

Windows Online™

"the Weekly"

May 9, 1992 - Issue 40

Windows Online "the Weekly" (WOLW) is published electronically every Saturday and distributed through many well known on-line services. WOLW contains articles and weekly columns by WOL's staff and freelance writers. Subject matter includes all aspects of MS Windows and its associated applications and utilities. Also covered are OS/2, Computer News, Communications and other GUI subjects. If you are interested in writing for "the Weekly" contact us through the numbers listed at the end of this publication.

In this issue:



New Product News _____ by Rich Young
hDC Power Launcher v2.0 - A Windows Program Launch Utility

Interesting Computer News _____ by Randy
Wong

Windows Tips _____ by Hugh
Hardie

To Buffer or Not to Buffer, That is the Question
Controlling the Actions of Users - Part Two
Uncle Bill's Tip of the Week

Visual Basics _____ by Barry Seymour
The Window Manager List

Windows Software Review _____ by Brian Parks
SmarText for Windows version 2.0
Lotus Development Corporation - Word Processing Division

Desktop Publishing _____ by Rob
Weinstein
CHARTIST--A flow chart creator



hDC Power Launcher v2.0

hDC presented Power Launcher version 2.0 at Windows World 1992. Power Launcher 2.0 has a new main window that provides a graphical control center to quickly and easily access the customization and productivity tools in Power Launcher. In addition, Power Launcher takes advantage of Object Linking and Embedding (OLE) and Drag and Drop capabilities of Windows 3.1 to provide the ultimate application launcher and manager for Windows. The new main window for Power Launcher is called Power Bar. This thin, horizontal toolbar across the top of the user's screen allows instant access to:

A convenient Command Line that executes any hDC Enhanced Command, including DOS commands (without running a DOS session) and includes an "auto completion" feature to save time when typing commands.

Key status information -- memory, resource, and disk space viewing

Task management -- find hidden and visible applications quickly

File management -- Windows File Manager is now one click away

Virtual Desktop map to organize work area using virtual screens

Customizable Power Buttons for favorite Enhanced Commands, and support for Drag and Drop, for functions like printing and deleting files.



Power Launcher can be used in addition to any shell, but also makes an ideal shell for the power user looking for a resource-efficient, yet powerful and configurable solution. Our new Action Editor is an OLE server for hDC Enhanced Commands. The hDC Action Editor is similar to the Object Packager application included in Windows 3.1, but supports the hDC Enhanced Command, a superset of the Windows run command. Using the Enhanced Command, users can launch multiple applications, documents and MicroApps, run scripts, play back macros, execute DDE commands, call DLLs and more in a single command. For example, the Action Editor can be used to perform an SQL database query to get the latest data for a report, send it to Excel and create a chart through DDE commands, and then paste it into the document. With the Drag and Drop capabilities in Windows 3.1, the customizable floating toolboxes created with the Power Toolbox can be used as drag targets. When a file is dropped onto a toolbox button, it will automatically run any hDC Enhanced Command assigned. For example, drag a file from the new Windows File Manager onto an hDC Power Toolbox button that automatically creates an email message, prompts for the recipient's name, and sends the message to the designated recipient. Power Launcher is the most powerful, convenient, and resource-efficient application launcher and manager for Windows. The product is compatible with Microsoft Windows 3.1 and Windows 3.0.



Pricing and availability

hDC Power Launcher version 2.0 will be available in April 1992 for a suggested retail price of \$99.95. Registered users of hDC Power Launcher V1.0 can upgrade for \$19.95 plus shipping and handling. If the previous version was purchased after February 1, 1992, the upgrade is free with proof of purchase.



System Requirements

Microsoft Windows-compatible personal computer
Microsoft Windows 3.0 or higher
1.2 MB disk drive or 720K 3.5"disk drive
Microsoft or compatible mouse

hDC has delivered innovative solutions for PC users since Windows 1.0. From doubling the size of the Windows 1.0 color palette, to creating the first user-friendly shell for Windows 2.0, to showing you the inner workings of Windows 3.0 memory management, hDC has been the technical leader. hDC products are designed to fully exploit the power of the graphical user interface in Microsoft Windows environments. In addition to the new hDC Power Launcher, the company markets hDC Windows Express, a menu system for corporate Windows standardization, and hDC FirstApps, FileApps, and Icon Designer, three products that use hDC's pop-up MicroApp approach to add customization and productivity tools to Windows. hDC was founded in 1986 and currently has an installed base of more than a half million . The company is located at 6742 185th Ave NE, Redmond, Washington 98052.

hDC Power Launcher, hDC PowerApps, Enhanced Command, hDC FirstApps, hDC FileApps, hDC Icon Designer, hDC Windows Express and MicroApps are trademarks of hDC Computer Corporation. Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation.



This press release researched, compiled and edited by Rich Young for Windows Online "the Weekly".





Computer News

*By
Randy Wong*



Inking The Pen



With Microsoft releasing Windows for Pen Computing, four manufactures, NCR, NEC, MicroSlate, and Momenta, are shipping hardware that runs on the Microsoft's Pen System. Oracle announced Oracle Card for Pen Computing, a graphical client/server application development tool for the Microsoft Windows for Pen Computing. The developer's version will be about \$699 and \$299 for a runtime, single-license version. On May 25, Slate's pen spreadsheet, At Hand, will be announced. Dan Bricklin, inventor of VisiCalc, precursor to Lotus 1-2-3, assisted with most of the work on At Hand. At Hand will be available on pen computers that run Go's PenPoint operating system. This application will be one of six pen applications that Slate will be shipping this summer.

