

Point Line Poly

A Technical Friend for PC ARC/INFO® Users
plus Data Automation Kit • ArcCAD® • Atlas GIS

Volume 6 Number 6

This Issue:



ArcView and
Spheroids: the Real
Story

1



The Low-down on
Label Errors

2



Save the
Environment!

5



Scratch Names and
Scratch Items

6



Atlas 4.0 Patch /
PLP OnLine

8



Project Files
(Continued)

9

Volume 6 Index

12



ArcView and Spheroids: the Real Story

Editor

In **V6N5** it was erroneously reported that ArcView assumes WGS84 as the spheroid for lat/long data. The truth is, when projecting/unprojecting data, *ArcView assumes that lat/long data use the same spheroid as the projection.* Thus, if your lat/long shapefiles are based on the WGS84 spheroid and you set your View's projection to TM/Clarke 1866 (e.g. to bring in a DRG image), your data will not match up!

Although a datum conversion sample extension (a port of the Molodensky transformation) is included with ArcView, the fact that lat/long is not handled in a consistent manner has frightful implications for creating and publishing data. It's also a strong argument to avoid lat/long and only work with projected data.

PC ARC/INFO and DAK

Once again, PC ARC/INFO and DAK (Data Automation Kit) prove themselves to be essential components of a robust yet low-budget GIS.

Both packages contain a full implementation of Host ARC/INFO's data projection facility. This means that you can readily convert between nearly any projection and spheroid. NADCON, the Federal standard

for NAD27- NAD83 conversion, is also supported.

Both packages also contain tools for the rapid development and maintenance of **topologically correct** data. For example, instead of digitizing adjoining polygons and then trying to match up the edges, you can simply digitize the common boundaries once and then automatically build the polygons.

Overlay Analysis

Anyone who has attempted true overlay analysis (identity, union, clip, dissolve, etc.) in ArcView knows just how poorly it works. PC ARC/INFO, however, contains a complete suite of overlay functions with a computational efficiency that blows ArcView right out of the water.

If you're an ArcView user and don't have PC ARC/INFO or DAK, drop by the product demo area at the User Conference and find out for yourself just what you're missing!

PLP

Icon Key



Macro



Buried Jewel



Lesson / Tutorial



Editorial



Review



Off the Wire



Good Ideas



Positions



Calendar



Doctor's Column



Letters



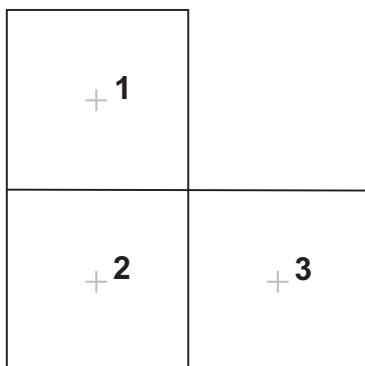
Announcements

1998 User Conference Issue



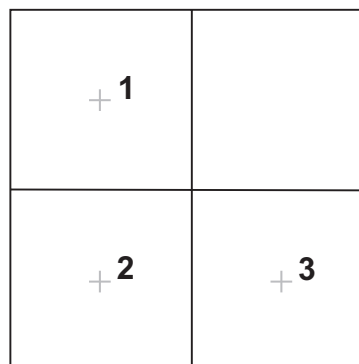
The Low-down on Label Errors

A routine aspect of data creation and maintenance is dealing with label errors in a polygon coverage. For example, given coverage TEST:



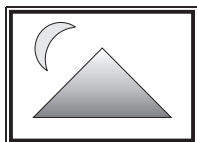
TEST_	TEST_ID	CODE
1	0	
2	1	A
3	2	B
4	3	C

the goal is to add another polygon:



One conventional procedure for adding new polygons to a coverage is as follows:

- 1) Add the necessary arcs in ARCEDIT(W)
- 2) CLEAN the coverage
- 3) Use NODEERRORS [cover] DANGLE to identify dangles
- 4) Fix any dangles in ARCEDIT(W) and CLEAN or BUILD as necessary



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? "Being female, the woman had not survived the flight."

Michael Avallone

Beneath the Planet of the Apes

- 5) Find the ID base for CREATELABELS (i.e., if the highest user ID in the coverage is 3, the smallest ID base is 4)
- 6) Use CREATELABELS [cover] [id_base] to generate labels with unique user IDs
- 7) Re-BUILD the coverage

Why is an ID base used in this case? In Host ARC/INFO, using CREATELABELS without specifying an ID base doesn't cause any problems! The reason, of course, is that PC ARC/INFO uses the user ID to join attributes after a BUILD, whereas Host ARC/INFO uses the internal ID.

When you run CREATELABELS on the coverage without a specified ID base, new label points will be created for every polygon, having the following IDs:

+ 1	+ 2
+ 3	+ 4

but the PAT still has the following IDs:

+ 1	+ 0
+ 2	+ 3

Thus if you attempt to edit the labels in ARCEDIT you get the following message:

WARNING User-IDs between LAB & PAT do not agree.

Exit and re-BUILD the coverage.

