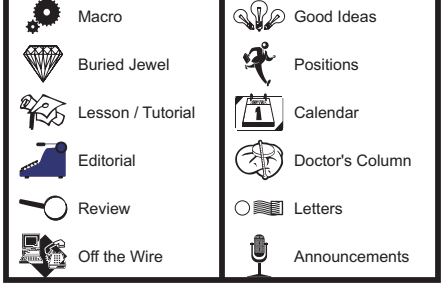
Point Line Poly

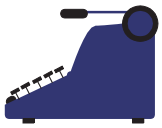
A Technical Friend for PC ARC/INFO® Users and Other ESRI Desktop Software

Volume **7** Number **1**

Inside:

Icon Key





De-evolving GIS

Editor

GIS for Everyone?

Jack Dangermond, at the 1998 User Conference, expounded an uplifting vision of integrating GIS with everyday life. Nonetheless, while the idea of putting a map on a cell phone is dandy, I am upset by the rapidly spreading practice of shielding GIS users from an understanding of the work being done.

As is Duane F. Marble, who in the Spring 1998 ARC News (V20N1) stated: "As GIS technology has become generally available, as well as more user-friendly, an erroneous notion has developed that these changes mean that the technology can be mastered by almost anyone with minimal effort."

No matter how "simple" the software, performing GIS analysis without a technical knowledge of geography is an invitation to the production of bad or inappropriate data.

GIS for Everything?

An interesting note was struck at the Conference by keynote speaker Karl Steinitz, who pointed out the following limitations of GIS as an analytic tool:

1) The chief problems regarding environmental management are not quantitative, but cultural.

2) GIS can support a decision, but not automate it.

3) Without the support of a body of experience, the virtual reality represented by a model remain just that—virtual.

The Death of Science?

Which brings to mind a chilling anecdote (I heard it secondhand, and am unable to confirm its accuracy): a technician for an agency in the Midwest related that he was using GIS to analyze the correlation between iron content and vegetation growth in rivers. When asked what species of vegetation were involved, he replied that he didn't know. And when asked if field biologists were going to survey the vegetation types, he replied that there were no more field biologists, just the GIS.

Whether or not the story is true, the caveat remains: GIS is a component of scientific analysis, and not the science itself.

PLP

What's New?

A Review of the PC Technical Workshops at the 1998 User Conference

[I was hoping to be able to go into greater depth with the help of the Technical Workshop presentations, but not all of them made it to the CD and—despite what was repeatedly said at the User Conference—they aren't available on the Web site. Oh, well...]

PC ARC/INFO and DAK

What's New: 3.5.2

By the time this article is in print, the free 3.5.2 upgrade should be shipping. As well as fixing some bugs, a number of new features have been added.

GUI and SML

- Graphic display cursor coordinate tracking, map or page units
- New windows look, closer to Win 95
- International character support
- Additions to Windows extensions
- &IF: double quotes force string comparison
  `&goto it_works! &if &ne "1" "1.0"`

Data Conversion and Product Compatibility

- SHAPEARC: accepts but skips measure or Z values
- AGFSHAPE/SHAPEAGF: supports version Atlas GIS 4.0 files
- DXFARC: supports R14 entities (except regions and 3D solids)

Miscellaneous

- Improved NT setup
- DRAWE/BACKE NODE ALL added to ARCEDIT
- DUMP added to TABLESW
- LINE option added to CLEAN
- BUILD requires POLY option for polygons
- MAPENV.SML (see **V6N6**) added to UTOOL

The list isn't complete, but after 3.5.2 ships there will be a follow-up article.

POINT LINE POLY is published by Pierssen Publishing, 3125 West Wilson Drive, Flagstaff, AZ USA 86001. Domestic subscriptions: 6 issues $35. Foreign: 6 issues $50. Single Issues $7 domestic, $10 foreign. Send subscriptions and fulfillment questions to *PLP* Circulation Dept., 3125 West Wilson Drive, Flagstaff, AZ USA 86001. Or call (520) 774-7905. Submit material and technical questions to The Editor, at the address above.

© Copyright 1998, Pierssen Publishing. All rights reserved. *Point Line Poly* is an independently produced publication of Pierssen Publishing.

ISSN: 1099-2324

ESRI, ARC/INFO, PC ARC/INFO, ArcView, and ArcCAD are registered trademarks; Atlas GIS, Data Automation Kit (DAK), Simple Macro Language (SML), and Avenue are trademarks of Environmental Systems Research Institute, Inc. All other company and product names mentioned are property of their respective owners.

POSTMASTER: Send address changes to:
Point Line Poly
Pierssen Publishing
3125 West Wilson Drive
Flagstaff, AZ USA 86001
e-mail: piersen@primenet.com

NOTICE: Due to differences in hardware and software configurations, and differing user requirements, information published in *PLP* may or may not be applicable to specific installations and user requirements. To ensure the accuracy of the information published in *PLP*, Pierssen Publishing specifically disclaims responsibility for errors and omissions or the ability of users to implement recommendations published in *PLP*.

Say, what? "There was a young fellow named Hector, who was fond of a launcher-erector. But the squishes and pops of acute pressure drops wrecked Hector's hydraulic connector." **Thomas Pynchon**, *Gravity's Rainbow* (one of the cleaner ones)

Note: this is the **last** DOS-compatible version to be released. Future releases will be produced for the 32-bit Windows environment only, although 3.5.2 will also be included on the CD for DOS users.

