

## Emulate UNIX on Windows NT/2000

by Stephen Leads, ArcInfo Technical Support Specialist, ESRI–Australia

Users migrating from UNIX versions of ArcInfo to Workstation ArcInfo on Windows NT or Windows 2000 may find that tasks they previously took for granted on UNIX now seem unfamiliar. This article describes some modifications you can make to ArcInfo on Windows NT/2000 that allow “UNIX-like” behavior. These changes apply equally to versions of ArcInfo from 7.2.1 to 8.1.

### Copying and Pasting Text

In UNIX, it is possible to copy and paste text by highlighting the relevant text and using the right mouse button. To allow this behavior on Windows NT/2000, make the following changes.

1. Start Arc in Workstation ArcInfo and right-click on the blue bar at the top of the Arc DOS window to bring up the context menu.
2. Choose Properties from the context menu.
3. In the Arc Properties dialog box, choose the Options tab.
4. Check both the QuickEdit Mode and Insert Mode checkboxes and click on OK.
5. The Apply Properties to Shortcut dialog box will ask if you want to apply these changes only to the current window or to the shortcut that started the program. Choose the second option to have copying and pasting available every time you start Arc.

The Insert Mode option allows text to be inserted on the command line without overwriting the existing text. To copy and paste text, highlight the text to be copied, click the right mouse button to copy the text, move the cursor to a new location, and click the right mouse button again to paste the text.

### Scroll Bars

Scroll bars can also be added to the ArcInfo DOS window. Access the Arc Properties dialog by following Steps 1 through 3 in the previous

section. In the Properties dialog box, choose the Layout tab and increase the Screen Buffer Size Height parameter to 2000. This will cause the scroll bars to appear once the screen has been filled.

### Reissuing a Command

Both the UNIX and Windows NT/2000 versions of ArcInfo allow previously issued commands to be reissued using Shift + 6 (^) and return. On Windows NT/2000, the up and down arrow keys scroll through the list of commands that have previously been issued.

### Using “ls” in Place of “&sys dir”

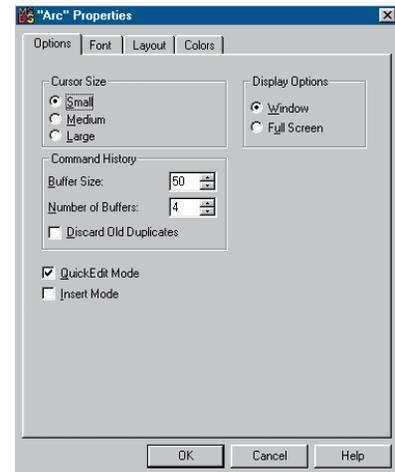
An ARC Macro Language (AML) script that emulates the UNIX ls (list) command within Windows NT/2000 can be downloaded from the ArcScripts page of the ESRI Web site. This functionality can be useful if you frequently switch between operating systems and forget that you must use &sys dir when on Windows NT/2000. This script accommodates most of the ls optional parameters. Visit the ArcScripts page and search on “&sys dir” in the keywords field to locate this script.

### Text Editors

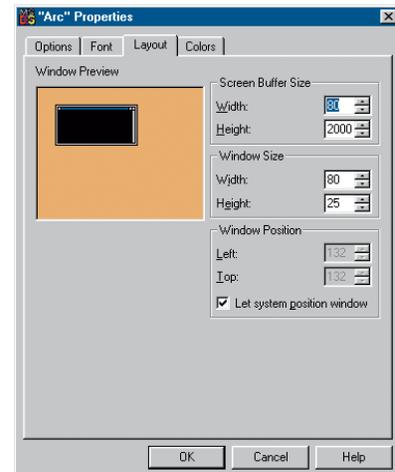
There are many programs available for Windows NT/2000 that emulate vi, the popular UNIX text editor. A 30-day evaluation copy of Lemmy, a good shareware vi emulator, can be downloaded from the Software Online Web site ([www.softwareonline.org](http://www.softwareonline.org)).

