One-To-Many Attribute Mapping – Linking and Joining One Theme to Many Records Based on a Key Field

These instructions enable you to map out attribute data that has been joined to a theme. In one-to-many relationships, using ESRI ArcView 3.x software, you perform either of the following:

- link table and shapefile, OR
- split table, join to shapefile, and symbolize

You will need to use the project **OneToManyMap.apr** because it contains the scripts required for either method. The scripts have been tied to buttons in the appropriate document GUI. In this example, the problem is to combine tabular data with a shapefile so that wildlife observation records from many years can be incorporated into the spatial analysis and mapping of township polygons. Both the 'One' table (destination shapefile) and the 'Many' table (source attribute table) must have a common or key field value.

The first method involves scripts that automate the process of ***linking*** tables. Link establishes a one-to-many relationship between the destination table (the 'One' table; e.g. township shapefile's attribute table) and the source table (the 'Many' table; e.g. wildlife observation table). One record in the destination table is related to many records in the source table. After the link is established, selecting a record in the destination table will automatically select the record or records related to it in the source table – not so useful for visual mapping, but useful for tabular analysis.

The second method involves scripts that automate the splitting of the 'Many' table so that a one-to-one or many-to-one relationship to the destination 'One' table can be established. This is the process of ***joining*** tables. Typically, the source table contains descriptive attributes of features that you wish to join into a theme's table so that you can symbolize, label, query and analyze the features in the theme using the data from your source table.

ORIGINAL DATA

ab_twp.shp	a shapefile of Alberta townships that has a key field **`Twp_id`** to relate to the records of the Many table
selected.dbf	a dBASE table of the Many records that has a key field **`Location`** to relate to the shapes in the shapefile

CREATED DATA

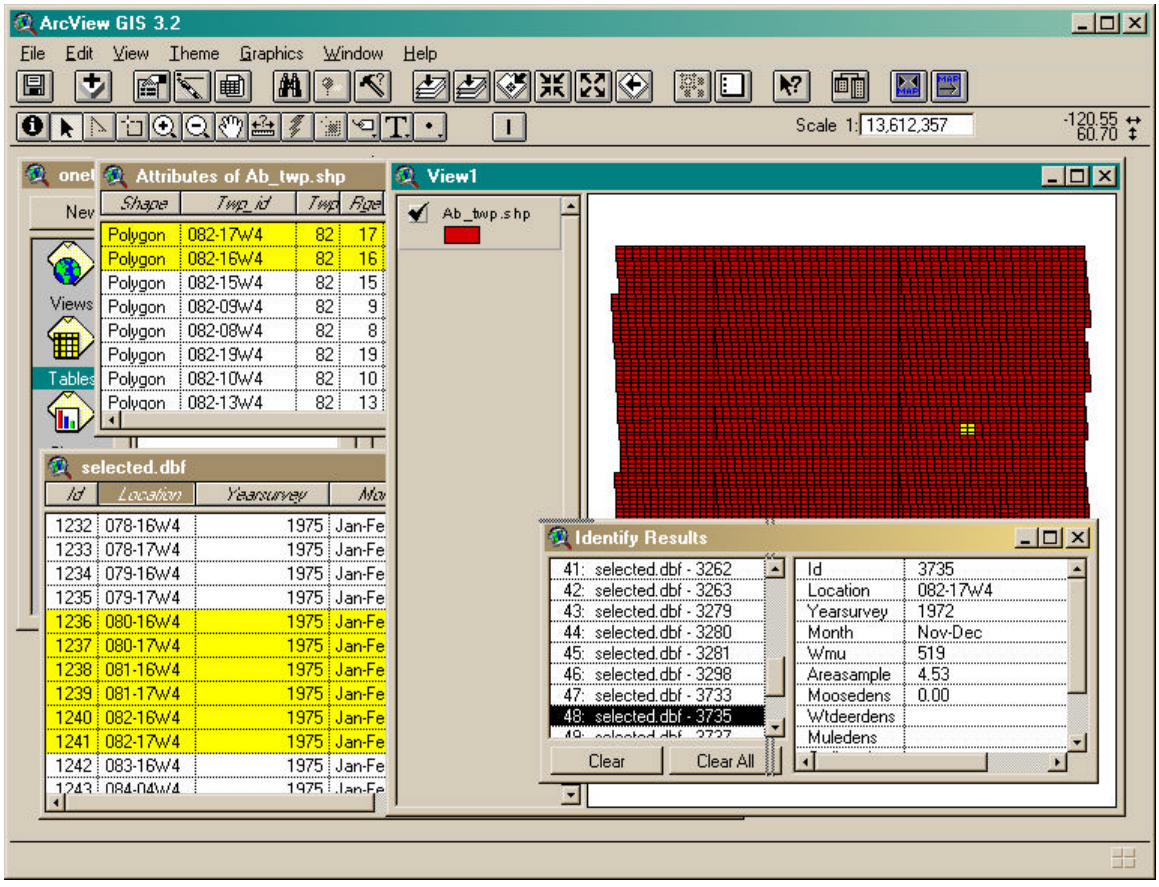
1969.dbf, …, 1997.dbf	various dBASE tables resulting from the split of the selected.dbf table based on the **`Yearsurvey`** field

