MICROSOFT FILES History 197

Table of Contents

The Story Begins

JAN 1975 MITS introduces the 8080-based Altair computer.

The story of MS-DOS begins in a hotel in Albuquerque, New Mexico. In 1975, Albuquerque was the home of Micro Instrumentation Telemetry Systems, better known as MITS. In January of that year, MITS had introduced a kit computer called the Altair. When it was first shipped, the Altair consisted of a metal box with a panel of switches for input and output, a power supply, and two boards. One board was the CPU. At its heart was the 8-bit 8080 microprocessor chip from Intel. The other board provided 256 bytes of memory. The Altair had no keyboard, no monitor, and no permanent storage. But it had a revolutionary price tag. It cost $397. For the first time, the term "personal computer" acquired a real-world meaning.

The real world of the Altair was not, however, the world of business computing. It was primarily the world of the computer hobbyist. These first users of the microcomputer were not as interested in using spreadsheets and word processors as they were in programming. Accordingly, the first software for the Altair was a programming language. And the company that developed it was a two-man firm, in Albuquerque, called Microsoft.

FEB 1975 Microsoft sells first BASIC to MITS for Altair computer.

The two men at Microsoft were Paul Allen and Bill Gates. Allen and Gates had met when they were both students at Lakeside High School in Seattle, where they began their computer-science education on the school's time-sharing terminal. By the time Gates had graduated, the two of them had already founded a company called Traf-O-Data. Traf-O-Data's main product was a machine based on the 8008 microprocessor, which was the predecessor of the Altair's 8080 chip. The Traf-O-Data machine automated the process of reading 16-channel, 4-digit, binary-coded decimal (BCD) tapes generated by traffic recorders to monitor traffic flow. Unfortunately, the company faced tough competition from state-owned—and therefore state-subsidized—processing facilities. Nevertheless, the experience with Traf-O-Data and the 8008 microprocessor gave Allen and Gates a running start when the Altair was introduced. They used their running start to write a version of BASIC for the 8080 and sold it to MITS. It was Microsoft's first product.

Originally, the only way to enter Microsoft's BASIC into the Altair was to flip the switches on the front panel to enter the machine code into the computer's memory. This process was soon automated by paper tapes that could store and load the code. But Allen, who had taken a position as Director of Software at MITS, quickly realized that a disk drive was the real solution to entering and storing code. So in February 1976, he asked Gates if he could write a disk-based version of BASIC. Since the Altair had no operating system and hence no method of managing files, the disk BASIC would also have to include some file-management routines. In other words, it would have to function as a rudimentary operating system.

FEB 1976 Gates writes first Disk BASIC.

That's when Gates, who was still a student at Harvard, flew to Albuquerque, checked into the Hilton Hotel with a stack of yellow legal pads, and asked not to be disturbed. Five days later, he checked out of the hotel, yellow pads filled, and started typing code into a DEC PDP-11 mainframe, on which time was leased from the Albuquerque public school system. After five days, Disk BASIC was up and running on the Altair.

Disk BASIC used a file-management strategy based on a file allocation table. The more conventional approach to file management was to include disk-allocation information with each file. In Disk BASIC, all of this information was kept in one location: the file allocation table. Gates chose this strategy for its speed. A few years later, when Microsoft wrote a stand-alone version of BASIC for the new 16-bit 8086 chip, they used the same approach. The file-handling routines in stand-alone Disk BASIC became, in turn, the model for the operating system that would eventually be known as MS-DOS.

But in 1976, people weren't thinking about 16-bit operating systems. They were thinking about 8-bit operating systems. Both users and developers were beginning to recognize the limitations of running applications on top of BASIC or some other language. MITS eventually used the code from Gates' Disk BASIC to build an operating system for the Altair. Another company, Digital Research, headed by Gary Kildall, had developed an operating system called Control Program for Microcomputers, or CP/M, a couple of years earlier. As new companies began to compete with the Altair for the microcomputer market, some of them sold CP/M as their operating system. But most introduced their own operating systems. The result was chaos.

> *"There was such a tower of Babel in operating systems. But then Lifeboat Associates came along and started actually offering software in CP/M-80 format. It was probably Lifeboat Associates more than anybody who really got things going."*
>
> —Bill Gates

JUL 1977 Microsoft introduces FORTRAN for 8080 computers.

At Microsoft, Gates and Allen continued to focus their attention on programming languages. They developed different versions of BASIC to run on the various operating systems that were appearing. They also introduced versions of COBOL, FORTRAN, and Pascal. When Lifeboat Associates, a software distributer in New York, began offering a line of software to run on CP/M-80, Microsoft was ready with FORTRAN and COBOL. With this distribution channel opened, several major manufacturers of microcomputers, including Northstar Computers and Processor Technology, decided to offer CP/M as their primary operating system, and an 8-bit standard began to emerge.

FALL 1977 **Microsoft introduces COBOL for 8080 computers. Microsoft licenses BASIC to Apple Computer and Radio Shack. Microsoft sells first BASIC in Japan.**

Although Microsoft was devoting most of its resources to developing languages, it was not ignoring operating systems altogether. In fact, the company wrote its own 8-bit operating system, which it called M-DOS. M-DOS was a true multitasking system, with good performance and an advanced disk structure. But it was large and pushed the 8-bit chip to its limits. And since CP/M had already won the majority of the 8-bit market—in large part, with the help of Microsoft's languages—Microsoft eventually turned its attention to UNIX, a multitasking, multiuser operating system that was beginning to achieve the status of an industry standard on larger computers. In early 1980, Microsoft licensed UNIX from Bell Labs, and a team of Microsoft programmers ported UNIX to several microprocessors under the name XENIX.

Microsoft recognized that the market for a multitasking, multiuser system like XENIX was significant. Yet two years earlier, the company had already begun to define another market for itself—the market for single-user automated workstations. In the late 1970s, this market was more vision than reality. And the vision was not particularly obvious in a world where, for most people, office automation meant dedicated word processors. The primary users of microcomputers were still hobbyists with their computers tucked away in dens, attics, and basements.

APR 1978 **Intel introduces the 16-bit 8086 chip.**

Then, in 1978, Intel introduced the 16-bit 8086 chip. The new chip was a giant step forward in performance and memory capacity. When Intel released the 8080 chip in 1974, the market for microcomputers was nonexistent. The developers believed they were designing a chip to make household appliances or industrial machines more intelligent. By the time the 8086 chip was developed, however, the microcomputer was a market reality, and the new design reflected the growing importance of this market.

Microsoft saw the 8086 as an opportunity to be a pioneer. The company had already developed and marketed all of the major languages as well as a linker and an assembler for the 8080 and Z80 systems. It had produced versions of BASIC for all of the major 8-bit processors on the market. As Gates and Allen looked at the future of Microsoft, they could see two possible directions. The first was to continue in the same direction and develop the remaining languages—FORTRAN, COBOL, and Pascal—for other 8-bit processors. Or they could turn to 16-bit technology.

The first alternative was not a pioneering effort. Gates and Allen did not really want to rewrite FORTRAN, COBOL, and Pascal to run on other 8-bit systems. But they also couldn't ignore the earning potential of this strategy, particularly with the growing popularity of the Apple II based on the 6502 microprocessor. Allen found the solution: turn the Apple II into a Z80 system. To accomplish this metamorphosis, Microsoft developed the SoftCard. The SoftCard was a Z80 microprocessor and a ROM version of the CP/M BIOS (Basic Input/Output System) on a plug-in card, with CP/M itself on disk. Microsoft licensed CP/M-80 from Digital Research for a fixed, one-time fee of $46,000. With CP/M-80 on the SoftCard, Apple users could run any of the programs and languages designed to run on the CP/M machines, including all of Microsoft's languages. Eventually, CP/M-80 on the SoftCard became the most popular version of CP/M-80 on any computer.

Gates and Allen were more interested in the second alternative: the 16-bit alternative. Their first step was the same as Microsoft's first step with the Altair. They developed another stand-alone version of BASIC. But this time, it was for the 8086.

APR 1979
Microsoft moves to Bellevue, Washington.

While Microsoft was working on its stand-alone BASIC, a man named Tim Paterson was working at a company called Seattle Computer Products. Seattle Computer Products built memory boards, but Paterson had become interested in the 8086 chip and was developing an 8086 card for use with an S-100 bus machine like the original Altair. In May 1979, Paterson called Microsoft. He wanted to know if Microsoft was developing any 8086 software that he could use to test his board. Microsoft told him that they had an 8086 BASIC—their stand-alone BASIC—ready to try.

MAY 1979
Paterson goes to Microsoft to test his 8086 card.

Microsoft had moved to Bellevue, a suburb of Seattle, in April 1979. So Paterson just drove across the bridge over Lake Washington to Bellevue with his new board in a Chromemco Z-2 box. Within a week, BASIC was running on Seattle Computer's board.

JUN 1979
Microsoft shows 8086 stand-alone BASIC on Paterson's machine at NCC.

The following month, June 1979, Microsoft went to the National Computer Conference in New York. Lifeboat Associates was now selling all of Microsoft's software and had invited Microsoft to share their small booth at the show. Microsoft, in turn, invited Paterson to join them there to show 8086 BASIC running on the S-100 8086 hardware.

At the conference booth, the discussion turned to operating systems and file management, and Paterson got an inside look at how Microsoft was handling files in stand-alone BASIC. He also learned about the internals of M-DOS. Paterson was impressed with Microsoft's approach and tucked the conversation away in his memory for future use.

FALL 1979 Microsoft contracts with Convergent Technologies to develop FORTRAN for 8086 machine. Digital Research announces plans to release CP/M-86.

By the fall of 1979, the 8086 chip was beginning to attract interest from many quarters. Microsoft, for example, found itself with a contract from Convergent Technologies to develop an 8086 version of FORTRAN. Digital Research had announced a planned release of CP/M-86—the 8086 version of its operating system—for December 1979. Still, not everyone agreed with Microsoft that 16-bit technology was the logical development path for microcomputing. A number of voices in the trade press warned that the industry investment in 8-bit equipment and software was too great to introduce a new standard.

"The implementation of microprocessors with completely different instruction sets, such as 16-bit units, creates major headaches for companies with large investments in proprietary software, such as word processing, BASIC interpreters, and operating systems."
—*InfoWorld*, December 22, 1980

At Seattle Computer Products, however, Tim Paterson continued to work on the 8086 board. When CP/M-86 was still not available in April 1980, Paterson decided it was time to take the matter of a 16-bit operating system into his own hands. He began to work on the system that would become MS-DOS.

APR 1980 Paterson begins work on an 8086 operating system.

Paterson recognized that a standard operating system was crucial if users were going to be assured of a wide range of applications software and languages. He also recognized that CP/M had already become a de facto standard. So translation compatibility—the ability to mechanically translate existing CP/M-80 applications to run on his 16-bit system—became a major goal in his effort. His 86-DOS, as it was called, mimicked CP/M-80's functions and its command structure, as well as its file control blocks (FCBs), its program segment prefix (PSP), and its approach to executable files.

However, Paterson had some complaints about CP/M, particularly about its approach to error handling. He was also dissatisfied with the file allocation system in CP/M, which was inefficient in its use of space as well as its speed of operation. He remembered how Microsoft had handled files in stand-alone BASIC and M-DOS, which was how Bill Gates had handled them in that first version of Disk BASIC, written in the hotel in Albuquerque. The file allocation table (FAT) became the core of his file-management system.

