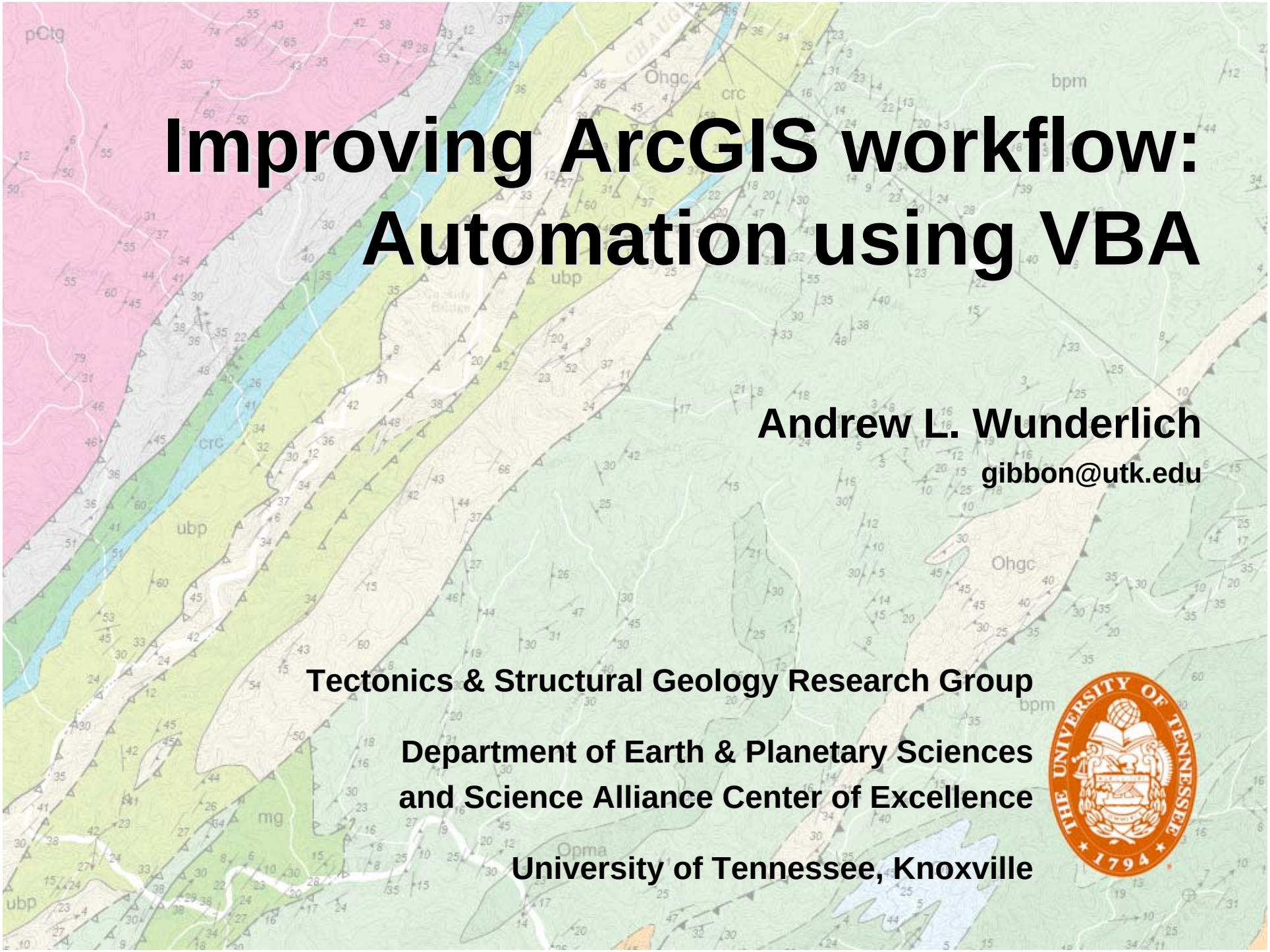


Improving ArcGIS workflow: Automation using VBA

Andrew L. Wunderlich

gibbon@utk.edu

Tectonics & Structural Geology Research Group

Department of Earth & Planetary Sciences

and Science Alliance Center of Excellence

University of Tennessee, Knoxville

Introduction

Many tasks associated with editing and QCing GIS datasets can be tedious, repetitive, and time consuming.

Automation of some of these tasks, such as systematic panning around a map during editing and inputting feature attribution, is very desirable.

BUT HOW?

Customizing ArcGIS

- Many forms of customization:
  - Model Builder
  - Layer definition files
  - Simple macros (keystroke combinations)
  - Coding with VBA
  - Stand-alone applications

- All ArcGIS applications (ArcMap, ArcCatalog, etc.) include the Visual Basic for Applications development environment.

VBA in ArcGIS

- From the ArcGIS Desktop Help:

  “Using VBA, you author macros that are stored within the document/template structure of the application you are extending by writing code using the Visual Basic language. You can also create custom commands and tools, called UIControls. UIControls are macros that also contain hooks into the application framework so that you can respond to actions that happen on the buttons or commands you create.”

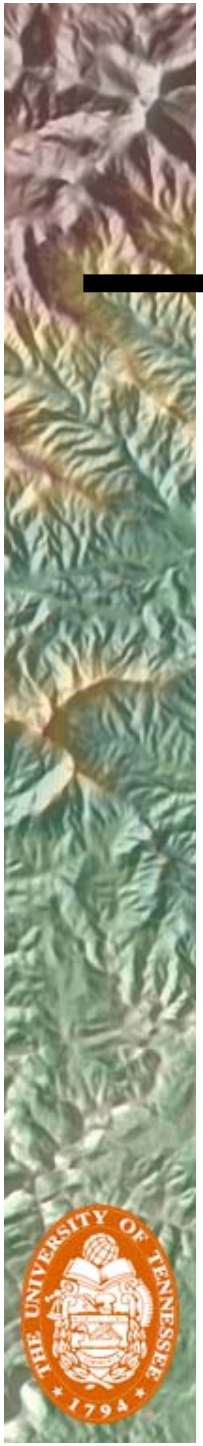
Programming can be scary…

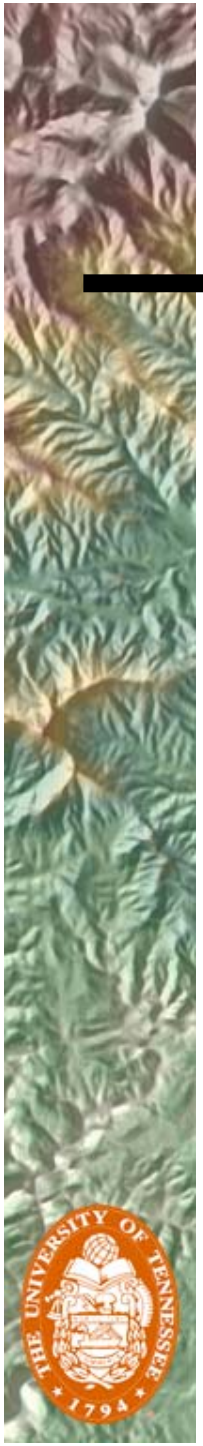
Don’t panic!

