Dataset, we change the default value for the Target Feature Dataset Wildcard to "Cadastre," since we know that the Site_P Feature Class is only found within the Cadastre Feature Dataset. Further, we change the default value 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.

4.4 Run the Tool and View Results

If running the tool with Background Processessing disabled, we can see which RPSUIDs are being updated (Fig. 4.4).



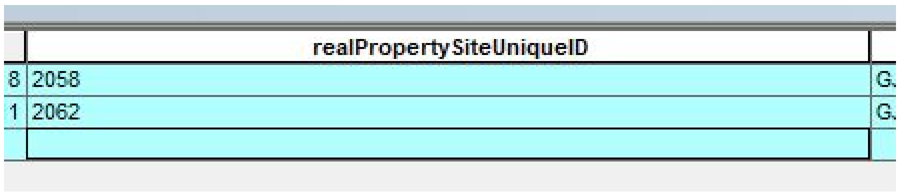
Executing (03) Calculate Feature RPSUIDs from Overlapping Polygon...

Close this dialog when completed successfully

```
Executing: calcRPSUIDfromPoly C:\Users\stevenconnorg\Documents\knight-federal-solutions
\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb C:\Users\stevenconnorg
\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete
\Example.gdb\Cadastre\Site_A REALPROPERTYSITEUNIQUEID Cadastre Site_P
realPropertySiteUnique* WITHIN
Start Time: Tue May  1 12:50:09 2018
Running script calcRPSUIDfromPoly...
Starting loop
REALPROPERTYSITEUNIQUEID = '1644'
Updating Site_P/REALPROPERTYSITEUNIQUEID from 1644 to 1644
Updating Site_P/REALPROPERTYSITEUNIQUEID from 1644 to 1644
REALPROPERTYSITEUNIQUEID = '2056'
Updating Site_P/REALPROPERTYSITEUNIQUEID from 2056 to 2056
Updating Site_P/REALPROPERTYSITEUNIQUEID from 2056 to 2056
REALPROPERTYSITEUNIQUEID = '2058'
Updating Site_P/REALPROPERTYSITEUNIQUEID from <Null> to 2058
REALPROPERTYSITEUNIQUEID = '2059'
Updating Site_P/REALPROPERTYSITEUNIQUEID from 2059 to 2059
REALPROPERTYSITEUNIQUEID = '2060'
Updating Site_P/REALPROPERTYSITEUNIQUEID from 2060 to 2060
REALPROPERTYSITEUNIQUEID = '2062'
Updating Site_P/REALPROPERTYSITEUNIQUEID from <Null> to 2062
```

Figure 4.4: Tool parameters

After the tool has run, we see that the 2 Site_P features with missing RPSUID values are updated accordingly 4.5).



	realPropertySiteUniqueID	
8	2058	G.
1	2062	G.

Figure 4.5: Updated attributes after running the tool

Chapter 5

Find Duplicate Geometry

5.1 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.

5.2 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.

5.3 How to Use

5.3.1 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. 5.1).

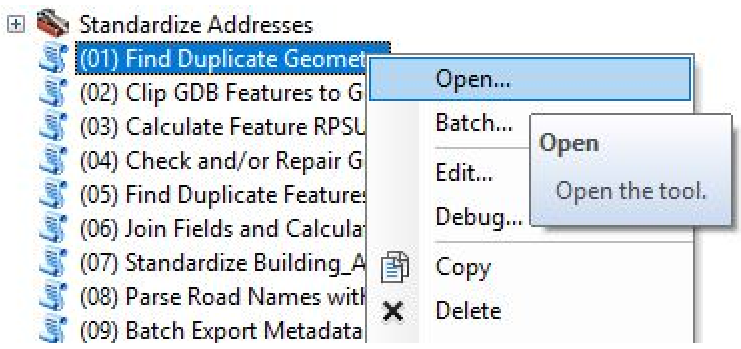
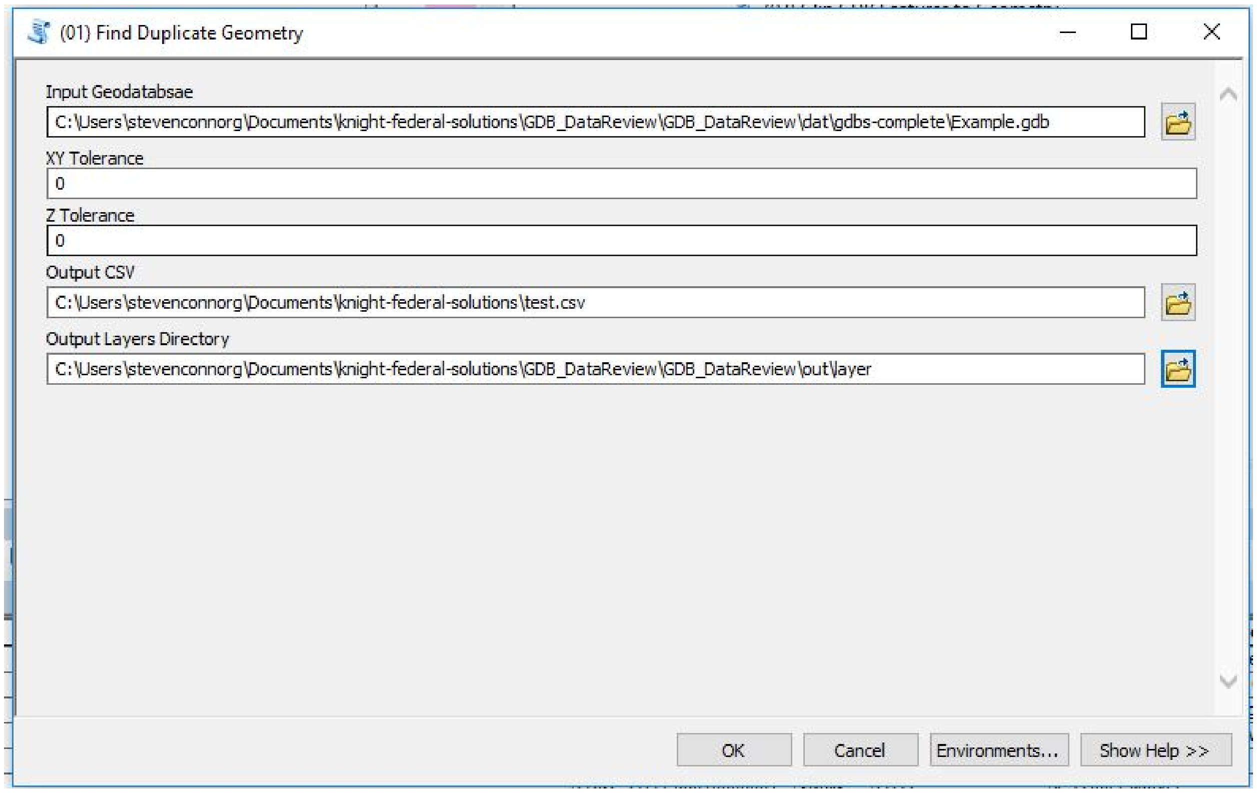

Standardize Addresses
(01) Find Duplicate Geomet
(02) Clip GDB Features to G
(03) Calculate Feature RPSU
(04) Check and/or Repair G
(05) Find Duplicate Feature
(06) Join Fields and Calcula
(07) Standardize Building_A
(08) Parse Road Names with
(09) Batch Export Metadata
Open...
Batch...
Edit...
Debug...
Copy
Delete
Open
Open the tool.

Figure 5.1: Opening the Find Duplicate Geometries tool

5.3.2 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. 5.2). 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.’


(01) Find Duplicate Geometry

Input Geodatabsae
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb

XY Tolerance
0

Z Tolerance
0

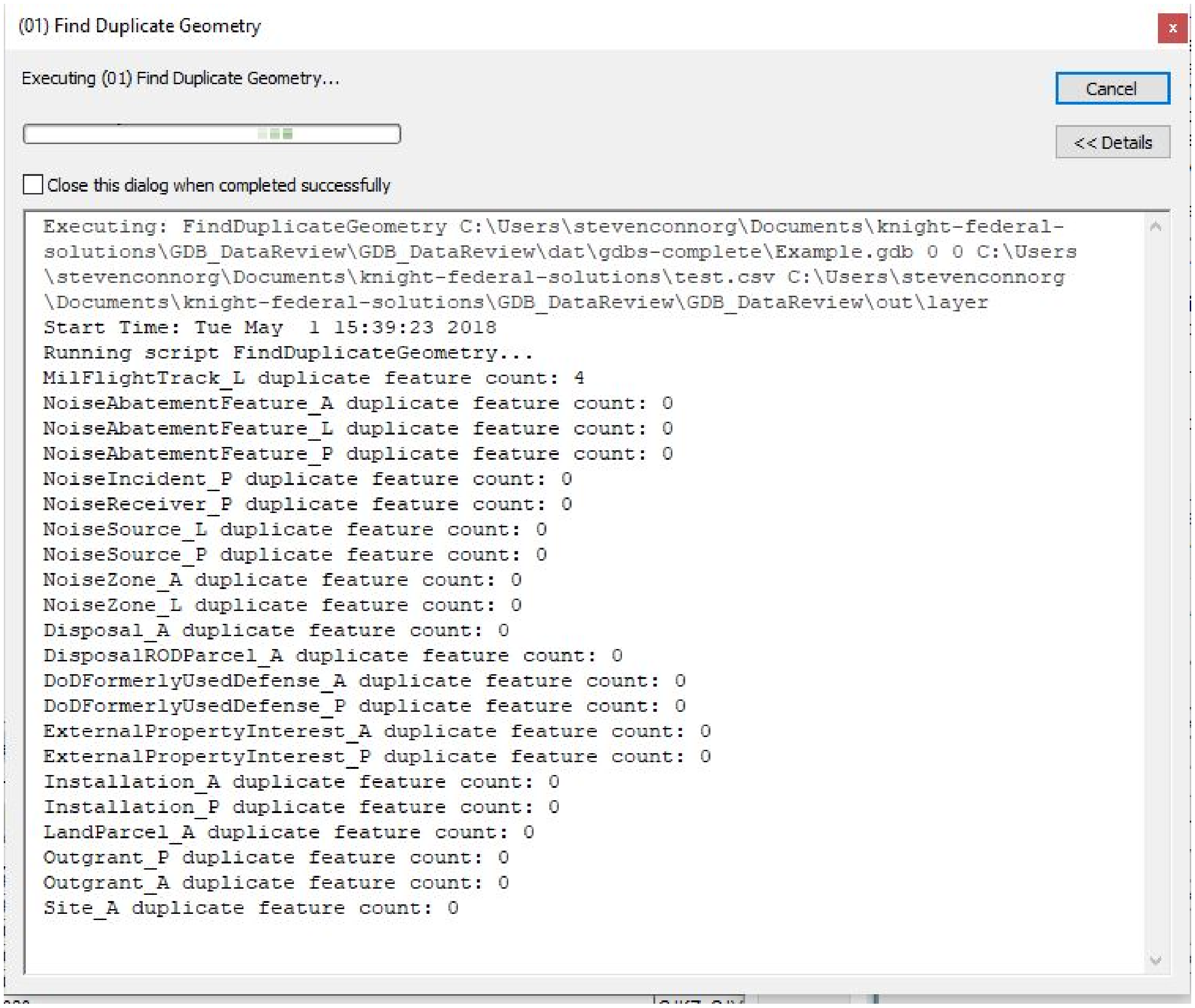
Output CSV
C:\Users\stevenconnorg\Documents\knight-federal-solutions\test.csv

Output Layers Directory
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\out\layer

Figure 5.2: Find Duplicate Geometries parameters

5.4 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. 5.3).


