PX 515

PX0515

Erik Stevenson

From: Scott Smith
To: bradsl; bradstr; craige; darbyw; davidcol; dennisad; doughe; glennt; jefft; neffb; tammyst
Cc: scottsm
Subject: WordPerfect status for Chicago meeting <long>
Date: Friday, November 05, 1993 11:35AM

The WordPerfect meeting is now set for the morning of 11/11 starting at 9:00 am. Bradstr has all the details on the meeting location /etc. Approx 12 or so WordPerfect people will be in this meeting.

--scott

==========
WordPerfect product listing for Windows.
WordPerfect 6.0 for Windows
Grammatik 5 for Windows (and other Grammar products. WordPerfect purchased Reference Software about 10 months ago).
WordPerfect In-Forms (forms designer and filler)
WordPerfect Office (groupware - email, scheduling, and calendar all integrated into one product).
WordPerfect Presentations

WordPerfects strategy is to support all platforms and not try to choose sides. They have products for DOS, Mac, Windows, OS/2, VMS, Unix, etc). They just released WP 5.2 for OS/2 and are doing some bundling deals with IBM. This product isn't getting very good reviews. Product was ported to OS/2 from Windows using Micrografx's Mirrors ToolKit.

You will be meeting with a number of people from WordPerfect representing most every product. Dave Moon is their Sr. VP of Development (all products report to him). There are a few other VP's coming and then Directors of Dev and Development Leads. A few of the people in the room work in their Shared Code group. These are the front line guys for working with any new operating systems. They deliver a base set of functionality to each product group.

WordPerfect's Position:
WordPerfect is working on delivering WordPerfect for Windows NT sometime in 1H'94.
This product was initially going to use Win32s but with the amount of thunks they would've had to do they decided this was too much of a hit on their dev. resources so they're going to deliver an NT only product. Other groups at WordPerfect will deliver NT products after the release of WP Win. The product that will most likely be second to come out is the WP Office server.

WP Win is one of the apps that will be shown at Fall Comdex running on

MS7093847
CONFIDENTIAL

EXHIBIT
5 1/8/09
Henrich

PX0515

IBM Power PC.
While you're in Orem, Tom Freeman, WP Win32 developer, will most likely be in Redmond
doing the work to get compiled and running on PPC.

OLE vs OpenDoc: WordPerfect will deliver products with OLE 2.0 support. They've been
working on this for probably 6 to 9 months now. However, their strategic direction is to convert
all OLE 2.0 work to OpenDoc code sometime in late'94? or early 95?. This is part of the
OLE competitor they're working on with Apple, IBM, Novell, and Borland. WP has committed
to delivering the Windows components. Last week they just announced their OpenDoc SDK
(from Wordperfect). yes, this does sound strange. they're supporting OLE but delivering a
competitive product and then planning on throwing their OLE code out. They're doing this
because they want control of their compound document format with OpenDoc. (they're
supporting OLE 2.0 now because they got beat up for not having OLE 1.0 support in a
reasonable timeframe and waiting for final OpenDoc will be too late as well).

Previous meetings with WordPerfect:
January 8th: President, VPs of Dev and VPs of Mktg met with BillG, JonL, and others here
in Redmond.
January: early Chicago and Win32 design preview
March: met to discuss their Win32 / Win32s commitment.
May: met to discuss their Win32 progress and OLE 2.0 progress.
Summer: WordPerfect spent 1 week in porting lab doing OLE 2.0 work.
September: Pre PDC Chicago meeting
(I'm probably missing some other ones here as well).

PX 517

PX0517

Memo

Date: April 7, 1994

To: WWSMM Attendees

From: Doug Henrich

Re: **DRG Business Plan -- ISVs**

Much of Windows' success can be credited to the broad range of available third party applications and hardware products. Our Windows strategy requires that we continue to extend and evolve Windows APIs, SPIs and DDIs which will require the continued support of ISVs (includes IHV efforts, too). Over the next 12 months the main focus of DRG will be to migrate today's Windows applications to Chicago, as well as target new applications for Chicago.

We are challenged by the movement to component software, providing a real paradigm shift for our competitors to exploit. Our initial ISV support for OLE 2 shows positive signs for our continued control of the next generation of APIs for software applications. This will be our 2nd priority.

Beyond the classic desktop productivity applications, it will be critical for DRG to build a strong set of applications supporting the "Enterprise" space. This will include historic mainframe/mid-range ISVs, vertical solution providers, "Solution Providers," and to some extent corporate developers. The platform for these applications will be much broader than the historical operating systems capabilities. Applications will be built with a variety of technology from low-level functionality based on Win32 and OLE 2 technology all the way up to Office-based applications utilizing tools like VBA.

Currently most of the ISVs DRG deals with are based in the US. A few key European companies (like SAP, SAG, Microfocus) also have contacts within DRG. Japan is a special situation where we have the opportunity to build acceptance of Windows applications. We can use the momentum of Windows 3.1J, Chicago and Windows NT-J to establish a dominant market share through localization of major Windows applications by top DRG ISVs.

ISV Taxonomy

When talking about ISVs it is helpful to use the following segmentation. We have adopted these groupings for the purposes of account management.

- **Chicago ISVs (Productivity and Desktop focus)** - Includes key applications in each major application category.
  - **Tier "A" ISVs** - These "key" ISVs represent the majority of application software sales. As developers of general interest (productivity) software they also receive the most attention from the trade press and corporate accounts. Their support of Windows is critical to the general perceptions of its success:
    Adobe Systems, Aldus, America Online, Arabesque Software, Attachmate, Banyan, Berkeley, Borland Int'l, Brightworks, Broderbund, Central Point Software, Claris, Compton's, CompuServe, Computer Associates, Corel, D&B, Datastorm, Delrina, Electronic Arts, Frye, Intuit, Lotus Development, Microcomm, Micrografx, Inc, Novell, Polaris, Prodigy, Quark, Saber, Software Publishing, Sony Publishing, Spinnaker Software, Symantec/Norton, Wall Data, Word Perfect
  - **Tier "B" ISVs** - These 'second tier' ISVs represent significant volume sales of software but often in less general areas of interest:
    Altamira, Artisoft, Autodesk, Inc, CMS, Cognos, Conner, DCA, DEC, Deneba Software, Eicon, Envisor Manager Software, Farallon, Folio, Frame Technology, Franklin Quest, FTP Software, FutureSoft, Great Plains, Grolier Electronic Publishing, Gupta, Helix Software, Hilgraeve, Inc., IBI, Information Resources, Inset Systems, Mathsoft, Meos Software, Netsoft, Ocean Isle, Pilot Software, Quest, SAS Institute, Shapeware, Shiva, SPSS, Sybase, The Periscope Co., Traveling Software, U-Lead Systems, Virtus Corp, Wolfram
  - **Tier "C" ISVs** - A universe of 1500 ISVs that bubble up and down from "A" or "B" status.
- **Development Tools** - Getting support for our new APIs requires the support of key development tools. The leading tools vendors include Borland Int'l, Digitalk, Powersoft, Symantec, and Blue Sky. Beyond these key ISVs there are additional players and categories that can not be ignored. The taxonomy broadly includes: low-level compilers & debuggers for many languages (C, C++, Fortran, Cobol, etc.), GUI layout tools, scripting languages, case tools, and high-level, application-builder products.
- **Enterprise ISVs** - A new category for DRG, with the potential of taking us into the corporate development arena, supports re-engineering and client/server projects. Including the following:

MS7045839
CONFIDENTIAL

Microsoft Confidential

Plaintiff's Exhibit
9497
Comes V. Microsoft

EXHIBIT
10/8/09
Heinrich

- – **Vertical Applications** -- DRG will align with the vertical markets targeted by SP marketing organization. These verticals will include segments like healthcare, financial services, banking, engineering, manufacturing, insurance, and several others.
- – **Mainframe & Mid-range ISVs** -- This segment includes many of the largest software vendors in the world. Historically, they have not done much work for the Windows platform, and what they have done is usually not innovative. Companies like SAG, SAP, CA, D&B, Lawson, and MSA will be targeted.
- – **AS/400** -- Compared to other platforms, the AS/400 market is a widely distributed software market. DRG has started working with a number of vendors who provide Mid-range and Mainframe business applications including: JD Edwards, SSA, Lawson, D&B, SAP, System 2000, Emphysis, etc.
- – **IBM** -- Though it has required time consuming efforts DRG has commitments for Windows NT versions of CICS and DB2.
- – **OLTP** -- Getting credibility for Windows NT in transaction processing requires that we get support from the OLTP community. In addition to working with IBM on CICS, DRG is working with Transarc and Unisys for a version of Tuxedo and with NCR on an implementation of TopEnd for Windows NT.
- – **Systems Management** -- Building our Hermes story involves working 50-60 players across the whole spectrum of systems management companies (including IBM, DEC, SNI, CA, Legent, Tivoli, SUN, HP, Brightworks).

