

Windows™/ DOS



Developer's Journal

- > [Author Information](#)
- > [Subscription Information](#)
- > [From the Editor](#)
- > [About *Windows/DOS Developer's Journal*](#)

Select one of the general topics above, or select one of the issues below to view the table of contents for a specific issue.

<u>1992Vol. 3, No. 1 January</u>	<u>Internationalization</u>
<u>1992Vol. 3, No. 2 February</u>	<u>Windows Programming</u>
<u>1992Vol. 3, No. 3 March</u>	<u>Multimedia</u>
<u>1992Vol. 3, No. 4 April</u>	<u>Hardware Manipulation</u>
<u>1992Vol. 3, No. 5 May</u>	<u>Interpreters</u>
<u>1992Vol. 3, No. 6 June</u>	<u>Debugging</u>
<u>1992Vol. 3, No. 7 July</u>	<u>Efficiency</u>
<u>1992Vol. 3, No. 8 August</u>	<u>Graphical User Interfaces</u>

Articles by Ron Burk

January 1992 Foreign Language Pre-Processor And String Externalization Tools
February 1992 A Reusable C++ Chunk Allocator for DOS and Windows
March 1992 Windows 3.1
August 1992 Windows 3 Developer's Workshop

Articles by Timothy Adams

June 1992

Intercepting DLL Function Calls

Articles by Charles B. Allison

April 1992

July 1992

Fast Times with the PC Clock

Using the UART's FIFO Queue

Articles by Arthur D. Applegate

June 1992

Preventing, Detecting, and Fixing Windows Memory Bugs.

Articles by Daniel Appleman

May 1992

Writing Dynamic Link Libraries for Visual Basic

Articles by Dirk S. Baum

March 1992

Using CD-ROM Device Drivers to Play Audio Discs

Articles by Paul Bonneau

<u>January 1992</u>	<u>Windows Questions and Answers</u>
<u>February 1992</u>	<u>Windows Questions and Answers</u>
<u>March 1992</u>	<u>Windows Questions and Answers</u>
<u>April 1992</u>	<u>Windows Questions and Answers</u>
<u>May 1992</u>	<u>Windows Questions and Answers</u>
<u>June 1992</u>	<u>Windows Questions and Answers</u>
<u>July 1992</u>	<u>Windows Questions and Answers</u>
<u>August 1992</u>	<u>Windows Questions and Answers</u>

Articles by Marc Briand

April 1992

.RTPatch

Articles by Frank D. Cavallito

May 1992

Interrupts and Lite Assembly in APL

Articles by John Cole

May 1992

The PCBUS Databus System

Articles by Jerry Cross

March 1992

Playing Live Digital Video Under Windows

Articles by Helen K. Custer

February 1992 [A Reusable C++ Chunk Allocator for DOS and Windows](#)
July 1992 [Inside Windows NT: An Introduction](#)
August 1992 [Inside Windows NT: The Win32 Protected Subsystem](#)

Articles by Guy Eddon

February 1992 OS/2 Power Under DOS

Articles by Peter Gultzan

January 1992 How To Compare Characters With European Collating Sequences

Articles by Karen Hazzah
April 1992 DMA Revealed

Articles by Robert L. Hummel

May 1992

BIOS Identification of Diskette Drives

Articles by Peder Jungck
August 1992 Generating an Icon Editor

Articles by Alex Leavens

January 1992 Using WindowsMaker Professional

February 1992 Building Free-Standing Control Buttons

Articles by Craig Murray

January 1992 DOS Internationalization Support

Articles by Stephen Nebel

August 1992

PUSHA/POPA Emulation for 8088/8086 CPUs

Articles by John G. Nelson

January 1992 Japanese Double-Byte Character Processing

January 1992 Uncovering the NEC-9801 PC

Articles by Chris Newbold

June 1992

Displaying Bitmapped Images In Dialog Boxes

Articles by Kanhom Ng

August 1992

Multi-page dialog boxes

Articles by Daniel Norton

June 1992

Soft-ICE/W from Nu-Mega Technologies

Articles by Tom Olsen

May 1992

Making Windows and DOS Programs Talk

Articles by Doug Overmyer

February 1992 A Font Previewer
August 1992 Colorizing Windows

Articles by Mark Peterson

July 1992

WINMEM: An Efficient Subsegment Memory Allocator For Windows 3.x

Articles by Matt Pietrek

[June 1992](#)

[Writing a Windows Debugger](#)

[July 1992](#)

[A Windows assert\(\) with Symbolic Stack Trace](#)

Articles by David Andrew Price

March 1992

Writing WordPerfect Documents and Merge Files

Articles by Brent Rector

July 1992

Windows WEP Problems

Articles by Dan Saks

February 1992 Programming Style

June 1992 Programming Style

August 1992 Programming Style - Genericity in C++, Part 2

Articles by Richard R. Sands

January 1992 Using Generic Windows API in Pascal

Articles by William Smith

June 1992

For a Consistent User Interface, Center Your Windows

Articles by George Toft

[July 1992](#)

[Fast, Indexed Help](#)

Articles by Victor Volkman

<u>February 1992</u>	<u>Fading-in Custom Controls For Windows 3: Part 1</u>
<u>March 1992</u>	<u>A Fader Custom Control For Windows 3: Part 2</u>
<u>April 1992</u>	<u>The FAX BIOS Standard</u>
<u>July 1992</u>	<u>The Advanced Power Management Specification</u>

Articles by Jonathan Wood

January 1992 Detecting The Display Hardware

Articles by Leor Zolman

<u>January 1992</u>	<u>Tech Tips</u>
<u>February 1992</u>	<u>Tech Tips</u>
<u>March 1992</u>	<u>Tech Tips</u>
<u>April 1992</u>	<u>Tech Tips</u>
<u>May 1992</u>	<u>Tech Tips</u>
<u>June 1992</u>	<u>Tech Tips</u>
<u>July 1992</u>	<u>Tech Tips</u>

About *Windows/DOS Developer's Journal*

Windows/DOS Developer's Journal is the monthly journal for advanced DOS and Windows programmers. We provide solutions to programming problems in a variety of languages, including C++, C, assembly language, Pascal, Visual Basic, BASIC, and others. Our focus is on presenting practical and innovative technical solutions to the programming problems that we all face in creating professional DOS and Windows programs. We favor the concrete over the abstract, and strive for clear and concise writing, in order to help you quickly absorb the technical information you need to help you do your job. Our goal is create a monthly forum for programmers to exchange the very latest and best programming techniques.

Windows/DOS Developer's Journal articles often come from the experienced programmers who make up our audience, rather than a handful of professional writers. We actively solicit manuscripts and article ideas from our readers, and the result is timely articles that reflect the best techniques available -- not just the standard information you can find in the manual. Select the "Author Information" topic below for information on submitting a proposal or manuscript to *Windows/DOS Developer's Journal*. For information on subscribing to *Windows/DOS Developer's Journal*, select "Subscription Information" below. To provide feedback on this online index or communicate with the editor, select "From the Editor".

[Author Information](#)

[Subscription Information](#)

[From the Editor](#)

Author Information

Windows/DOS Developer's Journal actively seeks submissions from its readers. Each issue of the magazine contains a call for papers (check the table of contents for the correct page number) for various upcoming themes. The call for papers lists suggested topics for each theme, but these are mainly intended to stimulate ideas, not to restrict your possibilities. In addition to theme articles, we are also interested in feature articles on a variety of DOS and Windows topics. Feel free to send us a proposal on any ideas you have that you think would be of interest to other programmers.

One good first step for any prospective author is to obtain a copy of our author guidelines. To have a copy of our guidelines mailed to you, contact the Managing Editor at:

Windows/DOS Developer's Journal
1601 West 23rd St., Suite 200
Lawrence, KS 66046-2743
Tel: (913) 841-1631
FAX (913) 841-2624

We prefer exclusive submissions; if yours is a simultaneous submission, please note that in your cover letter. Proposals should include a short abstract, preferably followed by a one-page outline of the article. Including a brief resume of your programming background is also helpful.

