VALIDATION AND SUBMISSION TOOL GUIDE

Version 4 Statewide Parcel Map Database Project

January 22, 2018

Guide Contents

☐ VALIDATE W/VALIDATION AND SUBMISSION TOOL

OVERVIEW OF TOOL 1
- Before Starting 2
- Troubleshooting 2
- About the Tool 3
- Glossary of Associated Files 4

TEST MODE 5
- 1 Overview of TEST Mode 5
- 2 Interpreting Geometric Issues 5
- 3 Workflow/Sequence for Edits 6
- 4 Resolving Broad-level Geometric Errors 7
- 5 Resolving In-line Geometric Errors 8
- 6 Resolving Broad-level Attribute Errors 9
- 7 Resolving In-line Attribute Errors 12
- 8 Repeat Sequence (As Needed) 16

FINAL MODE 17
- 9 Overview of FINAL Mode 17
- 10 Inputting the *Explain-Certification.txt* file 18
- 11 Saving the Mandatory .ini Submission Form 19

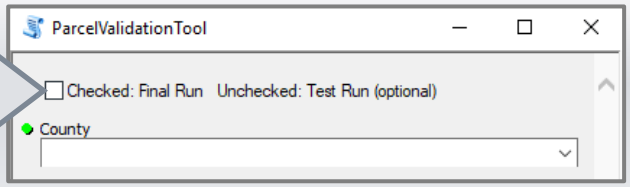
☐ ZIP & SUBMIT

SUBMIT .INI SUBMISSION FORM + DATA 20

Overview of Tool

☐ Leave box **UNchecked** to run tool in **TEST** Mode

☑ **Check** box to run tool in **FINAL** Mode



TEST mode = For PREPARING/TESTING/VALIDATING a parcel dataset
- Find and fix errors within a Searchable Format parcel submission
- Provides easy-to-use descriptions of each error so that the proper resolutions can be implemented

FINAL mode = For CREATING FINAL .INI SUBMISSION FORM for parcel dataset
- **Check parcel dataset submission**, validating the existence, degree, and nature of any errors
- **Enter *Explain-Certification.txt* attachment**, which explains any of your special circumstance
- **Create the .ini submission form**, which *must* be included in final zipped submission package

Before Starting . . .

Gather Materials

- The Submission Documentation and V4 webpage
- The county staff member(s) who will submit the data
- The parcel dataset for submission, with data **already prepared to the schema standards**

Install the Tool

- Install the updated Validation and Submission Tool, downloaded from www.sco.wisc.edu/parcels/tools
  - The *Validation and Submission Toolbox* contains an ArcPy script tool.
  - **Unzip** the downloaded zip package to the directory of your choice.
  - Place the tool within a directory that is close to your parcel submission working directory.
  - **Open ArcMap or ArcCatalog** and navigate to the newly created directory containing the unzipped files. You should see a toolbox (if you do not see any toolboxes, hit **F5** to refresh the directory).
  - The tool should be ready to run upon unzipping.
  - If you experience trouble starting or executing the tool, see the Troubleshooting section below.

Get Ready to Run the Tool

- **First, FORMAT YOUR DATA to the schema standards**
  - The Validation Tool's TEST mode is meant to be executed on an parcel layer that is as-close-as-possible to the schema specifications.
  - An excessive number of errors will likely be found on a parcel dataset that does not closely adhere to the schema, even if it is an well designed parcel dataset on its own.
  - The tool is designed to test against the statewide parcel schema only.
  - Testing a dataset before the proper preparations have been applied may cause the tool to not complete execution, as it is designed to gracefully exit execution if the requirements are not intact in the tested feature class.
  - Running the tool in TEST mode before the data is properly prepared may also cause an excessive number of simple errors to be flagged, which can be overwhelming and may obfucscate more important errors.

- **Test the tool**
  - To test how the tool works, you try it on the test data, included within the zipped tool package.

- **Run the "Null Fields and Set to UPPPERCASE Tool"**
  - Run the Null Fields and Set to UPPERCASE Tool.
  - The statewide parcel schema specifies that leading and trailing white spaces should **not** exist.
  - All strings should written in UPPERCASE.
  - Empty string cells (e.g. " ") should be annotated as true SQL <Null> values (and **not** a string of text characters that spells out the word "NULL").
  - The tool will flag any instance of the these errors and these types of errors can add up quickly..
  - Executing the Null Fields and Set to UPPERCASE Tool **before** running **Validation tool** can thus help avoid excessive errors.

Troubleshooting

My tool does not show-up in the directory I have unpacked to.
- If 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, contact the State Cartographer's Office (SCO).

I get an error when attempting to run the tool.
- Restart ArcCatalog if you continue to get errors. A file lock may be a cause of the error.
- Right-click » Refresh on the tool's toolbox, may help resolve an issue.
- If running tool results in an error, ensure the tool runs correctly on a test dataset, available in tool's zipped package.
- If the tool does not run successfully over the test data, try to interpret the error message (which is the resulting red text within tool's dialogue). Submit the error message (via screen capture or cut and paste) SCO.

I get a properties dialogue instead of a tool input dialogue when opening the tool.
- Refresh the directory in ArcCatalog (hit F5 or *right-click* » *Refresh* to refresh the directory).

Who can I contact for help?
- David Vogel, SCO, djvogel2@wisc.edu | Codie See, SCO, 608-890-3793, csee@wisc.edu

About the Tool

About TEST Mode

Executing the tool in TEST mode will allow you to assess your parcel submission and **determine specifically what improvements need to be made** to the dataset before zipping up the files and submitting the data.

TEST mode is **designed to flag all conditions that violate schema specifications** and **note them in two places**:

- **Summary file called *ValidationSummary.txt*** – A file called *ValidationSummary.txt* is a summary file named and written to the directory specified by the user in the tool's dialogue.
- **In-line comments** – Comments that are automatically created **within the output feature class**, whose output location is also specified by the user in the tool's dialogue.

The purpose of TEST mode is to eliminate errors—by helping to pinpoint them—and thus direct the user to take the appropriate actions to correct them.

Resolving errors may necessitate the execution of one of the Project's tools, custom automated solutions, and/or manual edits.

It is anticipated that **the Validation Tool may need to be executed several times in TEST mode** in order to troubleshoot all errors that may exist. The tool should be executed subsequent times until all possible fixes to errors have been resolved. The tool should always be executed one final time in TEST Mode before proceeding to FINAL mode.

Some errors uncovered within TEST mode may not be possible to resolve. **If any errors remain because it is not possible to resolve them, they must be explained within the *Explain-Certification.txt* file** that is entered when running the tool in FINAL mode.

ValidationSummary.txt file

The error status is always summarized within the output *ValidationSummary.txt* file, pictured in Figure 1. This file provides a general overview of the condition of the dataset. It summarizes error status for "In-line errors" and displays in full "Broad-level errors."

In-line errors are detailed in an output feature class

Broad-level errors are summarized in the text of this *ValidationSummary.txt* file

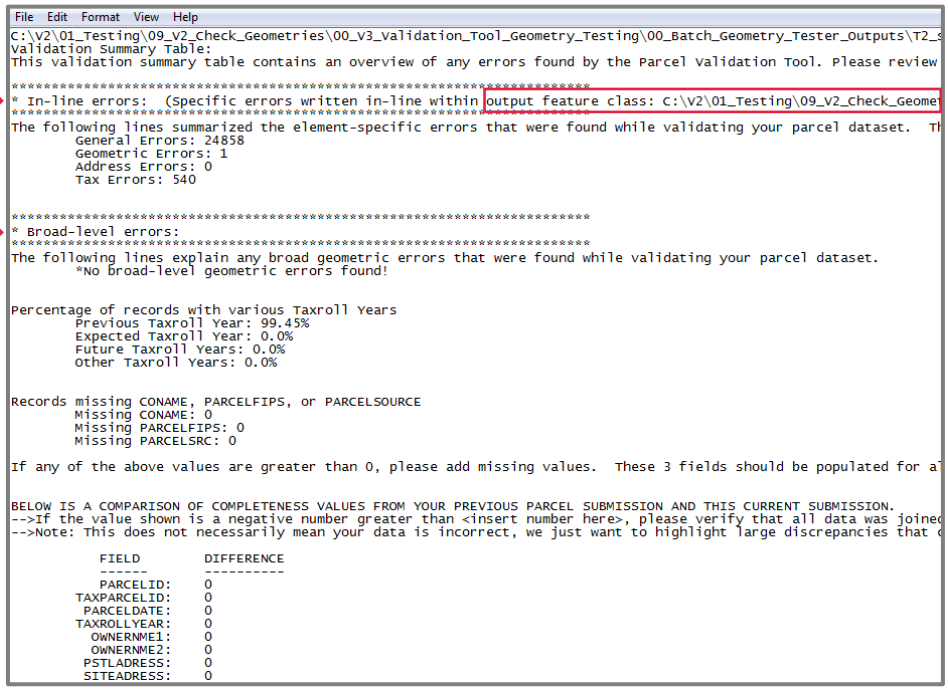


Figure 1. ValidationSummary.txt file (example). This summarizes error status for "In-line errors" and displays in full "Broad-level errors"

Once the parcel data is ready for submission—as determined by an error-free TEST mode run and a corresponding *ValidationSummary.txt* file that tells you no errors have been found, the tool can then be executed in FINAL mode.