What's Coming: 4.0

Work has already begun on 4.0, but the features to be implemented are far from finalized. Some of the new features include:

- True 32-bit Windows executables
- Image backgrounds
- Wintab support for digitizers (the old drivers will also be kept)
- ArcView symbol sets (the centering problem will be fixed—see p. 6)

Lots of SML enhancements are also being implemented:

- More local variables
- Increased variable size
- Named local variables (without compilation)
- No compiler needed for &IF ... &ELSEIF ... &ELSE ... &END
- New logical operators for &IF: &LT, &LE, &GT, &GE, &AND, &OR (&XOR ?)
- &IFNOT directive for &GOTO and &GOBACK

Will SDTS be supported? As far as I know this hasn't been decided yet. If you're having problems with Sol Katz's SDTS2DLG utility you might want to push for it. Also, vote for your favorite image formats! User feedback is provided at the PC ARC/INFO and DAK Discussion Forum:

http://nt1.esri.com/community/community.cfm

Atlas GIS

What's New: 4.0

Version 4.0 is now shipping (the pre-release version was reviewed extensively in **V6N5**). Some of its new and outstanding features include:

- Atlas GIS is a 32-bit OLE server application!
- Working document is WYSIWIG
- ODBC/SQL support
- Support for standard desktop formats
- Seagate Crystal Reports included
- Nationwide geocoding from 1 CD (current to 12/97)
- Ranged fill and proportional fill
- AGF files can have multiple feature classes
- Powerful projection/datum transformation engine

Also, the new 4 CD set of ESRI Maps and Data ("darn near 9 gigs worth of data") will be packaged in 4.0, and will be made available to existing 4.0 users for basic cost ($49).

What's Coming: OCX, 4.1

An ActiveX control (OCX) for Atlas GIS (4.0a) will be available in a few months for Visual Basic/C++ programmers. Version 4.1 is in the works; hopefully further information will be available soon.

ArcCAD

What's New: 14

Version 14 is principally a compatibility release. Some new features include:

- Shape import/export
- PC A/I's projection utility
- Image support dropped (now handled by R14)

The marketing emphasis for ArcCAD has shifted: it is now viewed as an add-on utility to edit GIS data in AutoCAD. Also, the price has dropped significantly: at an incredible $495, no other ESRI package I know of has more geoprocessing functionality for the buck.

What's Coming: ???

Now that: 1) AutoDesk is no longer an ESRI partner, and 2) Visual Basic for Applications (VBA) allows one to bring MapObjects and ARC/INFO 8 into AutoCAD, ArcCAD's future is uncertain. As with ESRI's other PC products, it remains up to the users to determine the future direction.

Year 2000 Resource Center

ESRI has published a site to answer questions about Y2K, compliance standards, and how various software packages stand up. The site is at:

http://www.esri.com/software/y2000/index.html

Although testing remains to be completed for many packages, all ESRI PC software should be compliant or compliant "with minor issues". The basic issue is that while dBASE files store years in 4 digit format, unless the four digits are explicitly given by the user, the software will pull the century from the system clock (as is the case with PC A/I).

AGF/Shape Converters

Two utilities to convert between shapefiles and Atlas GIS .AGF files are available for free download at:

http://www.esri.com/software/atlas/converters.html

GIS Job Site!

Here's a valuable site that could really take off in the near future:

http://www.gisjobs.com/

PLP Online

http://www.primenet.com/~piersen/PLP

User Name: plpv7n1

Password: nz8bks7j

PLP

(Continued from Page 3)

ArcView

What's New: 3.1

ArcView 3.1 was made available for sale at the User Conference. Some of the new features include:

- New front ends for some extensions, including labeling tools and wizards for geoprocessing, custom legends, and grids and graticules
- Seagate Crystal Reports included
- More View and TOC customization
- 3D shapes (measures or Z values)
- Support for more formats: R14, TIFF 6, MrSID
- Customizable image support
- New shape and graphic classes: Ellipse, GeoCurve, SplineText
- New symbol classes: VectorPenMarker, VectorPenVertexMarker, CompositeArcInfoPen
- More symbols

Although 3.1 fixes some bugs (but not all of them), unfortunately it introduces some new ones. The nastiest bug from my point of view is its inability to render vector fills with a pen size less than 1.0! A patch may be issued, but as of writing this article one hasn't come out yet. If and when it comes out, it should be posted at the patch page:

http://www.esri.com/software/arcview/avsoftware.html

What's Coming: Projection Utility, 4.0

As was stated in **V6N6**, ArcView does not intrinsically support datum conversion. However, a standalone utility is in the works (based, I hear, on the Atlas GIS engine!!!) to convert between standard projections and datums.

I found out very little at the User Conference about future directions of ArcView. I have heard rumors that, as is being done with ARC/INFO 8, ArcView 4.0 will be re-engineered to the Component Object Model (COM) standard and packaged with VBA. Undoubtedly the next User Conference will provide more details.

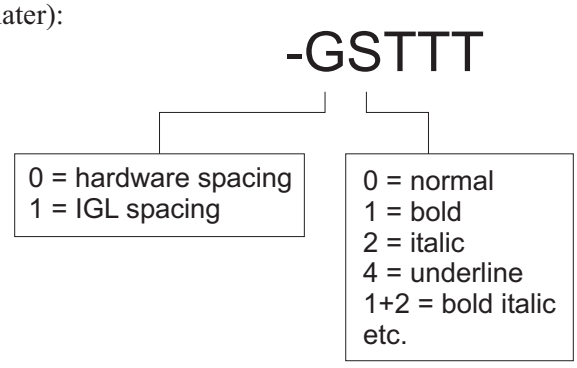
PLP

Windows Fonts, Colors, and Patterns

This article is long overdue, because PC ARC/INFO has had the ability to use Windows fonts, colors, and patterns since version 3.4.2; the goal is to introduce new and experienced users to their convenience, attractiveness, and power.

TrueType Fonts

Windows (TrueType) fonts may be used in text sets, marker sets, line sets, and any Windows extension commands that refer to font numbers, such as TEXTFONT. TrueType font numbers are distinguished from IGL font numbers by having a negative value. The rightmost three digits (000-999) of the number refer to the font family, the next digit (10^3) refers to the style, and the next digit (10^4) refers to the character gap method (more on that later):



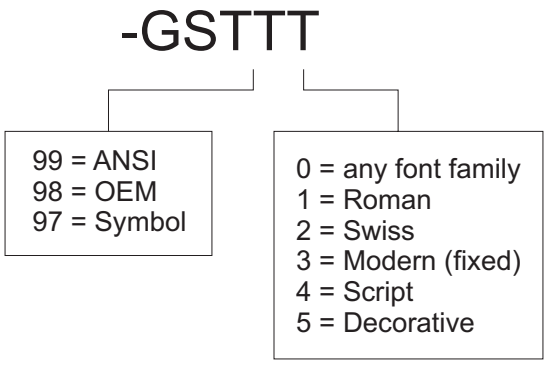
Note that style numbers are additive; thus if Arial were the second font in the TrueType list, and the styles were bold and underline (standard spacing), the font number would be -5002. To determine the font number for a particular font and style, the "Fonts..." entry may be picked from the graphic display's "Options" menu. The font dialog lists the available fonts and styles, and displays the corresponding font number:

The dialog may also be invoked by entering "WIN EXEC 9" at the command prompt. Because the resulting font number is returned to variable 1, it may be used to set the TEXTFONT at the ArcplotW prompt or in an SML application[1]:

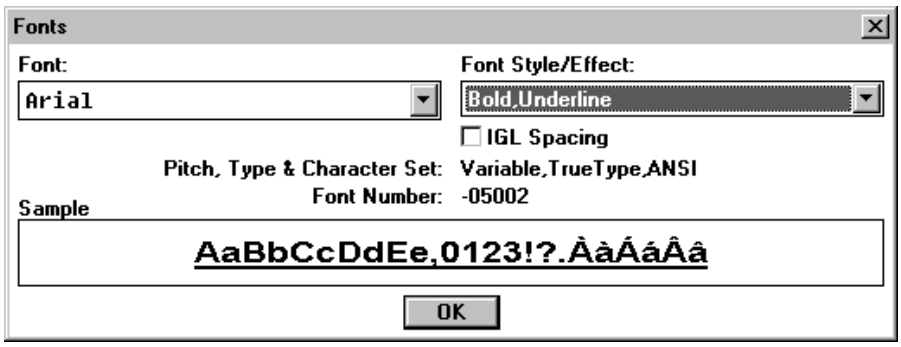
```
WIN EXEC 9
TEXTFONT %1
```

*[**Warning:** Font numbers vary with the Windows font installation. If you add or remove a font from your computer, you probably will need to revise accordingly any PC A/I symbol sets and SML scripts that use Windows fonts.]*

Because font numbering varies with font installation, PC A/I supports several generic font families for which Windows will try to find the closest match. The rightmost digit (0-9) refers to the generic font family and the next two digits (97-99) refer to font classification (ANSI,OEM,Symbol):



Two symbol sets utilizing TrueType fonts are provided as of version 3.5: TRUETYPE.TXT and TRUETYPE.MRK. Additional information may be found in the on-line help under "Windows Fonts".

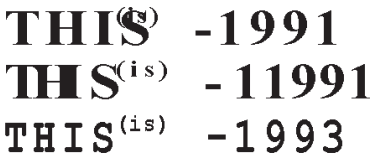


IGL Spacing

The IGL spacing option forces constant width characters for proportional fonts:

THIS is NORMAL spacing
THIS is IGL spacing

As you can see, this isn't too pretty. You're actually better off just using a fixed font (e.g. Modern), which is also better at handling IGL typesetting commands:



Unfortunately, fixed pitch fonts (other than variants of Courier) are hard to find. I'll put up a few useful links on the PLP OnLine version of this article.

Using Windows Fonts as Markers

TRUETYPE.MRK is similar in structure to other marker sets, save that the FONT and PATTERN items have been expanded to accommodate extra characters:

```
ITEM NAME      WIDTH    TYPE    N.DEC
SYMBOL             3       N        0
COLOR              3       N        0
PATTERN           11       N        0
FONT              11       N        0
SIZE               8       N        6
```

Assuming that font -972 defaults to Wingdings, symbol 108:

```
COLOR      PATTERN         FONT      SIZE
   1           108          972 0.126000
```

will plot as a solid black circle. The Windows accessory "Character Map" is useful for examining the glyphs available in a font; unfortunately, it only gives the keystroke and not the pattern number. The following SML will return the pattern number (33-127) of a given keystroke (space-Control+Backspace):

patnum.sml

```
&rem Return ASCII number of character
&rem (33-127 except 34)

&sv -2 " ! #$%&'()*+,-./01234567"
&sv -3 "89:;<=>?@ABCDEFGHIJKLMNO"
&sv -4 "PQRSTUVWXYZ[\]^_`abcdefg"
&sv -5 "hijklmnopqrstuvwxyz{|}~ "
&cv -9 lpos "%-1" "%-2"
&goto false001 &if &eq %-9 0
   &type "Pattern = %<%-9 + 31>
   &return
&label false001
&cv -9 lpos "%-1" "%-3"
&goto false002 &if &eq %-9 0
   &type "Pattern = %<%-9 + 55>
   &return
&label false002
&cv -9 lpos "%-1" "%-4"
&goto false003 &if &eq %-9 0
   &type "Pattern = %<%-9 + 79>
   &return
&label false003
&cv -9 lpos "%-1" "%-5"
&goto false004 &if &eq %-9 0
   &type "Pattern = %<%-9 + 103>
   &return
&label false004
&type "Usage:  &r PATNUM [character]"
&type "        Space ( ) = 32"
&type "        Double Quote ("") = 34"
&return
```

In the above code, " " stands for character 127 (Control+Backspace). Running the SML without an argument will return the code for a space and a double quote:

```
Usage:  &r PATNUM [character]
        Space ( ) = 32
        Double Quote (") = 34
```

Note that above utility is not needed for patterns greater than 127, which are represented by the keystrokes Alt+0nnn, where nnn = the pattern number. Similarly, you can use Alt+0nnn to find the glyph corresponding to a certain pattern number.

One current problem with using TrueType fonts as marker symbols is that the glyphs will not center exactly[2]:

Here, symbol 177 of TRUETYPE.MRK is compared to symbol 2 of PLOTTER.MRK. The difference may be sufficiently small to be acceptable for many applications, but use of TrueType markers is not advised for high accuracy cartography.

TrueType markers may also be used in line symbols. For example, the following entries:

```
LAYER TYPE COLOR OFFSET  ODIA  IDIA
INTERVAL PATTERN          OPTION
1     7    1     0.000  0.100 0.100
0.100    041               74
2     1    1     0.000  0.025 0.000
0.050    73                 0
```

will create the following line symbol:

Of course, TRUETYPE.MRK needs to be specified before drawing the line.

One final note: although Windows fonts generally draw faster than IGL fonts, each one that is used by PC A/I takes up system resources. Thus, a using large number of different fonts in a plot may slow down graphic execution.