If you have a fairly short (less than 1000 words, less than 200 lines of code) programming tip or technique, please consider submitting it as a "Tech Tip". We pay at least \$50 for each tip that we print. Send your submission to:

Tech Tips
Leor Zolman
1601 West 23rd St., Suite 200
Lawrence, KS 66046-2743

or, via email to:

Internet: techtips@rdpub.com
CIS: >INTERNET:techtips@rdpub.com
uucp ...!uunet!rdpub!techtips

[About *Windows/DOS Developer's Journal*](#)
[Subscription Information](#)
[From the Editor](#)

Subscription Information

For the most current subscription information and prices, contact

Windows/DOS Developer's Journal
1601 W. 23rd St., Suite 200
Lawrence, KS 66046-2743
Tel: (913) 841-1631
FAX (913) 841-2624

As of this writing (July 24th, 1992), our subscription prices were as follows:

1 year (12 issues)	\$29
2 years (24 issues)	\$54
3 years (36 issues)	\$77

Please allow up to six weeks for delivery of first issue. Orders outside the US must be prepaid in US funds. CANADA/MEXICO subscriptions are: 1 year - \$53, 2 years - \$88, 3 years - \$121. Overseas subscriptions are: 1 year - \$64, 2 years - \$120, 3 years - \$174.

[About *Windows/DOS Developer's Journal*](#)

[Author Information](#)

[From the Editor](#)

From the Editor

Thank you for obtaining our online index. My name is Ron Burk and I am the editor of *Windows/DOS Developer's Journal*. Although a lot of programming magazine editors like to think they are still programmers, I really am! My time is divided evenly between being a magazine editor and being a software developer. My well-thumbed copies of Duncan's *Advanced MS-DOS* and Petzold's *Windows* tome are testimonies to my search for knowledge that just isn't in the Microsoft manuals. That's the gap that *Windows/DOS Developer's Journal* helps fill.

We have run a slew of useful technical articles in past issues and that's why I created this online index. I got tired of thumbing through issues to find that article that shows how to make a multiline edit control accept carriage returns, or to find which of Paul Bonneau's columns showed how to detect keystrokes in a DOS VM. Now, with a few clicks, I can usually directly locate the exact article I want and then pull it off the shelf.

If you are a subscriber, I hope you find this a useful reference to your back issues. If you are not a subscriber, I hope this index gives you a feel for what our magazine is about. We are a growing magazine, but still small enough that we listen to all the feedback our readers provide. If you have feedback on this online index or the magazine itself, I encourage you to write us a letter. You can write us at the address listed below, or submit your letter electronically by sending it to:

Internet: wdletter@rdpub.com
uucp ...!uunet!rdpub!wdletter
CompuServe >INTERNET:wdletter@rdpub.com

I also enjoy hearing from readers personally, to hear criticism, topic ideas, article proposals, or just that someone found a particular article useful. You can contact me by writing to:

Ron Burk
Windows/DOS Developer's Journal
1601 W. 23rd St., Suite 200
Lawrence, KS 66046-2743

or send me personal electronic mail at:

CompuServe: 70302,2566
BIX: rlburk
uucp: ...!uunet!rdpub!ronb
Internet: ronb@rdpub.com

I reply to all my electronic mail, so if you don't get a reply, it means that either your message or my reply got eaten by the network. Thanks for using our online index and being part of the *Windows/DOS Developer's Journal* community of programmers.

-Ron Burk

[About Windows/DOS Developer's Journal](#)
[Author Information](#)
[Subscription Information](#)

Windows™/DOS

 Developer's Journal

Vol. 3, No. 1 January 1992

Internationalization

- How To Compare Characters With European Collating Sequences** **11**
If you thought DOS code pages handle your European language needs, think again.
Peter Gulutzan
- Japanese Double-Byte Character Processing** **23**
Learn the pitfalls involved in handling double-byte character sets like kanji.
John G. Nelson
- Uncovering the NEC-9801 PC** **35**
Learn about Japan's most popular - and most incompatible - PC.
John G. Nelson
- DOS Internationalization Support** **39**
A quick review of DOS National Language Support, the foundation for Windows and OS/2 internationalization.
Craig Murray

Features

- Using Generic Windows API in Pascal** **43**
Should you skip OWL and access the Windows API directly? Here are the pros and cons.
Richard R. Sands
- Detecting The Display Hardware** **51**
The most complex video detection methods are not always the best.
Jonathan Wood

Product Previews

- Foreign Language Pre-Processor And String Externalization Tools** **61**
Ron Burk

User Reports

- Using WindowsMaker Professional** **67**
A user view of a Windows user interface development tool.
Alex Leavens

Columns

- Windows Q&A** **5**
Paul Bonneau
- Tech Tips** **57**
Leor Zolman

January 1992

Vol. 3, No. 1 Internationalization

Internationalization by Design
Shea Murphy

Abstract

If you are designing a DOS or Windows program, this article can acquaint you with the design issues you should consider in order to produce a program that can easily be adapted for foreign markets.

Windows Questions and Answers

Paul Bonneau

Key words: Windows; dialog boxes; C; icons; cursors; menus

Abstract

Chris B. Turner asks how to use a colored brush for the background of a dialog box class. Paul shows the proper way to do this and includes a little trick for speeding up background erasing.

Niels Erik Holm asks how to display an icon that is not part of your .exe. To answer this, Paul shows the internal structure of a Windows icon. This is not documented by Microsoft, but it is the same in both Windows 3.0 and Windows 3.1:

```
typedef struct ICN
{
    POINT ptHot; /* hotspot */
    int dx, dy; /* width, height */
    WORD cbLine; /* width of mono bitmap */
    BYTE cpln; /* # of color planes */
    BYTE cbit; /* bits per plane */
    BYTE rgb[1]; /* mono/color bitmaps */
} ICN;
```

Deenar Toraskar asks how to use function keys as accelerators in modal dialog boxes, since modal dialog boxes have their own message loop that does not translate accelerator keys. Paul provides a reusable function, AcceleratedDialog(), to handle this problem.

Charles H. Roth asks a variety of questions about speed and size versus memory models under Windows. Paul's reply also indicates how to keep edit controls from using up your local heap.

Many different people have asked how to change the desktop cursor. The answer is:

```
SetClassWord(GetDesktopWindow(), GCW_HCURSOR, LoadCursor(NULL, IDC_SIZE));
```

This sets the desktop cursor to the four-headed resize cursor -- you would substitute a handle to whatever cursor you wanted.

Finally, Paul looks into how you can capture the image of each menu in an application (for example, to use in technical documentation). Paul provides a function to return the handle of a popup menu window, which you could then use as the source of a BitBlt().

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)

May 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992 [Windows Questions and Answers](#)

How To Compare Characters With European Collating Sequences **Peter Gultzan**

Key words: C; assembly language; DOS; internationalization; code pages

Abstract

Microsoft's National Language Support solves some, but not all of the problems of handling foreign languages. This article demonstrates the problems involved in trying to write software for the European market and reveals solutions for dealing with comparisons in European character sets. The code includes a small assembly language program to demonstrate using DOS's collating sequence tables and a basic C translation routine, for translating strings to their collating equivalents.

The author provides an invaluable list of problem characters for the following languages: Albanian, Croatian, Czech, Danish, Dutch, Esperanto, Estonian, Finnish, German, Hungarian, Icelandic, Latvian, Lithuanian, Norwegian, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish, Turkish, and Welsh.

January 1992

Vol. 3, No. 1 Internationalization

Japanese Double-Byte Character Processing **John G. Nelson**

Key words: C; DOS; internationalization; kanji; NEC 9801 PC; Japan

Abstract

One major obstacle to adapting your software to the Japanese market is the double-byte character set. Most code that deals with characters, both in strings and on the screen, has to change. This article covers the problems and solutions involved in handling double-byte character sets. The code includes a function to translate from Shift-JIS to JIS and functions to correctly handle extracting and inserting characters with double-byte character strings. The code also handles detecting the presence of a machine that supports the Japanese AX architecture.

Other Articles by John G. Nelson

January 1992 [Uncovering the NEC-9801 PC](#)

January 1992

Vol. 3, No. 1 Internationalization

Uncovering the NEC-9801 PC
John G. Nelson

Key words: DOS; text-mode video; graphics-mode video; hardware; internationalization; NEC 9801 PC; Japan

Abstract

Unlike many foreign markets, the Japanese market consists of a mix of PC software and hardware environments that contain significant incompatibilities. Worse, the most popular PC in Japan is also the least compatible with the standard American PC. This article acquaints you with the various versions of DOS and hardware standards that make up the Japanese PC market and points out some of the problems you have to avoid if you want to port to the NEC-9801 PC. Understanding the incompatibilities can help you make design decisions that produce software that ports more easily to the Japanese market.

Other Articles by John G. Nelson

January 1992 [Japanese Double-Byte Character Processing](#)

January 1992

Vol. 3, No. 1 Internationalization

DOS Internationalization Support

Craig Murray

Key words: DOS; internationalization

Abstract

DOS functions in a wide variety of different countries and languages. DOS provides a variety of features to help you write international applications. This article looks at what DOS can and can't help you with as you design an international application.

