

Articles

This appendix contains technical articles on these topics:

- Microsoft Windows for Pens
- Quarterdeck's QEMM-386 and Windows 3.1
- PC-NFS and Windows 3.1
- FastDisk: An Introduction to 32-Bit Access

Contents of this appendix

Windows for Pens.....	
Why Pens?.....	
Technical Highlights.....	
The Internal Architecture of RC Manager.....	
Pen for Windows Support Resources	
Quarterdeck's QEMM-386 and Windows 3.1	
QEMM-386 Features for Windows	
Troubleshooting for QEMM	
Getting Additional Help	
PC-NFS and Windows 3.1.....	
Installation Tips.....	
Using PC-NFS With Windows	
FastDisk: An Introduction to 32	
What Is 32-Bit Disk Access?	
The User Interface for 32-Bit Disk Access	
The Components of the 32-Bit Disk Access System	
The Risks of 32-Bit Disk Access	

Windows for Pens

Why Pens?

With pen computing, you can use a pen-like stylus on a computer with a special display to print, draw, and annotate information much as you would with pen and paper. Beneath the computer's display is a digitizer, an electronic sensing device that captures data and displays it as "ink" or interprets it as text or as a command. Pen computing supports:

- **Creation of highly portable devices.** With the pen, hardware developers are no longer constrained by the keyboard, and can deliver smaller, lighter machines that can be used while standing or walking.
- **Development of new markets.** With smaller, more portable pen PCs, new categories of applications such as notetakers and calendars become possible.
- **More intuitive features for users,** combining the familiar feel of pen and paper with the power of the computer. Pen computers may resemble clipboards, notebooks, and pocket calendars.
- **Improved efficiency.** The pen is a superb pointing device for accurate data selection and object manipulation. Gestures (described later in this paper) make data manipulation economical because they can simultaneously communicate position (or object selection) and initiate an action.
- **Flexibility.** The pen is a great drawing tool for creating detailed graphics and, with the Windows graphical user interface and Object Linking and Embedding features, can support creating compound documents. The pen is ideal for taking notes, sketching, and annotating documents.

Who Will Use Pens?

Consider the following scenarios:

- Nurses and doctors can track and record patient histories as they make their rounds, using clipboard-like computers and pens to look up and enter

patient information over a wireless network from servers throughout the hospital.

- Insurance claims adjusters can assess automobile damages on site, looking up relevant cost information with the tap of the pen, then printing the estimate and writing a check to the repair shop at the end of the visit.
- Sales representatives can track inventory and the effect of promotional campaigns in retail stores, using a pen computer. At the end of the day, the information is transmitted through a phone line back to headquarters.
- Financial analysts can perform “what if” analyses in meetings with customers or managers, using a pen to change a number or drag a line on a chart.
- Business people can maintain their calendars on computers that feel more like notepads and calendars than current computers. They jot down notes during meetings, conduct interactive presentations, annotate documents, or edit using simple handwriting.

Pen users will include:

- Service providers such as field sales and service representatives, field engineers, route managers, and insurance adjusters who need to look up information on site, or capture it in a format that is machine-readable to be used in the field or to be communicated to other users through cellular communications or other network solutions.
- Users of laptop and notebook PCs, who will find that the next generation of notebook PCs will most likely be “convertible,” allowing users to detach the display for pen-exclusive uses. These

pen computers will be used as data entry devices, and may support smaller, less expensive compact disk storage and applications for data retrieval.

- Professionals who will add pen devices to existing hardware, controlling the pace and cost of the migration to pen computing.

What Is Windows for Pens?

The modular design of the Windows operating system allows the integration of pen extensions into the basic Windows 3.1 product. Windows for Pens is an OEM product to be packaged and sold with a variety of new pen computers.

Windows for Pens offers a base of pen-specific APIs to allow Windows developers to build pen-centric applications. Windows for Pens supports existing Windows applications without special modification through a special compatibility layer so users can:

- Point and tap as with the mouse.
- Issue commands using a standard set of 13 system-wide gestures.
- Issue commands using 26 user-assigned gestures.
- Print characters in any text field. Hand-printed entries in Windows applications are translated and appear as typed text.

Windows for Pens provides an open, modular approach to handwriting recognition, so you can replace or supplement the Windows for Pens recognizer with third-party recognizers.

Windows for Pens uses the Windows-specific functions such as scalable fonts, Advanced Power Management, support for Flash EPROM, IC card storage, and the ability to run in ROM. Windows for Pens also includes pen-specific DLLs and device drivers so that applications can run on new pen hardware platforms. Applications can run in portrait or landscape mode and can support scaling for different sized displays.

Why Choose Windows for Pens?

Windows for Pens, a complete operating environment for pen computers, offers compelling benefits for all groups involved in developing and using pen applications:

- **For new computer users,** Windows for

Pens provides the building blocks for creating easy-to-use pen-centric applications. Unlike other environments, Windows for Pens does not impose a specific metaphor on end-users, leaving the choice of user interface design and functionality to the application developer.

- **For existing computer users,** Windows for Pens draws on the Microsoft Windows environment, so they can use the same applications in the office or on the road. Users can use the pen with Windows applications to point and tap as they would with a mouse, issue commands using gestures or hand-printed characters to be translated as text.

Windows for Pens, because it is based on the Windows environment, supports a multitude of communication options, printing devices, database engines, and file formats.

- **For Independent Software Vendors (ISVs),** Windows for Pens provides: comprehensive API support for controlling ink, recognition, and gestures; extensive driver support for various pen platforms and OEM hardware; support for multiple recognition engines to facilitate access to domestic and international markets, and all the capabilities provided by the Windows 3.1 environment. Windows for Pens offers developers maximum flexibility to design and implement applications using the Windows interface or innovative metaphors. With Windows for Pens, software developers can use their programming experience and the code base created for Microsoft Windows, and can choose from over 120 different development tools— third generation languages, object oriented development environments, quick UI and application generators, and database front ends. Because of the large and growing installed base of Windows users, ISVs have access to established markets and to new market segments that require connectivity solutions.
- **For corporate MIS,** Windows for Pens provides an environment where developers can draw on a market of skilled developers and consultants, or use their in-house Windows experience with development tools and application generators that shorten the development cycle. New pen-centric applications can be integrated with existing desktop solutions to create enterprise-wide solutions, avoiding the inefficiencies associated with supporting specialized

niche environments, and eliminating the need to maintain different operating systems, code bases, and skill sets.

