The Future of DOS

1994 by Paul Williamson

What's in store for DOS users and developers in the coming year?

As the time approaches for the release of *WINDOWS95* more and more people are asking "What do you think is going to happen to DOS?". Some are very sure that there will no longer be a DOS and that "DOS is dead". I just don't believe that's the case!

It is my professional opinion that DOS will be with us for many years to come...and not just as standalone computers shunted into the corners of major businesses to be relegated as *dumb* 3270 terminals. There are a multitude of DOS based systems running DOS based applications still in use today doing real work reliably- all over the world.

In order to speculate on the future of DOS with any degree of ball park accuracy, one needs to have an understanding and appreciation of the brief history of the Microsoft's Disk Operating System or MS-DOS. Since the introduction of version 1.0 in 1981, DOS has gone through more than 10 major changes. Version 1.0 was originally designed for the IBM PC and only supported single-sided diskettes. In 1983, version 2.0, the first major revision, was released adding support for a hierarchical directory structure and the implementation of fixed (hard) disks. 1984 brought versions 3.0, 3.1, 3.2 and the highly acclaimed version 3.3 to market, which added support for the 1.2Mb diskettes, Microsoft networking support, 3.5 inch floppy drives and disk partitioning.

It was at this point in time, that the business community was taking a very serious look at the microcomputer to off-load some of the processing burden from their mainframes. At least two major industries invested heavily in the development of DOS based applications, betting on its future as well as their own, the banking industry and the insurance industry. DOS 3.3 provided the most stable platform available at the time and there are some who still believe that. Many mainframe and minicomputer manufacturers also saw the potential that the personal computer and DOS provided and started including the PC as an integral part of their product.

The introduction and acceptance of DOS 3.3 became more evident as time went by without further updates from Microsoft. Hardware manufacturers sprang up in numerous garages and basements throughout the country. DOS 3.3 and the Personal Computer became dominant during this period. It wasn't until late in 1988, four years after the release of version 3.3, that Microsoft released the next

major version: MS-DOS 4.0. This was the first release to add support for expanded memory and logical disks or partitions greater than 32 megabytes. This was also the initial release of a graphical shell. Using this interface, almost everything that had previously been accomplished from the command line, could now be done by pointing and clicking with a mouse. No matter how many good things were put into MS-DOS 4.x, there was so much bad with it, that many users, and developers alike, were forced to retreat to DOS 3.3. But the concept, the ideas, and the methodology brought to the market by DOS 4.x dictated that this trend would have to be continued and it was!

Probably the most powerful version of DOS came in 1991, with the introduction of MS-DOS 5.0. Now, users of the 80386 and 80486 processors, and even some 80286 machines, regained hundreds of kilobytes of conventional memory. The memory management capabilities of DOS 5.0 enabled DOS itself to load in the High Memory Area, which only exists because of a quirk in the way DOS uses the Offset:Segment addressing scheme. DOS 5.0 also allowed TSR (Terminate and Stay Resident) programs and device drivers to be loaded in to the Upper Memory Area, which previously had been reserved for hardware devices and the system BIOS shadowing. Additionally, DOS 5.0 included an upgraded shell which allowed for task swapping, a full-screen editor replacing the archaic, but functional, Edlin. ***BASIC*** **was replaced with QBasic, which allowed for a full-screen editor. DOS 5.0 soon became the "operating system of choice" for the '90s.**

The year before the release of MS-DOS 5.0, Microsoft introduced the most exciting and revolutionary graphical interface environment: Windows 3.0. Windows 3.0 took the computing public by storm. According to many, Windows is one of the most significant software products to be developed for microcomputers, if not ***the most significant*****. Even though developers and users alike were entrenched in using DOS and DOS based applications, Windows offered the end-user the control interface that had been missing since the invention of the PC. Within a very short period of time, corporate America, as well as most other developers, recognized the need to not only develop new and exciting software for this environment, but also to** ***port*** **existing DOS based applications into Windows. But, there already existed a large installed base of applications that ran under DOS. Microsoft provided for this by allowing Windows to create DOS virtual machines (VMs) so existing DOS applications could still be used while maintaining the Windows graphical interface, functionality and Windows applications.**

