# Technical manual for tools provided

Tools for LPIS preparation

Background & Technical manual

Contents

[1. Rationale 3](#_Toc412143747)

[2. Technical description and outline 4](#_Toc412143748)

[2.1 Removing large administrative parcels 4](#_Toc412143749)

[2.2 Splitting LPIS 4](#_Toc412143750)

[2.3 Duplications 4](#_Toc412143751)

[2.3 Parcel ID and tracking changing parcels 5](#_Toc412143752)

# Rationale

The Land-Parcel Identification System (LPIS) consist of polygon feature classes, where each feature represents a land-parcel. Each feature class is representative of a single year. A number of fields provide information for each of the land-parcels, including size and land-use.

Originally the LPIS was created an administrative tool to assist authorities and farmers with the single-farm payment scheme as part of the European Union’s (EU) common agricultural policy. As such the LPIS has been constructed for a very specific purpose and is subject to certain limitations when utilising outside mentioned purpose.

1. Large administrative parcels – Some areas within the LPIS (mainly island groups within the Atlantic Ocean) are covered by single large polygons, which according to Mallon Technology represent administrative boundaries.
2. Multiple Polygons Per Parcel – On review of the LPIS dataset it has been identified that there can be multiple polygons and records in the dataset for the same parcel. This is particularly prevalent in relation to commonage areas where multiple farmers can be claiming for the same shared area. The result of this duplication is that the dataset has an area approximately 25% larger than the actual area of Ireland. It also means that there are multiple records in relation to the land-use type and livestock figures.
3. Topological & Geometric Issues – In addition to the multiple polygons above there are issues around overlapping polygons, this is where the boundary of one parcel overlaps with another meaning that technically the same bit of land is being accounted for. This results in difficulties in accurately assigning the land-use type or determining the livestock figures for this area.
4. Geometry Changes on Parcels – The changing of the geometry of parcels from year to year leads to the difficulties in tracking change over time. For example it is common that the boundary of a field parcel would change due to splitting fields, selling land etc. There appears to be no relationship between the Parcel ID’s of new and old parcel boundaries. For example if a field is split 2 new Parcel ID’s are created and there is no link back to the original ID meaning users cannot identify and account for the change.
5. Multiple Crop / Land-use Descriptions – There is no standardisation of the crop descriptions that are used within LPIS. It appears that when the data is collected it is entered into free text forms instead of selecting from standardised data types / lists. The data is collected in a single field but can have multiple crop types as entries within the same field (e.g. Grass, Rough Grazing, and Permanent Pasture). This has resulted in more than 1,200 unique entries and is often due to a variation in how the same information is collected, for example “Grass, Rough Grazing” could also be in as “Rough Grazing, Grass”. The impact of this is that the data needs to be cleaned and standardised before it can be used for reporting, assessments and data creation (e.g. Corine 2012).

In order to utilise LPIS for scientific purposes a number of operations are required to adapt the system. The nature of these steps may depend on the proposed use for the LPIS and the decision which steps to use has to be made by the respective researchers/users.

# Technical description and outline

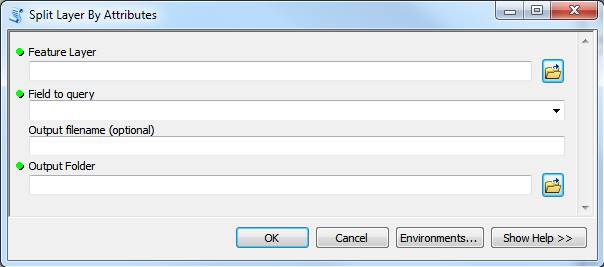
Depending on the problem to be solved operations are either carried out using native ArcGIS 10.1+ tools or require purpose built tools. Each step and the use of the tools required are explained in the following section.

## 2.1 Removing large administrative parcels

This step is carried out manually. Either by optically scanning each LPIS dataset and selecting obviously large parcels that span multiple islands or large fresh-water bodies (i.e. lakes) and removing them using the editor function within ArcGIS, or by selecting all parcels larger than a previously selected area and deleting them using the ‘Delete features’ tool within the ‘Data management tools/Features’ toolbox.

