The Windows Clipboard Does More Than Cut And Paste

By Jim Douglas

Windows has radically changed the way users interact with software. Armed with a mouse, users are no longer forced to input information in a predetermined sequence, but can now take control of the interface, clicking through forms, buttons, and input fields in any order and at any time. The immense popularity of Windows applications gives some indication of just how important this control is to users.

Business Basic developers can follow this popular trend and give their users additional control over the Windows environment by adding clipboard functionality to their Visual PRO/5® applications. The clipboard, an electronic scratch-pad controlled by the user, gives users the ability to copy data at will between arbitrary unrelated programs.

In Visual PRO/5 1.0, BASIS introduced three conrols--EDIT, TXEDIT, and LISTEDIT--that support the Windows clipboard's cut-and-paste functions using the standard keystrokes <CTRL-X> for Cut, <CTRL-C> for Copy, and <CTRL-V> for Paste. This functionality can also be made available to the user through a menu system by defining menu items with three reserved menu IDs: 32027 for Cut, 32028 for Copy, and 32029 for Paste.

Visual PRO/5 2.0 supports the simple clipboard interface from version 1.0, but it also expands on it with the addition of the following verbs and functions:

Verb or Function	Description
`fmt=CLIPREGFORMAT("format")`	Register a format or request the number of a registered format.
`CLIPLOCK`	Get exclusive access to the clipboard.
`tf=CLIPISFORMAT(fmt)`	Find out if data in a specific format are currently available on the clipboard.
`CLIPFROMFILE fmt,file$`	Write data to the clipboard from a file.
`CLIPTOFILE fmt,file$`	Read data from the clipboard and

	write them to a file.
`CLIPFROMSTR fmt,str$`	Write data to the clipboard from a string variable.
`str$=CLIPTOSTR(fmt)`	Read data from the clipboard into a string variable.
`CLIPCLEAR`	Clear all data from the clipboard. This is usually only done when exiting from a program and only if the user has copied a large amount of data to the clipboard. Always ask the user before modifying the clipboard.
`CLIPUNLOCK`	Release the clipboard.

Using these new verbs and functions, developers can offer users the ability to cut, copy, and paste in more sophisticated ways from within a Visual PRO/5 program. Users can then exchange data between a Visual PRO/5 application and other Windows programs, such as Microsoft Word or Microsoft Access.

To ensure that a user can copy data from a Visual PRO/5 application to the program of his or her choice, the developer should write data to the clipboard in as many formats as possible. By providing a variety of formats, an application reading the clipboard will be more likely to find a format it understands. For example, Microsoft Excel writes data to the clipboard in several spreadsheet and graphics formats, as well as Rich Text Format (RTF) and plain text. This wide variety of formats makes it possible for Excel data to be easily copied and pasted to a simple text editor or to a more sophisticated word processor. Each program would select the best available format for that particular program. For instance, Windows Notepad would select plain text format from the clipboard while Word would select RTF.

There are sixteen formats with fixed format numbers, and Windows also preregisters several other formats, like RTF and Comma Separated Values (CSV). Developers can retrieve the format numbers for preregistered formats, or register their own customized formats, with the CLIPREGFORMAT() function, but other Windows-based programs won't know about customized formats defined by developers. These formats will only be useful for exchanging data within a single application. Here are the most common formats:

Format	Windows Name	Description
1	CF_TEXT	Unformatted text, ANSI character set

2	CF_BITMAP	Bitmap (BMP) original format
3	CF_METAFILEPICT	Windows Metafile picture format (WMF)
4	CF_SYLK	Windows Symbolic Link Format (spreadsheet)
5	CF_DIF	Data Interchange Format (spreadsheet)
6	CF_TIFF	Tagged Image File Format (commonly used for fax images)
7	CF_OEMTEXT	Unformatted text, OEM character set
8	CF_DIB	Device Independent Bitmap (BMP) newer format
9	CF_PALETTE	Color palette information for CF_DIB data
10	CF_PENDATA	Used by Windows for Pen Computing
11	CF_RIFF	Resource Interchange File Format (sound format)
12	CF_WAVE	Windows sound format (WAV)
13	CF_UNICODETEXT	Unformatted text, UNICODE character set (Windows NT only)
14	CF_ENHMETAFILE	Windows Enhanced Metafile (picture) format (EMF)
15	CF_HDROP	Formatted list of files (not in Windows 3.x)
16	CF_LOCALE	Identifies the language of the CF_TEXT data (not in Windows 3.x)
RTF	CLIPREGFORMAT("Rich Text Format")	Rich Text Format used to exchange data with word processors (typically Microsoft Word)
CSV	CLIPREGFORMAT("CSV")	Comma Separated Values

