PLAINTIFF'S EXHIBIT
1354
Comes v. Microsoft

Microsoft Confidential
Chicago

Chicago Strategy Document

"Making People More Productive"
June 16, 1992

Send feedback to: Brad Silverberg; David Cole; Eric Rudder; John Ludwig; Mack Mccauley

What is Chicago

Chicago is the code name for the next major release of Windows from the Personal Systems Group. It's an integrated and complete Windows operating system which starts with the basic functionality found today in MS-DOS, Windows, and soon to be released Windows for Workgroups. It adds overall improved ease of use, enhanced workgroup function, ease of finding and sharing information, advanced operating system performance, and enables new application capability.

The phrase "integrated and complete Windows operating system" is key, it means that Chicago will be the first high volume desktop release of Windows which does not sit on top of or require any version of DOS. Chicago will be a compelling upgrade for existing Windows users and a compelling new computing environment for people who have not yet moved to Windows.

The Competition

Competition in the operating system business is intense. There are a number of dire competitive threats which Chicago must address.

- IBM is after the desktop. OS/2 2.0 is not dead, in fact it's doing better than expected. (expectation was that it would pretty much ship dead.) OS/2 is selling because of it's DOS app support, appeal of the workplace shell, image of being a stable system, better multitasking, and promise of power from pre-emption, threads and 32-bits. We must respond with a powerful, 32-bit system, which is much smaller and much faster than OS/2 so it runs on average machines. It must be completely compatible with DOS and Windows apps and device drivers. By meeting these goals, Chicago will put OS/2 into it's grave.
- Novell is after the desktop. As you know, they have acquired Digital Research and are now working hard to tightly integrate DR-DOS with Netware. We should also assume they are working on a Windows clone and/or that they are working on a virtualized DOS environment which will run standard mode Windows as a client. This is perhaps our biggest threat. We must respond in a strong way by making Chicago a complete Windows operating system, from boot-up to shut-down. There will be no place or need on a Chicago machine for DR-DOS (or any DOS). Since we don't expect Novell to go away anytime soon, Chicago must ship with great Novell client support. We must build all the things into Chicago which are required to make it THE ideal Netware client.
- Lotus is trying to set the agenda for workgroup computing with Notes and mail technology. We are behind and must catch up fast. The first release of Windows for Workgroups will put us in the game. Chicago must put it's stake in the ground by defining a low-end base platform for workgroup computing and thus trap Lotus at the high where volume is low.
- The collapse of PC OEMs has left the industry visionless and at the mercy of IBM and Apple. Practically no ease-of-use innovations have materialized. Microsoft has the moral imperative to fill the void and drive the PC industry forward, particularly in ease of use. We are the only company which can really take leadership for the rest of the industry. Chicago must deliver on the vision of making the PCs easier to use by enabling plug-and-play of peripheral devices. Chicago absolutely MUST be easier to use in all respects than today's system. It's essential to our leadership role that Chicago be the product that cements Windows as the driving force behind innovation in the PC industry.
- There are many vendors out to kill Windows in the x86 graphical user environment; Next, Solaris, GeoWorks, just to name a few. Chicago must be a significant and compelling upgrade from Windows 3.1 to maintain the high volume needed to keep ISVs, Corps, and end users thinking about Windows, Windows, Windows.

What makes Chicago great

Chicago will be very competitive and compelling product because:

- It's an integrated and complete Windows operating system from boot-up to shut-down. A user will never "return to DOS" to perform a task. The system will be very stable.
- Exploits 32-bits to deliver new levels of performance and power. Great multitasking, and high capacity.
- It's easier to setup, configure, and use.
- Will make a PC fundamentally easier to use. Enables plug-and-play; the user just plugs in a new board and Chicago does the rest to. Improved user interface will make interacting with the system just easier.
- Makes the top 10 common tasks simpler and faster to perform. The user will find that day-to-day common tasks are easier.
- Enables workgroups to be more productive.
- Enables powerful new application capability. 32-bit API, pre-emption, threads, and new APIs for powerful workgroup and application interoperability capabilities.
- Great on common hardware, a 4 meg 386 machine is a great Chicago machine.
- FAST. Chicago will beat all competitors. It will match or beat Windows 3.1.
- Compatible with existing DOS and Windows apps and drivers

Overview of the Chicago Product

There are 5 key areas of improvements which will make Chicago a compelling and competitive product, and an obvious upgrade from all previous versions of Windows. (Requirements like complete compatibility with existing Windows and DOS applications and device drivers, a no-brainer upgrade from previous versions of DOS and Windows, and broad hardware support are assumed obvious and won't be covered further here.) The 5 areas are:

1. Usability
2. Workgroup functionality
3. Connectivity
4. Operating system power
5. New application capability

The rest of this document covers the specific improvements in each of these 5 areas.