For more information on these modifications, please contact Stephen Lead at [slead@esriau.com.au](mailto:slead@esriau.com.au)

*Stephen Lead is a technical support specialist for ArcInfo with ESRI–Australia who previously worked for the Australian Survey and Land Information Group (AUSLIG) as a cartographer.*



*Enabling QuickEdit and Insert Mode will let you copy and paste text by highlighting it and using the right mouse button.*



*Add scroll bars by changing the Screen Buffer Size Height parameter to 2000.*

## PC ARC/INFO

### Make Shorter Work of Commands

By Joe Zastrow, PC ARC/INFO Product Specialist

Using abbreviations for long command names can save time and possibly limit typing errors. Generally speaking, a shortened form of any command can be used if it does not conflict with an existing command in PC ARC/INFO. For example, DRAWENVIRONMENT, the command for specifying which feature classes are displayed for the

edit coverage when the DRAW command is issued, can be abbreviated as DRAW, DRAWEN, DRAWENV, or DRAWENVIR but not as DRAW.

The Vocab subdirectory, located in the Arcexe directory, contains vocabulary files that list PC ARC/INFO commands. The vocabulary files for ARC, ARCEDIT, and

ARCPLLOT can be opened with NotePad and list all the internal commands and any existing abbreviations for them (e.g., EC for EDITCOVER). Alternatively, typing COMMANDS at a command prompt will generate a list of the available commands.

## MapObjects

### Working with Color in MapObjects

Whenever you reference a color in MapObjects, you are really working with a Win32 COLORREF data type. A COLORREF is simply a 32-bit unsigned integer that has the hexadecimal form 0x00bbggrr. The low-order byte contains a value for the relative intensity of red, the second byte contains a value for green, and the third byte contains a value for blue. The high-order byte is usually zero. If the high byte is not zero, Visual Basic uses the system colors, as defined in the user's Control Panel settings and by constants listed in the Visual Basic object library or the MapObjects object library in the Object Browser. The maximum value for a single byte is 0xFF.

MapObjects definitions for some common colors are shown in Figure 1. A list of all 23 defined colors can be found in the MapObjects help under Color Constants. You may also define entirely new colors by specifying different numeric values. The Visual Basic example in Figure 2 uses the Color property to control the color of the symbol used to render a MapLayer. The code and additional discussion can be found in the MapObjects help entry for Color Property. Most development environments have a similar utility routine to construct a COLORREF for you.

To try this example, paste the code into the Declarations section of a form containing a CommonDialog control, a CommandButton named Command1, and a Map named Map1 that contains one MapLayer with polygon features. Press F5 and click Command1. The first time you click the button, you can make a selection from the Color dialog and click OK. Subsequent clicks on the button illustrate different ways to set Color in code.

#### 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.*

Constant	Decimal	Hex	RGB
moBlack	0	0	(0, 0, 0,)
moRed	255	FF	(255, 0, 0)
moGreen	65280	FF00	(0, 255, 0)
moBlue	16711680	FF0000	(0, 0, 255)
moMagenta	16711935	FF00FF	(255, 0, 255)
moCyan	16776960	FFFF00	(0, 255, 255)
moWhite	16777215	FFFFFF	(255, 255, 255)
moLightGray	12632256	C0C0C0	(192, 192, 192)
moDarkGray	4210752	404040	(64, 64, 64)
moGray	8421504	808080	(128, 128, 128)
moPaleYellow	13697023	D0FFFF	(255, 255, 208)
moLightYellow	8454143	80FFFF	(255, 255, 128)
moYellow	65535	FFFF	(255, 255, 0)
moLime Green	12639424	C0DCC0	(192, 220, 192)
moTeal	8421440	808040	(64, 128, 128)
moDarkGreen	32768	8000	(0, 128, 0)

Figure 1

#### Option Explicit

```
Private Sub Command1_Click()
    Dim fRed As Double, fGreen As Double, fBlue As Double
    Static lNumClicks As Long

    With Map1.Layers(0).Symbol
        Select Case lNumClicks
            Case 0
                CommonDialog1.ShowColor
                .Color = CommonDialog1.Color
            Case 1
                fRed = 255 * Rnd
                fGreen = 255 * Rnd
                fBlue = 255 * Rnd
                .Color = RGB(fRed, fGreen, fBlue) 'random
            Case 2
                .Color = QBColor(10) 'light green

            Case 3
                .Color = vbActiveTitleBar
            Case 4
                .Color = moOrange
            Case Else
                MsgBox "No more examples. Thanks anyway!", vbInformation
        End Select
        lNumClicks = lNumClicks + 1
    End With
    If lNumClicks < 6 Then Map1.Refresh
End Sub
```