- **For Original Equipment**

Manufacturers (OEMs), Windows for Pens offers a systems solution that can be scaled across platforms and form factors. Its modularity ensures that it can run on small silicon-memory-based pen computers as well as larger clipboard-type machines and convertible notebooks. Its modular design means that developers can scale down Windows for Pens to under 2 MB — most of which can be run in ROM. Because Windows for Pens can run on a variety of hardware platforms, OEMs can capitalize on their investment in hardware design and implementation.

Technical Highlights

Windows 3.1 includes several special pen features and capabilities. These features, described in detail later in this paper, include:

- **Gestures**, implemented as 13 standard system-wide gestures available across all applications plus 26 application-mappable gestures.
- **The Pen Palette** is a floating toolbar that features buttons for invoking the Writing window, the Trainer, the Gesture Manager, and the On-Screen Synthetic Keyboard.
- **Control Panel settings for pen and ink**, so that users can specify their own custom database of handwriting samples.

The core components are described in “The Internal Architecture of Pen for Windows” later in this paper. These components include:

- **The RC Manager.** This includes the pen, display drivers, and the recognition engine. These components handle the data that flows between the pen and the application.
- **The Windows for Pens Compatibility Layer.** Also referred to as the Pen Message Interpreter, this function interacts with standard Windows applications, so they can use gestures and handwriting recognition.

Gestures

A gesture is a special handwriting symbol that issues a command, such as Copy, or produces a non-printing character, such as a carriage return or backspace.

Figure D.1 Gestures

Gestures can save time, especially for editing or correcting text. With gestures, the user can simultaneously specify an object or function and initiate an action. Windows for Pens gestures are similar in form and purpose to common handwritten editing symbols. Three types of gestures are supported by Windows for Pens:

- Thirteen system-wide gestures that are

uniform across applications.

- Application-mappable gestures that allow application-specific commands to be mapped to an unused set of symbols.
- User-defined gestures, so that users can define their own symbols.

The Pen Palette

The Pen Palette is displayed whenever Windows for Pens is started, providing a pen interface (menus and dialog boxes to execute the most common pen commands), an expandable writing area, and an on-screen keyboard. The Pen Palette is not required for applications that call the pen API directly. To support the feel of pen and paper, Windows for Pens was designed so that users can write anywhere in the text area of an application rather than in a special writing window.

Figure D.2 The Pen Palette

The other elements that support the Pen Palette are:

- **The Writing window and Edit box**, invoked with an edit gesture that resembles a check mark.

The Writing window and Edit box provide a large area for entering, editing, and reviewing hand-printed text. The Writing window appears below the Pen Palette.

- **The Trainer** supports sophisticated on-the-fly handwriting training. Users are not required to train the system before using it and can call the Trainer as needed, without closing work in progress.
- **The Gesture Manager** controls the definition of 26 circled character gestures based on the 26 characters in the alphabet.

Figure D.3 The Wiring window

- **The On-Screen Synthetic Keyboard** provides a tool for entering characters, CTRL and ALT key combinations, and keyboard shortcuts that can be used in conjunction with the Gesture Manager to replace keyboard-based menu accelerators with circled character-based gestures.

The Control Panel

By defining values in the Control Panel, users can control the width of the ink generated by the pen, the orientation of the menus, and the type of user (left- or right-handed). The Control Panel supports multiple users and different display orientations and also defines when recognition occurs—for example, when the pen is lifted or after a time-out.

Figure D.4 Control Panel items for pens

The Internal Architecture of Pen for Windows

This discussion describes how Windows for Pens is for applications programming, and is intended for readers who understand some basics of Windows programming.

Adding the pen to Windows involves enhancing both input and output. For input, the pen device driver provides pen data to the system. For output, pen data is echoed on the display as “ink” while drawing takes place. This ink layer is handled through a new device driver entry point and involves no changes to the GDI interface. The presence of ink has no impact on Windows applications. At this lowest level in the system, connection between pen input and ink on the display is immediate. This is identical to the mouse cursor location being updated at interrupt time.

Above the device-driver level, pen input is interpreted and given meaning. The device driver-to-device driver communication in the hardware guarantees that users see the pen as a writing tool that can freely leave ink on the display. The levels above interpret the ink as text, shapes, and gestures.

Windows applications interact with the pen through the RC Manager, which handles the pen device driver, recognizers, and dictionaries on behalf of all programs. The RC Manager sends pen input to the appropriate window and coordinates pen input processing for that window.

The diagram in Figure D.5 shows the various components of the pen enhancements to Windows. The RC Manager is in the Systems components layer, which also includes MS-DOS functionality, input handling (labeled “Windows” to designate the functionality in the Windows Kernel and USER.EXE libraries), and graphics output (handled by GDI.EXE). The “RC” in RC Manager stands for Recognition Context.

Figure D.5 The pen enhancements to Windows

RC Manager

The RC Manager is the heart of the extensions to Windows provided in Windows for Pens, providing the architectural component that orchestrates pen input handling. Code for the RC Manager is contained in PENWIN.DLL, so the terms “RC Manager” and “Pen DLL” are synonymous.

Whenever a windows program wants to work with the pen extensions, that program obtains an RC, fills in the fields of the RC appropriately, and hands the RC to the

RC Manager. Within the RC data structure are fields that specify exactly how the pen data is to be handled. In this sense, the RC is patterned after the Device Context used with GDI.

In general, one RC is used with each inking “transaction” between the user and the program. Often, programs will find it convenient to allocate an RC permanently and fine-tune it as needed (just like an “own” DC).

Fields within the RC identify which recognizer should be applied to the ink and which dictionaries used to select the “best guess” string. Other fields specify what conditions will end the inking transaction (such as the pen going up, or the pen going down in a specific location, or a time-out for pauses). The RC also provides parameters used by the recognizer and the dictionaries themselves.

At the end of each inking transaction, the RC Manager returns the results of ink processing to the window that initiated the transaction. If the window does not fill in an RC and initiate an inking transaction with the RC Manager, the pen will function exactly like a mouse.

The Pen and Pen Driver

Pens come both tethered and untethered, to serve different markets, but both styles of pen act the same under Windows for Pens. The pen works together with electronics in the PC to encode (x, y) locations and other information. The pen must sense the difference between touching and not touching the drawing surface, typically accomplished with a tip-switch that acts just like the left mouse button. Many pens actually report pressure levels at the surface, but this additional information is not required. Some pen systems also provide location even when the pen is not quite touching the surface (called proximity sensing).

Many pens will also provide a button on the barrel. An OEM might also implement a pen that records the slant of the pen or rotation of the pen barrel. This information is not required by Windows for Pens, but a standard method is provided for applications to determine whether the information is present and, if so, to access the information.

All pen devices must be capable of recording at least 120 events each second, and must have a resolution of at least 200 spots per inch. Many pens will be able to record data at much higher resolution or rates. These requirements guarantee sufficient data for character recognition by the vector-based character recognizers used with Windows for Pens.