1. Usability

We must continue making the system easier to use, our customers are begging us for it. They are paying others lots of money for it. The current dissatisfaction with ease of use in Windows is best quantified by looking at the number of products targeted at improving Windows ease of use; Norton Desktop, hDC Power Launcher, Xsoft's Rooms for Windows, are just to name a few. Our focus on ease of use must be broad and encompass the entire PC system. This means that Chicago must not only improve the Windows 3.1 interface and user model, it must enable plug-and-play of peripheral devices so users don't have to struggle with getting their PC setup properly. In addition to improving the specific areas listed below, we must make the top 10 common tasks simpler and faster to complete than in previous versions of Windows. The ultimate test in overall usability is to ask the question "What would you recommend your mother buy?, Chicago must be the answer.

1.1. Using the Windows system

Note that the following discussing of shell issues merely reflects currently thinking and is not absolute plan of record.

Desktop: The desktop will be utilized to efficiently present all system oriented function like file management, program starting, email, and printing. This means MDI will not be used. Windows for system oriented function will appear directly on the desktop. Drive icons will appear on the desktop as well. Commands for managing the desktop will be easily accessible. [We may also utilize multiple "virtual" desktops to reduce screen clutter and help organize tasks by project. If support for these isn't "built into" Chicago, it can be added via a Topdesk type tool.]

Drag/drop: All drag/drop targets will be readily accessible on the desktop. At the users choice, they will also be available when using full screen applications.

Common commands: Common system commands will be readily available. This will be done via the use of drag/drop targets, built in command palette or toolbar, and a command cache which contains the last "n" system commands performed by the user.

Popup menus: The right mouse button will be use to activate popup menus for items on the desktop. Popup menus will contain shortcuts to common commands for the specific desktop item selected. Popup will replace the menu bar where appropriate, such as cases where the menu bar takes up more room than the client area.

Task switching: The task manager will be replaced with a graphical method of switching between apps. A simple flavor will be tied to the Alt+Tab key sequence for just switching, a more full flavor will be tied to Ctrl+Esc which will enable task management like termination, tiling, etc.

Help: Help will be readily available to the user at all times. This includes context sensitive help in all dialogs as was done in Windows 3.1 and a new dynamic help mode. While in this mode, the user can get help on anything by just pointing the mouse at it.

Command line: To leverage the millions of MS-DOS users in the world, a Windows command line interface with complete command history will be included. All system commands which DOS users are familiar with will be accessible via this interface.

1.2. Use of OLE 2.0

The user interface will be heavily utilize OLE 2.0 and it's user model to deliver better ease of use. Providing a consistent model in user interface for the system with what applications will use is very important to overall usability. This will require a tremendous investment on the Windows and OLE teams to make this truly great, but it must be done.

1.3. Finding and viewing files

Searching: Users will be able to search for files based on MS-DOS attributes of the file and simple ANSI content of the file. All searches can be performed in the background using threads, with the option to have the search window continually updated as in PC Mail. Search by example will be used to help the user fill in proper search criteria. Search criteria can easily be saved and used again at a later time. We need to explore what special search features are needed for searching a peer server.

Viewers: Users will be able to view the contents of a file without loading the application which owns the file. This allows the user to quickly identify if they've located the correct file before taking the time to fully open it with the associated application.

Wizards: Wizards will help users through the process of defining search criteria for locating files. We will also explore using Wizards to help users with other file system tasks, like organizing data files, making backups, etc.

1.4. Setup and configuration

Chicago will make setting up a PC and Windows much easier. This is very hard today. Everything from physically installing a hardware board, to installing DOS and getting all the config.sys and autoexec.bat stuff right, to installing Windows and the right drivers, to configuring all the Windows system parameters, to configuring the desktop arrangement is very hard today and support intensive.