Setup in ArcView 3.2:

1. Open the **OneToManyMap.apr** project in ArcView 3.x
2. In the VIEW GUI, add the **ab_twp.shp** theme
3. In the TABLE GUI, add the **selected.dbf** table
4. Optionally, set the Projection in the VIEW → PROPERTIES

First Method – Linking Tables

5. In the VIEW GUI, click on the SET ONE-TO-MANY RELATIONSHIP button 
6. Select **`Twp_id`** as the key field
7. Select **selected.dbf** as the 'Many' table
8. Select **`Location`** as the key field for the 'Many' table
9. Select the appropriate ID label field
10. Click YES to the question
11. Click on the **I** tool and click on the theme to select/identify attributes 
12. In the TABLE GUI, open up both tables

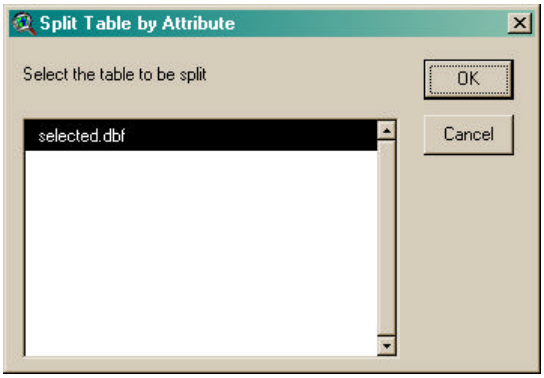


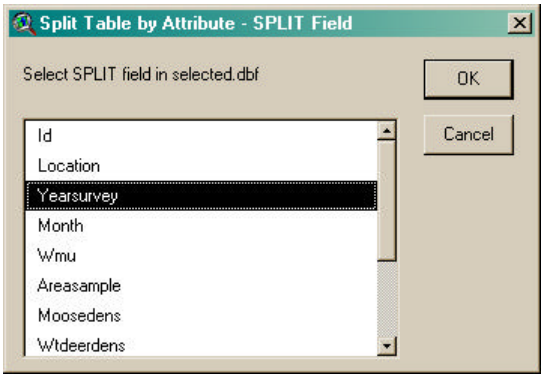
13. The select identified records in the "Attributes of Ab_twp.shp" table are highlighted, as are the corresponding records of the other table

14. You may also interactively select records in “Attributes of Ab_twp.shp” table to select the corresponding records of the other table
15. Before going on to the second method:
    - Choose TABLE → REMOVE ALL LINKS for the tables
    - Click on the SELECT NONE button 
    - Close the Identify Results window
    - Remove “Attributes of Ab_twp.shp” from the TABLE GUI