Windows Colors

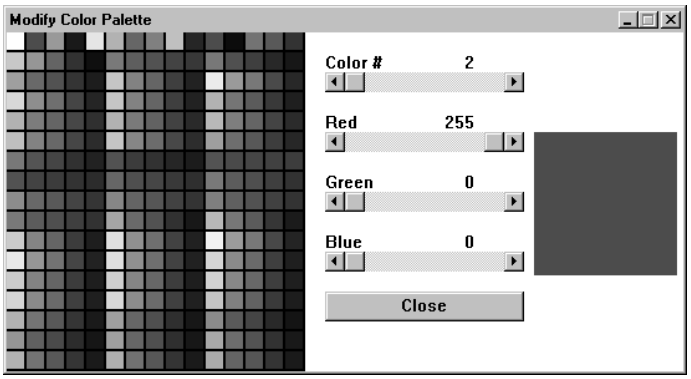
PC A/I supports 255 colors; symbol sets COLOR255.LIN, COLOR255.MRK, and COLOR255.SHD take advantage of them. By default, the colors are defined by the file SCRWIN.COL in SYMBOLS directory, which contains RGB values for each color number. For example, color 31 (orange) has the following entry:

```
31 255 127   0
```

In this example, the value 255 corresponds to 100% red and 127 to 50% green. SCRWIN.COL may be edited to alter the color definitions, or new color files created. (Note, however, that colors 1-15 are system colors, and should not be edited; color 1 cannot be changed at all.) For example, a quick way to lighten up the colors a bit for inkjet plotters would be to change all 0 values in colors 16-255 to 64.

COLORS.CML, included in the UTOOL directory, is a handy utility for experimenting with new color definitions. Run it in ArcplotW:

```
display 4
@colors
```



To assign color definitions from a different color file, use the WIN COLOR command:

```
win color r c:\arcexe\symbols\scrwin34.col
```

Note that color definitions are reset to SCRWIN.COL every time DISPLAY 4 is issued.

Windows Patterns

PC ARC/INFO supports 100 patterns; the symbolset PATTERNS.SHD uses all 100 patterns. Patterns 1-7 are hardware patterns (used in HARDWARE.SHD) compatible with pen plotters. Patterns 8-100 are bitmap patterns, based on 8x8 stipples (same as Windows desktop patterns or ArcView's stipple-based raster fills), and are defined by default using the file SCRWIN.PAT. The advantage of using bitmap patterns in shadesets is that they draw very rapidly; the disadvantage is that they are opaque, and always draw with a white background.

SCRWIN.PAT stores a pattern definition as hex string where each hex character corresponds to a 4-bit pixel pattern:

0 = 0000
1 = 0001
2 = 0010
etc.

Where "0" is a foreground pixel (color) and "1" is a background pixel (white). (The Windows calculator accessory has a scientific view which is convenient for converting between hex and binary values.) For example, pattern 15:

```
 15   1D 0E 17 BB 71 E0 D1 BB
Diagonal thatches
```

translates to the following bit pattern:

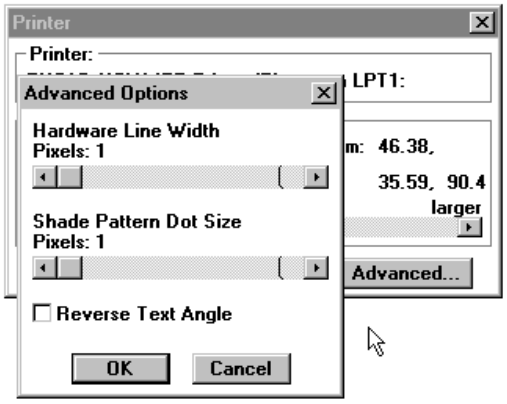
PAT.CML, shown here, may be run in ArcplotW to create a new pattern and calculate the corresponding hex string.

As with colors, other pattern files may be specified using the WIN PATTERN command:

```
win pattern r c:\arcexe\symbols\scrwin34.pat
```

Note also that patterns 8-34 in SCRWIN.PAT match the patterns available in DOS graphic displays.

When printing shades using bitmap patterns, you will probably want to adjust the shade pattern dot size. The Print dialog has an "Advanced..." button which brings up the following:



A pixel size of 4 works well with 600 dpi printers.

Colors and Patterns in Shade Sets

Colors and patterns may be combined in a shadeset by setting TYPE to 1 and OPTION to an appropriate value. Positive values of OPTION will combine a color number and a hardware pattern (1-7) as follows:

shade code number = (color number - 1) * 7 + pattern number

In other words, take the color number, subtract 1, multiply by 7, and add the pattern number. Negative values of OPTION will combine a color number and any pattern number as follows:

negative shade code number = INT ((color number - 1) / 100) * 10000 + (pattern number - 1) * 100 + MOD (color number - 1,100) + 1

A simpler rule of thumb is as follows:

For colors 1 to 100:

001 to 100 = colors 1 to 100, pattern 1
101 to 200 = colors 1 to 100, pattern 2
etc.

For colors 101 to 200, use the same number scheme as for the first 100 colors but add 10000:

10001 to 10100 = colors 101 to 200, pattern 1
10101 to 10200 = colors 101 to 200, pattern 2
etc.

For colors 201 to 255, use the same scheme as above but add 20000:

20001 to 20055 = colors 201 to 255, pattern 1
20101 to 20155 = colors 201 to 255, pattern 2
etc.

Next: Altering Table Structures

[1]Version 3.5.2's TEXTFONT * command automatically invokes the font dialog and applies the returned number. Note also that the font number is not returned if the dialog is launched from the "Options|Fonts..." menu.
[2]This problem will be fixed in Version 4.

PLP

SHAPEARC and Overlapping Polygons

Because ArcView's shapefile format, being nontopological, allows overlapping polygons, Host ARC/INFO's SHAPEARC command will either bring in polygons as arcs (assigning attributes to the AAT) or will create a region class which then must be reprocessed using the REGIONPOLY command if attributes are to be reassigned to the polygon class.

