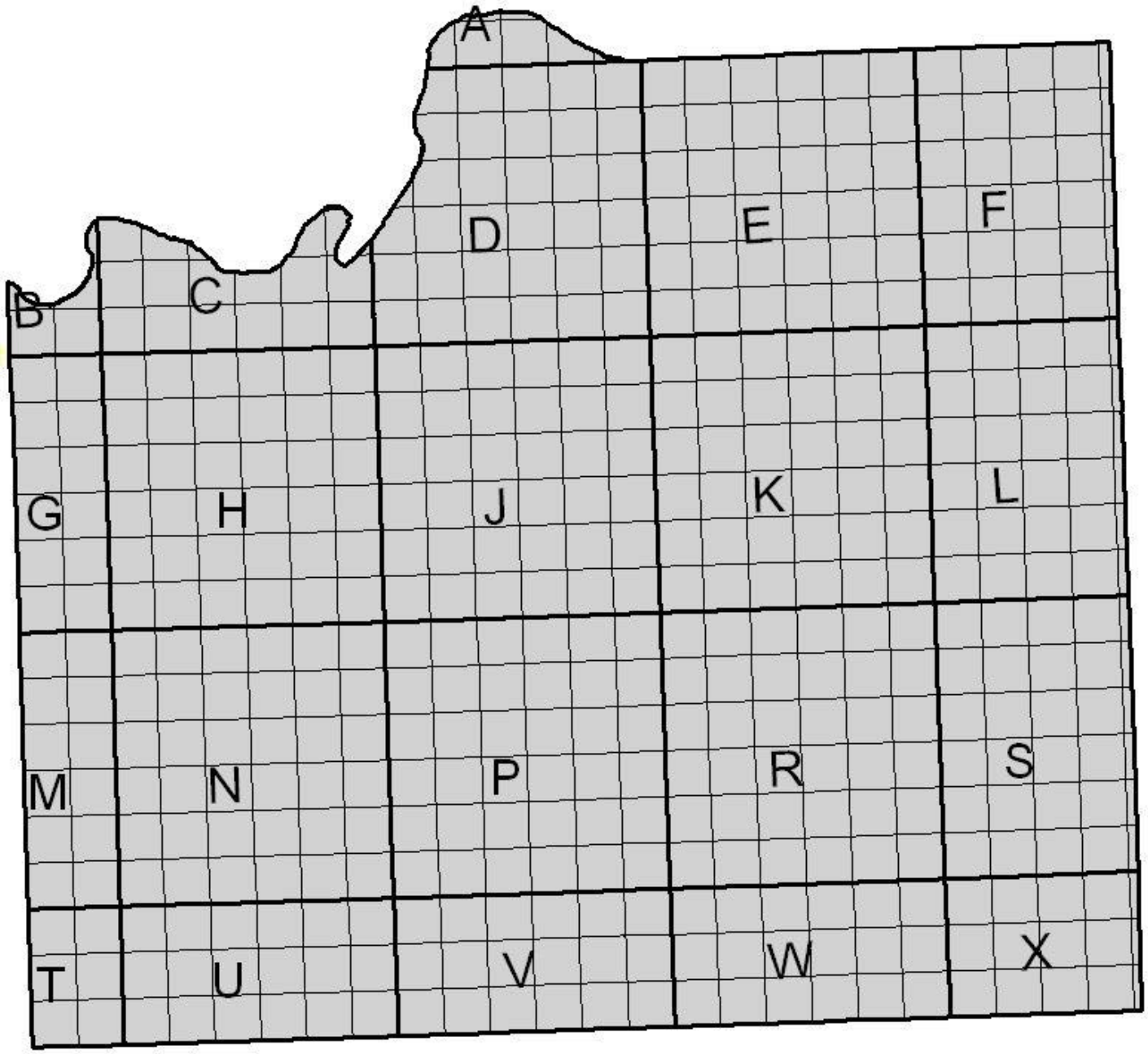
Creating Merged Themes from Tiled Data

Dan Steen

Johnson County, KS

Automated Information Mapping Systems


A
B
C
D
E
F
G
H
J
K
L
M
N
P
R
S
T
U
V
W
X

Need for a Solution

- Various datasets are distributed to data licensees based on the area they have licensed.
- New application development calls for countywide coverages.
- Users want single coverage solutions.

Solution

- Create an interface for customizing and controlling the process
- Allow the user to access all data sets and types
- Ability to join tables to shapefiles

Design Tools

- ArcView Dialog Designer – Used about five separate dialogs to control design.
- ArcView Extensions – creates a portable, easy to use application.

Extension

- Used ESRI's samples for creating extension.
- Run a startup script that sets up data paths and loads data files into hidden directory.

Application Considerations

- Prevent users from getting into main attribute tables.
- AV errors out after certain number of ftabs have been opened for editing.

Attribute Table Control

```
if (av.GetProject.FindGUI("Hidden Tables") <> nil) then
    return nil
else
    av.GetProject.MakeGUIName("Hidden Tables")
    newGUI = av.FindGUI("Table").clone
    newGUI.SetName("Hidden Tables")
    newGUI.SetDocBaseName("Hidden Tables")
    newGui.SetTitle("Hidden Tables")

newGUI.SetVisible(false)
av.GetProject.AddGUI(newGUI)
end
```

Performance

- About 2 hours to merge 506 property coverages into one file
- Resulting shapefile is about 120 megs in size

Code for Merge Themes

```
for each rec in theRecordsToMerge
   newRec = mergeFTab.AddRecord
   inField  = inFTab.FindField( "Shape" )
   outField = mergeFTab.FindField( "Shape" )
   mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue(
   inField, rec ))

   if (fieldList.Count > 0) then
     for each f in fieldList
       fName    = f.GetName
       inField  = inFTab.FindField( fName )

       ' Skip field if not found in inFTab...
       if ( inField <> Nil ) then
         outField = mergeFTab.FindField( fName )
         aValue    = inFTab.ReturnValue( inField, rec )
         mergeFTab.SetValue( outField, newRec, aValue )
       end

     end   ' for each f
   end   ' if count
end   ' for each rec
```

Read list from text file

```
if (byFile.IsSelected = true) then
atextFile = FileDialog.Show("*.txt", "*.txt", "Select Text File")
  tf = TextFile.Make( atextfile, #FILE_PERM_READ )
  source = tf.Read( tf.GetSize )
    tf.Close

'Interpret the file into a string
'File must be a text file with returns between each value and a return at the
    end.
'each tile must have a 0 if it is a single digit (i.e. a08)

newString = {}
while (source.count >3)
newstring.Add(""+source.Left(4)+"")
source = source.Right(source.count-4)
end
theTileList.DefineFromList(newstring)

End
```