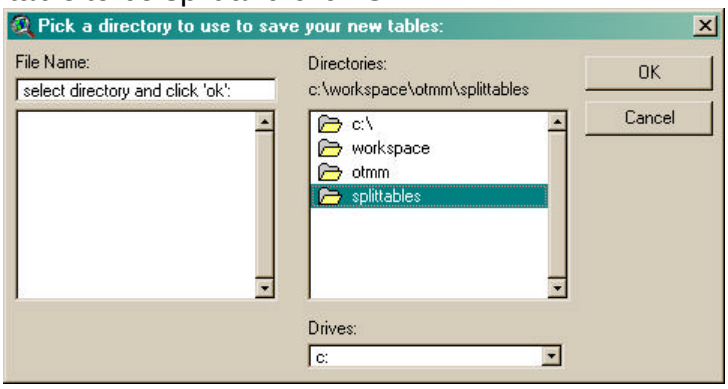
Second Method – Joining Tables

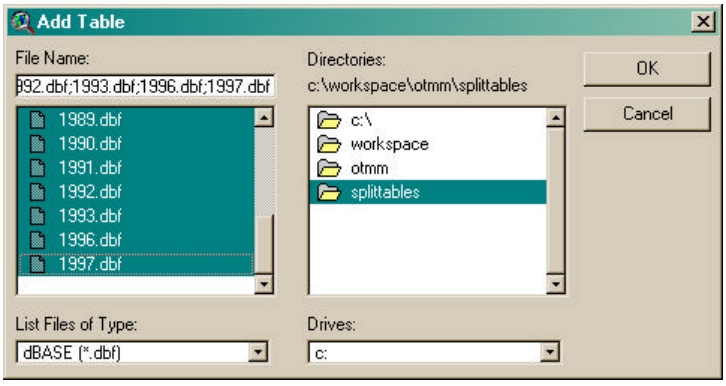
16. In the TABLE GUI, make sure **selected.dbf** is open and click on the SPLIT TABLE BY ATTRIBUTE button 





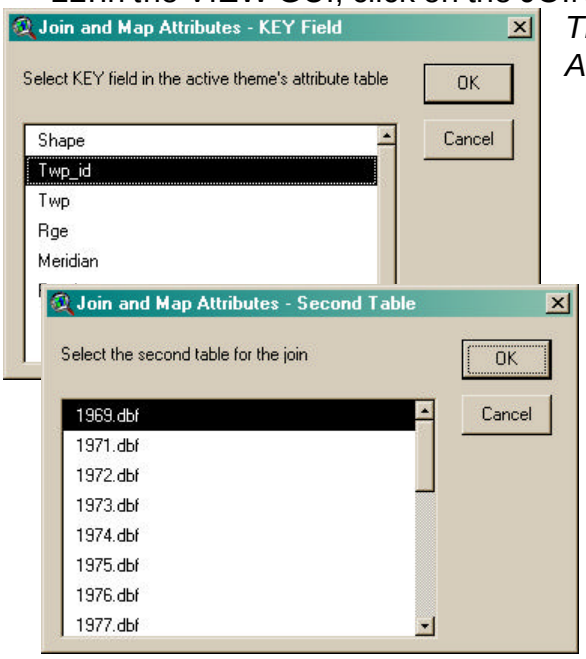
17. Select **selected.dbf** as the table to be split and click OK
18. Select **`Yearsurvey`** as the attribute split field and click OK
19. Navigate to where you want to save the split table outputs and click OK
20. Add the newly created tables to the project
21. Close them all to reduce “clutter” in the window



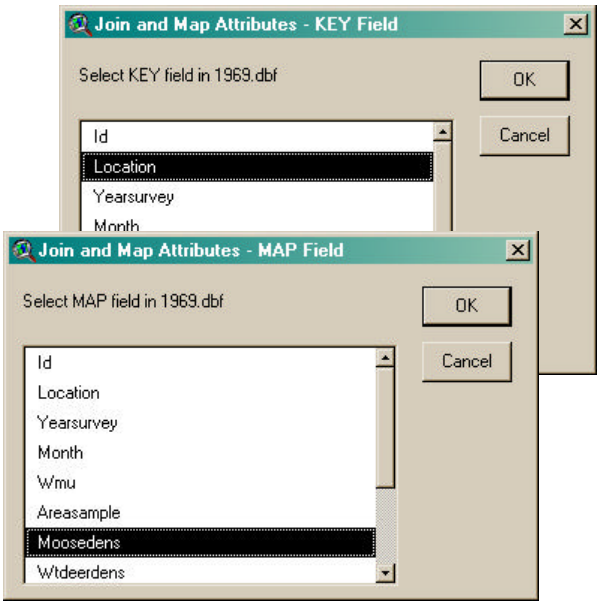


22. In the VIEW GUI, click on the JOIN AND MAP ATTRIBUTES button 

This script works on the FIRST ACTIVE THEME in the view!