Plug-and-play: In addition to just designing understandable user interfaces and implementing them. Chicago will provide plug-and-play where ever possible. The theme behind plug-and play is that the user just plugs a new device into their PC and Chicago will make sure it properly configured and the right software gets installed to make it work. For example, today it's very complex to install things like sound cards, CD-ROM drives, network cards, and FAX modems. You have to know about irq lines, i/o ports, and memory addressing. plus what all the other devices in the system are using so you don't conflict. Then you're suppose to know what driver to install and how to set it's options so it all works. Basically, it's so hard today that the industry is stifled. While good detection code can help out, that alone can't solve the whole problem. We are working with OEMs and hardware vendors on standards which enable plug-and-play to really happen for Chicago.

System Setup: System setup will do a great job at detecting all the hardware in the system, including things like audio cards, network cards, and CD-ROM drives. It will automatically install and configure the proper drivers. If the user adds a new device after Windows was installed, Windows will recognize this when ever possible and automatically re-configure the system to use the proper drivers, including prompting the user for the appropriate driver disk if needed. Remember, plug-and-play!

The user interface for both express and custom install will be made even simpler than in Windows 3.1. We will explore using wizards to guide the beginning user through custom install should they need to use it. System setup will need to include a host of diagnostics and trouble shooting code as Chicago is an enhanced mode only product. Setup also will to do a great job of upgrading over previous versions of DOS and Windows, we need to learn from our DOS 5.0 and Windows 3.1 experiences and get creative about this.

Configuration management: Outside of initial system setup, all functions to add/delete and modify the configuration needs to get consolidated into one end user tool. It needs to utilize all the same hardware detection and plug-and-play used in initial system setup. This tool is the one the user goes to when they want to change the screen resolution or color depth for example. It makes sense to merge all of this into the control panel. The improvements should include the ability to undo a parameter change, or backout installation of new driver. We will explore where drag/drop installing of things like fonts makes the most sense.

Since Chicago is a complete Windows operating system, configuration management encompasses everything. If the user adds a new device, then Chicago must do all the necessary copying of files and editing of config.sys, autoexec.bat, and the ini files. The user should only install a device once, not once for DOS and once for Window. Same for administering the configuration.

A highly intelligent tool must be developed to handle safe editing of the CONFIG.SYS, AUTOEXEC.BAT and INI files. We will explore using the NT registry model to help manage this mess. User, machine, and workgroup profiles will be utilized to bring management of the system under control. Remote install and configure will be investigated.

Application install/uninstall: A standardized engine for application install and uninstall will be provided in Chicago. A version of this engine already exists and is shipped in the Windows 3.1 SDK.

Auto adaptation: Chicago will self adjust to a new hardware environment automatically. For example, when a mobile computer is plugged or unplugged from a docking station, Chicago will automatically switch to the right set of drivers needed for it to function properly in it's new environment.

1.5. Complete set of system tools in Windows

Chicago will contain a complete set of Windows tools which to manage the system. This includes complete disk management tools; backup, restore, defrag, undelete, virus detection, etc. The user should never return to "DOS" to perform any system management task.

Monitors for disk performance, CPU usage, critical resource usage, and more will be provided so users can easily understand what is going on in the system and make adjustments as required.

A complete, Windows hosted diagnostic tool will be included to aid the user and support staff in tracking down and resolving problems. This tool is currently code named Malone.

1.6. Printing

Printing is one of the most common tasks performed in Windows and one of the highest support issues. There are many ways to improve. Installing network printers can be made easier by leveraging off an existing printer setup on the network. Have a printer drag/drop target on the desktop for each desired printer. Graphical interface for configuring a printer and monitoring status, utilize bi-di communication to make this really useful. Allow MS-DOS apps to print through the Window print spooler. Make the spooler 32-bit so it gets pre-emptively scheduled, use a separate thread per printer. For the mobile market, enable deferred printing.