January 1992

Vol. 3, No. 1 Internationalization

Using Generic Windows API in Pascal

Richard R. Sands

Key words: Windows; Turbo Pascal for Windows; ObjectWindows Library

Abstract

Turbo Pascal for Windows comes with its own object-oriented interface to Windows: Object Windows Library (OWL). You can, however, ignore OWL and directly access the Windows API. The result looks very much like C Windows programs. This article looks at the pros and cons of directly accessing the Windows API from TPW and presents a sample program, BYE, which gives you a quick exit from Windows.

January 1992

Vol. 3, No. 1 Internationalization

Detecting The Display Hardware **Jonathan Wood**

Key words: C; assembly language; DOS; BIOS; text-mode video

Abstract

Nearly all high-quality text mode DOS programs write directly to the screen. Unfortunately, the need to handle a variety of video adapters sometimes leads to lowest-common-denominator code -- code that cannot exploit color capabilities or is unnecessarily slow to avoid snow problems. This article presents a well-tested method for reliably detecting the type of video monitor installed and reporting its characteristics. The result is a small assembly language function that uses BIOS functions rather than the trickier (and sometimes incorrect) methods that directly access the video hardware.

Tech Tips
Leor Zolman

Key words: C; assembly language; DOS; serial communications; NEC V20; modems

Abstract

How to Really Defeat Call Waiting

Dan Goldberg discusses how to solve the problem that call waiting presents to modems. The beep that call waiting uses can cause a loss of carrier, with disastrous effects to any serial communications session you have going on the line. In some areas, you can send a certain key sequence that suspends call waiting for the duration of the current phone call. Unfortunately, that service is not available everywhere or doesn't work properly. Dan shows how to configure a Hayes-compatible modem to ignore the .75-second loss of carrier caused by the call waiting beep.

Reliable NEC V20 Detection

Anthony V Ingenoso supplies a simple Turbo C program to detect the presence of a NEC V20 CPU. The function depends on the fact that the NEC responds differently than the 80x86 to a variation of the AAD instruction.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

March 1992 [Tech Tips](#)

May 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Foreign Language Pre-Processor And String Externalization Tools **Ron Burk**

Key words: C; DOS; internationalization; kanji

Abstract

Probably the biggest job you face in internationalizing software is getting the strings out. Every string constant in your program is a candidate for translation to the language of the target country. This article examines two software packages from Network Dynamics designed to eliminate the drudgery of getting the strings out of C programs.

The String Externalization Tools package provides two programs: one for extracting strings from your C programs and another for loading in the strings when your program starts up. The program provides kanji support and a flexible system that lets you control which strings get extracted and allows you to add new strings and then run the extract program again, without affecting previous work.

The Foreign Language Pre-Processor allows you to maintain a database of different string translations for a single program. You then generate the correct source code for the desired language from this database. This method produces a separate, language-specific, .exe for each target country.

Other Articles by Ron Burk

February 1992 A Reusable C++ Chunk Allocator for DOS and Windows

August 1992 Windows 3 Developer's Workshop

March 1992 Windows 3.1

January 1992

Vol. 3, No. 1 Internationalization

Using WindowsMaker Professional **Alex Leavens**

Key words: Windows; C; user Interface; CASE

Abstract

WindowsMaker/Pro is a Windows development tool that allows you to interactively design your C application's user interface. Alex looks at the new features this upgrade offers, including more convenient source code regeneration and a new animation test mode.

Other Articles by Alex Leavens

February 1992 [Building Free-Standing Control Buttons](#)

Windows™/DOS

Developer's Journal

Vol. 3, No. 2

February 1992

Windows Programming

A Reusable C++ Chunk Allocator for DOS and Windows 5
Plug this chunk allocator into all your classes for a 100 percent speedup.
Ron Burk and Helen K. Custer

Fading-in Custom Controls For Windows 3: Part I 14
Learn about custom controls with this fader control.
Victor Volkman

A Font Previewer 27
Let the user pick from a plethora of fonts -- visually.
Doug Overmyer

Building Free-Standing Control Buttons 53
The ins and outs of coding custom buttons in Windows.
Alex Leavens

Features

OS/2 Power Under DOS 49
Combine Phar Lap's 286 DOS Extender and the Microsoft C OS/2 libraries to access the OS/2 API from DOS!
Guy Eddon

Columns

Windows Q&A 41
Paul Bonneau

Programming Style 63
More About Set Types
Dan Saks

Tech Tips 70
Leor Zolman

February 1992

Vol. 3, No. 2 Windows Programming

A Reusable C++ Chunk Allocator for DOS and Windows **Ron Burk and Helen K. Custer**

Key words: C++; DOS; memory management

Abstract

A simple, well-known method for speeding up C++ memory management is with a chunk allocator. A chunk allocator allocates a chunk of several fixed-size objects at a time, then doles out each object in the chunk as requested. This reduces the number of calls to the more general-purpose, but slower new and delete. The problem with chunk allocators is that you have to write one for each class, and each class derived from such a class. This article shows you how to write a single chunk allocator that handles all your classes and typically cuts your memory management overhead in half.

Other Articles by Ron Burk

January 1992 Foreign Language Pre-Processor And String Externalization Tools

August 1992 Windows 3 Developer's Workshop

March 1992 Windows 3.1

Other Articles by Helen K. Custer

July 1992 Inside Windows NT: An Introduction

August 1992 Inside Windows NT: The Win32 Protected Subsystem

February 1992

Vol. 3, No. 2 Windows Programming

Fading-in Custom Controls For Windows 3: Part I **Victor Volkman**

Key words: Windows; C; custom controls

Abstract

Custom controls allow you to provide structured extensions to the standard Windows user interface tools. In this two-part article, Victor demonstrates the principles involved in constructing custom controls, using a slide control (like the fader slide switch on a stereo) as an example.

Other Articles by Victor Volkman

April 1992 [The FAX BIOS Standard](#)

March 1992 [A Fader Custom Control For Windows 3: Part 2](#)

July 1992 [The Advanced Power Management Specification](#)

February 1992

Vol. 3, No. 2 Windows Programming

A Font Previewer **Doug Overmyer**

Key words: Windows; fonts; Turbo Pascal for Windows; ObjectWindows Library

Abstract

The number of fonts available in Windows can be quite large, especially if you are using Adobe Type Manager. If your application allows the user to select from the available fonts, a simple listbox may be insufficient -- it's just too hard to remember what a font looks like from the sometimes-cryptic font names. This application demonstrates how to write a font previewer in Turbo Pascal for Windows. The utility allows you to quickly view any available font.

Other Articles by Doug Overmyer

August 1992 Colorizing Windows

Windows Questions and Answers

Paul Bonneau

Key words: Windows; hypertext; cursors; edit controls; comboboxes

Abstract

David Lee inquires if anyone knows of a "sleazy hack" to determine if a combo box's listbox is down. In lieu of any cleaner solution to this problem, Paul provides the required sleazy hack for Windows 3.0; fortunately, Windows 3.1 provides a notification message for this case.

Stephen Chung asks why his custom edit control does not receive WM_CHAR messages inside of a dialog box. Paul explains the problem and points out that you have to respond to the WM_GETDLGCODE message with DLGC_WANTCHARS.

How do you display an hourglass cursor when a modal dialog box is present? Sounds easy, but as usual, there are some gotchas. Paul provides a MySetCursor() function to finesse the problem.

Amari Elammari writes to ask why he can't create a caption for an edit child control with the following code:

```
hwnedit = CreateWindow("edit", "edit",
    WS_CHILD | WS_VISIBLE | WS_OVERLAPPEDWINDOW
    | WS_CAPTION | WS_HSCROLL | WS_VSCROLL
    | WS_OVERLAPPED | ES_LEFT | ES_MULTILINE
    | ES_AUTOHSCROLL | ES_AUTOVSCROLL,
    0,0,0,0, hWnd, 1, hInstance, NULL);
```

Paul investigates and discovers that the edit control explicitly removes the WS_BORDER style from the window and, since a caption only gets created when both WS_BORDER and WS_DLGFAME are on, you get no caption. Paul speculates that this was a crude solution, on the part of the edit control, to the problem of whether a WM_SETTEXT would refer to the edit control's text or the edit control's caption -- if you can't create a caption, there's no ambiguity. The column contains some listings to provide captioned edit controls by creating a pair of windows to handle the edit control and the caption text.