If you were to re-BUILD the coverage, however, the user IDs of the labels would be used to join the attributes, resulting in a mess! Whereas the polygons originally had the following attributes:

+ A	
+ B	+ C

they're now coded as follows:

+ A	+ B
+ C	+

The solution is not to re-BUILD the coverage but to recalculate the user IDs of the PAT in TABLES:

```
TABLES
SEL TEST.PAT
CALC TEST_ID = TEST_ - 1
Q
```

Now the PAT user IDs match the LAB user IDs; IDEDIT isn't even necessary!

Multiple Labels

When a coverage has been CLEANed with more than one label point in a polygon, the LAB and PAT will not agree. The LAB IDs look like this:

+ 1	+ 5 + 4
+ 2	+ 3

whereas the PAT IDs look like this:

+ 1	+ 4 + 4
+ 2	+ 3

Whatever attributes may have been assigned to label point 5 are already gone; only point ID 4 has a corresponding record in the PAT. Thus, running CREATELABELS without an ID base and then recalculating the PAT IDs in TABLES will effectively eliminate the extra label point in polygon 2; the user IDs of polygons 2, 3, and 4 will change, but the attributes will remain correct:

+ A	+ D
+ B	+ C

The Universe Polygon

The presence of one or more label points in polygon 1 (the universe polygon) is usually an undesired result of a particular operation (e.g. deleting a polygon's arcs but not the label point). Should one or more label points exist in the universe polygon, running CREATELABELS without an ID base and then recalculating the PAT IDs in TABLES will eliminate the label point(s) but preserve the attributes in record 1 of the PAT.

It is desirable at this stage to erase any universe polygon attributes. Otherwise, should more polygons be added later on, record 1 attributes will be assigned to those polygons lacking label points. Overlay operations may also be affected.

A Routine to Fix Label Errors

Given the above knowledge, a routine can be created that will automatically fix all label errors in a coverage that has been CLEANed or BUILDED, and erase any attributes in record 1:

CL.R:

```
&routine cl

&define cover 51 &var

&if &eq "x%-1" "x" &do
    &delim < >
    &type "Usage:  &r CL [cover]"
    &delim [ ]
    &return
&end
&if &nf % -1 \pal &do
    &type "%-1 does not have polygon
topology."
    &return
&end
&if &fn % -1 \msk &do
    &type "%-1 must be CLEANed or
BUILDED."
    &return
&end

&value [cover] -1
CREATELABELS [cover]
TABLES cl_tab
&return

&routine cl_tab

&define wksp -19 &var
&define cover 51 &var

&value [wksp] WKSP
SEL [cover].PAT
CALC %[cover]_ID = %[cover]_ - 1
RES $RECNO = 1
&openw [wksp]t$items.lis
ITEMS
&closew
&sv -11 .FALSE.
&open [wksp]t$items.lis error
&while &do
    &read -10 [break]
    &extract -1 -10 1
    &if &ne "x%-1" "x" &and &nm % -1 &do
        &extract -2 -10 2
        &value -5 -2 %< len "%-2" - 2 >
        %< len "%-2">
```



Save the Environment!

Joseph Zastrow

Problem:

You create a map composition and then later try to use it in another session, but you have forgotten what page size and units were used to create the map composition. So the map composition does not come up correctly.

Solution:

Save the page environment information in the map composition as an SML file that you can later restore.

The following SML saves the page environment information in the specified map composition in the file "env.plt". Note the file must start with an "e" and end with ".plt" or the map composition will become invalid. To restore the page environment do the following:

```
@[map_name]\env.plt
```

Possible enhancements: you could also add mapext, mappos, mapunits etc if needed.

MAPENV.SML

```
&goto usage &if &eq " " "%-1"
&openw %-1\env.plt i 3
show pagesize -11 -12
&write "pagesize %-11 %-12" 3
show pageunits -11
&write "pageunits %-11" 3
&closew 3
&return
&label usage
&type "Usage: &r mapenv [map_name]"
&type " Write the current page
environment to the map composition"
&type " [map_name]. [map_name] must
already exist. You can then use:"
&type " @[map_name]\env.plt"
&type " or"
&type " &r [map_name]\env.plt"
&type " to restore the page
environment."
&return
```

Joseph Zastrow is ESRI's one-man programming team for PC ARC/INFO, DAK, and Atlas GIS.

PLP

(Continued from page 4)

```
&value -6 -2 %< len "%-2" - 1 >
%< len "%-2">
&if &eq "%-5" "_ID" &or &eq "%-
6" "_I" &do
    &sv -11 .TRUE.
&elseif &eq %-11 .TRUE. &do
    &extract -3 -10 4
    &if &eq %-3 N &or &eq %-3 D
&do
    CALC %-2 = 0
    &elseif &eq %-3 C &do
    MOVE ' ' TO %-2
    &end
&end
&end
&close
& DEL [wksp]t$items.lis
ASEL
Q
```

```
&return
&label error
&type "Error opening
[wksp]t$items.lis"
Q
&return
```

Use COMPSML CL N to generate CL.SML and CL_TAB.SML.

One final note: because this routine regenerates label points for all polygons, you will not want to use it on a coverage containing label points which you have placed at specific coordinates (e.g. for LABELTEXT purposes) and which you don't want relocated.