- **Server ISVs** - Establishing Windows NT as a credible server platform will require convincing the leading server applications to port from UNIX and OS/2.

  Lotus Notes, Oracle, Ingres, Informix, IBM w/DB2, Brieve Technologies, FileNet, Keyfile, Viewstar, Wang Labs, Borland International (InterBase), Coromandel Industries, Cracchiolo & Fedor, Desktop Data, Kofax Image Products, Lucas Management Systems, Micro Decisionware, Inc., Saros Corporation, Imara Research. Plus 20 other interesting applications, comm, text, image, telephony, object dbms.

- **UNIX & Technical Workstation ISVs** - Windows NT has the capabilities (and we have the tools support) to make it a credible alternative to UNIX for workstation applications. While nearly all of these ISVs are new to the PC marketplace, DRG has had success with a wide range of nearly 200 ISVs including:

  Parametric, Intergraph, Autodesk, Bentley Systems, Cadence Design Systems, Computervision, Intergraph, Ithaca Software, Mentor Graphics, National Instruments, PAN Data, Parametric Technologies Corporation, Viewlogic, Visual Numerics, Accel Technologies, Adra Systems, Ansoft, Cadbooks, CNC Software, Consensys, Erdas, GrayTech Software, Masstock, Ltd., Orcad, PADS Software, Recal-Redac, Rasna, SDRC, Spatial Technology, Swanson Analysis Systems

- **Multimedia** - Over the past few years Apple has established the perception of a significant lead in multimedia applications. DRG is supporting a select group of home, entertainment, games, and in particular, content related ISVs for development of titles for Windows.

- **Personal Devices** - DRG is evangelizing over 60 ISVs developing applications for WinPAD.

- **IHVs** - DRG has targeted the key players in each device area (such as display, net cards, SCSI, CD-ROMs,..). More than 2,000 hardware vendors are being evangelized in one-to-many fashion to build PnP hardware.

In addition to the groups listed above there are roughly 3,000+ ISVs being tracked that have identified themselves as developing Windows applications or supporting hardware. They are largely supported by one-to-many programs like broadcast email, fax, mailings and call-downs.

Objectives

The goal of DRG's ISV programs is to maximize support of our systems products and technologies. The following are the goals for the next year.

⇒ **Products**

- **Chicago**
  - – Move 16-bit windows applications to 32-bit and exploit Chicago opportunities (shell extensibility, viewers, PnP events, LFNs, OLE 2, new UI, MAPI, TAPI). 100 key productivity and desktop applications are targeted.
  - – Have major Windows application available at Chicago ship (Lotus, Word Perfect, Borland...)
  - – Get new exciting applications for Chicago. Multimedia titles, content, and games will be a huge opportunity.
  - – Focus on exploiting the *Information Highway* hype. Work with CIS, America Online, Prodigy, and the many apps, utilities, info services, and communication products to own this emerging marketplace. This is a could be a huge win for our platforms.
  - – Work to insure full compatibility of existing 16-bit and 32-bit shipping applications.

MS7045840
CONFIDENTIAL

- **Windows NT** (Daytona) - Continued focus on server applications (~10 major applications) and technical workstation applications (>200 ISVs). DRG will use Windows NT as the foundation for moving enterprise applications (including verticals) from high-end platforms to Windows NT.
- **EMS** - ISVs identified in 7 major categories of workgroup computing. WGA is driving business arrangements with top tier ISVs in each category, 2nd tier ISVs are being evangelized by DRG to exploit Capone, MAPI, and EMS.
- **Cairo** - Get early adopters to demonstrate potential Cairo solutions. Various new features and interfaces "open processed" with the ISV community. Interfaces like DNA & OCX, plus many other new capabilities introduced will have wide spread ISV adoption by the time Cairo ships.
- **RISC** - Leverage OEMs of Windows NT. Utilize VC++ targeted for MIPS, Alpha, and PowerPC to motivate ISVs to build applications for all the Windows NT platforms and ship on a single CD. Leverage the RISC OEMs (IBM, DEC, MIPS) to do the majority of the work.
- **WinPAD** - Establish WinPAD as the premier PDA-like device through supplying an interesting assortment of ISV applications at product ship. More than 60 ISVs currently building WinPAD targeted applications.

⇒ **Technologies**

- **Win32** - Transition all application development to the 32-bit world of Win32 and OLE 2 (32-bit). Chicago will drive the major players to move to Win32. Leverage MS applications shipping for Windows NT to drive more widespread adoption of desktop productivity applications. More than 300 Win32-based application now shipping.
- **OLE** - Continue the momentum established by the initial wave of ISVs shipping OLE 2 enabled applications. Move OLE 2 support to be a required feature of all Windows applications. Motivate ISVs to exploit new interfaces defined by OLE 2, including DNA and the new control architecture (OCX). Work with DEC, and other platform vendors to establish COM as cross platform.
- **WOSA** - establish each service API (ODBC, MAPI, TAPI...)
- **PnP** - establish PnP as the defacto standard for next generation hardware for all major devices (networking, display, SCSI, storage...)
- **Multimedia** - Establish Windows as the leading platform for multimedia applications and titles. More than 30 development tools, and several hundred titles.

Strategy

- Continue DRG's efforts to migrate today's base of applications to exploit Chicago. Get new exciting applications for the platform (networking and communications products, multimedia, games, PnP, etc...). Focus the DRG Porting Lab on moving existing Windows applications to Chicago. Create demand for new "Chicago" applications through tradeshow exhibits of applications, roadshow displays, and various press-related activities. Ensure that Windows NT applications continue to be developed. Make sure Chicago applications are tested and compatible with Windows NT.
- Expand the universe of ISVs we target. Our motto is "a new ISV everyday." Target new ISVs on all ends of the spectrum - high-end enterprise ISVs, info highway players, new, low-end solutions centered around WinPAD, etc... Target the new community of content providers to create more exciting multimedia titles. Own the content industry for our platforms.
- Establish an evangelism effort focused on the complete "enterprise" ISV applications needed. This effort will require significant product and platform investment to provide the foundation for successful deployment of new systems and compatibility with existing applications.
- Provide a broader set of services to the ISV community. This set of services should include:
  - leverage of non-systems products (primarily DDT products and other developer services like MSDN)
  - establishment of marketing services to help ISVs better establish their products in the new markets - component s/w world, the PDA space, and new hardware spaces (RISC, PnP, WinPAD). Some of these may require joint marketing from MS to establish the platform.
- Leverage RISC OEMs of Windows NT (DEC, MIPS, and IBM) each of who has an ISV program to support their implementations.
- Actively sell ISVs into the SP channel as appropriate. More ISVs should (and do) want to sign SP deals to ship and support MS products.

- Account Management - DRG evangelists have responsibility for direct contact with the top 300 ISVs

Competitive Strategies

Our major competitors (Novell, Lotus, and IBM) have all established DRG-like efforts. So far, most have been primarily one-to-many evangelism efforts.

- Lotus is focused on Notes add-in products. We are targeting these same people with EMS and MAPI.
- It is not clear what IBM is doing with their developer related programs. They seemed to be on a path to clone MSDN. Lately, this has been re-organized and massively downsized. The PowerPC group porting Windows NT is cloning DRG's strategy and targeting our Win32 ISVs to recompile their Windows NT applications for the PowerPC version of Windows NT.
- Novell does not have a coherent developer effort. They have a developer program, but not a DRG-like team. Brainshare is a annual PDC style event with no clear focus for developers. It is much more like TechEd focused on covering all developer issues (not just strategic issues). AppWare has been targeted at specific ISVs (mainly tools vendors) to provide components (ALM - AppWare Loadable Modules). It is not clear how effective this approach is, but we can not discount this effort.
- The OpenDoc initiative has its own evangelism effort. They are good at targeting key ISVs, but lack the coordination to do broad evangelism.

