

Final Report

Version 1 Statewide Parcel Map Database Project

November 18, 2015

CONTENTS

- OVERVIEW 2
- 1 DATA ASSESSMENT METRICS 3
- 2 RECOMMENDATIONS FOR ASSISTING COUNTIES 4
- 3 RECOMMENDATIONS FOR FURTHER DEVELOPMENT 5

- APPENDICES 10
 - A. Address Parsing Tool Guide..... 11
 - B. DOR XML Parse Tool Guide 24
 - C. Data Standardize Tool Guide..... 29
 - D. Condo Stack Tool Guide..... 36

- TO RETURN TO TABLE OF CONTENTS [Click ↻](#)

Wisconsin State Cartographer’s Office
384 Science Hall
550 North Park Street
Madison, WI 53706-1491
608-262-3065
sco@wisc.edu
www.sco.wisc.edu

Wisconsin Department of Administration
Wisconsin Land Information Program
101 East Wilson Street
Madison, WI 53703
608-267-3369
wlip@wisconsin.gov
www.doa.state.wi.us/WLIP

OVERVIEW

The Version 1 Statewide Parcel Map Database Project (V1 Project) was a collaboration between the State Cartographer's Office and the Wisconsin Land Information Program (WLIP). This report describes the V1 Project, part of the Statewide Parcel Map Initiative established by Act 20 of 2013. The primary project objectives of V1 were to:

- Establish a statewide parcel GIS map layer by integrating county-level datasets
- Recommend a searchable format for parcel attributes for V2 and beyond
- Make recommendations on WLIP Strategic Initiative grants for parcel mapping activities in the form of "benchmarks" for parcel dataset development

The V1 Project successfully aggregated all known digital parcel datasets within the state. The resulting statewide GIS parcel layer totaling 3.34 million parcels was made publically available online on July 31, 2015.

Differences Between Interim and Final Reports

Interim Report Scope

- Recommendation for a standard searchable format for parcel attributes as required by s. 59.72(2)(a)
- Identification of metrics to assess completeness of digital parcel attributes needed to meet s. 59.72(2)(a) requirement
- Identification of metrics to assess quality and completeness of county parcel datasets *beyond* s. 59.72(2)(a) requirement
- Quantitative assessment of each county parcel dataset
- Recommendations for assisting counties to improve quality and completeness of parcel datasets by achieving benchmarks
- Description of tools used to standardize individual county and municipal datasets
- Recommendation of a preferred county data export standard for integration into statewide parcel database

Final Report Scope

- Sustainability recommendations: How to maintain and update the statewide parcel database
- Data submittal recommendations: How to automate, streamline, and improve the submission of local datasets and updates to those datasets
- Other recommendations from V1 Project lessons learned
- Tools to Aid Benchmark Achievement
 1. Address Parsing Tool – Parse site addresses into sub-address elements
 2. DOR XML Parse Tool – Translate Wisconsin Department of Revenue Tax Roll XML into a GIS table
 3. Data Standardize Tool – Standardize file geodatabase feature class data via the creation of a lookup table
 4. Condo Stack Tool – Model condos by stacking condo parcel geometries by owner

1 DATA ASSESSMENT METRICS

1.1 Metrics to Assess

1.1.1 Identification of metrics to assess quality and completeness of county parcel datasets

In order to identify metrics to assess quality and completeness of county parcel datasets, benchmarks were established in the *V1 Interim Report*:

- Benchmark 1 – Parcel and Zoning Data Submission
- Benchmark 2 – Extended Parcel Attribute Set Submission
- Benchmark 3 – Completion of County Parcel Fabric
- Benchmark 4 – Completion and Integration of PLSS

The Department of Administration adopted these benchmarks as part of the 2016 WLIP grant application. The benchmarks will thereby set some of the terms of Strategic Initiative grant agreements.

1.2 Quantitative Assessment of County Data

1.2.1 Quantitative assessment of each county parcel dataset according to metrics identified to assess quality and completeness

Quantitative assessment of county data for V1 was reported in the *V1 Interim Report*. In keeping with the goal of incremental improvement, the V1 Project was a vehicle through which recommendations for future versions of the statewide parcel map were produced. Achievement of Benchmarks 1, 2, and 3 will be assessed via the data contributed to the statewide parcel project. Each county will be asked to submit parcel datasets on an annual basis, through at least Version 4 of the statewide parcel map (to be completed in 2018).

This quantitative assessment will take two forms for V2 Project:

- **County land information plans:** the county land information plan must identify the layer status of its parcel mapping and PLSS in the Foundational Elements chapter of the plan. If the county has not yet achieved the parcel benchmarks, it is required to include project plans for achieving them in its county land information plan.
 - Project Plan to Achieve Searchable Format (Benchmarks 1 & 2)
 - Project Plan for Parcel Completion (Benchmark 3)
 - Project Plan for PLSS (Benchmark 4)
- **Data ingest and verification system:** The county will need to fill out a data submission form and report to what standard it is submitting its data for V2, whether that be the searchable, export, or other (not meeting either standard). SCO staff will then review data for achievement of the stated standard.

As “completion and integration” of PLSS varies at the local level according to business needs, topography, and other factors, the judgement of whether a county has achieved Benchmark 4 will be left to county land information councils. However, the *V1 Interim Report* does provide some guidance, in the form of guidelines and standards to assess whether a county’s PLSS has reached a stage of satisfactory completion.

2 RECOMMENDATIONS FOR ASSISTING COUNTIES

2.1 Recommendations for assisting counties to improve quality and completeness of parcel datasets

The *V1 Interim Report* features recommendations for assisting counties to improve quality and completeness of parcel datasets, by way of 2016 and 2017 WLIP Strategic Initiative grants. Strategic Initiative funds are an investment to assist counties to achieve the benchmarks. Counties must prioritize the use of Strategic Initiative funds toward achieving the parcel benchmarks.

Beyond providing grant funding to assist local governments, on the technical side, four tools were developed to assist counties to meet the benchmarks.

- *Address Parsing Tool* – Parse site addresses into sub-address elements
- *DOR XML Parse Tool* – Translate Wisconsin Department of Revenue Tax Roll XML into a GIS table
- *Data Standardize Tool* – Standardize file geodatabase feature class data via the creation of a lookup table
- *Condo Stack Tool* – Model condos by stacking condo parcel geometries by owner

A webpage was created where users could download the packaged files for installing the tools, as well as PDF guides and YouTube video tutorials covering tool installation and execution.

Tool guides are included as Appendices to this report. The tool packages are available for download at <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>

3 RECOMMENDATIONS FOR FURTHER DEVELOPMENT

3.1 Sustainability Recommendations

Throughout the design and development of the V1 Project as well as the LinkWISCONSIN Address Point and Parcel Mapping Project, the project team researched long-term solutions for efficiently sustaining parcel and zoning layers. How to maintain and update the statewide parcel database involves supplanting the parcel aggregator role with a steward role, the implementation and enforcement of standards, and the provision of solutions to county contributors.

3.1.1 Supplanting the Aggregator Role with a Steward Role

For the V2 Project, staff at the State Cartographer's Office partnered with the DOA as a consultant under a Memorandum of Understanding. Part of SCO's role in the project is to collect, translate, standardize, and aggregate data from local contributors, in what is considered an "aggregator" role. The aggregator role is relatively expensive and time consuming when compared to its alternative, the "steward" role.

Aggregator of Statewide Data	Steward of Statewide Data
Collects data from local contributors	Local contributors submit data
Translates and standardizes data	Local contributors translate and standardize data
Aggregates data from local contributors	Aggregation is automated, according to a contributor model

The Parcel Initiative was designed to phase out the aggregator role over the next two cycles of development and instead be replaced with role of a "steward" which will be less costly and time consuming. The key to achieving this transition lies in:

- The benchmarks and standards implemented as a part of the V1 Project.
- The adoption of a "contributor model" for collecting, homogenizing, and aggregating data (discussed below).
- Long-term availability of a "steward" for the project. The person(s) in this role will oversee the automated contributor model, provide troubleshooting when necessary, continue to research and implement forward-looking procedures, and make alterations to the model as necessary. The steward would also be involved in various stewardship tasks such as interactive mapping, data distribution, and community feedback.

3.1.2 Standards

A high degree of emphasis must continue to be placed on clear and thoughtful standards for data contributors to meet, as well as clear and thoughtful documentation of submission file formatting. Without homogenizing the data submitted through these standards, the contributor model will not be successful. Thus, standards like those of the benchmarks outlined in the 2016 WLIP grant application are essential to future success of the Parcel Initiative.

3.1.3 The Provision of Solutions to County Contributors

Continuing to provide or support solutions, such as the tools to aid benchmark achievement included in this report, will be beneficial in assisting counties to meet the standards and file formats required for successful submission under the contributor model. Supporting these solutions would ideally be a function of the steward. In addition, as technology changes, new solutions should be considered and developed as appropriate.

3.2 Data Submittal Recommendations

The project team also considered how to automate, streamline, or improve the submission of local datasets and updates to those datasets. Here too, the contributor model offers an attractive solution, due to its automation and streamlining of data submission. At the same time, it is important to recognize that an increased onus is placed on the data contributor to ensure their data meets required formats and standards.

3.2.1 Aggregator Versus Contributor Models

The V1 Project followed an aggregator model, as illustrated in the table below and Figure 1.

Parcel Aggregation Models	
Contributor Model	Aggregator Model
<ul style="list-style-type: none"> An interface to which local-level parcel data stewards can independently field-map and aggregate their parcel data to a statewide parcel feature service 	<ul style="list-style-type: none"> An interface to which an authoritative entity (such as the State Cartographer's Office) can field map, aggregate, and upload local-level parcel data into a statewide parcel feature service The model for the V1 and V2 projects, to be phased-out in future development cycles

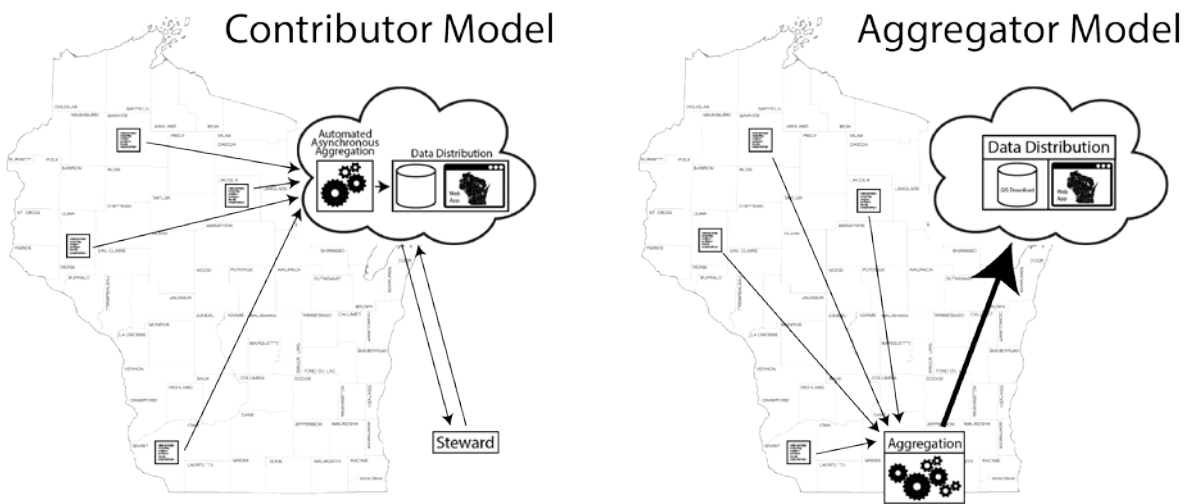


Figure 1. The distinction between the two models lies in where aggregation takes place and who homogenizes the data in preparation for aggregation

Under the aggregator model, the aggregator (the project team) took the role of coordinating, standardizing and formatting the data in preparation for aggregation. The contributor model is grounded on the contributor, who is responsible for achieving this data homogenization.

The contributor model stipulates that aggregation happens on a server through an automated process. Because the process is highly automated, there is no direct involvement of the steward in aggregation of the data. This type of automation is only made possible through coordinated, standardized, and well-formatted data submissions from contributors.

The contributor model features a web interface for contributors to submit their authoritative data. Through the web interface, the user is able to manually configure a data submission. The web interface can also support stored submission routines for further automation. Once a contributor submits data, the submitted data asynchronously replaces the contributor's previously existing data in its entirety. As this is an automated and asynchronous process, a data contribution could be performed at any given time and at any interval.

Role of the Steward of Statewide Data

It is anticipated that the V2 Project will continue to accommodate an aggregator model approach, but the aggregator approach will be phased out as the Parcel Initiative progresses. However, there will still be a need for a central steward of the statewide parcel map. A steward will be necessary to monitor whether counties are meeting the requirements for data submittal, modernize the software and applications used, market the statewide layer to potential users, and to continue to improve and enhance the statewide layer.

3.2.2 The Four A's – Authoritative Automated Asynchronous Aggregation

The contributor model offers benefits of sustainability, efficiency, and longevity by supporting authoritative automated asynchronous automation. These “four A’s” support the submission of individual datasets at any time or interval by county data stewards by automatically merging the local data with the most current statewide database.