Next: TrueType Fonts

PLP



Scratch Names and Scratch Items

Routine SCRATCHN emulates the AML [SCRATCHNAME] function:

```
&r SCRATCHN {-PREFIX [prefix]}
  {-SUFFIX [suffix]} {-FILE |
  -DIRECTORY | -INFO} {-WKSP}
```

arguments

{-PREFIX [prefix]} - specifies a prefix. Will be truncated if more than 5 characters long. The default is XX.

{-SUFFIX [suffix]} - specifies a suffix. Will be truncated if more than 3 characters long. The default is blank.

{-FILE | -DIRECTORY | -INFO} - type of object for which the name is generated.

{-WKSP} - specifies that the full pathname to the %WKSP% directory will be included.

notes

- The arguments may be given in any order. The return value may be retrieved using the **&rv** directive. For example:


```
&r scratchn -directory -prefix t_
&rv -19
```
- The returned value is PREFIXn.SUFFIX (7.3 format), where n is a numeric string; if the -WKSP option is included, the returned value also includes the full pathname to the %WKSP% directory.
- The -INFO option sets SUFFIX to DBF for internal purposes but strips it from the return value.
- The -DIRECTORY option sets SUFFIX to blank. If the first character of PREFIX is 0-9, "A" will be substituted. Otherwise, the routine does not check for illegal coverage characters.
- If the intended object is not subsequently created, the next time SCRATCHN is run it will return the same value.

```
&routine scratchn
```

```
&rem **** Emulates the AML SCRATCHNAME
function
```

```
&define scratch -1 &var
&define i -9 &var
&define p -11 &var
&define s -12 &var
&define type -13 &var
&define inwksp -14 &var
&define wksp -18 &var
&define temp -19 &var
```

```
&sv [i] -1
&sv [p] XX
&sv [s]
&sv [type] FILE
&sv [inwksp] .FALSE.
&value [wksp] WKSP
&while &rn [i] -6 -1 &do
  &extract [temp] %[i] 1
  &if &eq "x[temp]" "x" &do
    &break
  &elseif &eq -PREFIX [temp] &do
    &dec [i]
    &extract [p] %[i] 1
    &value [p] [p] 1 5
  &elseif &eq -SUFFIX [temp] &do
    &dec [i]
    &extract [s] %[i] 1
    &value [s] [s] 1 3
  &elseif &eq -FILE [temp] &do
    &sv [type] FILE
  &elseif &eq -INFO [temp] &do
    &sv [type] INFO
  &elseif &eq -DIRECTORY [temp] &do
    &sv [type] DIRECTORY
  &elseif &eq -WKSP [temp] &do
    &sv [inwksp] .TRUE.
  &else
    &delim < >
    &type "Usage: &r SCRATCHN {-
PREFIX [prefix]} {-SUFFIX [suffix]}"
    &type "                  {-
FILE/-DIRECTORY/-INFO} {-WKSP}"
    &delim [ ]
    &return ERROR
  &end
  &dec [i]
&end
&if &eq [type] INFO &do
```

```

    &sv [s] DBF
&elseif &eq [type] DIRECTORY &do
    &value [temp] [p] 1 1
    &if &cn A0123456789 [temp] &do
        &sv [temp] A
        &value [p] [temp] 1 1 1 1
    &end
    &sv [s]
&end
&sv [i] 0
&while &do
    &sv [scratch] 0000000
    &value [scratch] [p] 1 %<len [p]> 1
    %<len [p]>
    &value [scratch] [i] 1 %<len [i]>
    %<8 - len [i]> 7
    &if &ne "x[s]" "x" &do
        &sv [scratch] [scratch].[s]
    &end
    &if &eq [inwksp] .TRUE. &do
        &sv [scratch] [wksp][scratch]
    &end
    &if &nf [scratch] &do
        &break
    &end
    &inc [i]
&end
&if &eq [type] INFO &do
    &value [scratch] [scratch] 1 %<fpos
. [scratch] - 1>
&end

&return [scratch]

```

Routine SCRATCHI returns a unique item name for the currently selected table in TABLES. It will also work in ARCEDIT(W).

```
&routine scratchi
```

```
&rem **** generates unique item name
for current table in TABLES
```

```

&define scratch -1 &var
&define i -9 &var
&define j -10 &var
&define p -11 &var
&define numit -12 &var
&define fn -13 &var
&define wksp -18 &var
&define temp -19 &var

```

```

&sv [p] XX
&value [wksp] WKSP
&if &eq "x%-1" "x/?" &do
    &type "Usage:  &r scratchi

```

```

{prefix}"
    &return
&elseif &ne "x%-1" "x" &do
    &extract [p] -1 1
    &value [p] [p] 1 5
&end
&openw [wksp]t$items.lis
ITEMS
&closew
&sv [i] 101
&sv [numit] 0
&open [wksp]t$items.lis error
&while &do
    &read -10 [break]
    &extract -1 -10 1
    &if &ne "x%-1" "x" &and &nm %-1 &do
        &extract -2 -10 2
        &value %[i] -2
        &inc [i]
        &inc [numit]
    &end
&end
&close
& DEL [wksp]t$items.lis
&sv [i] 0
&while &do
    &sv [scratch] 0000000
    &value [scratch] [p] 1 %<len [p]> 1
    %<len [p]>
    &value [scratch] [i] 1 %<len [i]>
    %<8 - len [i]> 7
    &sv [fn] .FALSE.
    &sv [j] 1
    &while &rn [j] 1 [numit] &do
        &value [temp] %<[j] + 100>
        &if &eq [scratch] [temp] &do
            &sv [fn] .TRUE.
            &break
        &end
        &inc [j]
    &end
    &if &eq [fn] .FALSE. &do
        &break
    &end
    &inc [i]
&end
&return [scratch]

&label error
&type "I/O error:  SCRATCHI"
&return