Not to be left out, Apple will be showing off its pen based, handheld computer, code named the Newton, at the Consumer Electronic Show. Users will be able to enter information by writing it on the screen. A unique feature of Newton, is that it will be able to communicate with a desktop computer, wireless, up to six feet away, by using infrared lights. The videocassette sized handheld device will be priced less than \$700.00, and will be available in the beginning of 1993.



DEC Seeks Edge With NT



Digital Equipment Corp. and Microsoft Corp. tighten the bonds that hold them together. DEC has confirmed that Microsoft Windows NT will run on DEC's next generation Alpha platform. The two companies' latest agreements include an assortment of ways to expand support and service by DEC of the Microsoft Windows environment and its products. The arrangement also calls for DEC to readjust its Network Application Support (NAS) software platform to conform with those of Microsoft's Windows Open Services Architecture (WOSA). This will allow Windows applications to interact on a multiplatform environment using NAS. Microsoft Windows NT is expected to appear later this year, and will run on the Intel Corp., Mips Computer Systems, Inc., and now the DEC Alpha hardware platforms. DEC's Alpha chip will begin

shipping sometime in the first quarter of next year. A workstation using both of these technologies is a year away.

Icon Based SQL Database

IntelligenceWare, Inc. is scheduled to release Iconic Query, a Microsoft application for allowing users to access database information by using graphical icons, instead of dealing with query language commands. Iconic Query will offer read access to database systems, such as Borland's Paradox, Oracle's Oracle, Sybase's Sybase, and IBM's DB2. The product will include a library of icons, for attaching to tables. A hypertext function allows the user to relate or link variables. Double clicking on icons can let a user query through the contents of a database. Behind the icons is the link between tables, and full access to joined databases. Data files from spreadsheet applications, statistical analysis applications, data management applications, or graphics packages are candidates that might be joined into a database.

Reach Out And Touch

Another alternative to the uprising pen computing, is the touchscreen computing. Touchscreen computing is beginning to start its climb as another ease of use interface, and affordable alternative to the mouse, keyboards, or pens. Touching the screen with your finger or any object, will cause the command to be executed. Not all pointing devices is right for someone, such as people have coordination problems with moving the mouse, dentists and doctors have some allergy with keyboards, and users who get tangle with the pen cable. Whatever the reason is, the ease of just a touch to do something is still the best way to do things. Software compatibility for this rising technology is another great feature. The touchscreen computers can work with most pointer driven software, such as Microsoft Excel. Several corporations are beginning to experiment in using this technology. They are Grace Hospital, McDonald Corp., Blockbuster Enterprises, and others.

Ringing Out The Weeds

Using computers, criminal hackers steal credit card and telephone account numbers, then charge thousands of dollars in goods and telephone calls. Two teenagers were arrested last week in New York city for charging on stolen account numbers. The two were discovered when their order arrived at the real card holder's home. This prompted the FBI to begin the crackdown of over 1,000 computer criminals using these tactics. The credit card and phone accounts were stolen from

the Equifax credit bureau and other unknown sources. Currently on the trail left behind by the two, authorities arrested two Ohio hackers and the seizure of computer equipment in three cities.



Compaq Goes To The Orients



Compaq will be going outside of the United States to produce their low cost notebooks. Following Compaq's specifications, Citizen, a Japanese manufacturer, will be building all three notebooks for Compaq. Announcement of these machines is expected sometime in June, and the starting base price is \$999.

Windows Tips

To Buffer or Not to Buffer, That is the Question

Controlling the Actions of Users - Part Two

Uncle Bill's Tip of the Week

To Buffer or Not to Buffer, That is the Question

Well, its buffer controversy time again. It seems that the Windows User's Guide has caused a spot of puzzlement about what it means when it talks about buffers. The offending page is Number 522.

Page 522 states the following:

"In general, you should set **buffers=20** or less. If you are using SMARTDrive, you can set **buffers=10**. If you are using MS-DOS version 5.0 or later, you can set the **buffers=30**."

The controversy is raging over what to set buffers to, if you are using a

combination of the products mentioned above. What if you are using SMARTDrive and MS-DOS 5.0. Is the number suggested for SMARTDrive overridden by the settings for MS-DOS 5.0 if you use them both. Suppose you are still on MS-DOS 4.01 and running SMARTDrive, what then?

The argument has raged fast and furious across Windows messaging conferences around the world. Everyone has their own interpretation of the meaning of the words. At times it takes on the airs of a discussion of religious principles. Advocates argue loftily and, sometimes, not so loftily for their individual positions.

Still others try out the various permutations and combinations and report back with glee that nothing much changes whatever you choose to do. These individuals, of course, represent the atheists.

I decided that this was too important a question to be left to the soothsayers and so I attempted to reach a representative of the individual who created Windows and MS-DOS 5.0. I was informed that, after careful examination of and rumination on, the technical scriptures that the answer is:

If you are running any kind of disk cache, SMARTDrive, PC-KWIK, Hyperdisk or whatever, you should set **buffers=10**. This is irrespective of what version of DOS you are using.

If you are running without a cache and are using a version of DOS **prior** to 5.0, you should set **buffers=20** since those versions of DOS could not create any more than 20 buffers.

If you are running without a cache and are using DOS 5.0, you should set **buffers=30**. This is appropriate for 99.5% of all DOS version, application type and hardware combinations.