All pen data is reported by the Pen Device Driver to

the system at 0.001" resolution. This standard resolution allows improvements in pen hardware. It also is the same as the HIENGLISH Mapping mode in the GDI, and is therefore convenient for programs.

The Pen SDK includes sample device drivers for WACOM and Scriptel tablets. These drivers can be easily modified to support any pen whose hardware uses an interrupt to return data through I/O addresses.

Display Driver

The display driver under Windows for Pens is an enhancement of the standard display driver for Windows 3.1. The extensions support echoing ink on the display as it is drawn by the user, and moving the origin of the screen to support landscape and portrait orientations. Ink support should be provided in such a way that removing the ink does not generate redraw messages to Windows programs. The display driver must also provide a new cursor—the Pen cursor.

Inking is supported by two function calls in the display driver, **InkReady()** and **GetLPDevice()**, and is perhaps the easiest of the functions to implement. Supporting portrait and landscape modes is somewhat more difficult. The Pen SDK includes a sample driver for VGA devices.

Recognizer

A recognizer is a dynamic-link library (DLL) that collects ink from the device driver and interprets that ink as characters in one or more alphabets. When inking begins, the RC Manager calls the recognizer specified in the RC for the transaction. Windows programs do not directly interact with recognizers, though programs may use application-specific recognizers.

Recognizers are required to handle an inking transaction according to the fields set in the RC for that transaction. The recognizer will end the transaction when the conditions specified in RC fields are met. The recognizer will examine the ink for characters in the alphabets and character sets named in the RC. The recognizer will only return characters from the character sets named, and, within those sets, only characters recognized at the level of confidence specified.

During recognition, the recognizer collects and analyzes the ink. The original ink is returned to the calling program along with the results of analysis, which is returned in a symbol graph that is a sequence of 32-bit values. Each 32-bit symbol is either a recognized character or a meta-character used to group alternate possibilities where more than one

character meets the confidence level set for the transaction in the RC. For example, the string of symbols "clea{r|n}" would be a symbol graph representing two possible strings ("clear" and "clean"). In this example, "c," "l," "e," "a," "r," and "n" represent recognized characters, each of which is associated with a confidence level. The "{," "|," and "}" symbols are meta-characters used to define the graph of alternatives.

All recognized characters in a symbol graph are associated with both the confidence the recognizer assigns to that character and the specific strokes in the ink that make up the character. If the recognizer cannot assign a character to some part of the ink with the required confidence, it specifies "don't know" at that position in the symbol graph.

A recognizer returns the results of its analysis to the RC Manager, which arranges to return the results to the calling program. The recognizer returns ink, a symbol graph, symbol elements (containing the symbol value and confidence), and symbol correlations (which list the strokes for a recognized character). Recognizers may return partial results during an inking transaction.

Beyond the requirements for collecting ink and returning results, there are no constraints placed on how recognizers work. Windows for Pens was built from the ground up to accommodate recognizers for languages using non-Roman alphabets. The recognizer interface is open and published, and third-party recognizers will be widely available. At least some Windows programs will want to provide custom recognizers for special symbols.

Compatibility with Windows Applications

The Pen Message Interpreter allows the pen to be used with existing Windows (and non-Windows) applications by acting as a “middleman” between these applications and Windows for Pens. Most pen input is treated as mouse input (for example, a downward motion that depresses the pen point is treated as a mouse click) and is passed to the application’s mouse buffer for processing. Existing Windows applications already buffer mouse input separately from keyboard input, simplifying the pen message interpreter’s tasks.

However, when the user moves the pen into a text entry box, Windows for Pens recognizes that the user is now over a writing area. This causes several changes: the cursor shape changes to that of a pen; the display device driver temporarily throws a transparent overlay over the existing screen to show the “ink” left by the pen; and pen input is sent to the handwriting recognition engine for processing, then sent to the application’s keyboard buffer. Gestures are handled similarly and mapped to command events, such as “copy.”

Pen for Windows Support Resources

CompuServe

CompuServe features a special forum for Windows for Pens beta developers. The forum is used daily by engineers and support specialists who answer technical questions and place technical articles that are of wide interest to Windows for Pens developers. To access this forum, developers need to be on the Windows for Pens beta program. Section 12 under “winbtdev” is dedicated to Windows for Pens issues.

Contact: 800/ 848–8199

We recommend using TAPCIS, The Access Program for the CompuServe Information Service. Put together by Support Group, Inc.

Contact:

Phone: 800/ 872–4768 or 301/ 387–4500

Fax: 301/ 387–7322

Pen for Windows Consultants

Andersen Consulting

Andersen Consulting is a world wide information

systems integrator, with offices in 146 cities. In the U.S., there is an Andersen Consulting office in all major metropolitan areas. Since installing the first business computer in 1954, Andersen Consulting has helped leaders in virtually every industry to apply information technology for competitive advantage and to manage the profound change associated with it. We offer clients our expertise in industry, business, and advanced technology, plus the services necessary to integrate complex business and technology systems. This unmatched blend of capabilities has made Andersen Consulting one of the world’s largest commercial systems integrators. A single source for the total solution, from strategic planning and systems design and development to change management, facilities management, hardware, and software.

As one of the largest systems integration firms in the world, our clients list is extensive. We provide professional services to a significant proportion of the Fortune 500 and to the leading companies in most local markets. In industries like Insurance, Aerospace, and Telecommunications, we are presently working with nearly all the major entrants.

Product Specific Experience: Andersen Consulting is a leader in the development of custom Windows-based applications for major corporations. As part of our ongoing application platform research, we have followed the development of pen-based technology with interest since its inception. We have been a participant in the Windows for Pens beta program since its inception, evaluating it as a basis for client solutions and demonstrating its capabilities to existing clients.

Pen Consulting Services: For the organization considering the application of pen technology, Andersen Consulting can provide a number of services, including:

- Information technology strategy
- Information planning
- Analysis of competitive and management issues
- Systems development planning
- Systems design and installation
- Organizational change management
- Systems support and management

Applying any new technology simply for its own merits is a waste of scarce corporate resources. Andersen Consulting assists its clients to identify and to exploit the opportunities a new technology can provide. We work with our clients to develop solutions that can provide significant, sustained competitive advantage.