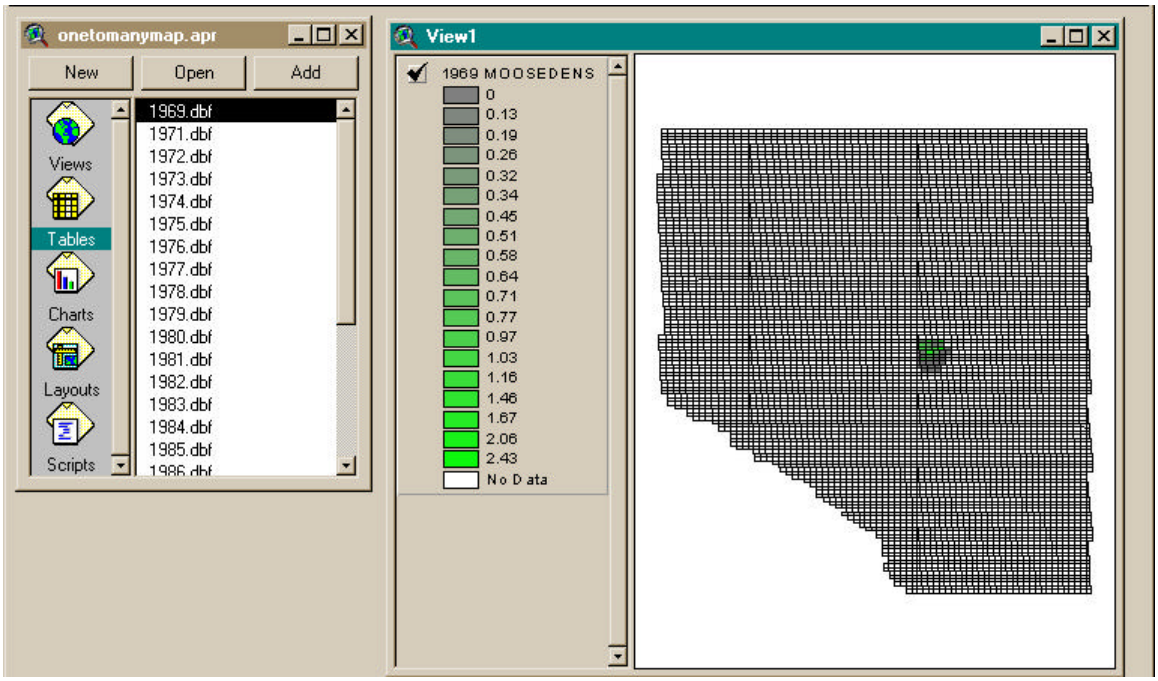


23. Select **`Twp_id`** as the KEY field for the first table and click OK
24. Select **1969.dbf** as the second table for the join and click OK
25. Select **`Location`** as the KEY field for the second table and click OK



26. Select the ID field and click OK
27. Select the MAP field and click OK

You may choose any of the joined attribute fields to map first, and change the field for subsequent mapping. A preset legend scheme is applied to the symbolization of the MAP field values.



28. Click on the MAP NEXT UNIQUE ATTRIBUTE button 
29. Select a new joined field attribute to map
30. To join a different year table to the shapefile, click on the JOIN AND MAP ATTRIBUTES button again, but select another year's table (e.g. 1971.dbf)

The scripts automate what could very well be a tedious process if you were to perform individual joins from scratch. Note that the shapefile is conveniently ***renamed*** *(as the year and attribute) with each join and attribute symbolization. You will notice that if there is no data for a particular attribute (e.g. 1997's COYOTEDENS) then the shapefile is symbolized as No Data.*

The underlying shapefile is still ***Ab_twp.shp****. Each subsequent mapping of the next attribute uses the same underlying shapefile, but symbolizes on your selected join field. To make the join permanent, and create a new shapefile with the joined attributes, choose THEME → CONVERT TO SHAPEFILE.*

If you wish to change the class colors for the legend, open the script named ***_View.SetClass*** *to change the* ***firstColor*** *and* ***secondColor*** *variables to any of:*

- *GetBlack*
- *GetBlue*
- *GetCyan*
- *GetGray*
- *GetGreen*
- *GetMagenta*
- *GetRed*
- *GetWhite*
- *GetYellow*

If you wish to change the color ramp classification scheme then replace ***currentLegend.Unique(theFTheme, mapField.AsString)*** *with any of the following:*

- *Interval (theFTheme, mapField.AsString, 5)*
- *Natural (theFTheme, mapField.AsString, 5)*
- *Quantile (theFTheme, mapField.AsString, 5)*
- *StdDev (theFTheme, mapField.AsString, 5)*

You may replace the example value "5" with a number of classes of your choice.

Click on the COMPILE button when finished modifying the scripts!