If this is not clear, call Uncle Bill and get him to explain.

Controlling the Actions of Users - Part Two

In the last column, we started to investigate ways in which the activities of users could be controlled in larger environments where the importance of standards overshadows everyone's right to do their own thing.

Windows offers a number of controls over user access to features and functions of Windows. These controls are implemented through commands entered in the PROGMAN.INI file. In the last column we discussed the effects of the NoRun, NoFileMenu and NoClose commands. The commands left to examine are:

NoSaveSettings=

EditLevel=

These commands are used in the following manner:

The NoSaveSettings command can be used to prohibit a user from saving any changes he or she made to Program Manager settings. The user may still be able to make changes such as creating new groups and group items but the new settings would not be saved. 0 allows saving, the default, and 1 prohibits saving.

The EditLevel command controls in greater detail what users can and cannot do to items under the control of the program manager. There are four levels of control which are implemented by placing any number between 1 and 4 after the equals sign which follows the command. The four levels are as follows:

1 - File menu commands; new, move, copy and delete are available only when a group is selected. The command prevents these file menu commands from being used on groups as opposed to programs within groups.

2 - File menu commands; new, move, copy and delete are not available plus all restrictions for level 1.

3 - File/Properties/Command line; text cannot be changed for program items plus all restrictions for levels 1 and 2.

4 - File Properties dialog box; no commands can be changed, only viewed plus all restrictions for levels 1, 2 and 3.

Uncle Bill's Tip of the Week

Switching between applications is easiest when you can click on the icon of the application you would like to use next. This might be easy if you were running the current active application in a window as opposed to full screen.

I tend to run my most frequently used application, word processing, full screen. The lack of icons to click on is complete but Uncle Bill has a solution.

The ALT/ESC keys, pressed simultaneously, cause all of the icons on your desktop to pop through whatever is on the screen at the time. Switching applications is easy, just point and click.

Thanks, Uncle Bill!



The Window Manager List

In tales of old, a magician gained power over someone if he discovered their true name. Using that name in spells and incantations, a sorcerer could wreak mischief or havoc, bestow favors or curses on the person whose name he had learned.

In Windows, the secret name which is the key to power is the window's handle. Once you have a window's handle, you can send messages to the window to restore, minimize, maximize or destroy itself. You can determine the window's task ID, display contents, coordinates, windowstate or caption. Having a window's handle gives you control over that window, provided you know how to access the required API spells. Note that this applies to any window and any program, not just the forms you create in Visual Basic.

Windows maintains a list of all active windows called the Window Manager's List. As its name indicates, this is a list of all existing parent and child windows. You can use this list to get information about everything that's going on in the current session of Windows; you can get a list of handles, task IDs, and captions for every process in the system. If you wished, you could limit that list only to visible windows, or you could limit the list to top-level windows, ala the Task Manager. Once you can access the Window Manager's List the choice is up to you.

The API call `GetWindow` allows you to traverse the Window Manager List, getting the handle of each window. The call, along with constants for required parameters, looks like this....
(remember the `<+>` signifies a line join.)



```

Declare Function GetWindow Lib "User" (ByVal hWnd As Integer, <+>
ByVal wCmd As Integer) As Integer
    Const GW_HWNDFIRST = 0
    Const GW_HWNDLAST = 1
    Const GW_HWNDNEXT = 2
    Const GW_HWNDPREV = 3
    Const GW_OWNER = 4
    Const GW_CHILD = 5

```



This function takes two parameters; first, the handle of a window; second, a parameter which specifies what kind of related window handle to retrieve. Since you're accessing the Window Manager's List, the descriptive constant names give you a clue as to their purpose. (All descriptions taken from the Microsoft Window's Programmer's Reference.)

GW_HWNDFIRST ID's the first sibling window of a child window, otherwise the first top-level window in the list.

GW_HWNDLAST IDs the last sibling window for a child window. Otherwise it returns the last top-level window in the list.

GW_HWNDNEXT IDs the next window in the Window Manager's List.

GW_HWNDPREV IDs the previous window in the Window Manager's List.

GW_OWNER Identifies the window's owner

GW_CHILD ID's the window's first child window

If you want to access the list of all active windows from the beginning, you need to get the handle of a top level window, then issue a `GetWindow` call with that handle and the `GW_HWNDFIRST` parameter. After that you can cycle through the whole list with each subsequent window handle and the `GW_HWNDNEXT` parameter.

At first glance this seems to be a bit of a conundrum; in order to **get** a handle, you need to **provide** a handle. Fortunately VB can give you the handle of any Visual Basic form through the `.hWnd` property. For example, to get the handle of `Form1`, you simply reference `Form1.hWnd`. After you obtain a handle to any window the API call `GetParent` allows you to 'climb the tree' from that window to it's

topmost parent. If a window has no parent the function returns zero, and you know you're at the 'top of the family tree'.



Declare Function GetParent Lib "User" (ByVal hWnd As Integer) As Integer



To sum up the process...

1. Get the handle of your VB form using .hWnd.
2. Climb to that window's topmost parent using GetParent.
3. Use GetWindow with the topmost parent's handle and the GW_HWNDFIRST parameter to get the first window in the Window Manager's List.
4. Use GetWindow with the GW_HWNDNEXT parameter to get the handle of the rest of the windows in the list.

A code snippet (actually a mondo code **chunk**) from this week's example demonstrates this three step technique...