Gil Romano is incorporating some hypertext capabilities in his Windows program and wants to know what strategy is best. Paul discusses the problems and provides sample code to implement a transparent window that handles detecting a button hit on "hot text" that lies beneath the window.

Other Articles by Paul Bonneau

May 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992 [Windows Questions and Answers](#)

February 1992

Vol. 3, No. 2 Windows Programming

OS/2 Power Under DOS

Guy Eddon

Key words: C; DOS; OS/2; DOS extenders

Abstract

OS/2 offers some important advantages over DOS, including multi-threading and access to extended memory. What you may not realize is that, if you have Phar Lap's 286 DOS Extender and the OS/2 libraries from Microsoft C, you can write DOS programs that access many OS/2 API functions. This article shows you how and lists the functions that you can use.

February 1992

Vol. 3, No. 2 Windows Programming

Building Free-Standing Control Buttons **Alex Leavens**

Key words: Windows; C; user Interface; buttons

Abstract

Windows makes it easy to use buttons in your window or dialog box. What if you want something besides simple text on a button? This article shows you how to make custom buttons, displaying your own bitmap on the button face instead of text.

Other Articles by Alex Leavens

January 1992 [Using WindowsMaker Professional](#)

February 1992

Vol. 3, No. 2 Windows Programming

Programming Style

Dan Saks

Key words: Pascal; Modula-2; sets

Abstract

Following up on the December Programming Style column about the built-in set types of Pascal and Modula-2, Dan shows how these two languages implement sets and set operations. Understanding this can help you implement your own set operations in other languages.

Other Articles by Dan Saks

June 1992 [Programming Style](#)

August 1992 [Programming Style - Genericity in C++, Part 2](#)

February 1992

Vol. 3, No. 2 Windows Programming

Tech Tips
Leor Zolman

Key words: C; assembly language; DOS; OS/2; BASIC

Abstract

Mousing Around With QuickBASIC

Noel Nyman shows how to use QuickBASIC's CALL INTERRUPT (documented in Phil Weber's August 1991 article) to handle the mouse from QuickBASIC.

Accessing I/O Ports Under OS/2 Protected Mode

Donna Campanella relates the trials and tribulations of trying to perform direct I/O to the keyboard under OS/2. The included code shows how to place code that does device I/O in a segment with the correct IOPL (I/O privilege level).

Other Articles by Leor Zolman

January 1992 [Tech Tips](#)

March 1992 [Tech Tips](#)


May 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Windows™/DOS

 Developer's Journal

Vol. 3, No. 3

March 1992

Multimedia

Using CD-ROM Device Drivers to Play Audio Discs 22

A complete application for controlling audio CDs via your CD-ROM drive.
Dirk S. Baum

Playing Live Digital Video Under Windows 39

An application that demonstrates Intel's new Audio/Video Kernel for Digital Video Interactive (DVI).

Jerry Cross

Features

A Fader Custom Control For Windows 3: Part 2 5

Victor wraps up with the necessary dialog editor interface functions.
Victor Volkman

Writing WordPerfect Documents and Merge Files 49

Add WordPerfect support to your application with this reusable code.
David Andrew Price

Product Previews

Windows 3.1 53

An overview of the new Windows 3.1 API.

Ron Burk

Columns

Windows Q&A 17

Patching Another MLE Bug.

Paul Bonneau

Tech Tips 59

Leor Zolman

March 1992

Vol. 3, No. 3 Multimedia

A Fader Custom Control For Windows 3: Part 2 **Victor Volkman**

Key words: **Windows; C; custom controls**

Abstract

Custom controls allow you to provide structured extensions to the standard Windows user interface tools. In this two-part article, Victor demonstrates the principles involved in constructing custom controls, using a slide control (like the fader slide switch on a stereo) as an example.

Other Articles by Victor Volkman

February 1992 [Fading-in Custom Controls For Windows 3: Part I](#)

April 1992 [The FAX BIOS Standard](#)

July 1992 [The Advanced Power Management Specification](#)

March 1992

Vol. 3, No. 3 Multimedia

Windows Questions and Answers **Paul Bonnaeu**

Key words: Windows; C; serial communications; debugging; edit controls

Abstract

Shalom Halevy writes about a problem with multiline edit controls (MLEs) created with the ES_CENTER style. If the user enters a line of text the width of the control, then enters a single space character, either Windows encounters a UAE, or white horizontal bands the entire width of the display appear! Paul tracks down the problem (down to the very instruction), caused by two separate bugs in Windows, and then provides a solution.

Stephen Baker writes to ask for information about the kernel debugger, which Paul often uses to track down nasty bugs. Paul replies with comments about the various debuggers and his own debugger preferences.

Moshe Rubin of Triton Software Ltd. writes about a bug encountered when using full duplex with the Windows COM driver. Not being an expert on COMM.DRV, Paul cites John Loram of Bio-Engineering Research Labs, who claims that there is indeed a bug and the third-party driver, TurboComm, fixes it (and also supports the 16550 UART for more reliable higher-speed serial communications).

Other Articles by Paul Bonnaeu

February 1992 [Windows Questions and Answers](#)

May 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992 [Windows Questions and Answers](#)

March 1992

Vol. 3, No. 3 Multimedia

Using CD-ROM Device Drivers to Play Audio Discs

Dirk S. Baum

Key words: C; assembly language; DOS; device driver; CD-ROM; sound; multimedia

Abstract

The Windows multimedia extensions handle a lot of the work of dealing with CD-ROMs. DOS programs are still on their own, however. This article covers the essentials of CD-ROM access by demonstrating how to play audio tracks on your CD-ROM drive. The code includes a complete application for controlling audio CDs, written mostly in assembly language.

March 1992

Vol. 3, No. 3 Multimedia

Playing Live Digital Video Under Windows

Jerry Cross

Key words: Windows; C; hardware; multimedia; Digital Video Interactive

Abstract

An important part of multimedia is the ability to incorporate live video (from a television signal or VCR, for example). Intel is setting a new standard in this area with their Audio/Video Kernel (AVK). This article covers the capabilities of the AVK by developing a Windows application that displays live video in a window.

March 1992

Vol. 3, No. 3 Multimedia

Writing WordPerfect Documents and Merge Files **David Andrew Price**

Key words: Pascal; DOS; file formats

Abstract

For many applications, understanding the file formats of a standard word processor can be a plus. This article provides a set of routines that allow Turbo Pascal programs to write WordPerfect v5.0 files, readable by both WordPerfect v5.0 and WordPerfect v5.1.

March 1992

Vol. 3, No. 3 Multimedia

Windows 3.1

Ron Burk

Key words: **Windows; Windows 3.1**

Abstract

After a long beta involving thousands of programmers and users, Windows 3.1 is ready to ship. This article summarizes the new features of Windows 3.1, including OLE, DDEML, File Manager, TrueType fonts, the common dialogs, multimedia, toolhelp.dll and more. A sidebar covers the big picture, explaining what Windows NT, Win32, and other new developments mean for the developer.

Other Articles by Ron Burk

January 1992 [Foreign Language Pre-Processor And String Externalization Tools](#)

February 1992 [A Reusable C++ Chunk Allocator for DOS and Windows](#)

August 1992 [Windows 3 Developer's Workshop](#)

Tech Tips
Leor Zolman

Key words: Windows; C; assembly language; DOS; Visual Basic; I/O redirection

Abstract

Redirection Autodetection

Some programs require command-line I/O redirection to function as intended; others cannot function correctly under such redirection. In both cases, a method to detect that one of the standard I/O handles has been redirected is needed. Daniel Kian McKiernan provides both a C and assembly language routine to detect the redirection status of the five standard file handles under DOS.

Visual Basic: Load Time Problem Solution

When you launch a Visual Basic application, the mouse cursor changes to an hourglass. Unfortunately, the cursor reverts back to an arrow, long before VB finishes loading the application. If you are using a big form with a lot of objects, this delay can be quite pronounced and leave the user with the feeling that your application just isn't going to start. Al Meadows provides a small Startup form to solve this problem. You can use this form to display a message while VB loads the rest of your application.

A Text-Mode Busy Cursor

Windows etiquette calls for you to change the cursor to an hourglass whenever your application is busy and unable to respond to user input. Steven Van Dyke supplies the DOS text-mode equivalent, a C program that produces a spinning cursor to let the user know your program really isn't hung.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

January 1992 [Tech Tips](#)

May 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Windows™/DOS

Developer's Journal

Vol. 3, No. 4

April 1992

Hardware Manipulation

DMA Revealed

5

Learn the complexities of DMA transfers in DOS, Windows, and OS/2, on a ISA or Microchannel bus.