Paterson bootstrapped his way to an operating system by first writing a translator that translated 8080 code to 8086 code. He then wrote an 8086 assembler in Z80 assembly language and used the translator to translate it. It took him four months to build a working version. When it finally ran, 86-DOS was just 6K bytes. Anyone who wanted to run a CP/M-80 program on the 8086 could use the translator he had built to translate the source code and be reasonably certain it would run under Paterson's operating system with little or no editing.

Once 86-DOS was running, Paterson again went to Microsoft, this time to ask them to write a BASIC to run on his system. It was September 1980.

AUG 1980 Microsoft introduces SoftCard for Apple II.

While Paterson was developing 86-DOS, things were stirring in a Boca Raton research facility for IBM. IBM had, to all appearances, ignored most of the developments in microcomputing. But now, within the company that had made computing an industry, several groups were competing for the right to develop an IBM computer based on a microprocessor. The product was to be aimed at the market that IBM knew already: the business market. But IBM was unsure of both the microcomputing technology and the microcomputing market. And with its history of long development cycles, it was also unsure of its ability to develop a machine and accompanying software in the competitive time frames that were inevitable in the rapidly changing world of microcomputing.

AUG 1980 IBM's Boca Raton study group visits Microsoft.

So a study group at Boca Raton, headed by Jack Sams, began to look around for people who looked like they knew what they were doing in the field. One of the places they looked was Microsoft.

The Boca Raton group first visited Microsoft in August 1980. They told Microsoft that IBM wanted to get involved in the "low end." They knew that the low end was very fast moving and that they couldn't afford their typical five-year design cycle to build a new machine. But they also had a strategy: They would use off-the-shelf products. All of the hardware they needed was available. The problem was the software.

AUG 1980 Microsoft signs consulting agreement with IBM.

The group that visited Microsoft wanted to know: If IBM gave Microsoft the specifications for an 8-bit computer, could Microsoft write a ROM BASIC for it? Could they do it by the following April?

Microsoft said yes, but why not consider a 16-bit machine?

After the meeting at Microsoft, Jack Sams and his study group went back to IBM with the proposal for Project Chess, a low-end, 16-bit workstation.

SEP
1980
IBM asks Microsoft to develop BASIC, FORTRAN, Pascal and COBOL for its new machine.

Sams returned to Microsoft a month later and asked if Gates and Allen could also provide FORTRAN, Pascal, and COBOL in the same time frame—namely, before April of the following year. Gates explained to Sams that, to run those languages, they would need an operating system. He explained that BASIC was unique because it had been designed as a stand-alone product. Microsoft did not have a stand-alone FORTRAN or COBOL. Gates suggested that Sams call Digital Research, since he knew that CP/M-86 was in the works there. In fact, Gates placed the initial phone call to Digital Research for IBM.

"They didn't understand that our BASIC was kind of funny in a way. Stand-alone BASIC worked. But we said: you've got to have an operating system."
—Bill Gates

The IBM team went to Digital Research the next day to talk about operating systems, but the people at Digital Research were nervous about IBM's nondisclosure requirements, and the meeting never got off the ground. In the weeks that followed, IBM continued to communicate with Digital Research, but without much success.

Meanwhile, Microsoft was reassessing its own goals and resources. Gates and Allen wanted to write the languages—all of them. In order to do that in six months, they needed some assurances about the operating system that IBM was going to use. They needed to know that it would be ready on time. They also needed to know that they would be able to get information about some of its internals, since the ROM BASIC would have to interact intimately with the Basic Input/Output System (BIOS). Digital Research was already almost a year behind its own announced schedule for release of CP/M-86, so Gates and Allen began to wonder if Digital Research could provide the system on time. They also had heard rumors that Digital Research was buying a BASIC from one of Microsoft's competitors to bundle with its operating system. Such a move would make things very difficult for Microsoft in the BASIC market. All of these considerations made CP/M-86 seem less desirable as the operating system for the IBM product.

SEP
1980
Paterson shows Microsoft his 86-DOS.

Then, Gates and Allen started looking at the figures. The proposal for the four languages and an assembler and loader came to 400K of code. They estimated that to add an operating system would require only another 30K or so. They had also just seen Paterson's rudimentary operating system for the 8086, designed to be compatible with CP/M applications, and knew exactly where it stood.

SEP 1980
Microsoft decides to write MS-DOS.

It was Sunday night, September 28, 1980. Gates and Allen were in Gates' office on the eighth floor of the Old National Bank building in Bellevue. They were reviewing all the facts with Kay Nishi, a Microsoft vice president and president of ASCII Corp. in Japan, who was preparing the proposal for IBM.

OCT 1980
Microsoft buys rights to 86-DOS.

The more they talked about it, the more it seemed possible—even preferable—to write their own operating system. Finally Nishi got up and said, "Let's do it! We've got to do it!"

> *"People take it for granted that IBM was going to be big in the business and that you should follow when they command, but there were a lot of smart companies doing small computers, and it wasn't clear that IBM was going to have the impact that it eventually did. We thought it was worth pursuing, but it wasn't like today where it seems so obvious in retrospect."*
>
> —Bill Gates

OCT 1980
Microsoft submits proposal to IBM for 400K of code—four languages, an assembler, a loader, and MS-DOS.

Allen's first step was to contact Tim Paterson at Seattle Computer Products. He told Paterson that Microsoft wanted to develop and market the operating system and that he had an OEM customer for it. SCP, which was not in the business of marketing software, agreed. Microsoft paid SCP a fee for ownership 86-DOS. SCP also received a license to use and sell Microsoft's languages and any 8086 versions of the operating system.

With 86-DOS in hand, Microsoft submitted its proposal to IBM in October 1980. The proposal included Microsoft's plan for developing the operating system. They told IBM that they wanted to do the entire package. The time frame, they explained, was so short and the boundaries between the languages and the operating system were so unclear that Microsoft needed to control the development of the operating system if they were going to guarantee delivery by spring of the following year.

NOV 1980
Microsoft signs contract with IBM.

In November, IBM signed the contract.

THANKSGIVING 1980
Microsoft receives first prototype of IBM PC.

A prototype of the IBM machine arrived at Microsoft at Thanksgiving, and a small DOS team began working long, tedious, and sometimes chaotic hours. The immediate challenge was to bring up the rudimentary operating system they had bought from SCP on the new machine in an environment of constant change. The hardware was changing. At SCP, Paterson was continuing to modify the operating system. And Microsoft was changing its own specifications for the operating system as it developed.

The first problem was getting the copy of the operating system that they had gotten from Seattle Computer compiled and married to the BIOS, which Microsoft was also helping IBM write. The media complicated this task. 86-DOS was on an 8-inch floppy disk. The IBM machine used a 5 1/2-inch disk.

"If I was awake, I was thinking about the project."
—DOS programmer

To Microsoft fell the task of determining the format of the new disk and then finding a way to get the operating system, which was written in assembly code, from the old format to the new. The work went like this: They would move sections of code from the 8-inch disk, and compile it, then convert it to Intel HEX format, upload it to the DEC-20, and then download to an Intel development machine. They also developed the BIOS on the DEC-20 and downloaded it to the Intel machine, converted it to HEX, moved it to the IBM development system, and then crossloaded it over to the prototype.

FEB 1981 MS-DOS runs on IBM PC prototype for the first time.

Dependence on the physical characteristics of the machine added to the tedium of the task. Paterson's goal for 86-DOS was logical device independence. But at the time that Microsoft began to work with the system, it still worked with fixed physical sectors that were 128 bytes long. The IBM prototype had 512-byte sectors. So Microsoft had to convert the operating system to handle logical records that were independent of the physical record size.

Paterson eventually improved the operating system's logical independence by adding functions that streamlined the reading and writing of multiple sectors and records as well as records of variable size.

In addition to making his own changes, Paterson worked closely with the Microsoft developers as they got the operating system up and running. Microsoft would give him dozens of changes that they wanted to implement—from changes in the initial messages that appeared when MS-DOS started up to changes in EDLIN, which was the editor that Paterson had written for his own use. Throughout this process, however, Microsoft could not tell Paterson anything about the system except that it was an OEM machine. Under IBM's security restrictions, Microsoft certainly couldn't show him the prototype. So Paterson was working blind. The only serious clue he had about the identity of Microsoft's customer was a phone call. One day, someone from IBM called SCP with some questions about DOS. A little surprised, the people at SCP asked again who the person was. The caller replied, "Oh, never mind," and hung up.

APR
1981
Paterson joins Microsoft.

Paterson found out the truth in April when he left Seattle Computer and joined Microsoft.

The secrecy requirements imposed by IBM created other strains on the project. All of the work had to be done in a secure room, which turned out to be about 10 x 6 feet. It had no windows, and the door was to be kept locked at all times to protect the secrecy of the project. IBM even requested that Microsoft install chicken wire above the ceiling tiles to secure the room from above. In this small, closed room was a large Intel hard-disk development system with an In-Circuit Emulator as well as two prototypes. The heat generated by these machines created frequent hardware malfunctions, not to mention programmer irritability.

Even without the heat, the hardware was not stable. The DOS team spent countless hours tracking down software problems that turned out to be hardware problems—like a serial card that would respond when it should not be responding or that would corrupt areas of memory.

The project encountered other hardware problems that slowed the development process. One of the most frustrating was a constraint in the BIOS. The BIOS could not transfer data across a physical 64K boundary. DOS would transfer a number of sectors from a certain address. If they crossed over the 64K boundary, the system would crash. The DOS team didn't realize this constraint until April.

JUN
1981
First rumors of the IBM machine appear in InfoWorld.

While IBM was a demanding customer in terms of schedule and confidentiality, Microsoft found the IBM team to be receptive and helpful throughout that early development process. They were buying Microsoft's expertise in an area where they themselves were uncertain. So they did not dictate. They supplied information. They responded overnight to hardware problems. And they listened.

The new operating system ran on the prototype for the first time in February 1981. In the months that followed, Microsoft worked feverishly to expand and refine it for use by a new class of users—for business managers and other office workers. IBM accepted the final product, and MS-DOS made its debut on the machine called the IBM Personal Computer.

The First Release

AUG
1981
IBM announces the IBM PC. Microsoft releases Version 1.0 of MS-DOS.

The world got its first look at Version 1.0 of MS-DOS when IBM introduced the IBM Personal Computer in August 1981. This first version had 4000 lines of assembly-language source code and ran in only 12K. It was organized into three separate files: The IBMBIO.COM file contained the disk and character input/output system. It interfaced with the BIOS that IBM had developed. The IBMDOS.COM file included the program interface, the disk-file manager, and a character I/O handler. The COMMAND.COM file was the external command processor.

MS-DOS was not the operating system that Microsoft wanted to build when it first thought of building a 16-bit operating system. Everyone at Microsoft saw it as a compromise. It was a compromise with time. More important, it was also a compromise with market demands that made program-translation compatibility from CP/M-80 the overwhelming design goal. The large base of CP/M software included the two most popular applications: WordStar and dBase II. Microsoft's own languages also ran under CP/M-80. MS-DOS, like 86-DOS, was designed to allow software developers—including Microsoft itself—to mechanically translate source code that ran on the 8080 to run on the 8086.

AUG
1981
DOS 1.0 inherits many features from CP/M.