## 2.2 Splitting LPIS

LPIS datasets tend to be very large (around 3 GB per year) which will cause problems with many standard issue computers. It is therefore recommended to split the LPIS dataset into smaller subsets. Depending on data requirements, the subselection criteria are at users’ discretion. Previous work has shown that dividing the LPIS into counties creates files small enough for further use.

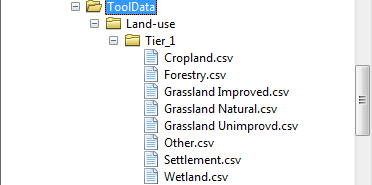
To split the LPIS shapefiles into we recommend using a separate polygon shapefile representing the desired regions. Using the “Spatial join” tool (ArcGIS tool Box/Analysis tools/Overlay), joining the shapefile with the regions to the LPIS shapefile. Once the join has been applied to the LPIS file the ‘SplitLayerByAttribute’ (provided with the tool set, originally downloaded from <http://arcscripts.esri.com/details.asp?dbid=14127>, author: Dan Patterson) can be used to split the original LPIS layer according to the newly added field. While the provided parcel ID’s first letter represents the county, it is not a reliable indicator of the location of the parcel since the letter indicates the residence of the land-owner.

## 2.3 Duplications

Duplications are generally a result of commonage. Identifying or removing such duplicates can be carried out using the ‘Find Identical’ and ‘Delete Identical’ within ArcGIS 10.1+ (Data management tools/General). There are two possible methods, either by ‘PARCEL\_ID’ as commonage parcels generally have identical IDs, or by ‘Shape’. The second option provides the security that only parcels of the exact same shape will be deleted, however it has occurred that due to the size of LPIS the tool does not always work probably.

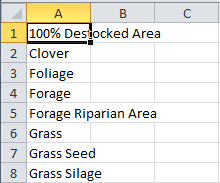
An additional tool ‘Search Duplicates’ was created which allows selecting between simple identification and deletion based either on location or a selected field. It also allows to select a field which will limit deletion if the value in the field is not identical in otherwise duplicated parcels (e.g. if a different land-use is reported for otherwise duplicated fields.

## 2.4 Summarise land-use categories

A tool to summarise the crop descriptions provided in LPIS into overarching categories. In order for the tool to work, the crop descriptions have to be grouped into different .csv table according to the user’s requirements. The table name will be used as the category name. All tables need to be in a single folder. Within the table, each crop description should be in a single row.

*Figure 2: Structure of .csv table containing land-use classes to be summarised (Example: 'grassland improved.csv')*

Figure 1: Folder structure for .csv tables of cropland categories.

In the tool window select the shapefile you want the tool to be applied to. Select the crop description field you want to summarise from the drop down menu. After that you need to select the folder containing the .csv tables and provide the name for the field containing the summarised category names.

## 2.5 Parcel ID and tracking changing parcels

While the provided ‘PARCEL\_ID’ field generally provides a single ID that will stay identical through time there are certain limitation when using this identifier. If fields are changed, even slightly, a now ID will be assigned to the field with no notification of the previous ID. This can make tracking a single parcel through time very difficult as it requires constant use of spatial identifiers such as centroids.

In order to limit the use of spatial identifiers a tool was developed to add a new unique ID which can be fully traced through the whole timeframe. Furthermore is it possible to add specific criteria at which to consider parcel identical (i.e. a maximum area difference and a maximum centroid shift).

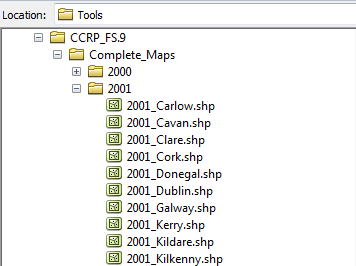
The tool will assign a unique ID throughout the selected timeframe to each parcel considered identical with regards to the criteria. If a parcel is considered to change outline, a new ID will be assigned, however a new fields containing the old ID in the newer year and a field containing the new ID in the previous year will be added. The tool provides a field containing information on the type of change a parcel undergoes, if it changes ID. The following classes are defined:

1. No previous information – This applies to the first year in a timeframe as no information of previous parcel outline is available.
2. Removed next year – The specific land-parcel will drop out of LPIS in the subsequent year.
3. Added parcel – The parcel did not exists in the previous LPIS dataset
4. Changed outline – The parcel has changed its outline beyond the previously defined criteria and was assigned a new ID
5. Changes outline – The parcel will change outline in the subsequent year and will be assigned a new ID