```

Both routines must be compiled with COMPSML.
To display the usage supply "/" as an argument:

```
&r scratchn /?
```




Atlas GIS 4.0 Report Patch

The first cut of the Atlas GIS 4.0 CD was missing some files needed for Crystal Reports support. Affected users should have received a floppy containing the fix, and the CD has been remastered.

The patch is also available for download at:

<http://www.esri.com/base/products/atlas/reportspatch.html>

If you haven't seen it yet, ESRI's new and vastly improved scripts repository is worth a visit. Check it out at:

<http://andes.esri.com/arcscripsts/scripts.cfm>

PLP

PLP Online

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

User Name: plpv6n6

Password: iw0p8vp2

(Continued from page 7)

One TABLES bug present in Version 3.5.1 is that a SORT on mixed field types will not work if the first sort field is numeric. The following SML uses SCRATCHI to create a temporary character item to put in front of the sort list:

SORT.SML

```
&rem **** sort routine for TABLES for
sorting mixed field types
&rem **** when the first sort field is
numeric
```

```
&goto usage &if &eq "x%-1" "x"
&r scratchi
&rv -19
&sv -20
ADDITEM
%-19
1
```

```
C
%-20
%-20
MOVE 'x' to %-19
SORT %-19 %-1 %-2 %-3 %-4 %-5 %-6 %-7
%-8 %-9 %-10 %-11 %-12 %-13 %-14 %-15
%-16 %-17 %-18
DROPITEM %-19
Y
&return

&label usage
&type "Usage:  &r SORT [item...item]"
&return
```

PLP



Project Files (Continued)

In the last article (**V6N5**), I showed you how to implement project files in the EDTTOOLS application, allowing the user to save and load editing sessions. The goal was to store as little information as possible in SML variables, ideally only those parameters not accessible through the SHOW command. Routine SAVEP assumes that edit features are stored in variables 31-34 and back symbols are stored in 35-38, and writes the project file as an SML that when executed restores the session appropriately. Also, the project file name is stored in variable 51 ([project]) and the saved flag is stored in variable 52 ([saved]).

Now we're going to take a look at how that information is maintained when opening and removing edit and back coverages. When opening an edit coverage, the user is automatically asked to set the edit feature, so the only change necessary is to add some lines at the end of routine SETEDITF:

```
*REM **** store edit feature
&sv -2 1
SHOW EDITCOVER 0 -3
&while &rn %-2 1 4 &do
    SHOW EDITCOVER %-2 -4
    &if &eq "x%-4" "x%-3" &do
        &value %<%-2 + 30> -1
        &break
    &end
    &inc -2
&end
&return %-1
```

The four edit cover slots are searched for a match to editcover 0, and the edit feature (variable -1) is stored in the appropriate location. Also, the following line is placed at the end of the "*PICK SETCOV SETFEAT 106" entry in routine ETOOL:

```
&sv [saved] NO
```

In the last article some entries were added to E_MAIN.MNU to add back coverages and remove edit or back coverages; now we'll add some *PICK directives to routine ETOOL to support them:

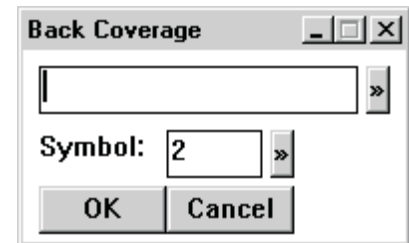
```
*PICK REMCOV
    *OPEN remcov H EDIT
```

```
*PICK BACKCOV
    *OPEN setbackc H
    &sv [saved] NO

*PICK REMBACK
    *OPEN remcov H BACK
```

Let's look at routine SETBACKC first. It uses a new dialog named E_SETBC.DLG:

```
BEGIN      0  0.00  0.00  6.00 24.00 v Back
Coverage
EBOX      101  0.50  1.00  1.50 20.00 v
PBUT     -102  0.50 21.50  1.50  1.50 v »
LTEXT      0  2.50  1.00  1.00  8.00 v
Symbol:
EBOX      103  2.50  9.00  1.50  6.00 v 2
PBUT     -104  2.50 15.50  1.50  1.50 v »
PBUT     -105  4.25  1.00  1.50  8.00 v OK
PBUT     -106  4.25  9.00  1.50  8.00 v
Cancel
```