PC A/I's SHAPEARC command is a mixed blessing because, on one hand, undesired region classes are never created; on the other hand, overlapping polygons are handled ungracefully. Figure 1 depicts a shapefile with overlapping polygons, and Figure 2 shows the resulting coverage. Perhaps the nastiest situation involves polygons coded "D" and "E", which overlap to such an extent that two label points end up in one polygon.

Currently there is no workaround for this in PC ARC/INFO (unless, perhaps, Version 4 adds a LINE option to the SHAPEARC command). There is, however, a way to "condition" a shapefile in ArcView so that it will process correctly in SHAPEARC. The script "regpoly.ave" emulates to a certain degree the REGIONPOLY command: it outputs a polygon theme (representing a self-intersection of the input theme) and a table containing attributes; a unique id field allows a one-to-many relationship of the output theme to the output table (Figure 3). Note that, unlike REGIONPOLY, the output polygons are unsorted.

RegPoly.ave

```
theView = av.GetActiveDoc
theFTab =
theView.GetActiveThemes.Get(0).GetFTab
if (theFTab.GetShapeClass.GetClassName
<> "Polygon") then
   MsgBox.Error("Not polygon
FTheme","RegionPoly")
   return nil
end
shpname =
FileName.GetCWD.MakeTMP("Theme","shp")
shpname =
FileDialog.Put(shpname,"*.shp","Output
Shapefile")
if (shpname = nil) then
   return nil
end
tabname =
shpname.ReturnDir.MakeTMP("Pat","dbf")
tabname =
FileDialog.Put(tabname,"*.dbf","Output
Attribute List")
if (tabname = nil) then
   return nil
end
idname = MsgBox.Input("Name of relate
field","RegionPoly","poly_id")
if (idname = nil) then
   return nil
end
if (theFTab.FindField(idname) <> nil)
then
```

Figure 1

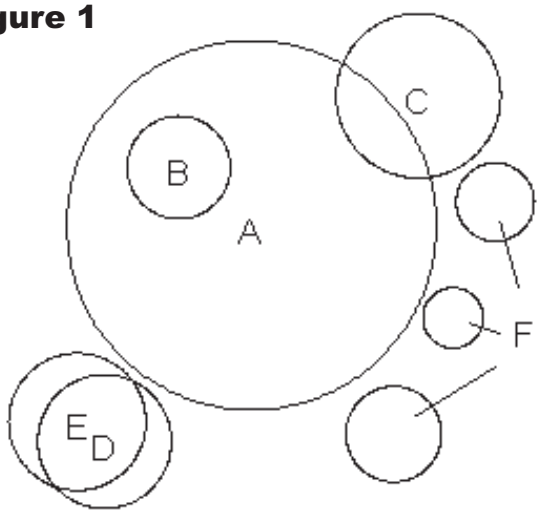
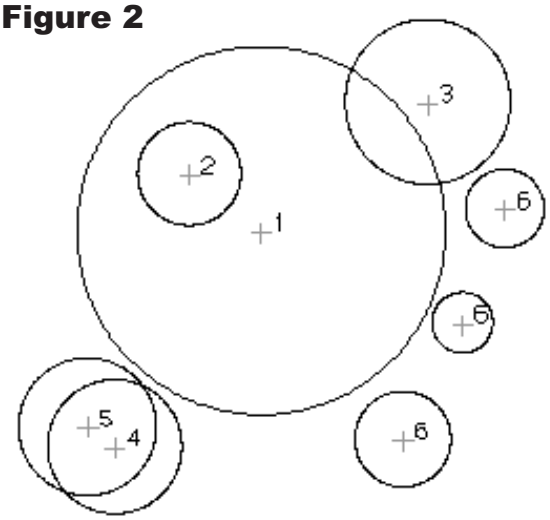
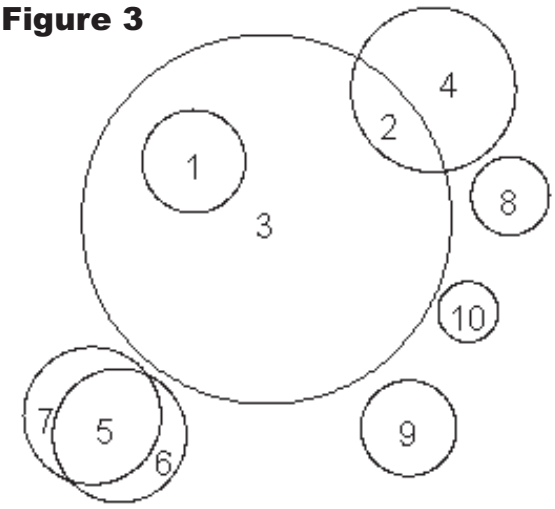


Figure 2



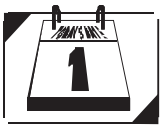
```
   MsgBox.Error("Field
Exists.","RegionPoly")
   return nil
end

'**** create and define table
structures

PShape = FTab.MakeNew(shpname,
Polygon)
idf =
Field.Make(idname,#FIELD_DECIMAL,11,0)
PShape.AddFields({idf.Clone})
PTab = VTab.MakeNew(tabname,dbase)
PTab_flist = List.Make
PTab_flist.Add(idf.Clone)
for each f in theFTab.GetFields
   if (f.IsTypeShape.Not) then
      PTab_flist.Add(f.Clone)
   end
end
PTab.AddFields(PTab_flist)
sf = theFTab.FindField("shape")
psf = PShape.FindField("shape")
pif = PShape.FindField(idname)
tif = PTab.FindField(idname)

'**** get list of shapes to self-
intersect

plist = List.Make
for each r in theFTab

plist.Merge(theFTab.ReturnValue(sf,r).
Explode)
end

'**** generate list of intersected
shapes

av.ShowMsg("Performing self-
intersection...")
av.ClearStatus
rec = 1
numrec = plist.Count
i_list = {plist.Get(0)}
for each i in 1..(plist.Count - 1)
   p = plist.Get(i)
   chopp = p.Clone
   j_list = List.Make
   for each s in i_list
      p1 = s.ReturnIntersection(p)
      if (p1.IsNull.Not) then
         j_list.Merge(p1.Explode)
         p2 = s.ReturnDifference(p1)
         if (p2.IsNull.Not) then
            j_list.Merge(p2.Explode)
         end
         chopp =
chopp.ReturnDifference(p1)
      else
         j_list.Add(s)
      end
   end
   if (chopp.IsNull.Not) then
      j_list.Merge(chopp.Explode)
   end
   i_list = j_list
   rec = rec + 1
   av.SetStatus(rec / numrec * 100)
end
plist = nil
```