About FINAL Mode

FINAL mode walks you through prompts that build the mandatory *.ini submission form*. The *.ini submission form* is an automatically created **.ini** file that is named with

- The county name + the output file name, which is determined by the user:
- e.g., "**DANE_Final1.ini**" – where "DANE" is the county specified and "**_Final1**" is the output file name chosen by the user.

The tool need only be executed once in FINAL mode. The only output for FINAL mode is the *.ini submission form*, which must be zipped up with the GIS files for submission.

Tool Interface

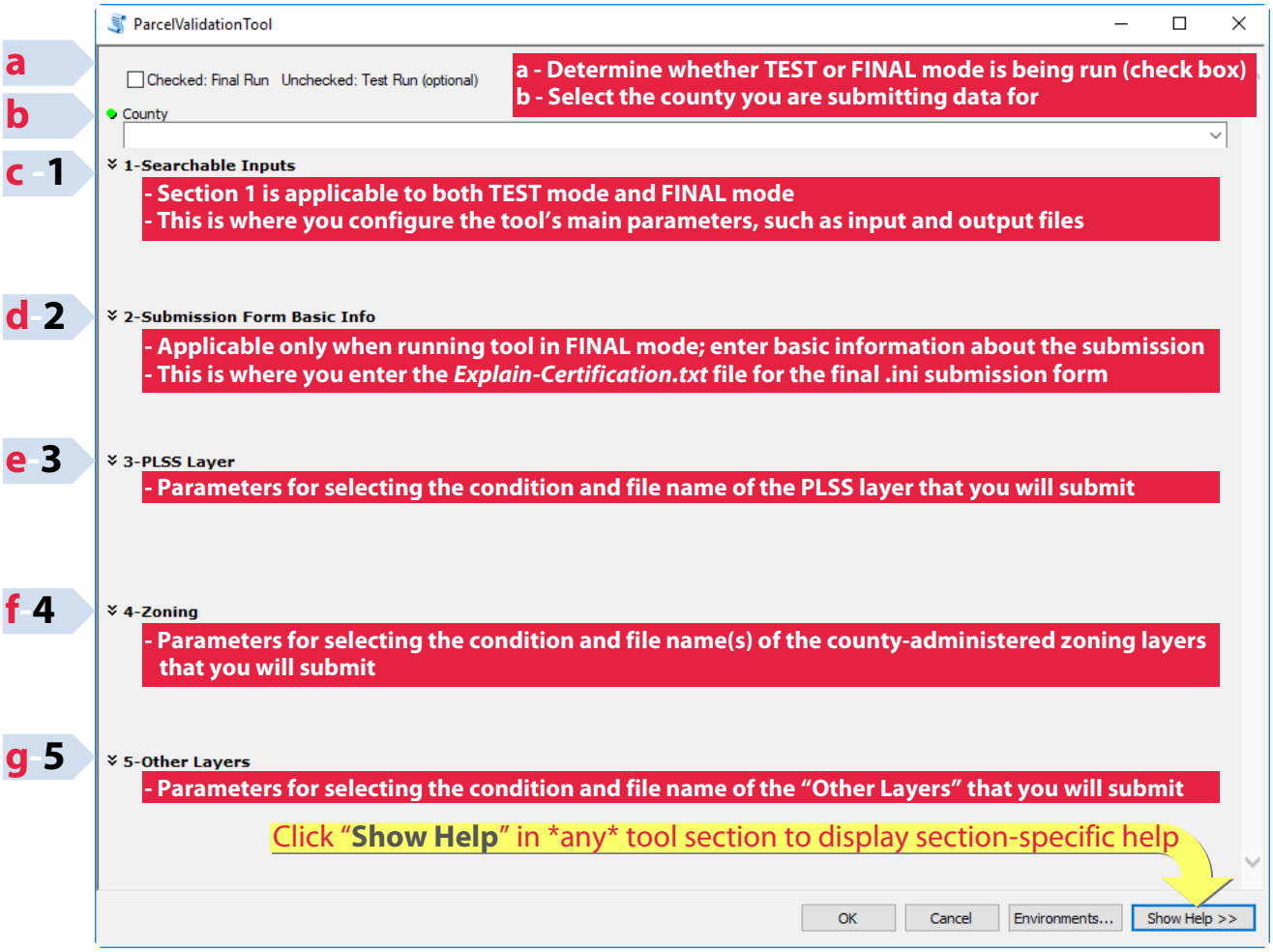


Figure 2. Tool Interface

Glossary of Associated Files

ValidationSummary.txt – A text file that results when you run the tool in TEST mode. It summarizes errors.

Explain-Certification.txt – A plain text file that the county LIO fills out and saves, to later enter in the tool. The template *Explain-Certification.txt* file can be found at the root level of the tool's zipped package. It is where an explanation of any special situations and unresolved errors should be placed. It is entered into the tool as an attachment. This attachment is entered in the "2-Submission Form Basic Info" section of the tool when executing in FINAL mode. If there are no issues, type "NONE" in *Explain-Certification.txt.*

.ini submission form – An .ini format file that is the result of successfully running the tool in FINAL mode. This file provides all of the information you have inputted in FINAL mode (including the *Explain-Certification.txt*). It is **MANDATORY for all submissions**.

Output Feature Class – The feature class that results from running the tool in TEST mode, where errors can be viewed as values within the attribute table as "In-line errors," written in-line, inside the attribute table of output feature class.

TEST MODE

1 Overview of TEST Mode

TEST mode may yield several thousand errors or merely a handful of errors upon its first running. The number of errors will be influenced by the size of the parcel dataset, its adherance to schema specifications, and the amount of preparation that has already been applied to the layer.

1.1 Finding errors manually

The Validation Tool will check your data for errors, but it is advisable to manually check for errors first. A manual check is the process that a county may have employed in the past to double-check that the parcel submission adheres to the specs of the Submission Documentation. Ensure that the dataset is vetted with a **manual error check** before moving to the next step.

1.2 Finding errors via the Validation Tool

The Validation Tool is designed to **identify** common file, geometric, and attribute issues.

- **Geometric Issues.** The geometric assessment results can be found within the output *ValidationSummary.txt* file and within the geometric error field ("GeometricElementErrors") of the tested parcel feature class.
- **Attribute Issues.** The attribute assessment results can be found within the output *ValidationSummary.txt* file and within the general error field ("GeneralElementErrors"), address error field ("AddressElementErrors"), or tax error field ("TaxElementErrors") of the tested parcel feature class.

2 Interpreting Geometric Issues

2.1 Definitions for Types of Geometric Issues

Geometric errors may be classified in one of three locations pending the nature of the error: Immediate errors, In-line errors, or Broad-level errors.

Immediate errors

- "Immediate" errors exist on parcel feature classes under conditions where the feature class cannot be further tested by the tool without resolving them.
  - ***Example***: There is only one immediate geometric error type. It happens when the dataset is projected to a Coordinate Reference system that differs from that of the statewide schema. If this condition is found, the tool will automatically stop executing.

In-line errors

- "In-line" errors exist on individual parcel features within the dataset and apply to only the specific feature at the given OBJECTID within the feature class.
  - ***Example***: A parcel geometry was found that has no length value or no area value, thus indicating that it contains a "Null" geometry or invalid geometry
- Notes that describe the nature of In-line errors can be found **within the output feature class** that is created by running the tool in TEST mode. These error notes will exist in the "GeometricElementErrors" field that is automatically created when running the tool in TEST mode. A summary of the number of In-line geometric errors found is also written to the *ValidationSummary.txt* file for the purpose of quick reference when reviewing the results.

```
5  ********************************
6  * In-line errors:  (Specific err
7  ********************************
8  The following lines summarized t
9      General Errors: 0
10     Geometric Errors: 0
11     Address Errors: 0
12     Tax Errors: 101
```

Figure 3. In-line errors, summarized from ValidationSummary.txt

Broad-level errors

- "Broad-level" errors are broader in nature and typically apply to an entire parcel feature class or a large portion of a feature class.
- Notes that describe the nature of Broad-level errors can be found **within the *ValidationSummary.txt* file** under the "Broad-level errors" section.
- The summary file will read "***No broad-level geometric errors found!**" if no errors of this type are found when running in TEST mode.
- Note that Broad-level errors will **not** be found within the "GeometricElementErrors" field, as this type of error is not specific to any one feature (rather, they apply to many features or the feature class as a whole).