The code for routine SETBACKC is as follows:

```
*ROUTINE setbackc
e_setbc.dlg 1

*REM **** get back cover

&define i -10 &var
&define pos -11 &var
&sv [i] 1
&sv [pos] 0
&while &rn [i] 1 4 &do
    SHOW BACKCOVER %[i] -2
    &if &eq "x%-2" "x" &do
        &cv [pos] 34 + [i]
        &break
    &end
    &inc [i]
&end
&if &eq [pos] 0 &do
    WIN MB 1 'No more room for back
```

```

coverages'
    &return
&end
&sv -1
&sv -2
*OPEN e_setbc.dlg
*PICK 102
    *FILE 101 C * 'Enter coverage'
*PICK 105
    &sv -1 %101
    &sv -2 %103
    *CLOSE
*PICK 106
    *CLOSE
*ENDPICK
&if &eq "x%-1" "x" &do
    &return
&end
&if &eq "x%-2" "x" &do
    &sv -2 1
&end
BACKC %-1 %-2
&value %[pos] -2
&sv [saved] NO
&return

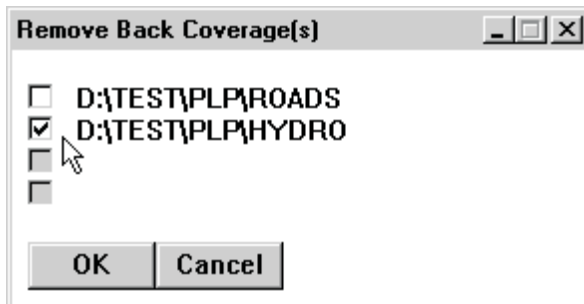
```

The code is almost identical to routine SETEDITC save that it supports an extra widget (103) for the back symbol. (*Support for widget 104, launching the symbol picker, will be discussed in the next article.*) In fact, it would be more efficient to combine SETEDITC and SETBACKC, as is done in routine REMCOV. Routine REMCOV uses E_GETCC.DLG:

```

BEGIN 101 8.00 40.00 8.00 36.00 v
CBOX 102 1.00 1.00 1.00 1.50 v
LTEXT 103 1.00 4.00 1.00 31.00 v
CBOX 104 2.00 1.00 1.00 1.50 v
LTEXT 105 2.00 4.00 1.00 31.00 v
CBOX 106 3.00 1.00 1.00 1.50 v
LTEXT 107 3.00 4.00 1.00 31.00 v
CBOX 108 4.00 1.00 1.00 1.50 v
LTEXT 109 4.00 4.00 1.00 31.00 v
PBUT -110 6.00 1.00 1.50 8.00 v OK
PBUT -111 6.00 9.00 1.50 8.00 v Cancel

```



```

*ROUTINE remcov
e_getcc.dlg 1

*REM **** remove edit or back cover

&define type -8 &var
&define i -9 &var
&value [type] -1
&openw [winfile]
&if &eq [type] EDIT &do
    *W S 101 Remove Edit Coverage(s)
&else
    *W S 101 Remove Back Coverage(s)
&end
&sv [i] 1
&while &rn [i] 1 4 &do
    &if &eq [type] EDIT &do
        SHOW EDITCOV %[i] -1
    &else
        SHOW BACKCOV %[i] -1
    &end
    &if &eq "x%-1" "x" &do
        *W G %<[i] * 2 + 100>
    &else
        &value %<-10 - [i]> -1
        &rem &value -2 -1 %<lpos \ %-1 +
1>
        *W S %<[i] * 2 + 101> %-1
    &end
    &inc [i]
&end
&closew
&sv -1 CANCEL
*OPEN e_getcc.dlg
*PICK 110
    &sv -1 OK
    *CLOSE
*PICK 111 49
    *CLOSE
*ENDPICK
&if &eq %-1 CANCEL &do
    &return
&end
&sv [i] 1
&while &rn [i] 1 4 &do
    &value -1 %<[i] * 2 + 100>
    &if &eq %-1 1 &do
        &value -2 %<-10 - [i]>
        &if &eq [type] EDIT &do
            REMOVEE %-2
            &type " "
            Y
            &sv %<30 + [i]> NONE
        &else
            REMOVEB %-2
            &sv %<34 + [i]> 0

```

```

        &end
    &end
    &inc [i]
&end
&sv [saved] NO
&return

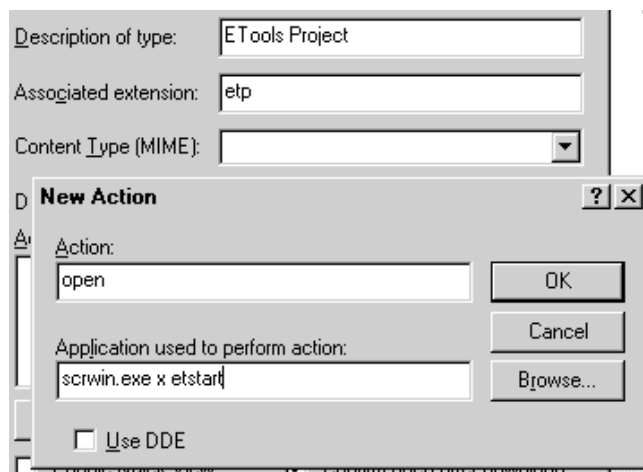
```

First, the routine sets the title of the dialog box. Then it searches through the four slots of edit or back coverages: where one is filled, the appropriate text widget is assigned that value, otherwise the widget is left blank and the checkbox grayed. If a particular coverage is removed, its edit feature or back symbol variable is cleared.