'Use handle of vbex12 window & climb window's family tree to the top...

```
WindHandle% = VBEX12.hWnd
OldHandle% = WindHandle%
WindHandle% = GetParent(WindHandle%)
If WindHandle% <> 0 Then
    Looping = -1 'TRUE
    While Looping
        OldHandle% = WindHandle%
        WindHandle% = GetParent(WindHandle%)
        If WindHandle% = 0 Then 'TOP FOUND
            WindHandle% = OldHandle%
            Looping = 0 ' false
        End If
    Wend
Else 'if WindHandle% = 0: This window has no parent! Back down one..
    WindHandle% = OldHandle%
End If
```

'Obtain the first toplevel window in the Window Manager's List...

```
WindHandle% = GetWindow(WindHandle%, GW_HWNDFIRST)
StartPlace% = WindHandle%
```

```
Looping = TRUE
```

```
While Looping ' Get info on ALL windows in the Window Manager's
List...
```

```
  'get window caption...
```

```
  StrLen% = GetWindowTextLength(WindHandle%) + 1
```

```
  StrText$ = Space$(StrLen%)
```

```
  Result% = GetWindowText(WindHandle%, StrText$, StrLen%)
```

```
  WindowTask% = GetWindowTask(WindHandle%)
```

```
  'Assemble a string for addition to the task listbox..
```

```
  NewStr$ = "Task ID: " + LTrim$(Str$(WindowTask%)) + Chr$(9)
```

```
  NewStr$ = NewStr$ + "Handle: " + LTrim$(Str$(WindHandle%))
```

```
  'add it to the list....
```

```
  NewStr$ = NewStr$ + Chr$(9) + StrText$
```

```
  List1.AddItem NewStr$
```

```
  'get the next window handle;
```

```
  'quit if you return to the start or reach the end...
```

```
  Result% = GetWindow(WindHandle%, GW_HWNDNEXT)
```

```
  Select Case Result%
```

```
  Case StartPlace%, 0
```

```
    Looping = 0 'FALSE
```

```
  Case Else
```

```
    WindHandle% = Result%
```

```
  End Select
```

```
Wend
```



As you get all these windows, you can also get the Task ID for each window. According to the *Microsoft Windows Programmers' Reference*, "a task is any program that executes as an independent unit. All applications are executed as tasks, and each instance of an application is a task." Couldn't have said it better myself.

Getting the task IDs for each window allows you to determine which running program each window belongs to. This is achieved with an easy call to **GetWindowTask...**



Declare Function GetWindowTask Lib "User" (ByVal hWnd As Integer)

<+>
As Integer



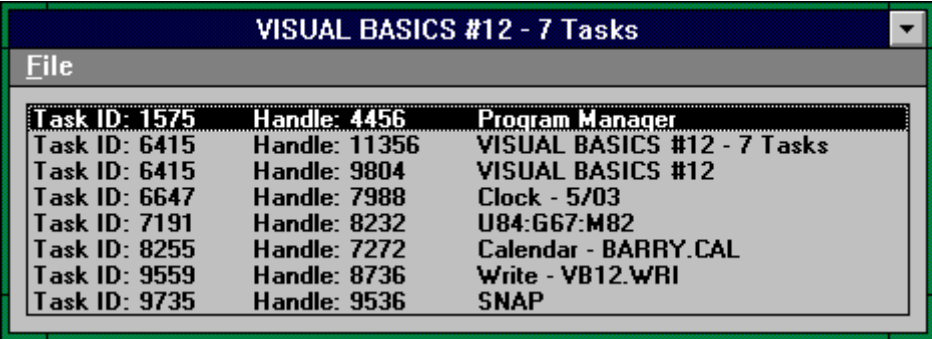
Send the window handle; GetWindowTask returns it's task ID. If you need to get the total number of tasks, another simple API call is GetNumTasks. Make the call and you're returned the number of active tasks....



Declare Function GetNumTasks Lib "Kernel" () As Integer



This week's example scans the Window Manager List and displays the information. The Task ID, window handle and window caption for each **visible window** are placed in a sorted list box, hence grouping together windows with the same task ID. A timer checks the total number of tasks at regular intervals. If that number changes, the listbox is cleared, the list is scanned again and the listbox is refilled. When running, it looks like this...



Task ID: 1575	Handle: 4456	Program Manager
Task ID: 6415	Handle: 11356	VISUAL BASICS #12 - 7 Tasks
Task ID: 6415	Handle: 9804	VISUAL BASICS #12
Task ID: 6647	Handle: 7988	Clock - 5/03
Task ID: 7191	Handle: 8232	U84:G67:M82
Task ID: 8255	Handle: 7272	Calendar - BARRY.CAL
Task ID: 9559	Handle: 8736	Write - VB12.WRI
Task ID: 9735	Handle: 9536	SNAP

You can modify the program to show information on all windows in your system; you may be surprised at how much is going on behind the scenes! You can also modify the program to show just top-level windows; this gives you the same list that Windows' Task Manager shows. I've included commented-out sections of code for all three options in the source code.

Also note that the invisible 'parent' window every VB application has shows up on this list! In this case it's Task 6415, handle 9804. Visual Basic applications in effect have two parent windows; one visible, one not! Go figure.

Once you have this kind of information you can control all these windows with further program modifications. See the text file WINAPI.TXT for more parameters to use with the SendMessage call -- you might just get drunk with power!