A few evangelists started proclaiming that the future of DOS is limited and that it will only be a matter of time before DOS is completely eliminated and replaced by Windows. This may be true, but not in the foreseeable future. Microsoft's implementation of Windows 3.0 was not without problems, however, and a lot of those problems centered around handling DOS applications in the DOS Virtual

Machine (VM). Windows 3.1 has become even more popular because of its major improvements over version 3.0. These improvements do have their price though, in a more complicated environment. No longer is the system configuration maintained in just the CONFIG.SYS and AUTOEXEC.BAT files. DOS applications and those who support them, must also survive in the Windows environment.

Early in 1993, Microsoft announced their latest Disk Operating System - - DOS 6.0. This version of DOS is very similar to DOS 5.0 simply because the base functionality of DOS 6.x is built on DOS 5.0, but that's where the similarity ends. DOS 6.0 became a full blown package of utilities and applications. DOS 6.0 introduced better memory management, a utility called MemMaker that is designed to take the guesswork out of establishing memory management and optimization to get the best possible use of the HMA and UMA. Many users, including this author, feel that the combination of MemMaker and EMM386.EXE can do as good a job as other memory managers, except in the case of *stealth* . The *stealth* ability to move ROM shadowing to extended memory provides more UMBs for TSRs and other drivers. Microsoft's implementation does not offer the stealth capability, but not everybody really needs it. Given the price, one would be very hard pressed to surpass the Microsoft memory manager on a cost -performance basis.

The addition of a hard disk compression utility included with the operating system was a unique concept, to say the least. Notwithstanding the excellence of the concept the original implementation left a lot to be desired. Disregarding the recent court battles and settlements, DblSpace and DrvSpace are ideal disk compression utilities. Again, Microsoft provides the best bang for the buck although not necessarily the most optimal when compared to other disk compression providers.

Other additions that Microsoft has included in the latest release of DOS, include a system information/diagnostic report generator called MSD (MicroSoft Diagnostics). This tool is also included with releases of the Windows software, which Microsoft also touts as being an *Operating System* . (We will save discussion on what makes an operating system for a later date.) Multiple configuration support coupled with the different start up options, single-step and clean booting, and the CHOICE command, round out the top of the list of improvements made in DOS 6 since DOS 5.

It must be noted that most of the changes, at least the major changes, weren't made to the Operating System itself, but to the utilities and support modules for users to better use the existing operating system. Does this mean that DOS has reached the end of the line as far as core development goes? More than 5 years ago, Microsoft designed a new operating system - DOS. That operating system has flourished and matured and fulfilled the design criteria. Some statistics show

that 80% of the personal computers in use today, sport the MS-DOS operating system. What more can be done? Inducing from the plans released by Microsoft, Novell and IBM for the software to be released within the next 6 to 12 months, it certainly doesn't appear that much serious development will be done on DOS as we know it.

However, DOS isn't dead, and it won't die from lack of use either. There is still a very large installed base of DOS applications out there to take into consideration. Latest reports show that Novell may not make any more versions of NW-DOS, formerly DR-DOS. Speculation has it that there just isn't any more that Novell wants to do, so there isn't a need to do it. With the IBM PC-DOS 6.x versions, very little has been done to the operating system itself, with most changes being to the utilities. No new PC-DOS versions are on the immediate horizon. The next "planned" version from Microsoft will probably be MS-DOS 7.0 and won't be available until after *WINDOWS 95* is released.