3 Workflow/Sequence for Edits

3.1 Recommended sequence of editing

Three levels of errors may be found when running the tool in TEST mode. It is recommended you resolve these issues in the following order, but please note this workflow is only one of many possible acceptable workflows:

1) **Immediate errors – Resolve 1st**
   - General Errors
   - Geometric Errors (NOT APPLICABLE)
   - Address Errors (NOT APPLICABLE)
   - Tax Errors (NOT APPLICABLE)
2) **Broad-level errors – Resolve 2nd**
   - General Errors
   - Geometric Errors
   - Address Errors
   - Tax Errors
3) **In-line errors – Resolve 3rd**
   - General Errors
   - Geometric Errors
   - Address Errors
   - Tax Errors

Immediate errors – Resolve 1st

Immediate errors are errors found within the data that **have a direct effect on the proper execution of the tool**. If an error of this nature is found, the tool will gracefully stop executing and alert the user of the problem through the tool's dialogue and within the *ValidationSummary.txt* file.

Generally, no other errors will be listed when this type of error is encountered. The user must resolve the immediate error, using the directions provided in the tool's dialogue box or *ValidationSummary.txt* file before proceeding with the next execution of the tool in TEST mode. Once the immediate error is resolved, the tool can be executed again for Broad-level and In-line errors.

Broad-level errors – Resolve 2nd

Broad-level errors are recommended for resolution after immediate errors, but before In-line errors. This is due to the nature of Broad-level errors, which **affect a great number of features**. The various resolutions to Broad-level errors tend to be well suited for batch troubleshooting. Executing resolutions on large batches of features or the entire feature class may positively impact the number of In-line errors, which are can require more manual intervention to resolve.

In-line errors – Resolve 3rd

In-line errors are recommended for resolution last. Remember that In-line errors are found within the output feature class, in the attribute table.

Recommendation for Handling Error Types

At the above error levels, there are four categorical types of error. The condition of the data, software used, and other factors may cause a more obvious workflow to be desirable, but it is recommended that these types be handled in the following order:

I. **General Errors** – errors otherwise unclassified by the three types below
II. **Geometric Errors** – errors that are a function of parcel geometry
III. **Address Errors** – errors specific to address-related attribute elements
IV. **Tax Errors** – errors specific to tax roll-related attribute elements

4 Resolving Broad-level Geometric Errors

4.1 Reading Broad-level results within the ValidationSummary.txt file

Broad-level geometric errors come in the following forms. Please read the directives associated with each error to begin troubleshooting.

Geometric File Errors

Geometric file errors will occur if the parcel feature class appears to have file qualities that do not meet submission standards. The following types of Geometric file errors or flags might be thrown:

- **Error**: **Invalidated Coordinate Reference System**
  - **Nature**: Occurs when the feature class's coordinate reference system (CRS) cannot be accurately validated by the Validation Tool
  - **Fix**: Follow the Parcel_Schema_Field_Mapping_Guide to project native data to the Statewide Parcel CRS and ensure that the feature class adheres to the CRS parameters as so:
    - Datum: NAD_1983_HARN_Wisconsin_TM
    - WKID: 3071
    - Authority: EPSG
    - Projection: Transverse Mercator
    - False Easting: 520000.0
    - False Northing: -4480000.0
    - Central Meridian: -90.0
    - Scale Factor: 0.9996
    - Latitude of Origin: 0.0
    - Linear Unit: Meter (1.0)

- **Error**: **Geometry Type Error**
  - **Nature**: Occurs when the feature class contains non-polygonal features, such as points or lines. Only polygon feature classes should be submitted.
  - **Fix**: Prepare a feature class containing polygon parcel features

Geometric Misplacement Flag

A Geometric Misplacement Flag will occur if parcel geometries appear to be spatially misplaced when comparing them against the previous statewide parcel version's parcel geometries. This existence of this issue is indicative of a **re-projection error** but could also be caused by other reasons.

- **Purpose of this test**: This test calls attention to the possibility that parcel geometries might have had geometric error introduced. Past parcel submissions have occasionally contained subtle to significant degrees of dataset-wide geometric misplacement.
  - This condition often results from merging the local parcel dataset with its native coordinate reference system, with the statewide parcel template (GISTemplates.zip), which has the statewide coordinate reference system applied.
  - To best preserve the geometric quality of the local parcel dataset, the local dataset should first be projected from its native CRS to the CRS of the statewide parcel layer, and *then* merged into the template file.
  - See Parcel_Schema_Field_Mapping_Guide on how to project data to the statewide parcel CRS.
- **How is this error tested?** Within the logic of the Validation Tool, parcel geometries of the submitter's parcel dataset are checked at ten evenly dispersed intervals across the parcel dataset. Each of these tests involve querying the most recent statewide parcel feature service by PARCELID, accessing the geometric properties of the queried parcel, and comparing parcel centroids of each polygon. If the two parcel centroids contain more than a 0.01 meter difference, then the test will fail. If all ten of these tests fail, the Geometric Misplacement Flag is thrown.
- **False positives**: Given the nature of this test's logic as described, there is a possibility of a false positive test for geometric misplacement. The following may be reasons a geometric misplacement error may be incorrectly displayed:
  - Any reason that all, or a significant number of parcel features, were spatially adjusted over the course of the year.
  - The PARCELID within the parcel dataset does not match that of the parcel ID submitted for the previous year.
  - The parcels submitted to the previous version of the Project were misplaced and were not corrected when preparing the statewide layer.
  - **If any of these conditions for false positives are true, or another exceptional condition exists, make note of this** in the *Explain-Certification.txt* section of the tool's FINAL mode under "2-Submission Form Basic Info."
- **Validating and correcting this issue**: If this flag is thrown and it is clear that the false positives above are not explanations of the cause, then the following steps are recommended:
  - **Validating**
    1) Open a new ArcMap session and bring up the parcel feature class
    2) Add an orthorectified imagery layer—use the best/most reliable layer available

3) Add the native parcel dataset—make sure that the native layer has the county coordinate reference system applied.
4) Change each of the two parcel layers symbology to have no fill and stroke with two distinct colors.
5) Zoom-in on parcels across several different areas across the county, check to ensure that the geometries of the two parcel datasets align (it may be necessary to zoom in beyond 1:1 scale).
6) If there is a difference between the location of the two parcel datasets when inspecting parcels across the county, then a geometric misplacement exists.

- **Correcting**
  1) If the parcel dataset was prepared using steps akin to merging the native parcel dataset with the GISTemplate without first projecting the native data to the statewide coordinate reference system, then this is likely the cause of the geometric misplacement.
  2) Follow the steps outlined in the Parcel_Schema_Field_Mapping_Guide to re-prepare the parcel geometries.

5 Resolving In-line Geometric Errors

5.1 Reading results within the "GeometricElementErrors" field

Geometric errors that are specific to an individual feature will be written in-line within the *GeometricElementErrors* field. This field is automatically created when running the tool in TEST mode. **Open the output feature class in ArcMap after running this Validation Tool and sort this field descending on the *GeometricElementErrors* field.** Please read the directives associated with each error to begin troubleshooting.

Error and Flag Overview

The following types of "geometric element errors" or "flags" might be thrown when running a validation. These results will be reflected within the GeometricElementErrors field.

- **Error/Flag: Sliver Polygon**
  - **Nature**: This type of flag will occur if the parcel's geometry appears to have the geometric qualities of a sliver polygon. Specifically, if the polygon has any of the following:
    - "Sliver polygon: AREA" → area < 0.01 Meters
    - "Sliver polygon: LENGTH" → length < 0.01 Meters
    - "Sliver polygon: AREA/LENGTH" → area / length < 0.01 Meters
  - **Fix**: *Note that this type of issue is annotated as a "flag" and thus not necessarily an indication of an error. In many parcel datasets, this type of sliver polygon is used to maintain a spatially contiguous parcel dataset, even though there may be small areas where parcels did not COGO together perfectly. While these types of features are not preferred, they are acceptable by the statewide schema standards, as long as one of the following actions are applied:
    - ***If the sliver exists intentionally as a way to fill a geometric gap in the parcel dataset***, the parcel should be annotated as "GAP" within the PARCELID field.
    - ***If the sliver exists as an unintentional fragment of another parcel***, the parcel should be geometrically merged with the parcel that shares a common PARCELID.

- **Error/Flag: Corrupt Geometry**
  - **Nature**: This type of flag will occur if the parcel's geometry cannot be accessed within the script. Use the following directives accordingly:
    - ***"Corrupt Geometry: The feature's area and/or length could not be accessed."***
      Or
    - ***"Corrupt Geometry: The feature's geometry could not be accessed."***
      → A rare general error that occurs when a parcel's geometry could not be accessed. This type of error is indicative of a corrupt parcel geometry and is general in nature. As this is a general error, there may be various reasons that the error was thrown.
  - **Fix**: If either of these error messages are found on a feature, try the following techniques to troubleshoot (in the following order):
    - Navigate to the feature within ArcMap by selecting the feature and choosing to zoom to it. If the act of zooming to the feature brings you to no result, or to some location outside of the county, then the parcel likely has a null geometry. Features with null geometry should be deleted or re-created.
    - If navigating to the feature by selecting the feature and zooming to it brings you to a parcel feature, explore the nature of the parcel geometry.
      - Does it self-intersect?
      - Does it contain more than 2 vertices?
      - Do any of its vertices exist perfectly on top of one another?
      - Is it represented as existing within the county boundaries?