Contact for Andersen Consulting Information:

Timothy M. Boudreau
69 West Washington Street
Chicago, IL 60602
Phone: 312/ 507-6526
Fax: 312/ 507-2548

K.C. Watkins
801 Second Avenue
Seattle, WA 98104
Phone: 206/ 233-8241
Fax: 206/ 386-8615

Joseph McCall & Company

Joseph McCall & Company provides professional consulting services in North America in the following functional areas:

- Strategic services: information planning and technology audits
- System design and implementation
- Software evaluation and selection
- Project management and planning
- System testing and production systems support
- Training design and execution

Major accounts include: Department of Business Regulation, State of Florida; Allis-Gleaner (AG Co.); Continental Telecom (Contel); Alabama Gas Company; Blue Cross-Blue Shield of Florida.

Pen-Based Experience: In early 1990, Joseph McCall & Company made a strategic decision to pursue pen-based technology. Since that time we have established ourselves as strategic marketing partners with Microsoft and NCR Corporation and have successfully implemented one of the largest applications of pen-based technology. Our goal is to provide innovative pen-based solutions that meet the information processing needs of our clients. With the experience and knowledge gained from our involvement in pen-based projects, we have successfully applied our expertise across various industries including: Insurance and Healthcare; Consumer Products; Federal, State and Local Government; and Manufacturing.

Windows Resource Kit

Pen Consulting Services: Joseph McCall and Company is an information systems consulting firm that has been providing professional services for over six years. Our focus is on helping our clients achieve the promise of today's advanced technologies. Our goal is to provide innovative pen-based solutions that address the specific challenges facing our clients. We have a dedicated team of professionals, with expertise in all phases of the systems development life cycle. Our challenge as systems integrators is to help companies to build on their investment in existing technologies, while also allowing them to effectively assimilate pen-based technologies into their current systems architecture. Through the strategic application of pen-based technology we can assist organizations in realizing the significant benefits offered by today's advanced technologies.

Contact for Joseph McCall & Co. Information:

Scott J. Brady, Manager, Pen-Based Development
1819 Peachtree Street
Suite 520
Atlanta, GA 30309
Phone: 404/ 352-4241
Fax: 404/ 351-3728

Knowledge-Based Solutions (KBS)

The areas of expertise for Knowledge-Based Solutions include our QUICK START rapid prototyping program for early adopters of pen-based systems; domain experience includes field maintenance, sales force automation, financial, health care, and insurance, with projects throughout U.S., though primarily New York and New England. Representative clients include:

- Startup software company (health care)
- Windows for Pens hardware OEMs
- Windows for Pens ISVs
- Government agency

Product Experience: Experience with all major Windows-based development tools, including but not limited to ObjectVision, Visual Basic, Windows for Pens SDK, as well as connectivity with databases, wireless networks, and other Windows applications.

Pen Consulting Services: We are experts in the deployment of field automation systems, specifically involving pen-based systems. Our early work on pen computing with Microsoft and other pioneers, combined with extensive contacts, provides KBS clients with access to the most complete base of experience with this technology.

KBS works closely with vendors as an approved NCR Software Alliance Partner and a Microsoft Consultant Relations Program member.

Our low-cost, low-risk QUICK START program provides clients with a chance to quickly produce a “proof of concept” system to get started with pen-based computing. Other KBS services include full system development, executive overviews, technology transfer, and strategic analysis. Please contact us for a copy of our QUICK START brochure.

Contact for KBS Information:

Randy Parker
65 Westbourne Terrace, Suite 300
Brookline, MA 02146
Phone: 617/ 566-2451
Fax: 617/ 734-8442

Microsoft Consulting Services

Microsoft Consulting Services are available worldwide. MCS Consultants have assisted organizations throughout the world. MCS has already served hundreds of clients with tasks ranging from developing feasibility studies to providing onsite, long-term consultants. We’ve worked with dozens of industries, including aerospace, public utilities, law, finance, government, hotels and resorts, telecommunications, health care, automobile, manufacturing, and others.

Microsoft Consulting Services is staffed by consultants with a wide range of expertise in the field of information technology management and world-class technical consultants in areas such as the Microsoft Windows graphical environment, OS/2 operating systems, Microsoft Visual Basic modern programming system for Windows, Microsoft Mail for PC Networks, LANs, multimedia, enterprise networking, cooperative processing, development tools, and office automation. And because MCS is an integral part of Microsoft, clients have excellent resources. MCS is directly linked to Microsoft product development and support groups and is able to obtain their tools, services, and expertise.

Major clients include: Baxter Healthcare, British Petroleum, Coca-Cola Company, Dun and Bradstreet, Eli Lilly and Company, JP Morgan Company, Nike International, and Texaco are a partial list of organizations served by MCS worldwide.

Pen Consulting Services: MCS offers four primary services, tailored to meet the needs of individual clients and devised to get the job done for the mobile professional:

- Planning
- Design
- Development
- Implementation and support

We are committed to helping you leverage your investment in information technology to your best advantage. We can provide assistance with all phases of a project in developing systems that take advantage of the portability and the familiarity of pen computing. Our mission is to help you apply the leading information technologies such as pen computing to the information needs of your organization.

Contact for MCS Information:

Western Regional Office:

Microsoft Consulting Services
Microsoft Corporation
One Market Plaza, Suite 1550
Spear Street Tower
San Francisco, CA 94105
Phone: 415/ 905-0235

Eastern Regional Office:

Microsoft Consulting Services
Microsoft Corporation
200 State Street
Boston, MA 02109
Phone: 617/ 951-2780

Sales Technologies

Sales Technologies (ST) develops, installs, and supports tailored, networked systems that improve sales force productivity for corporations with mobile field sales forces. ST's software product consists of five core application modules: Territory Management, Electronic Mail, Electronic Forms, Electronic Reports Distribution, and Spreadsheets. Pen-based systems are already being used with the Territory Management module, and further development of pen-based software is underway. ST also offers a broad range of services to keep customers' systems running smoothly, including design consulting, database development, 24-hour help line, data center, hardware repair and replacement services, and comprehensive user training.

ST has developed sales force productivity solutions for companies in many different industries, including consumer package goods, industrial, oil and gas, and health care.

Major clients include: 3M, General Mills, Gillette, Mobile Oil, SmithKline Beecham, The Upjohn Company.

Pen Consulting Services: Sales Technologies offers custom solutions based on its core products. Our primary services include:

- Planning
- Design
- Development
- Implementation and support.

Contact for Sales Technologies Information:

Gitesh Ramamurthy
3399 Peachtree Rd. NE
The Lenox Building, Suite 700
Atlanta, GA 30326
Phone: 404/ 841-4000
Fax: 404/ 841-4115

Quarterdeck's QEMM-386 and Windows 3.1

Quarterdeck's QEMM-386 versions 6 and above support Windows 3.11, which is invisible to MS-DOS. QEMM-386 enhanced mode. Windows and MS-DOS 5.0 provide memory management features, but QEMM-386 remains popular as a 386/486 memory manager because it combines performance with an easy-to-use package. In complex Windows installations, where insufficient memory for non-Windows applications or memory conflicts are a problem, QEMM-386 can often provide solutions.