Figure 3



Relate table:

```
POLY_ID CODE
      1 A
      1 B
      2 A
      2 C
      3 A
      4 C
      5 D
      5 E
      6 D
      7 E
      8 F
      9 F
     10 F
```



CA/HI/NV/Guam Regional Conference

The CA/HI/NV/Guam ESRI Regional Users' Group will host their annual conference on January 27-29, 1999, at the San Bernadino Hilton in San Bernadino, CA, USA. Contact Chuck Johnson, US Bureau of Reclamation, (916) 978-5290 or e-mail: cbjohnson@mp.usbr.gov for more information. Additional information is also available at their web site:
http://www.mp.usbr.gov/mp400/cahinv/cahinvpg.html

(Also, information and links to other Regional Group events is maintained at ESRI's web site: **http://www.esri.com/usersupport/usergroups/ugconf.html**)

PLP

ERRATUM:

In **V6N4** ("ArcCAD and Object Data") the language of the article implied that R14's Object Data was a redefinition of the older Extended Entity Data (EED) model. It has since been pointed out that the two technologies are totally separate, though both are supported in R14.

(Continued from page 10)

```
'**** add record(s) as appropriate

av.ShowMsg("Building records...")
av.ClearStatus
rec = 1
numrec = i_list.Count
for each s1 in i_list
   r1 = PShape.AddRecord
   PShape.SetValue(psf,r1,s1)
   PShape.SetValue(pif,r1,rec)

theFTab.SelectByShapes({s1},#VTAB_SELT
YPE_NEW)
   for each rr in theFTab.GetSelection
      s2 = theFTab.ReturnValue(sf,rr)
      if (s2.Contains(s1).Not) then
         continue
      end
      r2 = PTab.AddRecord
      for each ff in theFTab.GetFields
         if (ff.IsTypeShape.Not) then
            val =
theFTab.ReturnValue(ff,rr)
            f =
PTab.FindField(ff.GetName)
            PTab.SetValue(f,r2,val)
         end
      end
      PTab.SetValue(tif,r2,rec)
   end
   rec = rec + 1
   av.SetStatus(rec / numrec * 100)
end
av.ClearMsg
av.ClearStatus

theBitmap = theFTab.GetSelection
theBitmap.ClearAll
theFTab.UpdateSelection
i_list = nil
PShape.Flush
PTab.Flush
av.PurgeObjects

'**** ask if add to View

if (MsgBox.YesNo("Add shapefile as
theme to the view?",
    "Convert to Shapefile",true)) then
   fthm = FTheme.Make(PShape)
   theView.AddTheme(fthm)
   theView.GetWin.Activate
end
```

PLP

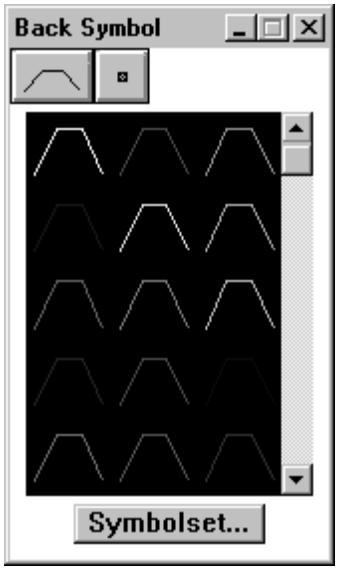
COORD vs. UBUT

SML PROGRAMMING

To conclude this series on GUITOOL, I'd like to discuss two different approaches to image mapping in a dialog box. In the last article (**V6N6**), dialog box E_SETBC.DLG and routine SETBACKC were added to the EDTOOLS application to allow setting background coverages. Widget 104 is a PBUT intended to launch a graphical symbol picker; to keep things simple we'll launch it modally:

```
*PICK 104
   *OPEN getsym I %103
   &rv -2
   &openw [winfile]
   *W S 103 %-2
   &closew
```

Routine GETSYM launches the symbol picker and returns the symbol number to widget 103 (the back symbol number EBOX). My original intention was to design the symbol picker to allow the user to browse line, marker, and text symbols. Unfortunately, ArceditW has no TEXT command, and the only way I could pull it off was to make ArcplotW calls—because that approach slowed things down too much, I dropped text browsing altogether. The COORD version of the symbol picker dialog (E_SYM.DLG) is as follows:




```
BEGIN    0 10.00 40.00 16.00 20.00 v
Back Symbol
UBUT  -201  0.00  0.00  1.75  5.25 b x
UBUT  -202  0.00  5.25  1.75  3.5  b x
PBUT  -204 14.25  4.00  1.25 12.00 v
Symbolset...
BRECT  205  2.00  1.00 12.00 16.00 v
COORD -206
VSBAR -207  2.00 17.00 12.00  2.00 v
END
R 207 1 255 3 15
```

Because UBUT widgets are invisible when no file specification is given, phony "x" values are supplied to keep them visible while playing around in EDITDLG; their actual icons will be created in the routine via the WIN PLOT command and assigned via the S dialog command[1]. Note also that the BRECT widget has an ID to allow assigning an image. The R command sets the properties of the VSBAR widget; in this case, the full range of values is 1-255, the small step increment is 3 (a row of symbols), and the large step increment is 15 (an entire pane).

The COORD widget stores a coordinate when a portion of the dialog (not corresponding to another return widget) is clicked. The coordinate value is a six digit number with the left three digits in eighth-row units and the right three digits in quarter-column units.