- Is it a multi-part feature? If this is a multi-part feature, the same issues should be examined on each part of the multi-part feature.
    - Does it self-intersect?
    - Does it contain more than 2 vertices?
    - Do any of its vertices exist perfectly on top of one another?
    - Is it represented as existing within the county boundaries?
- *If any of the above are true, actions should be taken to correct the issue, the feature should be deleted, or re-created.

6 Resolving Broad-level Attribute Errors

6.1 Reading Broad-level results within the ValidationSummary.txt file

Broad-level attribute errors will come in the following forms within the *ValidationSummary.txt*. Please read the directives associated with each error in the summary file to begin troubleshooting.

```
C:\Users\csee\Documents\GitHub\V3ValidationTool\TestData\Sample_Searchable_Submission\OutputFiles\WAUK8_ValidationSummary.txt
Validation Summary Table:
This validation summary table contains an overview of any errors found by the Parcel Validation Tool. Please review the contents of this file and make changes to yo

**********************************************************************
* In-line errors:  (Specific errors written in-line within output feature class: C:\Users\csee\Documents\GitHub\V3ValidationTool\TestData\Sample_Searchable_Submissi
**********************************************************************
The following lines summarized the element-specific errors that were found while validating your parcel dataset.  The stats below are meant as a means of reviewing
    General Errors: 0
    Geometric Errors: 0
    Address Errors: 0
    Tax Errors: 101


**********************************************************************
* Broad-level errors:
**********************************************************************
The following lines explain any broad geometric errors that were found while validating your parcel dataset.
    *No broad-level geometric errors found!


Percentage of records with various Taxroll Years
    Previous Taxroll Year: 100.0%
    Expected Taxroll Year: 0.0%
    Future Taxroll Years: 0.0%
    Other Taxroll Years: 0.0%


Records missing CONAME, PARCELFIPS, or PARCELSOURCE
    Missing CONAME: 0
    Missing PARCELFIPS: 0
    Missing PARCELSRC: 0

If any of the above values are greater than 0, please add missing values.  These 3 fields should be populated for all records submitted.


BELOW IS A COMPARISON OF COMPLETENESS VALUES FROM YOUR PREVIOUS PARCEL SUBMISSION AND THIS CURRENT SUBMISSION.
-->If the value shown is a negative number greater than <X>, please verify that all data was joined correctly and no data was lost during processing.
-->Note: This does not necessarily mean your data is incorrect, we just want to highlight large discrepancies that could indicate missing or incorrect data.

        FIELD       DIFFERENCE
        ------      ----------
     PARCELID:      -153562
   TAXPARCELID:     -153562
    PARCELDATE:     100
   TAXROLLYEAR:     -154177
     OWNERNME1:     -154177
     OWNERNME2:     -10298
    PSTLADRESS:     -152544
     SITEADRESS:    -144156
  ADDNUMPREFIX:     -62955
       ADDNUM:      -142937
  ADDNUMSUFFIX:     -214
       PREFIX:      -31433
    STREETNAME:     -144242
```

Figure 4. Broad-level Attribute Errors – Summarized in ValidationSummary.txt below " Broad-level errors" Heading*

Note that in TEST mode, the Validation Tool is designed to compare some portions of the Searchable Format submission you are testing for against previous data specific to YOUR county (that being the standardized county data from the previous statewide parcel layer), or values that might be "expected" based on the statewide schema.

Percentage of records with various Taxroll Years

- This heading displays results of a summary on your V4 test file's TAXROLLYEAR attribute.
- It is expected that the majority of tax records should be the year prior to the year in which a parcel submission takes place.
  - For V4—with the data request in 2018—the majority of tax records should be "**2017**." (There are cases where an exception can be made to this rule, for cases of parcel splits/new parcels, as detailed in the Submission Documentation, under the attribute definition for TAXROLLYEAR.)
- The "Percentage of records with various Taxroll Years," and **ALL error messages must be interpreted in the context of YOUR dataset**. A human with knowledge of the native data can interpret the output best.
- Consider the sample report below, followed by an explanation of the various types of errors you might get:

```
Percentage of records with various Taxroll Years
      Previous Taxroll Year: 0.0%
      Expected Taxroll Year: 63.1%
      Future Taxroll Years: 17.0%
      Other Taxroll Years: 0.0%
```

- **Error/Flag: All 4 stats add up to less than 100%**
  - **Nature**: Occurs when there are <Null> values in the TAXROLLYEAR field; <Null> values are not counted within this summary.
  - **Fix**: Ensure that all <Null> values exist because they are parcel splits/new parcels, or non-parcel features such as GAP/ROW/HYDRO, etc. If they are not parcel splits/new parcels or non-parcel features, then they should have a valid tax roll year associated with them.
- **Error/Flag: Any value higher than 0.0% exists in "Other Taxroll Year"**
  - **Nature**: Occurs when there are TAXROLLYEAR values that are anything other than the previous, future, or expected taxroll year. For example, in the case of V4 project "2015" is not valid and would be flagged here.
  - **Fix**: Ensure that all TAXROLLYEAR values are valid and make sure to update other attributes appropriately so that this data is of the appropriate vintage. If TAXROLLYEAR values cannot be of the appropriate vintage, please make an explanation of this in the *Explain-Certification.txt*.
- **Error/Flag: Any value higher than 0.0% exists in "Previous Taxroll Years"**
  - **Nature**: Occurs when there are TAXROLLYEAR values that meet the previous tax year, future, or expected tax roll year. For example, in the case of V4 project "2016" is not valid and would be flagged here. The majority of V4 tax roll year values should be "2017"
  - **Fix**: Ensure that all TAXROLLYEAR values are valid and make sure to update other attributes appropriately so that this data is of the appropriate vintage.
- **Error/Flag: An exceptionally large value exists in "Previous Taxroll Years"**
  - **Nature**: Occurs when there are TAXROLLYEAR values that meet the previous tax year, future, or expected taxroll year. For example, in the case of V4 project "2016" is not valid and would be flagged here. The majority of V4 tax roll year values should be "2017"
  - **Fix**: Ensure that all TAXROLLYEAR values are valid and make sure to update other attributes appropriately so that this data is of the appropriate vintage.

Records missing CONAME, PARCELFIPS, or PARCELSOURCE

- This heading speaks to three fields in the parcel schema that **must be 100% populated**: CONAME, PARCELFIPS, and PARCELSOURCE.
- Consider the sample report below, followed by an explanation of the various types of errors you might get:

```
Records missing CONAME, PARCELFIPS, or PARCELSOURCE
      Missing CONAME: 4
      Missing PARCELFIPS: 100
      Missing PARCELSRC: 55
```

- **Error/Flag: Any of the three fields (CONAME, PARCELFIPS, or PARCELSOURCE) contain missing values**
  - **Nature**: Occurs when there are blank or <Null> records within any of these three fields
  - **Fix**: Ensure that all three of these fields are 100% populated with valid domains as defined in the Submission Documentation. You can isolate these records by sorting each field in descending order, and using field calculator to populate as appropriate. In most cases, all parcels in the dataset will have the same values for CONAME, PARCELFIPS, and PARCELSOURCE.

BELOW IS A COMPARISON OF COMPLETENESS VALUES FROM YOUR PREVIOUS PARCEL SUBMISSION AND THIS CURRENT SUBMISSION.

- It is expected that parcel submissions continue to grow in quality and attribute completeness, as well as natural increases in quantity of records.
- Significant differences, however, in the number of records populated from one submission to the next (e.g., from V3 to V4) are indications of possible error or possible improvement.
- The table associated with this feedback section of the *ValidationSummary.txt* file is **created by comparing your current submission against what was established in the previous year's parcel data** (the standardized V3 statewide parcel layer).

- Consider the sample report below, and the various types of errors you might get.
- **Error/Flag: VALUE IN "DIFFERENCE" COLUMN IS SIGIFICANT IN NEGATIVE DIRECTION**
  - **Nature.** In this example, OWNERNME1 has a significant reduction in populated values. This would be indicative of a problem within the OWNERNME1 field, indicating that it may need attention.
- **Error/Flag: VALUE IN "DIFFERENCE" COLUMN IS SIGIFICANT IN POSITIVE DIRECTION**
  - **Nature.** In this example, ZIPCODE has a significant increase in populated values. This difference could be indicative of an improvement in data.