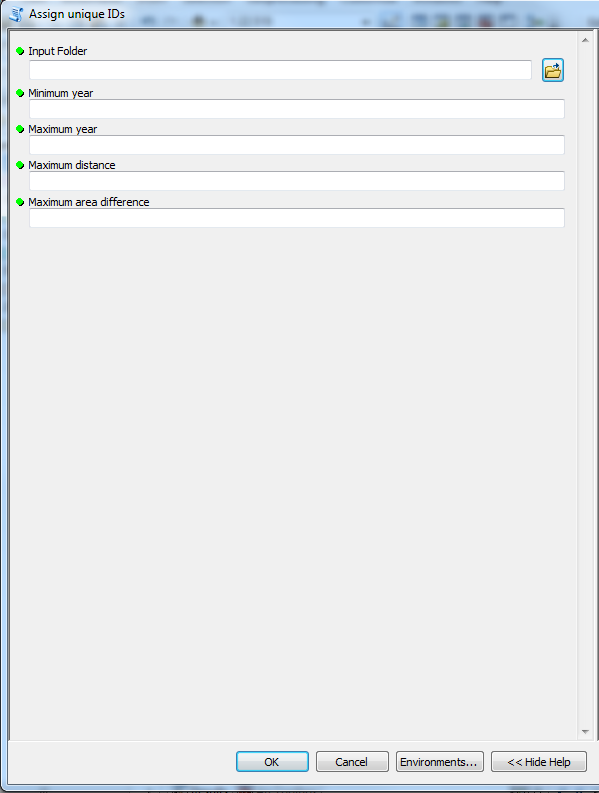
The classes can also be combined, since a changed parcel can for example drop out of LPIS in the subsequent year.

Additionally the toolbox contains a second tool that will add an ID history to each dataset in the timeframe, containing all IDs at a given location.

Using the tool:

The tool requires a specific data structure to work. If the LPIS has previously been split, which is recommended, it is important that the parts of a single year in LPIS are kept in a single folder named after the year. Furthermore, the folders representing each year need to be in a single parent folder. The single files should follow a given naming convention, which is ‘[Year]\_Name.shp’. The name is required to be constant throughout different years. Furthermore, to prevent repeated IDs in different LPIS subsets it is crucial to keep all subsets in the given folders. Assigning IDs to separate subsets will lead to IDs starting at 1 for each subset. New years can be added to an existing timeline. Select the last year of the previous assignment as the minimum year in the current tool run. The tool will recognise that assignment has already been carried out and will, in case of new parcels appearing continue to assign IDs from the highest previous ID on.

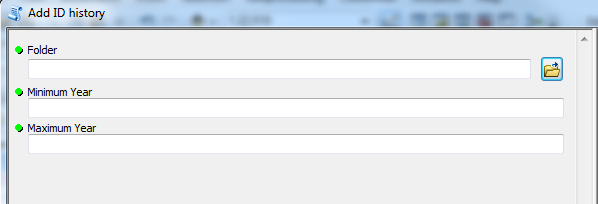
*Figure 3: Example of naming convention and folder required folder structure.*



Tool dialog (Assign unique ID)

The tool requires the parent folder, containing the subfolders for each year, the minimum and maximum years of the required timeframe as well as the criteria for considering parcels unchanged over time (i) maximum centroid distance in meters and minimum area difference in hectares.

*Figure 4: Assign unique ID tool dialog*

Tool dialog (Add ID history)

*Figure 5: Assign ID history tool dialog.*

The Add ID history tool requires only the folder containing the subfolders of the whole timeframe as well as the minimum and maximum years of the observed timeframe.

## 2.6 Determine land-use change

This tool reads the values of a given land-use field of two subsequent years and adds a new field to the later year containing the land-use change in the form ‘x to y’ or ‘x remaining x’. The tool can either be applied to a folder containing subfolders of the same naming conventions given in section 2.3 (Folder based selection) or it can be applied on two specific files selected by the user (file based selection). The file based selection simply requires adding two LPIS files. The land-use field can then be selected in a drop down menu.