3.2.3 Contributor Model Options

The contributor model could be implemented through various technical means. Three technical solutions were assessed for feasibility: Esri’s Community Parcels, a custom/hybrid solution, and partnering with the Wisconsin State Legislature’s Legislative Technology Services Bureau for GIS data hosting and display.

Community Parcels

Community Parcels is an Esri solution, workflow, and set of tools designed to serve as an out-of-the-box mechanism for executing a contributor model of parcel aggregation. Community Parcels offers an ArcGIS Desktop-based interface to which local-level parcel data contributors can independently field-map and aggregate their parcel data to a statewide parcel feature service in ArcGIS Online. The solution is designed to achieve three general tasks:

- *Aggregate Data.* Community Parcels is designed to achieve local to state data aggregation through field mapping local dataset attributes to a common attribute schema within a desktop tool. By default, the common attribute schema is defined as the “Community Parcels Attribute Schema,” but this can be tailored to any degree.
- *Upload to ArcGIS Online.* Community Parcels is designed to achieve data upload to ArcGIS Online automatically through the desktop tools provided. They are also designed to upload only the new data (asynchronous update), leaving non-updated jurisdictions as-is on ArcGIS Online.
- *Automate Procedures.* The solution can be designed to support the ability to be executed from ArcMap, ArcCatalog, or the Windows Task Scheduler on an automatically defined schedule.

Community Parcels requires that data has already been appropriately “staged” for aggregation. This means that all of the necessary parsing, normalizing, and QA/QC must be complete on the local level before beginning any part of the above models. The community parcels toolset does not accomplish any degree of database staging, this will need to be performed before any dataset would fit the Parcel Initiative data model.

Custom/Hybrid Solution

The V1 project team also considered a custom solution for statewide parcel aggregation which would follow a hybrid of the contributor and aggregator models. There are two parts to the hybrid model:

- *ArcGIS Desktop Tool.* An ArcGIS Desktop tool would be developed that would be distributed to all county and municipal data contributors. This tool would conform all data submissions to the Parcel Initiative’s data model through user configuration. Like all three technical solutions described in this section, this process would require the input data to be in the searchable format. The tool would validate the submission, provide error reports as necessary, and then upload the data to a data repository as a simple zipped file.
- *A Python Tool.* The tool would be administered by the Parcel Initiative steward, and would be engaged and executed each time there is a new data submission to the data repository. This tool would automatically delete the old and merge-in the new data asynchronously.

This technical solution can be classified as a hybrid model because the data aggregation actually happens on the data Parcel Initiative steward’s local machine. The process on the steward’s machine would be automated and should not require manual processing, but would likely require occasional troubleshooting.

Benefits of the hybrid model include:

- The technical process does not rely on a geospatially-enabled server such as ArcGIS Online (Community Parcels) or ArcGIS Server (LTSB), which are required of the other two solutions.
- This model would be dependent on the Parcel Initiative steward’s computing environment, while the other two require dependence on ArcGIS Online or ArcGIS Server. This is beneficial as a lower degree of complexity and cost.

The hybrid solution also entails some challenges:

- *No Precedents.* No logic/code has been created to implement this to this point, while the other two solutions are pre-existing and have a pre-existing support.
- *Relies on Steward's Local Computing Environment.* This process is not independent of the steward's local computing environment. The aggregation will take place on a local machine, which as a result has more of an impact on the steward's role. A merge tool would run in the background on the steward's machine or machines, a process not exclusive to one single steward.

Partnering with the Legislative Technology Services Bureau

In searching for cost-effective and efficient solutions to hosting and display of statewide parcel data, the project team considered other entities who engage in collection or display of statewide GIS data. One of those organizations is the Legislative Technology Services Bureau (LTSB) of the Wisconsin State Legislature.

There are two possible avenues for future collaboration:

- LTSB hosting of the statewide layer (this is currently designed to be implemented).
- LTSB collecting parcel data for future iterations of the Parcel Initiative

The option of LTSB hosting and displaying the parcel layer is attractive for a few key reasons:

- *Preferred alternative to ArcGIS Online.* The ArcGIS Online option of hosting the statewide parcel layer was not effective. Initial attempts to publish the statewide parcel layer through AGOL failed due to the existence of parcel features with geometry errors in the statewide feature class. In troubleshooting this problem, the project team worked closely with Esri Customer Support. Esri Support was able to determine that if there were geometry errors in a feature class being uploaded to AGOL and used to create a feature layer, then a random selection of features would be missing from the resulting feature layer.¹
- *GIS staff support.* LTSB has full-time staff dedicated to GIS data. The LTSB GIS team is responsible for providing technical support to the Legislature for legislative redistricting, voting information, district maps, census tract maps, and district boundary information related to both the legislature and the public.
- *Achieve lean government objectives through efficiencies.* LTSB has developed a proven process that can be expanded to meet the needs of the Parcel Initiative with low relative investment, as the core tools and procedures have already been developed.
- *No cost solution.* In the spirit of working towards government efficiency, transparency, and data utilization, the LTSB has offered the hosting solution at no cost, presuming that administrative work would still be the onus of the Parcel Initiative steward and dependent on future elaboration of roles.
- *History of previous success with counties with WISE-Decade.* LTSB released the WISE-Decade CBAS/Data Collection online tool, which offers an intuitive interface and workflow. The tool received positive feedback from counties. Further, integrating additional data submission in the same interface and web location further improves efficiency and intuitiveness of data submissions.
- *Data hosting and aggregation in one place.* It is likely that the LTSB will be the home to the feature service used for hosting of the statewide layer for the foreseeable future. It is an advantage to keep the submission, aggregation, and hosting logic under the same general roof.

Before moving forward with the LTSB partnership, it will be required for the LTSB to vet this partnership with their legislative representatives, from both an administrative and political perspective. If arrangements can be made, it is likely a collaboration will take place between the DOA/SCO and LTSB for this partnership.

¹ Regarding the obstacle of geometric errors, Esri Customer support and the ArcGIS Online team elected not to log a bug with ArcGIS Online. ArcGIS Online requires data to be free of geometric errors and it cannot be expected that ArcGIS Online would behave correctly if these errors exist. However, ArcGIS Customer Support determined that it can be expected that the "Repair Geometry" tool (a commonly used tool provide with ArcGIS Desktop designed for this exact purpose) should catch and fix these errors. As a result, Esri Customer Support logged a bug ("BUG-000090827 – Repair Geometry does not identify (or fix) all geometry errors which are identified by Check Geometry"), which will be beneficial to continue to monitor.

Recommended Solution

The custom/hybrid solution has been employed to host the V1 statewide layer to date. When considering the three options outlined above for the future, the SCO recommends pursuing the partnership with the LTSB as the best technical solution. The other two solutions are viable as backup options should the partnership with LTSB be unavailable.

3.3 Other Recommendations from V1 Project Lessons Learned

3.3.1 Tax Roll Cycle Timing

Parcel geometries and tax roll data are maintained and updated on different cycles at the local-level and are slightly incongruent due to this difference in timing. A finalized tax roll will only match one-to-one with a historic version of the parcel map.

Tax roll data (including assessed value, estimated fair market value, total property taxes) for a given year is based on parcels as they existed as of January 1st on any given year. It takes the whole year to reevaluate and assess the taxes for parcels. The tax roll data is not finalized until the end of the year. For example, 2015 tax roll data is according to parcels as they existed on January 1, 2015, but the 2015 tax roll data will not be available for the statewide parcel layer integration until early 2016.

Local parcel geometries are updated on various schedules in counties throughout the state, ranging from daily to annually. According to the *2014 WLIP Report*², about 24 of 72 counties have an update cycle of quarterly or more. For those that update annually, parcel update occurs at the beginning of the year in order to provide assessors with updated maps.

Monthly Update Cycle for Parcel Geometries

It is recommended that tax data required for the Parcel Initiative be submitted during the first quarter of each year, as this will be the point at which the tax roll data and parcel geometries will be most current, although they will not be completely congruent.

3.4 Layer Usage

Within the first week, the file geodatabase was downloaded over 175 times. Other usage and downloads statistics appear below.

Item	Usage	Date
Database		
V1_Wisconsin_Parcels_2015_3	776 downloads	As of Nov 12, 2015
V1.0.1_Wisconsin_Parcels_2015.gdb	408 downloads	As of Nov 12, 2015
V1.0.1_Wisconsin_Parcels_2015_uncompressed.gdb	37 downloads	As of Nov 12, 2015
V1.0.1_Wisconsin_Parcels_2015_ESRI9.2.gdb	43 downloads	As of Nov 12, 2015
	1,264 downloads	
	of database	
Web Mapping Application		
Wisconsin Statewide Parcels Version 1	6,616 views	As of Nov 12, 2015
	of web mapping application	

² http://www.sco.wisc.edu/images/stories/publications/2014_WLIP_Report_09.24.14.pdf

APPENDICES

Address Parsing Tool Guide

The *Address Parsing Tool* parses a full site address field into individual elements. It will parse any address contained as a field in a file geodatabase feature class into individual elements that fit the FGDC’s *United States Thoroughfare, Landmark, and Postal Address Data Standard*.¹

Use this guide if your county’s parcel site address data is not available as fully parsed address elements meeting the Parcel Schema for V2, Appendix F of the *V1 Interim Report*.

When to Use Address Parsing Tool		
	Benchmark 1 Parcel and Zoning Data Submission	Benchmark 2 Extended Parcel Attribute Set Submission with Parsed Address Elements
EXPORT FORMAT OPTION	Parsed address elements not required	✓ This tool can assist
SEARCHABLE FORMAT OPTION	✓ This tool can assist	✓ This tool can assist

The tool, which is run in several cycles, is designed to accomplish as much automatic parsing as feasible. However, for some parcel feature’s addresses, the workflow may require altering an input site address or manual parsing.

Why is This Guide Necessary?

The attribute schema of the statewide parcel layer requires parcel site addresses and sub-address elements to be modeled according to FGDC and other standards.

For those counties who wish to use Esri’s ArcGIS geocoding tool called “Standardize Addresses,” be aware that the tools from SCO—the *Address Parsing Tool*, and the *Data Standardize Tool*—were created to overcome some of the weaknesses in the Esri tool:

- The Esri tool does not parse to the FGDC addressing standard but rather to Esri address locators
- Highways are intermingled with street names (when they should be included as street prefixes)
- Some prefix directions are incorrectly included as street names
- Some street names are incorrectly included as sub address type (unit type)
- Some street names and street types are incorrectly included as sub address unit (unit ID)
- Some street names split incorrectly and are distributed to various incorrect elements
- Grid addresses numbers are not parsed according to the FGDC standard

The *Address Parsing Tool* will parse addresses but achieves limited element standardization. To meet the standardization requirements of the searchable format, users are encouraged to employ the *Standardize Tool*.

How to Use This Guide

It is recommended you have the following materials at hand:

- The *Version 1 Statewide Parcel Map Database Project Interim Report* and Digital Appendices
- Your GIS parcel data – Must be in file geodatabase feature class form but can be of any geometry type (point, line, polygon)
- The county staff member(s) who will submit the data for V2
- The *Address Parsing Tool* from www.sco.wisc.edu/projects/v2-parcels.html

¹ https://www.fgdc.gov/standards/projects/FGDC-standards-projects/street-address/index_html