1.7. Running DOS based applications

Setting properties for DOS apps in the program manager will directly utilize the PIF settings and interface. Multitasking settings in the PIF editor will be simplified. A tight, dynamic link between PIF settings the Settings command on the system men for DOS apps will be created. The title bar of DOS apps will dynamically reflect the currently running app in that VM. Windows apps can be started from a DOS command prompt.

1.8. Improved set of accessories

Chicago will contain a complete set of accessories for managing common tasks. All accessories will be great examples of new concepts we introduce to the system; OLE 2.0, new visuals, use of new controls, new APIs, drag/drop, toolbars, pop-up menus, exposed commands, etc. They will be 32-bit applications and written with the AFX composer. Source code will be included in the SDK for all possible.

- Write: Write will be dropped and replaced with either a WinWord Jr, or an OLE word processing server who's primary use is to compose email. The WinWord team obviously favors the OLE server approach. In either case, the WinWord groups owns this component.
- Cardfile: Should be a simple database front end, use ODBC for accessing a standard list format in the system, and use a new standard grid control.
- Calendar: Use schedule +, expose scheduling APIs.
- Paintbrush: Focus on turning this tool into editor for wall papers and icons, with simple bitmap operations.
- Terminal: Build in scripts for accessing common online information. Accessing Microsoft support services or the Windows driver library are obvious targets. We will also look at developing a protocol for graphical communication,
- Recorder: Explore allowing users to edit macros. Perhaps MS Test be used.
- MSD: Trim down to focus just on windows install/boot issues.
- FAX Manager: consult IFAX group
- Phone Manager: Are there enough standards to do this? This needs to be explored.
- Draw: Add TT special effects to OLE 2.0 MS Draw and ship
- Briefcase: easy to pack up data for home or travel
- Multimedia accessories: these must be improved. Consult MM tech and Foghorn groups.
- Daily views/schemes: These make Windows more fun. This applet will allow third partied to sell packages of views.

1.9. Wizards to help with the difficult tasks

We will explore using Wizards to guide users through any difficult tasks, this can be thought of as another level of help if you will. Guiding a user through the hard process of setting up terminal to communicate with Compuserve is a difficult task and a prime target for a wizard.

1.10. New visuals and controls

We will make use of new 3-d visuals as much as is compatible with existing applications. There will be need for new user interface controls as well, although they are not determined at this time. We should also make use of multimedia in the interface in the interest of being fun.

2. Workgroup functionality

Chicago must establish a platform for workgroup computing that shuts Lotus Notes out of the low end market. To do this it must be simpler to use, easier to install, configure and administer. Not much of the platform here is figure out; things like which security model, what sharing model and namespaces, and ipc and distribution methods must be thought about. However, Chicago is targeting to deliver the end user features listed below.

2.1. Sharing documents

A simple document library service needs to be provided to help workgroups share documents. This includes features like, check in and out, viewing the documents history, and perhaps a standard set of merging API. Making it easy to mail references to documents will be done as well.

2.2. Workflow management

Enabling easy shared preparation of documents needs to be provided. For example, a user should be able to assign a specific section of a document to someone else in the workgroup and have that person get notified they have some work to do. This means Chicago should enable users to assign whole or partial documents to others from the File Manager or application. The status and assignments should be tracked in Scheduler

2.3. Better mail and schedule integration with shell

Mail and scheduling are system oriented functions, and thus need to be integrated well with other system oriented functions. Multiple mail folders should be allowed and presented to the user as directories in the file system, this allows the user to browse messages as though they were just files. Mail box and to-do lists should be drop targets on the desktop.

2.4. Mail and schedule improvement

There are a number of improvements for each of these in to better integration with the shell. Mail: rich text support, mode finder features, integrated logon/setup with Windows, multiple provider support, etc. Schedule: categorization, printing, OLE, list management, rich reminders, shared calendars, and more.

2.5. Automating work group activities.

Bulletin board services and whiteboard/conferencing services should be included to help workgroups be more productive. Applets like a Ticker tape, Voter, Draw straws, Office pool, and some games will be considered.

2.6. Standard address book in the system

There should be a standard address book used in the system for all addressing. Parts of this should come from a sharable source. The address book should be used for email, FAX numbers, phone numbers, etc.