For the folder based selection the tool requires a parent folder and the starting and finishing year of the timeframe. The field has to be typed in manually.

In order for both tools to work the LPIS files need to have the unique IDs added by the tool described in section 2.5.

LPIS extraction and field value listing tools for ArcGIS

Background & Technical manual

Contents

[1. Rationale 3](#_Toc398897830)

[2. Technical description and outline 4](#_Toc398897831)

[2.1 Extracting tool 4](#_Toc398897832)

[2.2 Listing tool 4](#_Toc398897833)

[3. Manual 7](#_Toc398897834)

[3.1 Folder based input selection 7](#_Toc398897835)

[3.1.1 Extraction tool 7](#_Toc398897836)

[3.1.2 Listing tool: 8](#_Toc398897837)

[3.2 File based input selection 10](#_Toc398897838)

[3.2.1 Extraction tool 10](#_Toc398897839)

[3.2.2 Listing tool 11](#_Toc398897840)

# Rationale

The Land-Parcel Identification System (LPIS) consist of polygon feature classes, where each feature represents a land-parcel. Each feature class is representative of a single year. A number of fields provide information for each of the land-parcels, including size and land-use.

For certain analysis it may be necessary to extract only parcels that represent certain criteria during some stage within a given timeframe (e.g. parcels that were reported as cropland at some stage during a defined timeframe). Furthermore, as each LPIS feature class represents a single year, a methodology to list all values in a given field over a given time-frame in a single feature class will be required when summarising and assessing changes over time.

The aim of the tools is to provide means of:

(1) Extracting land-parcels from the LPIS that at some stage throughout a given timeframe display a selected field value (extraction tool), and

(2) Of listing all field values of a selected field that occur in a given timeframe in a new feature class (listing tool).

Both tools have been designed to allow the user to input the timeframe, the field that is to be tested and a list of values, either as manual input or imported from a .csv table that the selected field is to be tested for.

The outputs of the tools are:

(1) A subset of the original input files consisting only of features that displayed the selected field value at some stage during the timeframe, and

(2) A feature class with a separate field for each step during the timeframe displaying the specific field value displaying the field value the specific land-parcel showed at a given time-step.

Due to the nature of the LPIS, the outline of the land-parcels may change over time. The tools are designed to include those changing parcels in both, the extraction tool, and the listing tool. Within the listing tool this leads to multiple differently shaped features, representing different steps in the given timeframe. This means that a single feature may not represent the full selected-field history at a given location. In order to provide a full history for each location that is covered by a feature in the LPIS, the listing tool will produce a second output, which is a dissolved version of the original output. In case of multiple, differently shaped features, at a single location, the tool will produce new features representing the common outlines of these parcels, with the full selected field history for the given location.

# Technical description and outline

## 2.1 Extracting tool

The tool is based on unique parcel IDs given to each feature within the LPIS feature classes, where each ID represents a parcel of a unique geometry, however independent of the year. Instead of actually extracting features from the input feature classes, the tool creates a copy of each input file and then removes features not meeting the given criteria.

As the tool objective is to extract parcels that, meet the criteria during some stage within the given timeframe, it comprises a three-step approach to carry out the extraction, as simply removing features that do not meet the criteria in a single year would not allow to trace the specific features history.

1. In the first step the tool copies all input files representing the different time steps of a given LPIS feature class.
2. The tool then iterates through the copies using a search cursor (arcpy.da package) and memorises the unique IDs of all parcels that meet the given criteria in a single list. To include parcels that change outline (and therefore the unique ID) the tool requires the ID history of each parcel included. From each feature meeting the criteria the ID history will be extracted and, if not already present, added to the list of IDs.
3. In a last step the tool will iterate through all copies using an update cursor (arcpy.da package) and remove all features which unique IDs are not present within the list.

## 2.2 Listing tool

Again the tool is based on the unique parcel ID to list all values of a selected field in a single feature. The basic operation of the tool is to create a new feature class containing the unique IDs and the values of a selected field for an initial year and then to add a new field for each following year, populating it with the selected field values where the unique IDs match. In case a unique ID is not present, a new feature will be added to the feature class.