QEMM-386 Features for Windows

- QEMM-386 combines the functions of HIMEM.SYS and EMM386.EXE with a smaller “footprint” in memory below 640K (3K to 4K).
- New Stealth technology provides up to 211K of upper memory blocks in which to load resident programs by hiding machine ROMs and allowing UMBs to be mapped over them.
- Squeeze technology allows larger resident programs to be loaded by squeezing them temporarily into the EMS page frame while loading. With the increased amount of UMBs that Stealth provides, Squeeze generally eliminates the need to disable EMS support to load resident programs into UMBs.
- Manifest, Quarterdeck's memory analysis program, is currently included free. Manifest provides extensive information about the computer where it is running and is helpful for optimizing a system or diagnosing a memory problem or conflict.
- The Optimize program automatically loads TSRs and device drivers into the optimum regions of upper memory.
- The Analysis program identifies memory conflicts that might exist on your machine.
- QEMM-386 provides the option to map ROMs into fast RAM—a feature that often results in substantially better performance, especially where screen update speed is important.
- QEMM-386 can manage shadow RAM

automatically makes this memory available for Windows or non-Windows programs.

- QEMM-386 supports Windows in both standard and 386 enhanced mode, whether or not QEMM-386 is active. Currently, all other 386 memory managers only support Windows 386 enhanced mode. No special setup is required; just start Windows in the mode you prefer. Real mode is also supported for Windows 3.0.
- QEMM-386 automatically frees unused UMBs to support Windows translation buffers. This further lowers the overhead of Windows in MS-DOS and allows you to run larger non-Windows programs.
- The Loadhi programs allow TSRs and drivers to be loaded into specific upper memory locations, for efficient use of upper memory. With the Optimize program, optimum configuration is achieved automatically.
- EMS and XMS “share” extended memory, so there’s no need to divide extended memory between extended and expanded. Applications can use up to the maximum amount of memory available as EMS, XMS, or a combination of the two, without reconfiguring or rebooting the system.
- QRAM and QEMM, companion products, allow users of 80286 machines to use Loadhi technology, while maintaining an interface consistent with 386 and 486 machines running QEMM-386.

Installing QEMM

To install QEMM-386 on a system where Windows has already been installed:

- Run the QEMM-386 Install program from

the command prompt.

The Install program copies the necessary files to your disk and makes the necessary changes to CONFIG.SYS and AUTOEXEC.BAT. Install detects whether Windows is on the machine and adds the entry **SystemROMBreakPoint=false** to the [386enh] section of SYSTEM.INI, so that Windows can support the QEMM-386 Stealth feature. Then Install runs the Optimize program to set the system up in an optimum configuration to support Windows and other non-Windows applications you might run. After installation, Windows 3.1 should operate correctly in both modes with QEMM-386.

To install Windows if QEMM-386 is already installed:

1. Leave the QEMM-386 drivers installed and active on your machine. The Windows Setup program detects QEMM-386 and installs in a way that is completely compatible with it.
2. Run the Qwinfix program in the QEMM directory, which adds the entry **SystemROMBreakPoint=false** to the [386enh] section of SYSTEM.INI, so that Windows can support the QEMM-386 Stealth feature.
3. Run the Optimize program in the QEMM directory to ensure that the system is still configured properly after Windows adds or updates drivers in CONFIG.SYS or AUTOEXEC.BAT.

Tips for the Optimize Program

The Optimize program runs automatically when you install QEMM-386 for the first time. This sets up your system to load the resident programs and drivers you are currently using into the most efficient locations in upper memory.

However, if you add or upgrade drivers later, memory requirements may change. Therefore, whenever you change CONFIG.SYS or AUTOEXEC.BAT or install programs or upgrades (such as Windows) that may change these files, always run Optimize from your QEMM-386 directory. It takes only a couple of minutes and keeps your system configured correctly.

Troubleshooting for QEMM-386

Quarterdeck rarely sees problems related to QEMM-386 when running Windows in standard mode. However, because of the range of hardware and software combined in some systems, we have seen problems in some configurations when trying to run Windows in 386 enhanced mode. These problems include failure of Windows to run in 386 enhanced mode, system crashes when exiting Windows, and, in some cases, problems running Windows in 386 enhanced mode.

Windows Fails to Run in 386 Enhanced Mode

If you have trouble starting Windows in 386 enhanced mode, check your system to be sure that it has been set up as suggested in the QEMM documentation. The following items are important to 386 enhanced mode operations.

- **Be sure you have enough free memory.** You should have at least 1 MB of XMS memory free after QEMM-386 has been loaded to run Windows in either standard or 386 enhanced mode. The Manifest utility reports this memory under “Extended/XMS” memory.

There may be conditions under which you can run Windows in either mode using HIMEM.SYS and EMM386.EXE, but cannot with QEMM-386. This is because QEMM-386 creates more upper memory space than the Microsoft utilities and, as a result, uses more extended memory. If your machine has a smaller amount of extended memory, you may have to reduce the amount of upper memory created by forgoing the Stealth option, or decrease the amount of extended memory being used by SMARTDrive.

- **Be sure you have enough free disk space.** Unless you created a permanent swap file, you should plan to keep 2 to 3 MB of free disk space on the drive that Windows uses for its swap files. Otherwise Windows may not run in 386 enhanced mode.
- **Be sure you have enough file handles available.** Microsoft recommends setting **files=30** in CONFIG.SYS. QEMM’s FILES.COM program can be used to load some of these files into upper memory. However, we recommend having at least **files=15** in your CONFIG.SYS file and loading the rest of your files with FILES.COM. Windows may not run in 386 enhanced mode if you do not have a **files=** statement in

your CONFIG.SYS, or if it is set below 15.

- **Remove the SORT parameter if your machine has different speeds of memory.** And if your machine usually has less than 640K conventional memory, where QEMM-386 is filling the memory to 640K, you must use the NOFILL (NO) parameter. Although QEMM-386 supports memory sorting and conventional memory filling, Windows cannot support these features and cannot run in 386 enhanced mode when they are active. It is possible to use these features with Windows standard mode, since in this mode QEMM remains active. Reinstall QEMM-386 from the original disk and allow the Optimize program to run. The Install and Optimize programs will automatically reset many of the parameters necessary for Windows to work correctly, in case any of these have been deleted.

If the Preceding Suggestions Fail . . .