Karen Hazzah

Fast Times with the PC Clock

37

Obtain high-resolution time stamps with a little hardware manipulation.

Charles B. Allison

Product Previews

The FAX BIOS Standard

21

A new specification to provide standard access to FAX boards from DOS and Windows.

Victor Volkman

Product Spotlight

.RTPatch

63

A patch program that makes software upgrades a breeze.

Marc Briand

Columns

Windows Q&A

49

A W4W Conundrum; Vertical Scrollbars in Menus.

Paul Bonneau

Tech Tips

61

Leor Zolman

April 1992

Vol. 3, No. 4 Hardware Manipulation

DMA Revealed

Karen Hazzah

Key words: Windows; DOS; device driver; DMA; OS/2; MicroChannel; hardware; Virtual DMA Services

Abstract

This article is a practical guide to programming the Direct Memory Access (DMA) controller for the PC, AT, and Microchannel architectures in the DOS, Windows and OS/2 environments. The relatively simple operation of the original PC DMA controller has been complicated by the introduction of protected mode and multitasking environments. This article provides example code that demonstrates how to program the DMA in these situations.

April 1992

Vol. 3, No. 4 Hardware Manipulation

The FAX BIOS Standard **Victor Volkman**

Key words: Windows; DOS; FAX; specifications

Abstract

As FAX boards become cheaper, more and more applications are supporting them. The ability to FAX data is becoming as ubiquitous as the ability to send data to a printer. As with printers, the problem is that each FAX board has its own distinct interface, creating a headache for developers.

The FaxBios Specification is a new, hardware-independent API for accessing FAX boards from DOS or Windows. The DOS interface is via a TSR and the Windows interface is via DDE. This article provides a detailed, technical look at the capabilities of this new standard for accessing FAX hardware.

Other Articles by Victor Volkman

February 1992 [Fading-in Custom Controls For Windows 3: Part 1](#)

March 1992 [A Fader Custom Control For Windows 3: Part 2](#)

July 1992 [The Advanced Power Management Specification](#)

April 1992

Vol. 3, No. 4 Hardware Manipulation

Fast Times with the PC Clock
Charles B. Allison

Key words: C; DOS; timers; clock; 8253 counter; hardware; interrupts

Abstract

The DOS timer tick is so slow that it does not lend itself to high-resolution applications. This article shows you how to obtain a high-resolution time stamp by reading the timer chip's registers directly.

Other Articles by Charles B. Allison

July 1992 [Using the UART's FIFO Queue](#)

April 1992

Vol. 3, No. 4 Hardware Manipulation

Windows Questions and Answers **Paul Bonneau**

Key words: Windows; C; user Interface; menus; listboxes

Abstract

A W4W Conundrum

In his December 1991 column, Paul tracked down a multi-line edit control (MLE) bug in Windows and supplied a DLL that fixes the problem. When Paul installed version 2.0 of Word for Windows, he discovered that his DLL causes it to crash! The problem lay in the non-standard way W4W takes over the standard edit control. Paul traces the problem and provides a new version of DLL that still fixes the MLE bug, but does not crash when W4W does its nasty business.

Shivram Venkatasubramaniam writes to ask how to set up scroll bars for a menu that contains many items. Paul looks into some possible solutions and provides the code for one such solution: moving a child listbox control under the top -level menu item when it is activated.

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)

May 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992 [Windows Questions and Answers](#)

April 1992

Vol. 3, No. 4 Hardware Manipulation

Tech Tips
Leor Zolman

Key words: Windows; subclassing; assembly language; print screen

Abstract

Re-Enabling PrintScreen

Homer Tilton contributes some assembly language to re-enable the PrintScreen key and view the current state of the PrintScreen key (enabled or disabled).

Extending WndExtra Data of Existing Controls in Windows

Subclassing standard controls is an easy way to extend the Windows user interface. Unfortunately, you often want to attach your own data to the control and using Windows properties is slow and inconvenient. Gregory C. Peters shows how to register your subclassed control with the extra data bytes you need and conveniently access the resulting storage.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

January 1992 [Tech Tips](#)

March 1992 [Tech Tips](#)

May 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

.RTPatch
Marc Briand

Key words: **DOS; data compression**

Abstract

.RTPatch is a product that aims to change the way programmers distribute software upgrades. Rather than sending out a complete new version of your product, .RTPatch allows you to send a compressed "patch file" that contains only the information that changed from one version to the next. Customers then use this patch file along with a small patch program to update their old version of your software.

One advantage of this product is that it can greatly reduce the number of floppies required to distribute an upgrade (or a new beta to beta testers). Also, since the patch file is useless to anyone who does not own a previous version of your product, you can safely distribute upgrades electronically.

Windows™/DOS

Developer's Journal

Vol. 3, No. 5

May 1992

Interpreters

Interrupts and Lite Assembly in APL

5

Use your favorite debugger to access assembly language from APL.

Frank D. Cavallito

The PCBUS Databus System

16

A look at the design of the Databus interpreter.

John Cole

Features

Making Windows and DOS Programs Talk

21

The Windows virtual device driver provides the most elegant solution for making Windows talk to DOS.

Tom Olsen

Writing Dynamic Link Libraries for Visual Basic

35

Learn some tricks of the trade for writing DLLs tailored for Visual Basic programs.

Daniel Appleman

BIOS Identification of Diskette Drives

45

Reusable code, callable from any language, for determining floppy drive types.

Robert L. Hummel

Columns

Windows Q&A

53

Paul looks at the pitfalls of floating-point exceptions for Windows programs.

Paul Bonneau

Tech Tips

65

Leor Zolman

May 1992

Vol. 3, No. 5 Interpreters

Interfacing to Personal REXX
Charles Daney

Abstract

May 1992

Vol. 3, No. 5 Interpreters

An Interpretive Screen Builder System
Steven Van Dyke

Abstract

May 1992

Vol. 3, No. 5 Interpreters

NTH - the Not-To-Hard Thread Interpreter
Ross Barnes

Abstract

May 1992

Vol. 3, No. 5 Interpreters

MUMPS on PCs: An Overview
Richard F. Walters

Abstract

May 1992

Vol. 3, No. 5 Interpreters

TCL - A Threaded Command Language
Nabajyoti Barkakati

Abstract

May 1992

Vol. 3, No. 5 Interpreters

Interrupts and Lite Assembly in APL

Frank D. Cavallito

Key words: assembly language; DOS; APL

Abstract

Most PC implementations of APL provide some way to access the BIOS and DOS interrupts. This article shows how to use these features and also how to use the "assemble" option of a debugger to code and execute small assembly language routines for maximum speed.

May 1992

Vol. 3, No. 5 Interpreters

The PCBUS Databus System

John Cole

Key words: interpreters; database

Abstract

Databus was originally designed as an interpreter for Datapoint corporation. This language provided BASIC-like features and database services, all in 16Kb of memory. This article looks at the language and its implementation.

Making Windows and DOS Programs Talk

Tom Olsen

Key words: Windows; C; assembly language; DOS; TSR; VxDs; interrupts

Abstract

Windows 3.0 has resulted in large demand for Windows versions of existing DOS programs. Unfortunately, for some programs, moving to Windows is not so much a porting problem as a total rewrite. For that reason, it is important to understand the option of placing a Windows wrapper on an existing DOS program.

This article examines three potential approaches to making DOS and Windows programs communicate. The simplest is to use the Windows clipboard for communication. A more flexible approach is to write a DOS TSR that can pass arbitrary information between a Windows program and any DOS Virtual Machine. The most powerful and elegant approach, however, is to use a Windows VxD (virtual device driver) to provide communications between DOS and Windows; both DOS and Windows programs invoke the VxD functions via a software interrupt.

This article provides useful source code for all three approaches.

May 1992

Vol. 3, No. 5 Interpreters

Writing Dynamic Link Libraries for Visual Basic **Daniel Appleman**

Key words: Windows; C; Visual Basic; DLLs

Abstract

Visual Basic is one of the most productive tools for constructing user interfaces. However, many programmers will want to meld the flexibility of Visual Basic with the speed of compiled DLLs. This article tells you everything you need to know about writing DLLs that you can easily access from Visual Basic.