As previously mentioned, this process may lead to multiple overlapping features of different geometry if features change outline and therefore ID over time. In a following step the feature class is overlaid with itself in a union operation, to create features of common geometry, which then are dissolved to single features containing the complete field history.

As the original outlines of the input feature classes may be needed the tool produces two outputs. The first contains the features as present in the input feature classes, including possible overlapping features if geometries change over time.

The workflow to produce this outcome comprises of the following steps:

1. For reasons of simplification, instead of creating a new feature class, the initial time steps feature class is copied into a given output folder, all unnecessary fields are removed, and the selected field for listing the history is renamed to represent the given time step.
2. For all following years, the output feature class will be extended by a field representing the current year. Then the tool will iterate through the current input feature class using a search cursor (arcpy.da package) and write the necessary fields (unique ID, feature geometry, selected field) into list objects. This is done to improve tool performance, as list items can be directly accessed by their index, while using the cursor requires repeated iteration to locate single items. An update cursor (arcpy.da package) is then used to iterate through the output feature class, checking if the ID in each row is present within the list derived from the current input file. If that is the case, the information derived from the input feature class is written into the row and the list entry is flagged as being accounted for. When the cursor is finished, the all list entry not flagged as accounted will be added to the list feature class as new features, using an insert cursor (arcpy.da package).

The second output contains the dissolved features. The output feature class does not contain any overlapping features. Therefore any location within the feature class is represented by a single feature. However, the features may not be identical to the input files, as in case of overlapping features, new intersection features are created.

The second output is created using the following workflow:

1. The output of the first workflow within the tool is overlaid with itself using an ArcGIS union operation. In locations where features overlap the operation will create two new features representing the area of overlap, containing information of the respective parcels that overlap. Areas that do not overlap will not be affected by the operation.
2. To identify features with identical outlines, the coordinates of the centroid are added to the union feature class. While it is possible to use the geometry directly to identify identical parcels, using these coordinates as string is significantly faster when carrying out the following operations.
3. Using a search cursor (arcpy.da package), the centroid coordinates, as well as the listed field values from the union feature class are written into a list object. This list will then be used to dissolve the identical feature created by the union operation into single features containing information from all overlapping parcels. Achieving that takes three steps:
   1. Grouping all items within the list object by their location, i.e. the coordinates of the centroid.
   2. Collapsing the group back to a single item, containing all field values listed for the given timeframe.
   3. Using an update cursor (arcpy.da package) to write the collapsed lists into the union feature class.

The following example is included to visualise the process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Location** | **Time 1** | **Time 2** | **Time 3** | **Time 4** | **Time 5** |
| **A** | *Entry a1* | *Entry a2* | *<empty>* | *<empty>* | *<empty>* |
| **A** | *<empty>* | *<empty>* | *Entry a3* | *Entry a4* | *Entry a5* |
| **B** | *Entry b1* | *Entry b2* | *Entry b3* | *<empty>* | *<empty>* |
| **B** | *<empty>* | *<empty>* | *<empty>* | *Entry b4* | *Entry b4* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Location** | **Time 1** | **Time 2** | **Time 3** | **Time 4** | **Time 5** |
| **A** | *entry a1, <empty>* | *entry a2, <empty>* | *<empty>, entry a3* | *<empty>, entry a4* | *<empty>, entry a5* |
| **B** | *<empty>, entry b1* | *<empty>, entry b2* | *entry b3, <empty>* | *entry b4, <empty>* | *entry b5, <empty>* |

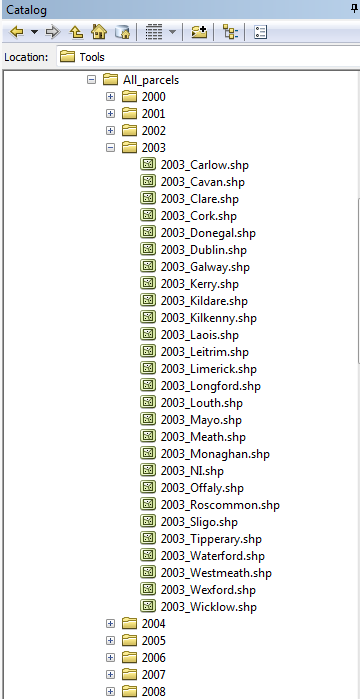
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Location** | **Time 1** | **Time 2** | **Time 3** | **Time 4** | **Time 5** |
| **A** | *entry a1* | *entry a2* | *entry a3* | *entry a4* | *entry a5* |
| **B** | *entry b1* | *entry b2* | *entry b3* | *entry b4* | *entry b5* |

1. In a final step the dissolved output is cleared of all duplicated features using the ‘Delete Identical’ tool in the ‘arcpy package’ and to remove any small areas ( < 0.1 ha) that might have been created during the union operation.