If it is difficult to get Windows to run in 386 enhanced mode, here are some other things to check:

- If the system has shadow RAM, try QEMM-386's NOSHADOWRAM (NOSH) parameter. If you are not sure, try it anyway. The parameter will not have harmful effects if your system does not have shadow RAM.
- If random characters appear on the display, accompanied by beeping, and Windows drops back to the command prompt when you try to run 386 enhanced mode, check for the file WINHIRAM.VXD in the QEMM directory on your hard disk. If you are running QEMM-386 version 6.0 or greater, WINSTLTH.VXD should also be present. Also, make sure these files are identical to the ones that came with

the version of QEMM-386 you are running.

- Run the Windows Setup program from the command prompt (not from Program Manager) and make sure you have properly identified your machine. The default Setup selection specifies "MS-DOS System," and many users accept this option. However, Windows provides specific setups for many common PC compatibles such as AST, NCR, NEC, Toshiba, Everex, and Hewlett-Packard. Check this list if you are having problems. If your machine is on the list, use that selection.
- Run Manifest and, in the "Expanded" memory section, verify that the EMS Page Frame has not been set lower than C000 or higher than E000. Although QEMM-386 can establish the frame outside these limits, Windows will not support a frame outside this range and will not run in 386 enhanced mode. If the frame is outside these limits, consult the QEMM-386 documentation regarding moving it.

In some cases the following modifications to SYSTEM.INI have solved problems:

- Add the entries **DualDisplay=True** and **EMMExclude=A000-FFFF** to the **[386enh]** section of SYSTEM.INI.
- If you have an XGA card, replace the line **Display=XGAVDD.386** with the line **Display=*VDDVGA** in the **[386enh]** section of SYSTEM.INI.
Or use the QEMM "EXCLUDE" parameter to exclude the first 4K of the 32K region where the XGA ROM is located in the 640K-1024K area. For details, see the Quarterdeck Technical Note #208, "XGA Video and Quarterdeck Products."

Problems Running Windows in 386 Enhanced Mode

When Windows runs in 386 enhanced mode, control of memory (including all of the upper memory area) passes to Windows, using drivers that were co-developed by Quarterdeck and Microsoft. When Windows runs in 386 enhanced mode, QEMM-386 is inactive. When you quit Windows, QEMM-386 again takes control. As a result, under 386 enhanced mode, QEMM-386 can neither directly cause problems nor prevent them if they occur. However, QEMM may have loaded some resident programs or may have made changes to the system before being disabled. These changes may later present problems in Windows 386 enhanced mode.

One way to determine whether QEMM-386 causes the problem is to replace the QEMM driver with HIMEM.SYS and EMM386.EXE. Then see if the problem persists. If it does, the problem is not related to QEMM. Follow the troubleshooting steps in Chapter 13, "Troubleshooting Windows 3.1," of the *Windows Resource Kit* to find a solution.

Programs Fail When Loaded High

Some resident programs fail when loaded into the upper memory area because they contain logic that fails when the program is loaded above 640K. With the growing popularity of memory managers such as QEMM-386 that load programs above that barrier, most of these programs have been rewritten, so the likelihood that you will run into one of these programs becomes less and less. Still, if you are having a problem and find the problem goes away when you remove one of your resident programs, you should try loading that program into conventional memory.

Expanded Memory TSRs

QEMM-386 can load resident programs (TSRs) into upper memory and provide expanded memory for their use. However, not all resident programs are compatible with Windows, particularly when they are using expanded memory. If you are running such programs before running Windows and are experiencing problems, try removing them. If the problems go away and the programs can be configured to use XMS, try configuring them that way. If problems persist, contact the manufacturer of the program to ask about compatibility with Windows.

Bus-Mastering Devices

Bus-mastering devices do their own direct memory addressing (DMA). DMA is done without going through the machine's processor. Currently, the most common bus-mastering devices are SCSI hard disk controllers, but technically, other types of devices can also be bus-mastering.

The problem with these devices is that they put data into absolute memory addresses and assume that the content of those memory addresses always remains constant. However, when running Windows 3.1, this can be an incorrect assumption. When Windows is managing memory, the same physical memory addresses can at any given moment hold different data, depending on which virtual machine is current.

To run Windows 3.1 properly with bus-mastering devices, you should obtain a driver from the device manufacturer that meets the VDS (Virtual DMA Services) specification. With such a driver, Windows should be able to operate properly with your bus-mastering device.

When you are not running Windows in 386 enhanced mode, QEMM-386's DISKBUF (DB) parameter can be used to work around these problems. But again, because QEMM-386 must be disabled when Windows runs in 386 enhanced mode, the DISKBUF parameter will have no effect. SMARTDrive can perform this sort of buffering as well. So if you are having trouble running Windows in 386 enhanced mode, loading SMARTDrive may help.

Windows Runs Slowly in 386 Enhanced Mode

A full hard disk impairs the Windows paging abilities and reduces the size and number of programs you can run in Windows. If Windows seems to be slower or your system runs out of memory, check that you have at least 2 MB of free disk space on the drive to which the TEMP environment variable points.

Speed can also be affected by fragmentation of your hard drive. In these cases, setting up a permanent Windows swap file, as outlined in the *Windows User's Guide*, might be useful.

Finally, if your machine does not provide caching of video ROM into fast RAM, you should try using QEMM's ROM parameter, which provides this feature through software. Often, this will speed up the Windows screen display.

Getting Additional Help

If you have problems you have not been able to resolve while running Windows with Quarterdeck products, help is available through the following sources:

- **Q/FAX: 213/314-3214**

FAX-on-demand service. Contains up-to-date technical notes, product and upgrade information. Available 24 hours a day, 7 days a week. Call from your fax machine.

- **Electronic Bulletin Board Service:
310/314-3227**

Available 24 hours a day, 7 days a week.

- **Telephone Support: 310/392-9701**

Available business days, from 9:00 a.m. to 5:00 p.m. Pacific Standard Time

- **CompuServe and BIX**

CompuServe subscribers can reach the Quarterdeck Support Forum by typing **GO PCVENDB** at the prompt. For BIX subscribers, type **GO DESQVIEW**.

PC-NFS and Windows 3.1

The tips that follow will let you more effectively use PC-NFS. For more information about these topics.