(01) Find Duplicate Geometry

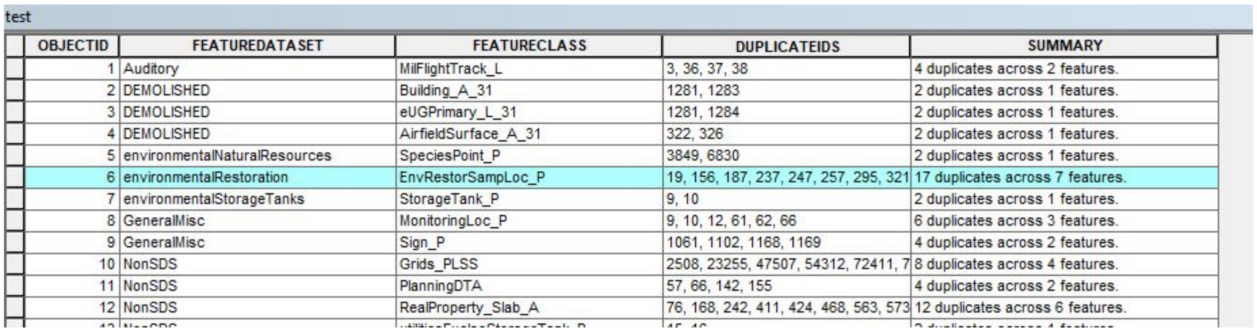
Executing (01) Find Duplicate Geometry...

Close this dialog when completed successfully

```
Executing: FindDuplicateGeometry C:\Users\stevenconnorg\Documents\knight-federal-
solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb 0 0 C:\Users
\stevenconnorg\Documents\knight-federal-solutions\test.csv C:\Users\stevenconnorg
\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\out\layer
Start Time: Tue May  1 15:39:23 2018
Running script FindDuplicateGeometry...
MilFlightTrack_L duplicate feature count: 4
NoiseAbatementFeature_A duplicate feature count: 0
NoiseAbatementFeature_L duplicate feature count: 0
NoiseAbatementFeature_P duplicate feature count: 0
NoiseIncident_P duplicate feature count: 0
NoiseReceiver_P duplicate feature count: 0
NoiseSource_L duplicate feature count: 0
NoiseSource_P duplicate feature count: 0
NoiseZone_A duplicate feature count: 0
NoiseZone_L duplicate feature count: 0
Disposal_A duplicate feature count: 0
DisposalRODParcel_A duplicate feature count: 0
DoDFormerlyUsedDefense_A duplicate feature count: 0
DoDFormerlyUsedDefense_P duplicate feature count: 0
ExternalPropertyInterest_A duplicate feature count: 0
ExternalPropertyInterest_P duplicate feature count: 0
Installation_A duplicate feature count: 0
Installation_P duplicate feature count: 0
LandParcel_A duplicate feature count: 0
Outgrant_P duplicate feature count: 0
Outgrant_A duplicate feature count: 0
Site_A duplicate feature count: 0
```

Figure 5.3: Find Duplicate Geometries parameters

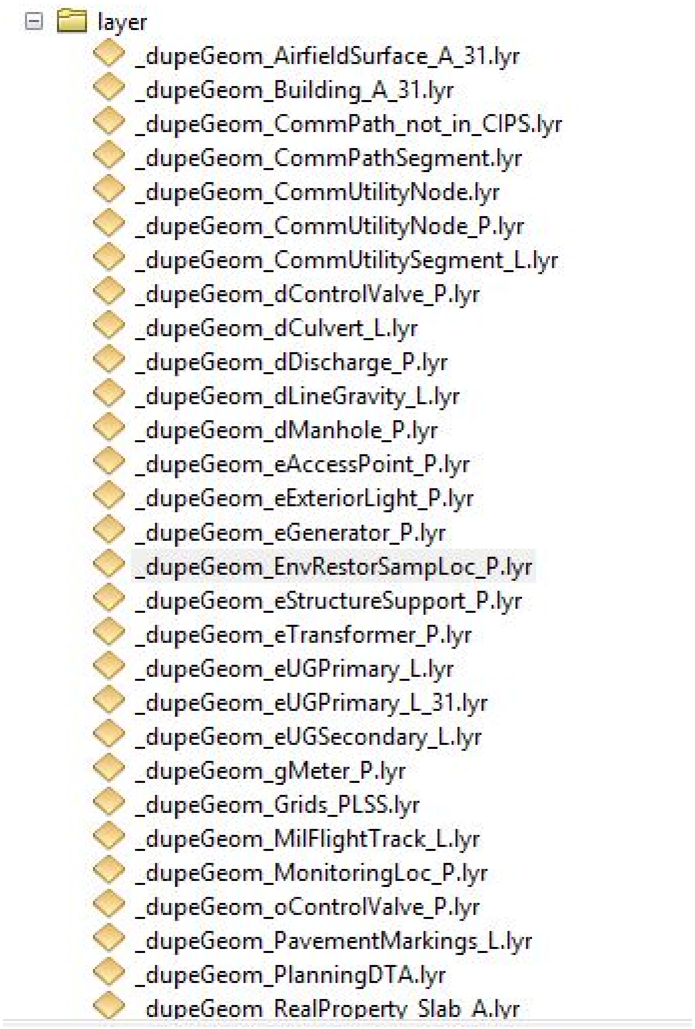
After the tool has run, we can open the output .csv we specified in the tool parameters to examine which Feature Classes have duplicated geometries . For example, we find that the EnvRestorSampLoc_P Feature Class within the environmentalRestoration Feature Dataset has 17 total duplicates spread across 7 unique geometries (Fig. 5.4).


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 5.4: 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. 5.5).



layer
_dupeGeom_AirfieldSurface_A_31.lyr
_dupeGeom_Building_A_31.lyr
_dupeGeom_CommPath_not_in_CIPS.lyr
_dupeGeom_CommPathSegment.lyr
_dupeGeom_CommUtilityNode.lyr
_dupeGeom_CommUtilityNode_P.lyr
_dupeGeom_CommUtilitySegment_L.lyr
_dupeGeom_dControlValve_P.lyr
_dupeGeom_dCulvert_L.lyr
_dupeGeom_dDischarge_P.lyr
_dupeGeom_dLineGravity_L.lyr
_dupeGeom_dManhole_P.lyr
_dupeGeom_eAccessPoint_P.lyr
_dupeGeom_eExteriorLight_P.lyr
_dupeGeom_eGenerator_P.lyr
_dupeGeom_EnvRestorSampLoc_P.lyr
_dupeGeom_eStructureSupport_P.lyr
_dupeGeom_eTransformer_P.lyr
_dupeGeom_eUGPrimary_L.lyr
_dupeGeom_eUGPrimary_L_31.lyr
_dupeGeom_eUGSecondary_L.lyr
_dupeGeom_gMeter_P.lyr
_dupeGeom_Grids_PLSS.lyr
_dupeGeom_MilFlightTrack_L.lyr
_dupeGeom_MonitoringLoc_P.lyr
_dupeGeom_oControlValve_P.lyr
_dupeGeom_PavementMarkings_L.lyr
_dupeGeom_PlanningDTA.lyr
_dupeGeom_RealProperty_Slab_A.lyr

Figure 5.5: 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. 5.6).