So MS-DOS had to look a lot like CP/M-80. And it did. It inherited such features as the 8-character file names with 3-character extensions. It used the same conventions for identifying disk drives in its command prompts, and it used the same command syntax as CP/M-80. For the most part, it used the same command language, it offered the same file services, and it had the same general structure as the 8-bit standard.

"The challenge of 1.0 was just basically getting it done on time and making sure it was reliable and had the extra error-recovery that IBM wanted."

—DOS programmer

But although the people at Microsoft were aware of the constraints imposed by history, they also had a vision of the future. It was a future that would continue to bring rapid change in all of the technologies on which MS-DOS was dependent: disks and other media, memory boards, and microprocessors. The rapid development of these technologies promised to make increased computing power available to a growing number of users at a lower price. Yet it also created the potential for chaos in the world of software in general, and of operating systems in particular. Microsoft also realized that the market potential was vast and that IBM was not likely to be the only player in that market.

To be successful over the long term, Microsoft's new operating system would have to be flexible enough to accommodate change. Toward that end, Microsoft sought to make its operating system less dependent on any specific hardware configuration—that is, to virtualize its operating system in every way it could. The first steps were device-independent input and output, variable record lengths, relocatable program files, and a replaceable command processor.

AUG 1981 DOS 1.0 treats peripheral devices as files.

MS-DOS made input and output device independent by treating peripheral devices as though they were files. It assigned a reserved file name to each of the three devices it recognized: CON for the console (screen and keyboard), PRN for the printer, and AUX for auxiliary serial ports. If one of those names appeared in a file control block of a file named in a command, all operations were directed to the device named rather than to the disk file.

Device independence provided benefits for applications developers and users alike. For developers, it meant that application programs could use one set of read or write calls rather than using different calls for different devices. It also meant that no modifications of the application would be necessary as new devices were added to the system. For users, device independence meant flexibility. If a program were designed for disk I/O only, for example, the user could direct the output to the printer or use a file for input. The user could, in fact, use the devices with any command that specified a file name as an argument.

AUG 1981 DOS 1.0 uses variable record lengths.

Another step toward logical independence was variable record lengths. In CP/M-80, logical and physical record lengths were identical: 128 bytes. Files could be accessed only in units of 128 bytes. The IBM disk had physical sectors of 512 bytes. But with MS-DOS, a user never needed to know about this physical requirement since MS-DOS marked the end of the file. If the user opened a file with a logical length that differed from the physical record length, MS-DOS remembered exactly where the file ended—to the byte. Practically speaking, this logical independence meant that MS-DOS users could have files with different logical record lengths.

AUG 1981 DOS 1.0 defines relocatable program files.

Another new feature in MS-DOS that was designed to give it logical independence was the relocatable program file. MS-DOS could load two types of program files: .COM files and .EXE files. The .COM files mimicked the binary files in CP/M-80. They started at location 100H in memory segment in which they were loaded. All segment addresses referred to this starting location. The .EXE files did not set a physical starting location. They were loaded at the lowest available address. The command processor then used the load address in the file header information to relocate the segment addresses automatically.

> *"The basic difference between CP/M and MS-DOS was that CP/M had no distinction between a physical sector on the disk and a logical record of the program. MS-DOS had logical independence."*
>
> —Tim Paterson

AUG 1981 DOS 1.0 uses a replaceable command processor.

MS-DOS also tried to virtualize the user interface—the command processor—by making it a separate, relocatable file just like any other program. The COMMAND.COM file was designed as an executable file and could be replaced by any other command processor. For example, it might be replaced by a menu-driven command processor by simply naming the new command processor COMMAND.COM.

Building on the logical independence created by these features, MS-DOS could also claim some important strides forward in performance. The speed of MS-DOS operations was due in part to the cleverness of Tim Paterson in finding and taking advantage of hidden time gaps in operations. But more important were some fundamental design decisions. The most important of these was perhaps the file allocation table.

AUG 1981 DOS 1.0 incorporates the file allocation table—FAT.

The file allocation system that Gates had first implemented back in Albuquerque in 1976 made the allocation of disk space efficient and the search for long files very fast. CP/M-80 had used directory entries to store information about file locations. Each directory entry had a memory map, which was a list of sixteen 1K allocation units (groups of contiguous physical sectors) on the disk—where successive parts of the file were stored. CP/M might require several directory entries, or extents, to map the entire file if the file was larger than 16K. It would read the directory for the first extent, go load the blocks according to the memory map, then return to the directory on the disk and search for the next extent, read the memory map, and continue.

MS-DOS also allocated disk space in "allocation units." The use of these allocation units was mapped in one central file allocation table—or FAT—which was always in memory. MS-DOS also used a directory entry for each file, but the entry did not include the allocation map as in CP/M. Instead, it simply pointed to the first allocation unit in the file allocation table. Each entry in the table pointed to the next unit that was associated with the file. As a result, MS-DOS had a complete in-memory analysis of all file components, the file size, and all of the available disk space without having to access the disk.

When Paterson was initially developing the operating system at Seattle Computer, he made extensive calculations to evaluate the efficiency of different methods of file allocation with different states of the disks. He compared the amount of space used by the FAT to the amount required by CP/M-type directory entries as well as the time to analyze the information in these two different formats. The use of a file allocation table in memory proved to be the most efficient way of using disk space. For long files it was definitely faster: Even if MS-DOS had to search every allocation unit for the disk, it could complete the search faster than it could load the head on a disk to search for a single directory entry.

AUG 1981 DOS 1.0 reads and writes multiple sectors.

Two other important features—the ability to read and write multiple sectors and the transient use of memory by the command processor—provided more efficiency for both users and developers. The independence of the logical record from the physical sector laid the foundation for the ability to read and write multiple sectors. If an application were reading multiple records in CP/M, it had to issue a "read" function call for each sector, one at a time. With MS-DOS, it would issue one read function call, giving the operating system the beginning and ending records to read. MS-DOS would then load all of the corresponding sectors automatically.

AUG 1981 DOS 1.0 has a transient command processor.

> *"The code was really tight. There wasn't any fat in that original version of MS-DOS at all."*
> —DOS programmer

The other performance feature in version 1.0 was the transient command processor. When the Microsoft programmers designed the command processor COMMAND.COM, they wanted to include the most common user functions, such as DIR and COPY, within it. For speed and ease of use, they made these functions intrinsic to the command processor. The tradeoff was the larger size of the command processor. The user would have to pay a premium in memory for this speed and ease of use. To minimize the cost in memory, Microsoft designed the command processor in two parts: a resident part and a transient part. The added functions were placed in the transient part which could be overwritten by large applications programs. Then, when the user exited the application, the resident part of the command processor would reload the functions in the transient portion if necessary.

Beyond these structural features designed for future compatibility and enhanced performance, Version 1.0 of MS-DOS introduced several services

AUG
1981
DOS 1.0 improves error handling.

and built-in utilities to make life easier for users and applications developers. Chief among these were improved error handling, automatic logging of disks, date and time stamping of files, and batch processing of command files.

From the beginning, IBM had stressed the need for data integrity, and everyone recognized that loss of data is most likely to occur when the user encounters an error message and responds incorrectly. To reduce this risk, MS-DOS replaced the traditionally cryptic error messages with messages that were clear and complete. These messages were used consistently across MS-DOS functions and utilities. Microsoft encouraged developers to use the same messages in their applications when appropriate.

To further reduce risks to data, MS-DOS trapped hard errors, such as permanent disk errors, which had previously been left to the hardware-dependent logic to handle. Now the hardware logic could simply report the nature of the error to MS-DOS, which again would treat it in a consistent way. MS-DOS could also trap the Control-C key sequence, so that an application program could protect against accidental termination by the user or provide a graceful termination when appropriate.

AUG
1981
DOS 1.0 provides automatic date and time stamping.

To reduce errors and simplify use, MS-DOS also automatically updated information about the disk when it was changed. In CP/M, users had to log in a new disk each time they changed disks. This manual login was particularly cumbersome for users with single-disk systems or those who needed to store their data on several different disks. MS-DOS automatically logged new disks as long as no file was currently open.

AUG
1981
DOS 1.0 has batch processor.

Another innovation in MS-DOS was date and time stamping. From the very earliest prerelease versions, MS-DOS tracked the system date and displayed it every time a user started MS-DOS. So when it turned out that the operating system only used the first 16 bytes of a file control block for the file-header information, the DOS programmers decided to use the remaining 16 bytes for date and time information as well as information about the size of the file. This information was then automatically displayed with the file names by the DIR command. It was also available to the linker, so that when users were editing their programs, the linker would only recompile those relocatable files with earlier date and time stamps than the source code.

It was IBM's own internal testing needs, rather than considerations of the need for a programmer's tool, that led to the addition of another capability in MS-DOS. This was batch processing of files. IBM wanted to test various functions of the system by running scripts—or a sequence of commands or other operations—one after the other. To run these scripts, they needed an automated way of calling one routine after another. The MS-DOS batch processor solved this problem for IBM but also meant that users could take advantage of the speed batch processing.

AUG 1981 DOS 1.0 allows programs to be locked in memory.

Finally, MS-DOS increased the options available to a program when it terminates. In more rudimentary operating systems, applications and other programs remained in memory only as long as they were active. When terminated, they were removed from memory. MS-DOS added a Terminate But Stay Resident function that allowed the program to be locked into memory. So the program could, in effect, become part of the operating-system environment until the system was shut down. The program could also call another program when terminated, allowing programs to be stacked in memory.

"Even our most optimistic view of the number of machines using that thing really wouldn't have matched what really ended up happening."
—Bill Gates

It was inevitable perhaps that MS-DOS, when released, would be compared with CP/M-80 and eventually with CP/M-86. The main issue was compatibility: To what extent was MS-DOS compatible with the existing standard? Few people imagined that MS-DOS was bound to replace that standard within a very short time.

Player," presumably in anticipation of a vast inventory of CP/M applications for the IBM PC. They led their readers to assume that the IBM PC was a CP/M machine.

MAR 1982 Lifeboat Associates offers CP/Emulator for IBM PC. Vendex Corporation introduces Baby Blue expansion board to run CP/M-80 software on the PC.

Confusion also arose around the name. When Paterson wrote the first version of the operating system at Seattle Computer Products, he called it 86-DOS. When Microsoft began developing the system for IBM, they called it 86-DOS for a while, but by the time the PC was ready for the market, everyone at Microsoft—and at IBM—was calling it MS-DOS.

"IBM is in for some rough competition. Xerox, which announced its model 800 personal computer last month, is expected to give IBM the biggest headache in the small-computer competition, but TAB Office Products, Burroughs, and Lanier Business Systems are all working on similar products to announce later this summer. None of these firms share IBM's hesitancy to use Digital Research's CP/M as an operating system, so IBM may find itself with a competitive system, but little or no software to compel potential buyers."
—*InfoWorld*, July 20, 1981

Then IBM decided to call it IBM Personal Computer DOS, and the press was quick to call attention to the fact that "PC-DOS" had some features that MS-DOS—the generic system that Microsoft was licensing to other manufacturers—did not. These were primarily a few utilities that IBM added. To add to the confusion, when Lifeboat Associates agreed to help promote MS-DOS, they decided to call it Software Bus 86, because they had a line of Software Bus products, including SB-80 which was their version of CP/M-80. Since Software Bus was a trademark for Lifeboat's product line, Microsoft could not call their own product SB-86, even though it was identical to the product that Lifeboat would be selling. To further complicate matters, some of the first companies that licensed MS-DOS wanted to use their own names. So Compaq called it Compaq DOS, and Zenith called it ZDOS.