Installation Tips

- Install Windows before you install PC-NFS.
- Make sure you have 1 MB of free disk space available to perform a Basic PC-NFS installation, or 3 MB of disk space for a Full PC-NFS installation.
- Sun recommends that you do not use the Windows Setup “Network” option when installing PC-NFS.
- The PC-NFS installation program performs all of the Windows-specific configuration tasks necessary to use PC-NFS with Windows, including modifying files such as `PROGMAN.INI` and `SYSTEM.INI`. If you choose not to let the installation program modify system files, you must edit `SYSTEM.INI` and `PROGMAN.INI` manually. See Chapter 3 of the PC-NFS documentation for information about modifying these files.
- PC-NFS installation copies several `.EXE`, `.DLL`, `.HLP`, `.GRP`, `.ICO`, `.SYS`, `.DRV`, and `.PIF` files to the installation directory, and creates new Windows files, including:

<code>WINHOSTS.FIL</code>	Names of NFS file servers to browse
<code>WINHOSTS.PRN</code>	Names of print servers to browse

Also, PC-NFS installation modifies these existing Windows files:

<code>PROGMAN.INI</code>	Adds <code>PCNFS.GRP</code>
<code>SYSTEM.INI</code>	Adds lines that define PC-NFS drivers

During its operation, PC-NFS modifies these files:

<code>WINFILE.INI</code>	Adds parameters used by File Manager
<code>WIN.INI</code>	Adds connected drives
<code>PCNFS.INI</code>	Lists default values for mounting drives and printers

- During PC-NFS installation, you will be prompted to assign an

interrupt number to any 3COM Ethernet adapter you will use. The installation program uses IRQ 3 as its default value. You must change this number for the adapter to operate properly with Windows. Change the interrupt value to any unused interrupt. IRQ 5 is a commonly available value.

-
- If you upgrade from PC-NFS version 3.5 to version 4.0, remove any /Y options from the statement **device=pcnfs.sys** to your CONFIG.SYS file.

Using PC-NFS With Windows

- The PC-NFS terminate-and-stay-resident (TSR) component called the Resident Transport Module (RTM) must be loaded before Windows to use PC-NFS and its utilities. This file cannot be started with a WINSTART.BAT file. Sun recommends adding the command-line **rtm.exe** to the AUTOEXEC.BAT file of computers that will run Windows.
Another way to use the RTM is to create a batch file that loads the RTM, runs Windows, and then unloads the RTM when you exit Windows.
- If you plan to receive messages using the PC-NFS Console Messaging utility, you must load the Listener TSR program before starting Windows.
- Sun recommends a minimum of 4 MB of memory for running PC-NFS with Windows.
- All swap files and temporary files used by Windows must reside on a local (not a network) disk drive. Therefore, be sure that each user's TEMP environment variable refers only to a local disk drive. Also, the SYSTEM.INI settings shown below should not refer to a network drive:

```
[386enh]                [NonWindowsApp]
PagingDrive=            SwapDisk=
```

- If you manually edit the SYSTEM.INI file, make sure the following entry appears in order to avoid "General Failure - Abort, Retry, Ignore" errors:

```
[386enh]
UniqueDOSPP=TRUE
```

To run PC-NFS, Windows, and Norton Desktop 1.0, add the following entry to the PCNFS.INI file in the WINDOWS directory:

```
[Compatibility]
Norton1.0=1
```

Figure D.7: Disk access in 386 enhanced mode without FastDisk

FastDisk: An Introduction to 32-Bit Disk Access

The new 32-bit disk access feature in Windows 3.1, also known as “FastDisk,” is a system of Windows drive components that allows Windows to talk directly to the hard drive, bypassing the BIOS.

Normally, in MS-DOS, when an application wants to read from the disk, it follows this sequence:

1. The application makes an INT 21H call to MS-DOS.
2. MS-DOS decides where the requested part of the file exists on the disk.
3. MS-DOS sends an INT 13H call to the disk BIOS, which contains a device driver for each hard disk in the system.
4. The BIOS driver talks directly to the hard drive controller. Each hard drive controller requires its own disk BIOS driver, usually specific to the controller board or adapter that connects the hard drive to the rest of the system.

This process is illustrated in Figure D.6.

If your system has an off-the-shelf disk drive, your computer’s ROM BIOS may not know how to talk to it without a specific device driver. Such a driver might be installed in your CONFIG.SYS or, for example, if you use a CD-ROM drive, the MSCDEX file is the device driver that provides the BIOS interface. These drivers trap INT 13H calls and attempt to access the drive without passing it to ROM BIOS.

What Is 32-Bit Disk Access?

Figure D.6: Disk access without FastDisk

32-bit disk access replaces the disk BIOS. Like MSCDEX, it serves as a device driver that talks to the hard drive controller, watching for INT 13H calls and handling them directly or passing them on to the BIOS.

32-bit disk access must exactly match the hardware controller, hence a specific hard drive. Windows 3.1 supports the standard introduced by Western Digital for 1003 controllers. This standard supports all types of hard drive controllers except ESDI and SCSI. And 32-bit disk access is an open standard that Microsoft promotes to all manufacturers of hard disk controllers, so they can write a FastDisk device to support this functionality for their hardware.

However, 32-bit disk access is not required. If you don’t have a FastDisk device installed, your system will run the same way it did in Windows 3.0.

For Windows 3.1 in 386 enhanced mode:

- 32-bit disk access speeds up disk access.
- 32-bit disk access allows you to run more non-Windows applications, and it speeds up the system when you are running non-Windows applications in the background.
- 32-bit disk access provides a new mechanism for ISVs to write powerful disk utilities.

FastDisk Speeds Up Disk Access

On many machines, the disk BIOS is slow and unwieldy. When a FastDisk replaces the BIOS functions, many systems show impressive performance improvements. Also, 32-bit disk access allows Windows to handle BIOS requests in protected mode, rather than in real mode. Under 386 enhanced mode, the disk-reading sequence described at the beginning of this article becomes more complicated, because Windows and Windows applications run in protected mode, yet the system must switch to real mode (or virtual 8086 mode) in the background to run code such as MS-DOS and the BIOS. Switching modes is time consuming.

For this process in 386 enhanced mode, the system begins in virtual mode while running the application. When the application makes a call to read from a file,

Windows traps this interrupt and switches to protected mode, where several virtual devices check the call for actions they need to perform. If none of the devices intervenes, Windows hands the call to MS-DOS and switches back to virtual mode. MS-DOS finds the requested location on the disk, then generates an INT 13H call to the BIOS. Again, Windows traps that call and switches to protected mode for additional processing before handing it back to the BIOS, then switches back to virtual mode. The BIOS passes the call to the hard drive controller to set the physical read in motion, waits until that physical read is finished and returns from the INT 13H call. Windows switches back to protected mode to perform more processing, then returns to virtual mode to let MS-DOS see the return and perform more processing before returning from INT 21H back to protected mode. Windows again performs additional processing before finally switching back to virtual mode for the application that started this process.

Figure D.8: Disk access in 386 enhanced mode with FastDisk

With 32-bit access, a FastDisk device is inserted to trap INT 13H calls and handle them in protected mode. This saves two mode transitions and perhaps more, because MS-DOS might have performed many INT 13H calls for the task. This is illustrated in Figure D.8.