2.7. Mobile computing needs

Chicago will address mobile computing needs be delivering features like: a dialup server support so you can access email and shared data while on the road; wireless media NDIS drivers, deferred printing, deferred email, FAX gateway, Briefcase applet for packing up data and apps for the road; auto adaptation for when the mobile machine is plugged into a docking station; and more.

2.8. View Files

Chicago enable users to look at standard View Files. This enables users to share information without everyone buying a copy of the appropriate application. All that's needed is that the person sending the document around save it in the View File format. [Viewing "View Files" is actually a feature in the help system.]

2.9. Wizards to help with difficult stuff

As is a common theme, wizards will be used to help guide users through difficult tasks.

3. Connectivity

Chicago will be a great system for sharing information and getting at information which is not on your local PC.

3.1. Great Novell support

Chicago isn't a product without great Novell support. There are 3 basic ingredients to this: 1) Owning the VxD and client components and making sure they are bug free, 2) shipping everything in the box needed to make upgrading to Chicago while using a Novell network completely hassle free; and 3) utilize all the Novell user lists and logon data for Chicago security and email.

3.2. Comm sharing/redirection

The ability to share and redirect comm ports and modems in a peer-to-peer networking environment is an obvious thing to do.

3.3. Easy online information access

Terminal (or replacement) will contain built in scripts to allow easy access to services like Compuserve, Online America, and MS PSS services. Getting connected should be as easy as specifying which service and your own location. The scripts will automatically determine the correct phone number and comm settings, log you on and get to the correct service. If it can't get this automated, we'll make use of wizards to guide people through it. We will also do as much as possible to unify the user model for connecting to one of these services and accessing information on a LAN.

3.4. Dialup support for networking

Chicago will utilize the phone as the network. Enabling dialup from anywhere there are phones and modems.

3.5. FAX management

The system will incorporate FAX management as a basic system function, just like email and printing. This includes a simple FAX applet and close integration with email and the forms support built into Chicago.

3.6. Better user interface to the network

Chicago will have a better network resource browsing model. This includes things like flat lists of shares and printers in a workgroup. We will improve the interface for sharing files, printers, and other types of data.

3.7. View Files

This was mentioned in the workgroup section as a primary way to share data without the need to have the app that created the data.

4. Operating system power

Chicago is a high performance, integrated, reliable, 32-bit operating system. It'll be highly responsive and fast system with no internal system limits which prevent users from being productive. Above all else, the system will be reliable. It should never crash and should be protected from bad applications. {Remember that compatibility is an assumed requirement and should be part of every design you think about.]

4.1. Integrated and complete, 32-bit Windows system

Chicago is an integrated Windows operating system, complete with protect mode DOS, protect mode device drivers, and a complete set of tools to manage the system. The user never returns to DOS. In fact, the user is in the Windows system from boot-up to shut down. Chicago is primarily a 32-bit system, it will provide the ultimate in "smooth" multitasking.

4.2. VxD improvements

VxD can now be pagable and dynamically loadable and unloadable. This is important since it gives us the option to page-tune this part of the system which is typically pretty big.

4.3. No Windows heap limits

All internal USER and GDI heap limits will be removed by managing the critical resources in 32-bit heaps.

4.4. Windows improvements

We will improve the windows core in a number of ways. Better DIB processing with a new DIB driver. New APIs to address high demand requests from ISVs.

4.5. Reliable

Chicago will be a very reliable system. We will track down and understand where GP faults and coming from in Windows 3.1, and make sure they don't happen in Chicago. This may be just fixing system bugs, or adding additional protection from bad 16-bit applications. 32-bit applications will run in their own address spaces and the system will have additional levels of protection against them being bad.

4.6. Small, works great on 4 meg 386 systems

Chicago will be great on a 4 meg system. Chicago will perform equal to or better than Windows 3.1 on these systems for all function common to both. All design work, and tuning should be done with a 4 meg system in mind running 2 major applications which are engaged in OLE transactions. The 32-bit app block tuner from Nathan's group will be used where ever possible to help do a great job at making the working set small.

4.7. FAST

As we learned with Windows 3.1, high performance can be one of the most compelling features of a system. Every aspect of Chicago must be optimized to be as fast as possible with the 4 meg machine target machine in mind. This includes basic file system I/O, network access, graphics, user interface, accessory, mail, etc. Chicago performance needs to blow away everything operating system on the market, including previous versions of Windows.