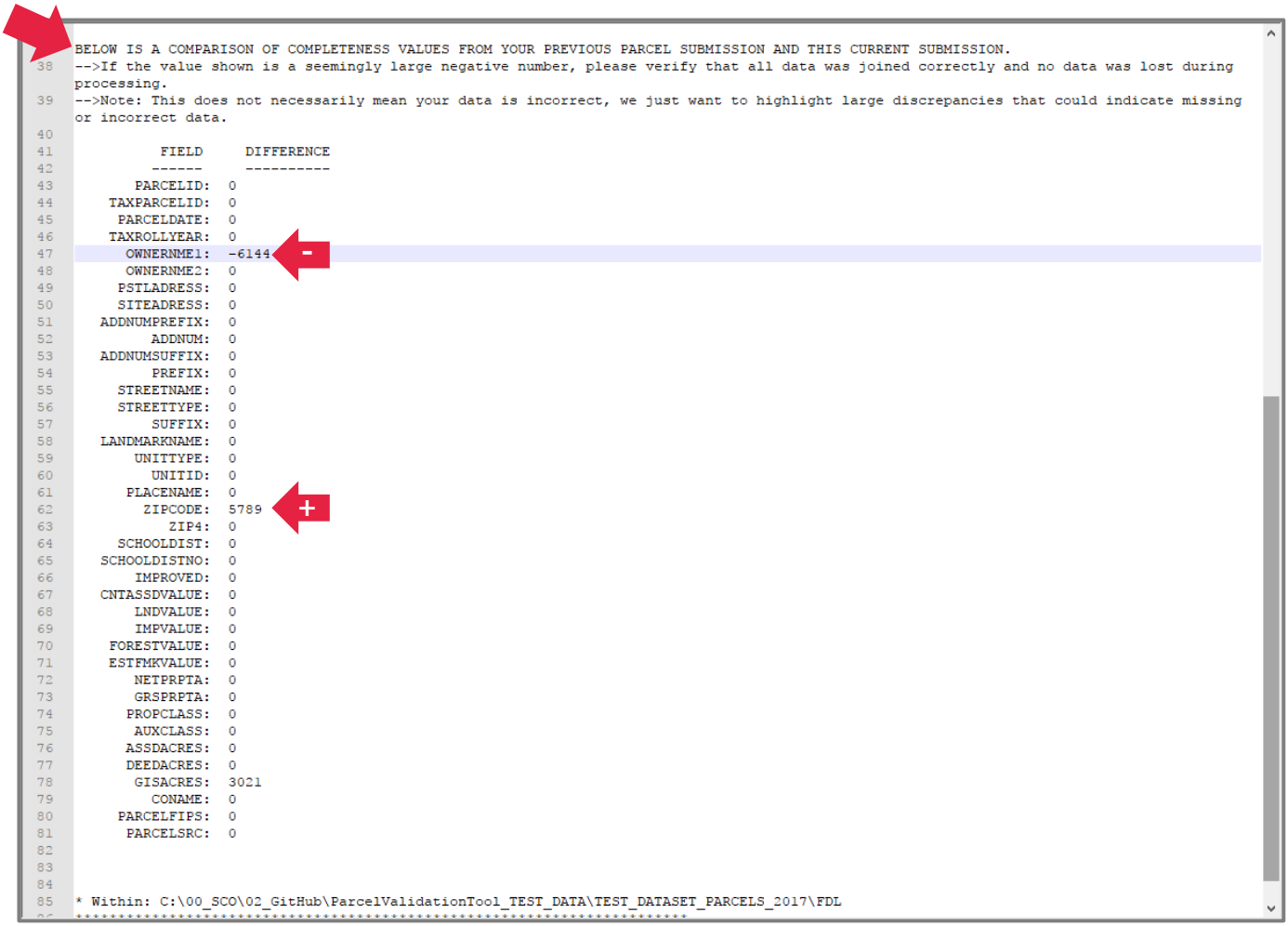


Figure 5. Sample completeness report

- **Error/Flag: Missing attribute values across any field listed.**
  - **Nature**: If a negative number is displayed within this table, then the annotated number represents how many fewer records are populated within the parcel layer being tested in contrast to the currently live parcel layer. If a positive number is displayed, it represents the number of new records that are populated. A negative or positive number does not necessarily mean your data is correct or incorrect, however, large discrepancies *could* indicate missing or incorrect data.
  - **Note that these statistics are compiled against the most recent and curated parcel data (in this case, the standardized V3 data).**
  - **Fix**: If any field's difference appears to be abnormally significant, review the field and identify whether this difference is due to missing or incorrect data.
    - **If difference is due to missing or incorrect data**: Fix the issue so that the most complete and contemporary data as possible is being submitted. If it is not possible to achieve this, then make a note within the *Explain-Certification.txt* section of the tool.
    - **If difference is not due to missing or incorrect data**: Make note of the explanation within the *Explain-Certification.txt* section of the tool.

7 Resolving In-line Attribute Errors

7.1 Errors in GeneralElementErrors/AddressElementErrors/TaxElementErrors fields

Attribute errors cover the bulk of expected and possible errors that the tool will find. Because of the quantity and nature of these types of issues, the issues are broken into three categories:

- **GeneralElementErrors** – Attribute errors that are not specific to address or tax elements
- **AddressElementErrors** – Attribute errors that are specific to address-related elements
- **TaxElementErrors** – Attribute errors that are specific to tax-related elements

7.2 Reading results in the GeneralElementErrors/AddressElementErrors/TaxElementErrors

- General, address, and tax in-line errors are specific to an individual parcel feature.
- These and are written **in-line** within their respectively named fields, **in the output feature class** that results from running the tool in TEST mode.
- Like GeometricElementErrors, these three fields are automatically created when running the tool in TEST mode.
- To read results:
  - After running the Validation Tool, open the output feature class in ArcMap.
  - **Sort in descending order** on each of these fields, ***one at a time***.
  - After sorting, there will be one or more messages on the lines at the top of the table, delimited by a pipe (|)
  - In the attribute table of the output feature class, read the directives associated with each error to begin troubleshooting.

7.3 Interpreting these directives

General directives are listed below. Note that error messages contain contextual descriptions that have large numbers of permutations. Thus, it is not possible to list all possible messages and the message descriptions below. This documentation intended to provide general guidelines.

Legend for syntax – For interpreting the error directives below

[*FIELD*] – This syntax represents a parcel schema field name as a way of annotating that this is a variable and could be interchanged for any field name.

[*ADDRESS FIELD*] – Similar to [*FIELD*], this syntax represents a standardized field name as a way of annotating that this is a variable. In this case, the field is specific to **address** element fields only.

<X> – Indicates a count of something, which could be any value—from small to very large.

- **Error/Flag: An unknown issue occurred with the [*FIELD*] field. Please inspect this field's value**
  - **Nature**: This type of "general element error" or "flag" might be thrown when an unexpected condition exists within the annotated field. As the message states, the issue is unknown and uncommon. See the fix below for strategies to resolve this.
  - **Fix #1**: *Note that this type of issue is annotated as a "flag" and thus not necessarily an indication of an error. These records should be inspected within their respective field. To inspect a record, open an Editing session in ArcMap on the output feature class, and navigate to the record in question to explore:
    - **Does anything appear abnormal about the content of the cell?**
    - **Copy the record from the cell and paste it into a text editor**
      - **Do any new characters appear?**
      - **Are there new lines below the text?**
      - **Are there characters on new lines?**
    - If any of the above conditions, or other strange/undesirable conditions exist within the cell, make the appropriate corrections. If no obvious solution appears, proceed to fix #2.
  - **Fix #2**: If there are no obvious problems with the record, check the record to make sure that it is meeting s statewide schema specifications. If there is no obvious reason that this error exists, note the following within the *Explain-Certification.txt* as a part of your submission.
    - *--There are **<X>** instances of the following error "unknown issue occurred with the ADDNUM field. Please inspect this field's value." These issues are unexplainable by the county LIO."*

- **Error/Flag: Value in [*FIELD*] doesn't appear to be a numeric value**
  - **Nature**: Occurs when there are non-numeric elements existing within the field when only numeric values should exist.
  - **Fix**: Ensure that all values within this field are numeric. Correct the issue by starting an editing session in ArcMap to type in a valid value, or use the *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution, or re-joining anew may be warranted.

- **Error/Flag: Null Found on [*FIELD*]**
  - **Nature**: Occurs when a <Null> value was found within a field that should not contain <Null> values.
  - **Fix**: Ensure that all values within this field are free of <Null> values. Correct the issue by starting an editing session in ArcMap to type in a valid value, or use *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution, or re-joining anew may be warranted.

- **Error/Flag: Appears to be a duplicate value in [*FIELD*]**
  - **Nature**: Occurs when a duplicate value was found within a field that should never contain duplicate values. **NOTE:** This error message will flag the second and all subsequent instances of duplicate values present, but does not flag the first occurrence of a value. Rights of way, hydrography, gaps, and other non-parcel features are, by design, not included in this assessment. However, if items similar to these non-parcel features are flagged, their error can be disregarded and noted as an exception in the *Explain-Certification.txt* file.
  - **Fix**: Ensure that all values within this field are not duplicative. Correct the issue by starting an editing session in ArcMap to type in a valid value, or use *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution, or re-joining anew may be warranted.
  - **NOTE**: It is possible that exploded multi-part parcels will be flagged by this tool. It is not required to dissolve single parcels represented by multiple non-contiguous geometries. If this condition is the reason why duplicate polygons exist, then these features may be submitted as they exist, but this must be reported within the *Explain-Certification.txt* file.

- **Error/Flag: Redundant information in TAXPARCELID and PARCELID fields.**
  - **Nature:** Occurs when TAXPARCELID values are duplicates of the values existing in the PARCELID field.
  - **Fix:** Ensure that this field is <Null> for all records unless the values are distinct/different from those in the PARCELID field.