Launching Project Files

Launching a project file from Windows Explorer or File Manager is simply a matter of making a program association:

- 1) Start Windows Explorer
- 2) Click "View | Options..."
- 3) Click "File Types"
- 4) Click "New Type..."
- 5) Under description, type "ETool Project"
- 6) Under extension, type "etp"
- 7) Click "New..."
- 8) Under action, type "open"
- 9) Under application, type "scrwin.exe x etstart"



10) Optional: change the icon to
%ARC%\CMD\ARCEDITW.ICO

ETSTART.SML, which should reside in
%ARC%\UTOOL, contains the following code:

```

&value 51 -1
&sv 52 "ETP"
arceditw etool
QUIT

```

All that's left is to write some code so that ETOOL knows a project is being launched. First, the following line in routine ETOOL:

```
DRAWE ARC LABEL NODE ERRORS TIC IDS
```

is replaced with:

```
&r init
```

Routine INIT has the following code:

```

*ROUTINE init

&sv -1 1
&while &rn %-1 1 4 &do
    &sv %<%-1 + 30> NONE
    &sv %<%-1 + 34> 0
    &inc -1
&end
&if &ne "x[saved]" "xETP" &do
    &sv [project]
&end
&if &ne "x[project]" "x" &do
    &r [project]
&else
    DRAWE ARC LABEL NODE ERRORS TIC IDS
    BACKE ARC
&end
&sv [saved] YES

```

Thus, when a project file is opened in Explorer, its name is passed to ETSTART.SML, which assigns the filename to variable 51 and the "ETP" flag to 52. Routine INIT, after clearing the edit feature and back symbol variables, runs the project file if the "ETP" flag is set.

Next: COORD vs. UBUT

PLP

Volume 6

Chronological Index

Title	Author	Page
Volume 6 Number 1		
The Next Year ArcView and AVX.	Editor	1
Text Editing Buried jewel - editing annotation text.	PLP Staff	2
Handling Coordinates SML programming - reading and implementing coordinate data in SML.	Mark Cederholm	3
Extending Lines Macro (SML) - extending an arc to intersect another.	Mark Cederholm	8
ArcView - Acre Calculations Macro (Avenue) - modification of CALCAPL.AVE to handle acreage values.	Maggie Smith	12
ArcCAD - Adding Parcel Numbers from an AutoCAD Drawing Tricks using DDPEAT.	Mark Scott	15
Volume 6 Number 2		
Your Opinion Counts PC ARC/INFO enhancements depend on user input.	Editor	1
PC ARC/INFO 3.5.1 Enhancements New enhancements, including new features in WEDIT.	PLP Staff	2
Faking Dynamic Segmentation Macro (SML) - generates physical features from a route event table.	Mark Cederholm	3
More Fun with Widgets SML programming - edit boxes, user defined buttons.	Mark Cederholm	8
ArcCAD - Some Help with Utility Mapping Macro (AutoLisp) - create pipe feature between two manholes, populate pipe attributes from manhole.	PLP Staff	12
Miscellaneous Bits Announcements - Atlas GIS, User Conference, credits for acreage macro, user enhancement request form.	PLP Staff	15
Volume 6 Number 3		
Welcome Atlas Atlas GIS now featured in PLP.	Editor	1
Linesymbols with Text Buried jewel - using LINEEDIT to add glyphs to line symbols.	PLP Staff	2
Duplicate Record Search Macro (SML) - temporarily sorts tables to tag duplicate entries.	Mark Cederholm	6
Atlas - Geocoding: Logical Options Taking advantage of Atlas' Relax options.	Steve Lackow/Elio Spinello	9
Projection Tips with DATUM/NADCON Options Using the right datum for input <u>and</u> output.	Eric Edelstein	11
Nonmodal Dialog Boxes SML programming - supporting multiple dialog boxes.	Mark Cederholm	12
ArcCAD - Symbol Sets Using symbol sets to symbolize features.	Mark Scott	17

Volume 6 Number 4

Pardon Our Dust New publisher for PLP, also PLP OnLine.	Editor	1
What's New? New product developments unveiled at 1997 User Conference, plus discussion of year 2000.	PLP Staff	2
ESRI Newsgroup Off the Wire - new internet newsgroup for ESRI users.	PLP Staff	3
Setting Up Digitizers Getting your digitizer to work in PC ARC/INFO.	PLP Staff	4
DXF and Elevation Attributes Buried jewel - supporting elevation in ARCDXF and DXFARC.	Eric Edelstein	6
Sizing Graphic and Text Windows Macro (SML) - repositioning the text window and graphic display to prevent overlap.	PLP Staff	7
Atlas GIS: Graphics Tips and Tricks Pasting page elements, using images.	Steve Lackow	8
GUITOOL SML programming - using GUITOOL to design applications.	PLP Staff	10
ArcCAD and Object Data Using DDPEAT to capture extended entity data.	Adena Schutzberg	14
Mouse Chording Buried jewel - using mouse button combinations to supply key numbers.	PLP Staff	15
Match 'em Up! Macro (SML) - rotate and translate features based on match points.	PLP Staff	16