- Many resources available for examples and help with VBA development.
- Huge amount of sample applications and scripts already available for common tasks.
- Simpler syntax makes understanding VB code easier than other languages.

REMEMBER:

Scripting does not have to be complex!





Programming is like illustration…

Imagine that you have been asked to create an illustration of a dog…

Programming is like illustration…

Imagine that you have been asked to create an illustration of a dog…

Programming is like illustration…

Imagine that you have been asked to create an illustration of a dog…

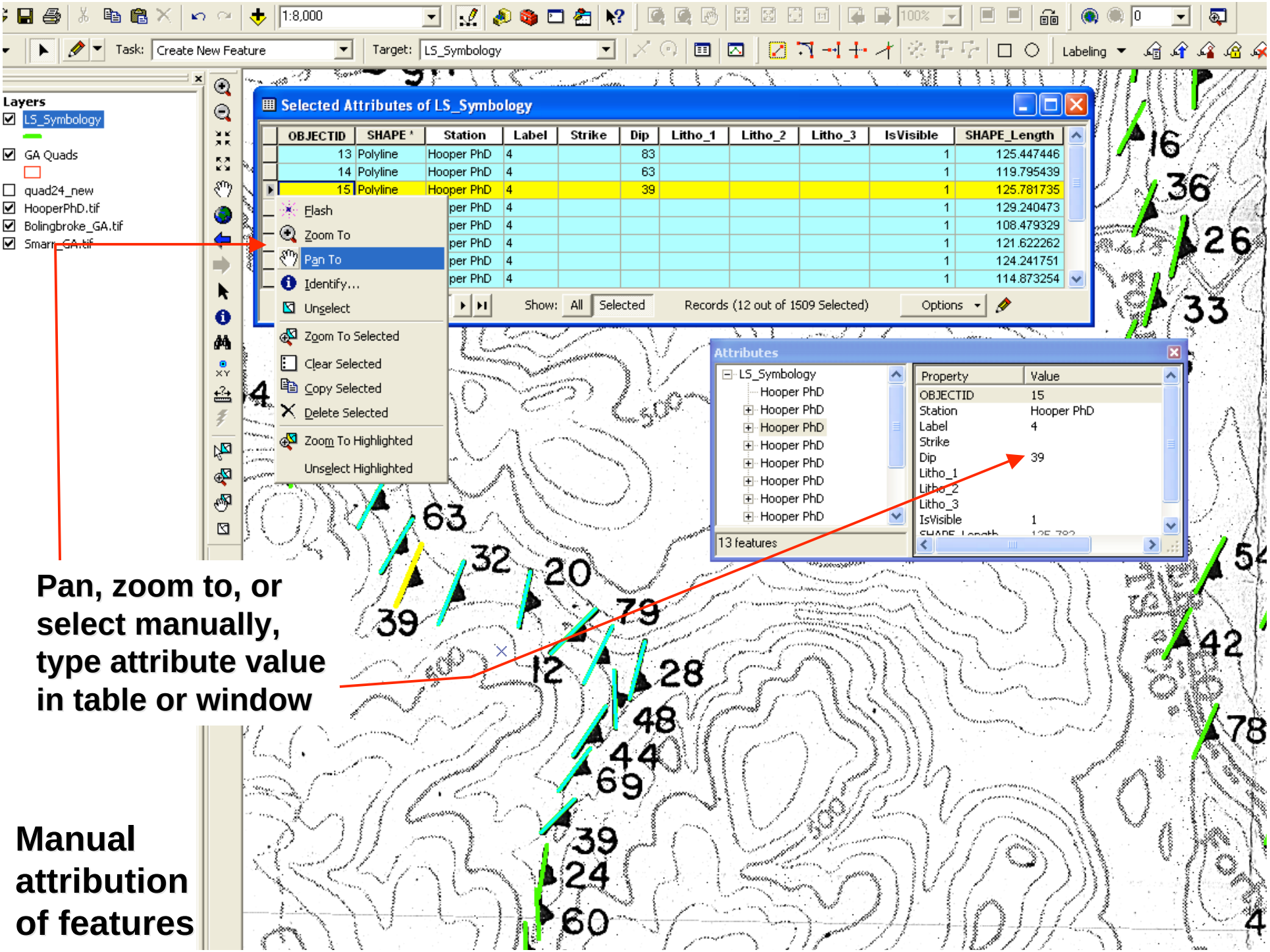
It’s about time!

- Both illustrations are fundamentally the same: they are dogs!
- Vary the complexity and functionality of scripts based on criteria and TIME
- Ask yourself:
  - Does the script need to be bare-bones or robust?
  - How much time can be budgeted for development?
  - How much time will be saved? In the short-term? Long-term?
  - How can I justify the time I need to develop a custom solution to management/supervisor?

Scenario for automation…

Features digitized from georeferenced scan of an old paper map need attribution lifted from labels near points.

1) Start edit session.
2) Open attribute table, select features for attribution.
3) Right-click entry, pan/zoom to feature.
4) Type attribute in table window or feature attribute window and press <Enter>.
5) Pan to or click next feature and repeat.


Task: Create New Feature
Target: LS_Symbology
Layers
LS_Symbology
GA Quads
quad24_new
HooperPhD.tif
Bolingbroke_GA.tif
Selected Attributes of LS_Symbology
OBJECTID | SHAPE * | Station | Label | Strike | Dip | Litho_1 | Litho_2 | Litho_3 | IsVisible | SHAPE_Length
13 | Polyline | Hooper PhD | 4 | | 83 | | | | 1 | 125.447446
14 | Polyline | Hooper PhD | 4 | | 63 | | | | 1 | 119.795439
15 | Polyline | Hooper PhD | 4 | | 39 | | | | 1 | 125.781735
Flash
Zoom To
Pan To
Identify...
Unselect
Zoom To Selected
Clear Selected
Copy Selected
Delete Selected
Zoom To Highlighted
Unselect Highlighted
Show: All Selected
Records (12 out of 1509 Selected)
Options
Attributes
LS_Symbology
Hooper PhD
13 features
Property | Value
OBJECTID | 15
Station | Hooper PhD
Label | 4
Strike
Dip | 39
Litho_1
Litho_2
Litho_3
IsVisible | 1
Pan, zoom to, or select manually, type attribute value in table or window
Manual attribution of features

Time the process

- Attribute some features manually (maybe a few dozen) and record the time.
- Divide the time it took by the number of features attributed.
  - In this scenario it took about me about 25 seconds per feature.
- Extrapolate using total number of features to get the total time estimate.
  - In this scenario there are about 1500 features to attribute so…

25 x 1500 = 37500 seconds / 3600 = ~10.5 hours!