Process Overview

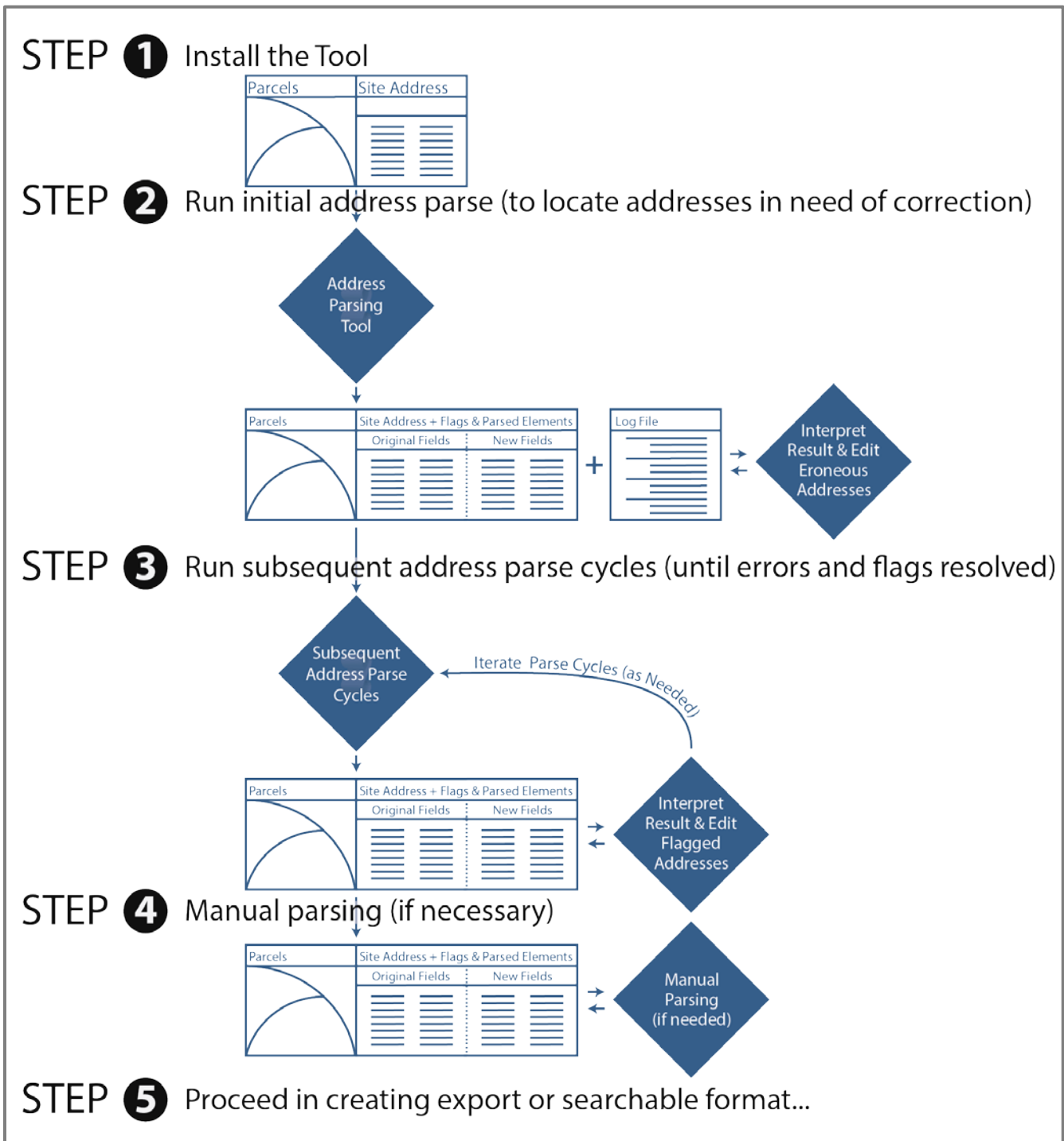


Figure 1. Address Parsing Tool workflow overview

1 Install the Tool

1.1 Download and unzip the tool

Download the zipped package with the *Address Parsing Tool*, an ArcPy script tool, from <http://www.sco.wisc.edu/images/stories/publications/V2/tools>.

Unzip to the directory of your choice. Then open ArcCatalog and navigate to the new directory. You should see toolboxes, labeled with their respective ArcGIS version compatibility. Choose the toolbox that fits your ArcGIS install. If you do not see any toolboxes, hit F5 to refresh the directory. Once the tools are visible, move on to the next step.

1.2 Including Python library dependencies

The tool will not be ready to run immediately. Instead, a directory of Python library files need to be placed within your ArcGIS 10.X Python package. Note that copying files to this directory may require administrative privileges to the machine.

To install the files:

- Navigate to the directory the tools were unpacked in. Included within the .zip file is a directory called */libraries/*.
- Copy the **contents** of the */libraries/* directory.
- Navigate to your ArcGIS 10.X install of Python27. The full path to this directory may vary depending upon where ArcGIS is installed. A typical path to the Python27 install will look like this:
 - **C:\Python27\ArcGIS10.3**
 - If you cannot find the path to your ArcGIS 10.X install of Python27, from ArcCatalog, open the Python Window. Within the Python Window, enter:
 - **import re** (and hit enter)
 - **re.__file__** (and hit enter)

Figure 2 illustrates what this would look like in the ArcCatalog Python Window.

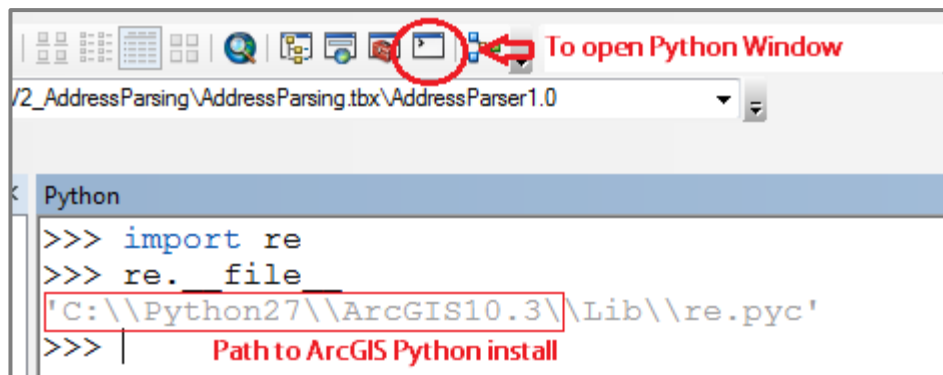


Figure 2. Locating Path to ArcGIS 10.X Install of Python27

- Once the path to Python27 is identified, navigate to this path, making sure to follow the path of the ArcGIS version that you are currently running. Navigate to the *\\Lib\\site-packages* sub-directory.
 - For example: **C:\\Python27\\ArcGIS10.3\\Lib\\site-packages**
 - Copy and paste the contents of */libraries/* from the .zip file into this directory

The tool should now be ready to run. If you experience trouble starting or executing the tool, see the Troubleshooting section below.

1.3 Tool overview

The *Address Parsing Tool* is designed to parse a site address within a file geodatabase feature class of type point, polygon, or line. The tool has the ability to create:

- A copy of the input feature class with:
 - New address fields containing the address elements parsed by the tool.
 - Address Flag Fields containing information to identify poorly constructed addresses. Note that new flag and address fields will be cleared and overwritten when subsequent parses are run over a previous tool's output feature class.
- An **Error Log** as output text file, containing information on addresses the tool was unable to parse.

As some outputs are optional, choosing to exclude optional elements may allow the tool to execute faster. Figure 3 depicts the tool's parameters.

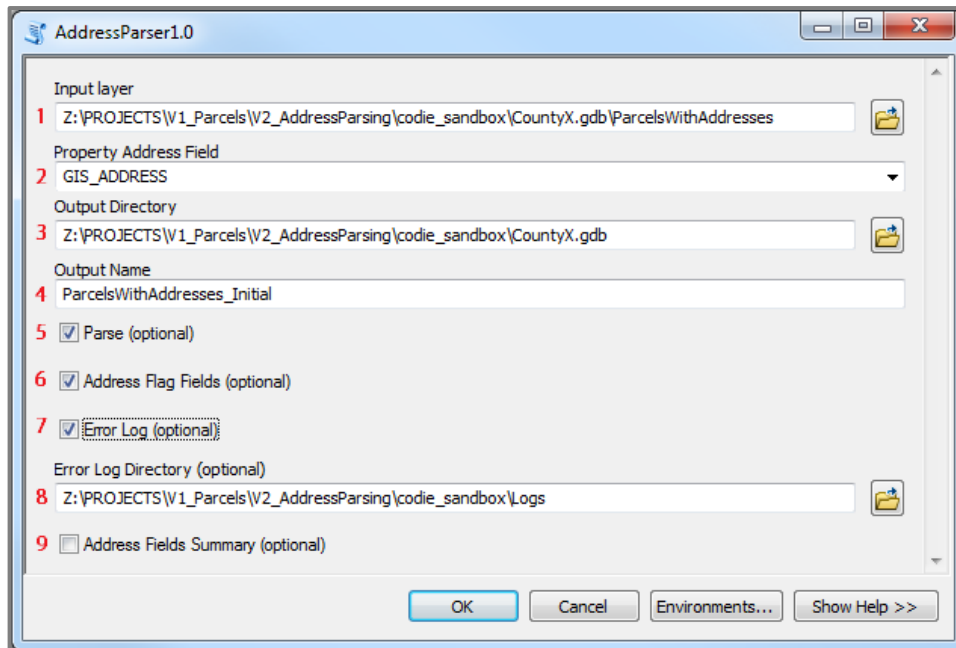


Figure 3. Address Parsing Tool parameters

Address Parsing Tool Parameter Definitions

1	Input Layer	The feature class containing the site address field to be parsed.
2	Property Address Field	The field containing the site address.
3	Output Directory	The path specified for the output feature class. This must be a path to a file geodatabase, or file geodatabase feature dataset.
4	Output Name	The name given to the output feature class and error log file (if applicable).
5	Parse (Optional)	<p>Optional functionality to parse the site address field into the appropriate address elements. By default, this field is checked.</p> <p>Parse adds these fields to the feature class:</p> <ul style="list-style-type: none"> ● NEW_ADDNUMPREF ● NEW_ADDNUM ● NEW_ADDNUMSUFF ● NEW_PREFIX ● NEW_STREETNAME ● NEW_STREETTYPE ● NEW_SUFFIX ● NEW_UNITTYPE ● NEW_UNITID <p>Consult the <i>Address Reference Guide</i> for definitions of address fields that are added to the feature class.</p>
6	Address Flag Fields (Optional)	<p>Optional functionality to create additional fields in the output feature class containing detail on potential poorly constructed addresses. By default, this field is left unchecked.</p> <p>Address Flag Fields generates the following:</p> <ul style="list-style-type: none"> ● Character_Flag: Any characters found in the site address that are not found in properly constructed addresses. Some characters flagged in this field are acceptable depending on the location of the character within the address. For example, hyphens in an address number range (100-110 Main St.) are unacceptable, but in a unit ID range (100 Main St. Apt 1-10) they are acceptable. See the <i>Address Reference Guide</i> for specific information regarding the character in question. ● Incomplete_Data_Flag: An indication of potential missing address elements based on the notion every address must have an address number and street name. Valid addresses must contain these elements. See the <i>Address Reference Guide</i> for more information on required address elements. ● Parse_Error_Flag: A numeric field indicating parse error with a "1" or a "0" / Null for no parse error. ● Extraneous_Data: Any elements not parsed to schema address elements are dumped into this field. The field often contains data not included in a valid address and can be removed from the site address altogether. However, there are cases where an address will be parsed, but parsed incorrectly placing some elements in this field. See the <i>Address Reference Guide</i> for more information regarding extraneous address data.
7	Error Log (Optional)	Optional functionality to output a text file containing information on addresses the tool was unable to parse. By default, this option is unchecked.
8	Error Log Directory (Optional)	The path for the output error log file. This field only needs to be completed if the Error Log checkbox is selected. If left blank, the default directory is the ArcGIS current workspace.
9	Address Fields Summary (Optional)	Optional functionality to summarize address fields.

② Run initial address parse (to locate addresses in need of correction)

2.1 Run initial parse

If the addresses to be parsed were constructed in a consistent, conventional, error-free manner, it may only be necessary to run the *Address Parsing Tool* once. However, **if the addresses contain extraneous data, or are occasionally incomplete or erroneous, this tool will help identify where problematic addresses exist, so that they can be corrected within the full (input) site address and parsed again.** This tool may be able to uncover problematic addresses that would otherwise go unnoticed.

To begin the parsing workflow and establish a baseline for site address corrections needed in the file geodatabase feature class, run an initial address parse. When running an initial address parse, the optional parameters should be checked, as shown in Figure 4.

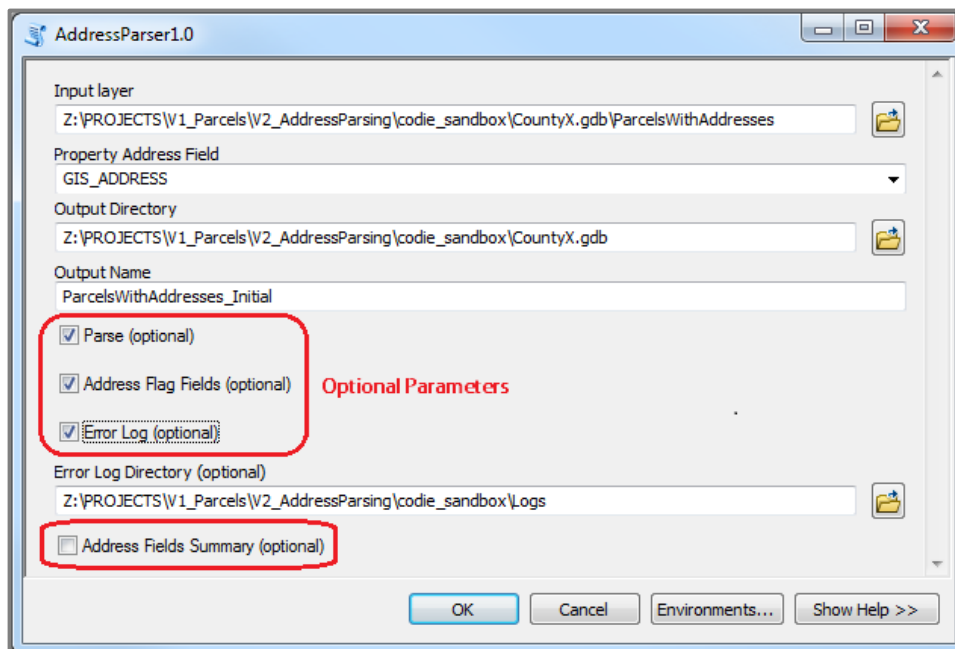


Figure 4. Select optional tool parameters for initial parse

2.2 Interpret results of initial parse

Upon completion of the initial parse, the parse tool will output a new feature class and error log.