FastDisk Allows You to Run More Non-Windows Applications

Users often complain that when the Help About dialog box shows, for example, 14 MB of free memory they can run large Windows applications, but they get out-of-memory messages when trying to start another non-Windows application. And even though all that memory is available, the Windows applications run slowly and hit the disk constantly.

These discrepancies occur because of the difference between virtual memory (of which you often have a great deal) and physical RAM (which is usually still limited). Virtual memory might be four times the amount of physical RAM, or even more under Windows 3.1. Although virtual memory is great for Windows applications, it doesn't always fill the need.

Each non-Windows application runs in its own virtual machine, taking up 640K or more of virtual memory. But when that virtual machine is running, all of it must occupy physical RAM, even if the application is idle in the background.

Windows applications can be paged to disk so that

only the parts being used take up physical memory. But paging for non-Windows applications presents problems. If part of a non-Windows application were allowed to be paged to the disk, it might happen to be the application's data buffer. Windows would think that it's okay, because it can read the data back when the application tries to touch it. But suppose the application makes a call to MS-DOS to read data from the disk into that buffer. MS-DOS begins handling the call, then sends an INT 13H call to the BIOS, which in turn talks to the hard disk controller until data begins to be transferred into the buffer. Windows see that it needs to page the buffer that's currently on disk back into memory, but it can't call MS-DOS, because MS-DOS is already busy and cannot be reentered—if Windows calls again, the call fails. Windows can't read the data by calling the BIOS, because the BIOS is also in the middle of a call and also can't be reentered. So Windows can't let the current call finish until it pages in the buffer, and it can't page the buffer back in until the current call finishes.

The only way to ensure that this situation does not occur has been to avoid paging in anything while the non-Windows application is in the middle of an MS-DOS or BIOS call. Windows has done this by never paging an active non-Windows application out to disk. In other words, all 640K or more of that virtual machine had to be in physical memory.

This rapidly reduces free RAM. The Help About box might show that your system has plenty of memory, but it is virtual memory. If you try to start another non-Windows application, which requires yet another 640K of physical RAM, it can't be done.

In Windows 3.0, if you had a swap file, Windows could page using INT 13H BIOS calls. Windows 3.1 can do this with calls to the 32-bit disk access system, which can queue multiple requests and is completely reentrant. It doesn't rely on MS-DOS or the BIOS, so it can always page data in from the hard drive. The application in the virtual machine can be completely paged, no longer taking up physical RAM.

With 32-bit access, if you start three non-Windows applications, and then start another, it will work. When the Help About dialog box shows 14 MB, you can actually use all 14 MB.

FastDisk Makes the System More Responsive for Non-Windows Applications

By paging non-Windows applications with 32-bit access, Windows can free more RAM for running multiple non-Windows applications. You can run more applications, and the system runs more quickly.

You may also notice a difference when switching between non-Windows applications.

Without 32-bit-access, if a non-Windows application were configured so that it could not run in the background, then it could be swapped out to disk. But when you press ALT+TAB, it could take a long time for the application to be read into RAM. By paging with 32-bit disk access, Windows can load it into just the amount of memory that the application is actually using, so task switching becomes almost instantaneous.

FastDisk Enables New Classes of Applications

32-bit disk access devices provide services to other virtual devices. Those services can also be provided to applications. New applications and utilities will soon appear that rely completely on 32-bit disk access. These utilities might, for example, allow applications to perform asynchronous I/O, reading and writing unobtrusively in the background, or might use a FastDisk for a tape drive to back up your system from inside Windows. These applications could not be written without 32-bit disk access, but this means that 32-bit access will be required for some applications.

The User Interface for 32-Bit Disk Access

32-bit disk access is turned on and off in the Control Panel through a checkbox in the 386 Enhanced dialog box that appears if FastDisk drivers are available for one or more of the controllers on the machine. Click the checkbox in Control Panel to toggle the value for **32BitDiskAccess=** in the **[386enh]** from **Off** to **On**. You must restart Windows for the change to take effect.

You can also run Windows at the command prompt with the **win /d:f** switch to disable FastDisk temporarily (for that Windows session).

The Components of the 32-Bit Disk Access System

The 32-bit disk access system consists of these 386 enhanced mode virtual devices:

- **WDCtrl**, which is the FastDisk device that talks to standard Western Digital 1003 or ST506 hard drive controllers (about 90 percent of existing disk controllers). This device is only installed if Windows Setup detects a compatible hard drive controller.
- **BlockDev**, which coordinates devices calling block I/O services and the FastDisk devices that provide those services for specific hard drive controllers. This device is always installed.
- **PageFile**, which handles the virtual memory paging file. It calls through BlockDev if any FastDisk devices are available. This device is always installed.
- **Int13**, which traps and emulates INT 13H BIOS calls by calling BlockDev. This device is only installed when at least one FastDisk device is present.

Because WDCtrl is the only FastDisk device included with Windows 3.1, the standard components only support 32-bit disk access on Western Digital-compatible controllers. When you run Setup, it automatically detects these controllers and, if these controllers are present, adds the following lines to the **[386enh]** section of SYSTEM.INI:

```
32BitDiskAccess=off
device=*int13
device=*wdctrl
```

The 32-bit disk access is turned off by default. If you turn on 32-bit disk access in the Virtual Memory dialog box, then **32BitDiskAccess=** is reset to **On**. If you turn off 32-bit disk access, you do not need to delete any files from your hard disk, because the devices are built into WIN386.EXE.

The Risks of 32-Bit Disk Access

Microsoft has identified machines where WDCtrl can crash the system, usually in cases where hard disk controllers appear to be 1003 compatible, but are not. WDCtrl has been made as safe as possible. Each time it starts, it tests to ensure that it communicates properly with the hard drive controller. After much redundant checking, WDCtrl tries to read data from the disk, and, if that passes, it tries writing to the disk and reading it back. Only if all this passes does FastDisk continue and Windows is started.

However, there are drives that appear to be 1003 compatible but are not, and that are not detected. Also, there can be problems even with compatible controllers on some portable computers, specifically those that power down the hard disk to conserve power without telling the running software. If the disk is powered down and back up while its state is being tracked by WDCtrl, the state information will be incorrect, and serious damage can result.

This is why 32-bit disk access is not turned on by default. You must decide to turn it on in the Control Panel. For OEMs who preinstall Windows on a known system, however, 32-bit disk access can be turned on by default in complete safety.

If the initial WDCtrl tests fail, you will not know that anything is wrong. You just won't receive the benefit of 32-bit disk access. If a later test fails, an error message appears, warning you that something has gone wrong and advising you to reboot.