Break the process down

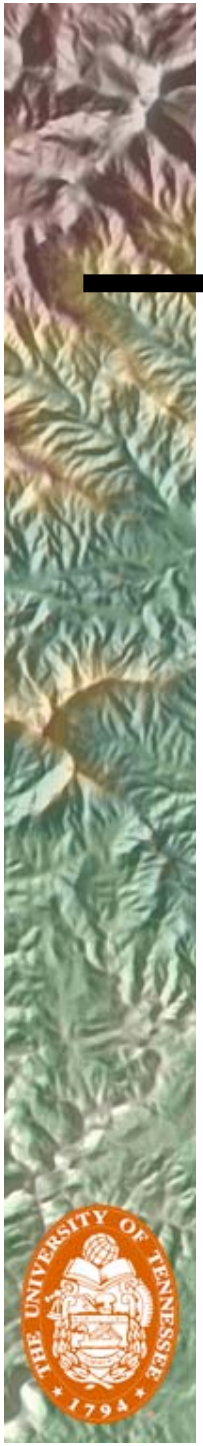
- In it's simplest form, the script needs to:
  - See the selected features in a map layer
  - Pan/zoom to the first selected feature
  - Pop up a window to allow user input
  - Find the field for storing the user input and write the value to that field
  - Move to the next feature
- Also:
  - Must allow user to cancel
  - Must handle basic errors without scaring the user too badly