First, find addresses the tool was unable to parse, by going into the attribute table of the output feature class and sorting the Parse_Error_Flag field. A value of "1" indicates a parse error, while a value of "0" or Null means the address was parsed correctly or skipped, respectively.

Make edits in the site address field to allow for proper parsing. To get a better understanding of why a parse error occurred, the **Error Log** provides additional information about the parse attempted and where in the address an error occurred.

At the bottom of each log is a count of the total number of parse errors found. Figure 5 shows a sample Error Log with a single parse error.

```

----- OBJECTID: 153 -----
ERROR: Unable to tag this string because more than one area of the string has the same label
ORIGINAL STRING: 586 W 10TH ST & 729 S Thunderbird AVE
PARSED TOKENS: [(u'586', 'AddressNumber'), (u'w', 'StreetNamePreDirectional'), (u'10TH', 'StreetName'), (u'ST', 'StreetNamePostType'),
(u'&', 'AddressNumber'), (u'729', 'AddressNumber'), (u'S', 'StreetNamePreDirectional'), (u'Thunderbird', 'StreetName'), (u'AVE', 'StreetNamePostType')]
UNCERTAIN LABEL: AddressNumber
When this error is raised, it's likely that either (1) the string is not a valid address or (2) some tokens were labeled incorrectly
To report an error in labeling a valid address, open an issue at https://github.com/datamade/usaddress/issues/new - it'll help us continue to improve usaddress!
For more information, see the documentation at http://usaddress.readthedocs.org/

Total Number of Parse Errors: 1

```

Figure 5. Sample Error Log output

Each log record begins with the **OBJECTID** of the record that could not be parsed, followed by a statement about the error. Most errors are caused when the parser tries to place more than one address element into the same address field.

In Figure 5, the parser encountered an address in the form of an intersection, making the parser try to assign multiple elements to the **AddressNumber** field. Standards dictate that an intersection is not an acceptable address. In order to remedy this particular error, one address must be removed from the site address field. Consult the *Address Reference Guide* for more specific information regarding standards and address formatting.

Sometimes the error will indicate elements parsed to address elements not found in the schema. This is a result of using a separate Python module to complete the parsing.

The error log option should be used as many times as necessary, until all parse errors are removed from the data.

Look through the Extraneous_Data field after the first iteration of the address parser. Any address elements that do not fit the schema are placed in this field. Like with parse errors, sort by the field to find the extraneous data. There are two scenarios in which data ends up in the **Extraneous_Data** field:

- Data is not considered part of an address (e.g., a property descriptor such as “barn”)
- Address elements parsed incorrectly

Make note of any records in which adjustments made to the site address still result in errors for the final step of the address parsing process.

③ Run subsequent address parse cycles (until errors and flags are resolved)

3.1 Configure another Address Parsing Tool cycle

After all problems identified by the **Error Log** have been eliminated or corrected, configure another parse through the *Address Parsing Tool*. This time, the **Error Log** option may be left unchecked.

This parse should be executed on the output of the previous parse (or the feature class last edited). It is recommended that a versioned naming convention be adopted to keep input and output feature classes and their versions in order, as depicted below.

Feature Class Input Name	Cycle	Feature Class Output Name
ParcelsWithAddresses	Initial Parse >	ParcelsWithAddresses_Initial »
» ParcelsWithAddresses_Initial	Cycle 1	ParcelsWithAddresses_v1
ParcelsWithAddresses_v1	Cycle 2	ParcelsWithAddresses_v2
ParcelsWithAddresses_v2	Cycle 3	ParcelsWithAddresses_v3

3.2 Interpret results of parse cycle

Again, look for incorrectly parsed addresses in the Extraneous_Data field. This field may contain different address elements depending on the adjustments made from the first tool iteration. Similar steps should be taken with addresses that contain extraneous data as above (section 2.2) in when interpreting results of a parse.

Make corrections based on extraneous data.

Next, sort on the Character_Flag field. This field will include a list of characters found in the address that are not acceptable in a properly formatted address. In some specific cases, characters flagged by the tool are acceptable. Consult the *Address Reference Guide* for details regarding character acceptance.

Remove unacceptable characters from the site address(es).

Sort through the Incomplete_Data_Flag field to locate missing elements. In this field, comments will indicate if an address is potentially missing a key element (e.g., an address number, street name, or street type). These results are based on the initial parse and do not include post-parsing processing within the tool.

This process should be followed with each iteration of the parse cycle.

Once no new flags have been thrown and the rest of the flags accounted for, it is time to move to the final parsing step.

3.3 Final run

To configure the parse tool, check the **Parse** option and **Address Fields Summary** option, and execute the final run.

④ Manual parsing (if necessary)

4.1 Identifying manual edits

The final run of the *Address Parsing Tool* will contain parsed addresses in their most accurate form. However, it is possible that a valid address is an anomaly to the logic within the address parsing tool. **The only way to handle especially challenging addresses is through manual editing of the parsed address fields.**

Having a list of known records from previous steps can help with the identification process. The text file outputs from the **Address Fields Summary** option are another source for locating elements in fields they should not be in.

It is important to scan through all of the records to confirm an accurately parsed dataset. If a correct address is identified to be parsed incorrectly or causing an error, make note of the address (in tabular form if several exist) and submit them to the SCO. The project team will work to better support these addresses.

For your records, save a copy of the table of parsed addresses along with parcel IDs.

⑤ Proceed in creating export or searchable format

Troubleshooting for Address Parsing Tool

My tool does not show-up in the directory I have unpacked to.

If the tool does not appear in the directory you have unpacked it to, first try refreshing the directory in ArcCatalog (Right click directory » Refresh...). If the problem continues, it may be because you are using a legacy version of ArcGIS (ArcGIS 9.0 – ArcGIS 10.1). Tools supporting ArcGIS legacy versions are available at <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>

I get an error when attempting to run the tool.

If running the tool results in an error, first ensure that the tool runs correctly on the test data provided in the zipped package.

- If the tool does not run successfully over the test data, try to interpret the error message in finding a solution and submit the error message (via screen capture or cut and paste) to Codie See at csee@wisc.edu.

I get a properties dialogue instead of a tool input dialogue when opening the tool.

Refresh the directory in ArcCatalog (hit F5 to refresh).

Who can I contact for help?

Codie See, State Cartographer's Office, 608-890-3793, csee@wisc.edu

Credits

Supporting libraries for tool development, thanks to: the datamade/parserator project, <https://github.com/datamade/parserator>

Address Reference Guide

This reference accompanies the *Address Parsing Tool*. It describes the address components participating in full street addresses, the elements the *Address Parsing Tool* outputs, how to check the parsing results, and troubleshoot address parsing errors and flags.

Use this guide if you are utilizing the *Address Parsing Tool* to parse street addresses and want to learn about how to optimize the results of your address parse.

The *Address Parsing Tool* was developed as a part of the Wisconsin Statewide Parcel Initiative as a pathway for data contributors to process unparsed addresses into address sub-components meeting the statewide layer's attribute schema. The schema's address elements are based on the FGDC-endorsed *United States Thoroughfare, Landmark, and Postal Address Data Standard*² and include common elements with definitions. This tool is designed to interpret local addresses and output address components meeting the FGDC and Parcel Initiative standards to the highest degree possible, although variations in data at the local level may impede accuracy.

Why is This Guide Necessary?

In order to achieve the most optimal parsing results from this tool, it is important for the user to understand the nature of the address elements within the Parcel Initiative's schema as well as the various outputs provided as a result of this tool.

This is a reference guide specifically for those using the *Address Parsing Tool*, developed to meet the needs of the schema for the Statewide Parcel Map Parcel Initiative.

Address Element Examples

1.1 Address standards and formatting

The attribute schema for the Wisconsin Parcel Initiative can be found in Appendix D, Schema Requirements for V2. The address elements in the table below are the targeted outputs of the Address Parsing Tool.

Address Element Outputs	
Schema Name (and Alias)	Corresponding Tool Output
ADDNUMPREFIX (Address Number Prefix)	NEW_ADDNUMPREFIX
ADDNUM (Address Number)	NEW_ADDNUM
ADDNUMSUFFIX (Address Number Suffix)	NEW_ADDNUMSUFF
PREFIX (Prefix)	NEW_PREFIX
STREETNAME (Street Name)	NEW_STREETNAME
STREETTYPE (Street Type)	NEW_STREETTYPE
SUFFIX (Suffix)	NEW_SUFFIX
UNITYTYPE (Unit Type)	NEW_UNITYTYPE
UNITID (Unit ID)	NEW_UNITID

Note that this tool aims to parse street addresses only and does NOT parse the following to output fields:

- Landmark Names

² <https://www.fgdc.gov/standards/projects/FGDC-standards-projects/street-address/index.html>

- Place Names (City/Town/Village Name)
- Zip-Codes

If Landmark Names, Place Names, or Zip-Codes exist within the address, they will be written to the extraneous data field while street address will be parsed to their respective elements.

Some address examples, and their expected *Address Parsing Tool* outputs are depicted in Figure 6. The address and ordering of sub-components to outputs depicted meet the specifications of the Parcel Initiative’s Schema.

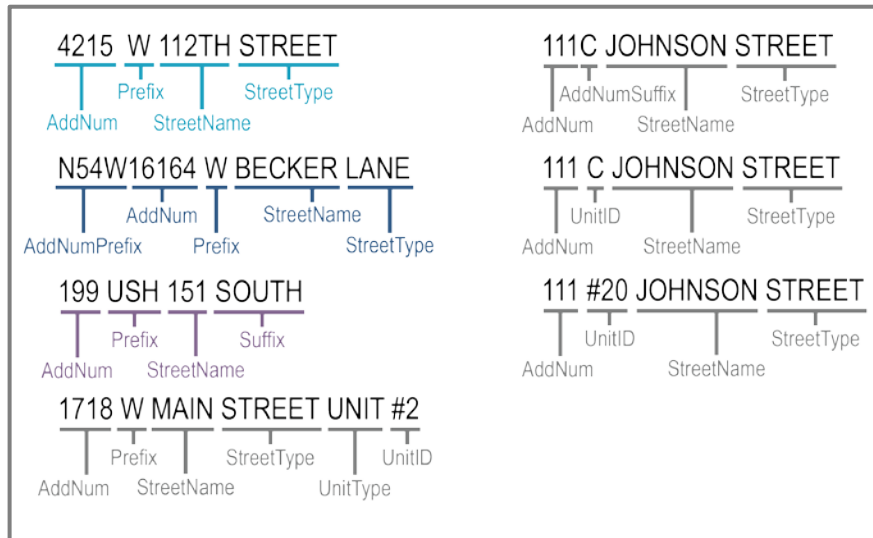


Figure 6. Sub-components of address outputs

Correcting Flags and Errors

2.1 Special characters

The address parsing tool is very sensitive to the inclusion or location of non-alphanumeric characters in the site address field. For this reason, character cleanup is vital to achieving optimal results. In many cases, you will need to manually remove characters flagged by the tool.

Characters flagged by the tool should be removed, with some exceptions where a character is acceptable.

Special Characters		
	Acceptable	Unacceptable – Requires Change
<p>Hyphen (-) The hyphen character is most often used in a site address for indicating a range or compound words.</p>	<p>Hyphens are acceptable in compound street names, concurrently running streets, or unit ID ranges.</p> <ul style="list-style-type: none"> • 100 W MAIN ST UNIT A-E • 100 TRI-COUNTY RD • 100 BUS 18-151 	<p>Hyphens are not allowed to indicate an address number range. A range of address numbers indicates multiple site addresses. There should only be one site address per parcel. To solve this problem, either remove the address number range or duplicate the parcel assigning each parcel a different number in the range.</p> <ul style="list-style-type: none"> • 100-104 W MAIN ST should be 100 W MAIN ST
<p>Forward Slash (/) The forward slash is used frequently in a site address to denote different street names.</p>	<p>Forward slashes are acceptable when denoting concurrently running streets.</p> <ul style="list-style-type: none"> • 100 STH 25/58 	<p>Forward slashes are unacceptable when indicating a street name alias. To solve this problem, remove one of the street names and the forward slash.</p> <ul style="list-style-type: none"> • 100 STH 26/MAIN ST should be 100 STH 26 or 100 MAIN ST
<p>Pound sign (#) The pound sign is used to</p>	<p>Pound symbols are acceptable when used to indicate a unit.</p>	<p>All other situations. To solve this problem, remove the character and other associated text. Then add</p>

indicate a unit. For example, instead of "Unit 1" another way to write it would be "#1."

- 100 W MAIN ST #1
- 100 #1 W MAIN ST

number if known.

- **NO#** W MAIN ST should be 100 W MAIN ST or W MAIN ST
- **UNKNOWN #** W MAIN ST should be 100 W MAIN ST or W MAIN ST

2.2 Extraneous Data

The **Extraneous_Data_Flag** field is generated when you run the parser over the site address field. It contains various data that the parser was unable to assign to a common address element. **In most cases this extraneous information should be removed from the full address prior to re-running the parse tool.**

A few examples of data that could be found in this field include the latter portion of an address range or some type of property descriptor, or an alternate street name attached to the full address.

Please note that while in most cases this information is not an acceptable address element, it is possible that the parser incorrectly placed an acceptable element into this field. It is recommended you take time after your final parse to manually move elements that may have inadvertently ended up in the **Extraneous_Data_Flag** field to their appropriate field.