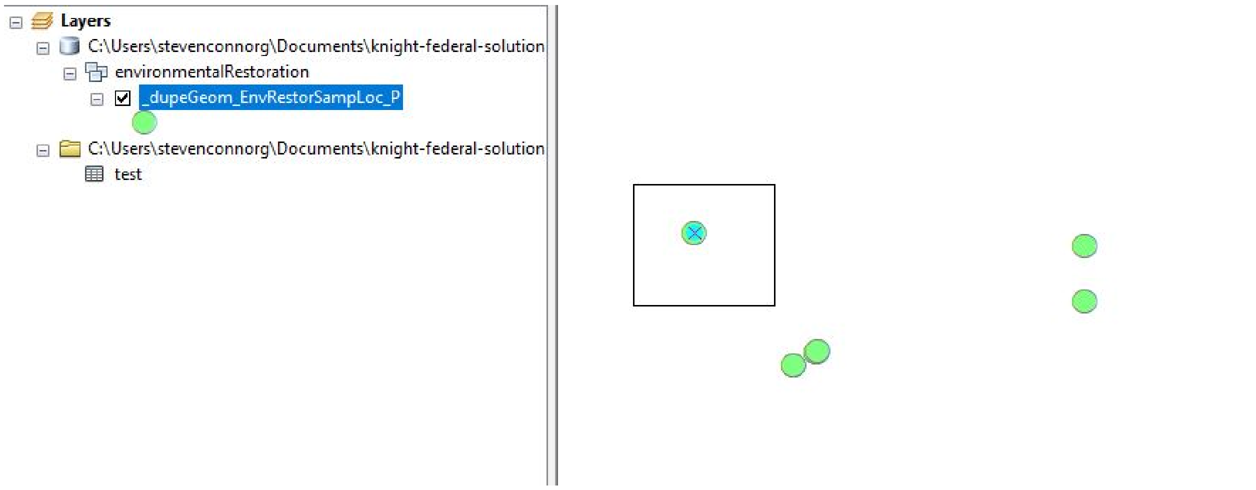
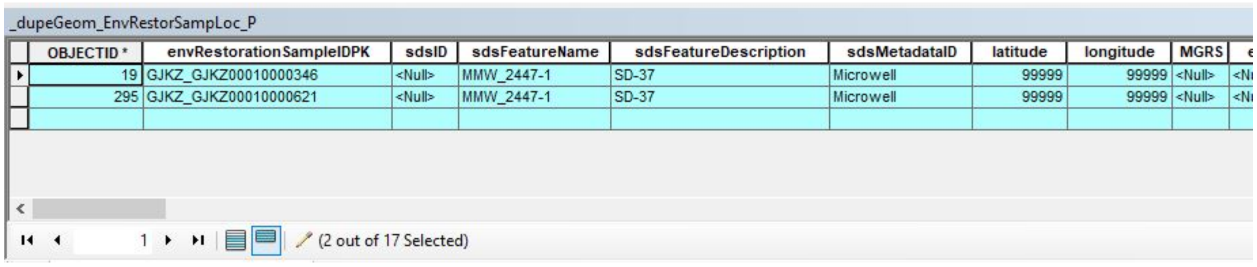

Layers
C:\Users\stevenconnorg\Documents\knight-federal-solution
environmentalRestoration
_dupeGeom_EnvRestorSampLoc_P
C:\Users\stevenconnorg\Documents\knight-federal-solution
test

Figure 5.6: 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. 5.7). 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	e
19	GJKZ_GJKZ00010000346	<Null>	MMW_2447-1	SD-37	Microwell	99999	99999	<Null>	<Nu
295	GJKZ_GJKZ00010000621	<Null>	MMW_2447-1	SD-37	Microwell	99999	99999	<Null>	<Nu

1 (2 out of 17 Selected)

Figure 5.7: Attributes of duplicated features

we can pull in the layer files created for each Feature Class to manually inspect the duplicated features to determine if/which features should be amended or deleted.

Chapter 6

Delete Duplicate Features

6.1 Overview

The Find 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.

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'.

6.2 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).

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 Features' script tool to open (Fig. 6.1).

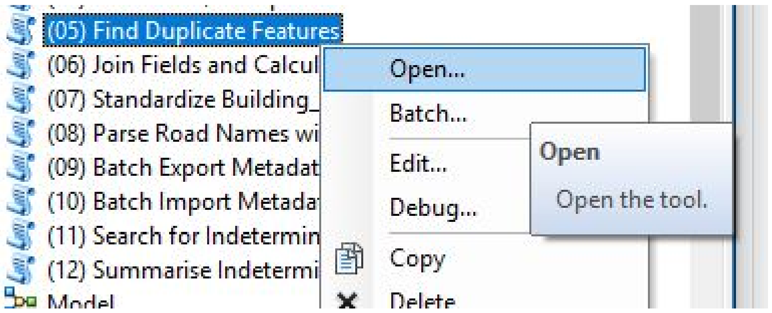
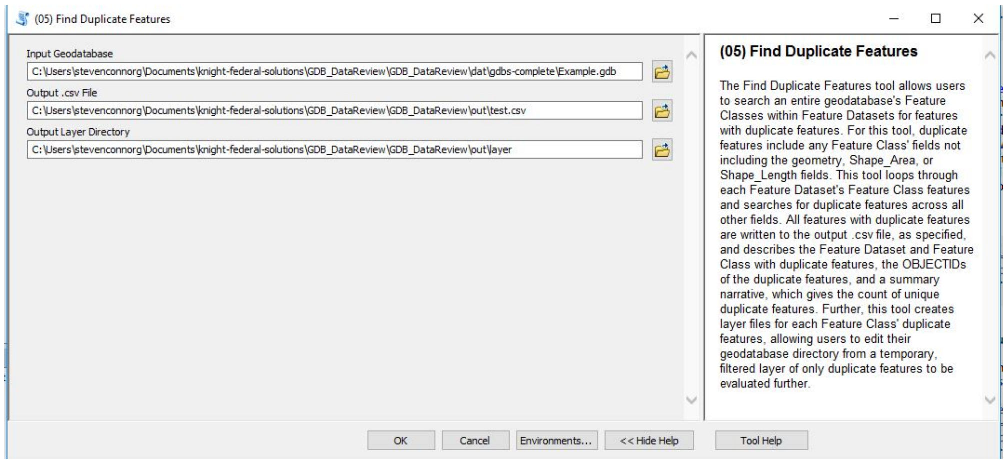


Figure 6.1: Opening the Delete Duplicate Features tool

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 features (Fig. 6.2). 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.


(05) Find Duplicate Features

Input Geodatabase
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb

Output .csv File
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\out\test.csv

Output Layer Directory
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\out\layer

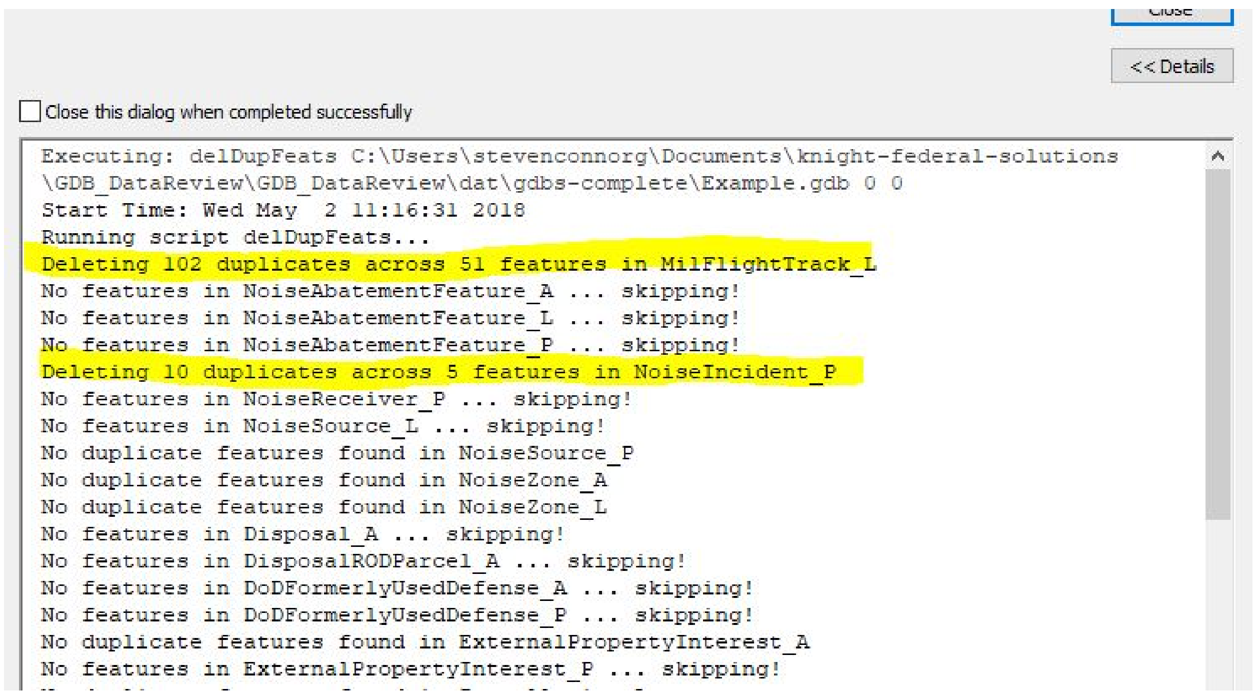
(05) Find Duplicate Features

The Find Duplicate Features tool allows users to search an entire geodatabase's Feature Classes within Feature Datasets for features with duplicate features. For this tool, duplicate features include any Feature Class' fields not including the geometry, Shape_Area, or Shape_Length fields. This tool loops through each Feature Dataset's Feature Class features and searches for duplicate features across all other fields. All features with duplicate features are written to the output .csv file, as specified, and describes the Feature Dataset and Feature Class with duplicate features, the OBJECTIDs of the duplicate features, and a summary narrative, which gives the count of unique duplicate features. 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.

Figure 6.2: Delete Duplicate Features parameters

6.3.3 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. 6.3). 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).



```
Executing: delDupFeats C:\Users\stevenconnorg\Documents\knight-federal-solutions
\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb 0 0
Start Time: Wed May  2 11:16:31 2018
Running script delDupFeats...
Deleting 102 duplicates across 51 features in MilFlightTrack_L
No features in NoiseAbatementFeature_A ... skipping!
No features in NoiseAbatementFeature_L ... skipping!
No features in NoiseAbatementFeature_P ... skipping!
Deleting 10 duplicates across 5 features in NoiseIncident_P
No features in NoiseReceiver_P ... skipping!
No features in NoiseSource_L ... skipping!
No duplicate features found in NoiseSource_P
No duplicate features found in NoiseZone_A
No duplicate features found in NoiseZone_L
No features in Disposal_A ... skipping!
No features in DisposalRODParcel_A ... skipping!
No features in DoDFormerlyUsedDefense_A ... skipping!
No features in DoDFormerlyUsedDefense_P ... skipping!
No duplicate features found in ExternalPropertyInterest_A
No features in ExternalPropertyInterest_P ... skipping!
```

Figure 6.3: Delete Duplicate Features messages

Chapter 7

Check and/or Repair Geometries

7.1 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.

7.2 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. **Installation_Review_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.

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 'Check and/or Repair Geometry' script tool to open (Fig. 7.1).