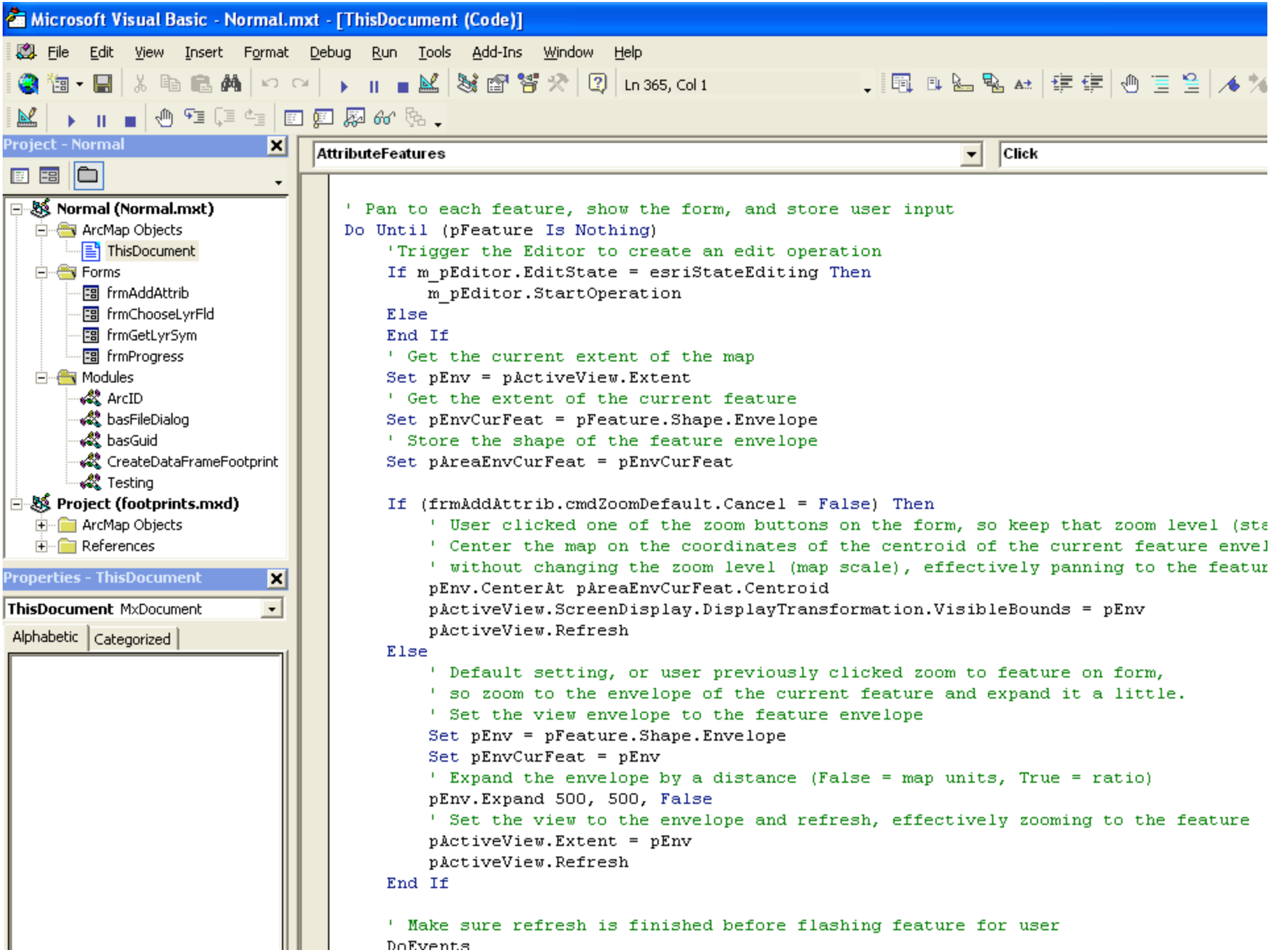
What I came up with…

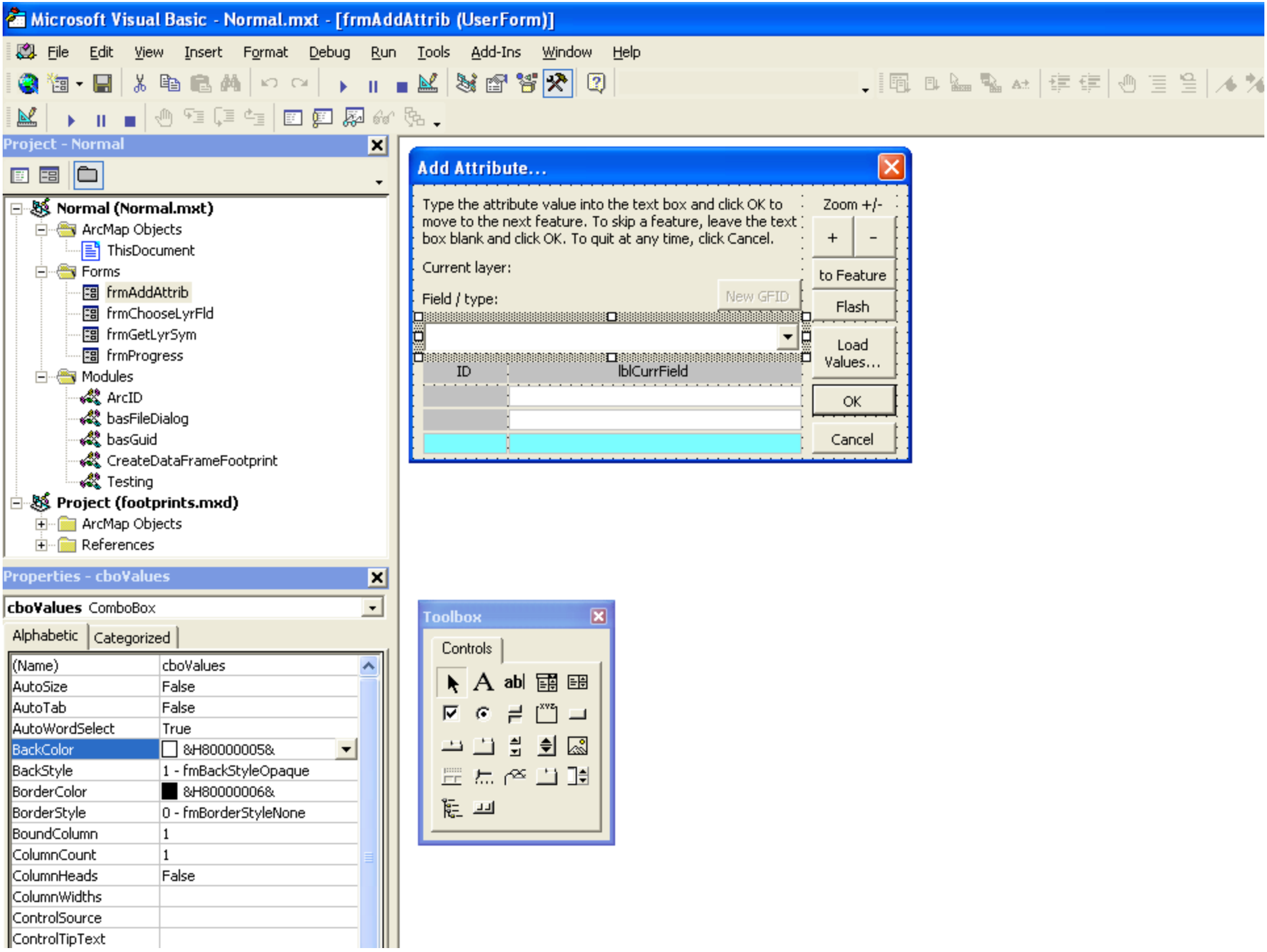
- Using tutorials and code samples I was able to develop a *very* basic attribute tool in a day.
  - Not sophisticated, worked with the selected features in the highlighted layer in the TOC.
  - Handled features one by one using a simple feature cursor:
    - Loops through features
    - Sets view to the current feature “envelope”
    - Pops up an “InputBox”
    - Writes value to predefined field in table
  - Very simple error handling: script simply quits (gracefully) when a problem is encountered
- With this simple script, time was cut down to around 4-5 seconds per feature! (~2 hours)