Microsoft recognized that there is indeed something in a name, particularly when you want to build an industry standard. The company finally insisted that everyone call the operating system MS-DOS. Eventually, everyone except IBM did.

IBM's persistence in calling their operating system simply DOS gave the trade press grounds for speculating that even if the IBM PC were wildly successful, its success would not guarantee wild success for MS-DOS. As late as the end of 1983, many industry watchers were still doubters.

MAR
1982
Microsoft releases FORTRAN for MS-DOS.

Microsoft faced the same dilemma as its customers in the software-development community. MS-DOS represented only a small fraction of its own revenues. Most of its products were languages and applications. For two years after MS-DOS was introduced, the growth of CP/M-86 and its look-alikes nearly paralleled that of MS-DOS. Microsoft found itself in the position of selling languages to run on CP/M-86 as well as MS-DOS, thereby contributing to the growth of software for its competitor.

"It's PC-DOS that dominates. . . . I feel it's very logical that IBM will offer versions of PC-DOS that will differ from the current PC-DOS and they will not necessarily be purchased from Microsoft. I see a turnkey specific market that will require things that Microsoft is not interested in doing."

Jean Yates
—*InfoWorld*,
September 26, 1983

If the software developers were confused by the press, they also didn't get much guidance from the hardware manufacturers. From the start, many of the software developers took a "wait and see" attitude, and what they were waiting to see was what the hardware manufacturers would do. The hardware manufacturers, in turn, were often confused by the issue of compatibility.

MAR
1982
MS-DOS software appears for the first time in InfoWorld's software review column.

Microsoft had always had translation compatibility between CP/M-80 and MS-DOS software as a goal. But many people misunderstood the meaning of compatibility in operating systems. Compatibility did not mean that software designed to run on the 8080 or Z80 microprocessor could be run directly, without adaptation, on an 8086 microprocessor, regardless of the operating system. In the gloss of compatibility, a lot of people expected that CP/M-86 would automatically make the installed base of CP/M-80 software available to the IBM PC and other 16-bit computers.

"We were in a very serious competitive situation, with MS-DOS vs. CP/M in those days. I mean, very serious."

—Bill Gates

APR
1982
IBM releases CP/M-86 for the IBM PC.

At the same time, some developers of both software and hardware believed that it would be easier to transport their CP/M-80 software to a CP/M-86 environment than to an MS-DOS environment. In fact, MS-DOS provided better support for CP/M-80 calls than CP/M-86 did. Furthermore, mechanical translation applied only to applications written in 8080 or Z80 assembly-language source code. Much of the installed base of 8-bit software could not be mechanically translated to either CP/M-86 or MS-DOS because it was written for other processors, such as the 6800 or 6502. Many applications were not written in assembly language at all, but were written in a high-level language. Most of those high-level language programs were written in Microsoft language, and all of Microsoft's languages ran on MS-DOS. This was the story that Microsoft had to tell over and over as they worked to promote MS-DOS.

APR 1982 Microsoft releases its COBOL compiler for MS-DOS.

In the United States, IBM's market force helped sell MS-DOS to some computer manufacturers. But in the beginning, IBM did not ship to Europe, and so Microsoft was on its own there. A key victory for Microsoft was its agreement with Victor, a company that was very successful in Europe. Initially, Victor had licensed CP/M-86, but Microsoft worked closely with Victor, providing special development support, and finally convinced the company to offer its networking and languages only on MS-DOS.

In Japan, it was difficult to sell 16-bit computers at all. The Japanese had a huge installed base of 8-bit computers. The most popular computers there continued to be Z80 machines. Mitsubishi was offering 16-bit machines, but they, too, initially licensed CP/M-86. Microsoft never wrote off a customer, however. Their strategy was to help companies do what they wanted to do and eventually convince them that what they wanted to do was offer MS-DOS as their "operating system of obvious preference." In the case of Mitsubishi, Microsoft helped the company get Multiplan and FORTRAN running on the CP/M-86 system. Out of this relationship of cooperation, they gradually won Mitsubishi's support of MS-DOS.

MAY 1982 DEC introduces four 16-bit computers running CP/M.

Microsoft did not ignore the popularity of the 8-bit machines in Japan either. After MS-DOS was released, they asked Paterson to write an 8-bit operating system that used the same file format as MS-DOS. Microsoft released MSX-DOS in September 1983 and eventually sold millions of copies in Japan.

Making MS-DOS the "operating system of obvious preference" was not as easy as getting hardware manufacturers to offer it. Microsoft's list of MS-DOS customers grew steadily from the time it was introduced, but many of them offered CP/M-86 along with MS-DOS. Digital Research also convinced several manufacturers to include both the 8080 and 8086 chips in their machines. CP/M compatibility took on a new meaning when 8-bit and 16-bit software was used on the same machine, since the user would probably want to use the same disk format for both types of software. MS-DOS used a different disk format, and was thus at a disadvantage in these dual-processor machines.

Ultimately, Microsoft sold MS-DOS on its technical superiority. When

DEC decided, somewhat late in the game, to make MS-DOS the primary operating system for its Rainbow computer, they pointed to the richer set of commands and "dramatically" better disk performance as reasons. The improvement in disk performance was the result of the unique file system in MS-DOS, which did not need to access the disk nearly as often as CP/M. Of course, by the time DEC decided to offer MS-DOS, most software developers had already made up their minds.

"IBM absolutely was not enough to create a standard for MS-DOS."
—Bill Gates

JUN 1982
DOS 1.1 runs on double-sided disks on the IBM PC. DOS 1.25 supports double-sided for non-IBM customers.

While Microsoft was gaining its reputation for technical superiority, it was already working to improve MS-DOS technically. In 1982, IBM released an upgraded PC that used double-sided disks. Microsoft provided Version 1.1 to IBM to handle the new disks. Version 1.25 did the same for the generic MS-DOS.

"As [Rainbow's] MS-DOS had been out only for a month . . . we expected to find 10, 20, or maybe even 30 packages. We were astounded to find 250 programs from more than 60 different manufacturers in 28 different application areas. That's more software than is available under CP/M-86 for the Rainbow, and CP/M-86/80 has been available for more than a year!"
—*Digital Review,* January 1984

Even before these intermediate releases, though, visions were beginning to dance in the minds of a few people at Microsoft. The DOS team was still small, perhaps the smallest at Microsoft. But they were talking to a much larger team, the team that had been developing XENIX. For the first version of MS-DOS, there had been no time to think about borrowing features from XENIX. There had also been no room. IBM wanted a 16K system, and at that time, XENIX was closer to 100K. But when IBM told Microsoft that the next major release of their personal computer was going to be the XT with a 10-megabyte (10MB) fixed disk, the possibility of a larger, more powerful version of MS-DOS led the DOS team to take a closer look at XENIX.

IBM might well have been satisfied if this new version had simply added a disk device driver for the hard disk. But Microsoft convinced them that a hard disk really called for the kind of tree-structured hierarchical file system that UNIX used. They also convinced IBM that they could adapt the file system in Version 1.0 to support hierarchical directories.

At the same time, Microsoft was becoming aware that many of the emerging third-party peripheral devices weren't working well with one another. Each company had its own way of hooking its hardware into DOS, but if two third-party devices were plugged into the machine at the same time, they would often fail. What was needed in the next version of MS-DOS was a better mechanism for encouraging people to write device drivers and to use a consistent method for installing these device drivers.

JUN 1982
Microsoft releases SoftCard III for Apple III.
Microsoft releases BASIC compiler for MS-DOS.

Finally, the word "task" was becoming important. Multitasking had always been a strength of UNIX, which was used in large, multiuser environments where many people need to be able to use the resources of the computer to perform different tasks at the same time. MS-DOS was a small, single-user environment. But the DOS programmers recognized that, in this environment, too, the concept of tasks could be useful, and they began to think about the situations in which users might want to perform certain tasks in the background.

> *"At that time, MS-DOS was about the smallest cookie on Microsoft's plate."*
> —DOS programmer

So while the rest of the world was debating the pros and cons of MS-DOS versus CP/M-86 as an industry standard, the DOS programmers were redefining MS-DOS to make it a more powerful and more flexible operating system.

New Power in a New Release

MAR 1983
IBM announces the IBM PC-XT. Microsoft releases MS-DOS Version 2.0.

Version 2.0 of MS-DOS was released in March 1983. In the development of this new release, the major design issue had been the file system. Version 1.0 had a single directory for all of the files on a floppy disk. With a 10MB fixed disk, a single directory would quickly become unmanageably large, since the disk might hold as many as 1000 files.

The designers of CP/M had approached the problem of large media by using a partitioning scheme. This system divided the disk into 10 user areas of fixed size, each with its own directory. In effect, it made the fixed disk look like 10 floppy disk drives. XENIX, which had traditionally dealt with larger systems, used a hierarchical file structure, in which the user could set up as many levels of directories, subdirectories, and files as needed to organize files and make them readily accessible.

In development meetings, IBM and Microsoft had debated the merits of these two approaches. The arguments in favor of partitioning were familiarity, size, and ease of implementation. Many small system users—especially software developers—were already familiar with partitioning from their experience with CP/M. The code necessary to develop a partitioning scheme would be minimal compared with the code to manage a hierarchical file system and would also take less time to implement.

Partitioning had two inherent disadvantages, however. First, partitioning becomes a less viable solution as the storage media grow. It may be an adequate system for handling 10 megabytes of storage, but it ceases to provide adequate access to files when storage sizes grow to the range of 100 megabytes per disk, as is the case with large hard disks and optical storage disks. In 1983, Microsoft was already anticipating this kind of growth in storage capabity of media.

The second disadvantage in partitioning was its dependence on the physical device. If the size of the disk changes, either the number or size of the partitions must change. These changes would then necessitate changes in both operating-system code and the code for application programs. For Microsoft, with its commitment to device independence, partitioning would have been a step backward.

A hierarchical file structure could be implemented to be device-independent. A disk could be partitioned logically rather than physically to support the needs of its users. For example, if two or more users shared a fixed disk, they could divide the disk into one directory for each user rather than being limited by an arbitrary number of partitions. Then, within each directory, they could set up their own individual subdirectories for the different projects or types of data and programs that they needed for their work. In short, a hierarchical system let the user determine the best way of organizing a disk.

Microsoft had another reason for preferring a hierarchical file system. That reason was compatibility with its XENIX operating system. Compatibility seemed important for several reasons. When the choice of file systems was being made, Microsoft was working to make XENIX a viable operating system for smaller computers. It seemed possible that as XENIX became available on smaller computers, users of systems like the IBM PC would want to move from DOS to XENIX. The growing interest in networking also suggested that XENIX compatibility would be important, since users of both operating systems would be likely to want to share data and resources. If the systems shared the same file structure and file-naming conventions, it would be possible for software developers to write source code that could be compiled and run in either environment, thereby saving them the expense of developing two different sets of source code using two different approaches for the two operating environments.

MAR 1983 DOS 2.0 introduces a hierarchical file system.