- **Error/Flag: A value provided in the PSTLADRESS field may contain an incomplete address. Please verify the value is correct or set to <Null> if complete address is unknown.**
  - **Nature:** Occurs when a postal address value contains 'UNAVAILABLE', 'UNKNOWN', '00000', NONE', etc.
  - **Fix:** Include a full postal address if available. If this is not possible, correct the issue by starting an editing session in ArcMap, and convert values such as these to <Null>.

- **Error/Flag: Value provided in PLACENAME doesn't contain required LSAD descriptor.**
  - **Nature**: Occurs when a place name value does not include the necessary LSAD descriptor, such as:
    - **CITY OF**
    - **TOWN OF**
    - **VILLAGE OF**
  - PLACENAME examples:
    - CITY OF CHIPPEWA FALLS
    - TOWN OF MADISON
    - CITY OF MADISON
    - VILLAGE OF LAKE HALLIE
  - **NOTE** *All* tax parcels must have a PLACENAME value, even parcels that have not been assigned an address.
  - **Fix**: Ensure that all values within this field contain their appropriate LDAD descriptor. Correct the issue by starting an editing session in ArcMap to type in a valid value, or use *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution, or re-joining anew may be warranted.

- **Error/Flag: Value provided in [*FIELD*] not in acceptable domain list.**
  - **Nature**: Occurs when an invalid domain exists within the annotated field.
  - **Fix**: Ensure that all values within this field contain a valid value. Correct the issue by first checking the statewide schema definition for the field within the Submission Documentation. Also, consult the parcel domain list for a comprehensive list of domains on fields as appropriate. Use an ArcMap editing session or *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution may be desired, such as use of the Data Standardize Tool.

- **Error/Flag: Value provided in [*FIELD*] does not appear in list created from V4 data. Please verify this value is correct.**
  - **Nature**: Similar to the previous error, occurs when an unrecognized value exists within the annotated field. However, in this case the [*FIELD*] containing the value does not have a comprehensive list of domains. This is simply a flag to call attention to the field, so that a specialist can make the correct decision on the validity of the value.

- **Fix**: Check all values containing this flag and verify that they contain a valid value. There will not be a comprehensive list of domains for this field, so verifying the record means that it must be checked against the field's definition within the Submission Documentation. Also, consult the parcel domain list for a comprehensive list of domains on fields as appropriate. Use an ArcMap editing session or *Field Calculator* to correct the records. If this issue exists on a large number of records, a more automated solution may be desired, such as use of the Data Standardize Tool.

- **Error/Flag: [*FIELD*] is Null but [*FIELD*] is populated. Please ensure elements are in the appropriate field.**
  - **Nature**: This flag will be thrown when one of a selection of fields that relate to one another are found to be blank or <Null> when they should not be blank or <Null>. For example, if SITEADRESS is populated—indicating that a site address has been assigned to the parcel—then we would also expect that STREETNAME would be populated. Under this example, if SITEADRESS is populated and STREETNAME is <Null>, then this flag would be thrown.
  - **Fix**: Check all values containing this flag and verify that they contain a valid value in context of the fields that they are flagging. Verifying the record may mean that it must be checked against each field's definition within the Submission Documentation. Use an ArcMap editing session or *Field Calculator* to correct the records.

- **Error/Flag: Null Found on [*FIELD*] field and value is expected.**
  - **Nature**: This flag will be thrown when a value is found to be blank or <Null> when a value would otherwise be expected.
  - **Fix**: Check all values containing this flag and ensure that they contain a valid value in context of the fields that they are flagging. Verifying the record may mean that it must be checked against each field's definition within the Submission Documentation. Use an ArcMap editing session or *Field Calculator* to correct the records.

- **Error/Flag: Value provided in ZIPCODE is either not 5 digits long or does not appear to be a Wisconsin zipcode.**
  - **Nature**: This flag will be thrown if the zipcode value does not meet specifications of a Wisconsin zipcode, or if the value is not five digits in length.
  - **Fix**: Check all values containing this flag and ensure that they contain a valid zipcode value. Note that the Submission Documentation specifies that this field should contain the **site address** zipcode, so only Wisconsin zipcodes are valid. Owner mailing addresses do not belong in ZIPCODE. To make corrections, use an ArcMap editing session or *Field Calculator* to correct the records.

- **Error/Flag: Value provided in IMPROVED doesn't correspond with IMPVALUE for this record - please verify.**
  - **Nature**: This flag will be thrown if the IMPROVED value does not properly correspond with the value in the IMPVALUE field. Please consult the Submission Documentation on IMPROVED definition for considerations on how this field should be populated in relation to the IMPVALUE field.
  - **Fix**: To make corrections, use an ArcMap editing session or *Field Calculator* to correct the records.

- **Error/Flag: Bad characters found in [*FIELD*]**
  - **Nature**: This flag will be thrown if uncommon characters were found within the specified field. Double-check this field to ensure that the values within it are correct, readable, and free of new lines/carriage returns. Note that these flags are thrown under particular contexts, so not all flags are errors and not all instances of these characters will cause flags **with the exception of new lines "\n" and carriage returns "\r"**—these characters should never exist in a parcel submission.
  - **Fix**: To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: The NETPRPTA value is greater than the GRSPRPTA value. See Validation_and Submission_Tool_Guide.pdf for further information.**
  - **Nature:** This flag will be thrown when NETPRPTA value is **greater** than the value provided in the GRSPRPTA field. This could be indicative of delinquent utility charges or non-property tax charges being included in the NETPRPTA field.
  - **Fix:** If possible, remove non-property tax charges from NETPRPTA value. If this is not feasible, then <Null> out value in the NETPRPTA according to statewide schema specifications.

- **Error/Flag: A value provided in PROPCLASS field is not in acceptable domain list.**
  - **Nature**: This error will be thrown if any values other than those listed as acceptable domains for PROPCLASS are found within the field. As defined in the Submission Documentation, the valid domains for this field are:
    - **1**
    - **2**
    - **3**
    - **4**
    - **5**
    - **5M**
    - **6**
    - **7**

- **Fix**: To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records. Note, for large numbers of errors, the Class of Property Dissolve Toolset may be of use in correcting this issue.

- **Error/Flag: A value provided in AUXCLASS field is not in AUXCLASS domain list. Please ensure mappings for these values are provided in the 'Explain-Certification' box of submission form.**
  - **Nature**: This error will be thrown if any values other than those listed as acceptable domains for AUXCLASS are found within the field. As defined in the Submission Documentation, the valid domains for this field are:
    - **X1**
    - **X2**
    - **X3**
    - **X4**
    - **W1**
    - **W2**
    - **W3**
    - **W4**
    - **W5**
    - **W6**
    - **W7**
    - **W8**
    - **W9**
  - **Note:** The above is not an exhaustive list of acceptable values. Other classifications you may have that are not included in the definition of AUXCLASS or PROPCLASS may be included within AUXCLASS, and **these additional values would not require standardization** as long as the definitions are specified within the *Explain-Certification.txt* file.
  - **Fix**: To troubleshoot this flag, first review the domains listed within the flagged field. If any of the values represent an AUXCLASS EXEMPT or AUXCLASS SPECIAL value, ensure that they are corrected to meet the values annotate in the Submission Documentation. If the value does not represent an AUXCLASS EXEMPT or AUXCLASS SPECIAL value, then the value can be left as is.
    - Make sure to define each of these additional values within the *Explain-Certification.txt* file.
  - To make corrections to a value, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records. Note, for large numbers of errors, the Class of Property Dissolve Toolset may be of use in correcting an issue.

- **Error/Flag: Duplicate values exist in [PROPCLASS or AUXCLASS] field.**
  - **Nature**: This flag will be thrown if a domain is observed to exist twice within either the PROPCLASS or AUXCLASS fields.
  - **Fix**: A domain should only be listed once within this comma delimited list. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: Both PROPCLASS and AUXCLASS fields are <Null> and a value is expected.**
  - **Nature:** This flag will be thrown if a <Null> value is found in both the PROPCLASS and AUXCLASS field and the record appears to be a valid taxable parcel.
  - **Fix:** Check all records containing this flag and ensure that at least one of these fields is populated for all valid taxable parcels. If values are not available, provide an explanation in the *Explain-Certification.txt* file.