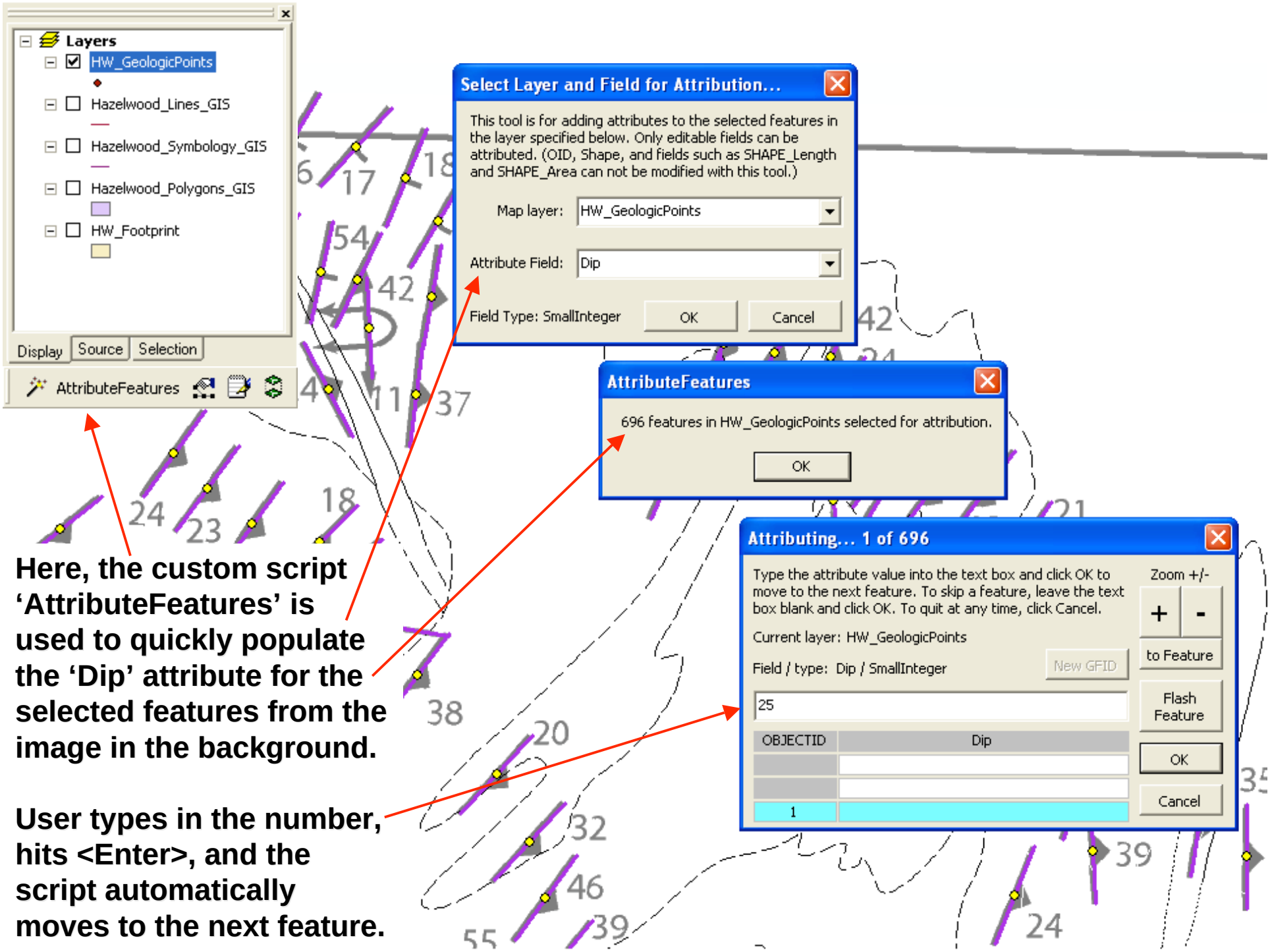
With a bit more effort…

- Additional code samples and research (and another day):
  - User can choose layer and attribute field, and query features if none are selected
  - Makes sure user input is appropriate for the target field
  - Cleaner, more efficient code
  - Better error handling
- Further developments (another day or two):
  - Ability to trigger edit session (undo!)
  - Ability to edit features classes that participate in relationships
  - Improved interface: buttons for zooming in and out and flashing current feature




Microsoft Visual Basic - Normal.mxt - [ThisDocument (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 365, Col 1
Project - Normal
Normal (Normal.mxt)
ArcMap Objects
ThisDocument
Forms
frmAddAttrib
frmChooseLyrFld
frmGetLyrSym
frmProgress
Modules
ArcID
basFileDialog
basGuid
CreateDataFrameFootprint
Testing
Project (footprints.mxd)
ArcMap Objects
References
Properties - ThisDocument
ThisDocument MxDocument
Alphabetic
Categorized
AttributeFeatures
Click
' Pan to each feature, show the form, and store user input
Do Until (pFeature Is Nothing)
'Trigger the Editor to create an edit operation
If m_pEditor.EditState = esriStateEditing Then
m_pEditor.StartOperation
Else
End If
' Get the current extent of the map
Set pEnv = pActiveView.Extent
' Get the extent of the current feature
Set pEnvCurFeat = pFeature.Shape.Envelope
' Store the shape of the feature envelope
Set pAreaEnvCurFeat = pEnvCurFeat
If (frmAddAttrib.cmdZoomDefault.Cancel = False) Then
' User clicked one of the zoom buttons on the form, so keep that zoom level (sta
' Center the map on the coordinates of the centroid of the current feature envel
' without changing the zoom level (map scale), effectively panning to the featur
pEnv.CenterAt pAreaEnvCurFeat.Centroid
pActiveView.ScreenDisplay.DisplayTransformation.VisibleBounds = pEnv
pActiveView.Refresh
Else
' Default setting, or user previously clicked zoom to feature on form,
' so zoom to the envelope of the current feature and expand it a little.
' Set the view envelope to the feature envelope
Set pEnv = pFeature.Shape.Envelope
Set pEnvCurFeat = pEnv
' Expand the envelope by a distance (False = map units, True = ratio)
pEnv.Expand 500, 500, False
' Set the view to the envelope and refresh, effectively zooming to the feature
pActiveView.Extent = pEnv
pActiveView.Refresh
End If
' Make sure refresh is finished before flashing feature for user
DoEvents


Microsoft Visual Basic - Normal.mxt - [frmAddAttrib (UserForm)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - Normal
Normal (Normal.mxt)
ArcMap Objects
ThisDocument
Forms
frmAddAttrib
frmChooseLyrFld
frmGetLyrSym
frmProgress
Modules
ArcID
basFileDialog
basGuid
CreateDataFrameFootprint
Testing
Project (footprints.mxd)
ArcMap Objects
References
Properties - cboValues
cboValues ComboBox
Alphabetic
Categorized
(Name) cboValues
AutoSize False
AutoTab False
AutoWordSelect True
BackColor &H80000005&
BackStyle 1 - fmBackStyleOpaque
BorderColor &H80000006&
BorderStyle 0 - fmBorderStyleNone
BoundColumn 1
ColumnCount 1
ColumnHeads False
ColumnWidths
ControlSource
ControlTipText
Add Attribute...
Type the attribute value into the text box and click OK to move to the next feature. To skip a feature, leave the text box blank and click OK. To quit at any time, click Cancel.
Current layer:
Field / type:
New GFID
ID
lblCurrField
Zoom +/-
+
-
to Feature
Flash
Load Values...
OK
Cancel
Toolbox
Controls



Here, the custom script 'AttributeFeatures' is used to quickly populate the 'Dip' attribute for the selected features from the image in the background.

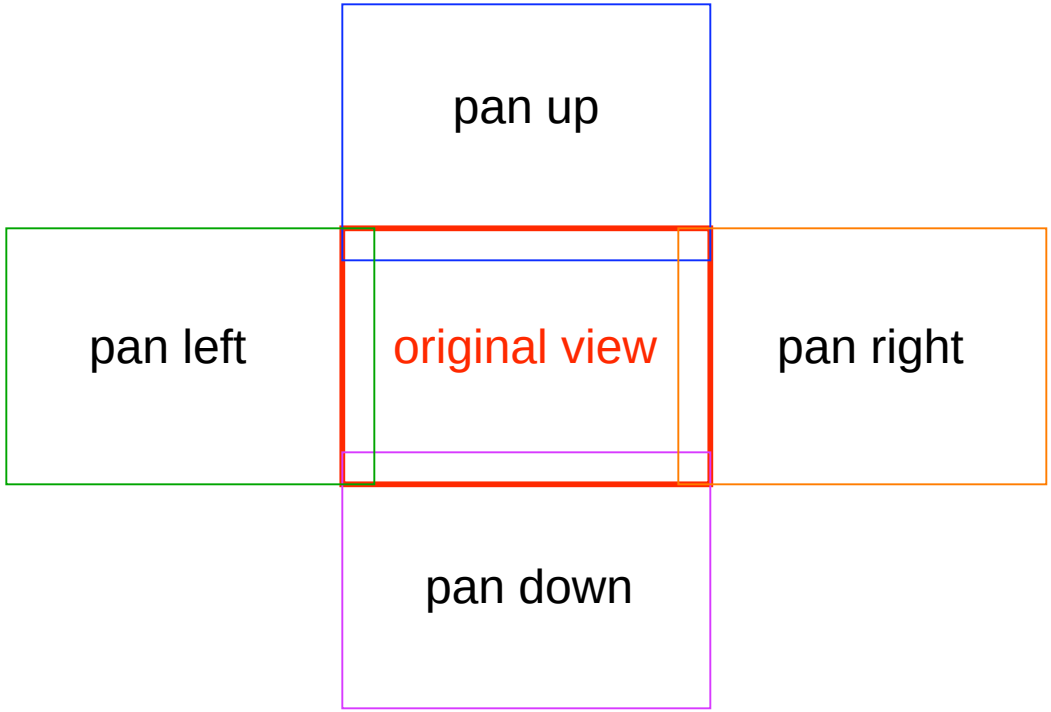
User types in the number, hits <Enter>, and the script automatically moves to the next feature.