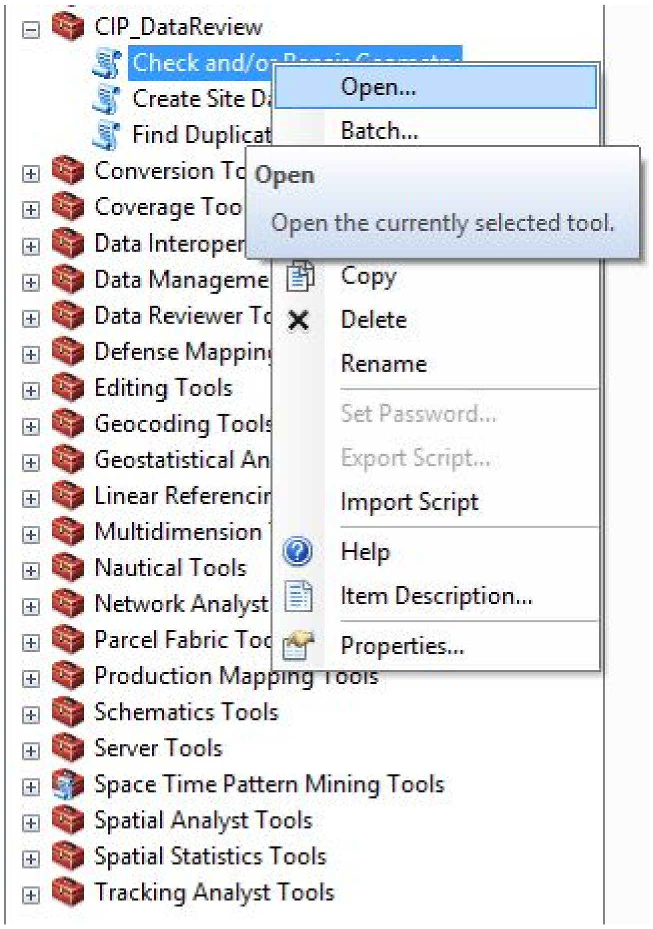
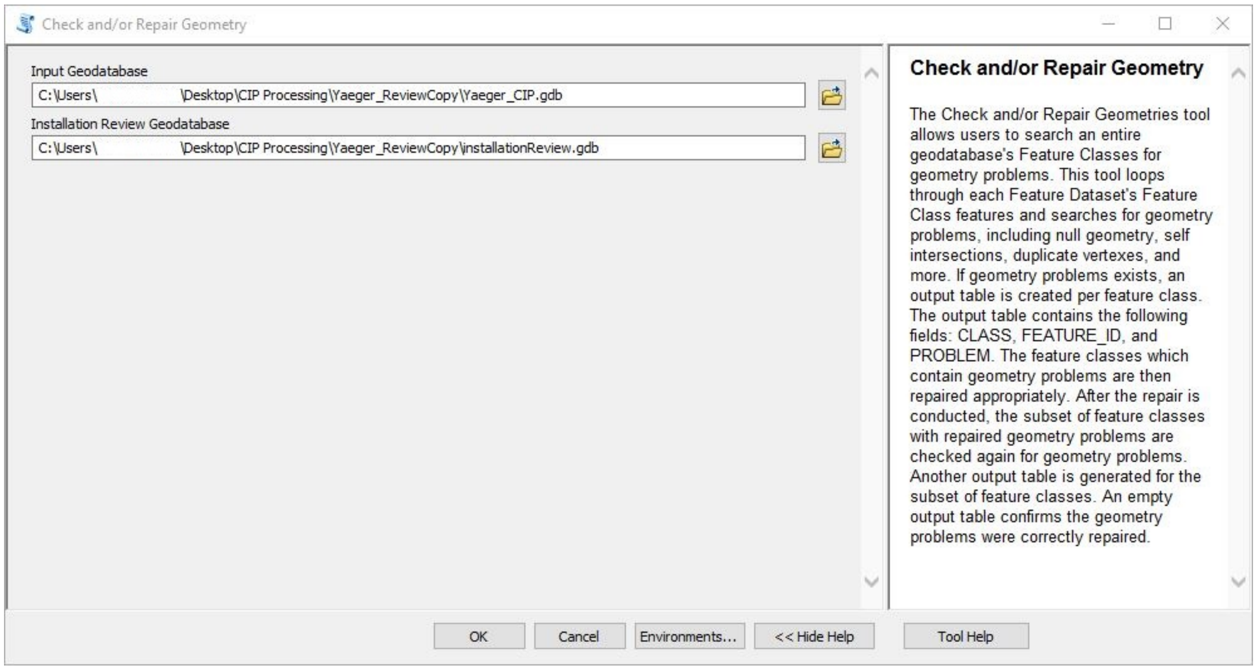


Figure 7.1: Opening the Delete Duplicate Features tool

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). Last, we specify where we want to output the resulting tables, prefereably in an Installation Review geodatabase specifically for holding CIP processing results.



Input Geodatabase
C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb
Installation Review Geodatabase
C:\Users\ \Desktop\CIP Processing\Yaeger_ReviewCopy\installationReview.gdb

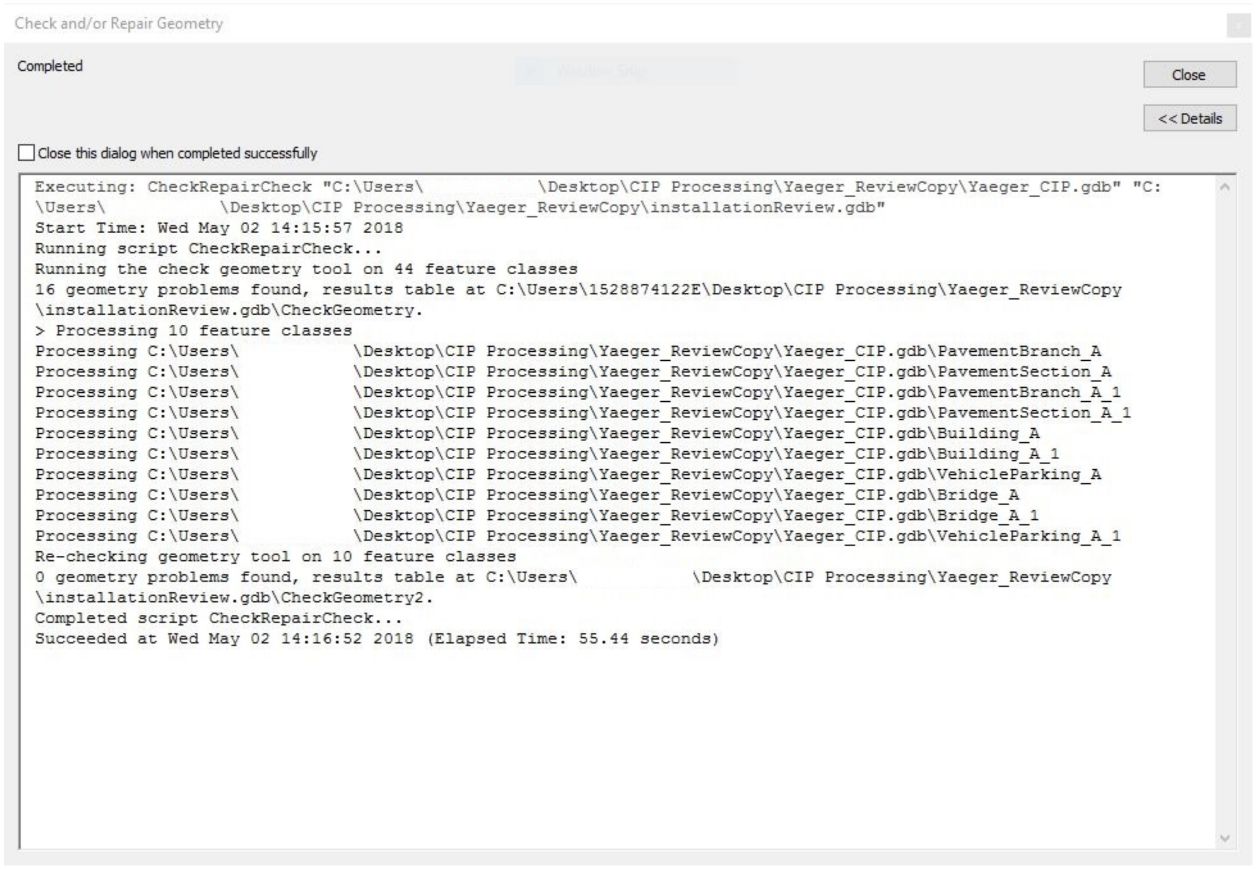
Check and/or Repair Geometry

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 per feature class. The output table contains the following fields: CLASS, FEATURE_ID, and PROBLEM. The feature classes which contain geometry problems are then repaired appropriately. After the repair is conducted, the subset of feature classes with repaired geometry problems are checked again for geometry problems. Another output table is generated for the subset of feature classes. An empty output table confirms the geometry problems were correctly repaired.

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 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. 7.3).


Check and/or Repair Geometry
Completed
Close
<< Details
Close this dialog when completed successfully

```
Executing: CheckRepairCheck "C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb" "C:
\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\installationReview.gdb"
Start Time: Wed May 02 14:15:57 2018
Running script CheckRepairCheck...
Running the check geometry tool on 44 feature classes
16 geometry problems found, results table at C:\Users\1528874122E\Desktop\CIP Processing\Yaeger_ReviewCopy
\installationReview.gdb\CheckGeometry.
> Processing 10 feature classes
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementSection_A
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementBranch_A_1
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\PavementSection_A_1
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Building_A
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Building_A_1
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\VehicleParking_A
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Bridge_A
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\Bridge_A_1
Processing C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy\Yaeger_CIP.gdb\VehicleParking_A_1
Re-checking geometry tool on 10 feature classes
0 geometry problems found, results table at C:\Users\           \Desktop\CIP Processing\Yaeger_ReviewCopy
\installationReview.gdb\CheckGeometry2.
Completed script CheckRepairCheck...
Succeeded at Wed May 02 14:16:52 2018 (Elapsed Time: 55.44 seconds)
```

Figure 7.3: Delete Duplicate Features messages

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. 7.4).