- **Error/Flag: The value provided in CONAME field does not match expected county name.**
  - **Nature**: This flag will be thrown if the CONAME does not match the name of the county expected, which would be the contributing jurisdiction. The CONAME is expected to be the same as the contributing county name.
  - **Note:** If you are contributing parcels on behalf of another county, this flag can be disregarded.
  - **Fix**: Ensure that the CONAME value is correctly populated. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: The value provided in PARCELFIPS field does not match submitting county fips.**
  - **Nature**: This flag will be thrown if the PARCELFIPS does not match the name annotated within PARCELSRC as would be expected.
  - **Tip: Make sure that leading zeros are intact within this field.**
    - For example, BAYFIELD has a FIPS code of 007, if the value of "7" were used in lieu of "007" this flag would be thrown.
  - **Fix**: Ensure that the value is correctly populated. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: The value provided in [PARCELSRC or PARCELFIPS] field does not appear to meet required domains.**
  - **Nature**: This flag will be thrown if the PARCELFIPS or PARCELSRC does not match the domain list as defined within the Submission Documentation.
  - **Tip: For PARCELFIPS, make sure that leading zeros are intact within this field.**
    - For example, BAYFIELD has a FIPS code of 007, if the value of "7" were used in lieu of "007" this flag would be thrown.
  - **Tip: For PARCELSRC, make sure that the word " COUNTY" is not included in this field.**
    - For example, an annotation of "BAYFIELD COUNTY" would cause a flag to be thrown.
  - **Fix**: Ensure that the value is correctly populated. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: The value provided in PARCELSRC field does not match submitting county name.**
  - **Nature**: This flag will be thrown if the PARCELSRC does not match the number annotated within PARCELFIPS as would be expected.
  - **Tip: Make sure that leading zeros are intact within the PARCELFIPS field**.
    - For example, BAYFIELD has a FIPS code of 007, if the value of "7" were used in lieu of "007" this flag would be thrown.
  - **Fix**: Ensure that the value is correctly populated. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records.

- **Error/Flag: The value provided in SCHOOLDISTNO is not within the acceptable domain list or is not 4 digits long as expected. Please verify value.**
  - **Nature**: This flag will be thrown if the SCHOOLDISTNO does not match the length of a valid school district ID (character length of four) or is not within the acceptable domain list.
  - **Fix**: Ensure that the value is correctly populated. See schema definition of SCHOOLDISTNO for hints. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records. See the Parcel_Domain_List for a comprehensive list of valid SCHOOLDISTNO and SCHOOLDIST values.

- **Error/Flag: One or both of the values in the SCHOOLDISTNO field or SCHOOLDIST field are not in the acceptable domain list. Please verify values.**
  - **Nature**: This flag will be thrown if the SCHOOLDISTNO or SCHOOLDIST values do not match valid school district name (SCHOOLDIST) or school district number (SCHOOLDISTNO) listed within the Parcel_Domain_List.
  - **Fix**: Ensure that the value is correctly populated. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records. See the Parcel_Domain_List.xlsx for a comprehensive list of valid SCHOOLDISTNO and SCHOOLDIST values.

- **Error/Flag: A value in SCHOOLDIST is not within the acceptable domain list. Please verify value.**
  - **Nature**: This flag will be thrown if the SCHOOLDIST field does not match a valid school district name within the updated domain list for school districts: Parcel_Domain_List.
  - **Fix**: Ensure that the value is correctly populated, meeting a valid domain as appropriate. To make corrections, use an ArcMap editing session or *Field Calculator* to overwrite and correct the records. See the Parcel_Domain_List for a comprehensive list of valid SCHOOLDISTNO and SCHOOLDIST values.

- **Error/Flag: Both the SCHOOLDISTNO & SCHOOLDIST are <Null> and a value is expected.**
  - **Nature:** This flag will be thrown if a <Null> value is found in both the SCHOOLDISTNO and SCHOOLDIST field and the record appears to be a valid taxable parcel.
  - **Fix:** Check all records containing this flag and ensure that at least one of these fields is populated for all valid taxable parcels. If values aren't available, provide an explanation in the *Explain-Certification.txt* file.

8 Repeat Sequence (As Needed)

The validation sequence articulated through the steps up to this point will need to be executed once and then the **tool executed again in TEST mode** to ensure that the errors were resolved.

Once the tool is run in TEST mode without error, the tool can then be run in FINAL mode to finalize the submission.

FINAL MODE

9 Overview of FINAL Mode

The Validation and Submission Tool's secondary purpose is to prepare the mandatory submission form that will accompany the zipped GIS file submission.

- The *.ini submission form* is an .ini format file that is yielded from successfully running the tool in FINAL mode.
- The *.ini submission form* provides all of the information you have inputted in FINAL mode—including the content of the *Explain-Certification.txt* file.

This part of the guide will explain what is needed to create the *.ini submission form* and how to package the submission.

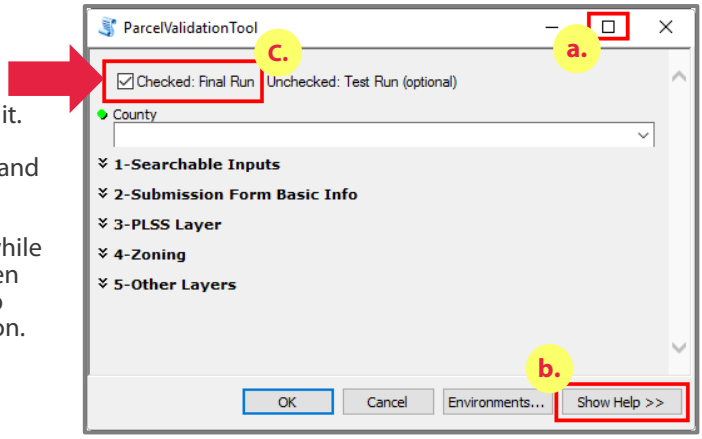
9.1 Getting Started/Installing the Tool

- Be sure to read sections 1-11 of this guide before attempting to run the tool in FINAL Mode.
- Subsequent sections of this guide presume that you have read sections 1-11 and have installed the tool.

9.2 Using the Tool and Configuring FINAL Mode

- To begin with FINAL mode, open the **ParcelValidationTool** from within the *Submission_Validation_Toolbox.tbx*.
- The interface of the tool will familiar if you have been using the tool in TEST mode.

- Checking the box **c.** will set the tool to execute in **FINAL mode**, and will affect the parameters that are available for populating within the five accordion toggles within the tool dialogue below it.
  - **Tip:** Use the "Maximize" toggle (item **a.** in the above figure) to gain more screen real estate and make configuring the tool easier.
  - **Tip:** Make sure to also open the "**Show Help**" section (item **b.**) for reference while configuring the tool. When you click on a given dialogue component, the corresponding help section will be brought-up in the "help" section.



9.3 Completing the Tool's Dialogue with Parameters

- Once you have properly configured the tool to run in FINAL mode and have set the County as appropriate, the remainder of the tool's configurations are nested across the tool's five accordion dropdowns.
- **Remember to use the tool's help section ("Show Help") if any parameter is unclear.**
- Fill out the tool's dialogue, working from top to bottom, **expanding each of the five accordion sections one at a time**.
- See Figure 2 for an explanation of the tool's accordion sections.
- If any parameter is grayed out, you can skip it and move along.
- If you miss a parameter, the tool will identify the missing parameter and will not allow you to execute the tool.
- Regardless of your submission type, a requirement to successfully completing the tool in FINAL mode is the inclusion of a completed *Explain-Certification.txt* file.
- **Complete and Save** the *Explain-Certification.txt* file. It can be found as a template within the root level of the tool's zipped package.
  - **Tip:** Use the ArcCatalog "Results" window to pull-up past configurations of the tool in order and revise the tool's configurations. To do this, open the results window from the File menu: *Geoprocessing > Results*. Within the results window, navigate to the most recent running of the tool and double click the script element. This will bring-up the former configuration(s) of the tool, as depicted in Figure 6.

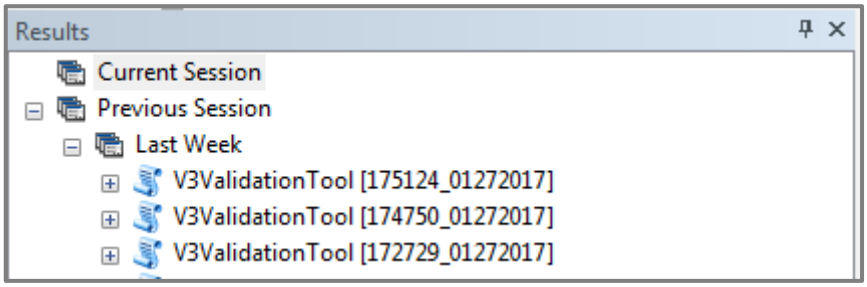


Figure 6. Past configurations of the tool, shown in ArcCatalog

10 Inputting the *Explain-Certification.txt* file

Regardless of your submission type, a requirement to successfully completing the tool in FINAL mode is the inclusion of a filled-out, complete *Explain-Certification.txt* file.

- The *Explain-Certification.txt* file template can be found within the root level of the tool's zipped package, available on the tools page.