Furthermore, if the tool was used to list the field history of a subset of parcels containing a selected value during the observed timeframe (i.e. an output of the extraction tool) the dissolve operation may produce outputs that do not show this specific field value. This happened when of two overlapping features of different geometry only one contains the specific field value, as then the parts of the other feature that do not overlap will not show the specific land-use.

# Manual

## 3.1Folder based input selection

Due to the size of the LPIS it is very likely that the operation will be conducted on a number of subsets of the overall datasets, as using the whole dataset may be too large to process. The tool therefore uses folders as input rather than a list of files. The folder should contain subfolders for each time step, which in return contain the LPIS feature classes. For the tool to work, the feature classes representing different time steps of the same locations should be named identically, however the tool is able handle references to the time step in the name (e.g. 2000\_Carlow.shp, 2001\_Carlow.shp, …). Figure 1 shows an example of the folder structure required.

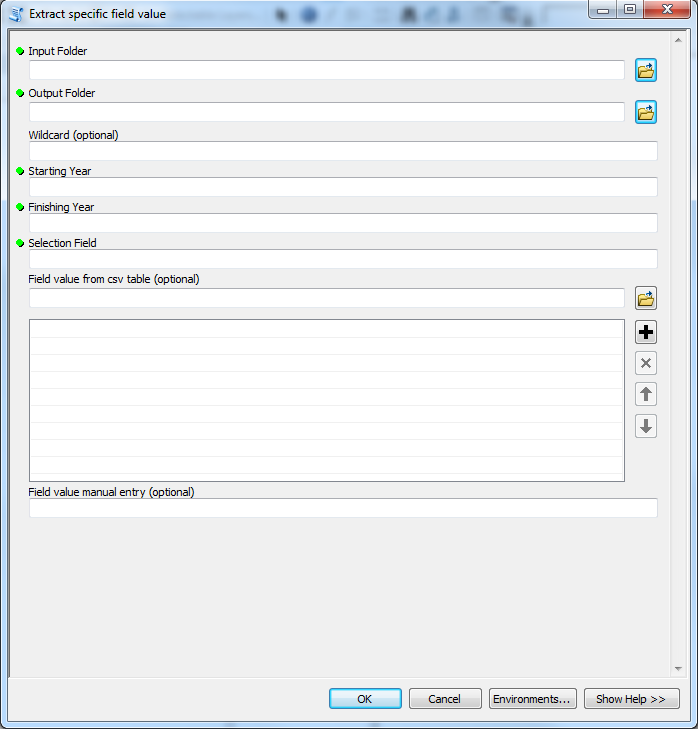
*Figure 1: Folder structure for using the folder based input selection*

### 3.1.1Extraction tool

*Data requirements*

To apply the tool the data to meet a number of criteria and must be structured in a specific way. The criteria are:

1. The presence of a unique parcel ID that does not change over time, the ID should be present in a float field named ‘UniqueID’.
2. For the tool to include parcels of changing outline, the ID history at a given location needs to be present in a string field named ‘ID\_Hist’. The name of the fields can be changed within the script if necessary.



Input parameters

1. Input Folder
2. Output Folder
3. Wildcard
4. Starting Year
5. Finishing Year
6. Selection Field
7. Field value (.cvs table)
8. Field value (manual)

*Figure 2: Tool window and parameter inputs (a to h) for extraction tool*

Figure 1 shows the tool window, as well as a list of the required inputs for the extraction tool. The data input requirements are as following:

1. The input folder requires input of the path in which the subfolders containing the specific shapefiles for each timestep are stored.
2. The output path will determine where the new output is saved. Within the output path new folders representing each timestep will be created and the respective output will be stored in the new folders.
3. The wildcard allows for limiting the input to files only containing a given string. The input follows the standard rules of wildcards, using asterisks to indicate string fragments (e.g. \*Carlow\* selects all files that at some position of the name contain the string ‘Carlow’). The wildcard is optional.
4. The minimum and (e) maximum steps of the timeframe. The input should be a whole number. The number needs to reflect the name of the folder in which the shapefiles for each shapefile are stored.
5. The field that will be used for the extraction query.
6. Import the selection value from a comma delimited table (.csv) (within the table each field value should be represented by a new row in a single column), or (h) as manual input. Again multiple values can be input, using a comma as separator.