MS is constantly positioned as the closed Proprietary solution. This will encourage continued EBM (everybody but Microsoft) efforts in the future. Our best response is:

- Continue to consume all ISV bandwidth so they can not target other competitive platforms. OLE2, Chicago, Cairo, new technology design previews are all used to consume developer bandwidth.
- Continue to provide compelling platform and business opportunities.
- Open Process - collaborate with the software development community to define defacto standards. Design previews used to get buy in from ISVs. Workshops / Account management keep ISVs on our path.
- Carefully evaluate the tactical & strategic ABM efforts. Give ground on tactical, meaningless battles to bolster "Open" industry perception. Attack aggressively ABM initiatives that are strategic.

We should not be paranoid but recognize that our competitors will continue to band together in a variety of efforts directed against us. Unanticipated mergers or acquisitions may also create more powerful competitors.

MS7045842
CONFIDENTIAL

PX 524

PX0524

MICROSOFT CONFIDENTIAL

To: Bill Gates, Steve Ballmer, Mike Murray

Cc: Mike Hallman, Paul Allen, Nathan Myhrvold, Jeff Raikes, Darryl Rubin, Brad Silverberg, Mike Maples, Cameron Myhrvold, Paul Maritz, Bob Muglia, Russ Werner, Jon Lazarus, Steve Madigan, Richard MacAniff, Brian Valentine, Dave Cutler, Russ Siegelman, David Thacher, Darren Remington

From: Jim Allchin

Date: 1/13/91

Subject: Thoughts on a Distributed Computing Strategy

The original purpose of this memo was simply to force me to privately think through some thoughts on our strategy. I am distributing it to you to record some of these thoughts and perhaps to act as a catalyst for future discussions. I have continued to refine my thinking since this was written, but I felt it would be good to begin getting feedback from everyone on my thoughts.

1. Today - Strategy and Market Position

Today, Microsoft networking products are perceived as 2nd or 3rd to Novell. Microsoft is directly compared to Novell. This isn't surprising since the strategy is "beat Novell". I call this the "out Novell Novell" strategy. It is a frontal attack. It is a features war, a comparison war, a mindshare war, SE war, *etc.* Clearly, Novell is doing well so it isn't wrong to go after them. However, how well such a strategy works depends on several factors:

- o correlation to long term vision (*viz.*, will this get us to where we want to go?)
- o competitive product feature set
- o the "right" channel(s)
- o cash to sustain attack

File: Networks

Microsoft Works

o Correlation to long term vision

I believe the vision is clear for Systems. We need to provide *the* platform for applications. That really is all an operating system is: a platform for applications. In the old days this meant abstracting the peripheral interfaces, efficiently managing the resources of the computer, and providing a secure environment for applications.

The goal is still the same today even though we live in a more rich and decentralized environment. All the basic functions that have heretofore existed in a "timesharing" system have appropriate counterparts in a decentralized system. For example, abstraction of physical resources, management of objects, security, accounting, *etc.* all must be provided. Clearly, the problems are much more difficult because of propagation times and a new range of potential failures; but, a globally transparent system is still the goal.

Nevertheless, there are many people asking what NBU should be when it grows up. They pose the question as "What business should we be in?" This is a clear sign of internal knowledge that the current path may not be the right one long term. I also have found many projects/ideas floating ... looking for the answer: Windows/N,

EXHIBIT Allchin 1 1-8-09 JA

DEPOSITION EXHIBIT 1 4/15/01

MS 0119610
CONFIDENTIAL

PX0524

Windows Server, buy this company or that company, application products to address LM weaknesses, *etc.* My conclusion is that the historic strategy built on "out Novelling Novell" cannot create long term, sustainable competitive advantage.

o Competitive product feature set

I believe the current LM 2.0 product is not terribly competitive with Novell (and certainly not Banyan). I say this for several reasons. First, Novell has a nice product; it's not great, but it's not bad either for their market segment. Second, LM 2.0 is not technically on par with Netware/386. (I don't want to start a religious war with that comment, but even though LM 2.0 has the checkoff items like fault tolerance, ease of use, *etc.*, the implementation is very weak. Consider fault tolerance. What good is a mirrored disk system if you can't mirror the boot device (the disk that probably receives the heaviest activity)? That means that two disk systems aren't supported and LM can not guarantee that random disk failure in any system won't disable the system.) Most importantly though, Novell has a huge installed base. There is no compelling reason why anyone should switch from Novell. Yes, MS can fight with Novell over new installations as the market continues to grow and that will be a significant market. However, Novell has the leader's mindshare advantage even with these new sites.

Furthermore, I can't see why LM 3.0 as planned will make the situation that much better. The Netware feature set will continue to expand. Basically, it's back to trying to out Novell Novell. Functional areas such as Directory and Security are simply expected. These areas won't be reasons to consider Microsoft over Novell. In addition, I believe the connectivity capability of Novell far exceeds Microsoft and is growing rapidly. LM has very primitive connectivity options planned. Worse yet, there are few developers here who understand this technology area. Remember Novell bought Excelan to gain their expertise.

What's most frightening is that DCE will totally change the face of the landscape during the next few years. LM 3.0 on its current path could be a niche product without backing from IBM or DEC. Many believe that DCE will lay the foundation for the services of the future. So, what will be the Microsoft advantage?

A fundamental problem is that Microsoft has fallen trap to the "networking is separate" model. That is, OS and networking are different pieces and can be developed and marketed separately. Novell is on their way to migrate to a more "general purpose" OS (with networking embedded). Microsoft, on the other hand, has OS and networking separate.

I know some may not like the following statement, but ... It is my belief that LM is making headway today because of only one reason: the Microsoft name and the IBM connection. MIS people says "hey, we better watch them, they are aligned with IBM". It's FUD. Consider what would happen if IBM sold Netware (as a strategic product -- not just their education group) or partnered with Novell on networking. With the current product and strategy, the game for Microsoft would be over with Novell the clear undefeated winner. Lucky for us, this is unlikely.

There is a separate low end to the market. LM today is also unsuited for this. LM is heavy duty -- not simple, inexpensive and easy to use. (All products should be easy to use, but small peer networks have unique requirements along this line.)

o The "right" channel(s)

Microsoft has a complex channel strategy. Even though Microsoft knocks Novell for their "7-11" outlet approach, Microsoft has another problem which could be severe. OEM distribution works great when a product is bundled with another orthogonal product (*e.g.*, Windows with new PCs) or there is not a "packaged" product available. Unfortunately, what has happened in LM is that OEMs have added value that is user visible (*e.g.*, LS logical

PX0524

server or admin UI). On top of this we've introduced retail sales. This, of course, leads to two problems: "Will the real LM stand up?" and OEMs competing with us.

Simply put, if an OEM is going to port the product AS IS (*e.g.*, only adding new drivers, *etc.*), it works great. If an OEM adds value, but there is no "standard" packaged product, then that also works great. Today, neither of those situations is true for LM.

Finally, the market that LM is targeting is high end. Only the most sophisticated resellers are capable of addressing the LM market. I'm sure the strategy is to sell to the top corporate customers and that will win with the smaller customers over time, *etc.* (Of course, this is the area of the market that requires WAN, Directory, network management, advanced security, *etc.* -- all the things that we don't have right now.)

I understand why we took the paths that we did and see how we are working our way out of the confusion. Nevertheless, it will take quite a while to change the perceptions in the marketplace. Networking may be leading the way for a continued movement away from OEM and toward packaged products. LM and Windows are both examples.

o Cash to sustain attack

Let's look at the current annual run rates extrapolated from November actuals. Expenses and overhead appear to be around $73M. Assuming no more headcount (we all believe that, huh?) breakeven won't be until around 73x1.18 = 86.1! (This is assuming that current COGS ratio.) In November, the revenue run rate was around $29M. Thus, the business must get 3x bigger to even break even. (It is also important to remember that the OEM revenue in most cases comes from honoring the contract minimums. If they back away (go with DCE, Novell or IBM!), then we are at risk in this area.)

While I'm sure that the plan shows great improvement in units, I also expect that it has extra headcount as well. Given my comments in the prior sections, it is clear to me that all in all this will be a *very* expensive and prolonged attack.