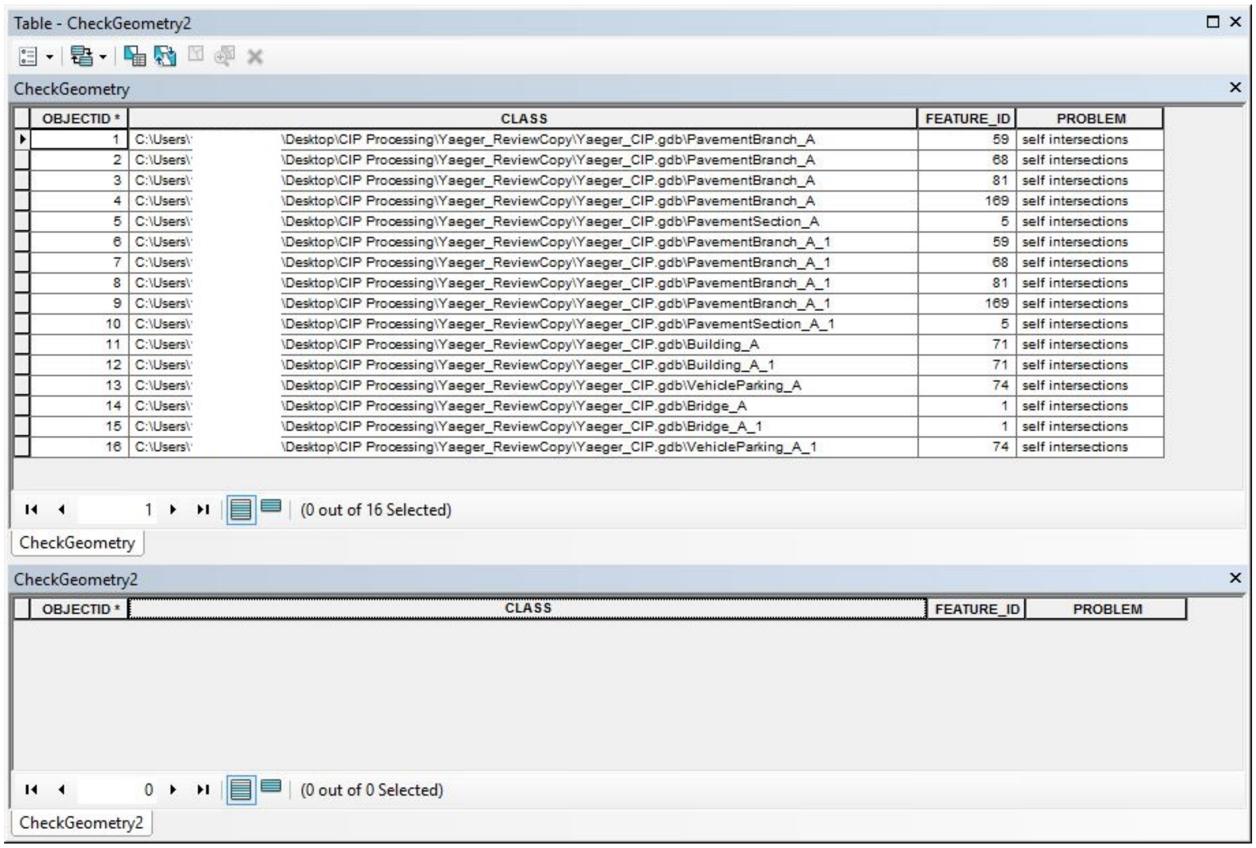


Table - CheckGeometry2

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 7.4: Delete Duplicate Features messages

Chapter 8

Standardized Address Field

8.1 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.

8.2 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.

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 'Find Duplicate Features' script tool to open (Fig. 8.1).

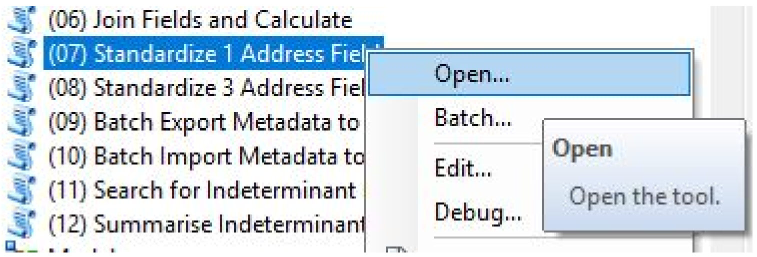
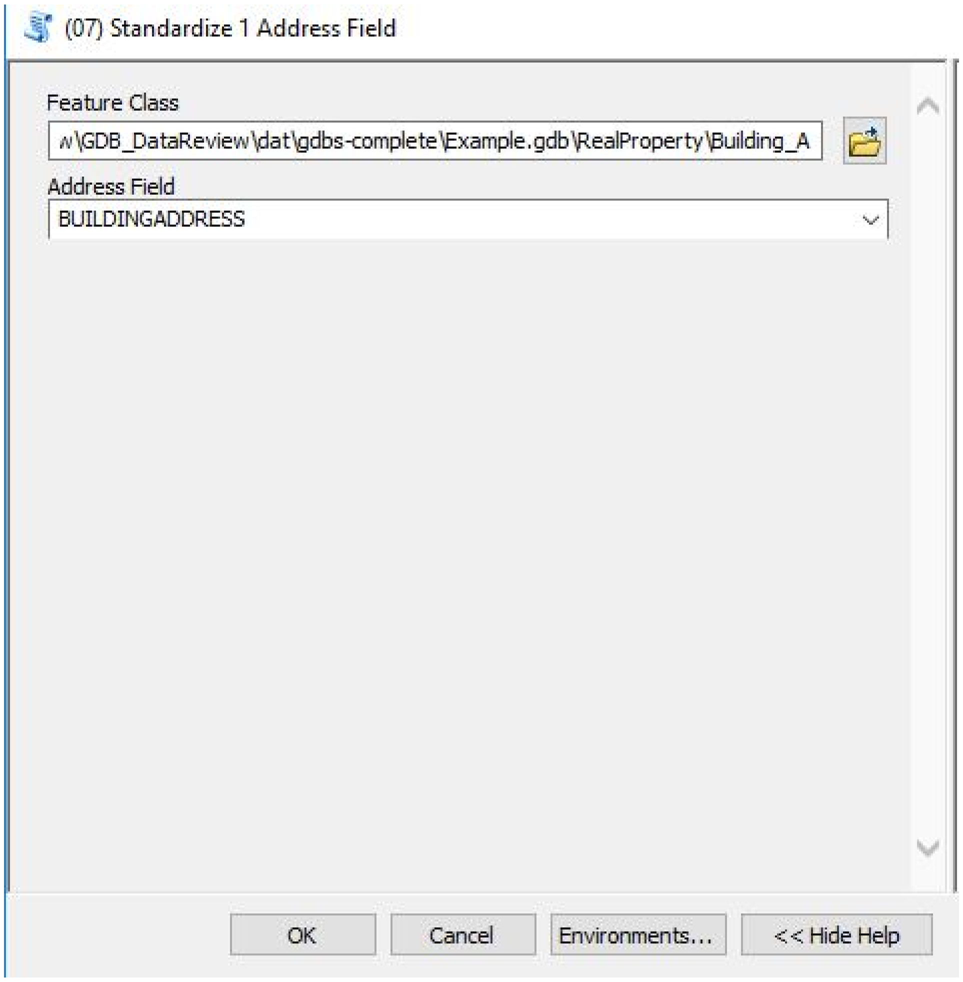

(06) Join Fields and Calculate
(07) Standardize 1 Address Fiel
(08) Standardize 3 Address Fiel
(09) Batch Export Metadata to
(10) Batch Import Metadata to
(11) Search for Indeterminant
(12) Summarise Indeterminant
Open...
Batch...
Edit...
Debug...
Open
Open the tool.

Figure 8.1: Opening the Delete Duplicate Features tool

8.3.2 Fill out the parameters

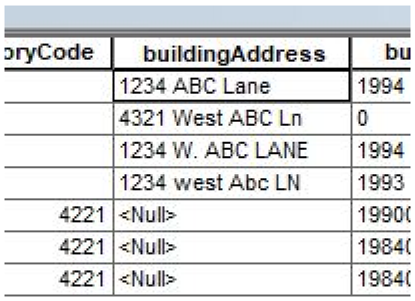
Next, fill out the parameters for the tool. Here, we want to update the buildingAddress field in the Building_A Feature Class in the Example.gdb (Fig. 8.2) because we notice none standardized addresses in the field (Fig. 8.3).


(07) Standardize 1 Address Field

Feature Class: w\GDB_DataReview\dat\gdbs-complete\Example.gdb\RealProperty\Building_A

Address Field: BUILDINGADDRESS

Figure 8.2: Delete Duplicate Features parameters

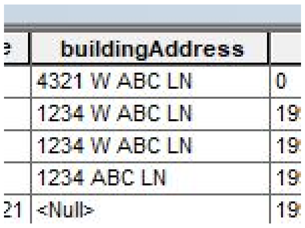


oryCode	buildingAddress	bu
	1234 ABC Lane	1994
	4321 West ABC Ln	0
	1234 W. ABC LANE	1994
	1234 west Abc LN	1993
4221	<Null>	1990
4221	<Null>	1984
4221	<Null>	1984

Figure 8.3: Delete Duplicate Features parameters

8.3.3 Run the Tool and View Results

After the tool has run, we can see that the building address values have been appropriately standardized (Fig. 8.4).



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 8.4: Delete Duplicate Features messages

Chapter 9

Standardized Road Prefix, Name, and Suffix

9.1 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.

9.2 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.

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 3 Address Fields' script tool to open (Fig. 9.1).