Other useful automations…

- Pan current view up, down, left or right with a small amount of overlap (~10%).
- Create data frame current view “footprints”.
- Densify features (lines and polygons) with additional vertices (part of footprint script, will not be discussed separately)
- Write layer symbology reports for map layers that utilize the unique value renderer.

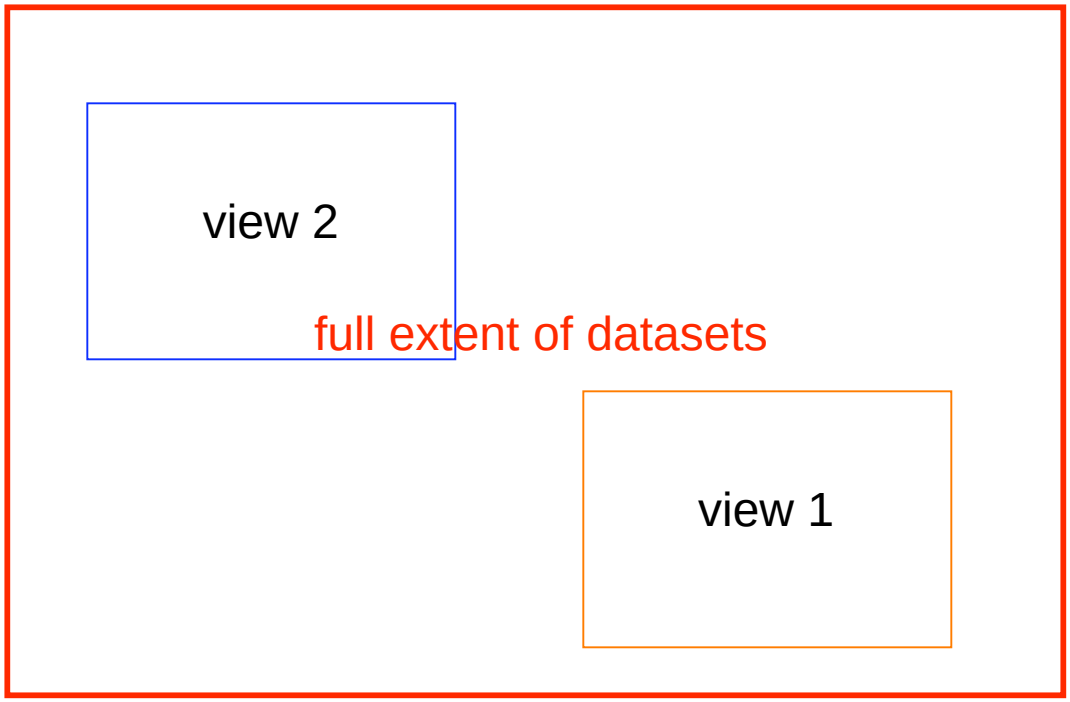
Systematic panning…

- Pan current view up, down, left or right with a small amount of overlap (~10%).
  - Speeds panning around a map for doing editing or QA/QC work.