There are a lot of other little API calls in this week's example that I can't go into for lack of space; **SendMessage** is used to quickly clear the ListBox. I use **GetWindowText** and **GetWindowTextLength** to obtain a window's caption. I also use the self-explanatory API call **IsWindowVisible**. You can see how they're used in the code.

To create this week's example, create the form VBEX12.FRM as follows...

```
Form VBEX12:
    FormName = VBEX12
    ControlBox = 0
    Maxbutton = 0
```

```
ListBox List1
    Sorted = -1 'TRUE.
```

```
Timer Timer1
    No need to set interval - code sets it to 1000.
```

Create a menu for VBEX12 with the following items...

```
CTLNAME  CAPTION
FileMain  &File
          FileExit  E&xit
```

When you're done, bulk paste the following code into **the form's Declarations section**. All code will fall into place.



```
' Booleans
Const TRUE = -1
Const FALSE = 0

'-----
'
' Method parameters '
'-----

' MousePointer (form, controls)
Const DEFAULT = 0      ' 0 - Default
Const HOURGLASS = 11  ' 11 - Hourglass

' WindowState (form)
Const NORMAL = 0      ' 0 - Normal
Const MINIMIZED = 1  ' 1 - Minimized
Const MAXIMIZED = 2  ' 2 - Maximized
```

```

Declare Function SendMessage Lib "user" (ByVal a%, ByVal b%, ByVal c%, ByVal d As Any) As Long
Const WM_USER = &H400
Const LB_RESETCONTENT = WM_USER + 5

Declare Function GetWindow Lib "User" (ByVal hWnd As Integer, ByVal wCmd As Integer) As Integer
Const GW_HWNDFIRST = 0
Const GW_HWNDLAST = 1
Const GW_HWNDNEXT = 2
Const GW_HWNDPREV = 3
Const GW_OWNER = 4
Const GW_CHILD = 5

Declare Function GetWindowText Lib "User" (ByVal hWnd As Integer, ByVal lpString As String, ByVal aint As Integer) As Integer
Declare Function GetWindowTextLength Lib "User" (ByVal hWnd As Integer) As Integer
Declare Function GetFocus Lib "User" () As Integer
Declare Function GetNumTasks Lib "Kernel" () As Integer
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) As Integer
Declare Function GetWindowTask Lib "User" (ByVal hWnd As Integer) As Integer
Declare Function IsWindowVisible Lib "User" (ByVal hWnd As Integer) As Integer

Dim NumberOfTasks As Integer

Dim ListHandle As Integer

Function StripString (StringToStrip As String) As String
    While InStr(StringToStrip, Chr$(0))
        Mid$(StringToStrip, InStr(StringToStrip, Chr$(0)), 1) = " "
    Wend
    StripString = LTrim$(RTrim$(StringToStrip))
End Function

Sub Form_Load ()
    List1.Left = 120
    List1.Top = 120
    VBEX12.Width = 7000
    VBEX12.Height = 1
    VBEX12.Visible = TRUE
    VBEX12.SetFocus
    List1.SetFocus
    ListHandle = GetFocus()
    Timer1.Interval = 1000
End Sub

Sub Timer1_Timer ()
'This sub won't execute unless the number of items in the task list
'changes.

    If NumberOfTasks = GetNumTasks() Then ' no change in number of tasks
        Exit Sub
    End If

    If VBEX12.WindowState <> 1 Then Screen.MousePointer = 11
    NumberOfTasks = GetNumTasks() ' get new number of tasks, change form caption
    VBEX12.caption = "VISUAL BASICS #12 - " + LTrim$(Str$(GetNumTasks() - 1)) + " Tasks"

'clear listbox with API call SendMessage. This is FAST!
Y& = SendMessage(ListHandle, LB_RESETCONTENT, 0, ByVal "")

'Use handle of taskmon window & climb window's family tree to the top...
WindHandle% = VBEX12.hWnd
OldHandle% = WindHandle%
WindHandle% = GetParent(WindHandle%)
If WindHandle% <> 0 Then
    Looping = TRUE
    While Looping
        OldHandle% = WindHandle%
        WindHandle% = GetParent(WindHandle%)
        If WindHandle% = 0 Then 'TOP FOUND
            WindHandle% = OldHandle%
            Looping = 0
        End If
    Wend
Else 'if WindHandle% = 0: This window has no parent! Back down one..
    WindHandle% = OldHandle%
End If

'Obtain the first toplevel window in the Window Manager's List...
WindHandle% = GetWindow(WindHandle%, GW_HWNDFIRST)
StartPlace% = WindHandle%

Looping = TRUE
While Looping ' Get info on ALL windows in the Window Manager's List...
    StrLen% = GetWindowTextLength(WindHandle%) + 1
    StrText$ = Space$(StrLen%)
    Result% = GetWindowText(WindHandle%, StrText$, StrLen%)
    WindowTask% = GetWindowTask(WindHandle%)
    NewStr$ = "Task ID: " + LTrim$(Str$(WindowTask%)) + Chr$(9) + "Handle: " + LTrim$(Str$(WindHandle%))

    'THREE OPTIONS FOR POPULATING THE LISTBOX:

```



```

'Option #1===== 'If the window's
visible, top-level,
'and has a caption, add it to the list.

If IsWindowVisible(WindHandle%) And StripString(StrText$) <> "" And GetParent(WindHandle%) = 0 Then
    NewStr$ = NewStr$ + Chr$(9) + StrText$
    List1.AddItem NewStr$
End If
'=====

'Option #2===== 'Add all windows
with a caption to the list, visible or not.
'If StripString(StrText$) <> "" And GetParent(WindHandle%) = 0 Then
    'NewStr$ = NewStr$ + Chr$(9) + StrText$
    'List1.AddItem NewStr$
'End If
'=====

'Option #3===== 'Add ALL items to
the list...
'NewStr$ = NewStr$ + Chr$(9) + StrText$
'List1.AddItem NewStr$
'=====

'Get the next handle & move on...
Result% = GetWindow(WindHandle%, GW_HWNDNEXT)
Select Case Result%
Case StartPlace%, 0
    Looping = FALSE
Case Else
    WindHandle% = Result%
End Select
Wend

List1.ListIndex = -1
Form_Resize
If VBEX12.WindowState <> 1 Then Screen.MousePointer = 0
End Sub

Sub Form_Resize ()
If VBEX12.WindowState = 0 And List1.Listcount > 0 Then
    OffSet% = VBEX12.height - ScaleHeight
    List1.height = VBEX12.TextHeight("A") * List1.Listcount
    VBEX12.height = List1.height + OffSet% + (List1.top * 2)
End If
List1.Width = ScaleWidth - (List1.Left * 2)
End Sub

Sub FileExit_Click ()
End
End Sub

```



As always, this column plus sample code is available on the Windows Online BBS in Danville, California, phone 1 510 736-8343. This column in Windows Write 3.0 format, along with the Visual Basic source code is in VBEX12.ZIP, and may be distributed as freeware.

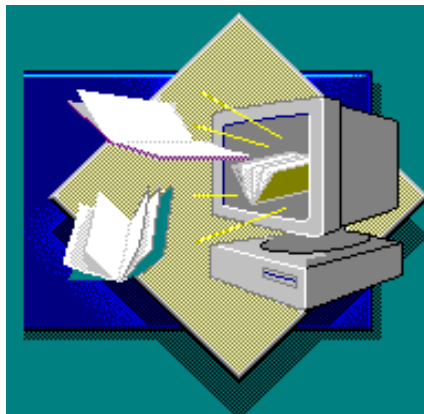
Barry Seymour
Marquette Computer Consultants
San Rafael, CA 415/459-0835

for **Windows Online "the Weekly"***



WINDOWS SOFTWARE REVIEW

Written by Brian A. Parks



SmarText for Windows version 2.0

Lotus Development Corporation

Word Processing Division

Curl up with a good book.

I first saw this product demonstrated at one of the many Lotus booths at the Windows World show in Chicago. To say that I was impressed would be putting it mildly. After the demonstration the only thing I could think of to say was, "**Where has this been hiding?!**" I had no idea that a product like this even existed, much less dared dream that I'd be seeing it in action. With SmarText, managing information efficiently and effectively will no longer be a lofty dream. SmarText for Windows is a complete hypertext software program for building, publishing, and reading electronic documents, minus the headache.

Using your existing text and graphics files, SmarText puts together an electronic document that includes a table of contents and index. But it doesn't stop there. It also creates hypertext links throughout your entire document....**Automatically**. Graphics can not be linked automatically in this version, you must do these manually. (Remember Apple's HyperCard, hypertext links that you had to create one at a

time, day after day? **Ha!** me neither.)

With SmarText you don't have to be a programmer to create an impressive on-line document.

SmarText features the ease of use of the Windows interface, which make learning the program a cinch. (The actual Users Guide for the Builder program is less than 115 pages long.) This leaves you with more time to be more creative.... or "**NOT!**"

What's in it?

SmarText for Windows 2.0 consists of two programs. The first is the **Builder Program**. With this program you may:

- Prepare and specify the source files
- Specify the build options
- Customize the document that SmarText creates

All in plain english. You tell the program what text and graphic files to use, what to do with them when it gets them and then add those finishing touches to it that make it your own. A SmarText document may contain more than one source file from any combination of ASCII text files, word processor files, graphic files, or keyword and stopword files. You may use up to 3500 of each of the above mentioned files.

How do I do it

Creating source files for the SmarText "DOCUMENT" is as easy as;

1. Opening a new file and giving it a name.
2. Select the type of files to include from a list in the "INCLUDE" dialog box.
3. Determine your unique "KEYWORDS and STOPWORDS" if any.

Creating or building the actual "Electronic Document" is just as simple.

1. Create a new, or open a previously created SmarText doc (*.STX)
2. Choose File/Build
3. Create the OUTLINE, TEXT LINKS, and WORD INDEX.
(If this is a new document, these options will be selected automatically.)
4. If you want to specify options for the text links, outline, and index choose "OPTIONS."

Now all that's left to do is sit back and let SmarText handle the transformation, which it can do in the background. Freeing you to do more important things...or not. It should be noted that SmarText does nothing to your original documents and graphic files. They are left as they are found.

Spreading the word

SmarText's Build program also contains a **READER PROGRAM**. This is used to access the information in the electronic document. Lotus also supplies a copy of ADOBE TYPE MANAGER ver 1.15 bundled with version 2.0 of SmarText for Windows. The separate READER program allows you to pass on the document that you have created to another user without fear that it will altered in any way. It is a read only version.

A special Publishers license is available on a case by case basis. There are other special options available to Publishers only. To find out more about this option, Publishers may contact Robin Sargent or Joel Ackerman at Lotus Development, Word Processing Division.