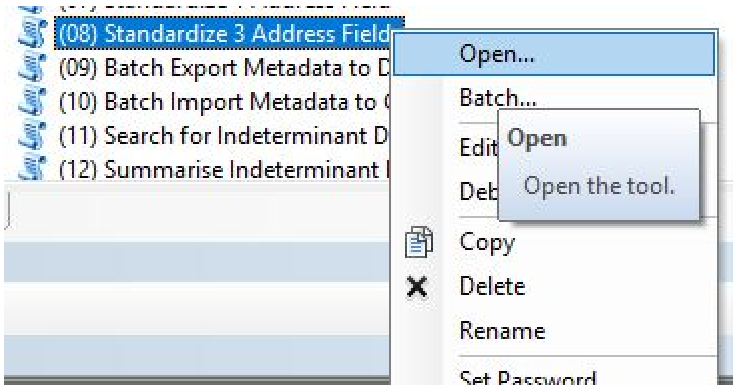
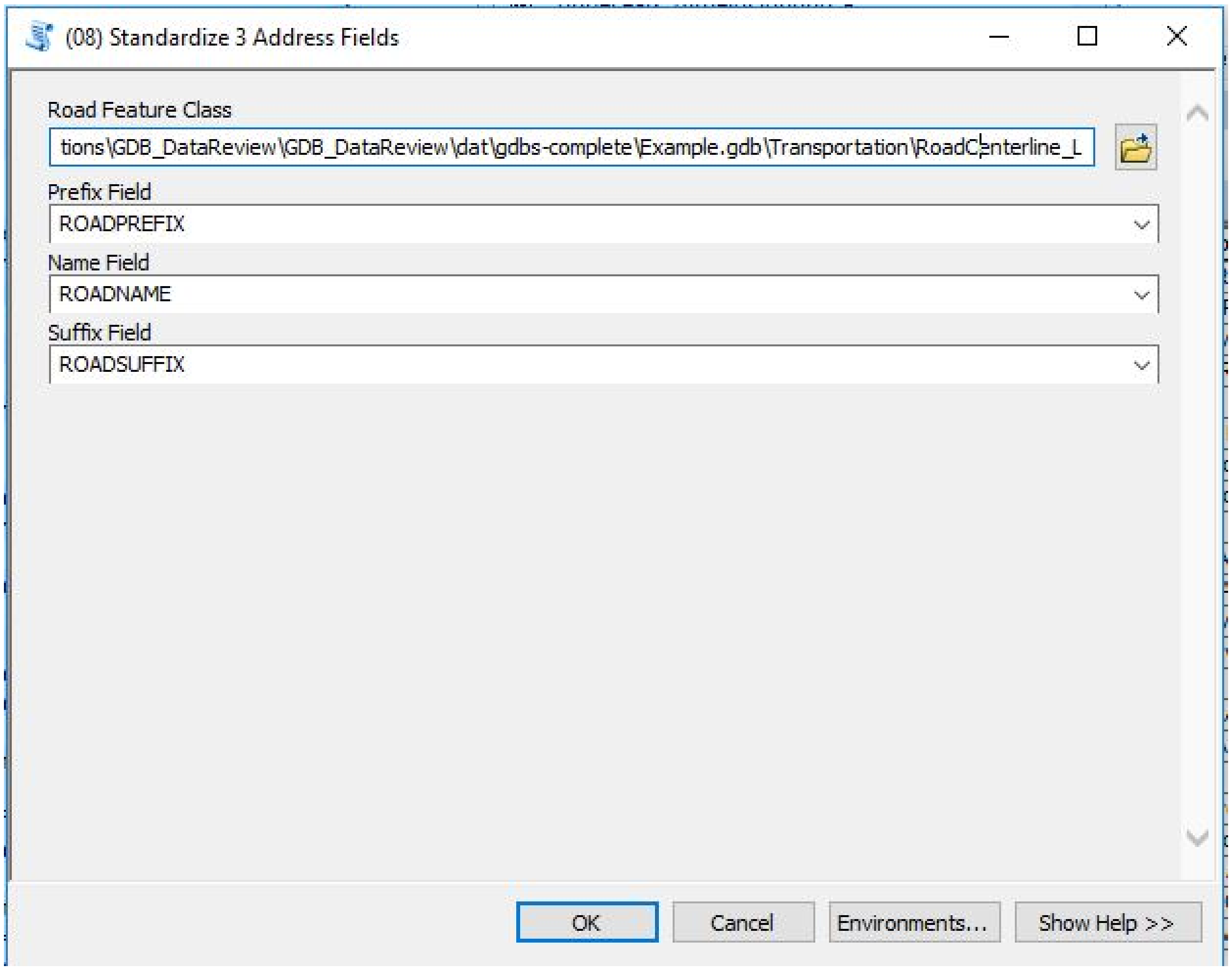

(08) Standardize 3 Address Field
(09) Batch Export Metadata to D
(10) Batch Import Metadata to
(11) Search for Indeterminant D
(12) Summarise Indeterminant
Open...
Batch...
Edit
Deb
Open
Open the tool.
Copy
Delete
Rename
Set Password

Figure 9.1: Opening the Delete Duplicate Features tool

9.3.2 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. 9.2). The fields can be derived directly from the Feature Class by using the drop-down menu.



Road Feature Class
tions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb\Transportation\RoadCenterline_L

Prefix Field
ROADPREFIX

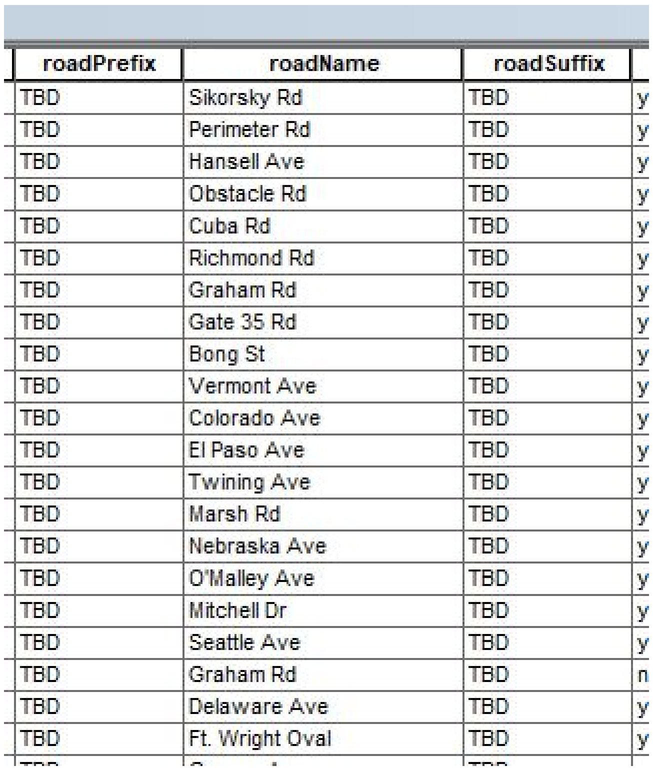
Name Field
ROADNAME

Suffix Field
ROADSUFFIX

Figure 9.2: Opening the Delete Duplicate Features tool

9.4 Run the Tool and View Results

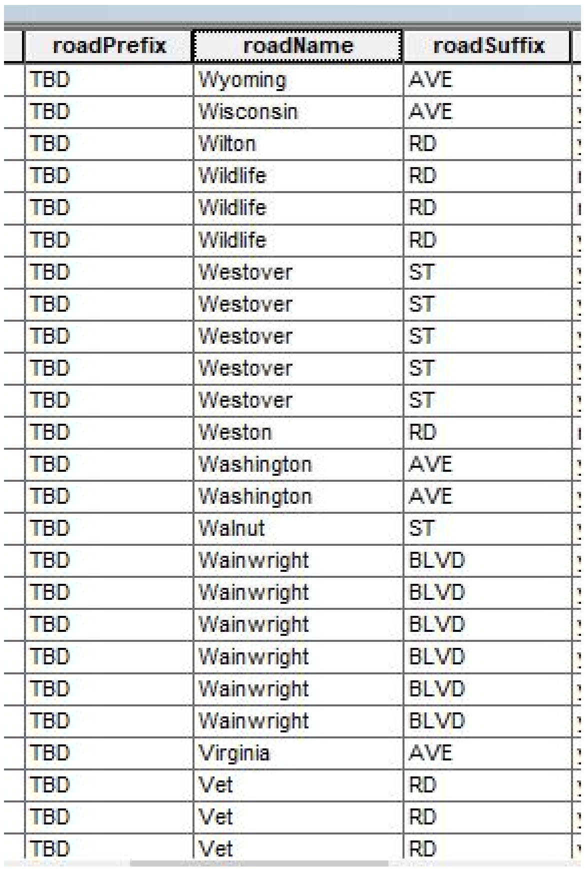
Before running the tool, we see that, indeed, the road prefixes and road suffixes are incorrectly populated inside the road name field Open the destinate Feature Class and view the update destination field values (Fig. 9.3).



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 9.3: Opening the Delete Duplicate Features 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. 9.4).



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 9.4: Opening the Delete Duplicate Features tool

Chapter 10

Search for Missing and Indeterminant Data

10.1 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,

[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

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.

10.2 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.

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 'Search for Indeterminant Data' script tool to open (Fig. 10.1).