May 1992

Vol. 3, No. 5 Interpreters

BIOS Identification of Diskette Drives

Robert L. Hummel

Key words: assembly language; DOS; BIOS; hardware; floppy drive

Abstract

How do you determine the capabilities of the current floppy drive? This article provides the answer in the form of an assembly language routine that handles all the special cases to correctly return the drive type. The code is callable from assembly language, BASIC, and C.

May 1992

Vol. 3, No. 5 Interpreters

Windows Questions and Answers **Paul Bonneau**

Key words: Windows; C; fonts; 8087; floating point; user Interface; listboxes

Abstract

Why Default?

Michael Dehoney asks why Paul puts his default cases at the beginning of his switch statement. Paul likes to put the default case first in order to make sure the switch always takes some branch. C does not require the default case to appear last in a switch statement.

Word Processing Conversions

Steve Gaetjens writes to ask if the Windows filter file (.FLT) used by Word for Windows and Lotus' Smarttext is a documented standard, and how to use the Word for Windows import DLLs. Paul points him to Aldus for the windows graphics import filter library standard. The "Microsoft WORD for Windows and OS/2 Technical Reference" documents the DLLs that W4W uses to import various text formats.

Trapping Floating-Point Exceptions

Gian Camillo Vezzoli asks how to trap floating-point exceptions in Windows 3.0, having been unable to write an alternative matherr() function that works. The problem turns out to be complex.

matherr() is only called to handle exceptions that occur inside the runtime floating-point routines. If a floating-point exception occurs outside of these routines, win87em.dll is called to handle the exception, but it does not call your matherr() routine. Instead, win87em.dll records the event, and the next time you call one of the C runtime floating-point routines, it generates a UAE. The problem is complicated by how Windows handles the FPU during a task switch. Paul delves in to problems Windows raises and provides some code to help deal with them.

Fixed Fonts in Listboxes

Shuchi Grover writes to ask how he can use spaces to align columns in a particular listbox in his application. The answer is a quick function call to use the fixed-pitch Courier font via a WM_SETFONT on the listbox.

How To Scroll Windows?

Shravan Arra writes to ask for general tips on scrolling windows that contain both text and bitmaps in them. Paul provides some strategies and some code to demonstrate them.

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992 Windows Questions and Answers

May 1992

Vol. 3, No. 5 Interpreters

Tech Tips **Leor Zolman**

Key words: Windows; C; assembly language; DOS; timers; clock

Abstract

C Macros for Code Performance Measurement

Robert Radcliffe demonstrates some preprocessor macros useful for timing sections of code. The code uses only standard C functions and works with Microsoft and Borland C compilers.

Early Calls to the Window Function

Yishai Sered writes to discuss the problem that your Windows window function receives window messages before CreateWindow () actually has returned a window handle to your program.

The Vampire Meets the Viper

Murray L. Lesser writes to add his comments to a previous piece on the DOS clock and the midnight rollover problem.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

January 1992 [Tech Tips](#)


March 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Windows™/DOS

 Developer's Journal

Vol. 3, No. 6

June 1992

Debugging

Writing a Windows Debugger

6

An inside look at the problems of designing a Windows debugger.

Matt Pietrek

Preventing, Detecting, and Fixing Windows Memory Bugs.

13

Techniques for avoiding the unique memory management problems that arise in Windows programs.

Arthur D. Applegate

Intercepting DLL Function Calls

22

Write your own SPY utility that intercepts calls from any application to any DLL!

Timothy Adams

Features

Displaying Bitmapped Images In Dialog Boxes

34

Dress your dialog boxes up with bitmapped images.

Chris Newbold

For a Consistent User Interface, Center Your Windows

40

Center your windows, for that extra touch of convenience.

William Smith

User Reports

Soft-ICE/W from Nu-Mega Technologies

47

The author of "Writing Windows Device Drivers" looks at a debugger for hard-core systems programmers.

Daniel Norton

Columns

Windows Q&A

28

Windows 3.1's internal window structure, EditItem tricks, tabbing, linear memory locations, and more.

Paul Bonneau

Programming Style

51

A new look at a past topic: genericity.

Dan Saks

Tech Tips

58

Leor Zolman

June 1992

Vol. 3, No. 6 Debugging

Debugging with macros
William Smith

Abstract

Other Articles by William Smith

June 1992 For a Consistent User Interface, Center Your Windows

June 1992

Vol. 3, No. 6 Debugging

Writing a Windows Debugger **Matt Pietrek**

Key words: Windows; debugging; toolhelp.dll

Abstract

In spite of the fact that Windows is basically a graphical DOS extender, writing a Windows debugger poses a different set of problems than writing a DOS debugger. The problems include how to load the debuggee without executing it, how to translate logical addresses to physical addresses, how to maintain breakpoints in the debuggee's code despite the fact that Windows could discard the code segment at any time, and more. In this article, Matt Pietrek looks at the unique design problems that Windows debugging poses and shows some possible solutions. The article shows how Windows 3.1's toolhelp.dll provides a tool for debugging that is documented, but less convenient than the old, undocumented windebug.dll.

Other Articles by Matt Pietrek

July 1992 [A Windows assert\(\) with Symbolic Stack Trace](#)

June 1992

Vol. 3, No. 6 Debugging

**Preventing, Detecting, and Fixing Windows Memory Bugs.
Arthur D. Applegate**

Key words: Windows; C; debugging; memory management; DLLs

Abstract

Memory management bugs have always been a problem in languages that require explicit memory management, such as C and C++. Windows, however, has its own unique set of pitfalls. This article looks at the memory management bugs that are unique to Windows and tells you how to prevent them, find them, and fix them.

Intercepting DLL Function Calls

Timothy Adams

Key words: Windows; C; assembly language; debugging; DLLs

Abstract

The Windows API provides functions that allow anyone to write debugging utilities that intercept window messages. Thus, anyone can write their own program to spy on windows or DDE conversations. Windows does not, however, directly support intercepting DLL calls. Although Petzold shows how you can write a debugging DLL you can link with your program to intercept calls to a specific library, that method does not allow on-the-fly debugging, nor does it allow you to debug other application's calls to arbitrary DLLs.

This article demonstrates a simple technique for intercepting DLL calls. The trick is to obtain a read/write selector to the segment containing the DLL function you wish to intercept. You can then insert a call to your own debugging function. The article includes a complete code example.

Windows Questions and Answers **Paul Bonneau**

Key words: **Windows; C; protected mode; toolhelp.dll; DLLs; Windows 3.1; comboboxes; undocumented**

Abstract

Subclassing a ComboBox's EditItem Window

Richard L. Rosenheim writes to ask for more details on how to superclass the EditItem portion of a ComboBox. He wants to detect double clicks and achieve other effects. Paul provides LpfnSubclassComboEdit() to make it easy to hook into the desired child window.

A reader asks how to access the physical memory location CA00:0000 from within a protected-mode Windows application. Fortunately, the kernel exports nine symbols for direct memory access, so the answer is as easy as:

```
extern WORD _C000H;  
LPSTR lpb = (LPSTR)MAKELONG(0xA000, (WORD)&_C000H);
```

Tabbing Trouble

Jeff Fulton writes about a problem he has getting tabbing to work in his dialog box (modelled after Petzold's hexcalc example); the TAB key does not move the focus along as expected. Paul points out how to call IsDialogMessage() to achieve the desired effect.

Internal Windows Structures

Several people have asked about how Windows internal window structure has changed under Windows 3.1. Paul provides the latest information he has gleaned on this undocumented internal Windows structure.

DLL Loading/Unloading

Samuel R. Blackburn writes to inquire how Windows knows to load and unload DLLs that various programs use. Paul explains the process and points to the new functionality in toolhelp.dll (supplied with Windows 3.1) that allows you to poke around in this area.

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)
May 1992 [Windows Questions and Answers](#)
January 1992 [Windows Questions and Answers](#)
March 1992 [Windows Questions and Answers](#)
April 1992 [Windows Questions and Answers](#)
July 1992 [Windows Questions and Answers](#)
August 1992 [Windows Questions and Answers](#)

June 1992

Vol. 3, No. 6 Debugging

Displaying Bitmapped Images In Dialog Boxes

Chris Newbold

Key words: Windows; dialog boxes; C; bitmaps

Abstract

Windows 3.0 does not provide an easy way to include bitmaps in your dialog boxes. This article shows you how to create dialog boxes with bitmaps on them, using any resource editor and a few extra function calls.

June 1992

Vol. 3, No. 6 Debugging

For a Consistent User Interface, Center Your Windows **William Smith**

Key words: Windows; C; user Interface

Abstract

In many situations, centering a child window or dialog box when it pops up makes for a more pleasing user interface. This sounds easy, but you have to consider more than one case. What if the child window is bigger than the parent? This article supplies simple functions that make it easy to center your windows correctly in all circumstances.