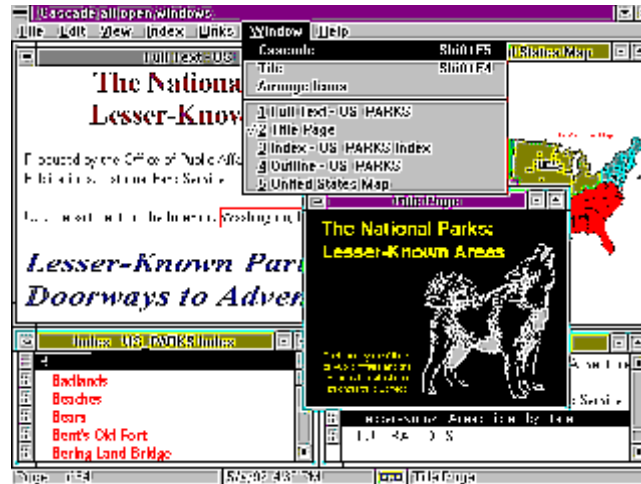
Read the book

You view the document as you would any other "book." When you start the Reader program you are presented with a "**BOOKSHELF**." Here you may select a book to read. The bookshelf contains icons that represent the documents that you have available.



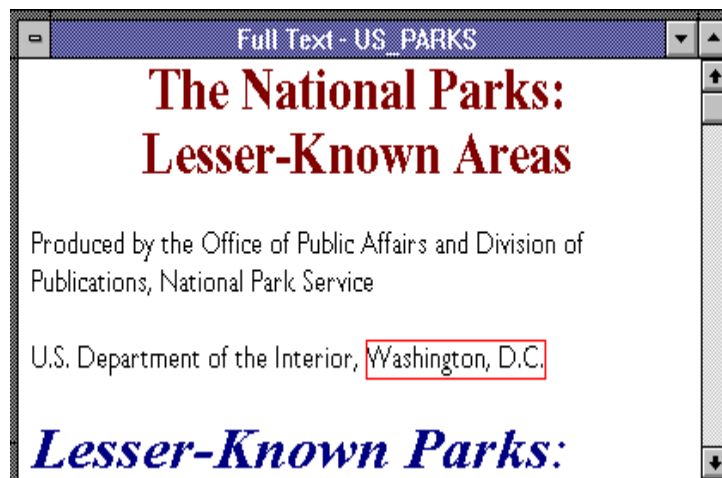
Double-clicking on an icon opens the book. You may also open a document using the File/Open command.

When you open a book, a "title page" can appear, if it is specified.



Objects or words on the title page can contain links to other parts of the text in the document. If no title page is specified, you are taken directly to the "full text" window. Outlines, indexes and graphics are displayed in their own windows. SmartText does not support the mixing of graphics and text in the same window.

As you read through the document you may often find words with boxes around them.



These are links. Some of the different kinds of links that you may run across are;

Text to text.

When you double-click on this type of link, SmartText displays related text in either another location within the same document or in another document (book) altogether.

Text to illustration.

When this type of link is activated, an illustration window is opened that contains a picture related to the initial link.

Text to executable application.

When you double-click on this type of link, SmartText launches another application such as a spreadsheet, communications, or other program.

Moving around in the document or accessing SmartText's features is made simple with the help of a user definable **SmartIcon** bar.



If by chance you customize the SmartIcon bar and then forget what an icon does. Simply pointing to that icon and pressing the right mouse button will display it's function in the upper right corner of the screen.

You can use SmartText's **SEARCH** feature to perform **Boolean searches**. This, my friends is a very handy feature. You can use it from within an open document or from the Bookshelf window. Use it to specify a search term or any desired Boolean operators. (Or, and, but not.) SmartText finds all instances of each search term, along with instances of the entire search query. It then shows you the results of the search in order of appearance.

SmartText's features list is long and impressive. Suffice it to say that in a corporate situation, SmartText will effectively end the paper chase or the paper avalanche.

SmartText is **NETWORK READY** right out of the box (nuff said)

Installation is as simple and straight-forward as any other Windows application. SmartText may be installed from the Program Manager using File/Run....A (or) B:\install. Or you can install it from the DOS prompt by typing, WIN A (or) B:\install. The program requires a minimum of 3 MB to install it's file on your hard drive. 1 MB of RAM is needed for the program to run correctly (2 MB's if running in Enhanced mode.) If you thought about running Windows in REAL mode and using SmartText...forget about it.

SmartText for Windows version 2.0 is available now. Retail pricing is \$495.00 for the Builder program .bundled with Adobe Type Manager ver 1.15. Separate Reader programs are also available with discounted pricing for quantity ordering.

For more information or to receive a free working demo contact;

Lotus Development Corporation

1000 Abernathy Road
Bldg. 400 Suite 1700
Atlanta Georgia 30328
1 800 831-9679

If you've unearthed a rare find, take a minute and drop me a line.
Brian Parks c/o Windows Online.

Copyright (c) 1992, Brian A. Parks
Copyright (c) 1992, Hopscotch Research Service

Desktop Publishing

CHARTIST--A flow chart creator

Recently I had a somewhat unusual project. I was approached by some business systems consultants who wanted me to put their latest report about a corporation into final presentation form. Most of the report as very straight forward text, organized in sections and sub-sections, with an index and footnotes. But a major part of the presentation consisted of four or five flow charts which were used to describe the corporate organization, inter-departmental relations and workflow.

