Desktop Notes

User Level
New/Casual Advanced

ArcView GIS

Creating a Composite Marker

By Stephen South, ESRI Technical Support

It may be useful to create a composite marker by overlaying two different symbol markers. ArcView GIS does not create fonts, but it can combine symbols created in another program.

First use a font-generating program such as Fontographer by Macromedia to create a TrueType font. When creating the font, be sure to save it as a TrueType font. After the font is created and saved to a folder, perform the following steps to create a composite marker in ArcView GIS.

Load the TrueType font onto the operating system.

1. From the Windows interface, go to Start > Settings > Control Panel.
2. Open the Fonts folder.
3. From the File menu select Install New Font.
4. Navigate to the location of the new TrueType font file (i.e., C:\stuff\test4.ttf).
5. The new font can be copied to the Fonts folder or it can be used in its current location.

Create a composite marker in ArcView GIS.

1. Start ArcView GIS.
2. Open a new view.
3. Hold down the Control key and press P (Cntrl + P) or choose Window > Show Symbol Window to bring up the Symbol Palette.
4. Click on the Palette Manager button located on the top right of the Symbol Palette.
5. In the drop-down box change Type from All to Marker.
6. Click on the Clear button.
7. Go back to the Project window and double click on the Script icon to open a new script window.
8. Type in the script shown in Figure 1.
9. Compile and run the script.

To change the marker colors, edit the SetColor()functions in the script to use any of the values shown in Figure 2.

The color of the second marker can be changed by using the Color Palette in the Symbols Palette. Open a view, click on the Draw Point tool in the button bar to make it active, and place a marker in the view. With this marker selected, use the shortcut (Cntrl + P) to bring up the Symbol Palette window. Click on the Color Palette and select the desired color. Open the Marker Palette from the Symbol Palette. It should contain only one marker—the one just created.

Figure 3 shows a sample composite marker. In this example, the number 36 does not line up with the state outline. To correct this, go back to the font-generating software and use the selection tool to move the contents of the cell. Note the coordinates because some experimenting is often necessary to get it right. Every time the font is changed, delete the old version from the operating system and install the new version using the procedure described previously. If the new version of the font is installed with the name used to create the marker in ArcView GIS, the marker will not need to be re-created.

FIGURE 1

```
'— This script uses two symbols to create a composite marker
av.GetSymbolWin.Open
av.GetSymbolWin.SetPanel(#SYMBOLWIN_PANEL_MARKER)
'— Set the properties of each marker
'— symbol according to your requirements
m1 = BasicMarker.Make
m1.SetStyle(#BASICMARKER_STYLE_PATTERN)
m1.SetSize(16)
'— Edit the next line to change the color of first symbol
m1.SetColor(Color.GetRed)
'— Lock the first color
m1.SetColorLock(true)
'— Change "Test4" in the next line with your font name
m1.SetFont(Font.Make("Test4","normal"))
'— Change the "B" in the next line to the letter in the
'— font generating software that represents the symbol
'— you want to use.
m1.SetCharacter("B".asascii)
m2 = BasicMarker.Make
m2.SetStyle(#BASICMARKER_STYLE_PATTERN)
m2.SetSize(16)
'— Edit the next line to change the color of second symbol
m2.SetColor(Color.GetBlue)
'— If you want to lock the second color,
'— you must uncomment the next line
'm2.SetColorLock(true)
'— Change "Test4" in the next line with your font name
m2.SetFont(Font.Make("Test4","normal"))
'— Change the "C" in the next line to the letter in the
'— font generating software that represents the symbol
'— you want to use.
m2.SetCharacter("C".asascii)
'— Make as many marker symbols as there
'— are layers in your composite symbol
cm = CompositeMarker.Make(SymbolList.FromList({m1,m2}))
mp = av.GetSymbolWin.GetPalette.GetList(#PALETTE_LIST_MARKER)
mp.Add(cm)
av.GetSymbolWin.RefreshPalette(#PALETTE_LIST_MARKER)
```

FIGURE 2

```
Color.GetBlack
Color.GetRed
Color.GetBlue
Color.GetMagenta
Color.GetCyan
Color.GetWhite
Color.GetGray
Color.GetYellow
Color.GetGreen
```

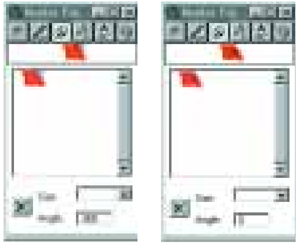

FIGURE 3
Use the font-generating software to select and fine-tune composite markers.