Volume 6 Number 5

Still Learning Getting the hang of things, new North American rate, upcoming features.	Editor	1
Local Pan and Zoom vs. MAPEXTENT Comparison: what they can and can't do.	PLP Staff	2
NADs Revisited Converting NAD27 data to DD for ArcView.	PLP Staff	6
Little Sister: Pre-release Review of Atlas GIS 4.0 Upsides and downsides of pre-release version.	Steve Lackow	7
GUITOOL and Dialogs SML programming - re-engineering dialogs for GUITOOL support, implementing project files.	PLP Staff	11

Volume 6 Number 6

ArcView and Spheroids: the <u>Real</u> Story The spheroid of lat/long data determines the accuracy of a particular projection in ArcView.	Editor	1
The Low-down on Label Errors An exhaustive discussion of the CREATELABELS command, plus an SML to automate label error correction.	PLP Staff	2
Save the Environment! Macro (SML) - saves the page environment for map compositions.	Joe Zastrow	5
Scratch Names and Scratch Items Macro (SML) - SCRATCHN emulates the AML [SCRATCHNAME] function, SCRATCHI creates a unique item name for the current table in TABLES, plus an example workaround for a SORT bug in TABLES.	PLP Staff	6
Atlas 4.0 Patch Off the Wire - a download link for the Atlas GIS 4.0 report patch.	PLP Staff	8
Project Files (Continued) SML programming - coverage handling tips, launching project files from Windows.	PLP Staff	9

Subject Index

Title	Author	Page
Articles		
ArcView and Spheroids: the <u>Real</u> Story The spheroid of lat/long data determines the accuracy of a particular projection in ArcView.	Editor	V6N6 / p. 1
ESRI Newsgroup Off the Wire - new internet newsgroup for ESRI users.	PLP Staff	V6N4 / p. 3
Miscellaneous Bits Announcements - Atlas GIS, User Conference, credits for acreage macro, user enhancement request form.	PLP Staff	V6N2 / p. 15
The Next Year ArcView and AVX.	Editor	V6N1 / p. 1
Pardon Our Dust New publisher for PLP, also PLP OnLine.	Editor	V6N4 / p. 1
PC ARC/INFO 3.5.1 Enhancements New enhancements, including new features in WEDIT.	PLP Staff	V6N2 / p. 2
Still Learning Getting the hang of things, new North American rate, upcoming features.	Editor	V6N5 / p. 1
Welcome Atlas Atlas GIS now featured in PLP.	Editor	V6N3 / p. 1
What's New? New product developments unveiled at 1997 User Conference, plus discussion of year 2000.	PLP Staff	V6N4 / p. 2
Your Opinion Counts PC ARC/INFO enhancements depend on user input.	Editor	V6N2 / p. 1
ArcCAD		
ArcCAD - Adding Parcel Numbers from an AutoCAD Drawing Tricks using DDPEAT.	Mark Scott	V6N1 / p. 15
ArcCAD and Object Data Using DDPEAT to capture extended entity data.	Adena Schutzberg	V6N4 / p. 14
ArcCAD - Symbol Sets Using symbol sets to symbolize features.	Mark Scott	V6N3 / p. 17
Atlas GIS		
Atlas - Geocoding: Logical Options Taking advantage of Atlas' Relax options.	Steve Lackow/Elio Spinello	V6N3 / p. 9
Atlas GIS: Graphics Tips and Tricks Pasting page elements, using images.	Steve Lackow	V6N4 / p. 8
Atlas 4.0 Patch Off the Wire - a download link for the Atlas GIS 4.0 report patch. See the Review section for additional Atlas GIS coverage.	PLP Staff	V6N6 / p. 8
Buried Jewels - Items shipped with PC ARC/INFO		
DXF and Elevation Attributes Buried jewel - supporting elevation in ARCDXF and DXFARC.	Eric Edelstein	V6N4 / p. 6
Linesymbols with Text Buried jewel - using LINEEDIT to add glyphs to line symbols.	PLP Staff	V6N3 / p. 2
Mouse Chording Buried jewel - using mouse button combinations to supply key numbers.	PLP Staff	V6N4 / p. 15
Text Editing Buried jewel - editing annotation text.	PLP Staff	V6N1 / p. 2

Lessons

Local Pan and Zoom vs. MAPEXTENT Comparison: what they can and can't do.	PLP Staff	V6N5 / p. 2
The Low-down on Label Errors An exhaustive discussion of the CREATELABELS command, plus an SML to automate label error correction.	PLP Staff	V6N6 / p. 2
Setting Up Digitizers Getting your digitizer to work in PC ARC/INFO.	PLP Staff	V6N4 / p. 4

SML Programming

GUI TOOL SML programming - using GUI TOOL to design applications.	PLP Staff	V6N4 / p. 10
GUI TOOL and Dialogs SML programming - re-engineering dialogs for GUI TOOL support, implementing project files.	PLP Staff	V6N5 / p. 11
Handling Coordinates SML programming - reading and implementing coordinate data in SML.	Mark Cederholm	V6N1 / p. 3
More Fun with Widgets SML programming - edit boxes, user defined buttons.	Mark Cederholm	V6N2 / p. 8
Nonmodal Dialog Boxes SML programming - supporting multiple dialog boxes.	Mark Cederholm	V6N3 / p. 12
Project Files (Continued) SML programming - coverage handling tips, launching project files from Windows.	PLP Staff	V6N6 / p. 9