According to published information from Microsoft, *WINDOWS 95* will fully support the current implementation of the DOS and the DOS memory structure. *WINDOWS 95* allows for an improved DOS VM as well as being able to create a stand-alone "real" DOS environment, which removes Windows completely. "Support for MS-DOS based applications, device drivers, and TSRs does not go away in Chicago. In fact, Chicago offers better compatibility for running MS-DOS based applications than Windows 3.1 does, including applications that are hardware-intensive, such as games." See *Microsoft® Windows™ "Chicago" Reviewer's Guide*

So what are these improvements? Let's say that DOS itself hasn't changed, as we have seen from the past, but the environment in which DOS and DOS applications run certainly has. According to Microsoft, the improvements in *WINDOWS 95* provide the following benefits from running DOS based applications in the Windows environment:

- Zero conventional memory footprint for protected-mode components
- Improved compatibility
- Improved robustness
- Better support for running DOS based games in a window
- Support for running existing DOS based applications without exiting *WINDOWS 95* or running MS-DOS externally
- Consolidated attributes for customizing properties of DOS based applications
- Toolbar access to features and functionality to manipulate the window environment while running DOS based applications

- **User scaleable window by the use of TrueType fonts in the DOS window**
- **Ability to gracefully end DOS applications without exiting the application**
- **Ability to specify local VM environment settings on a *per-application* basis**
- **Support for *new* DOS commands, providing tighter integration between the DOS command line and the Windows environment**

Zero Conventional Footprint Components

***WINDOWS 95* helps to provide the maximum amount of conventional memory available for running existing DOS based applications by providing 32-bit protected-mode components to replace many of the current 16-bit real-mode counterparts. The most notable sets of components include Microsoft and Novell network client software, CD-ROM drivers and extensions, Adaptec SCSI driver ASPI4DOS, SmartDrive disk caching software and the Microsoft Mouse driver. This can result in quite a remarkable memory saving that can be *over* 225Kb!**

Improved Compatibility

The goal of *WINDOWS 95*, ambitious as it may seem, is to support running DOS based applications better than is currently done in Windows 3.1. Not only to be able to run the existing "clean" applications that run flawlessly today, but also to support running the "bad" applications that tried to take over the hardware or required machine resources that aren't available under Windows 3.1.

Many things are being done to provide for better support, including better virtualization of system resources such as timers and sound device support which is needed for today's interactive games and multimedia processing. The use of 32-bit protected mode drivers which in turn freed up more conventional memory, allows a class of memory-intensive applications to run properly under *WINDOWS 95*.

To provide support for the most intrusive set of applications that only work under a unique DOS configuration that requires 100% access to the system components and resources, *WINDOWS 95* can provide "*Single MS-DOS application mode*." Entering this mode is like exiting Windows 3.1, then running the specified application under DOS, and then returning to Windows. In this mode, *WINDOWS 95* removes itself from memory except for a small stub, and provides the application with full access to all the resources of the computer.

Improved Support for Graphic-intensive Applications

Rather than forcing graphic applications to run in full-screen mode as in Windows 3.1, DOS based applications that use VGA graphic video modes will be

able to run in a DOS window. However, for the best level of performance, full-screen mode will still be the mode of choice.

Improved Memory Protection

Windows 95 will employ a *“global memory protection ”* attribute in the Program Properties that will allow the DOS system area to be protected from errant DOS based applications. When this attribute is set, the DOS system area sections are read-protected so that applications can’t write into this memory area and corrupt the DOS based application, support area or DOS based device drivers. In addition to the system area protection, enhanced parameter validation is performed for file I/O requests issued through the DOS INT 21h function.

DOS *“Defaults”* Changed

DOS applications running under *Windows 95* default to running in a Window with background execution set. With Windows 3.1, the Program Information File (PIF) had to be modified to run a DOS application in a Window or in the background. This change was incorporated to provide a more seamless integration between running DOS based applications and Windows-based applications.

Consolidation of Customization of DOS-based Application Properties

With Windows 3.1, the PIF Editor application was needed to create or change any property associated with running DOS-based applications. Numerous problems were encountered, including difficulty in accessing the PIF editor or changing PIF settings; the disassociation of the properties in a PIF file from the DOS-based application; the lack of a single location for storing PIF files other than placing them all in the Windows directory; and as previously discussed, the less-than-intelligent defaults for running DOS-based applications.