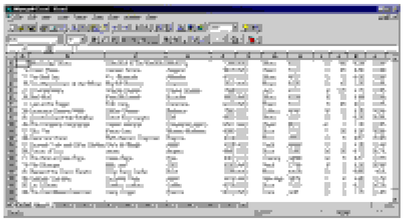
		format used to exchange tabular data between spreadsheet formats like Access and Excel

Out of the 16 standard formats defined in the Windows, a typical Visual PRO/5 program will probably use only CF_TEXT (1), CF_DIB (8), and possibly CF_WAVE (12). Most of the other formats are only of interest to C/C++ programs.

The following is an example using the CSV format to encode tabular information in a form that can be easily pasted to a spreadsheet program like Excel. This example program reads the sample data file **`datagrid.dat`**, which is included with Visual PRO/5 2.0 in the directory **`\basis\tools\guibuild\`**.

```
clip$=$$
cdfile=unt
open (cdfile)"datagrid.dat"
loop:
read (1,end=endloop)cd$,title$,artist$,label$,time$,type$,
+genre$,bin$,etc$
clip$=clip$+cd$+","+title$+","+artist$+","+label$+","
+time$+","+type$+","+genre$+","+bin$+","
+str(num(etc$(1,10)))+","+str(num(etc$(11,10)))+","
+str(num(etc$(21,10)))+","+str(num(etc$(31,10)))+$0d0a$
goto loop
endloop:
csv=clipregformat("CSV")
cliplock
clipfromstr csv,clip$
clipunlock
print "You can now paste the clipboard to Microsoft Excel."
end
```

Run this program, then load Excel and select Paste from the Edit menu. The data will be copied, nicely formatted, into Excel.



Data copied from the clipboard comes into Excel nicely formatted.

These new clipboard verbs can also be used to read data from the clipboard to an application. The following code reads CSV-formatted data from the clipboard into a string variable:

```
csv=clipregformat("CSV")
cliplock
csv$=cliptostr(csv)
clipunlock
```

The CLIPTOSTR() function returns a raw block of binary data that must be converted if it is to be interpreted as a text format like RTF or CSV. The text portion of the string will be null-terminated, so all data past the first **`$00$`** character must be discarded:

```
null=pos($00$=csv$+$00$)
csv$=csv$(1,null-1)
```

When you print **`csv$`**, the result looks like this *[output wrapped for online display -ed.]*:

```
000001,Mississippi Blues,John Hurt & The Ramblers,
Blind Pig,07256,AAD,Blues,N123,12,55,5.99,12.99
000006,Urban Blues,Various Artists,Alligator,05659
,AAD,Blues,N127,10,89,4.56,10.98
```

...several records omitted...

```
000091,Lay It Down,Cowboy Junkies,Geffen,05114,DDD
,Rock,YU10,13,3,6.22,14.78
000096,The Gold Medal Collection,Harry Chapin,Elek
tra,14013,AAD,Rock,AA8,32,2,7.78,22.48
```

With Visual PRO/5 2.0, developers gain a new set of verbs and functions that allows for a richer integration between Visual PRO/5 applications and the Windows clipboard. Applications that take advantage of these new verbs will give their users greater control over the Windows environment and enable them to work more effectively. The result is enhanced user appeal and functionality for any Visual PRO/5 application.

Jim Douglas*, a fourteen-year Business Basic veteran, is working with BASIS on its future product strategy,* Volcano *. For additional information on the popular RTF format, see the RTF specifications at ftp://ftp.microsoft.com/softlib/mslfiles/gc0165.exe. This file expands to a Word document, GC0165 update.doc.*