June 1992

Vol. 3, No. 6 Debugging

Soft-ICE/W from Nu-Mega Technologies
Daniel Norton

Key words: Windows; DOS; VxDs; debugging

Abstract

Windows programmers have several options for applications debugging. Many compiler vendors include a source-level debugger with the compiler. Third-party vendors like Multi-scope offer Windows debuggers rich with features.

Systems debugging, however, is a specialized area. Debugging DOS virtual machines (VMs), Windows Virtual Device Drivers (VxDs), or even Windows itself all require patience and knowledge. Daniel Norton, the author of "Writing Windows Device Drivers", looks at Nu-Mega's SoftIce/W debugger, which promises to make systems debugging easier.

June 1992

Vol. 3, No. 6 Debugging

Programming Style

Dan Saks

Key words: C; genericity

Abstract

Dan revisits the topic of compile-time genericity: the ability to have the compiler generate different functions for different data types, based on a single template that you specify.

Other Articles by Dan Saks

February 1992 [Programming Style](#)

August 1992 [Programming Style - Genericity in C++, Part 2](#)

June 1992

Vol. 3, No. 6 Debugging

Tech Tips
Leor Zolman

Key words: C; TSR; device driver; interrupts

Abstract

Copying Large Data Chunks in C

Professor Berthold K. P. Horn had to write code to handle bitmaps larger than 64Kb. He supplies a helpful function, `hmemcpy()`, that can copy bytes from a near buffer to a huge array.

Alternatives to the Three-Fingered Salute

Tired of using two hands to type Ctrl-Alt-Del? Tony Ingenoso provides a program (a TSR version and a device driver version) that reboots when it detects an NMI (which you can generate by pushing a button on some hardware debuggers). The assembly language code provides a look at the relative merits of TSRs and device drivers.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

January 1992 [Tech Tips](#)

March 1992 [Tech Tips](#)

May 1992 [Tech Tips](#)

July 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Windows™/DOS

 Developer's Journal

Vol. 3, No. 7

July 1992

Efficiency

WINMEM: An Efficient Subsegment Memory Allocator For Windows 3.x 5

A complete, efficient subsegment memory allocator in C.

Mark Peterson

Fast, Indexed Help 24

A BASIC implementation of Modified Indexed Sequential Access files for fast, online help.

George Toft

Using the UART's FIFO Queue 33

Using the 16550's FIFO queue is the key to efficient high-speed serial communications.

Charles B. Allison

Features

Inside Windows NT: An Introduction 12

Learn about the design of this new operating system from the author of "Inside Windows NT".

Helen K. Custer

Windows WEP Problems 39

Having problems with your WEP function? Here's why.

Brent Rector

A Windows assert() with Symbolic Stack Trace 49

An assert()-style macro that produces a symbolic Windows stack trace.

Matt Pietrek

Product Previews

The Advanced Power Management Specification 19

Learn how to make your application battery-power aware.

Victor Volkman

Columns

Windows Q&A 41

A Loader Program for 3.0 Installs

Paul Bonneau

Tech Tips 59

Focus on Floppies.

Leor Zolman

July 1992

Vol. 3, No. 7 Efficiency

A Subsegment Allocator DLL for Windows
William Smith

Abstract

Other Articles by William Smith

June 1992 For a Consistent User Interface, Center Your Windows

July 1992

Vol. 3, No. 7 Efficiency

WINMEM: An Efficient Subsegment Memory Allocator For Windows

3.x

Mark Peterson

Key words: Windows; C; memory management

Abstract

The Windows memory management API has several well-known limitations that make it inappropriate for general-purpose memory allocation: it is relatively slow, wasteful of space, and only supports about 8K memory blocks for all applications. C and C++ compilers are now starting to include more reasonable memory managers to use in place of the raw Windows API. All memory managers are not created equal, however. This article demonstrates an efficient, complete subsegment memory allocator written in C and compares it with the Borland and Zortech memory managers.

July 1992

Vol. 3, No. 7 Efficiency

Inside Windows NT: An Introduction **Helen K. Custer**

Key words: Windows; Windows NT; Win32 API

Abstract

One of the biggest contributors to the steep Windows learning curve is the lack of information on the design of Windows. It is very difficult for programmers to construct mental models of how Windows works from the low-level API specification Microsoft offers. Windows NT, however, will not suffer from this mistake. Three years ago, the NT development group at Microsoft brought Helen Custer on board to write "the book" on the design of Windows NT.

In this, the first of a series of excerpts from her upcoming book "Inside Windows NT", Helen provides an overview of the design of this technically advanced operating system. This article outlines the original goals of Windows NT's designers, and describes the significance of the features that Windows NT brings to PC programmers: portability, extensibility through the client/server model, object-based security, symmetric multiprocessing, structured exception handling, Unicode support, advanced virtual memory services, and more.

Other Articles by Helen K. Custer

February 1992 [A Reusable C++ Chunk Allocator for DOS and Windows](#)

August 1992 [Inside Windows NT: The Win32 Protected Subsystem](#)

July 1992

Vol. 3, No. 7 Efficiency

The Advanced Power Management Specification Victor Volkman

Key words: DOS; BIOS; specifications

Abstract

Portable computers bring their own peculiar set of problems with them. Conserving battery life is one of the central problems such computers face. Now, Intel and Microsoft are addressing the problem with the Advanced Power Management Specification. By following the APM specification, you can make your application help extend battery lifetimes and cope intelligently with the threat of power loss.

Other Articles by Victor Volkman

February 1992 Fading-in Custom Controls For Windows 3: Part I

April 1992 The FAX BIOS Standard

March 1992 A Fader Custom Control For Windows 3: Part 2

July 1992

Vol. 3, No. 7 Efficiency

Fast, Indexed Help
George Toft

Key words: **DOS; database; BASIC**

Abstract

Users have become sophisticated enough to expect applications to supply online, context-sensitive help. The help information for a complex application can consume a lot of disk space, so brute-force file access methods are often unacceptably inefficient. This article demonstrates, in BASIC, an efficient Modified Indexed Sequential Access method for implementing fast, context-sensitive help.

July 1992

Vol. 3, No. 7 Efficiency

Using the UART's FIFO Queue **Charles B. Allison**

Key words: C; DOS; serial communications; COM ports; UART; hardware; 8250 UART; 16550 UART

Abstract

Even if you have a fast '486 machine, serial communications can waste a lot of time; interrupting the CPU for each and every character at 9600 baud is not terribly efficient. Fortunately, newer PC UARTs have enough smarts to queue characters so that you can greatly reduce the number of interrupts needed to service the UART. This article uses a C program to demonstrate how to enable and use the new 16550 UART's FIFO queue for more efficient high-speed serial communications.

Other Articles by Charles B. Allison

April 1992 [Fast Times with the PC Clock](#)

July 1992

Vol. 3, No. 7 Efficiency

Windows WEP Problems **Brent Rector**

Key words: Windows; C; DLLs; WEP functions; Windows 3.1

Abstract

The Windows 3.0 SDK implies that you can use a DLL's Windows Exit Procedure (WEP) for most any kind of cleanup. Unfortunately, programmers have discovered the hard way that this just is not true. In this article, Brent lists all the problems and pitfalls related to using Windows WEP functions.

July 1992

Vol. 3, No. 7 Efficiency

Windows Questions and Answers **Paul Bonneau**

Key words: Windows; C; Windows 3.1; installation

Abstract

Question:

I have developed an install program for Windows to transfer the files from my floppies to my hard disk. The install program is on the first disk of a set and is to be run via the Program Manager's Alt+F+R sequence.

I found that when the install program prompts for a disk change, if the second disk has a different label than the first, I get a system error message, explaining that the system can't read from the specified disk drive.

How can I get around this problem other than ensuring all disks of a set have the same label? Also, I understand that install programs which automatically create new Groups and/or place new programs into the Program Manager do so via a DDE process. How is this done and how can I get the specifications?

Peter D'Agostino
9285 Lerwick Drive
Dublin, OH 43017

Answer:

Paul reveals that the root of the problem is a bug in Windows 3.0 that appears to be fixed in Windows 3.1. Paul supplies the complete source to small loader program that helps work around this bug if you need your installation program to work with Windows 3.0. He also outlines the simple steps needed to communicate with the Program Manager and establish a new group for your application.