2. Thoughts on a New Strategy

2.1 General Systems Strategy

How do the products of Systems fit into the overall mission of owning system software? After being here for several weeks, I'm still confused. I'm confused because we mix the marketing of a strategy with its implementation. The momentum of Windows has created "Windows" mania. I feel the following should be answered. Please note that we can change the marketing message and product names to ride the Windows wave later. Given the current names (applied to functionality), we need to answer the fundamental issues.

- What is Windows? Is it a GUI on DOS and NT? Or is it an operating system?

  I contend that this decision will dramatically change the strategy. Today we basically have a 3 operating system strategy (DOS, Windows, NT/OS2 -- 4 if you separate OS2 and NT). I strongly doubt whether such a tier environment can be built without functionality leakage eliminating needed differentiation and whether the marketplace could possibly understand a 3 OS strategy.

- What is the OS family? In defensible terms, what market or functionality is different between each member of this family? Clearly, if you have a family there must be a clear reason for each member.

PX0524

- What is the market and distribution plan for NT? That should drive the functionality and time to market issues.

- Why are we packaging and selling the basic building networking software separate from the operating system? We have legitimized the "NOS" concept instead of obsoleting it by putting this capability into the OS.

It is important when thinking about these issues to not get confused over the names...we could migrate to a total Windows named product line (or anything else) if we wanted; it's the functionality that counts.

2.2. Why is distributed computing important?

I believe there are three reasons why distributed computing is critical to Microsoft.

- Distributed computing is a fundamental building block for the future of computing. Concepts from distributed processing will permeate most every software system in the future. Microsoft must acquire and embed this knowledge for these future applications. There will be no Information at Your Fingertips without this knowledge.

- Microsoft set the application framework for the desktop. We must expand this framework for distributed applications -- or risk losing control of that business. Today, Microsoft has an advantage because the applications are written to a platform base (*e.g.*, DOS, Windows, OS/2) that is controlled by Microsoft. The platform for distributed applications is just as important.

- There are new business opportunities created by networks. There is a significant profit opportunity through providing distributed services for networks. In essence, these distributed services are a special type of distributed application which extends the standard OS features into the distributed arena.

PX0524

MICROSOFT CONFIDENTIAL

2.3 The Network Marketplace

The network systems marketplace today is divided into three market segments:

Segment	Examples	Volume	Price(server)	Key Features	Selling Method
Workgroup (<10 nodes-no interconns)	ELSI/II, Lantastic, 10Net	High	Low (Street 3k)	Simple, Simple, Simple Inexpensive	Naive user (the ultimate bottom up sale) (Everyone sells)
Department (small groups within larger orgs)	Netware 286, LM, LS, Vines	Medium	Medium (Street 1.5k)	Application compatibility Growth	Bottom up (Resellers+ Seminars, etc.)
Organizational (typically geographically dispersed departments)	Vines, Netware 386, LS, LM	Low	High (Street 5K++ - not price sensitive)	distributed mini capability Ease of Admin	Top down (IS - together with departments) (best with direct sales or high end resellers)

Today Novell owns the workgroup and the department. Banyan owns the organization.

It is interesting to note, that it is possible to have a product in the workgroup space without being in the Department or Organizational space. They are completely different areas. This isn't true in the other two areas. Simply by the nature of network growth, IS involvement, and competition, a company must offer products in both the Department and Organization space.

The Department,and especially the Organization, arena would have been owned by IBM historically. I'm sure that IBM sees retailing LM as a very significant threat. Once they lose the network, they are on their way to losing control of the account. Thus, it is clear to me that on the current path, IBM is a competitor -- not an ally. I do not think this is a good idea (...yet...). Networks require significant handholding. IS is beginning to understand this fact. IBM and handholding are synonymous. In order to really match IBM/DEC/*etc.* it will be necessary to get the channel as capable as direct SEs to support the networks. This is a very large undertaking. The larger, more complex the network, the more difficult the job. The resellers make little margin so they can't invest and their internal support staffs turn over rapidly so there isn't a constant support system in place. The point: LM is attacking the most complicated area and on a path of doing it alone.

The following is obvious, but reminding ourselves of the perceptions (in the NOS world) is important:

Novell	the leader, fast, a winner, "why shouldn't I buy them?"
Banyan	the dark horse, best technology, poor -- but improving marketing, directory, best WAN solution

MS 0119614
CONFIDENTIAL

PX0524

Microsoft IBM connection, significant investments, weak product, generally a winner -- but not in networking

2.4 An applications driven world

I completely agree with Bill's memo. Applications drive the world. Applications are the reason that the VAX was so successful. Applications make people switch computer systems and vendors. Assuming that LM stays in its current market segments, applications are the only thing that will allow Microsoft to win against Novell.

Therefore, we should strive at building the best distributed computing platform in the world. We should then promote this to every applications' writer to ensure that if it's distributed, then the server side *must* run in this environment.

Today, the general APIs used in the marketplace from LM deal with named pipes/mail slots (and even mail slots aren't used externally). There are two reasons. First, no other API is required to build a user oriented "client/server" application (*e.g.*, all DDBs have their own security model built in and the vendors know the LM security is poor and going to change). Second, few ISVs focusing on administrator oriented applications (*e.g.*, network backup) would write to the other APIs from LM because of marketshare. I contend that solving the user oriented application lock-in is much more powerful than the simple network utilities that could be written using the other APIs. Most all the currently available network utilities will eventually be built into systems so it is a transient lock-in. More importantly, end user productivity gains by applications on the network are a much more compelling reason for choosing a particular network.

No one is going to switch from their current system for a better file or print server. No one is going to switch for a "better" directory, *etc.* There has to be a significant reason which will increase their productivity (*e.g.*, an application, a new data model) to change or add a new product. The same logic holds for new sales. Why choose one system over another? LM needs a competitive advantage -- applications are the key. The applications could be Microsoft provided distributed services such as a licensing system or software distribution manager or something like a client/server version of the ASK system ported to OS/2. This represents the best Trojan horse strategy against Novell and the best long term lock on accounts in the future.

2.5 What technology is needed for Distributed Computing

Distributed applications need a framework to execute in. What are the missing pieces from a standalone operating system? I often think about OSI layer 7 being divided into sublayers for simplicity.

Distributed Applications	*e.g.*, Calendaring
Network Services	foreign file, foreign print, DDB, distribution, messaging, transaction processing monitor, *etc.*
System Services	Directory, Security, System Services Mgt, Time
Underlying Host OS/Transport System	OS + Transports + remote file and peripheral support

PX0524

The MAC is network ready. Unix is network ready. Both environments come with a standard interface into the network. (Macs use AFP/PAP through AT, and Unix uses NFS through TCP/IP.) Why is Microsoft shipping a desktop OS that is not network ready? (The number of PCs which are being connected is growing rapidly and we're letting others (Novell) put fundamental technology (a redirector) into the operating system.) The way to neutralize Novell is to embed this level of functionality (and more) into the operating system. It's the right thing to do even ignoring Novell for providing a consistent platform for decentralized applications.

In addition to redirection, the operating system should provide homogeneous transparent remote file and peripheral access. An addon service (as is done today) could be used, but as the semantics of the file systems become richer (e.g., transactions), it will become very difficult to layer these pieces. High performance is impossible to achieve without this tight coupling. Foreign clients accessing the node's resources can be built above the transparent structure already in place. This only works well if the semantics of the underlying system can accommodate the foreign system's semantics naturally.

The gaping hole today are the system service pieces. Again, the desire is to built a system which matches a typical minicomputer system -- only decentralized. Therefore, a directory, security, and service management model which spans nodes is fundamental. This should not be an addon.

The upper layer services are "applications". You could think of them as utilities in the old OS terminology. These distributed services can add significantly to ease to use, *etc.* They can add the intelligence to make such an environment manageable. *A fundamental problem with all systems today is that they are too hard to use. The system must figure more out using heuristic rules concerning the user, administrator and the current state of the system.* Customers will pay for ease of use. This follows the principle of making the user more productive.

2.6 The IBM Connection

Here is my observations of the IBM situation and its effect on our position:

- IBM feels screwed by Windows vs. OS/2.. They feel that they got left holding the bag on "old" development and, in addition, that Windows stands against the lock they were trying to establish for new applications. I believe there is a significant amount of old baggage hanging around dealing with past shared development projects, *etc.* (Building an operating system between two companies is certainly a difficult feat (impossible?); however, I believe there are many opportunities for a close development relationship with IBM.)