Let's look at routine GETSYM and see how it handles things:

```
*ROUTINE getsym
e_sym.dlg 1

&define sym -11 &var
&define pane -12 &var
&value [sym] -1
&sv [pane] LIN
&openw [winfile]
&r genicon LIN [sym]
&r genicon MRK [sym]
&r genpane [pane] [sym]
&closew
*OPEN e_sym.dlg
*PICK 201
   &rem **** switch pane to LIN
   &sv [pane] LIN
   &openw [winfile]
   &r genpane [pane]
   &closew
*PICK 202
   &rem **** switch pane to MRK
   &sv [pane] MRK
   &openw [winfile]
   &r genpane [pane]
   &closew
```

```
*PICK 204
   &rem **** choose symbolset
   &if &cn [pane] LIN &do
      &sv -1 Lineset
   &else
      &sv -1 Markerset
   &end
   &value -2 ARC
   WIN PATH %-2\symbols
   WIN FILE I [pane] 'Choose %-1'
   &if &ne "x[winrtn]" "x" &do
      &extract -2 [winrtn] 1
      &sv -3 .
      &value -2 -3 1 1 %<lpos \ %-2>
%<lpos \ %-2>
      %-1 %-2
      &openw [winfile]
      &r genicon [pane] [sym]
      &r genpane [pane]
      &closew
   &end
*PICK 206
   &rem **** set [sym] according to
COORD
   &cv -1 %206 / 1000 int - 15
   &cv -2 %206 mod 1000 - 3
   &if &rn %-1 1 96 &and &rn %-2 1 64
&do
      &cv -3 ( %-1 * 5 / 96 ) int
      &cv -4 ( %-2 * 3 / 64 ) int
      &cv [sym] %-3 * 3 + %-4 + %207
      &openw [winfile]
      &r genicon LIN [sym]
      &r genicon MRK [sym]
      &closew
   &end
*PICK 207
   &rem **** readjust pane according
to slider
   &openw [winfile]
   &r genpane [pane]
   &closew
*ENDPICK
&value -1 WKSP
& DEL %-1*.dis
&return [sym]
```

Remember that the top left corner of the BRECT is at row 2 column 1 (coordinate 16,4), and is 12 rows by 16 columns in size (96x64). Thus when the return value is 206, the first 3 digits of the stored coordinate (minus 15) go to variable -1 and the last 3 digits (minus 3) go to variable -2; those values are checked against the coordinate range of the BRECT to see if they fall inside. The symbol number is calculated by dividing the region into a 5x3 grid.

Routine GENICON generates an icon file for the line or marker UBUT and writes the appropriate S command to the dialog command file:

```
*ROUTINE genicon

&define type -11 &var
&define sym -12 &var
&define file -13 &var
&define wid -14 &var
&define wksp -18 &var
&extract [type] -1 1
&value [sym] -2
&value [wksp] WKSP
&sv [file] [wksp]t$[type].dis
WIN PLOT [file]
&if &eq [type] LIN &do
   LINEINDEX 0 0 0.67 0.33 [sym] [sym]
1
   &sv [wid] 201
&else
   MARKINDEX 0 0 0.33 0.33 [sym] [sym]
1
   &sv [wid] 202
&end
WIN PLOT
*W S [wid] [file]
&return
```

Routine GENPANE generates the symbol pane:

```
*ROUTINE genpane

&define pane -11 &var
&define sym -12 &var
&define wksp -18 &var
&value [pane] -1
&if &eq "x%-2" "x" &do
   &value -2 207
&end
&cv [sym] %-2 - ( %-2 mod 3 ) + 1 min
241
&value [wksp] WKSP
WIN PLOT [wksp]t$sym.dis
&if &eq [pane] LIN &do
   LINEINDEX 0 0 1.33 2 [sym] %<[sym]
+ 14> 3
&else
   MARKINDEX 0 0 1.33 2 [sym] %<[sym]
+ 14> 3
&end
WIN PLOT
*W S 205 [wksp]t$sym.dis
*W S 207 [sym]
&return
```

If a starting value isn't passed to the routine, it reads the value of the VSBAR. The symbol value is then constrained so that it mod 3 always equals 1 and doesn't exceed 241; this keeps the arrangement of the symbols even. Note also that the VSBAR is readjusted to that value.

Conceivably, one could develop even more complex image maps. A really ambitious person could create a lookup grid by mapping a coordinate to a character value in a string (e.g. position 1,1 maps to character 1 of string 1).

Because it generally takes come thoughtful coding to convert COORD values to meaningful data, it can be simpler to "cheat" by placing invisible UBUT widgets beneath the image. The size that I chose for the BRECT turned out to be unfortunate, because it didn't divide neatly into 5x3. After a bit of tweaking, however, I came up with the following:

```
BEGIN    0 10.00 40.00 16.00 20.00 v
Back Symbol
UBUT  -201  0.00  0.00  1.75  5.25 b x
UBUT  -202  0.00  5.25  1.75  3.5  b x
PBUT  -204 14.25  4.00  1.25 12.00 v
Symbolset...
VSBAR -207  2.00 17.00 12.00  2.00 v
UBUT  -211  2.00  1.00  2.50  5.25 b
UBUT  -212  2.00  6.25  2.50  5.50 b
UBUT  -213  2.00 11.75  2.50  5.25 b
UBUT  -214  4.50  1.00  2.25  5.25 b
UBUT  -215  4.50  6.25  2.25  5.50 b
UBUT  -216  4.50 11.75  2.25  5.25 b
UBUT  -217  6.75  1.00  2.50  5.25 b
UBUT  -218  6.75  6.25  2.50  5.50 b
UBUT  -219  6.75 11.75  2.50  5.25 b
UBUT  -220  9.25  1.00  2.25  5.25 b
UBUT  -221  9.25  6.25  2.25  5.50 b
UBUT  -222  9.25 11.75  2.25  5.25 b
UBUT  -223 11.50  1.00  2.50  5.25 b
UBUT  -224 11.50  6.25  2.50  5.50 b
UBUT  -225 11.50 11.75  2.50  5.25 b
BRECT  205  2.00  1.00 12.00 16.00 v
END
R 207 1 255 3 15
```

While I was developing the UBUT widgets in EDITDLG, I placed them after the BRECT and assigned phony "x" values to keep them visible. After saving the dialog file, I edited it to rearrange the widgets and strip the "x" values.