Leverage the Tool's Summary Tables

3.1 Address Fields Summary output tables

OBJECTID*	FREQUENCY	NEW_STREETNAME
1	1	<Null>
2	1	40 SUGAR MAPLE
3	1	44
4	1	73RD
5	1	BAYOU
6	1	BLACKSMITH
7	1	BONANZA
8	1	CRIPPLE
9	1	CRIPPLE CREEK
10	1	EAGLE
11	1	EAGLE KREST
12	1	ELIZABETH
13	1	FIRE LAKE
14	2	GOLDENROD
15	1	HOWELL LN HOOD
16	1	IMLAY
17	1	JUNCO
18	1	JUNCO BIRD
19	1	LAKE 5
20	1	MADLENE
21	1	MADISON
22	1	MAIN
23	3	MARSH
24	1	PACKERS
25	2	SENECA
26	2	SHADY LANE
27	1	SMITHWICK
28	1	SUGAR MAPLE
29	1	VALLEY
30	1	VV

Once you have successfully run through as many iterations as necessary to remove **Parse_Error_Flags**, corrected issues associated with **Incomplete_Data_Flags** and **Character_Flags**, it is recommended that you select the tool check box that will create **Address Fields Summary** for your Prefix, StreetName, StreetType, and Suffix fields. These summary tables will provide an additional method for double checking the parse success and identify any potential errors that may not have been identified by the various flags listed above. Below are some examples of the summary tables from the (SampleData.gdb » ParcelSamples) feature class provided with the tool.

Figure 7 shows a summary of the **StreetNames** from the **ParcelSamples** feature class. The two highlighted records "40 SUGAR MAPLE" and "HOWELL LN HOOD" are incorrect street names. Once you have read through the summary table and identified incorrect names, you can make corrections in various ways:

Option 1

Sort the attributed field alphabetically, open an editing session, navigate to the problematic street name and make the necessary correction. Be sure to make the correction to the full address field that you are running the parse over too, especially if you plan to re-run the parse.

Option 2

Another option is to use the **Select by Attributes** option within the attribute table, open an editing session and construct a definition query that will select the incorrect records and allow you to make the necessary corrections. Be sure to make the correction to the full address field that you are running the parse over too, especially if you plan to re-run the parse.

Figure 7. Highlighting of incorrect street names

OBJECTID *	FREQUENCY	NEW_STREETTYPE
1	5	<Null>
2	3	AVE
3	1	AVENE
4	1	BIRD
5	3	BLVD
6	1	CIR
7	1	CREEK
8	2	DR
9	1	KREST
10	2	LN
11	2	PL
12	1	PS
13	1	R
14	4	RD
15	6	ST
16	1	WAY

Figure 8. Highlighting of incorrect street types

Figure 8 shows a summary of the **StreetTypes** from the **ParcelSamples** feature class. The highlighted records are uncommon/incorrect street types. Once you have identified incorrect street types, you can make corrections in various ways.

Note that often when you see uncommon street types this is a result of a multi-word street name that is missing its street type. In many cases, the second word of the street name will be placed in the street type attribute field.

Option 1

Sort the attributed field alphabetically, open an editing session, navigate to the problematic street name and make the necessary correction. Be sure to make the correction to the full address field that you are running the parse over too, especially if you plan to re-run the parse.

Option 2

Another option is to use the **Select by Attributes** option within the attribute table, open an editing session and construct a definition query that will select the incorrect records and allow you to make the necessary corrections. Be sure to

make the correction to the full address field that you are running the parse over too, especially if you plan to re-run the parse.

DOR XML Parsing Tool Guide

The *DOR XML Parsing Tool* helps prepare tax roll data in XML format to be joined to parcel geometries. It will aggregate all XML datasets for a county, parse the XML into a database file that will contain one record per taxable parcel, and allow you to join the output to GIS parcels via a PIN or TaxKey ID.

Use this guide if your county’s tax roll data is already in XML format and you would like to use that XML data to help meet Benchmarks 1 or 2 in either the searchable or export format.

When to Use DOR XML Parsing Tool		
	Benchmark 1 Parcel and Zoning Data Submission	Benchmark 2 Extended Parcel Attribute Set Submission with Parsed Address Elements
EXPORT FORMAT OPTION	You have XML tax roll data which you’d like to submit separately from GIS parcels ✓ This tool can assist	You have XML tax roll data which you’d like to submit separately from GIS parcels ✓ This tool can assist
SEARCHABLE FORMAT OPTION	You have XML tax roll data which you’d like to submit already joined to GIS parcels ✓ This tool can assist	You have XML tax roll data which you’d like to submit already joined to GIS parcels ✓ This tool can assist

Why is This Guide Necessary?

The project team has worked to make the Parcel Initiative’s goals consistent with DOR’s preferred XML format standard for property tax roll data. While DOR’s XML format standard brings a county close to meeting Benchmarks 1 and 2 for the submittal of parcel datasets, it does not quite meet the needs of the Parcel Initiative for the following reasons:

- Up until August of 2015, the DOR XML standard did not include a “fair market value” attribute, which is required per state statute 59.72(2)(a). Estimated fair market value was added to 2016 assessment and tax roll XML schema. Data submitted to DOA in XML format must include the fair market value attribute.
- XML does not accommodate parcel geometry. While counties may use the XML format for attributes, submission of parcel geometries to DOA is required.
- The DOR XML format does not require parsing of street address elements, which is a requirement for Benchmark 2 in both the searchable and export formats.

How to Use This Guide

It is recommended you have the following materials at hand:

- The *Version 1 Statewide Parcel Map Database Project Interim Report* and Digital Appendices
- Your XML data – Must be a *validated* submission to DOR, can be individual jurisdiction files, or a county-wide aggregate
- Your GIS parcel data
- The county staff member(s) who will submit the data for V2
- The *XML Parsing Tool* from <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>

Process Overview

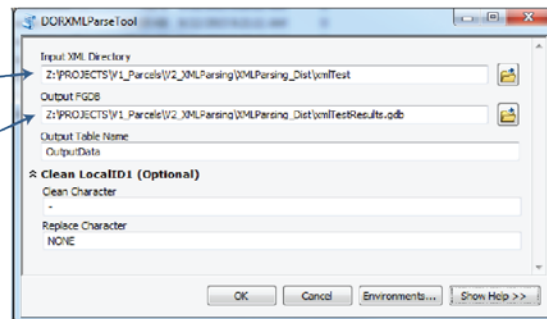
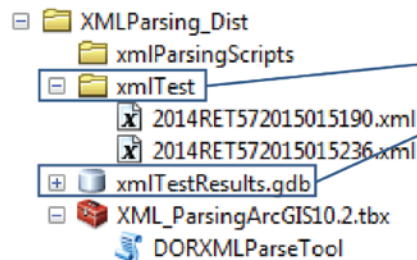
STEP 1 Install the Tool

STEP 2 Examine XML and parcel data

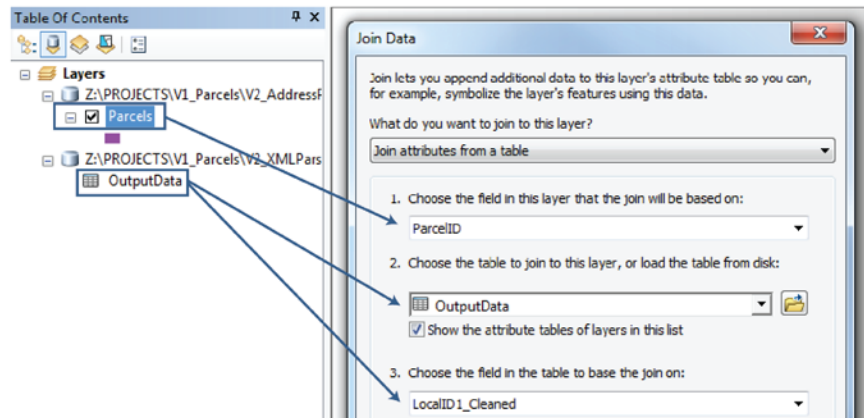
```
<PropertyInfo>  
<LocalID1>57-236-2-41-09-21-1</LocalID1>  
<LocalID2>38557</LocalID2>  
<Town>41</Town>  
<Range>09</Range>  
<RangeDirection>W</RangeDirection>  
<Section>21</Section>  
<Quarter40>NE</Quarter40>  
<Quarter160>NE</Quarter160>
```

OBJECTID *	Shape *	GISPIN
1	Polygon	57-236-2-41-09-21-1
2	Polygon	57-236-2-41-09-21-7
3	Polygon	57-236-2-41-09-21-9

STEP 3 Configure and execute tool



STEP 4 Table join the output .dbf to parcel geometries



STEP 5 Proceed in creating export or searchable format...

Figure 1. DOR XML Parsing Tool process overview

1 Install the Tool

1.1 Download and install

Download the zipped package with the *DOR XML Parsing Tool*, an ArcPy script tool, from <http://www.sco.wisc.edu/images/stories/publications/V2/tools>.

Unzip to the directory of your choice. Then open ArcCatalog and navigate to the new directory. You should see toolboxes, labeled with their respective ArcGIS version compatibility. Choose the toolbox that fits your ArcGIS install. If you do not see any toolboxes, hit F5 to refresh the directory.

The tool should be ready to run. The package includes two sample XML files in the *XMLParsing_Dist\xmlTest* directory that can be used to test the tool. If you experience trouble starting or executing the tool, see the Troubleshooting section below.

2 Examine XML and parcel data

2.1 Optional: Add Estimated Fair Market Value to your XML

Estimated Fair Market Value was added to 2016 assessment and tax roll XML schema, but was absent prior to that. If your XML data includes it, you may skip this step.

If your county does not currently support Estimated Fair Market Value in the XML data that you are about to parse, you have the option to add it to the XML data now. If adding Estimated Fair Market Value to the XML data is not feasible, this field and its values can be joined to parcels after completing the processes outlined in this guide.

2.2 Understand your XML data and the appropriate join PIN

The DOR XML schema supports three different types of PIN; **LocalID1**, **LocalID2**, and **ParcelID**. While **LocalID1** is the only required element of the three, it may be true that your county or jurisdiction has included **LocalID2** or **ParcelID** in the XML data. It may also be the case that **LocalID2** or **ParcelID** are essential to creating a proper join between tax roll information derived from the XML and parcel geometries.

Before proceeding, it is recommended that you understand your XML data and ensure that the appropriate PIN is available for joining the XML data to parcels. Note that it may be the case that a table separate from your parcel dataset contains the appropriate join key. It may also be the case that your Parcel or DOR XML PIN needs modification to correctly join.

Your county parcel dataset may include a PIN that is very similar to **LocalID1** of your XML data. For example, it may contain the same digits included in **LocalID1** but exclude hyphens. For this reason, the *XML Parsing Tool* includes a parameter for clearing-out a specific string in the output .dbf. This output is written to the **LocalID1_Clean** field. See Figure 2 for an example of the outcome if the **Clean Char** parameter is set to “-” and the **Char Replace** parameter is set to “NONE.”

XML:

```
<PropertyInfo>
  <LocalID1>57-236-2-41-09-21-1 01-000-000020</LocalID1>
  <LocalID2>38557</LocalID2>
  <ParcelID>57587</ParcelID>
  <Town>41</Town>
  <Range>09</Range>
  <RangeDirection>W</RangeDirection>
```

Output:

Output Field	Output PIN
LocalID1	57-236-2-41-09-21-1 01-000-000020
LocalID1_Cleaned	5723624109211 01000000020

Figure 2. PINs in XML data and the outcome of cleaning characters

③ Configure and execute tool

3.1 Use the *XML Parsing Tool* to parse the XML and turn it into a table that can be joined to GIS features

Execute the tool by entering an entire directory of validated DOR XML data as the **Input XML Directory** parameter and the tool will iterate through all XML in the directory and aggregate it. If you are only performing a parse on one file, only include that one file in the directory.

The **Output FGDB** parameter accepts a file geodatabase only. This will be the output location of the .dbf that contains the parsed/extracted data. The **Output Table Name** parameter accepts a string, which will be the desired name of the .dbf to be placed in the **Output FGDB**.