Flow charts are a highly specialized, technical tool, which are probably familiar to many of you because of their use in designing computer programs, and electronics equipment. The charts have what essentially amount to their own language, with specific symbols, flows and movement directions, each of which carry very specific meanings (in a workflow chart, for example, a rectangle signifies a step, while a diamond signifies a decision). They can be used to chart organizational structure, workflow, interaction, decision processes, or to graphically represent virtually any interaction.

In this case, I simply used CorelDRAW to draw the necessary flow charts, creating the basic shapes and then copying them as necessary and inserting the appropriate colors, fills and text. I then connected the

symbols with individually created arrows. The process produced beautiful charts, but was laborious and entailed using a drawing program for something which, while it functioned adequately, was not really in its venue.

So it was no surprise that when I saw a program called "Chartist", a flow charting program, on CompuServe the day after I finished the report, it attracted my attention. If you are ever faced with creating a flow chart of any type, this Windows shareware program should attract your attention also.

Chartist v. 1.02 is published by Novagraph, a company out of Dallas, Texas. This remarkable program does an excellent job of automating the task of creating flow charts, but takes the process one step beyond a simple graphic representation. Chartist creates a dynamic interaction between symbols (circles, squares, diamonds, etc.), so that the symbols and their connections are actually related. If you delete a symbol, for example, you delete all of the flows to and from that symbol, automatically.

Let me explain with a simple example. Chartist opens on a blank page, delineated with a quarter inch grid. The Chartist tool bar includes buttons for symbols, lines, symbol style, line style, printing, etc. There is a menu bar which includes which provides quick access to the standard Windows 3.x commands, as well as zoom and formatting capabilities.

If you select the Symbol button and place a symbol on the page, you are automatically presented with a dialogue box which allows you to choose from 48 pre-defined technical symbols--parallelograms, circles, ovals, reels, diamonds, rectangles, and so on. Each symbol can be filled with color (or not), and descriptive text can be placed within the symbol (in any available font). When you want a flow line connecting one symbol with another, you simply choose the Line button, click the cursor on the source symbol and drag the pencil cursor to the destination box. Chartist automatically places the necessary line, in the traditional format. And the line is linked to both boxes. If you delete the a box, the line goes also.

All of this doesn't mean Chartist does the work for you. But it does make the process of putting together a flow chart a whole lot easier, and for people who don't work with the charts all the time, far more accessible. Further, since Chartist is compatible with any Windows supported printer with graphics capability, it can be used in a wide range of environments. Finally, the documentation is simple, straight forward, and easy to follow. And it comes with several examples which

show some of the many uses--organizational design, decision trees, and several technical charts.

An evaluation copy of Chartist is available on Compuserve, or a registered copy is available for \$72.95 from Novagraph, 10014 Lawler Rd., Dallas Texas 75243-2612.

SCREENED BACK COPY

Graphic artists and designers have long used the trick of using screened text behind foreground type to create visually interesting effects. This is a technique which can be used in black-and-white or in color, and can turn simple text into a visually appealing graphic. What is nice is that the current sophisticated desktop publishing programs such as Arts & Letters or CorelDRAW, let you create this effect simply and on your screen, where it used to take a \$100,000 vertical camera to get almost the same results. Here is an example of the technique. Play with it!

1. Choose a simple, non-ornate text for the background text, and type it in a large size. In the example shown in the following figures, the numeral 50 was set in large, bold, san serif type. Set your text.
2. Fill the text with a 20 or 30 percent grey screen, and no outline.



3. Choose a different shaped and textured typestyle for your foreground type, and type your text. In the example, we used a script font, because the curved, italicized flow of the type contrasted well with the block shape of the background figure.



Try using different types of foreground and/or background type. In this figure the foreground text style is the same, but it is outlined with black and filled with white. The effect is dramatically different.



Finally, you can integrate color into the design. Depending on the color you choose, you may not have to use a percentage fill for the background type. This type is 100 cyan, for example, and doesn't overwhelm the foreground type. If the color you want in the background is bright red, try a 50 percent fill, so it won't be so overwhelming.



--By Rob Weinstein

Write for "the Weekly"

The WOLW is looking for a two or three writers to contribute a weekly column to this publication. Any of the following subjects: Windows, Windows Software Reviews (commercial or shareware), OS/2, Windows Programming, Windows related hardware, DeskTop Publishing, MultiMedia/Sound or Communications.

Qualified individuals will have backgrounds as computer consultants, Lan managers, Windows programmers, desktop publishing, graphic artist, technical support personal or just be knowledgable about Windows etc.. You must either have or be willing to acquire a 9600 modem.

There are **many benefits** to the special status you would have as a writer for Windows Online. If you are a professional and are interested in combining business with pleasure call me at the voice line listed below.

Frank Mahaney - Publisher

"the Weekly" is a product of,

Windows Online, (WOL), an on-line service dedicated to the MS Windows[™] operating environment and to other GUI subjects. WOL prides itself on having the most complete and up-to-date library of Windows shareware in the world. WOL currently has over 3,600 Windows 3.x programs and files for download. Call today for your free trial. Consider membership with the Worlds premeire Windows on-line service.

No reprinting of this document is permitted without the written permission of Windows Online. All articles are copyrighted by their authors. Opinions presented herein are those of the individual author and do not necessarily reflect those of WOL or "the Weekly."

Call Windows Online today!
(510) 736-8343 BBS



Frank J. Mahaney
Publisher and Editor
Windows Online
Box 1614
Danville, CA 94526-6614
(510) 736-4376 Voice/Fax

Copyright 1992, Windows Online