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)

May 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

August 1992 [Windows Questions and Answers](#)

July 1992

Vol. 3, No. 7 Efficiency

A Windows assert() with Symbolic Stack Trace **Matt Pietrek**

Key words: Windows; C; debugging; toolhelp.dll; .sym files; file formats

Abstract

Assertions are an important coding device for constructing correct, reliable code. C programmers are familiar with the assertion macros provided in <assert.h>. One problem with assertions is that a low-level assertion failure may not provide much information about the higher-level error that caused it. Typically, you then have to load the program into a debugger and cause it to fail again in order to see a stack trace and discover the function that really caused the problem.

Wouldn't it be nice if assert() could give you a symbolic stack trace automatically? That's just what the w_assert() macro in this article does for you under Windows. It uses the new toolhelp.dll (available in Windows 3.1 and backward compatible with Windows 3.0) and .sym files to provide a symbolic stack trace when an assertion fails. Besides providing a useful debugging tool for Windows programmers, this code provides insight into the format of .sym files and how to apply toolhelp.dll to practical problems.

Other Articles by Matt Pietrek

June 1992 [Writing a Windows Debugger](#)

Tech Tips
Leor Zolman

Key words: C; DOS; BIOS; floppy drive

Abstract

Managing Multiple Floppy Drive Letters on a Single-Drive System

Chris Howe points out that, on a single-drive system, DOS will occasionally butt in to emit an "Insert diskette for drive B: and strike any key when ready" message. This can wreak havoc on your carefully crafted interface screen and require time-consuming redraws.

To avoid this problem, Chris supplies two short Turbo C functions to help your application manage the drive. `SingleDrive()` lets you know whether you are running on a single-drive system. `SetFloppy()` lets you control whether DOS thinks that single floppy drive is currently drive a: or drive b:. Judicious use of these functions can make your application immune to those embarrassing DOS floppy drive messages.

A Cure for Floppy Drive Not Ready Errors

Dan Goldberg relates how his Novell menu script ran into trouble whenever the user did not insert the floppy as requested, or forgot to close the drive door, or produced a disk error. He supplies his solution: a C program that checks on the condition of the floppy drive and the floppy disk it currently contains. His program emits an intelligible error message if it detects a problem, and exits with an error code that you can test from within a batch file.

Other Articles by Leor Zolman

February 1992 [Tech Tips](#)

January 1992 [Tech Tips](#)

March 1992 [Tech Tips](#)

May 1992 [Tech Tips](#)

June 1992 [Tech Tips](#)

April 1992 [Tech Tips](#)

Windows™/DOS

Developer's Journal

Vol. 3, No. 8

August 1992

Graphical User Interfaces

Multi-page dialog boxes

13

A unique, convenient method to package lots of options in a single dialog box.

Kanhom Ng

Colorizing Windows

25

Learn how easy it is to create a floating toolbar in Turbo Pascal for Windows.

Doug Overmyer

Generating an Icon Editor

34

An icon editor for EGA/VGA displays.

Peder Jungck

Features

Inside Windows NT: The Win32 Protected Subsystem

5

Learn how Windows NT supports DOS, Windows 3.x, Win32, POSIX, and OS/2 - all at the same time!

Helen K. Custer

PUSHA/POPA Emulation for 8088/8086 CPUs

43

A clever software implementation of the PUSHA/POPA sequence of instructions.

Stephen Nebel

Columns

Windows Q&A

47

Paul shows how you can achieve DOS-like toggling in your Windows application.

Paul Bonneau

Programming Style

55

Dan extends his genericity discussion to C++.

Dan Saks

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Inside Windows NT: The Win32 Protected Subsystem **Helen K. Custer**

Key words: Windows; Windows NT; Win32 API; Win32s

Abstract

Far from being just an extension to Windows, Windows NT is a totally new operating system that can support DOS, 16-bit Windows, 32-bit Windows, POSIX, and even OS/2. How can one operating system provide multiple operating system environments? The key lies in Windows NT's "protected subsystem" architecture and, in particular, in the Win32 protected subsystem. This article explains protected subsystem concepts looks at the structure of the Win32 protected subsystem, which manages the screen and keyboard for environment subsystems and provides the all-important Win32 API.

Other Articles by Helen K. Custer

February 1992 [A Reusable C++ Chunk Allocator for DOS and Windows](#)

July 1992 [Inside Windows NT: An Introduction](#)

Multi-page dialog boxes **Kanhom Ng**

Key words: Windows; dialog boxes; C; user Interface

Abstract

Windows applications often have the need to display more options than can fit on one dialog boxes. Various approaches have evolved in order to keep the user interface simple without restricting the number of options. This article offers a unique solution to this problem with multi-page dialog boxes.

In this article, the Kanhom presents a simple API that allows you to design dialog boxes with multiple "pages". Each page is, in fact, a separate dialog box to the programmer. To the user, each page is a separate set of controls that appears in a specific area of the main dialog box. For example, you can set up a "Next" button that pages through dialog boxes containing sets of lesser-used options.

The end result is that the user sees a simple interface, but can access large numbers of options if necessary. Also, the API presented by the article makes it easy for the programmer to design and control all the pages within the main dialog box.

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Colorizing Windows **Doug Overmyer**

Key words: Windows; Turbo Pascal for Windows; user Interface; toolbars

Abstract

Windows applications that stick with the "standard" Windows 3.0 look and feel are beginning to look stodgy. New Windows applications make use of a variety of graphical techniques to produce flashier, easier-to-use user interfaces. One such innovation is the floating toolbar, a moveable window that contains an array or matrix of graphical buttons you can push to produce program effects. This article shows how easy it is to implement a floating toolbar in Turbo Pascal for Windows. The code also demonstrates some techniques for creating color backgrounds in windows.

Other Articles by Doug Overmyer
February 1992 [A Font Previewer](#)

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Generating an Icon Editor

Peder Jungck

Key words: C; DOS; graphics-mode video; graphics; EGA; VGA

Abstract

The EGA and VGA video hardware lends itself to graphical objects like icons. Once you write the software to display icons, however, you need a tool for creating them. This article demonstrates an icon editor for EGA and VGA devices that you can use as-is or extend into a full-featured tool.

PUSHA/POPA Emulation for 8088/8086 CPUs
Stephen Nebel

Key words: assembly language

Abstract

The 80186 introduced a handy set of new instructions, PUSHA and POPA, for pushing and popping a whole set of registers at once. Unfortunately, if you take advantage of these instructions, your code will not be backwardly compatible with 8086 or 8088 machines.

This article shows a clever way to implement your own software versions of PUSHA/POPA that work with any processor. You can also tailor this routine to save the register set that makes the most sense in your application. You call this routine at the beginning of a function in order to save the registers. The routine saves the registers and, before it returns, modifies the stack so that when your function returns, it will automatically transfer to code that pops the correct registers off the stack.

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Windows Questions and Answers **Paul Bonneau**

Key words: Windows; C; assembly language; BIOS; TSR; text-mode video; user interface; VGA

Abstract

Nathaniel Stitt writes to ask how to switch to 80x25 text mode from within a Windows application. He wants his application to respond to Alt-Enter by toggling between windowed mode and full-screen character mode, just as a DOS box does. Paul provides the answers in the form of a real-mode TSR that handles the necessary screen operations.

Other Articles by Paul Bonneau

February 1992 [Windows Questions and Answers](#)

May 1992 [Windows Questions and Answers](#)

January 1992 [Windows Questions and Answers](#)

March 1992 [Windows Questions and Answers](#)

April 1992 [Windows Questions and Answers](#)

June 1992 [Windows Questions and Answers](#)

July 1992 [Windows Questions and Answers](#)

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Programming Style - Genericity in C++, Part 2
Dan Saks

Abstract

Genericity is a family of techniques for writing reusable software. Dan continues his look at genericity by delving into C++ templates, the latest feature of this evolving language that provides a type-safe alternative to using macros for genericity.

Other Articles by Dan Saks

February 1992 Programming Style

June 1992 Programming Style

August 1992

Vol. 3, No. 8 Graphical User Interfaces

Windows 3 Developer's Workshop
Ron Burk

Key words: Windows; DDE; DDEML; custom controls

Abstract

This book is for the Windows C programmer who has already learned the basics. It covers debugging, making DLLs (including custom controls), and the new library from Microsoft designed to ease DDE programming. The book includes many source code examples.

Other Articles by Ron Burk

January 1992 [Foreign Language Pre-Processor And String Externalization Tools](#)

February 1992 [A Reusable C++ Chunk Allocator for DOS and Windows](#)

March 1992 [Windows 3.1](#)