PC ARC/INFO and ArcCAD

Convert ArcInfo Coverages

User Level
New/Casual Advanced

By Arron Heriford, ESRI Technical Support

PC ARC/INFO, ArcCAD, and ArcInfo coverages are not interchangeable in their native formats. The Workspace Translator, available from the ESRI Web site (www.esri.com/software/workspacetranslator/), is a stand-alone executable that translates a PC ARC/INFO or ArcCAD workspace into an ArcInfo 7.x workspace or vice versa.

This utility runs on Windows 95/98 or Windows NT. This download includes two versions of the program—a command line version (wkspcvt.exe) and a GUI version (wksptran.exe). Both versions are 32-bit programs. The GUI version is recommended, as it offers more functionality.

Install the Workspace Translator. The following steps describe how to use the GUI version of the program.

1. Run wksptran.exe from Explorer or Run. This brings up the Workspace Translator dialog box. In the section Input Workspace, enter the path name of the workspace to translate by typing it in the text box, or use the Browse button to locate it.
2. While still in the Input Workspace section, select the specific coverages to convert from the Coverage & Data files scroll box.
3. Use the Settings button to define specific details for the conversion. This dialog box can be used to direct the translator to skip coverages with edit mask files (MSK files) or coverages that have records marked for deletion, specify both the ASCII file format and the ArcInfo 7.x annotation subclass name, and create a report log.
4. In the Output Workspace area, enter the name and location for the new workspace.
5. When all the parameters have been set, click the Convert button.

Atlas GIS 4

Getting Block Group Data into a Geocoded Point Table

User Level
New/Casual Advanced

By Linda Schmidt, ESRI Technical Support

Follow these steps to add block group census data from the 1998 Data & Maps CD–ROM to an existing geocoded point table.

1. Copy the desired files containing block group data to a folder. Volumes 1, 3, and 4 of the Data & Maps 1998 CD–ROM contain block group data. For example, to use the block group files from Volume 1 for the State of Alabama, copy all the files named alblkgrp.xxx from the AL folder.
2. Open the shapefile (alblkgrp.shp in this example) in Atlas GIS. When the file is converted to a geo (.agf) file, it will be automatically linked to its corresponding table (.dbf) file.
3. Open the table and locate the column titled BKG_KEY. This column holds the block code data that will be put into the point table.
4. Open the point table to which you want to add block code data. Make sure the table has latitude and longitude columns that contain coordinates. Define an empty column, BLOCKCODE, for holding the block code data.
5. From the menu, choose Table > Assign Data By Location.
6. In the Assign Data By Location dialog box under the Set Assign Results To section, choose the target point table.
7. In the same dialog box under the From Containing Or Nearest Feature In section select the linked table that contains the block code data.
8. On the right side of the same dialog in the Columns To Assign section, set the Fill Column to the column that was defined to hold the block code data. This column should be empty.
9. Still in the Columns To Assign section, click on the selection button under With Expression and choose the BKG_KEY column in the table for the geo file that was created from the shapefile.
10. Click OK to complete the process.

The point table should now contain block codes that correspond to the point data.

ArcView Internet Map Server

Adding Identify Functionality for Grids

User Level
New/Casual Advanced

By Rex Hansen, ESRI Technical Support

ArcView Internet Map Server does not support an Identify function on grid themes in served views. The script shown in Figure 4 will add this functionality.

1. Start ArcView GIS. Choose File > Extensions and verify that the Internet Map Server extension is loaded.
2. In the Project window, double click on the Script icon to open a new script window.