In the end, IBM was convinced that a hierarchical file system was the better solution to the problem of fixed-disk support, even if it was more complex. So the DOS programming team set about the task of developing a file system that was physically consistent with the method of disk access developed in Version 1.0—namely, the file allocation table. At the same time, it had to be logically consistent with the XENIX file structure. The result was a system of directories, subdirectories, and files in which each file was identified by a unique pathname. There were no limits on the number of levels of subdirectories other than the limit on the length of the pathname, which could not exceed 64 characters.

Microsoft made one mistake in this process. At IBM's request, it used the backslash to separate file names in the pathname. XENIX had used a forward slash for this purpose. But Version 1.0 of MS-DOS had already allocated the forward slash for switches in the command line, borrowing from the tradition of DEC operating systems. So Microsoft decided to use the backslash for its pathname separator. This decision created several potential problems. First, it was not consistent with any UNIX-like operating systems. Also, MS-DOS ran on many machines, such as foreign machines, that did not have a backslash on their keyboards.

To address this problem, Microsoft invented a SWITCHAR configuration command that allowed the user to change the switch character from a slash to a hyphen. This solution, however, created yet another problem. If users wanted to exchange batch files, they would run into compatibility problems if they had changed the switch character. In the end, Microsoft decided not to document the SWITCHAR command. Unfortunately, users discovered the feature anyway, and Microsoft has had to spend a lot of time "weaning" users from SWITCHAR.

MAR 1983 DOS 2.0 uses file handles to call files.

In order to allow applications to fully exploit the new directory structure, Microsoft made another major change in the file-management system in Version 2.0. It was necessary to add a new way of calling fileservices. Version 1.0 of MS-DOS used file control blocks (FCBs) for compatibility with the old CP/M-80 programs. The file control blocks contained all of the pertinent information about the size and location of a file, except it didn't allow for specifying a file in a different directory. When a program needed to access a file, the programmer used function calls that directly manipulated the information in the FCB.

In version 2.0, MS-DOS added the ability to call files using file handles. File handles were another step toward logical independence. Whenever an MS-DOS program opened a file, MS-DOS returned a handle to the program. All further interaction with the file involved only this handle. MS-DOS itself made all the adjustments to the file control block so that the user never had to deal directly with the information about the file's location in memory.

Aside from simplifying the task of the programmer, file handles made it easier for programs to be ported from one system to another. And if future versions of MS-DOS changed the size or contents of file control blocks, the programmer would not need to rewrite any code, since the handle would be the only referent needed. The handle would not change.

MAR 1983 DOS 2.0 has ability to redirect file input and output.

Making the FCBs internal to the operating system and substituting file handles made it possible for MS-DOS to redirect file input and output. Once a file handle was assigned, a program could redirect any input or output to a new file by giving MS-DOS both handles. MS-DOS automatically used the new file whenever the original file was accessed. This capability was used in the COMMAND.COM file to allow the user to redirect output from a file to a printer, for example, or to "pipe" file data to a sort filter.

MAR 1983 DOS 2.0 introduces installable device drivers.

A major innovation in Version 2.0 was installable device drivers. One of the hallmarks of IBM's approach to the PC was open architecture, which meant that users could just slide new cards into the computer when they wanted to add new input/output devices such as a hard disk or a printer.

Unfortunately, Version 1.0 of MS-DOS did not have the corresponding open operating-system architecture. The Basic Input/Output System (BIOS) contained all the code that permitted the operating system to run the hardware. If independent hardware manufacturers wanted to sell their equipment for use with a computer manufacturer's DOS, they would either have to completely rewrite the entire set of device drivers, incorporating code for their device, or they would have to write a complicated utility to read the existing drivers, alter them, add their code, and produce a working set of drivers. These "patches" would often conflict if the user installed more than one device. Furthermore, each time MS-DOS was updated by the computer manufacturer, the suppliers of the peripheral devices would have to revise all of this code.

By the time Microsoft began working on Version 2.0, the DOS team recognized that the ability to install any device driver at run time was crucial. They implemented installable device drivers by making the device drivers more modular. IBMBIO.COM became, in effect, a linked list of device drivers. The linked list was expandable through the CONFIG.SYS file. So the manufacturers could simply write a device driver, and the user could install it at run time by including it in the CONFIG.SYS file. DOS then added it to the linked list.

MAR 1983 DOS 2.0 provides serial stream processing.

An extension of this capability to install device drivers was the ability to install or remove serial stream processing routines—for example, ANSI.SYS—to support the ANSI standard escape codes for cursor positioning and screen control.

Version 2.0 of MS-DOS also made it possible to do limited background processing. Background processing was the MS-DOS solution to the growing market demand for multitasking. To determine why the public wanted multitasking or whether they really wanted it at all, Microsoft studied the way that most business users of MS-DOS did their work. Did they, for example, really need to run WordStar and Lotus at the same time?

MAR 1983 DOS 2.0 introduces background processing.

The DOS team concluded that for many people and for most situations, background print spooling would be sufficient. To implement this kind of background processing, Microsoft added some internal calls. Using these calls, MS-DOS would automatically generate an Interrupt 28 to run the print spooler, PRINT.COM, whenever it had nothing to do. When the parent application became active again, PRINT.COM would be interrupted until the next Interrupt 28. This kind of background processing also opened the possibility of background communications such as electronic mail.

Hierarchical files, installable device drivers, and background processing were the major design decisions in Version 2.0. But for every major decision, there were dozens of smaller problems to be solved. And often, solving them required "clever hacks." For example, with the new fixed disk, it was necessary to modify the code for automatic logging of disks. The modification meant that the operating system had to access the disk more often, and file access became a lot slower as a result. One of the DOS programmers came up with a clever solution to this problem. He reasoned that if DOS had just checked the disk, there was some minimum time it would take for a user to change disks. If that time had not elapsed, the current disk information in RAM was probably still good. He took a stack of disks and tried cramming them into the disk drive as fast as he could, without caring if he bent the disks. The fastest he could change disks was about four seconds. Based on this empirical study, he had DOS check to see how much time had gone by since the last disk access. If less than two seconds had elapsed, DOS could assume that a new disk had not been inserted and it simply used the disk information in RAM. With this little trick, the speed of file handling increased considerably.

JAN 1983 Lotus releases Lotus 1-2-3.

When Version 2.0 of MS-DOS was ready for release, it had grown to 20,000 lines of code. The MS-DOS team had remained small, however, with only three or four programmers playing key roles at a time when Microsoft's total staff had grown to about 300. Tucked away in a couple of offices away from most of Microsoft's development work, the DOS team was almost invisible to the rest of the company. MS-DOS still did not represent a large share of Microsoft's earnings.

SPRING 1983 Microsoft introduces Microsoft Word.

But all of that was about to change. Within six months of its release, Version 2.0 had gained widespread acceptance, so wide, in fact, that when Digital Research released its version of CP/M-86 to support the hard disk six months after the XT was released, it was already too late for it to compete seriously with MS-DOS. Popular application programs like Lotus 1-2-3 took advantage of the features of the new version of MS-DOS, and thus helped to secure its future as the industry standard for 8086 processors. The challenge to MS-DOS was no longer the challenge to become an industry standard. It was the challenge to manage growth.

The Challenge of Growth

OCT
1983
Microsoft Press is formed.

The world of Version 2.0 was a world very much changed from that in which Version 1.0 of MS-DOS made its debut. When IBM had first released the PC almost a year and a half earlier, the market for business computers was uncertain and undefined. Now, IBM was releasing the XT into a known market, one that they themselves had built with the PC. Anyone who wanted to introduce a new computer knew that IBM was the competition to beat, and most manufacturers chose to compete with the IBM PC by emulating it. Software developers also had a new understanding of business computing. They felt confident that they could target their software for the market that IBM had defined. MS-DOS looked less risky in this new environment, and concerns about the existing base of CP/M-80 software faded as developers turned their sights away from home computing and toward the racing business market.

DEC
1983
DEC offers MS-DOS for its Rainbow computer.

In this environment, MS-DOS quickly secured its position as an industry standard, and the problem of promoting MS-DOS diminished. Microsoft now faced a new problem: the maintenance of an industry standard. Being a standard, MS-DOS had to be many things to many people. IBM had demands. The OEMs had demands. And sometimes the demands conflicted.

MAR
1984
IBM announces PCjr.
Microsoft releases DOS 2.1 to support PCjr.

When Version 2.0 was released, IBM was already planning to introduce its PCjr. The PCjr was designed with the ability to run programs out of ROM cartridges. IBM also used a slightly different disk-controller architecture for the PCjr as well as half-height 5¼-inch drives rather than full-height drives. To add to the variation, the drives were supplied by a different manufacturer, and they were not up to the hardware specs that MS-DOS assumed for the full-height drives. Because these drives were slower than the PC's and XT's and because MS-DOS couldn't tell which drives were being used, MS-DOS 2.1 had to be modified to handle the slower drives.

For the longer term, IBM was planning a faster, more powerful machine with a 20MB fixed disk. This prospect led Microsoft to take a new look at its file-management system since the larger storage capacity of the 20MB disk went beyond the size limitations for the file allocation table as it worked in Version 2.0.

IBM's real interest for the next major release of MS-DOS, however, was networking. Microsoft would have preferred to pursue some form of multitasking as the next stage of development for their operating system, but IBM was developing its IBM PC Network Adapter, a plug-in card with an 80188 chip to handle communications. So as soon as Version 2.0 was released, the MS-DOS team began work on a networking version of the operating system.

> *"We wanted to do multitasking, and IBM wanted to do networking. And we decided to do networking. Of course, it's not as simple as that. I mean, you can't just fill in the blanks. You can't just say, 'We wanted to do X and they wanted to do Y, and we decided to do Y.'"*
>
> —Bill Gates

At the same time, Microsoft's other customers were becoming more vocal about their own needs, which did not always match those of IBM. As it turned out, several of them did want a networking capability, too, which added weight to IBM's request. But a growing need among many of them was support for international products. They needed a version of DOS that could be sold in any country. That meant that it would have to be possible to translate all of the system messages and that programs would need a way to determine the local format for the time and date. They would also need to be able to choose the appropriate decimal sign and currency symbol and identify whether the currency symbol appears before or after the currency amount for a given country.

MAR 1984 Microsoft releases DOS 2.05 with international support.

At that time, IBM was not as interested in the international market. But Microsoft, as well as its other customers, wanted to make MS-DOS an international product. So while the DOS team was modifying the operating system to support the PCjr, it was also adding functions and a COUNTRY command that allowed users to set the date and time formats and other country-dependent variables in a CONFIG.SYS file.

Another international requirement also appeared at about the same time. The Japanese market for Microsoft was growing, and the question of supporting Kanji characters arose. The difficulty with Kanji characters is that they require a so-called dual-byte standard. For English and most European character sets, one byte corresponds to one character on the screen or printed page. In Japan, a Kanji character sometimes uses one byte per character and sometimes two. This variability creates problems in parsing when MS-DOS wants to back up: That is, with Kanji characters, MS-DOS doesn't know how far to back up. As a result, it has to go back and parse the string from the beginning in order to figure out where to stop. MS-DOS had to be modified to back up and parse in this way.

The support for country-dependent formats and Kanji characters appeared for the first time in Version 2.05. IBM, however, refused to use this version. So the modifications to support the PCjr appeared as a separate version, numbered 2.1, which went only to IBM.

MAR 1984 Microsoft releases DOS 2.11.