Now we can drop the *PICK 206 code from routine GETSYM and add the following:

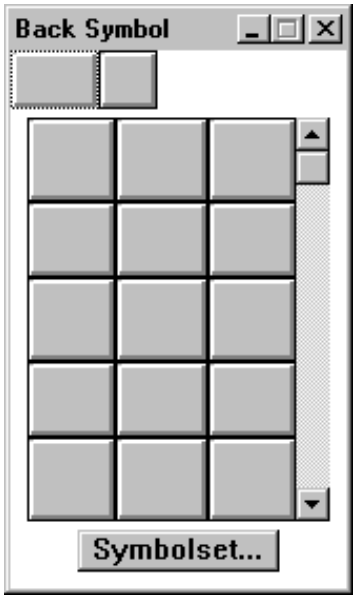
```
*PICK 211,225
   &rem **** set [sym] according to
UBUT
   &cv [sym] [winrtn] - 211 + %207
   &openw [winfile]
   &r genicon LIN [sym]
   &r genicon MRK [sym]
   &closew
```

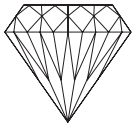
One may argue that developing appropriate UBUT widgets involves at least as much work as interpreting a single COORD value, and allows less flexibility in image mapping. True, I personally prefer COORD, but I encourage you to play with both approaches in your application, and develop your own preference.

Next: Faking Cursors

[1]To be perfectly honest, GETSYM isn't a very practical routine, but it is fun to click around and watch the buttons change!

PLP





PC ARC 3.5.x and Windows NT

On some computers, for some reason, PC ARC/INFO has stability problems in Windows NT 4.0: every now and then it will wink out of existence just as it's about to execute a command. The following steps will help to improve PC A/I's stability (version 3.5.2, if it detects NT, will automatically implement them on installation).

First, edit CONFIG.NT (in the WINNT\SYSTEM32 directory) to remove the REM from the following statement:

```
REM    NTCMDPROMPT
```

Also REM out or delete the following line if present:

```
SHELL=%SystemRoot%\SYSTEM32\COMMAND.CO
M /P /E:4096
```

Finally, replace %ARC%\CMD\ARCW.BAT with the following code:

```
@ECHO OFF
if a%1 == a goto error
if %1 == 0  goto okay
if %1 == 1  goto okay
if %1 == 2  goto okay
if %1 == 3  goto okay
if %1 == 4  goto okay
if %1 == 5  goto okay
if %1 == 6  goto okay
if %1 == 7  goto okay
if %1 == 8  goto okay
if %1 == 9  goto okay
goto error
:okay
set wksp=
set arcw=%1
SHIFT
SHIFT
IF A%OS% == AWindows_NT goto NT
ARC.BAT %0 %1 %2 %3 %4 %5 %6 %7 %8 %9
goto done
:NT
CMD /C ARC.BAT %0 %1 %2 %3 %4 %5 %6 %7
%8 %9
goto done
:error
echo echo This batch file may only be
launched by the SCRWIN program!
echo If you want to run ARCW, please
launch it by clicking on
echo the windows icon for it.
pause
:done
```

Note that this may not completely remove the stability problem, but it should significantly improve it (see also the on-line help under "Frequently Asked Questions"). This should not be a problem at all with version 4.

RENAMCOV

If RENAMCOV is giving you problems:

```
Cover XXX
has one or more Read-Only files that
can not be deleted.
Please change the attributes of these
files and delete them.
```

even though your coverage has no read-only attributes and all files were deleted, try editing RENAMCOV.BAT in the %ARC%\PTOOLS directory, replacing the following lines:

```
RMALL %1
IF EXIST %1\*.* GOTO READONLY
RMDIR %1
```

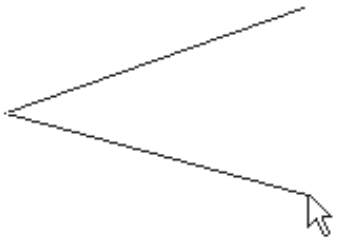
with:

```
DELETE %1 -NQ
IF EXIST %1 GOTO READONLY
```

PLP

Retrieving Angles

Routine GETANGLE allows the user to specify an angle using a three-point line (the following example returned -34.497393):



getangle.r

```
&routine getangle

&rem **** return angle based on user
3-point line

&define x0 -6 &var
&define y0 -7 &var
&define x1 -8 &var
&define y1 -9 &var
&define x2 -10 &var
&define y2 -11 &var
&define dx1 -12 &var
&define dy1 -13 &var
&define a1 -14 &var
&define d1 -15 &var
&define dx2 -16 &var
&define dy2 -17 &var
&define d2 -18 &var
&define a2 -19 &var
&define angle -20 &var

&rem **** get user line

&type "Input 3-point line to define
angle (<9> to cancel)"
&getxym -1 -2 -3
&if &eq %-3 9 &do
   &return
&end
&value [x1] -1
&value [y1] -2
&getxym -1 -2 -3 1
&if &eq %-3 9 &do
   &return
&end
&value [x0] -1
&value [y0] -2
&getxym -1 -2 -3 1
&if &eq %-3 9 &do
   &return
&end
&value [x2] -1
&value [y2] -2

&rem **** calculate angle

&cv [dx1]  [x1] - [x0]
&cv [dy1]  [y1] - [y0]
&cv [d1] ( ( [dx1] * [dx1] ) + ( [dy1]
* [dy1] ) ) ** 0.5
&cv [a1]  ( [dx1] / [d1] ) acos
&if &ne %<[dy1] abs> [dy1] &do
   &cv [a1] 0 - [a1]
&end
&cv [dx2]  [x2] - [x0]
&cv [dy2]  [y2] - [y0]
&cv [d2] ( ( [dx2] * [dx2] ) + ( [dy2]
* [dy2] ) ) ** 0.5
&cv [a2]  ( [dx2] / [d2] ) acos
&if &ne %<[dy2] abs> [dy2] &do
   &cv [a2]  0 - [a2]
&end

&cv [angle] [a2] - [a1]
&type "Angle = [angle]"

&return [angle]
```

Use COMPSML GETANGLE N to create getangle.sml. The following SML uses GETANGLE to rotate features in Arcedit:

```
SHOW SETANGLE -11
&r getangle
&rv -12
&goto end &if &eq "x%-12" "x"
SETANGLE %-12
ROTATE
SETANGLE %-11
&label end
```

PLP