### 

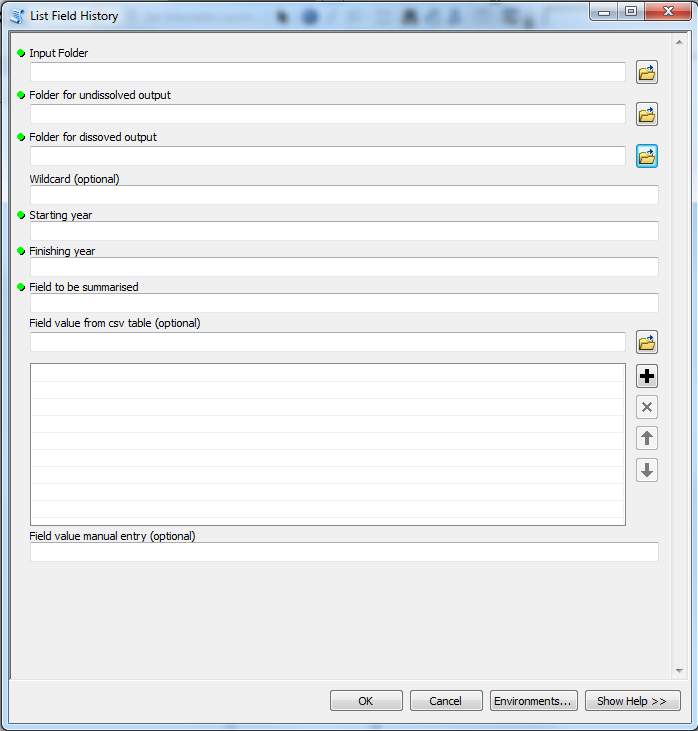
### 3.1.2 Listing tool:

*Data requirements*

The data requirements are similar to the extraction tool, with the exception that it does not require the ID history.

Input parameters

1. Input folder
2. Output folder for undissolved output
3. Output folder for dissolved output
4. Wildcard
5. Starting year
6. Finishing year
7. Field to list
8. Selection values (import from .csv table)
9. Selection values (Manual imput)



*Figure 3: Tool window and input parameters (a to i) for listing tool*

Figure 2 shows the tool window, as well as a list of the required inputs, for the listing tool. The data input requirements are as following:

1. Input folder, equivalent to the extraction tool, the folder should contain subfolders containing the shapefiles of the different steps in the timeframe
2. The folder for the undissolved output. Within the folder a new output shapefile for each timeline of inputs will be created
3. Equivalent to (b), but will contain the dissolved output.
4. The wildcard allows for limiting the input to files only containing a given string. Rules are equivalent to the extraction tool
5. The minimum and (e) maximum steps of the timeframe. The input should be a whole number. Equivalent to the extraction tool, the number needs to reflect the name of the folder in which the shapefiles for each shapefile are stored.
6. Setting the field which will be listed in the output
7. and (i) Setting an optional selection value, either from .csv table or manually. The value is used to filter the dissolved output to only contain features that contain the selected value at some point during the timeframe.