One of the problems that Microsoft faced as early as Version 1.25 was that most of its OEM customers wanted to get the same version that IBM got. Some manufacturers, such as Compaq, were selling 100 percent compatibility with IBM. For them, any difference in the operating system introduced the possibility of incompatibilities. Providing identical code to non-IBM customers was a difficult demand to meet in general. It wasn't until Version 3.1 that Microsoft was able to supply a system that its non-IBM users agreed was identical with the one that IBM got. They were certainly not happy when Microsoft shipped Version 2.05 to its OEM customers and a different version—2.1—to IBM. To satisfy the OEM customers, Microsoft had to combine 2.1 and 2.05 to create Version 2.11. Although IBM did not accept Version 2.11 because of the internationalization code, it became the standard MS-DOS version run by all non-IBM customers running anything in the Version 2.X series. It was sold worldwide and translated into about 60 different languages. Two other intermediate versions—Versions 2.2 and 2.25—provided Hangeul (the Korean character set) and Hangeul plus Kanji support, respectively.

JUN 1984 Microsoft becomes the first software company to reach $100 million annual sales.

After the release of 2.0, Microsoft also learned to appreciate the importance—and difficulty—of supporting people who are developing software on the world's leading operating system. Software developers worried about downward compatibility. They also worried about upward compatibility. But they didn't always adhere to programming practices that guaranteed that compatibility. If their programs were a success, it was Microsoft that was left with the burden of ensuring compatibility.

> *"To people who work on DOS, compatibility is a religion."*
>
> —Bill Gates

One of the problems was IBM's open architecture. Since the information about the internals of the BIOS and the ROM interface was published, software developers could work directly with the hardware, and often did in order to get more speed. In other words, developers sidestepped the operating system for some operations. Unfortunately, by choosing to work at the lower levels, they did not take advantage of the protection that the operating system provides against hardware changes. Thus, when low-level changes were made in the operating system, their programs didn't work. Or they couldn't run cooperatively with other applications.

Another problem was the continuing impact of the historical need for compatibility with CP/M. For example, in CP/M-80, programmers would call address 5 in order to request a function. In MS-DOS, Interrupt 21 was the function call. But to support old programs, the first version of MS-DOS also allowed a program to request functions by calling address 5. Although this feature was not documented, a number of existing programs continued to use it when they were converted to MS-DOS. One of these programs was WordStar. Microsoft could not afford to make changes in the operating system that would make it impossible to run a program as popular as WordStar. So each new version had to continue supporting CALL 5, even though it was never documented.

A more pervasive problem was the use of FCB-style calls. The Version 1 series had used FCB-style calls exclusively, as had CP/M. Version 2.0 introduced the more efficient and flexible XENIX-style calls. But Microsoft could not simply abolish the old FCB-style calls because many popular programs used them. In fact, some of Microsoft's own languages used them. So MS-DOS had to support both types of calls in the Version 2 series.

Microsoft's goal was eventually to eliminate the use of FCB calls. To encourage the use of the new XENIX-style calls, the company made it easy for everyone to upgrade to 2.0. They convinced IBM to require Version 2.0 for the XT, so that anyone who wanted fixed-disk support would have to use 2.0. They also encouraged software developers to require 2.0 for their applications. Both the software developers and the OEM customers were reluctant to require 2.0 because they were concerned about the installed user base of 1.0 systems, which meant that they had to support both sets of calls. They had to be able to detect which operating system the user was running. If they were running a Version 1 system, they would use FCB calls. If they were running any of the Version 2 systems, they would use the file handles. It was an awkward period of transition, but by the time Microsoft started working on Version 3.0 and the support for the 20MB fixed disk, it became apparent that the change had been in everyone's best interest.

The kinds of issues that began to emerge as Microsoft worked toward Version 3.0 exaggerated the problems of compatibility. Networking, with or without a multitasking capability, requires a level of cooperation and compatibility among programs that was never an issue in the earlier versions of MS-DOS. The great variability in programs and programming approaches that MS-DOS supported eventually proved to be one of the biggest hurdles to the development of a sophisticated networking system and, in the longer term, to the implementation a multitasking system.

JUN 1984 **200 computer manufacturers have licensed MS-DOS.**

By the time Microsoft began work on Version 3.0, the programming style in the DOS team had changed considerably. It was still a small team with a core group of just two or three people, but by now, the concerns for maintainability that had dominated programming philosophy in larger systems had percolated down to the MS-DOS world, and the desire to optimize for speed, using hacker's tricks, had to be tempered by the need for clarity and maintainability. In the same way, the small package of tightly written code that first characterized MS-DOS also had to be sacrificed for clarity and for the sake of long-term maintenance.

"You better believe that when we've got a new version, we go see Lotus and Ashton-Tate and people like that to make sure they feel good about the stuff that's going to be in there."

—Bill Gates

The working relationship with IBM also became more formal as MS-DOS proceeded toward Version 3.0. In the early days of Version 1.0 and even during the development of 2.0, the specifications from IBM were just a list of high-level requirements. Relying on Microsoft to define the lower level requirements, IBM did not provide formal specifications. By the time they began working on 3.0, however, the DOS team would receive formal specifications, which they could review and modify and then return to IBM for their review and modification until, after several exchanges, the specifications would be approved and Microsoft could begin work.

A Version for Networks

AUG 1984
IBM introduces the IBM AT. Microsoft releases MS-DOS Version 3.0.

The work on a version of MS-DOS for networks proved to be long and difficult. For a year and a half, Microsoft grappled with problems of software incompatibility, remote file management, and logical independence at the network level. When IBM was ready to release its new larger Personal Computer AT, the network software for MS-DOS wasn't quite ready. So in August 1984, Microsoft released Version 3.0 to IBM without network software.

AUG 1984
DOS 3.0 supports the AT's 20MB fixed disk.

Version 3.0 supported the AT's larger fixed disk, its new CMOS clock, and higher density floppy disks with door locks. It also included the international support that IBM had refused in 2.05 and 2.11. These same features were made available to Microsoft's OEM customers as Version 3.05.

Version 3.0 was not, however, a simple extension of Version 2.0. In laying the groundwork for networking, the MS-DOS team had completely redesigned and rewritten the MS-DOS kernel. Version 2.0 had been built on top of the structure of Version 1.0. In DOS 1.0, file requests used the FCBs. When a program made an FCB request, the request was passed to a big piece of FCB input/output code. Version 2.0 introduced file handles to specify files, so that a program simply requested a handle when it wanted to use a file. However, the handle calls would parse the pathname and then use the underlying FCB calls in the same way as Version 1.0. The redirected input and output in Version 2.0 further complicated the file system requests. In order to provide this capability to programs using the original CP/M system calls 1 through 12, Version 2.0 had created handle calls for these basic input/output calls. When a program used one of these CP/M calls, MS-DOS would first give it a handle and then turn the call back into an FCB call at a lower level.

AUG 1984
DOS 3.0 eliminates redundancy in file system requests.

Version 3.0 eliminated this redundancy by eliminating the old FCB input/output code of Versions 1.0 and 2.0. In its place was a standard set of I/O calls that could be called directly by both FCB calls and the XENIX-style handle calls. The look-alike calls for CP/M calls 1 through 12 were included as part of the set of handle calls. As a result of this restructuring, these calls were distinctly faster in Version 3.0 than in Version 2.0.

Even more important than the elimination of inefficiencies, this new structure made it easier to handle network requests. Microsoft was using the ISO Open System Interconnect as its model for networking. This model described a number of protocol layers. On the IBM PC network, the transport layer as well as the server functions were handled by IBM's PC Network

Adapter card. The task of MS-DOS was to support this hardware. For its OEM customers, Microsoft had to supply the transport and server functions as software. While Version 3.0 did not provide this general-purpose networking software, it did provide the basic support for IBM's networking hardware.

AUG 1984 DOS 3.0 provides redirector and sharer software.

This support consisted of redirector and sharer software. Microsoft adopted an approach to networking in which remote requests were routed by a redirector. The redirector would know how to interact with the transport layer of the network. The transport layer was the device drivers that could reliably transfer data from one part of the network to another. It was just before a call was sent to the newly designed low-level I/O code that a determination was made to see if the call was local or remote. If it was local, it would fall through to the I/O code. If it was remote, it would be passed to the redirector. The redirector, working with the operating system, would make the resources on a remote machine appear as if they were local. That was the primary function of the network.

NOV 1984 Microsoft releases MS-DOS Version 3.1 and Microsoft Networks.

Both the redirector and sharer software were in place in Version 3.0 when it was delivered to IBM. Version 3.1, released three months later, refined this network support. It was also available for use on non-IBM network cards in the form of Microsoft Networks.

Microsoft Networks was built on the concept of "consumers" and "services." Services were provided by a file server, which was part of the Networks application and ran on a dedicated computer. Consumers were programs on various network machines. They made requests for information to the server, which were then passed at a very high level to the server. There, it was the responsibility of the file server in Microsoft Networks to figure out where to find things on the disk. The requesting programs—the consumers—would not need to know anything about the remote machine, including what kind of file system it had.

NOV 1984 Microsoft Networks provides transport and server software for non-IBM users.

Microsoft Networks was designed to be hardware independent. However, the variability of the classes of programs that would be using its structures was a major hurdle to developing a networking system that was transparent to the user. To handle this variability, Microsoft identified three classes of programs. The first were the MS-DOS compatible programs. These were the programs that used only the documented Interrupt 21 call when requesting functions. These programs would run on any MS-DOS machine without problem.

The second class of programs were the MS-DOS based programs. They would run on IBM-compatible computers, but not necessarily on all MS-DOS machines.

The third class were what Microsoft labeled "incredibly misbehaved IBM programs." They used undocumented features of MS-DOS or they addressed the hardware directly. While these programs tended to have the best performance, they were the most difficult to support.

NOV 1984 DOS 3.1 supports networking with "well-behaved" software.

In the end, Microsoft decided to support officially only the MS-DOS compatible programs on the network. These programs all used the documented function requests. Their requests could thus be made to local or remote machines without any intervention by the user. In short, they would be invisible.

NOV 1984 DOS 3.1 controls the number of open FCBs.

While the change in file structure in Version 3.0 simplified file management on the network, it did not solve all of the problems. MS-DOS still had to handle FCB requests from programs that used them. Many programs would open an FCB and never close it. One of the functions of the server was to keep track of all of the open files on the network, and it ran into difficulties when an FCB was opened 50 or 100 times and never closed. To solve this problem, the designers of MS-DOS introduced a cache in Version 3.1. The cache allowed four FCBs to be open at any one time. If a fifth were opened, the least recently used FCB would be closed automatically and released. An FCBS command was added to allow the user to change the default number of FCBs that could be open at any one time.

NOV 1984 DOS 3.1 provides network printing.

The logical independence that had become a goal of MS-DOS in general acquired new meaning—and created new problems—with networking. One of the problems concerned printers on the network. A common use of a network was to allow a user who may not have a printer on his machine to use a printer in someone else's office down the hall. Some programs would open the printer, write to it, and close it. Such programs were easy to accommodate on the network. The "incredibly misbehaved" programs would try to use the direct IBM BIOS interface to access the printer. In order to handle this situation, the DOS designers had to develop a way for MS-DOS to intercept these BIOS requests and filter out the ones that the server would not be able to handle. With this technique, Version 3.1 was able to handle most types of printer output on the network.

NOV 1984 DOS 3.1 introduces installable file systems.