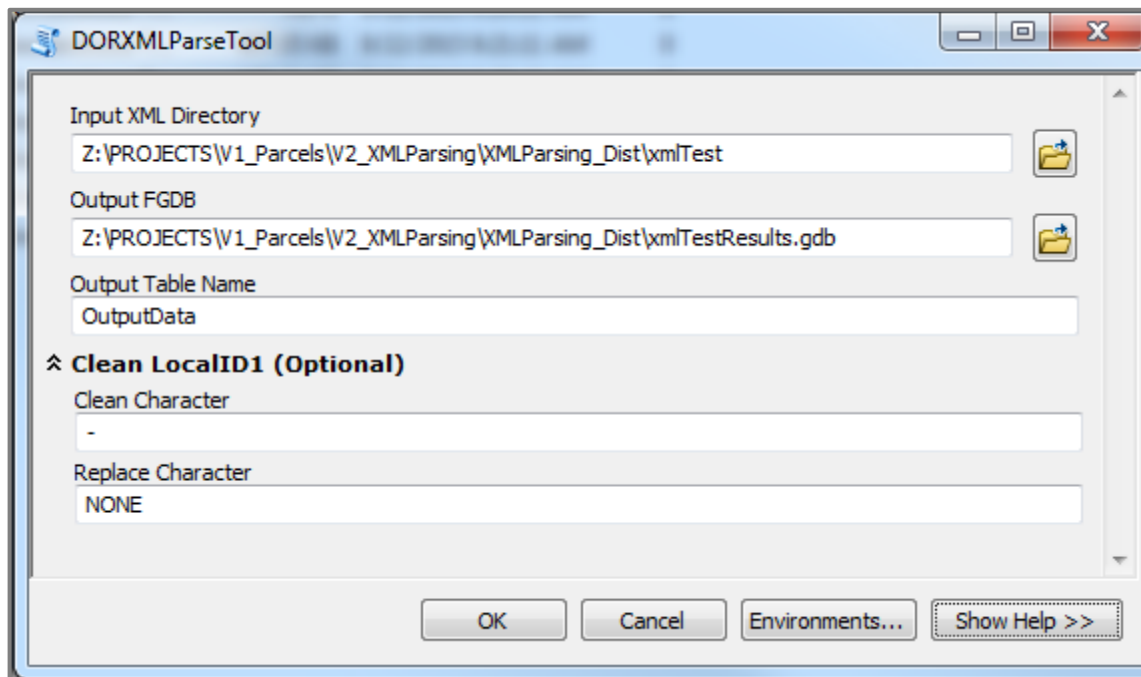


Figure 3. XML Parsing Tool

If you notice a slight difference between the **LocalID1** in your XML data and the PIN to be used to join the output XML to parcel geometries (such as the absence of hyphens), the optional parameters within the **Clean LocalID** dropdown may be useful.

Using the **Clean LocalID** option, a copy will be made of the **LocalID1** field called **LocalID1_Cleaned** in the output .dbf. The string parameter set in **Clean Character** will be the string stripped from the **LocalID1_Cleaned**. The string parameter set in **Replace Character** will replace the string identified in the **Clean Character** parameter. If the **Replace Character** parameter is set to "NONE" then the **Clean Character** will be replaced with nothing ("") in the output **LocalID1_Cleaned** field.

3.2 Click OK to run tool

④ Table join output .dbf to parcel geometries

4.1 Ensure that all features are joined

Upon completion of the *XML Parsing Tool*, ensure that you are able to accurately join the newly created .dbf table to your parcel geometries via a table join on the appropriate join fields.

If the XML correctly adheres to the DOR's schema, the *XML Parsing Tool's* output .dbf will represent each real property as an individual "taxable unit."

Note on condo modeling. If your parcel dataset models condos relationally, or does not model multi-owner units at all, extra processing is required. In these cases, either no taxable units or not all taxable units from the tax roll will be joined to the parcels due to the fact that one parcel geometry represents multiple taxable units. To correct this, use the *Parcel Stack Tool* available at <http://www.sco.wisc.edu/images/stories/publications/V2/tools>.

⑤ Proceed in creating export or searchable format

Troubleshooting for DOR XML Parsing Tool

My tool does not show-up in the directory I have unpacked to.

If the tool does not appear in the directory you have unpacked it to, first try refreshing the directory in ArcCatalog (Right click directory » Refresh...). If the problem continues, it may be because you are using a legacy version of ArcGIS (ArcGIS 9.0 – ArcGIS 10.1). Tools supporting ArcGIS legacy versions are available at <http://www.sco.wisc.edu/images/stories/publications/V2/tools>.

I get an error when attempting to run the tool.

If running the tool results in an error, first ensure that the tool runs correctly on the test data provided in the zipped package.

- If the tool runs successfully over the test data, then double-check your XML for errors against the DOR XML schema. Remember, your data must be validated by DOR to work with this tool.
- If the tool does not run successfully over the test data, try to interpret the error message in finding a solution and submit the error message (via screen capture or cut and paste) to Codie See at csee@wisc.edu.

My output is blank.

If the output is blank, make sure that there are XML files on the root directory of the "Input XML Directory." Nested folders will not be consumed by the tool (not recursive). Also, verify that the XML is valid and adhering to the DOR XML schema.

The .dbf result contains errors.

If you detect an anomaly or other problem in the result of the XML parse, please submit your observations to Codie See at csee@wisc.edu.

I cannot join my .dbf result to my parcel dataset.

Take a close look at the three different types of PIN within the DOR XML data; **LocalID1**, **LocalID2**, and **ParcelID** and compare these to any available PINs associated with your Parcel data. Try to note any similarities, even if the PINs are not a perfect match. It may be the case that a table separate from your parcel dataset contains the appropriate join key. It may also be the case that your parcel or DOR XML PIN requires modification to correctly join.

Note that there may be jurisdictional variations in PIN across the county, so some jurisdictions may join correctly while others will not. The Wisconsin Department of Revenue webpage documenting municipal parcel formats may be useful (www.revenue.wi.gov/ust/parcels.html).

Who can I contact for help?

Codie See, State Cartographer's Office, 608-890-3793, csee@wisc.edu

Data Standardize Tool Guide

The *Data Standardize Tool* standardizes parcel data field values to meet the requirements of the searchable format. It will generate a summary and lookup table of domains, which can be edited and utilized to standardize a field.

Use this guide if you wish to standardize fields to match the searchable format, outlined in Appendix D, Schema Requirements for V2, in the *V1 Interim Report*.

When to Use Data Standardize Tool		
	Benchmark 1 Parcel and Zoning Data Submission	Benchmark 2 Extended Parcel Attribute Set Submission with Parsed Address Elements
EXPORT FORMAT OPTION	Standardized domains/fields not required	Standardized domains/fields not required
SEARCHABLE FORMAT OPTION	✓ This tool can assist	✓ This tool can assist

Fields Requiring Standardization for Searchable Format	
Field Name	Definition
PREFIX	Prefix
STREETTYPE	Street Type
SUFFIX	Suffix
SCHOOLDIST	School District
SCHOOLDISTNO	School District Number
IMPROVED	Improved Structure
PROPCLASS	Class of Property
*PROPCLASS	Class of Property
*County Name	County Name
*PARCELFIPS	Parcel Source FIPS
*PARCELSRC	Parcel Source Name

*Asterisked fields require standardization for the searchable format, but do not necessarily need to be run through this tool, as they can simply be calculated with one domain within the field calculator or a similar instrument.

Why is This Guide Necessary?

The statewide parcel layer can be more consistent, understandable, searchable, maintainable, and user friendly if the data within is standardized. Standardization is required for the above fields within **searchable format** submissions. It may be valuable to standardize other fields in the submission, but this should be done at the submitter's discretion.

How to Use This Guide

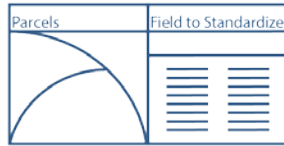
It is recommended you have the following materials at hand:

- The *Version 1 Statewide Parcel Map Database Project Interim Report* and Digital Appendices
- The data participating in the searchable format submission, in file geodatabase format
- The county staff member(s) who will submit the data for V2
- The *Data Standardize Tool* from <http://www.sco.wisc.edu/images/stories/publications/V2/tools>

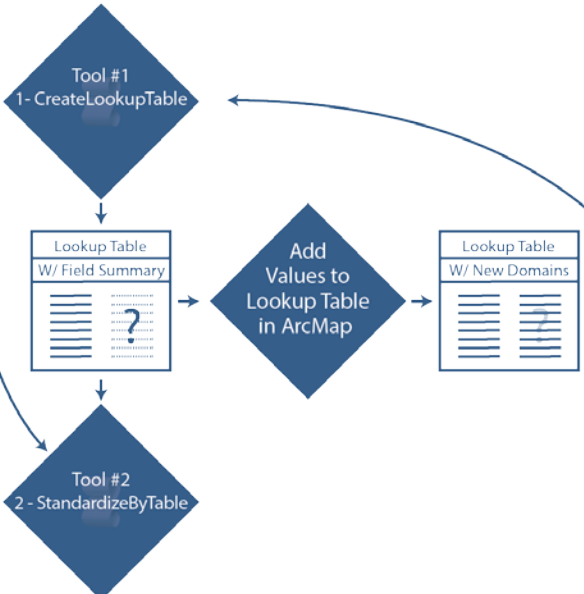
Process Overview

STEP 1 Install the Tool

STEP 2 Prepare the data to be standardized



STEP 3 Configure and execute tool #1, the 1 - CreateLookupTable Tool



STEP 4 Configure and execute tool #2, the 2- StandardizeByTable Tool



STEP 5 Repeat sequence (as needed)

STEP 6 Proceed in creating export or searchable format...

Figure 1. Data Standardize Tool workflow overview

1 Install the Tool

1.1 Install the Data Standardize tools

The *Standardize Domains Toolbox* contains two separate ArcPy script tools, both of which participate in the standardizing workflow. These tools can be installed by downloading the zipped package from <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>.

Unzip the download to the directory of your choice. Then open ArcCatalog and navigate to the new directory. You should see toolboxes, labeled with their respective ArcGIS version compatibility. Choose the toolbox that fits your ArcGIS install. If you do not see any toolboxes, hit F5 to refresh the directory.

The tools should be ready to run. If you experience trouble starting or executing either of the tools, see the Troubleshooting section below.

2 Prepare the data to be standardized

2.1 Determine the fields that require standardization

If you are using this tool as a step towards submitting data via the searchable format, the following fields require standardization.

Field Name	Definition	Field Name in County Land Information System
PREFIX	Prefix	
STREETTYPE	Street Type	
SUFFIX	Suffix	
SCHOOLDIST	School District	
SCHOOLDISTNO	School District Number	
IMPROVED	Improved Structure	
PROPCLASS	Class of Property	

2.2 Prepare for data standardization

These tools require a file geodatabase feature class as the main input GIS file containing the fields to be standardized. If the data does not yet exist as a file geodatabase feature class, take the appropriate actions to export or join the data into this format.

Also note that the steps outlined in this guide will need to be executed once for each field that is to be standardized.

③ Configure and execute tool #1, the 1 - CreateLookupTable Tool

The first tool of the sequence is the *1 - CreateLookupTable Tool*. This tool will summarize the field that is to be standardized and output a file geodatabase table containing the following:

- A field containing each unique value within the field of interest, named according to the input field
- A **FREQUENCY** field containing a count of the number of occurrences of the given attribute in the field of interest
- A blank field that will be manually edited in the next step of the cycle

To configure the *1 - CreateLookupTable Tool* and create a lookup table, configure the tool like the example in Figure 2.

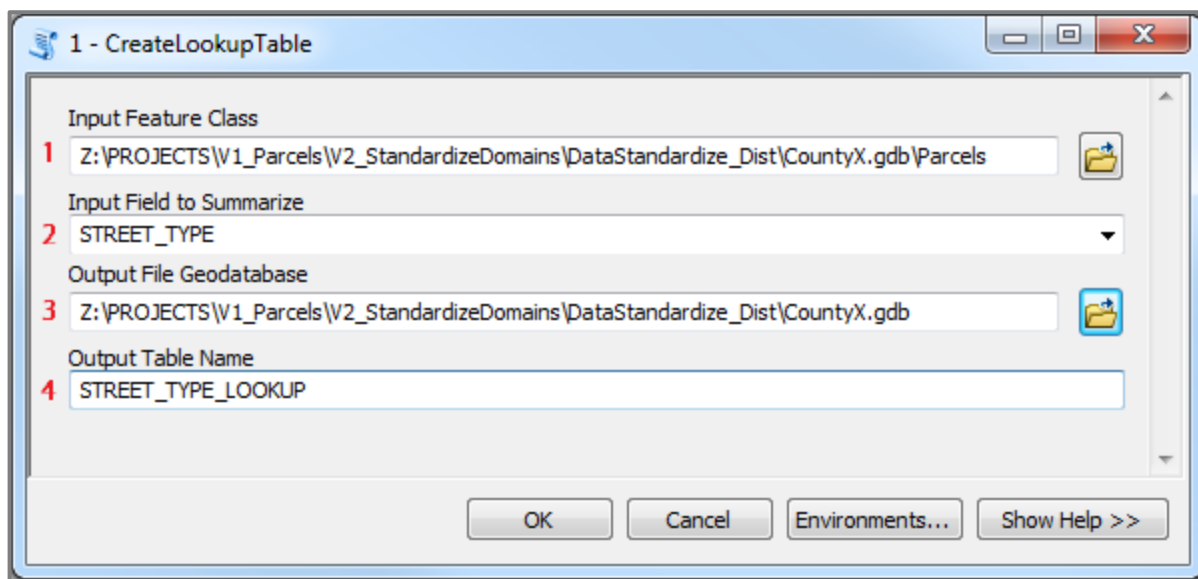


Figure 2. Configuring the 1 – CreateLookupTable Tool

1 – CreateLookupTable Parameter Definitions	
1 Input Feature Class	The feature class containing the field whose attributes are to be summarized and then standardized in 2 – <i>StandardizeByTable Tool</i>
2 Input Field to Summarize	The field whose attributes are to be summarized and then standardized in 2 – <i>StandardizeByTable Tool</i>
3 Output File Geodatabase	The output file geodatabase to which the output lookup table will be written
4 Output Table Name	The name of the lookup table to be written as an output

The outcome of *1 – CreateLookupTable Tool* is a lookup table named as appropriate within the file geodatabase, as seen in Figure 3.