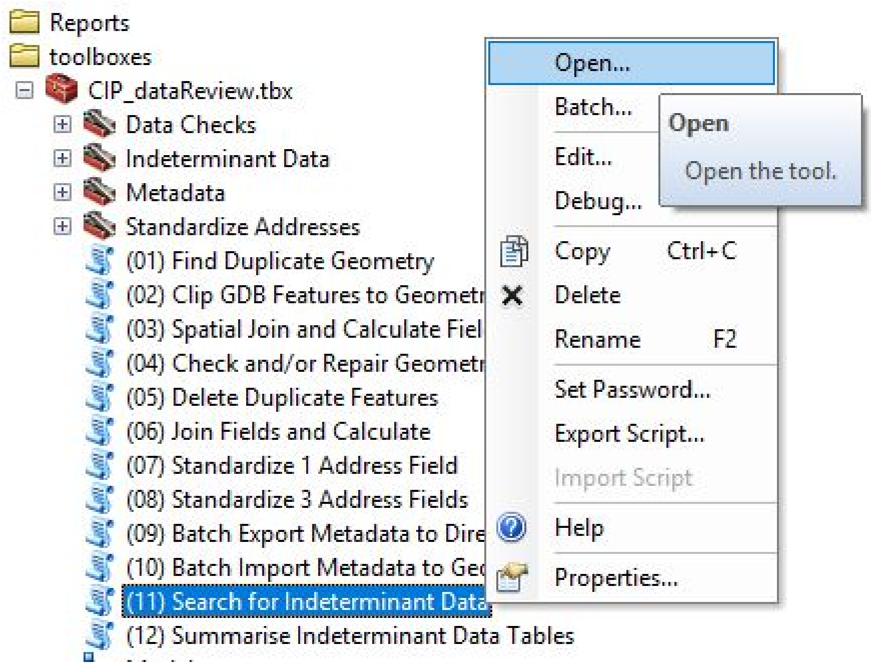
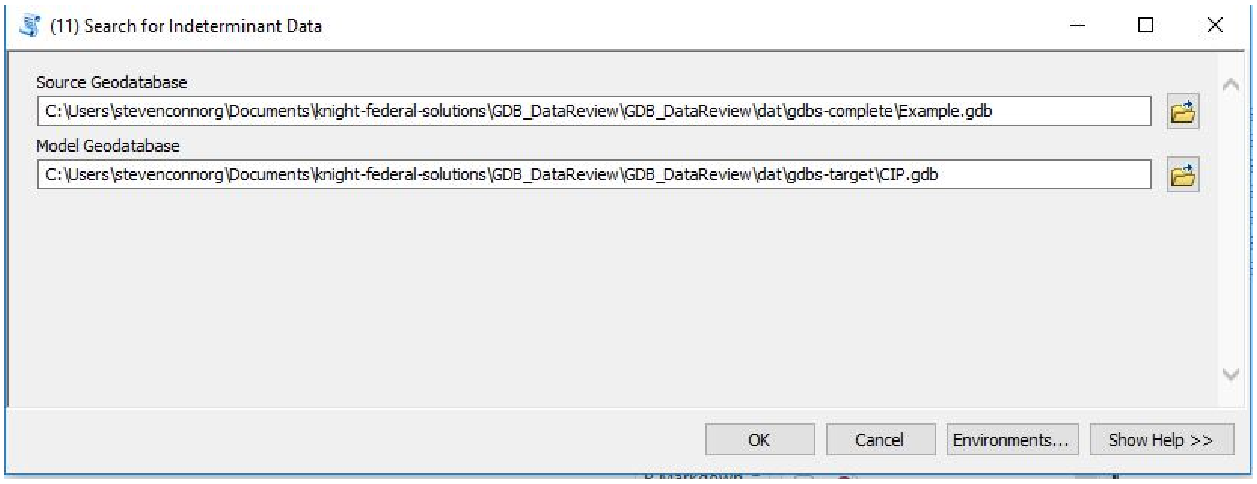

Reports
toolboxes
CIP_dataReview.tbx
Data Checks
Indeterminant Data
Metadata
Standardize Addresses
(01) Find Duplicate Geometry
(02) Clip GDB Features to Geometr
(03) Spatial Join and Calculate Fiel
(04) Check and/or Repair Geometr
(05) Delete Duplicate Features
(06) Join Fields and Calculate
(07) Standardize 1 Address Field
(08) Standardize 3 Address Fields
(09) Batch Export Metadata to Dire
(10) Batch Import Metadata to Ge
(11) Search for Indeterminant Data
(12) Summarise Indeterminant Data Tables
Open...
Batch...
Edit...
Debug...
Copy Ctrl+C
Delete
Rename F2
Set Password...
Export Script...
Import Script
Help
Properties...
Open
Open the tool.

Figure 10.1: Opening the Delete Duplicate Features tool

10.3.2 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. 10.2). The fields can be derived directly from the Feature Class by using the drop-down menu.

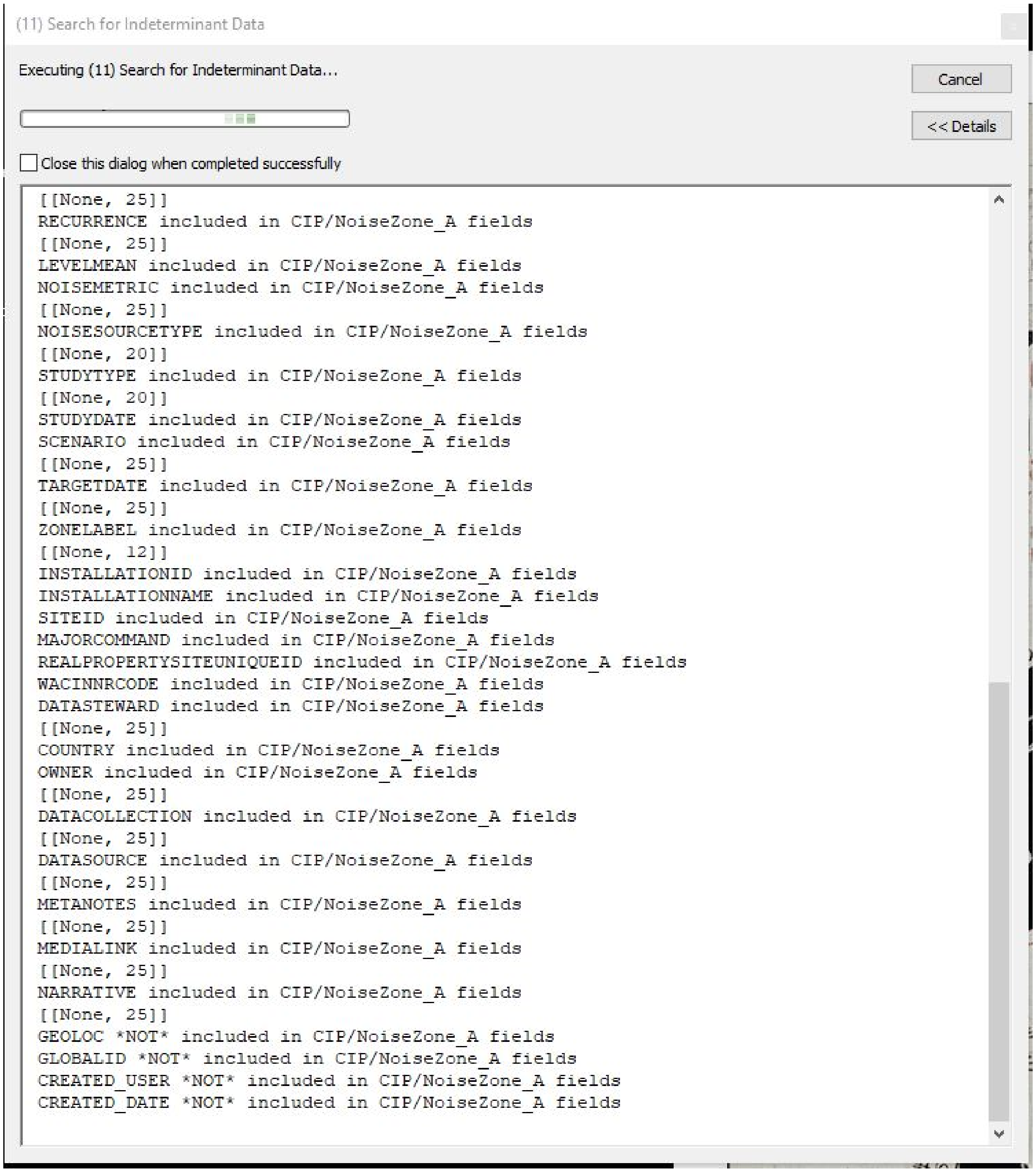


Source Geodatabase
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-complete\Example.gdb
Model Geodatabase
C:\Users\stevenconnorg\Documents\knight-federal-solutions\GDB_DataReview\GDB_DataReview\dat\gdbs-target\CIP.gdb

Figure 10.2: Opening the Delete Duplicate Features tool

10.4 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. 10.3).


(11) Search for Indeterminant Data
Cancel
<< Details
Close this dialog when completed successfully

Executing (11) Search for Indeterminant Data...

```
[[None, 25]]
RECURRENCE included in CIP/NoiseZone_A fields
[[None, 25]]
LEVELMEAN included in CIP/NoiseZone_A fields
NOISEMETRIC included in CIP/NoiseZone_A fields
[[None, 25]]
NOISESOURCETYPE included in CIP/NoiseZone_A fields
[[None, 20]]
STUDYTYPE included in CIP/NoiseZone_A fields
[[None, 20]]
STUDYDATE included in CIP/NoiseZone_A fields
SCENARIO included in CIP/NoiseZone_A fields
[[None, 25]]
TARGETDATE included in CIP/NoiseZone_A fields
[[None, 25]]
ZONELABEL included in CIP/NoiseZone_A fields
[[None, 12]]
INSTALLATIONID included in CIP/NoiseZone_A fields
INSTALLATIONNAME included in CIP/NoiseZone_A fields
SITEID included in CIP/NoiseZone_A fields
MAJORCOMMAND included in CIP/NoiseZone_A fields
REALPROPERTYSITEUNIQUEID included in CIP/NoiseZone_A fields
WACINNRCODE included in CIP/NoiseZone_A fields
DATASTEWARD included in CIP/NoiseZone_A fields
[[None, 25]]
COUNTRY included in CIP/NoiseZone_A fields
OWNER included in CIP/NoiseZone_A fields
[[None, 25]]
DATACOLLECTION included in CIP/NoiseZone_A fields
[[None, 25]]
DATASOURCE included in CIP/NoiseZone_A fields
[[None, 25]]
METANOTES included in CIP/NoiseZone_A fields
[[None, 25]]
MEDIALINK included in CIP/NoiseZone_A fields
[[None, 25]]
NARRATIVE included in CIP/NoiseZone_A fields
[[None, 25]]
GEOLOC *NOT* included in CIP/NoiseZone_A fields
GLOBALID *NOT* included in CIP/NoiseZone_A fields
CREATED_USER *NOT* included in CIP/NoiseZone_A fields
CREATED_DATE *NOT* included in CIP/NoiseZone_A fields
```

Figure 10.3: Opening the Delete Duplicate Features tool

After the tool has run, we can inspect the output tables within the 'Example.gdb' geodatabase (Fig. 10.4). Opening the CIP_MissingFDS table, we see that the Example geodatabase have no missing Feature Datasets 10.5).