The ability to pass a high-level request to a remote server without having to know the details of the remote server's file structure allowed yet another level of generalization of the system. In DOS 3.1, it became possible to access different file systems. For example, it would be possible to access a

XENIX file system and read data from XENIX machines. It would also be possible to have several different file systems on the network concurrently and still access them as though they were local. Version 3.1 does not completely support this abstraction of logical file systems, but provides the basic structure for it in the future.

JUN 1985
Microsoft joins the Intel/Lotus/Microsoft Extended Memory Specification.

Microsoft released one more intermediate version of MS-DOS: Version 3.2. This version supported 3½-inch floppy disks. It also moved the formatting function of a device out of the format utility routine and into the device driver. With this change, future devices will not have to supply their own format utilities. They will just provide the device driver, and the DOS formatting utility will call the driver to perform formatting.

AUG 1985
IBM enters joint-development agreement with Microsoft.

With the completion of Version 3.0, the complexion of the MS-DOS project at Microsoft changed. IBM recognized the significance of the IBM PC as more than a personal computer: It had become the automated office workstation and the relationship with Microsoft had been key to its development. So in August 1985, IBM entered into a joint-development agreement with Microsoft.

JAN 1986
Microsoft releases MS-DOS Version 3.25.

At the same time, the task of developing and maintaining MS-DOS mushroomed. Version 3.0 had about 40,000 lines of code—code that was necessarily much more complex since it had to support the complex environment of the network. To support this complexity, the MS-DOS team had to grow. Through the release of DOS 3.0, the team had been small, just four or five people at any given time. After the release of Version 3.0 and the joint agreement with IBM, the team grew quickly to 30 people. MS-DOS was no longer the "smallest cookie on Microsoft's plate."

> *"After DOS 3, it's a big deal. It's a BIG deal!"*
> —Bill Gates

A View Toward the Future

Throughout the development of MS-DOS, Microsoft kept a watchful eye on the marketplace. The expectations of IBM and other hardware manufacturers, of software developers, and of end users have shaped MS-DOS at each stage. And they will continue to do so in the future. The current interest in taking advantage of the protected mode of the 80286 chip, the continuing interest in expanded multitasking capability, the growing popularity of graphic interfaces, and the potential impact of the increased storage capacity of new media such as compact optical disks—all of these issues are part of the daily discussions among members of the MS-DOS development team.

For example, Microsoft's current approach to a graphics interface is embodied in Windows, a product that was released in November 1985 to run on top of MS-DOS. The use of Windows is optional. Microsoft recognizes that some people want it, and some don't. So Windows has been developed as a freestanding program that MS-DOS users can choose to use or not. This approach has an additional advantage: Microsoft can change features of MS-DOS to improve it without necessarily having to modify the Windows interface. Should 90 percent of MS-DOS users eventually begin using Windows, Microsoft may incorporate it into the operating system. But for now, it provides a flexibility that Microsoft believes is important.

The appearance of new media like compact disks raises the issue of data independence. Ultimately, the user should be able to access data without knowing where it is—that is, without knowing its physical location on the disk, which disk it is on, or even which machine on the network it is on. It will simply be the task of the operating system to find it. To provide this kind of location independence of data, MS-DOS will eventually need to support more than one file system at a time and will have to develop file systems that support very large media.

While issues such as these will shape the future development of MS-DOS, they do not define a general development philosophy. To understand the philosophy that will guide every innovation in MS-DOS, it is necessary to go back to the initial problem that IBM faced when it came to Microsoft in August 1980. That problem was the need to produce a state-of-the-art system in an industry of rapid technological change. The same problem continues to exist for both hardware manufacturers and software developers today. The hardware is evolving and getting better. The software

Major Releases – MS-DOS

August 1981	1.0	•First operating system on IBM PC
May 1982	1.1	•Double-sided disk support
March 1983	2.0	•Support for hierarchical files and hard disks
October 1983	2.1	•PC-DOS introduced with PCjr™
August 1984	3.0	•Support for 1.2Mb floppy disk on hard disk
March 1985	3.1	•Support for Microsoft Networks
December 1985	3.2	•IBM PC Convertible introduced •Support for 3.5-inch drives
April 1987	3.3	•Support for multiple partitions on hard disks •Improved support of foreign characters
November 1988	4.01	•Support for hard disk files over 32Mb •Ability to move portions of DOS into expanded memory •User shell
June 1991	5.0	

From: kgill@or.wagged.com
To: simsong@MIT.EDU
Subject: Microsoft timeline
Date: Thu, 17 Nov 94 10:39:00 PST

Please give me a call if you need any additional information!!

Thanks!
Kathy

Microsoft Timeline

1985
Nov. 20 Microsoft ships the retail version of Windows

1986
March 13 Microsoft stock goes public at $21 per share rising to $28 by the end of the first day of trading.
Nov. 17 Microsoft releases MS-DOS 4.0

1987
Jan. 01 Installed base for MS-DOS is over 21 million

Oct. 06 Microsoft releases Excel for Windows

1988
Jan 01 Microsoft becomes largest PC software company based on sales

1989
Jan. 01 Installed based for MS-DOS is over 39 million

1990
May 22 Microsoft announces the immediate, worldwide availability of Windows 3.0

1991
Jan 09 Microsoft announces Excel version for 3.0 for Windows, OS/2 and Macintosh platforms simultaneously.
June 11 Microsoft announces the immediate availability of MS-DOS 5.0
Nov. 14 Microsoft announces the Multimedia Edition on Works for Windows 2.0, Microsoft s first business application to incorporate multimedia.

1992
April 06 Microsoft ships Windows 3.1 with over 1,000 enhancements. The new version creates unprecedented users demand with over one million advanced orders placed worldwide.
May 20 Microsoft announces 3 million copies of Windows 3.1 shipped in the first six weeks after the product s release
Oct. 1 Windows for Workgroups is released
Oct. 27 Windows for Workgroups 3.1 ships worldwide.

1993
March 22 Encarta, the first multimedia encyclopedia designed on a computer, for use on a computer, ships
March 31 MS-DOS 6.0 ships
Nov. 01 Microsoft introduces MS-DOS 6.2
Nov. 01 Windows for Workgroups 3.11 ships

1994
Jan. 19 Shipments of Windows exceed 40 million units
April 4 Microsoft Windows 3.11 ships
April 18 Windows for Workgroups 3.11 becomes the world s best selling retail operating system, edging Windows 3.1 into the No. 2 spot.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 206 882 8080
Telex 160520
Fax 206 883 8101

Microsoft News Release

For more information contact:

Microsoft Corporation
Liz Sidnam or Marty Taucher
(206) 882-8080

Waggener Edstrom
Carrine Greason or Marianne Allison
(503) 245-0905

For Release 4 p.m. EDT
June 11, 1991

Microsoft Unveils MS-DOS 5

NEW YORK CITY -- June 11, 1991 -- Microsoft today announced the immediate availability of Microsoft® MS-DOS® 5, which contains major enhancements that bring greater functionality to all DOS* users whether novice or advanced.

Improved memory management in version 5 makes more memory available for DOS applications and data files, including those running under the Microsoft Windows™ graphical environment. A new Shell, online help, a task swapper and an undelete utility are among the newly added features that make MS-DOS 5 easier to use than previous versions.

As in the past, Microsoft will distribute MS-DOS through PC manufacturers for use on new computers. Currently, more than 130 PC manufacturers worldwide have licensed MS-DOS 5 for their customers. These 130 manufacturers accounted for nearly 90 percent of the DOS-based PCs shipped last year, according to Microsoft estimates. Version 5 is available from many of these PC manufacturers immediately.

To meet the needs of the current installed base of approximately 60 million MS-DOS users worldwide, Microsoft also for the first time ever will make an upgrade version available through the retail channel. This upgrade-only product will not install

---

* As used herein, "DOS" refers to MS-DOS and PC-DOS operating systems.

Microsoft®

without DOS 2.11 or higher already present on the machine. An intelligent, automated installation procedure upgrades all brands of PCs, even over a network, making the process safe and simple. The Microsoft MS-DOS 5 Upgrade has a suggested retail price of $99.95. Microsoft expects more than 7,000 reseller store fronts and sales offices to offer the MS-DOS 5 Upgrade.

"The PC industry has flourished over the past 10 years because of MS-DOS and the support it has had from PC manufacturers, software developers and PC users worldwide," said Bill Gates, chairman and CEO of Microsoft. "MS-DOS 5 represents a large investment toward advancing this acknowledged industry standard. With the overwhelming commitment of PC manufacturers and the record numbers of orders placed for the MS-DOS 5 Upgrade, we believe this is the largest initial demand in history we've seen for a PC software product.

"We worked closely with large companies, small businesses, educators, vendors, and user groups to define and test this product. More than 7,000 beta testers worldwide helped make MS-DOS 5 and the MS-DOS 5 Upgrade the most tested software programs in PC history, resulting in increased product stability, reliability and compatibility."

Because of the extensive testing, MS-DOS 5 is highly compatible with applications written for previous versions of MS-DOS. In fact, almost all DOS applications will run without modification.

More Conventional Memory for DOS Applications Alone or Under Windows

MS-DOS 5 provides much more space for user programs and data by using memory-management technology to move systems code out of the user's normal memory space.

With 80286-based or higher PCs, MS-DOS 5 provides significantly more conventional memory for DOS applications because most of MS-DOS 5 can reside in

WAGGENER EDSTROM
INTERNATIONAL PUBLIC RELATIONS COUNSEL
6915 SW Macadam Avenue
Portland, OR 97219

FAX: 503/244-7261
Telephone: 503/245-0905

DATE: November 15, 1994
TIME:

URGENT!

TOTAL NUMBER OF PAGES
INCLUDING COVER SHEET: 13

JOB #:

PLEASE DELIVER IMMEDIATELY TO:

SIMPSON GARFUNKEL
FAX #: 617/876-5999
PHONE:

FROM:

CAROL FOR COLLEEN LACTER
Waggener Edstrom

MESSAGE:

For Release 1 p.m. PST
Nov. 14, 1994

Microsoft Unveils Its Strategy for a New Interactive Online Service

The Microsoft Network Is Designed to Expand the Market for Customers and Content Providers

LAS VEGAS — Nov. 14, 1994 — Microsoft Chairman and CEO Bill Gates today outlined the company's strategy for its new interactive online service called The Microsoft® Network. Gates detailed the key elements of the strategy, which include providing a compelling business model and platform for content providers, easy and inexpensive access for users, and availability of rich and powerful development tools. The Microsoft Network, scheduled to go into beta testing this month, is designed to provide easy, affordable access to the rapidly expanding world of electronic information and communication for users of the Windows™ 95 operating system.

"Microsoft has long believed in the promise of personal computers enabling new ways of thinking and communicating. We call this vision 'Information At Your Fingertips,'" said Gates. "The Microsoft Network online service will represent a significant step toward the realization of this vision."

The Microsoft Network's technology and business model is designed to help content and service providers fully realize the potential of the online market. The Microsoft Network uses a platform model in which content and service providers will have maximum flexibility in creating products and pricing their services. Microsoft anticipates that providers will offer various pricing

- more -

options, such as subscriptions, online transactions and ticketed events. Other services will be supported by advertising and commerce.

Content and service providers also will have control over the look of their services. To enable the easy creation of rich multimedia content and services, Microsoft plans to provide a complete tool set and sponsor developer and design conferences to educate providers on how to make best use of the online medium.