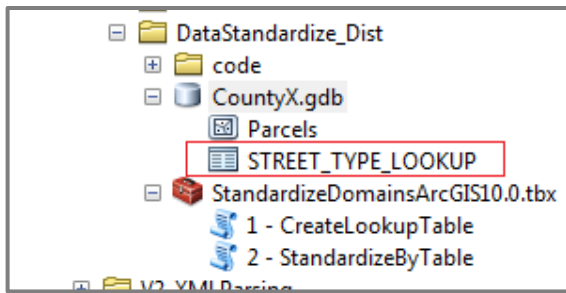


Figure 3. Output table from the 1 – CreateLookupTable Tool

3.1 Edit the newly created lookup table in ArcMap

The newly created lookup table will contain a blank field named as the original field’s name with an **_ST1** appended to the end. The second tool, 2 – *StandardizeByTable*, will consume this table and write out whatever corresponding value exists within this field as the standard domain.

Open the lookup table in ArcMap, start an edit session on this table, and manually edit the **_ST1 column as appropriate.** Figure 4 depicts an example of how this would be done to standardize the various forms of “road” and “street.”

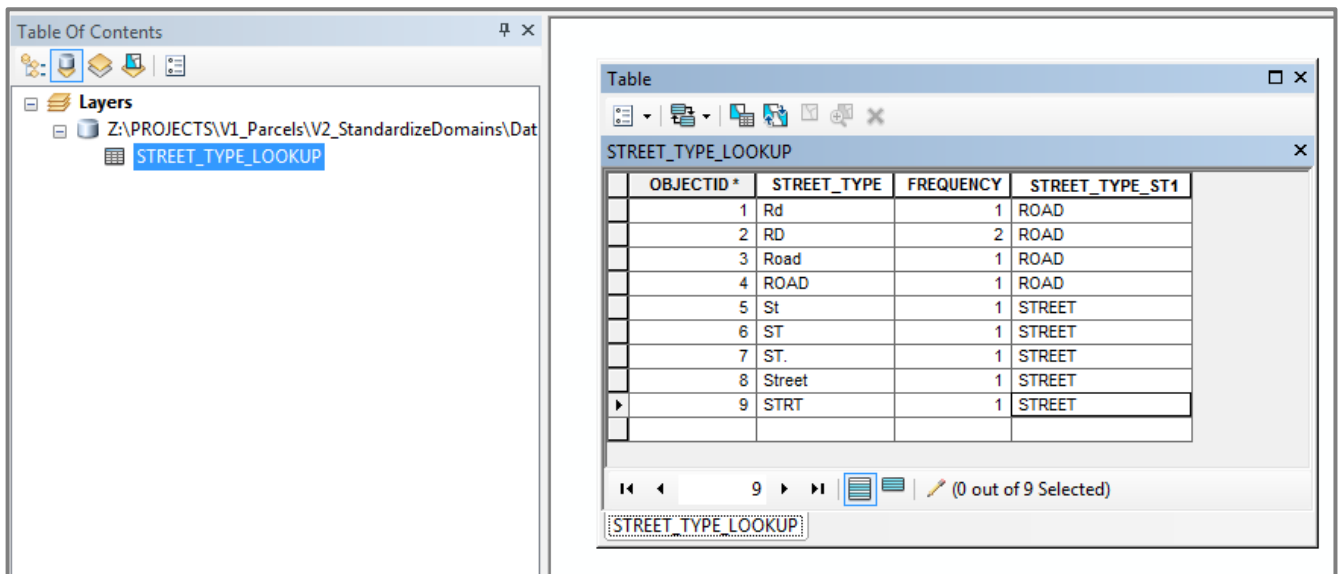


Figure 4. Editing the Lookup Table in an ArcMap editing session

The list of acceptable domains appears as [Digital Appendix J, V2_Parcel_Domain_List.xlsx](#), in the *V1 Interim Report*. Note that it may be efficient to use a table join to a list of acceptable domains in order to remove the attributes that already meet the correct domain standard.

④ Configure and execute tool #2, the 2 – StandardizeByTable Tool

The second tool of the sequence is the 2 – *StandardizeByTable Tool*. This tool will consume the newly edited lookup table and use the **_ST1** field to standardized and output a file geodatabase feature class containing a new field with the correctly standardized attributes.

Use the newly edited lookup table to configure the 2 – *StandardizeByTable Tool* by following the configuration illustrated in Figure 5.

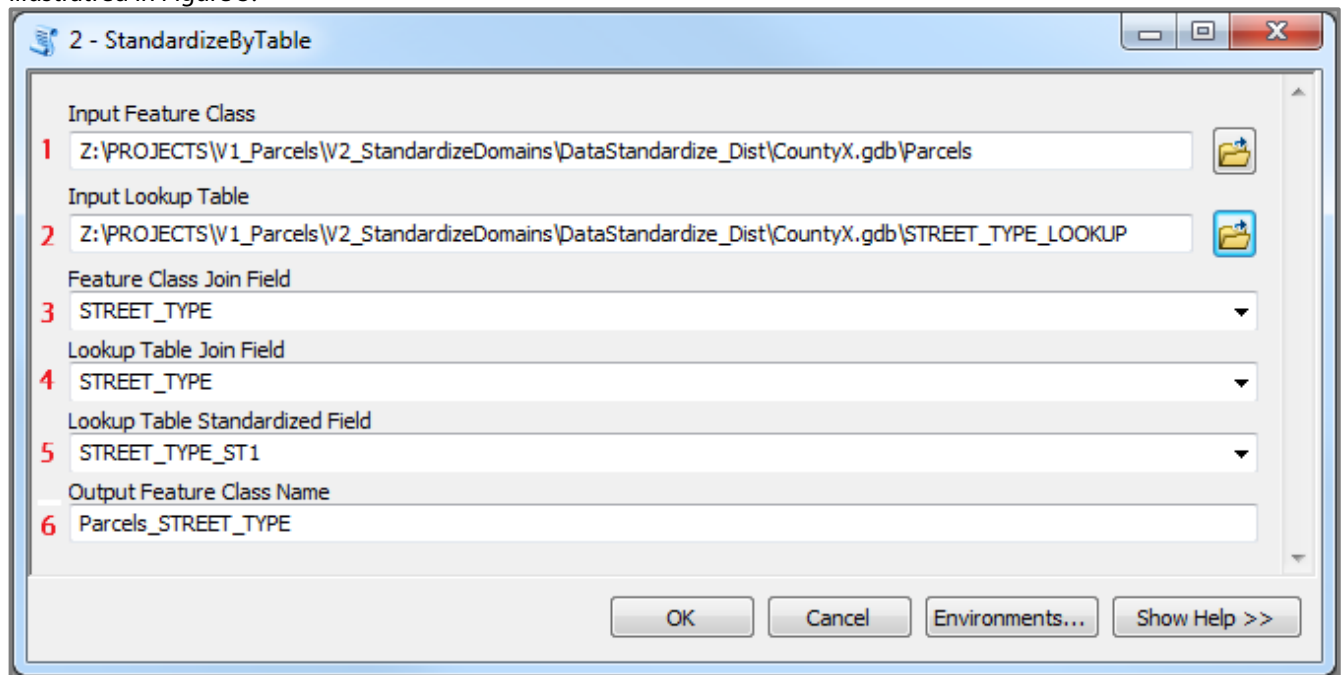


Figure 5. Configuring the 2 – *StandardizeByTable Tool*

2 – StandardizeByTable Parameter Definitions	
1 Input Feature Class	The feature class containing the field to be standardized.
2 Input Lookup Table	The input lookup table, created with 1 – <i>CreateLookupTable Tool</i> , containing a field summarizing all unique attributes in the field of interest and another field containing the appropriate domain values the attributes should be standardized to.
3 Feature Class Join Field	The field of interest within the feature class that contains the attributes to be standardized.
4 Lookup Table Join Field	The field within the lookup table containing a summary of all unique attributes in the field of interest.
5 Lookup Table Standardized Field	The field within the lookup table containing the appropriate domain values the output attributes should be standardized to.
6 Output Feature Class Name	The name of the output feature class. This feature class will automatically be written to the file geodatabase that the input feature class resides-in.

After running the 2 – *StandardizeByTable* tool to standardize the attributes in the field of interest, the outcome will be standardized fields within the newly created feature class. The original fields will remain intact as the standardized fields will be written to a field named **Original Field Name_ST2**. Figure 6 shows an example output.

OBJECTID *	Shape *	Original Field	New Field
1	Polygon	St	STREET
2	Polygon	STRT	STREET
3	Polygon	ST	STREET
4	Polygon	ST.	STREET
5	Polygon	Street	STREET
6	Polygon	Rd	ROAD
7	Polygon	ROAD	ROAD
8	Polygon	RD	ROAD
9	Polygon	Road	ROAD
10	Polygon	RD	ROAD

Figure 6. Final output of 2 – StandardizeByTable is standardized attributes in a new field

5 Repeat sequence (as needed)

The standardization sequence articulated through the steps completed to this point will need to be executed once for each field that is to be standardized. Make sure to save your lookup tables for use in subsequent standardization efforts. By joining a legacy lookup table to a freshly summarized lookup table, updating the `_ST1` field should require little additional effort.

6 Proceed in creating searchable format

Troubleshooting for Data Standardize Tool

My tool does not show-up in the directory I have unpacked to.

If the tool does not appear in the directory you have unpacked it to, first try refreshing the directory in ArcCatalog (Right click directory » Refresh...). If the problem continues, it may be because you are using a legacy version of ArcGIS (ArcGIS 9.0 – ArcGIS 10.1). Tools supporting ArcGIS legacy versions are available at www.sco.wisc.edu/projects/v2-parcels.html.

I get an error when attempting to run the tool.

If running the tool results in an error, first ensure that the tool runs correctly on the test data provided in the zipped package.

- If the tool does not run successfully over the test data, try to interpret the error message in finding a solution and submit the error message (via screen capture or cut and paste) to Codie See at csee@wisc.edu.

It may also be useful to restart ArcCatalog if you continue to get errors.

I get a properties dialogue instead of a tool input dialogue when opening the tool.

Refresh the directory in ArcCatalog (hit F5 to refresh).

Who can I contact for help?

Codie See, State Cartographer's Office, 608-890-3793, csee@wisc.edu

Condo Stack Tool Guide

The *Condo Stack Tool* helps to model condominium parcel geometries in a “stacked” geometric form that meets the requirements of the Parcel Initiative’s data model.

Use this guide if your county’s condominium parcel data is modeled relationally (1:M) or not modeled at all in your land information system and you would like to use that parcel data to join (1:1) to the county tax roll information. This tool can help certain counties meet Benchmarks 1 or 2 in either the searchable or export format by duplicating condo parcel geometries for every tax record in the tax roll, making it feasible to perform a 1:1 join between parcel geometries and tax records in the tax roll.

Geometric modeling of condos is required of both benchmarks and of both submission formats. Geometric modeling means there must be a one-to-one relationship between parcel geometries and records in the attribute table or tax roll . Each parcel must attach to one, and only one, record; each record must attach to one, and only one, parcel.

When to Use Condo Stack Tool		
	Benchmark 1 Parcel and Zoning Data Submission	Benchmark 2 Extended Parcel Attribute Set Submission with Parsed Address Elements
EXPORT FORMAT OPTION	You have condo tax information modeled relationally (1:M) or not modeled at all ✓ This tool can assist	You have condo tax information modeled relationally (1:M) or not modeled at all ✓ This tool can assist
SEARCHABLE FORMAT OPTION	You have condo tax information modeled relationally (1:M) or not modeled at all ✓ This tool can assist	You have condo tax information modeled relationally (1:M) or not modeled at all ✓ This tool can assist

Background & Why is This Tool Necessary?

Condos are not as simplistically mapped as other forms of real property because they incorporate land access to common facilities which are collectively owned. Although condos involve “units” of individual ownership, these units are abstracted from their geographic locations and do not typically adhere to a discrete legal description. When maintaining taxable information, each individual “unit” of ownership is easily annotated in table form but when joining these tabular elements to geographic and geometric elements, a more sophisticated strategy must be employed. This strategy varies by jurisdiction or dataset and takes two general forms; geometric models and relational models.

Relational Models:

Several counties and municipalities in Wisconsin practice relational modeling for their condos. Figure 1 illustrates the relational concept. Jurisdictions who practice this model may find the condo stack tool useful.