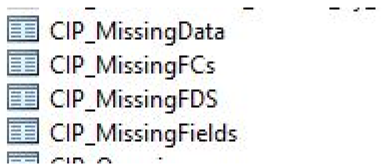
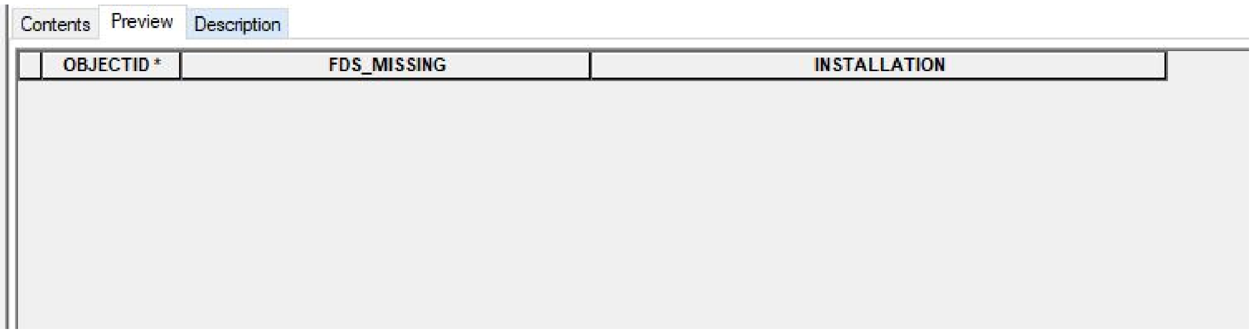


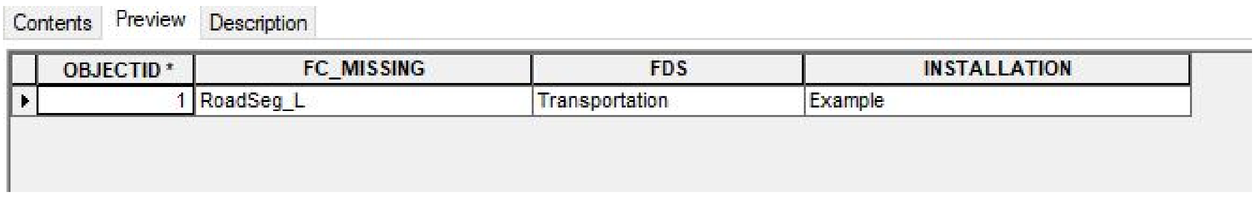
Figure 10.4: Opening the Delete Duplicate Features tool



OBJECTID *	FDS_MISSING	INSTALLATION

Figure 10.5: Opening the Delete Duplicate Features tool

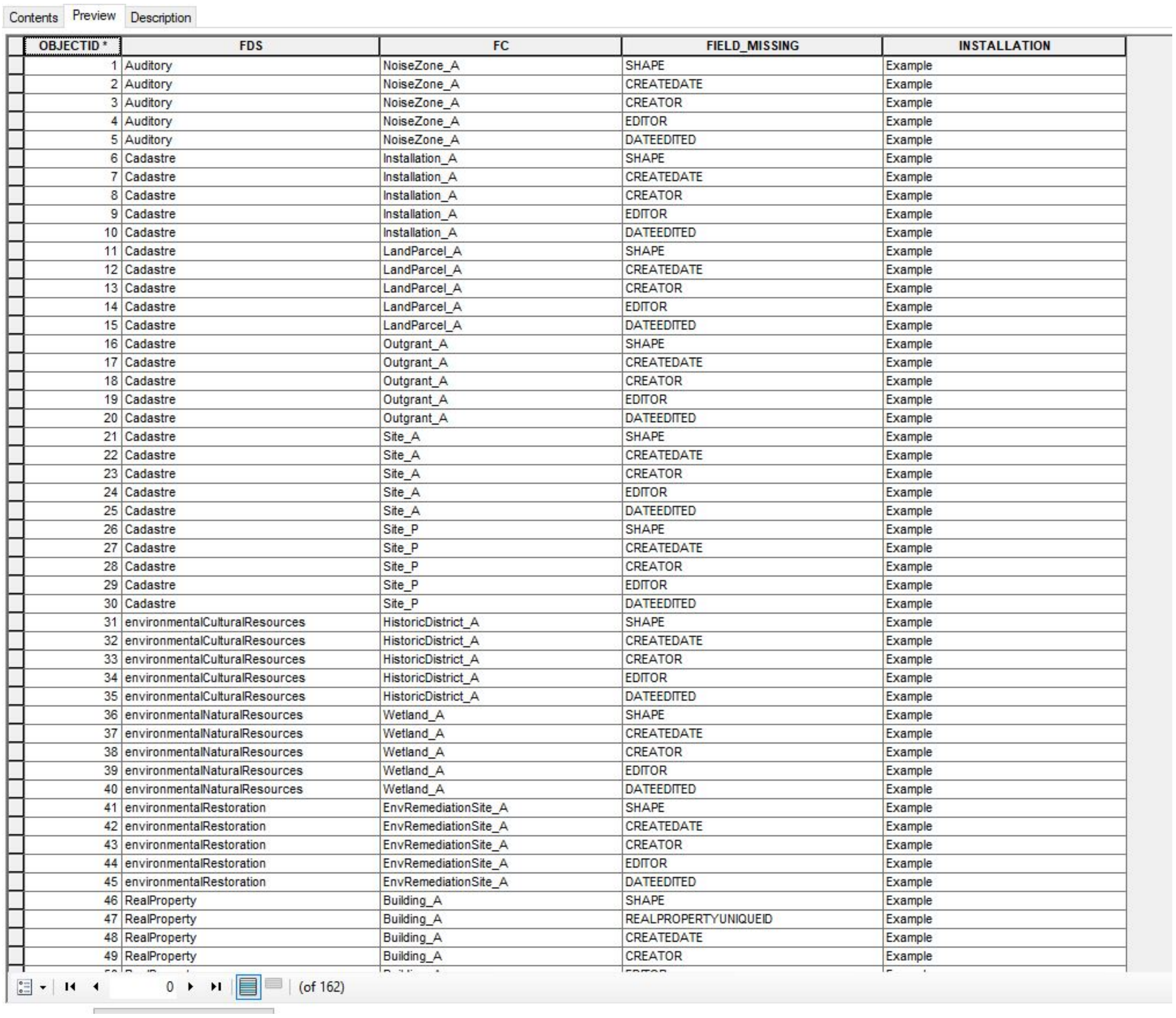
Examining the MissingFCs table, we see that the Example geodatabase has one Feature Class, RoadSeg_L from the Transportation Feature Dataset, missing when compared with the CIP geodatabase 10.6).



OBJECTID *	FC_MISSING	FDS	INSTALLATION
1	RoadSeg_L	Transportation	Example

Figure 10.6: Opening the Delete Duplicate Features tool

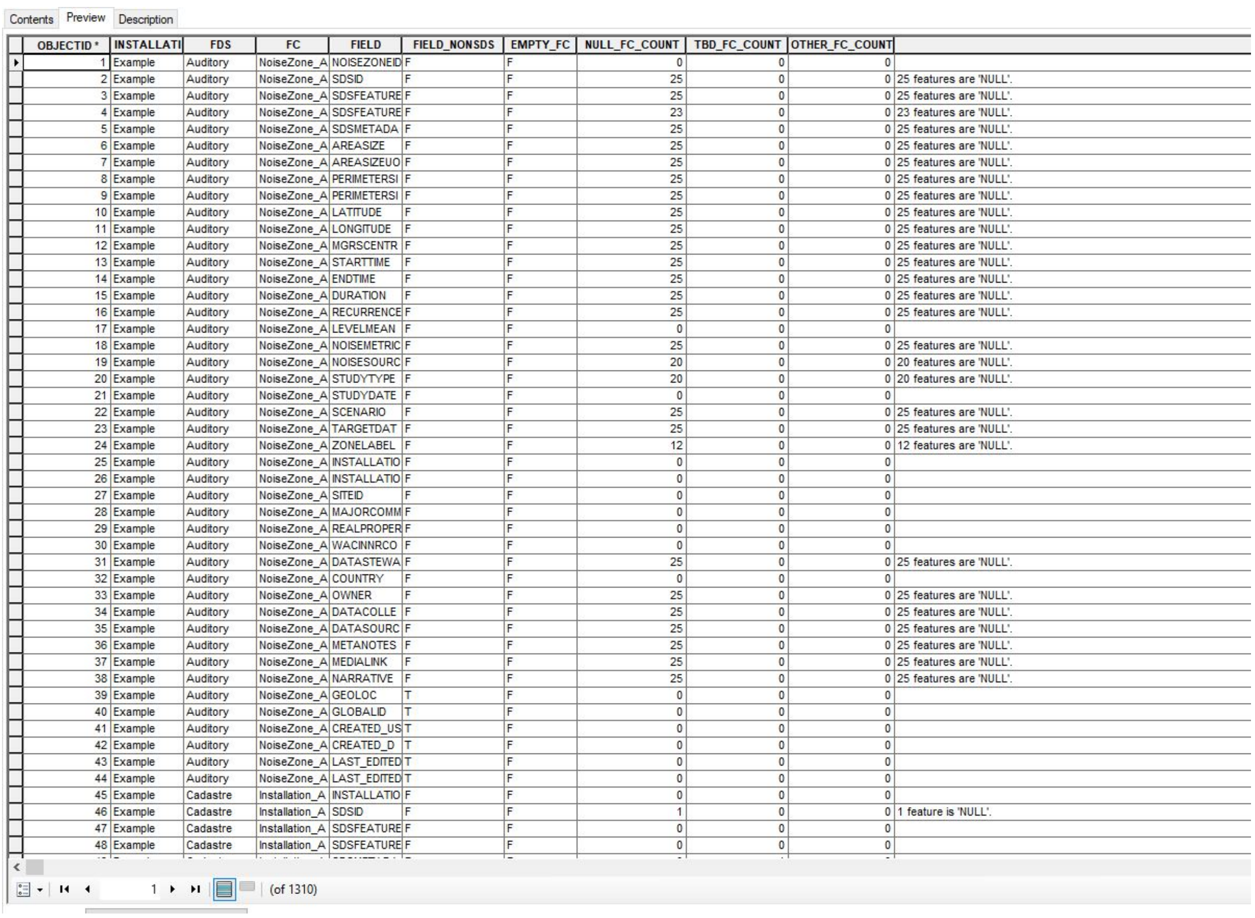
We can look at the MissingFLD table to see which fields are missing from each Feature Class from the target geodatabase that are included in the source geodatabase 10.7).



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

Figure 10.7: Opening the Delete Duplicate Features tool

To examine indeterminant field attribution, we can examine the MissingData table 10.8).



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 10.8: Opening the Delete Duplicate Features tool

Chapter 11

Summarise Indeterminant/Missing Data Tables

11.1 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.

11.2 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.