- We want IBM to be a distribution outlet. On the other hand, IBM wants to control the direction and, in particular, they want customers to perceive IBM as the "owner" of the future (MS is simply the implementor). I don't believe Microsoft is anywhere near as strong without IBM (marketing-wise). We are on a collision course with IBM today. I expect that someday we will eventually collide, but do we have to collide now? (I mean geez, I just got here and we're going to war?)

- IBM is committed to DCE. Their plan is not to adopt LM 3.0 long term. They want us to converge LS and LM so that they don't have to continue to invest resources in this deadend since they are moving to DCE. They also want to eliminate LM's temporary feature advantage over LS.

- If LS and LM don't converge, then IBM will undoubtable intensify their efforts of competition with us. They will show their migration strategy to DCE customers and how Microsoft is not with the program unless we have a great story. Nevertheless, IBM will win in large accounts. Actually, Novell will win the most because of the fragmented networking strategy between IBM and MS.

PX0524

- IBM doesn't understand DCE. This is clear in my past conversations with them. We have an opportunity to take a lead with them. (Block diagrams look so good. However, they are usually quite difficult to put together when they are done by different groups (using difficult fundamental models)).

2.7 Some straw ideas

Consider the following breakdown of products:

- Distributed Computing Tools. Tools are the needed catalyst for making distributed applications grow. RPC is the first on the list, but there are many more. For example, as soon as there is a standard for RPC, then remote debugging tools will be an opportunity. This is a new area but I believe there are many important tools such as remote transaction libraries, *etc.* that would really help the application writer. Furthermore, it is my belief that new extensions to languages (*e.g.*, "Distributed C++) are needed to address these new distributed environments. He who has them first will have the advantage.

- Windows/N. In my view, this product should be simpler than what is planned today, but based on the same concept. Create a common interface into the basic network functions. Make Windows truly "network ready". We want customers to want Windows if they run a network. That is, extend the concept of Winnet today. Customers will absolutely love the idea. In addition, it plays on our strength (Windows and the client); it insulates the networking companies from the user (so the user only sees Microsoft); it can be sold into all existing sites, *etc.*

  Marketing this product will be very easy. Even the "Windows" name can be leveraged as in "Microsoft provided me a (transparent) window into my computing environment."

  I believe that Windows/N offers advantages for non networked PCs. There will be lots of interesting stuff that can be done from home machines using integrated Fax, *etc.* For this reason, I suggest that we add the basic features to Windows. Then we have three basic packages (with one UI): basic Windows with "providers" disabled; "providers" enabled together with some sexy providers for fax, mci mail or the like; and finally an upgrade package for the installed base.

  I believe that this product should be shipped ASAP. The way to do that is not get carried away with remote management and all other wonderful things that are missing from networks today.

- Windows/N Server. I would propose that a business plan be written for a direct ELS I/II and Lantastic competitor with the most important advantage being ease of use. This is slightly different from what Nathan proposed recently. That is discussed below. This product would offer file, print, and admin capability for workgroup networks in a very simple package. The server side would be built, of course, under Windows. Client support should be for DOS and Windows. (No Macs, no fancy directory, *etc.*) The reason why this could be competitive is that ELS is hard to use -- Lantastic has won recently because it is easier to use. We should be able to make a *very* easy to install and use network in a Windows framework.

  Today, many PC manufacturers sell Windows loaded onto the PCs. We should be able to do the same thing with the board manufacturers. That is, if this is cheap enough, then we could perhaps take a royalty on every board (or every 5 pack or the like). Longer term, all the important PCs will be network ready with Ethernet or TRN available on the motherboard. Now would be the time to make a deal with them before they do it with Novell.

  Alternatively, we could bundle this functionality (one free provider) with Windows/N. I happen to like this idea. That would kill the workgroup systems of today.

PX0524

- OS N. We need to pick an OS to weave the networking code into the system. We can solve the ISV problems by creating Win32. We haven't solved it for the IHVs and the networking world. Clearly, the more operating system driver formats, conventions, *etc.* the harder it is to get momentum in the marketplace.

  Should the base OS be NT extended to support a rich set of protocols with a common interface to the application world: an inherently rich distributed file/peripheral system, directory, security, and basic admin to manage users and services? Or should we use Windows 3.2? That may sound silly, but it is worth considering. There will be Windows everywhere soon. It is much easier to convince someone to buy some distributed services (see below) to run on an operating system they understand than tell them they must add yet another operating system (*e.g.*, NT) to their organization. Would the OS/2 history repeat itself? (Even if we called it Windows+ or something, I don't think it would be hidden for long.)

  I propose packaging services on top of this environment. We need to market this as THE distributed computing platform. *Viz.*, write your application here. It cannot be perceived as a direct threat to a customer's installed base of network (*e.g.*, Netware). It is however a Trojan horse.

  Microsoft services and 3rd party services would simply fit under the basic service architecture provided in the system. This is not LM as we know it today. From this platform, LM could be built by adding the file, print, Mac, *etc.* support. Basically, I'm saying that distributed computing must have a good base. This must be integrated with the system from the ground up.

- Distributed Services. Outside of the operating system sale itself, this is where the money is. Services range from the mundane such as file/print to powerful services such as a global messaging engine for mail, inter-application communication, *etc.* There are two fundamental types of services:

  - user oriented. messaging, database, calendaring, computation, *etc.*

  - system administration and network management based. I personally see Nathan's Window Server as just a basic network service -- one for managing Windows clients. Other examples include remote PC debugging, network management, software licensing, software distribution, archiving, *etc.*

3. Some Other Thoughts

3.1 P&L and Church and State

There is an important separation between the Applications and Systems Divisions for marketing reasons. Even though not explicitly communicated, it is clear to me there are similar separations between the groups within Systems. I have run many P&Ls and know the benefit of pushing a manager to run their business to maximize their profits. However, the split today in Systems (at least for networking) may not encourage the right decisions to be made for the corporation in the long run. Much of this comes from my view that networking needs to be integrated into the operating system.

Some of the questions to ponder include:

- How should Netware be viewed by the operating system group vs. LM? I've heard many discussions over the DOS 5.0 upgrade shipping with the Netware redirector included on the disks. Being open is goodness; however, helping them put stuff into the OS? Our client? I understand the ease of installation issues. Nevertheless, I question this strategy.

PX0524

- I've heard a proposal that we host Portable Netware on NT. If a third party does this development, then fine. If we take any active role, then we ultimately will be drawn into a Netware on Unix vs. Netware on NT debate. This is wasted effort. We need applications -- not file/print systems. Furthermore, such a path would lead to a performance war game -- one which I can assure you that if someone using Unix wants to beat us, they will be able to. What would be the advantage?

  The real issue is the potential confusing message that it sends to the marketplace. Consider what would happen if NBU used Unix as a base operating system? Unix is far better than anything MS has today. However, this is clearly not in sync with our global mission. I believe the same holds for any *active* involvement with porting Portable Netware onto NT.

3.2 What's in a name?

We should consider whether changing the name of the NBU could help reposition us into another space. For example, would the Distributed Computing Business Unit (or Group) be better?. The reasons are as follows:

(a) DCE is attracting great attention. We could ride the wave by associating with this name.

(b) Distributed Computing sounds more encompassing and not associated with a physical network.

(c) Distributed Computing can be positioned to be the next logically step from client/server technology.

(d) It will be another marketing differentiation from Novell's story. (Recall they claim to be the "network computing" leader.) With IBM, DEC, *etc.* adopting DCE, our association with Distributed Computing would isolate Novell.

If asked to describe the difference I would say that DC encompasses all the technology/products to provide a distributed computing platform for applications whereas NC is a basic networking view stemming from a basic file server model. (This isn't what Novell means, of course, but it would probably work marketing wise.)

3.3 Strategic Product Planning Group

One of my biggest revelations is how many areas within Microsoft overlap. I've concluded a global coordination function would be beneficial to keep architectural consistency (and reduce redundant work) among the dispersed groups. Because of the far reaching effects of certain decisions and the complexity of the problems, I think we need a formal review group to monitor the implementation of the company's strategy. As an example, consider management of objects. What is the right split between the operating system, "database", network support, and an application? My guess is that between the OS group, database group, office group, and networking there are a lot of different opinions. This confusion will ultimately result in a inconsistent model or simply wasted effort.

I suggest a small team of the all around best thinkers be created to help plan and monitor the global issues of our future products.