"We believe that the success of The Microsoft Network depends on our ability to deliver a comprehensive platform that enables successful online businesses for our providers," said Russ Siegelman, general manager of the online services group at Microsoft. "While 40 percent of users of Windows have modems, only 10 percent of them, and only 4 percent of U.S. households overall, subscribe to any online service. That's a huge opportunity for content providers."

Access to The Microsoft Network will be offered as a feature to users of Microsoft Windows 95. The Microsoft Network is designed to fully harness the power and ease of use of Windows 95.

When The Microsoft Network becomes available in 1995, it will offer interactive online communities built around ideas, people, products and brands. It is designed to bring customers affordable and easy-to-use access to electronic mail, bulletin boards and "chat rooms" on a variety of topics, file libraries, and Internet newsgroups. Members will be able to access online tips,

add-ons, tools, product information and technical support directly from the Microsoft area of the service.

The Microsoft Network will be accessible in more than 35 countries, and its client application will be localized in 20 languages. In conjunction with today's announcement, four of the world's leading telecommunications carriers announced they are providing the worldwide network infrastructure to enable access to The Microsoft Network. Members of The Microsoft Network will be able to access the service with a local phone call and connect at speeds of up to 14.4 kilobits per second. The data center for The Microsoft Network, located in the Seattle area, uses scalable technology based on PCs running the Microsoft Windows NT™ Server operating system.

Beta testing for The Microsoft Network will begin with the shipment of Windows 95 beta version M7, which is slated for mid-November.

"We're excited about this first beta phase," said Siegelman. "This is just the first step in what we plan to be a long-term investment for Microsoft's information-highway efforts."

Founded in 1975, Microsoft (NASDAQ "MSFT") is the worldwide leader in software for personal computers. The company offers a wide range of products and services for business and personal use, each designed with the mission of making it easier and more enjoyable for people to take advantage of the full power of personal computing every day.

Editor's Note: Windows NT is a trademarked product name. Please do not abbreviate in any way.

For More Information, Press Only:
June McLaren, Waggener Edstrom (206) 637-9097

The Microsoft® Network:
The Easy-to-Use Interactive Online Service For Users of Windows™ 95

Backgrounder

November 1994

For more information, contact:

Waggener Edstrom
June McLaren, (206) 637-9097

Introduction

Microsoft Corporation has long believed in the promise of personal computers to enable new ways of thinking and communicating that are accessible, useful, personal and fun for all computer users. It calls this vision "Information At Your Fingertips." The introduction of The Microsoft® Network online service represents Microsoft's next step toward the realization of this vision.

While interactive online services are well-publicized throughout the print and broadcast media, today's services are surprisingly less popular with consumers than all the hype might suggest. For example, although 40 percent of users of the Microsoft Windows™ operating system have modems, fewer than 10 percent of users of Windows and 4 percent of U.S. households subscribe to any online service. The online-services business today remains in its infancy, with providers of existing online services working to find the right technical, business-model and usability solutions that will promote acceptance beyond the early-adopter audience that has sustained this category to date. But the potential of these services is tremendous. The online consumer market could become a $2 billion market within five years, according to SIMBA Research.

For this projection to come true, significant investments must be made to deliver the promise of "Information At Your Fingertips" and to establish mainstream viability for interactive online technology. Online services must offer easier access and a more compelling environment for all computer users to attract and maintain a broader audience. Content and service providers must be offered greater publishing abilities and viable long-term profit opportunities before consumers will see compelling benefits.

Introducing The Microsoft Network

Access to The Microsoft Network is a feature of Windows 95, the forthcoming version of the Microsoft Windows operating system. It is an online service that makes accessing electronic information and communications easy and inexpensive for any user of Windows 95. It removes the primary barriers to online service use — cost, difficult user interface and inertia. The Microsoft Network extends the Windows-based desktop to a worldwide community of people, ideas and information. It provides a setting for a worldwide electronic marketplace of products and services from Microsoft and third-party companies.

With The Microsoft Network, Microsoft hopes to expand the online market by delivering the needed technology and business model to provide an online experience that meets the needs and expectations of both consumers and content providers.

The Microsoft Network Strategy

To succeed in the marketplace, interactive online services must provide uniquely rich and valuable solutions to customer needs, offered in a more compelling and accessible way than other alternatives. This premise drives the strategy behind The Microsoft Network. The Microsoft Network extends the benefits of online services to a larger audience by addressing the limitations that curb the widespread adoption of online services today.

A Viable, Long-Term Business Environment

The Microsoft Network differs from existing online services in many ways. Among the chief differences is the fact that it offers a new platform model that is flexible enough to encourage and reward independent content and service providers for their participation. This model facilitates the rapid development of a broad range of content and services, attracting users and expanding the market.

The online business environment must reflect the diverse nature of business itself in order to realize its full potential. Companies succeed in business by adopting practices, branding, packaging and selling models that are tailored to their products and their customers. They invest in channels of distribution that provide the infrastructure and flexibility to support these strategies and their resulting profitability. Current online services inhibit the way providers can present their information and services, and limit the profits that independent providers can realize, often to a small share of a shrinking revenue model based on customer online connection-time charges.

The success of The Microsoft Network will be based on the success of Microsoft's content providers in reaching and motivating customers. So Microsoft is going to lengths to help ensure that success. The Microsoft Network provides a new and different business environment that puts significant revenue control in the hands of the content or service providers.

Providers aren't limited in the ways in which they realize revenues for their services. Variable revenue and pricing models such as subscriptions, online transactions, advertising subsidies, and ticketed events are at the provider's discretion. More important, providers retain the majority of the revenues that their content and services generate.

Easy and Inexpensive Access

Meeting customer demands for easier access to technology is a fundamental charter of Microsoft products. To that end, customers will find it easy to sign up and access The Microsoft Network as a feature of the Windows 95 operating system. The Microsoft Network fully harnesses the power and ease of use of Windows 95.

By extending the feature set and graphical interface of Windows 95, The Microsoft Network provides a familiar environment that facilitates easier exploration and interaction in the online world. In all respects, The Microsoft Network looks and acts just like Windows 95, offering customers easy, consistent and graphical functionality.
For example, The Microsoft Network services can be browsed using the Explorer in Windows 95 or from an icon-based container view. Actions such as downloading files are simple copy operations accomplished by drag and drop. Shortcuts enable personalized and efficient navigation. The Microsoft Network's e-mail and rich-text content documents are managed through the Information Exchange and WordPad services built

into Windows 95 with the same familiar user interface carried through all core communications functionality.

The Microsoft Network also offers affordable access. By adopting a business model that emphasizes member and content activity rather than connect time, The Microsoft Network delivers services at the lowest possible cost to its members.

A More Compelling Online Experience

The Microsoft Network supports and promotes personalized discovery and investment for members and content providers alike. Both are enabled with a new generation of technology and capabilities and are empowered to shape and evolve online communities.

The Microsoft Network interactive experience revolves around these dynamic content communities, each produced to make the most of the interactive medium and the specific topic, product or experience at hand. For example, content areas cover the following subjects:

- Arts and entertainment
- News and weather
- Business and finance
- Sports, health and fitness
- Science and technology
- Computers and software
- Community and public affairs
- Home and family

These communities will be facilitated by a select group of Forum Managers, who bring specialized knowledge, credibility and respect in their areas of expertise.

Members will have tremendous breadth and depth of technical, vertical-market and general-interest communities with which to explore and interact on a local or worldwide basis. Content and service providers will discover the opportunity to extend their brands, products and businesses in a graphically robust, interactive manner.

The Microsoft Network will further facilitate a higher quality of content and services by providing tools and technology to support a truly personalized, multimedia presentation.

World-Class Communications

Access to communication abilities that transcend the traditional boundaries of time, cost, resources and geography is one of the primary benefits of the Microsoft online system.

The Microsoft Network is designed from the ground up to accommodate full international access. The ability to access the network from anywhere in the world — and to exchange information with users anywhere in the world — makes the system even more compelling.

By integrating with the Information Exchange in Windows 95 and extending this functionality to bulletin board, file library and "chat" services, The Microsoft Network will offer the most robust, easy-to-use communication capabilities.

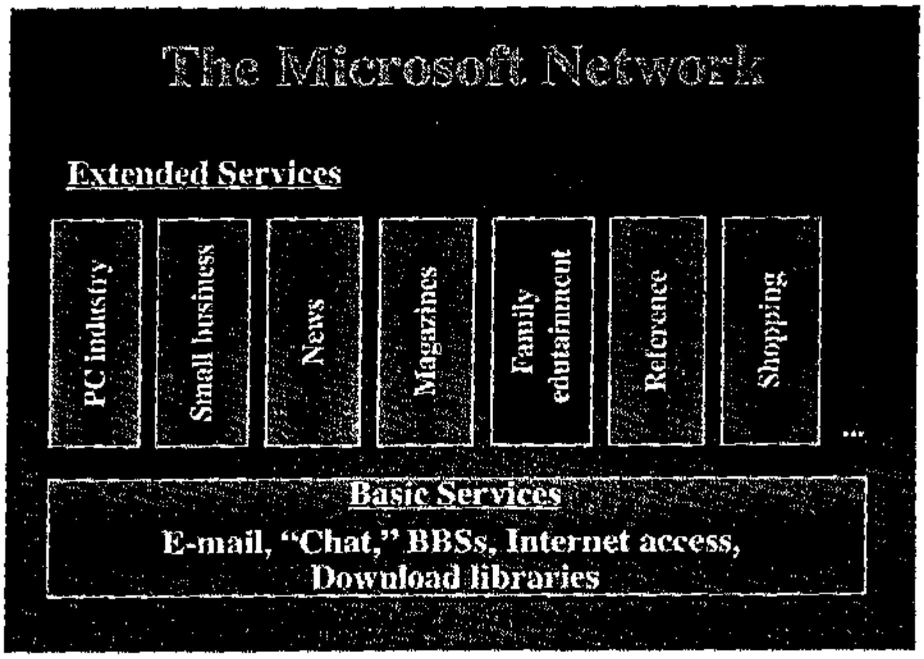
Consistent user interface, OLE support, drag-and-drop management, Explorer navigation, and rich-text formatting all extend the consistent experience of using Windows 95 within The Microsoft Network communication environment.

The Microsoft Network Services

When The Microsoft Network becomes available, subscribers will have access to the following basic services:

- Rich communications features, including e-mail, bulletin boards and "chat" services
- Internet access, including e-mail and news groups

- Information services, including news, sports, stock and weather reports, product and product-support information, and special-interest group information
- File download libraries — shareware, graphics and wave files, applets, product support, article archives, and the Microsoft Knowledge Base
- Microsoft information and support for customer service, product information and technical support



The Microsoft Network's extended services and products will include both Microsoft-branded and independently branded options available to users. Some will be available for an additional charge, and others will be available at no additional charge, with their revenues coming from advertising or shopping-transaction fees.

Conclusion

By combining unprecedented access and ease of use, a new business model to attract and reward independent content and service providers, a compelling online experience, and

- more -

world-class communications, The Microsoft Network will spur the market for online services, closing the gap that exists today between the potential and actual installed base for these services. In bringing highly functional and entertaining online service to millions of users, Microsoft is further realizing its mission of enabling "Information At Your Fingertips".

#########