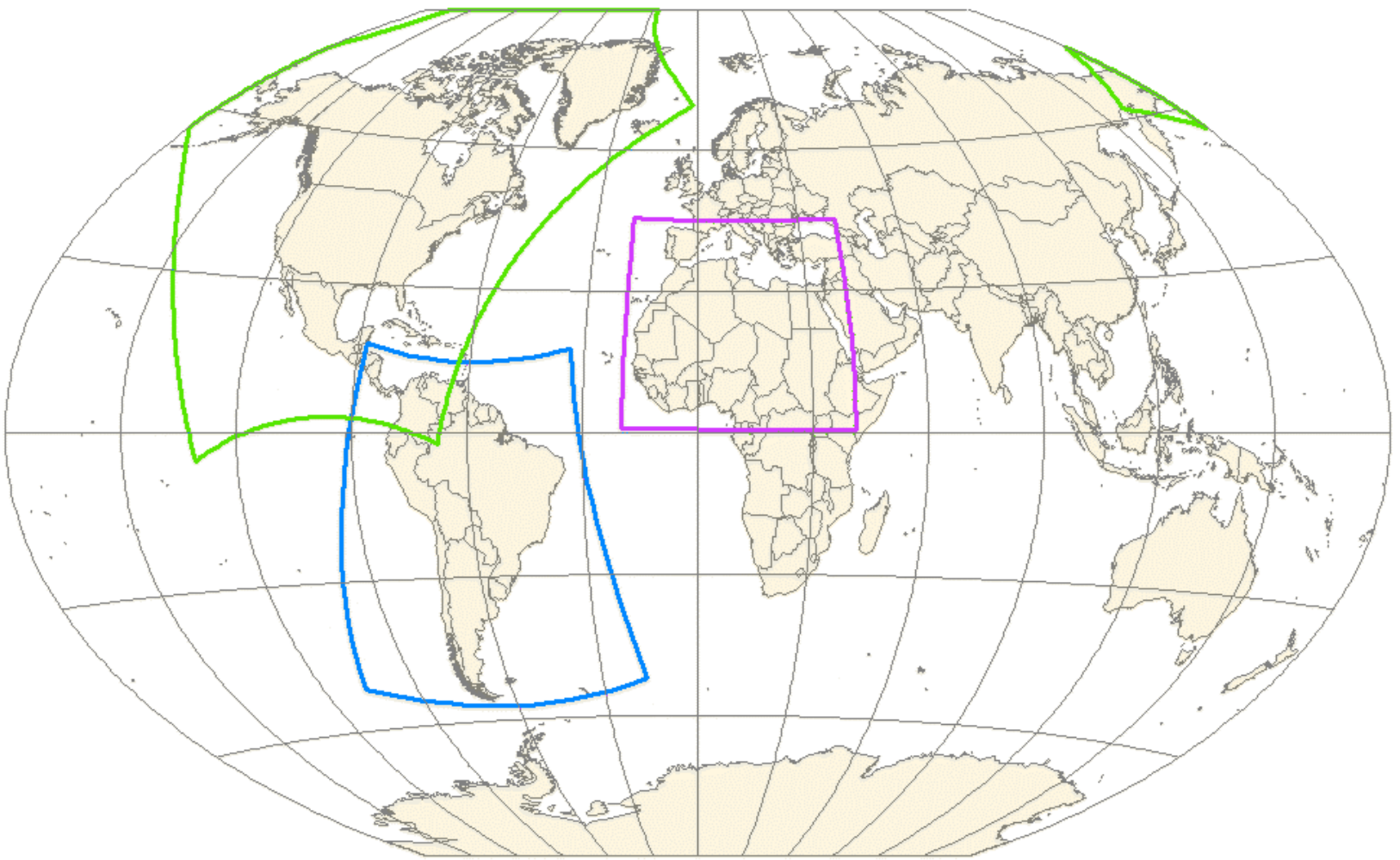


Data frame footprints

- Create a polygon shapefile that represents the current view extent.
  - Useful for creating data frame “footprints” for inset map locations or graphical indexing.



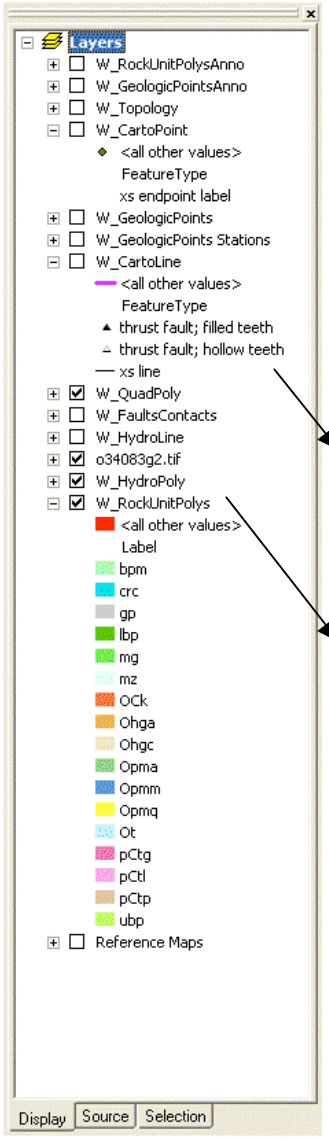
Example of graphical indexing



Layer symbology report

- Writes a comma separated values (CSV) text file containing the graphical attributes of layer symbology.
  - Useful for checking colors, line weights, etc. in layers that use a unique value renderer:
    - Line and Fill color in RGB and CMYK
    - Line (or outline) weight
    - Rotation (point classes)
    - Symbol size (point classes)
    - X, Y Offset (point classes)

Layer symbology report



Microsoft Excel - uvr_report.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Layer	Geometry	Field	Value	Graphic	R	G	B	C	M	Y	K	Width	Angle	Size	XOffset	YOffset
2	W_CartoPoint	Point	FeatureType	xs endpoint label	Fill	237	234	218	7	8	15	0		0	1	0	0
3	W_GeologicPoints	Point	Label50	ADCOH	Fill	0	0	0	0	0	0	100		0	6	0	0
4	W_GeologicPoints	Point	Label50	anticline inclined	Fill	0	0	0	0	0	0	100		90	18	0	0
5	W_GeologicPoints	Point	Label50	anticline overturned	Fill	0	0	0	0	0	0	100		0	18	0	0
6	W_GeologicPoints	Point	Label50	bedding horizontal	Fill	0	0	0	0	0	0	100		0	18	0	0
7	W_GeologicPoints	Point	Label50	bedding inclined	Fill	0	0	0	0	0	0	100		270	18	0	0
8	W_GeologicPoints	Point	Label50	bedding vertical	Fill	0	0	0	0	0	0	100		90	18	0	0
9	W_GeologicPoints	Point	Label50	fold axes surface	Fill	0	0	0	0	0	0	100		0	18	0	0
10	W_GeologicPoints	Point	Label50	foliation inclined	Fill	0	0	0	0	0	0	100		270	18	0	0
11	W_GeologicPoints	Point	Label50	foliation inclined mylonitic	Fill	0	0	0	0	0	0	100		0	18	0	0
12	W_GeologicPoints	Point	Label50	lineation inclined	Fill	0	0	0	0	0	0	100		0	18	0	0
13	W_GeologicPoints	Point	Label50	mine prospect quarry	Fill	0	0	0	0	0	0	100		0	12	0	0
14	W_GeologicPoints	Point	Label50	syncline inclined	Fill	0	0	0	0	0	0	100		90	18	0	0
15	W_GeologicPoints	Point	Label50	syncline overturned	Fill	0	0	0	0	0	0	100		0	18	0	0
16	W_CartoLine	Polyline	FeatureType	thrust fault; filled teeth	Line	0	0	0	0	0	0	100	10				
17	W_CartoLine	Polyline	FeatureType	thrust fault; hollow teeth	Line	0	0	0	0	0	0	100	10				
18	W_CartoLine	Polyline	FeatureType	xs line	Line	0	0	0	0	0	0	100	0.5669				
19	W_FaultsContacts	Polyline	Label50	contact certain	Line	0	0	0	0	0	0	100	0.432				
20	W_FaultsContacts	Polyline	Label50	contact approximate	Line	0	0	0	0	0	0	100	0.432				
21	W_FaultsContacts	Polyline	Label50	contact inferred	Line	0	0	0	0	0	0	100	0.432				
22	W_FaultsContacts	Polyline	Label50	contact concealed	Line	0	0	0	0	0	0	100	0.432				
23	W_FaultsContacts	Polyline	Label50	fault certain	Line	0	0	0	0	0	0	100	1.08				
24	W_FaultsContacts	Polyline	Label50	fault approximate	Line	0	0	0	0	0	0	100	1.08				
25	W_FaultsContacts	Polyline	Label50	fault approximate queried	Line	0	0	0	0	0	0	100	10				
26	W_FaultsContacts	Polyline	Label50	fault concealed	Line	0	0	0	0	0	0	100	0.432				
27	W_RockUnitPolys	Polygon	Label	bpm	Fill	173	255	181	32	0	29	0					
28	W_RockUnitPolys	Polygon	Label	crc	Fill	28	224	235	89	12	8	0					
29	W_RockUnitPolys	Polygon	Label	gp	Fill	204	204	204	0	0	0	20					
30	W_RockUnitPolys	Polygon	Label	lbp	Fill	76	199	0	48	0	80	22					
31	W_RockUnitPolys	Polygon	Label	mg	Fill	112	255	82	56	0	68	0					
32	W_RockUnitPolys	Polygon	Label	mz	Fill	230	255	242	10	0	5	0					
33	W_RockUnitPolys	Polygon	Label	OCk	Fill	245	115	56	0	51	74	4					
34	W_RockUnitPolys	Polygon	Label	Ohga	Fill	250	181	74	2	29	71	0					
35	W_RockUnitPolys	Polygon	Label	Ohgc	Fill	240	232	196	6	9	23	0					
36	W_RockUnitPolys	Polygon	Label	Opma	Fill	150	212	145	24	0	26	17					
37	W_RockUnitPolys	Polygon	Label	Opmm	Fill	89	153	217	50	25	0	15					
38	W_RockUnitPolys	Polygon	Label	Opmq	Fill	255	255	64	0	0	75	0					
39	W_RockUnitPolys	Polygon	Label	Ot	Fill	194	240	255	24	6	0	0					
40	W_RockUnitPolys	Polygon	Label	pCtg	Fill	240	110	181	0	51	23	6					
41	W_RockUnitPolys	Polygon	Label	pCtl	Fill	252	163	252	1	36	1	0					
42	W_RockUnitPolys	Polygon	Label	pCtp	Fill	227	196	158	0	12	27	11					
43	W_RockUnitPolys	Polygon	Label	ubp	Fill	196	255	102	23	0	60	0					
44	W_RockUnitPolys	Polygon	Label	water	Fill	255	255	255	0	0	0	0					
45																	
46																	
47																	
48																	