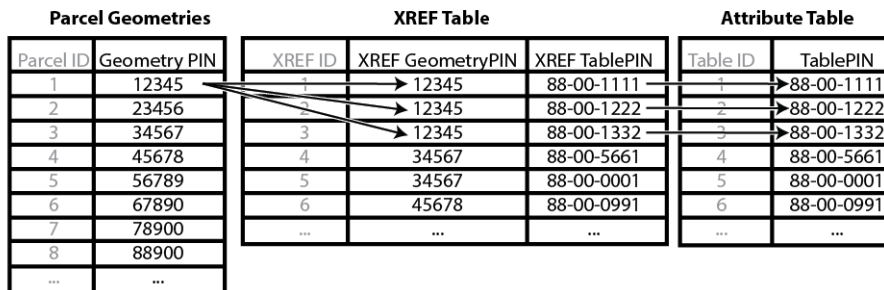
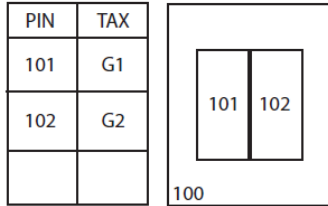


Figure 1. Relational Model Concept

Geographic Models:

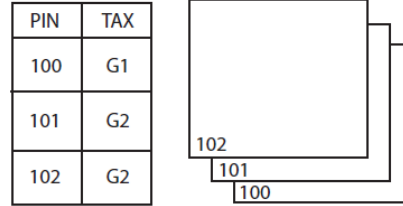
Through data submitted to the Version 1 Parcel Project, the Parcel Initiative team observed four distinct ways that condo parcels were geometrically modeled in GIS. These four types are further illustrated in Figure 2. All four models illustrated here are acceptable for both the export and searchable standards for the Parcel Initiative. However, the condo stack tool will help achieve the Condo Type #2 model illustrated in Figure 2.

Condo Type #1



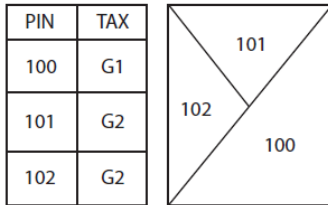
**No record in tax roll for PIN 100

Condo Type #2



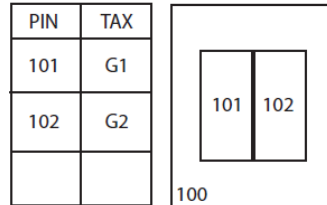
**Stacked parcels, 1 per owner

Condo Type #3



**Main parcel divided up into segments.
Not representative of individual unit/parcel geometry. (Common legal description)

Condo Type #4



**Follows same model as #1, but PIN 100 contains common taxable elements prorated across 101 & 102

Figure2. Condo Model Scenarios

Parcel-Attribute Relationships to Note:

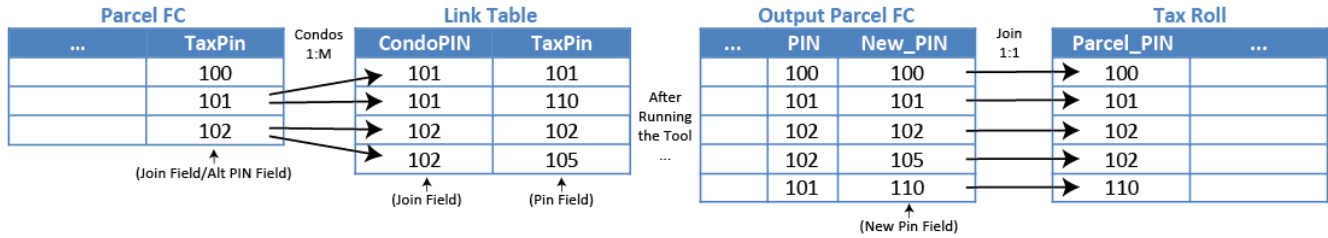
- There must be a one-to-one relationship between parcel geometries and records in the attribute table or tax roll. Each parcel must attach to one, and only one, tax roll record; each tax roll record must attach to one, and only one, parcel.
- Every record in the tax roll should attach to a single parcel geometry. If a record exists in the tax roll but not in the parcel geometry, it is a missing parcel geometry. There should be no missing parcel geometries.
- In the case of condos, or other collective real property ownerships, if there is more than one tax record for the same area of land, each record must attach to one, and only one parcel geometry.

Notes about Common Parcel-Link Table Relationships:

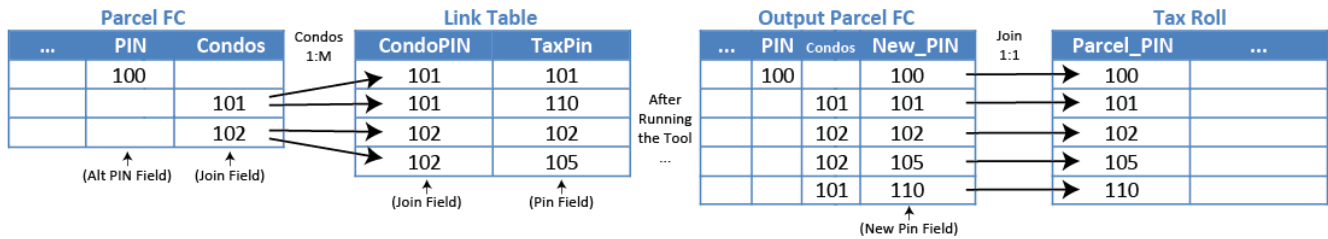
- There are 3 prominent and common relationships between a given condo parcel geometry, the link table, and tax roll records.
- **Standard:** Involves the parcel geometries, the required link table and the stand alone tax roll
- **Alternate PIN Field:** Involves the same elements as listed above, but the join field is specific to condos only, so the actual PIN that will be assigned to the standard parcels (i.e. the ones that don't need to be stacked) will be pulled from the Alt PIN Field
- **Tax Roll / Link Table:** Involves the elements in the standard relationship noted above, except for the fact that the Tax Roll and the Link Table are one in the same (i.e. this table contains both the join field, the respective PINs associated with the join field, and the related tax information). This is not as common as the other two relationship types, but does exist.

**If you need assistance when determining which fields to select for a given tool parameter, referring to Figure 3 could be a helpful way of understanding the relationship between specific fields and the function they serve.

1) Standard:



2) Alternate PIN Field:



3) Tax Roll / Link Table:

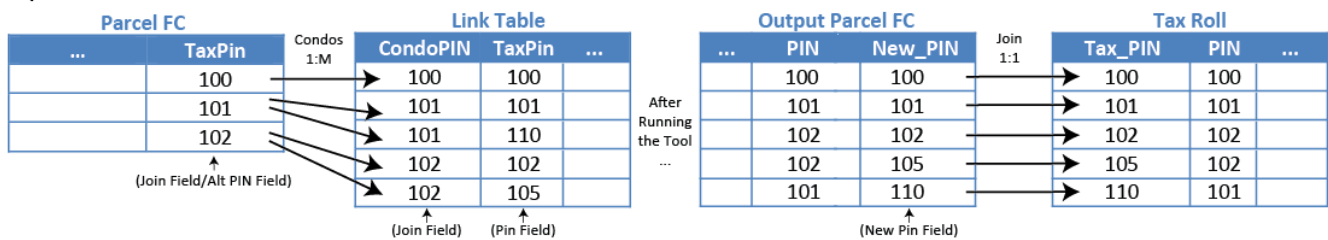


Figure 3. Parcel-Link Table Relationships

How to Use This Guide

It is recommended you have the following materials at hand:

- The *Version 1 Statewide Parcel Map Database Project Interim Report* and Digital Appendices
- Your GIS parcel data
- Your Link Table data in file geodatabase table (.dbf) format and containing:
 - Join ID (ID that correctly joins to parcel geometries)
 - Parcel ID (ID that is unique to each unit of ownership – e.g. each condo owner must have a unique ID)
- The county staff member(s) who will submit the data for V2
- The *Condo Stack Tool* from <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>

1 Install the Tool

1.1 Download and install

Download the zipped package with the *Condo Stack Tool*, an ArcPy script tool, from <http://www.sco.wisc.edu/images/stories/publications/V2/tools/>.

Unzip to the directory of your choice. Then open ArcCatalog and navigate to the new directory. You should see toolboxes, labeled with their respective ArcGIS version compatibility. Choose the toolbox that fits your ArcGIS install. If you do not see any toolboxes, hit F5 to refresh the directory.

The tool should be ready to run.

2 Configure the Tool

2.1 Understanding the fields

As described previously, the *Condo Stack Tool* was designed for counties that do not represent their condos in a stacked format. Since there are many other ways to model condos, it is important to first understand how the tool's fields align with the data being used.

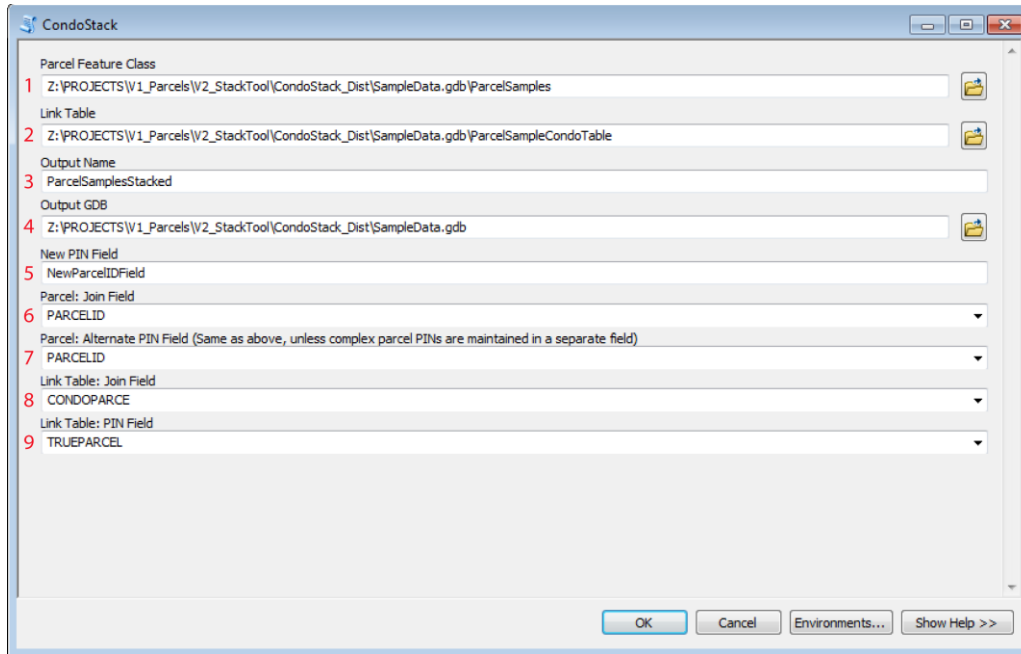


Figure 4. The Condo Stack Tool Interface

- 1. Parcel Feature Class:** Parcels to be stacked are included in the parcel feature class. The tool is capable of taking a subset feature class (e.g. only a feature class of condos subset from the county's parcel feature class) or a mixed feature class containing condos and that will not be stacked (i.e. a county feature class). The tool output will contain stacked parcels as well as any non-stacked parcels provided.
- 2. Link Table:** The link contains information required for stacking. In this table should be a field with pins that link a record to the parcel feature class (these values are duplicated for condos) and a field containing tax keys (these values should be unique and not duplicated). The link table is often a separate reference table constructed for condos and/or other complex parcels. However, it is also possible for the tax roll to contain a field for the parcel pins. The *Condo Stack Tool* relies on the information in the link table to perform the duplication.
- 3. Output Name:** Choose a name for the output feature class that will be created containing the stacked parcel geometries.
- 4. Output GDB:** Name of the output file geodatabase the new feature class will be created in.
- 5. New PIN Field:** Name of the new PIN field that will be created in the new feature class.
- 6. Parcel: Join Field:** The joining field from the parcel feature class.
- 7. Parcel: Alternate PIN Field:** Same as the above field ("Parcel: Join Field"), unless complex PINs are maintained in a separate field. If so, identify that field here.
- 8. Link Table: Join Field:** The joining field from the link table.
- 9. Link Table: PIN Field:** The field that contains the unique tax keys or PINs for the newly created stacked parcels.

③ Run

3.1 Execute the tool

Once all of the parameters have been correctly identified, proceed with running the tool.

④ Check the Newly Created Feature Class

4.1 Check to ensure that new stacked parcels have been created

After the tool has completed, the easiest way to check on the success of the tool is to compare the total feature count from the original feature class to the total feature count from the newly created feature class. This new feature count should be larger than the original. How much larger depends on the number of additional stacked parcels that were added to the new feature class. If you have a rough idea of the number of condos in your county, you would expect the parcel count to have increased by roughly that many features.

Troubleshooting for Condo Stack Tool

My tool does not show-up in the directory I have unpacked to.

If the tool does not appear in the directory you have unpacked it to, first try refreshing the directory in ArcCatalog (Right click directory » Refresh...). If the problem continues, it may be because you are using a legacy version of ArcGIS (ArcGIS 9.0 – ArcGIS 10.1). Tools supporting ArcGIS legacy versions are available at <http://www.sco.wisc.edu/images/stories/publications/V2/tools>.

I get an error when attempting to run the tool.

If running the tool results in an error, first ensure that the tool runs correctly on the test data provided in the zipped package.

- If the tool runs successfully over the test data, then double-check your XML for errors against the DOR XML schema. Remember, your data must be validated by DOR to work with this tool.
- If the tool does not run successfully over the test data, try to interpret the error message in finding a solution and submit the error message (via screen capture or cut and paste) to Codie See at csee@wisc.edu.

Who can I contact for help?

Codie See, State Cartographer's Office, 608-890-3793, csee@wisc.edu