Figure 2

# ArcView GIS

## Manipulating Polygons

In addition to splitting and merging polygons, the Edit menu in ArcView GIS contains the Combine Features and Subtract Features choices that can be used to graphically manipulate two overlapping polygon features in a theme.

### Create a Donut Polygon

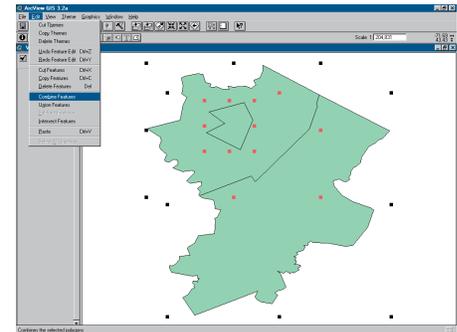
The Combine Features option can create a polygon with a hole inside it. This “donut polygon” can be used to represent a building footprint in a parcel or otherwise separate features. Using the Combine Features option merges the selected polygons but removes any overlap between polygons from the larger polygon.

1. Make the desired theme active and choose Theme > Start Editing. If the Start Editing menu choice is unavailable, either change the file attributes for the data from read only, copy the data source to a medium that can be changed, or choose Theme > Convert to Shapefile to save the theme to a new shapefile and add it to the View.
2. Choose one of the graphic drawing tools—the polygon, circle, or rectangle tool—from the drop-down toolbar and draw a polygon representing the area to be cut out.
3. Choose the Pointer tool. Select both of the polygons.
4. Choose Edit > Combine Features. The drawn polygon will be subtracted from the larger polygon. Choose Theme > Stop Editing to commit these edits. If the selected polygons do not overlap, Combine Features behaves the same way as Union Features and creates one larger polygon containing the area previously occupied by both polygons.

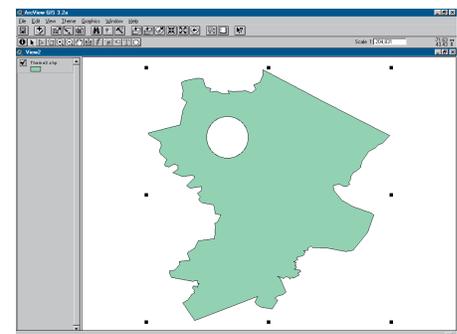
### Editing Overlapping Polygons

Choose Edit > Subtract Features to remove an overlapping area between polygons. Using this option subtracts the polygon on top from the polygon on the bottom. The top polygon remains unchanged. This functionality can be useful when adding a new polygon to a theme. To create a new polygon that abuts an existing polygon without generating sliver polygons, draw the new polygon so that the edges overlap the existing polygon slightly. Use Subtract Features to remove the area of overlap between the new and the existing polygon. The following steps, similar to creating a donut polygon, will accomplish this.

1. Make the desired theme active and Choose Theme > Start Editing.



*Both Combine Features and Subtract are choices in the Edit menu.*



*The Combine Features option can create a polygon with a hole inside it.*

2. Draw the new polygon, partly overlapping an existing polygon.
3. Use the Pointer tool to select both the new polygon and the existing polygon that will be modified. Subtract Features works only on two selected polygons at a time.
4. Choose Edit > Subtract Features to eliminate the area of overlap. Choose Theme > Stop Editing to commit the edits.

To reverse the order of the operation for Subtract Features (i.e., to subtract the bottom feature from the top feature so that the bottom feature is unchanged), hold down the SHIFT key while choosing Subtract Features from the Edit menu.

The Subtract Features option can be used to create an island polygon (a polygon that is totally surrounded by another polygon but which does not overlap the surrounding polygon). To create an island polygon, draw the polygon on top of the surrounding polygon, select both polygons, and choose Subtract Features.