Macros

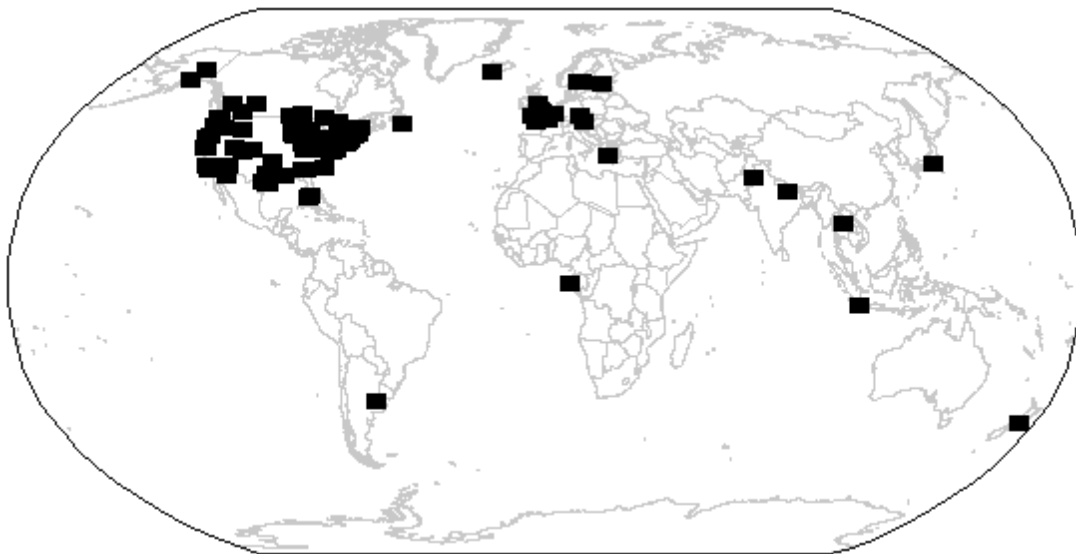
ArcCAD - Some Help with Utility Mapping Macro (AutoLisp) - create pipe feature between two manholes, populate pipe attributes from manhole.	PLP Staff	V6N2 / p. 12
ArcView - Acre Calculations Macro (Avenue) - modification of CALCAPL.AVE to handle acreage values.	Maggie Smith	V6N1 / p. 12
Duplicate Record Search Macro (SML) - temporarily sorts tables to tag duplicate entries.	Mark Cederholm	V6N3 / p. 6
Extending Lines Macro (SML) - extending an arc to intersect another.	Mark Cederholm	V6N1 / p. 8
Faking Dynamic Segmentation Macro (SML) - generates physical features from a route event table.	Mark Cederholm	V6N2 / p. 3
Match 'em Up! Macro (SML) - rotate and translate features based on match points.	PLP Staff	V6N4 / p. 16
Save the Environment! Macro (SML) - saves the page environment for map compositions.	Joe Zastrow	V6N6 / p. 5
Scratch Names and Scratch Items Macro (SML) - SCRATCHN emulates the AML [SCRATCHNAME] function, SCRATCHI creates a unique item name for the current table in TABLES, plus an example workaround for a SORT bug in TABLES.	PLP Staff	V6N6 / p. 6
Sizing Graphic and Text Windows Macro (SML) - repositioning the text window and graphic display to prevent overlap.	PLP Staff	V6N4 / p. 7

Reviews

Little Sister: Pre-release Review of Atlas GIS 4.0 Upsides and downsides of pre-release version.	Steve Lackow	V6N5 / p. 7
--	--------------	-------------

Tips (PC ARC/INFO)

NADs Revisited Converting NAD27 data to DD for ArcView.	PLP Staff	V6N5 / p. 6
Projection Tips with DATUM/NADCON Options Using the right datum for input <u>and</u> output.	Eric Edelstein	V6N3 / p. 11



Generated using PCARC/INFO

Put yourself on the map! Subscribe to Point Line Poly

- ☐ Yes! Please start my subscription to PLP today! Method of payment indicated below.
- ☐ I would like to see an article on the following topic: _____
- ☐ I would like to submit an article on the following topic: _____
- ☐ I have an e-mail address: _____

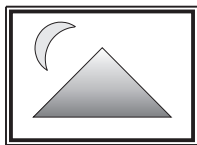
Enter address information below, or attach your business card:

Make payment in U.S. funds (credit card orders are automatically converted). Subscriptions within the United States, Canada, and Mexico are \$35 (6 issues per year); all other countries are \$50.

☐ Master Card Visa Number: _____ Expires: _____

(Please indicate if the card name or billing address is different from that on the mailing label.)

☐ Check or money order enclosed, made out to **Pierssen Publishing**



Pierssen Publishing
3125 West Wilson Drive
Flagstaff, AZ 86001
(520) 774-7905 Voice/Fax

<http://www.primenet.com/~piersen/>
piersen@primenet.com