4.8. Better DOS app support

Another lesson we've learned is that great DOS app support sells products. Some user interface improvements were mentioned about. In addition, Chicago will have: super exclusive mode for ultra-compatibility, maximize free memory in a DOS box, improve robustness and improve performance.

4.9. TrueType 2.0

First on the list of improvements here is the 32-bit rasterizer, this is not a TT 2.0 feature by the way. Chicago will include any of the obvious features in TrueType 2.0, direction on what these are will come from the STAT group.

4.10. Tuning of mail, schedule, and winball VxDs

Today these components are all pigs and make the overall system unresponsive and slow. These components must be significantly reduced in size and made fast to the point that using them is transparent.

4.11. Compression built in

Compression technology purchased for Astro will be integrated into the Chicago file system. All areas of the product needing compression should rely on this.

5. Programmability

Productivity centers on great applications. Application vendors should spend their creative efforts on making more powerful, more usable applications. Chicago will enable them to do this.

5.1. 32-bit API, multiple threads, Pre-emption

Chicago will deliver the premiere small, fast, high volume 32-bit application platform. This will enable developers to create more powerful and easy to use applications. Just as applications got more powerful when the system went beyond the 640kb limit and utilized expanded memory, and when the system made use of protect mode memory, user will see more powerful and easy to use applications as the system moves to provide a 32-bit, multi-threaded, pre-emptive environment.

5.2. OLE 2.0

Chicago will include a very good 32-bit implementation of OLE 2.0 which enables great application integration. The systems user interface and accessories will make user of OLE 2.0 and be models for other apps to follow.

5.3. Forms management

A forms architecture will be supplied in Chicago which will free ISVs from inventing their own over and over again. Standard forms in the system will help reduce the working set size for apps needing forms management. Forms registration will be provided to allow apps to pass forms between each other. We will also consider providing a forms creation, editing tool in the product for end users. We are currently investigating which forms technology to use.

5.4. New controls or AFX wrappers (rich text, grid)

A new control for rich text processing will be provided in chicago. It's crazy that ISVs need to keep inventing this on their own. Other standard controls for displaying information in a grid format should be included. The complete list of new controls to add still needs to be determined.

5.5. Workgroup APIs

There are many APIs being added to enable improved workgroup functionality. MAPI and CAPI to tightly integrate applications with system mail and scheduling. ODBC to standardize how databases are connected to. And document library APIs like; check in, check out, merge, view history, versioning to manage document creation.

5.6. Development environment

There are a number of improvements being considered to help make developing Windows applications less difficult. The exact list has not been determined but needs to be thought about.

Products to ship

While Chicago is being developed as a single integrated Windows operating system, it's being designing and built so that 3 specific retail products can be packaged up and sold separately. Which products actually ship other than full Chicago is a marketing issue.

1) Windows for Workgroups. This is the full Chicago product which includes all the function currently found today in MS-DOS, Windows, and Windows for Workgroups, plus the improvements outlined above.

2) Windows. This release contains all the function currently found today in Windows and MS-DOS, plus the non-workgroup related improvements above. This product would exist solely as a lower price point Windows. If the majority of users will pay more money for full Chicago, this product will not exist.

3) MS-DOS. This release contains all the function currently found today in MS-DOS, plus the improvements outlined above which not specific to Windows. This includes the ability to run multiple DOS apps, each in their own virtual machine, and pre-emptively scheduling them.

When does Chicago ship

The competitive environment is real, and it's intense. We must target to ship Chicago in the 9/93 timeframe. It's important that Chicago be a significant upgrade to Windows, it must contain enough new features to be a compelling upgrade. Ideally, full Chicago should be released 12 months from when Windows for Workgroups is released. Given that, the following timeline is what we are following.

	Base	User Interface
Product requirements complete	6/92	7/92
Spec Complete	7/92	9/92
Code Complete	10/92	2/93

	Full Product
Beta 1 (compatibility beta)	1/93
Beta 2 (function complete)	4/93
Beta 3	7/93
RTM	9/93

You will see future memos in the coming weeks describing product milestones in detail. Other than the 9/93 release to mfg date, use this only as a rough guide.