uvr_report

Ready

NUM

start

Microsoft O... Gmail - Inbo... Microsoft P... ArcGIS Des... Whetstone.... footprints.... Shapes Microsoft E... 12:50 PM

New developer tips…

- Comment code well; it makes scripts easier to follow and to edit.
- New tools can be based on old ones
  - Code that is needed in many of these scripts can be made into “functions” that can be called by any script.
- Save versioned copies of scripts.
  - If a mistake is made in a working script and it causes a crash, old code can be used for comparison or reversion.

And remember: great artists steal…

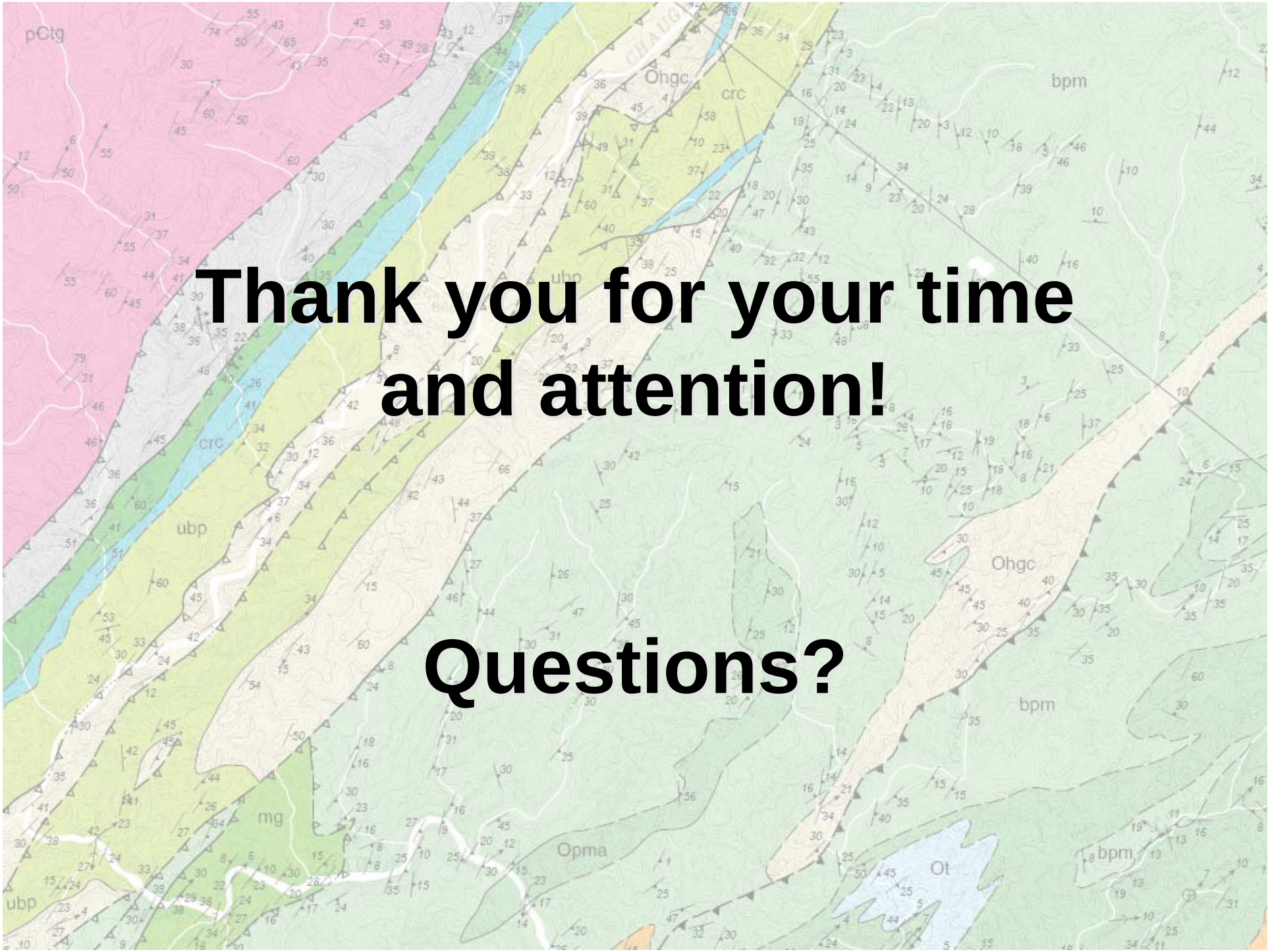
…so use these free resources!

- Get started with ArcGIS Desktop help:
  - Tutorials, code samples, and instructions for creating custom UIControls (Search for “vba”)
- ESRI Support website (support.esri.com)
  - Search key words for solutions: chances are, it’s been done already!
- ESRI Developer Network (edn.esri.com)
  - Advanced object reference and code samples
- Just Google it!

Final Thoughts

- Time needed to develop custom solutions can be justified through long-term savings.
- Basic programming skills can be achieved in a relatively short period of time with help from tutorials and the bevy of free code samples.
- All code is FREE, taken from PUBLIC resources and forums.
- Contact me if interested in getting source code for any of the scripts discussed:

  gibbon@utk.edu



Thank you for your time and attention!

Questions?