- You will be asked to configure the tool with your completed *Explain-Certification.txt* file within the "2-Submission Form Basic Info" section.
- Make sure to fill-out and save the *Explain-Certification.txt* form first.
- After saving the file, simply drag the *Explain-Certification.txt* file from ArcCatalog or from Windows Explorer onto the box outlined in red, or use the file picker to configure this parameter.
- **Important! Make sure to fill-out and save the *Explain-Certification.txt* form BEFORE running the tool.**

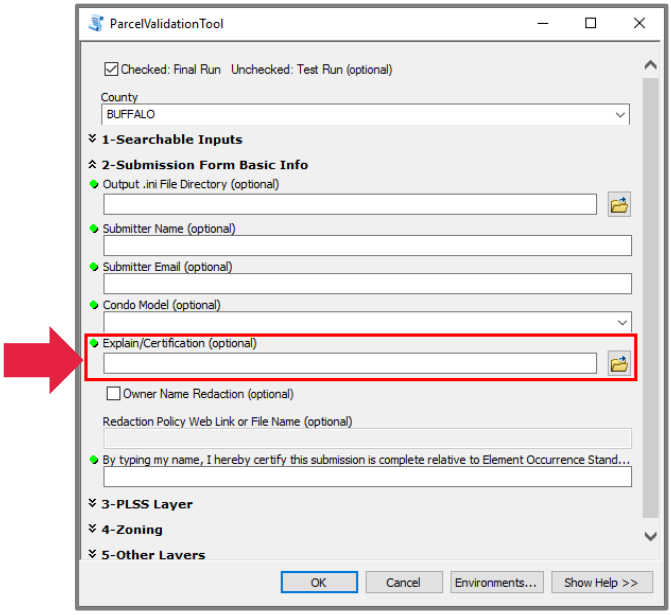


Figure 7. Where to input Explain-Certification.txt file (required)

About filling out the *Explain-Certification.txt* file:

- Open the *Explain-Certification.txt* file from the root level of the zipped package.
- Use the text editor of your choice, such as *Notepad++* or Windows default text editor *Notepad*.
- Fill out the information pertaining to the unsolvable errors, missing data, etc. by simply typing text into the line below.
- If the *Explain-Certification.txt* is left unaltered when used in configuring the tool in FINAL mode, the .ini will not be produced. If unsolvable errors, missing data, or other issues with adhering to the submission standards do not exist, write "NONE" within the *Explain-Certification.txt* file.
- The text you type out gets integrated in your final *.ini submission form*. It is **not** necessary to include the *Explain-Certification.txt* file itself in your zipped submission folder.
- Below is an example of a properly configured *Explain-Certification.txt* file (for sake of readability, the responses appear in green below):

```
**********************************************************************************************
Please provide a brief explanation of any unsolvable errors, missing
data, or other issues with adhering to the submission standards here
(if there are none, write NONE):
--There are 30 instances of the following error "unknown issue
occurred with the ADDNUM field. Please inspect this field's value."
These issues due to a unique address number situation in Badger
County that deviates from expected values due to a unique, non-
standard character in our address numbering system.
--There are 204 parcels that do not have a SCHOOLDIST or a
SCHOOLDISTNO. These are new parcels that have not yet been developed.
```

Figure 8. Explain-Certification.txt file (example)

11 Saving the Mandatory .ini Submission Form

Once the submission form is complete, the *.ini submission form* will be written to the output directory that was chosen under accordion 2 under "Output .ini File Directory."

- Navigate to the location of the .ini file and **copy the .ini file to your submission folder** (the directory where you have stored all of the files you will submit).
- Include county name in the name of your .ini file
  - e.g., **JUNEAU_Final.ini**
- You do **not** need to include the *Explain-Certification.txt* file in your submission folder (because entering it in the tool's section 2 transferred that text into your final *.ini submission form*).

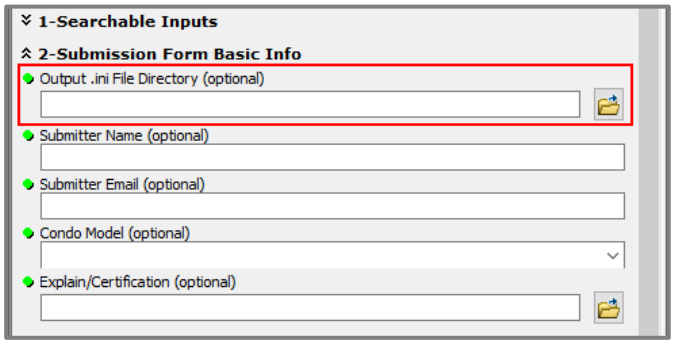


Figure 9. Where to find your final .ini Submission Form to include with zipped submission

11.1 Failure to Produce .ini submission form

If issues are found within the parcel data or the configuration of the Validation Tool in FINAL mode, the *.ini submission form* will not be created. If the *.ini submission form* fails to be created, the Validation Tool will provide the user with a message similar to the one pictured here. The following are some example scenarios under which the *.ini submission form* will not be created:

- If CONAME, PARCELFIPS, or PARCELSOURCE contain any number of missing values.
- If the parcel feature class does not meet the Parcel Initiative's searchable format CRS.
- If the parcel feature class does not contain the same attribute schema as that of the statewide schema.
- If the *Explain-Certification.txt* used when configuring the Validation Tool in FINAL mode is left unaltered. If no issues exist, type "NONE" within *Explain-Certification.txt* before inputting it.

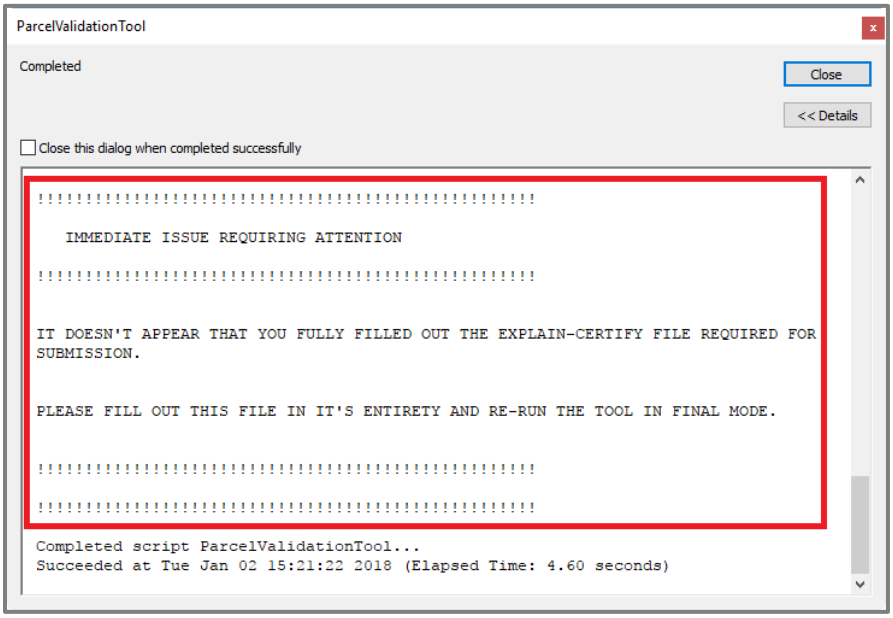


*Figure 10. Error message indicating .ini Submission Form has **not** been produced (example)*

SUBMIT .INI SUBMISSION FORM + DATA

Packaging the Submission

Before uploading to submit, include the following on the root level of your zipped package:

- .INI SUBMISSION FORM
- PARCEL FEATURE CLASS WITH TAX ROLL DATA
- OTHER LAYERS – PLSS / OTHER LAYERS – RML

Layer/Theme	Naming Convention (Required)
☐ **.INI SUBMISSION FORM**	**COUNTYNAME_Final.ini**
☐ **PARCEL FEATURE CLASS WITH TAX ROLL DATA**	COUNTYNAME_**PARCELS**.gdb\PARCELS
☐ **OTHER LAYERS:**	
PLSS	COUNTYNAME_**OTHER**.gdb\COUNTYNAME_**PLSS**_YEAR
Zoning – General (county-maintained)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**GENERAL**_YEAR
Zoning – Shoreland (county-maintained)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**SHORELAND**_YEAR
Zoning – Airport Protection (county-maintained)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**AIRPORT**_YEAR
Rights of Way	COUNTYNAME_OTHER.gdb\COUNTYNAME_**ROW**_YEAR
Roads/Streets/Centerlines	COUNTYNAME_OTHER.gdb\COUNTYNAME_**ROADS**_YEAR
Hydrography (line and/or polygon)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**HYDRO**_YEAR_POLY (or "_LINE")
Addresses	COUNTYNAME_OTHER.gdb\COUNTYNAME_**ADDRESSES**_YEAR
Buildings/Building Footprints	COUNTYNAME_OTHER.gdb\COUNTYNAME_**BUILDINGS**_YEAR
Land Use	COUNTYNAME_OTHER.gdb\COUNTYNAME_**LANDUSE**_YEAR
Parks/OpenSpace (e.g., county forests)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**PARKS**_YEAR
Trails	COUNTYNAME_OTHER.gdb\COUNTYNAME_**TRAILS**_YEAR
Other Recreation (boat launches, etc.)	COUNTYNAME_OTHER.gdb\COUNTYNAME_**RECREATION**_YEAR

Submit .ini Submission Form + Data

@ wisedecade.legis.wisconsin.gov

1. **Note the WISE-Decade browser requirements**
   - Compatible with **IE 10**+, **Firefox 28**+, **Chrome 33**+
   - If upload via WISE-Decade fails, there is an alternative upload page

2. **Look for a confirmation message after upload**
   - The Progress indicator will display "100%" after a successful upload, as well as a confirmation message.
   - You are done when you see the following confirmation message (appearance will vary by browser):