Windows 95 enhances the ability to define and change properties for running DOS-based applications by consolidating PIF files into a single location - the PIF directory within the directory where *Windows 95* is installed. Access is accomplished swiftly by using the secondary mouse button to simply click the icon or application window. The property information and the simplified user interface provides better organization of property settings through the use of property sheets.

MS-DOS Window Toolbar

Many Windows based applications provide a toolbar to implement quick access to common features and functions of a product. *Windows 95* extends this simplicity and power to making it easy to access functionality associated with an DOS application, including:

- cut, copy and paste buttons
- easy access to switching from windowed to full-screen mode
- quick access to the property sheets
- access to DOS VM tasking properties such as exclusive or foreground processing attributes
- easier access to font options for displaying text in a windowed DOS VM.

User-Scaleable MS-DOS Window

Since *WINDOWS 95* supports the use of a TrueType font in a windows DOS VM, the DOS window can then be scaled to any size, when the font size is set to “Auto”. With this setting, the DOS window is sized automatically to display the entire window within the user-specified area.

Local Virtual Machine Environment Settings

Under *WINDOWS 95*, a batch file can be optionally specified for a given DOS-based application allowing customization of the DOS VM on a local basis before running the application. This is like have a separate AUTOEXEC.BAT file for different DOS-based applications, which can set environment variables and install TSRs for the local VM only.

Support for Universal Naming Convention(UNC) Pathnames for Network Access

The *WINDOWS 95* shell allows users to browse and connect to network servers without mapping a drive letter to the network resource. *WINDOWS 95* supports the same functionality at an DOS command prompt and allows for using the contents of shared directories on servers from both Microsoft Network servers and Novell NetWare to:

- View the contents of shared directories

  dir \\server\sharename[\pathname]

- Copy files from shared directories

  copy \\server\sharename\pathname\file *destination*

- Run applications from shared directories

  \\server\sharename\pathname\filename

New MS-DOS Commands

The MS-DOS command processor and utilities have been enhanced to provide better integration between DOS functionality and the Windows environment.

- **Not only can DOS applications be run in a DOS window, but by using the start command, a user can start a Windows-based application from the command prompt.**

  Syntax: start <*application name*> | < *document name*>

  - √ **Start an application by specifying the name of a document to open, and *Windows 95* will launch the application associated with the given file type. For example, a user types "start myfile.doc" and the application associated with the file specification will start, assuming there is a valid association.**
  - √ **Start another DOS-based application in a *different* MS-DOS VM instead of the current VM. For example, to start a telecommunications session called TELECOM in another DOS VM, type "start c:\comm\telecom.exe" at the command prompt. A new DOS VM will be initiated and TELECOM.EXE will be executed.**
  - √ **Start a *Windows-based application* from an DOS command prompt by simply typing the name of the application. For example, to start Word for Windows, simply type WINWORD at the command prompt. This would be the same as typing "start winword".**

- **Commands and utilities that manipulate files have been extended to include support for long file names (LFN).**
  - √ **The dir command has been extended to show long file names in the directory structure, along with the corresponding DOS filename construct, commonly referenced as the *8.3 filename*.**
  - √ **The dir command has a verbose /v switch to provide additional file details.**
  - √ **The copy command has been extended to allow copying of long file names to/from short/long filenames. For example, typing: *copy myfile.txt "this is my file"* will create a new file with a long file name.**

So what happens to DOS and DOS applications when *Windows 95* becomes reality? Microsoft has pledged a commitment to continue to support DOS in the future. However, as history has shown, it doesn't necessarily follow that there will be any improvements or enhancements to MS-DOS itself, only to its surrounding environment. This means, that DOS will continue to exist tomorrow as it does today, but getting there will be different. Today, PCs boot into DOS and run Windows. Tomorrow, PCs will boot into Windows and run DOS.

Paul Williamson is an on-site consultant to Chase Manhattan Bank, a software developer and the RIME host for the DOS conference. He is a member of the editorial board of WindoWatch.

NEXT

ww