PX0524

MICROSOFT CONFIDENTIAL

4. Summary

4.1 Questions to answer:

This memo discussed several fundamental issues which must be answered to finalize a coherent strategy. (Please think through your positions on the following questions. I will discuss some of my personal conclusions at the meeting on 1/19/91.)

- What is Windows? ... GUI or OS?
- What is the OS family?
- What is the market and distribution plan for NT?
- What is the plan for OS/2? (public position and internal)
- How badly should we try to create a combined IBM/MS networking story given the apparent problem in OS strategies? Are we prepared for the consequences of a divorce?
- Do we need to win in the organizational network marketplace (as I defined earlier) given the costs?
- Should we implement a workgroup (and small department) network strategy also?
- Should Windows be shipped network ready? DOS?
- Which OS base is the "right" one to embed richer networking concepts to create the distributed computing platform discussed earlier?
- How important is interoperability with DCE? How important is "convergence"?
- Strategically, is Novell an enemy to Microsoft or only NBU? ...to Systems or only NBU? How can we ensure being open while at the same time integrating our own products better?

4.2 Conclusion

The "out Novell Novell" plan attacks Novell at their strength: selling file server software. Remember, the LAN Shootout between 3COM and Novell. Novell ate 3COM for lunch. Why? 3COM attacked Novell at their strongest point: speed. Microsoft is having another kind of shootout with Novell.

The way you win in such a battle is to change the game. If Novell is playing baseball, then we should convince the world that football is the game to play. Of course, we better be really good at football. (Segmentation and differentiation are of course the marketing terms.) The best argument provided by Microsoft to date has been that we created client/server architectures. This put Novell on the defensive. This worked because there was SQL Server to back up the claim. It also attacked Novell at a weak spot: their weak application engine (VAP/NLM) strategy.

Our future direction should crystallize the distributed applications platform by integrating networking into the operating system. That way, we change the game totally -- from the NOS world to an OS that is already network

PX0524

MICROSOFT CONFIDENTIAL

ready. Therefore, why use Novell? Moreover, we must find a way to leverage our natural strength: the millions of desktops running DOS and Windows. We must deal from strength – not weakness.

PX 529

EXHIBIT
BELFIORE - 14
1/13/09 - 9A

PX0529

From: Tom Reeve
To: Michael Holm
Subject: FW: Chicago ISV doc
Date: Wednesday, September 07, 1994 11:00AM

From: Lisa Maki
To: Consumer Division Program Managers
Subject: Chicago ISV doc
Date: Friday, September 02, 1994 8:46AM

I'm assuming that most of you already have seen this, and I may even have already sent it but, well, senility is my best excuse...

Anyway. Here is a doc that Joe Belfiore, the Chicago Shell PM, put together to provide ISVs with some basic information to ensure that their future apps run well under Chicago. Some information in this document may be old, but it gives a good set of guidelines that we should all understand. The original can be found on \\pyrex\user\joeb.

<<File Attachment: CHICOISV.DOC>>

CONFIDENTIAL
MSC 00656839

MX3217128
CONFIDENTIAL

PX0529

How to be a great app in the Chicago Shell

JoeB, revised 2/11/94

The list below describes important points to make your app a great app in the Chicago shell. This list is constantly changing, so check back occasionally. In adition to this list, be sure to study the Chicago UI Style Guide. This can be found on \\chicago\public, and is extremely valuable in describing the way your app should *look* in chicago.

1. Support long file names, and display them correctly. If you use commdlg open/save as, then you'll have a partial long filename solution. (Commdlg will hand you short names-- you'd still want to find out the correct long name to display in your title bar.) Displaying file names correctly means stripping the .3 extension off the end of the longname, so users see names like "My letter to Mom" instead of "My letter to Mom.DOC". We will provide helper routines for doing this.

2. Support UNC pathnames for files in your application. This enables users to browse the World directly and open your files without being forced to explicitly make a network connection. Be sure you support multi-user file sharing scenarios well.

3. Make full & correct use of the registration database. This means:
   - register icons for your document types in the database so they are displayed correctly in the shell
   - add data-specific commands to the database for your document types (eg. "Play" for a sound). These will then be displayed in context menus throughout the shell
   - support "print-to" in the reg database to enable drag/drop printing to specific printers in the shell
   - use the App extension path as defined in the Setup guidelines. You should specify the path to your "Application Extensions" directory, somewhere under 'c:\windows\application extensions', and keep all the files that end users won't find interesting in that directory (*.DLL, *.HLP, etc.)

4. Handle data transfers in a consistent manner with the shell. This means:
   - Support drag & drop extensively, including win31 style drag & drop of files and OLE 2.0 drag & drop to enable users to move your data to and from the desktop, folders and other applications. Be sure to support dragging with the right mouse button with a menu at the end, like the shell.
   - Be sure the menu-based transfer model in your app will work well with the one in the shell. You should test scenarios like copying a link or file in the shell, and then pasting it into your app.

5. Use commdlg, ESPECIALLY FILE OPEN/SAVE AS. If you use our file open and save as dialogs, for example, you will get links, long filenames (sort of) & direct browsing of the network for free. If you cannot use the commdlg open/save as dialogs, be sure your open/save as dialogs support the following features for consistency with the shell & apps/applets:
   - namespace hierarchy that's the same as the shell. "Desktop" is the root, followed by everything on the desktop: "My Computer", "My Network", etc....
   - support for links (opening a link should open its target)
   - display files with their correct icons and with their .3 extensions stripped, as in the shell
   - browse the network hierarchy directly
   - other miscellaneous features (context menus, properties sheet with ISV extensions, details view...)

6. Be sure your application visuals and control interactions are consistent with the shell and applets. See the UI Style Guide for Chicago (\\chicago\public) for more details.
   - make sure your dialog boxes use non-bold fonts and 3d effects like the rest of the system
   - use control panel button face/highlight colors on your 3-D buttons so they will reflect different 3D shades than just grey
   - take advantage of new system-supplied controls like Toolbar, Status Bar, Spin Buttons, etc.

PX0529

7. Maintain a consistent object paradigm between your app and the shell. This means:
   - use document names as the primary title bar text for your app window. (Document-centric!)
   - use context menus on mouse button 2 on controls and content like the shell does
   - follow the new OLE 2.0 UI guidelines in the Style Guide! You should rev your OLE UI to include these items:
     - support the new "object properties" dialog for your embeddings
     - make sure your icon visualization for embeddings is consistent with the shell. When a user drags an icon from the shell into your container, the icon & name should be the same.
     - make sure you support interactions with icon embeddings the same way the shell does– this includes selection appearance and fixed size
   - replace complicated dialogs in your app with property sheets where appropriate (we will supply a control)

8. Support Chicago-specific "Help" features. This means:
   provide context sensitive help bubbles in your dialogs and for your documents
   - use winhelp secondary windows for your procedural help, and take advantage of shortcut buttons and resizable topic windows to make help shorter and easier to read
   - investigate using Chicago's built-in Cue Card support

9. Be careful about multiple instances of your app being too easily started at the same time. This is a big problem for new users, and this is important to handle correctly with Chicago's data-centric paradigm. (See section below on the exact guidelines.)

10. Follow Chicago's guidelines for setting up your app properly in a chicago environment. More detail is available in another document, but here's a few points:
    - register your app properly as in point #3 above
    - try to create at most ONE folder underneath the c: root to place your app executable and interesting sample files. Place all the unfriendly file types (.DLL, .HLP, etc.) in the 'c:\windows\app extensions' directory
    - place a SINGLE icon (link) for your main application directly in the '\programs\programs menu items' folder, so that your app can be easily launched from the Start menu. DON'T overpopulate this menu! (you might want to prompt the use to choose which icons should be added during your setup)
    - support our "auto-setup" feature as described in the setup guidelines so that your installation program is run automatically by the shell (on some hardware, this will happen as soon as the floppy or CD is inserted.)