## File based input selection

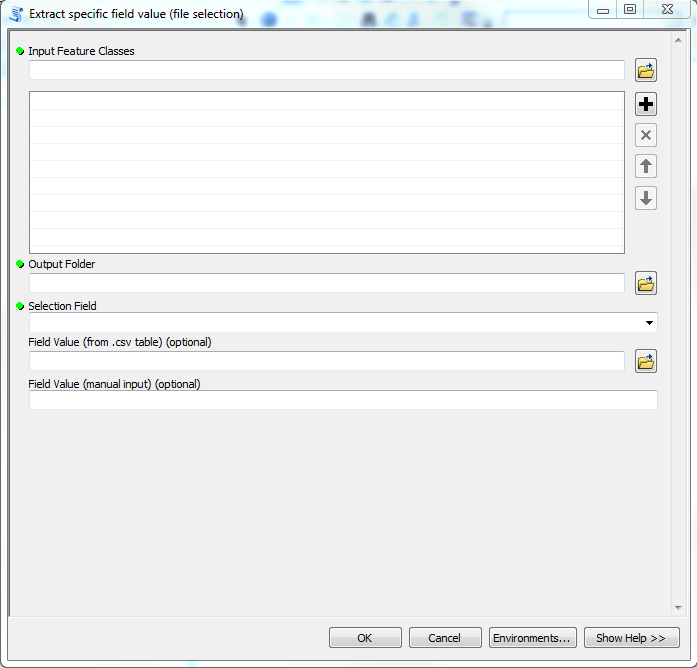
The file based approach is similar to the folder based in most regards with exception that instead of sorting the data into predefined folders, single files within a timeline can be manually selected. This allows applying the tool to single files without having to create appropriate folders and moving the files in question.

The data requirements are identical to the folder based selection.

### 3.2.1 Extraction tool

Input parameters

1. Input feature classes
2. Output folder
3. Field to list
4. Selection values (import from .csv table)
5. Selection values (Manual imput)



*Figure 4: Tool window and input parameters (a to e) for file based extraction tool*

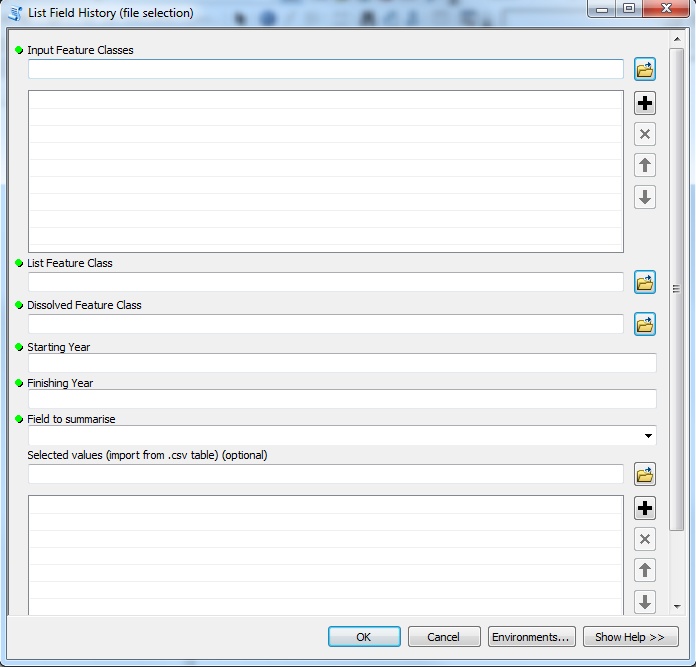
Figure 4 shows the tool window for the file based extraction tool. The parameters needed are as followed.

1. Input feature classes are the shapefiles from which those of a given land-use are to be extracted. The order of files defines the succession of the timesteps (top = oldest, bottom = newest).
2. As the tool creates multiple files the output destination is defined as a folder rather than single files. The name of the output files will be identical to the input files.
3. to (e): These parameters are identical to the folder based extraction tool.

### 3.2.2 Listing tool

Input parameters

1. Input feature classes
2. Undissolved output feature
3. Dissolved output feature
4. Starting year
5. Finishing year
6. Field to summarise
7. Selection values (import from .csv table)
8. Selection values (Manual imput)



*Figure 5: Tool window and input parameters (a to h) for file based list tool*

Figure 5 shows the tool window, as well as a list of the required inputs, for the file based listing tool. The data input requirements are as following:

1. Input feature classes are the shapefiles from which those of a given land-use are to be extracted. The order of files defines the succession of the timesteps (top = oldest, bottom = newest).
2. Output feature class for undissolved output
3. Output feature class for dissolved output
4. The minimum and (e) maximum steps of the timeframe. The input should be a whole number. The input is important for creating the output file. Therefore the length of the timeframe needs to be identical to the number of input feature classes.
5. to (h): Identical to folder based listing tool.