FIGURE 4

```
‘— This code returns the cell value
‘— of a grid in a served view.

‘— bolFlag is used to verify something was found
bolFlag = False
if ((thmActive.HasAttributes) or (thmactive.getclass.getclassname =
“GTheme”)) then
  if (thmActive.GetClass.GetClassName = “FTheme”) then
    ftbThe = thmActive.GetFTab
    lstFields = ftbThe.getFields
```

Continued on page 62

ArcView Internet Map Server

Continued from page 61

3. Choose Script > Load System Script and select AVINETMP.id. This script performs an identify on active themes in a view.
4. Note the first and last commented sections in the script fragment in Figure 4. Locate those sections in the AVINETMP.id system script. Replace the portion between those sections with the script fragment in Figure 4.
5. Save the script as AVINETMP.id.
6. Compile the script.
7. Make sure the grid theme is active in the served view. Serve the view.
8. Note the value of the identified cell in the information frame to the right of the map.

Disclaimer

The user assumes all responsibility for use of the sample routines as well as implementation of them to achieve the intended results. The user is responsible for fully testing each portion of the routine prior to relying on it. This information is offered as a sample only, and ESRI assumes no obligation for its operation, use, or any resultant effect in spite of this offer. This information and these sample routines are provided on an “as is” basis, without warranty of any kind.

Please remember to back up your data prior to using this information.

```
    ‘– If the results position is not below MapCafe
    ‘– then the results will be in vertical columns.
    if (INP.GetResultPosition <> #POSITION_PAGEBOTTOM) then
      for each fld in lstFields
        if(fld.isVisible) then
          bolFlag = True
          theVals = “<tr><td><font size=2><B>” ++ fld.GetAlias ++
“</B></font></td>”
        ‘theVals = “<tr><td NOWRAP><font size=2><B>” ++ fld.GetAlias ++
“</B></font></td>”
          for each x in lstAttribs
            jj= ftbThe.ReturnValueString(fld,x)
            if (jj = “”) then
              jj = “<I>No Data</I>”
            end
           ‘– Remove double quotes found in data
           jj = jj.Substitute(“”””,”/”””””)
           theVals = theVals ++”<td><font size=2>”++ jj ++ “</font></td>”
           ‘theVals = theVals ++”<td NOWRAP><font size=2>”++ jj ++
“</font></td>”
           end
         theVals = theVals ++ “</tr>”
         wlink.writeString(theVals++CR+NL)
       end
     end
    ‘– Otherwise put results horizontally
    else
      theVals = “<tr>”
      for each fld in lstFields
        if(fld.isVisible) then
          bolFlag = True
          theVals = theVals +”<td><font size=2><B>” ++ fld.GetAlias ++
“</B></font></td>”
        end
      end
      theVals = theVals ++ “</tr>”
      wlink.writeString(theVals)
      theVals = “<tr>”
      for each x in lstAttribs
        for each fld in lstFields
          if (fld.IsVisible) then
            jj= ftbThe.ReturnValueString(fld,x)
            if (jj = “”) then
              jj = “<I>No Data</I>”
            end
            jj = jj.Substitute(“”””,”/”””””)
            theVals = theVals ++”<td><font size=2>”++ jj ++ “</font></td>”
          end
        end
        theVals = theVals ++ “</tr>”
      end
      wlink.writeString(theVals++CR+NL)
    end

    ‘– Display grid cell values
    elseif (thmActive.GetClass.GetClassName = “GTheme”) then
      agrid = thmactive.getgrid
      p = point.make(xid,yid)
      vprj = viwtheview.getprojection
      val = agrid.cellvalue(p,vprj)
      thevals = nil
      thetext = val.asstring
    ‘– If the results position is not below MapCafe
    ‘– then the results will be in vertical columns.
    if (INP.GetResultPosition <> #POSITION_PAGEBOTTOM) then
      bolFlag = True
      theVals = “<tr><td><font size=2><B> Value </B></font></td>”
      ‘theVals = “<tr><td NOWRAP><font size=2><B> Value </B></font></td>”
      theVals = theVals ++”<td><font size=2>”++ thetext ++ “</font></td>”
      ‘theVals = theVals ++”<td NOWRAP><font size=2>”++ thetext ++
“</font></td>”
      theVals = theVals ++ “</tr>”
      wlink.writeString(theVals++CR+NL)
   ‘– Otherwise put results horizontally
   else
```