10. Extend the shell's ability to provide general information about your files. You can do this by:
    - using docfiles as your datatype and keeping general properties like author, word count, thumbnail, etc. in the OLE 2.0 docfile summary stream
    - you might consider adding pages to our file property sheet for files your app is interested in. We have a registration mechanism for people to indicate interest in adding pages to file properties

11. Support pen input for pen notebooks and desktop tablets. This means:
    - using Windows pen apis to activate the pen in your app, so that users can edit documents using gestures and input text using handwriting recognition
    - using ink-edit controls to allow users to enter scribbled notes, drawings, and signatures
    - enabling "ink" annotation of documents using the OLE 2.0 annotation server
    - adding other natural pen-oriented features and gestures to your app

12. Consider a major re-design of your user interface to take advantage of the new emphasis on data-centricity and shell integration. This may mean:

PX0529

7. Maintain a consistent object paradigm between your app and the shell. This means:
   - use document names as the primary title bar text for your app window. (Document-centric!)
   - use context menus on mouse button 2 on controls and content like the shell does
   - follow the new OLE 2.0 UI guidelines in the Style Guide! You should rev your OLE UI to include these items:
     - support the new "object properties" dialog for your embeddings
     - make sure your icon visualization for embeddings is consistent with the shell. When a user drags an icon from the shell into your container, the icon & name should be the same.
     - make sure you support interactions with icon embeddings the same way the shell does-- this includes selection appearance and fixed size
   - replace complicated dialogs in your app with property sheets where appropriate (we will supply a control)

8. Support Chicago-specific "Help" features. This means:
   provide context sensitive help bubbles in your dialogs and for your documents
   - use winhelp secondary windows for your procedural help, and take advantage of shortcut buttons and resizable topic windows to make help shorter and easier to read
   - investigate using Chicago's built-in Cue Card support

9. Be careful about multiple instances of your app being too easily started at the same time. This is a big problem for new users, and this is important to handle correctly with Chicago's data-centric paradigm. (See section below on the exact guidelines.)

10. Follow Chicago's guidelines for setting up your app properly in a chicago environment. More detail is available in another document, but here's a few points:
    - register your app properly as in point #3 above
    - try to create at most ONE folder underneath the c: root to place your app executable and interesting sample files. Place all the unfriendly file types (.DLL, .HLP, etc.) in the 'c:\windows\app extensions' directory
    - place a SINGLE icon (link) for your main application directly in the '\programs\programs menu items' folder, so that your app can be easily launched from the Start menu. DON'T overpopulate this menu! (you might want to prompt the use to choose which icons should be added during your setup)
    - support our "auto-setup" feature as described in the setup guidelines so that your installation program is run automatically by the shell (on some hardware, this will happen as soon as the floppy or CD is inserted.)

10. Extend the shell's ability to provide general information about your files. You can do this by:
    - using docfiles as your datatype and keeping general properties like author, word count, thumbnail, etc. in the OLE 2.0 docfile summary stream
    - you might consider adding pages to our file property sheet for files your app is interested in. We have a registration mechanism for people to indicate interest in adding pages to file properties

11. Support pen input for pen notebooks and desktop tablets. This means:
    - using Windows pen apis to activate the pen in your app, so that users can edit documents using gestures and input text using handwriting recognition
    - using ink-edit controls to allow users to enter scribbled notes, drawings, and signatures
    - enabling "ink" annotation of documents using the OLE 2.0 annotation server
    - adding other natural pen-oriented features and gestures to your app

12. Consider a major re-design of your user interface to take advantage of the new emphasis on data-centricity and shell integration. This may mean:

- if your app focuses on editting documents, you might think about making your app SDI so that users only think about the documents rather than the application as a workspace
- if you have a hierarchical containment namespace that contains specific non-ordered objects, think about integrating into the explorer as a "special folder"

Preliminary guidelines on #9 above... how to handle multiple instances of your app/documents. CHECK STYLE GUIDE FOR MORE COMPLETE/ACCURATE GUIDELINE.

1) DOUBLE-CLICK DOCUMENT (or other Open/Save kind of object):

Check to see if the document is already opened. If it is, the following is the preferred behavior:

- app checks to see if open copy has been opened by the current user (note: this check *should* be better than just comparing usernames, since the user could be logged onto >1 machine). If NO (someone else on the net has it), then:
  - the user is prompted with "This document has already been opened by <name>. Would you like to make a copy?". If the user does not want a copy, the app goes away, else it makes a copy.
- If the app figures out that the current user already has a copy open, then it just immediately restores the open version of the app and the correct MDI child window, if necessary.

2) DOUBLE-CLICK THE APP ICON after the first time. This may vary from app to app, but for the class of applications for whom double-clicking the app icon opens a blank document, behavior should be:

- present the user with a list of currently opened documents, with "New Document" as the default choice.

New Controls & Common Dialogs you can use

New Controls:
Toolbar
Status Bar
Column Heading
Slider
Scrollable button bar
Progress Indicator (Gas Gauge)
Tabs
ListView
TreeView
Property Sheet
Rich Text Control.

Common Dialogs:
Open
Save As
Print
Print Setup (Choose Printer)
Page Setup
Find
Replace
Color Picker

PX 530

PX0530

EXHIBIT
BELFIORE-18
1/13/09 - JA

Bill Gates

From: Bill Gates
To: Brian Fleming
Subject: FW: Shell Extensions
Date: Tuesday, November 08, 1994 6:41AM

Basicly we gave up because of marvel and paul will check into the add on pack..

From: Paul Maritz
To: Bill Gates
Subject: RE: Shell Extensions
Date: Monday, November 07, 1994 5:56PM

I met on Friday with DRG and Joeb to review following:

1. In M7, the Chicago Shell has been changed to force apps that use the iShellFolder interfaces to open into a separate window, ie. appear to be separate apps.

2. There were 4 groups using these interfaces (Capone, Marvel, Stac, Symantec). Capone, Stac, Symantec have found ways to not use them. However, the MARVEL guys have said that there is no way they can move off the current interfaces and still have chance of shipping with Win'95.

3. Based on this stance by MARVEL, we will not disable the interfaces, but will not document the iShellFolder in regular documentation, but we will have them documented in a resource kit so that if someone really, really does want to use them they can. I feel that if Marvel is using them, we have to say that in theory someone else could have done likewise. However we will tell ISVs that there is no guarantee that these API's will not get broken if future and we do not recommend their use as a result. If someone does use them, because the interfaces have been doctored to force app to open into a separate window, the legacy case can be handled relatively straightforward way by simply starting a copy of current explorer and then starting the app in same process.

4. I am not up to speed on details of O'Hare - will find out.

From: Bill Gates
To: Paul Maritz
Subject: FW: Shell Extensions
Date: Tuesday, November 08, 1994 5:32AM

I am a little confused by what is going on in this whole area.

HIGHLY
CONFIDENTIAL

MX 6025435
CONFIDENTIAL

MSC 00700466

PX 531

EXHIBIT
Ludwig - 4
1/21/09 - JA

PX0531

Novell

MICROSOFT MEMO -- DRAFT

Date: 10/13/93

To: Paul Maritz, Jim Allchin, Brad Silverberg

From: John Ludwig

Subject: Novell's AppWare

Novell's AppWare has gotten a tremendous amount of press coverage recently, and analysts are increasingly enamored of it. This memo outlines the goals of AppWare, the AppWare architecture, and Novell's major claims for AppWare. Agains each of these I outline possible MS responses. Attached for reference are the AppWare materials provided by Novell at the October Brainshare event.

* * *

AppWare Goals

Novell's Brainshare presentations define the Appware goals as:

"To stimulate growth of network applications by

- *Hiding complexity of the network*
- *Increasing development efficiency by standardizing access to services"*

An oft-repeated goal in the materials is to provide a cross-platform (client platform that is) solution, reflecting the increasing heterogeneity of customer networks.

⇒ MS response:

- These are fine goals, we support them.
- WOSA is our framework for standardizing access to network services. WOSA provides single, consistent API access to multiple backend services on Unix, Netware, NT, and hosts.
- We've been delivering on WOSA for several years -- MAPI, ODBC, Windows Sockets are all shipping. WOSA is not vaporware.
- We're layering our OLE2 object model on top of WOSA to make service access even easier, we've been at this for several years and have broad industry support for the OLE model.
- We are making OLE2 and the Windows API available cross-platform to enable cross-platform app deployment while preserving investment in existing Windows applications and tools. AppWare requires rewriting of all this code.

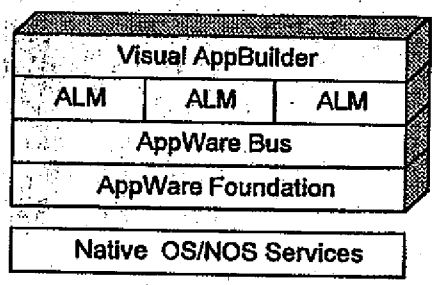
MS 0115589
CONFIDENTIAL

PX0531

- There is a huge array of tools available already for the Windows API and OLE development, targetting all levels of developers. There is 1 tool announced for AppWare.

AppWare Architecture

AppWare consists of 4 major components: the AppWare Foundation, the AppWare Bus, ALMs or AppWare Loadable Modules, and the Visual AppBuilder.



The **AppWare Foundation** provides a "common, cross-platform set of APIs...(which) allows developers to maintain a single-source base for all development platforms." Basically, this layer virtualizes all services of the underlying OSes on which it is hosted, insulating the developer from differences in these platforms. The services that are provided by this layer include:

Foundation Series	Connectivity Series	User Interface Series
Character Handling	Event Management	App instance support
Data conversion and storage	Pipe and Socket management	Box instance support
Error Handling	Object linking	Button instance support
File IO and file system management	Clipboard	Dialog control
Font Control	Task Manipulation	Display instance support
Graphics Package		Edit Text instance support
Instance Management		Help instance support
Keyboard Management		Interface data management
Memory Management		List instance support
Message Passing		Menu instance support
Module Management		Slider instance support
Pointer/cursor manipulation		Common Dialog support
User preference management		Table instance support
Printing support		Void instance support
Resource Management		Window instance support
OS attributes		

MS 0115590
CONFIDENTIAL

The upper interfaces of the AppWare Foundation are intended to be used by developers developing in C and C++, including the developers of the AppWare Bus from Novell. The Foundation interfaces will be documented in an SDK to be released shortly.

⇒ MS Response:

- The AppWare Foundation is an entirely new OS API. It offers virtually all the services of the OSes it is hosted upon, but with a brand new and different API set. All code has to be rewritten to use it, it provides no migration path from existing code bases.
- The AppWare Foundation is no simpler than any other OS API. The learning curve for it will be just as steep as that for Windows or the Mac or any other API set. Developers are already down these curves.
- Mixed mode development is not possible -- you can't easily write an app that is 1/2 native Windows, 1/2 AppWare, as the systems will not coordinate access to the screen or other resources. You have to commit to the AppWare environment fully.
- The Foundation, as a layer on the host OS, will always be slower and fatter than native code. You can't add a layer without adding size and losing speed.
- The Foundation is incomplete. It does not represent all the services of the host OSes. A single small development team cannot maintain pace with the thousands of developers at MS, Apple, Taligent, IBM, Sun, who are constantly innovating on these platforms.
- The Foundation is behind. By the time it ships, the next revisions of Windows and other OSes will be on the market, requiring a new version of the Foundation.
- The Foundation lacks some critical services, such as threads. This makes the Foundation basically incapable of taking advantage of the multiprocessing capabilities of the underlying OSes.

The layer above the AppWare Foundation is the **AppWare Bus**. The AppWare Bus provides "standardized access to prebuilt application services". The exact services provided by the AppWare Bus are not well documented, but conceptually the Bus seems to be the object model and object layer. The Bus handles bindings and interactions between objects. Per Novell, the object model will be compatible with OLE2.

⇒ MS Response

- The AppWare Bus is an entirely new object model. It is not OLE, it is not CORBA, it is not OPENDOC, it is a new thing. Everything has to be rewritten to it, there is no migration path from existing code bases.

- This new object model will sit side-by-side with the existing native object models of the various platforms -- more code, slower and fatter systems.
- Interoperability with existing native object models, and the hundreds of applications written to them, will never be perfect as the native object models evolve and the AppWare Bus must constantly be revised to keep up.

Residing on the AppWare Bus are **ALMs**, or **AppWare Loadable Modules**. These are objects, provided by Novell and others. The objects that Novell says they will provide (or will be provided by 3rd parties) include:

- File, Print
- Directory Services
- Messaging
- Database
- OLE/DDE
- Software Dist/Licensing
- Telephony
- Image/Document Management
- Network Management
- Multimedia
- Comm
- Security

⇒ MS Response:

- Again, these ALMs have to be completely rewritten from current code. There is no migration path from current drivers/libraries providing these features.
- The list of ALMs to be provided is vaporware. Windows and other platforms have these services now.
- The ALMs presented at Brainshare are not multi-vendor, heterogeneous ALMs. The directory ALM supports NDS. The Telephony ALM supports Netware Telephony services. The File/Print ALMs support Netware. Novell has promised to support other nets only as "market demand warrants". Other nets are supported today by WOSA.
- The ALM SDK is not shipping until the end of year at best. OLE SDKs are available now.

Finally, residing on top of the AppWare framework is **Visual AppBuilder**, a visual programming tool targetted at corporate developers. Constructing an application in Visual AppBuilder is a very graphical process -- lots of dragging and dropping of objects, dragging to establish relationships/actions between

objects, etc. There is little/no direct editting of code involved in creating an application.

Visual Appbuilder is not the only tool that Novell hopes customers use to develop AppWare apps. The Foundation, Bus, and ALM APIs are all published and Novell is encouraging ISVs to provide a variety of tools targetted at different levels of development expertise.

⇒ MS Response:

- Visual AppBuilder is a fine tool. It is too bad it can't be used with the native OSes on which it is hosted.
- Visual AppBuilder is one tool, targetted at one class of developer. There are hundreds of tools available on Windows and other platforms targetted at all levels of developers.
- Building substantial applications in Visual AppBuilder will be quite challenging, as it lacks a traditional IDE coding environment. All "code" must be entered as drag/drop actions and mouse clicks, there is no heavy duty coding environment.

AppWare Claims

Novell claims that the sum of the AppWare elements leads to greater developer productivity due to several characteristics of the environment.

- Application Performance. Apps based on the AppWare Foundation provide the same level of performance as applications based on native implementations.

  ⇒ MS Response: This is nonsense. The foundation is layered on the native OS services and is no simpler by Novell's admission; the layer adds code which means greater size and lower performance.

- Toolkit functionality. The AppWare Foundation provides developers all the functionality they need – GUI, OS services, inter-app connectivity.

  ⇒ MS Response: Again this is nonsense. Critical services like threads are missing, and are unlikely to be supported on platforms such as the Mac. The AppWare team due to its small size and separation from OS vendor teams will be consistently behind the capabilities of the native OSes. AppWare is not a superset, it is a least common denominator approach.

- Toolkit modularity. The AppWare Foundation architecture can be scaled, based upon the nedds of each application.

  ⇒ MS Response: This is true of each native OS, you can just use the capabilities you need. And since you cannot effectively do mixed-mode development, you can't use just part of AppWare – you need to make a wholesale conversion with the attendant learning curve.

PX0531

- Toolkit extensibility. The AppWare Foundation tookit enables developers to extend functionality by implementing featuers not directly provided by the toolkit.

  ⇒ MS Response: But you told us 2 points ago that the AppWare Foundation was complete and never needed to be extended. Which is it? And if I do extend it, the primary benefit claimed of portability is gone as I've now added platform-specific features.

- Development migration path. The AppWare Foundation architecture provides a method of mixing new and legacy code, since most organizations cannot afford to migrate an entire app to a new platform all at once.

  ⇒ MS Response: There is really no mixing supported. You can't draw on the screen using AppWare Foundation calls and native Windows calls and achieve any sensible and manageable results. Converting to AppWare is converting to a brand new OS API, and you need to do it completely.

- Integration with other tools. The architecture allows developers to work with their favorite 3rd party tools and easily integrate their own tools.

  ⇒ MS Response: So where are the tools? There are hundreds of compilers, IDEs, interpreters, debuggers, code management tools, etc. available for Windows and other OSes. So far, AppWare has one committed tool -- Visual AppBuilder -- and one promise from Borland to support it in the future.

- Portability. Applications developed to the AppWare Foundation or to ALMs/AppWare Bus will be portable to any of the AppWare Foundation-supported OSes.

  ⇒ MS Response: Yes, if you use only the services of the AppWare Foundation, and the provided ALMs which work only with Netware servers, then your app will be portable to other platforms supported by the AppWare Foundation.

  But most developers will want applications to be portable across server backends -- different SQL databases, different mail backends. This will require services outside of the AppWare environment. And with the increasing popularity of Windows API support across all the popular client OS platforms (OS/2, Taligent, Unix, etc), writing to Windows is a more compatible, lower learning curve approach to cross-platform development.

MS 0115594
CONFIDENTIAL