

PX 87

**Erik Stevenson**

**From:** Jim Allchin  
**To:** bradsi; paulma  
**Subject:** FW: shell issues  
**Date:** Friday, October 01, 1993 7:28PM

I just read this mail. I phoned both of you. You were both gone.

You both knew this was a critical issue to me. I can't believe that neither one of you discussed this with me personally. Paul if you say you came by and I was in meetings, then I'm sorry but I still don't accept this. You didn't even send me mail. Nothing this big has to be decided in a few minutes like this which impacts our work and more importantly positioning. You both knew this was a hot issue. Brad even though your comments are constructive and I appreciate them, I find this whole thing very upsetting. You didn't even try to contact me to discuss this.

This whole scenario is exactly why we do not a good working relationship.

There is no reason that I currently understand that everyone cannot use the Cairo interfaces (other than Capone if that ships bundled with Chicago) that I know of. Cairo will ship long before Office -- that is very clear. I don't understand any of this logic. Inside Microsoft people would be using the interfaces essentially today on our IDW. The SDK will be in people's hands before Chicago ships. (If fact, we even talked about this code being on the CD for the PDC this evening. We were only going to cover the interfaces, but having code there might make sense.)

Such a plan delays Cairo (but I'm not sure if it's a negligible amount or a large amount). It certainly makes Cairo bigger which is exactly what I'm killing myself to fix. It gives competitors an advantage over Microsoft Apps because Ray and everyone else will use the extensions before our apps group (for sure Office).

pretty damn discouraging...

jim

-----  
**From:** Brad Silverberg  
**To:** David Cole; Dennis Adler; Joe Belfiore; Kurt Eckhardt  
**Cc:** Steve Madigan  
**Subject:** shell issues  
**Date:** Friday, October 01, 1993 2:20PM

I talked to Paulma today about some shell issues, especially at the global level, such as apps plans and cairo.

Paul said that the apps group wants to get out of their "end of the year" cycles they are on now and into a mid year cycle. [One reason this is so important is to put them in a much better position for the Comdex timeframe awards.] This means that they will do Chicago revs of the apps with Chicago, or at worst, 60-90 later. There's not that much they can get done, so will be limited to things like 32-bit, Chicago visuals, better OLE, some multithreading, or similar stuff. These apps will require Chicago.

They will not be in a position, as a result, to have their "Chicairo" apps

until very late '95 or '96, and certainly would not require the Chicairo shell at that point.

This means that they plan to write a bunch of shell extensions to the Chicago v1 shell. Capone, for starters. In addition, Ren is going to be transferred over to work for Chris Peters (who is going to own Office), and be done as a Chicago shell extension. Other things that Office wants from the shell will be done as Chicago shell extensions. So will the Online Services project under RussS.

This means, of course, that Cairo is going to have to run these Chicago shell extensions.

And this means, that we really have to work well with the Cairo guys to develop the extensions so they can support them. Both groups -- Chicago and Cairo -- need to look at this in a non-religious way. Cairo knows they don't have a choice but to support Chicago extensions now, and further, some of the special things they wanted to do that invalidated a more common approach, they may have to give up or do differently. The key is to treat this, by both sides, as an \*engineering\* issue, not a religious or political one.

The ideal is to have an extension mechanism for applications that is based on ole2 or something that is simply wrapped to ole2. Maybe a better version of the Ren approach makes sense. What would be nice is an approach like we are pursuing with monikers/links, where the interface is ole2 but underneath is a different, small/fast implementation.

In addition, it's ok if the mechanism used by apps (Capone, Online services, Ren) is different than the one used internally by the system for say control panel. The latter can be very optimized, and it's ok if, when running an app that uses the former method doesn't run in 4M. That is, the shell itself needs to stil run in 4M. But when running an app that extends the shell, it's ok to run in 8M.

Paul has discussed this in detail with Bobmu and he fully supports. Bob accepts now that it has to be this way, that Cairo will have to support the chicago extensibility approach, and that they may have to give up some of their pet ways of doing things.

So please, we need to work together with them on this as an \*engineering\* problem, not a religious one, and they agree to do the same.

Do I have your support in this? Thanks.

PX 88

**Erik Stevenson**

---

**From:** Brad Silverberg  
**To:** Bill Neukom (bilin); Debra Vogt (debrav)  
**Subject:** FW: WfW, DDJ  
**Date:** Monday, October 04, 1993 9:15AM

**From:** Andrew Schulman  
**To:** Brad Silverberg  
**Subject:** RE: WfW, DDJ  
**Date:** Sunday, October 03, 1993 10:40AM

Here's your message from the other day. Thanks too for the message re: NeXT. I think you should add Mitch Kapor's "ON" to the list of serious competitors too. :-)

----- Forwarded Message -----

**FROM:** INTERNET:bradsi@microsoft.com, INTERNET:bradsi@microsoft.com  
**TO:** Andrew Schulman, 76320,302  
**DATE:** 01-Oct-93 at 00:38:31

**SUBJECT:** RE: WfW, DDJ

**Sender:** bradsi@microsoft.com  
**Received:** from netmail.microsoft.com by dub-img-1.compuserve.com (5.67/5.930129sam)  
 id AA13996; Fri, 1 Oct 93 00:37:52 -0400  
**Received:** by netmail.microsoft.com (5.65/25-eef)  
 id AA09074; Thu, 30 Sep 93 21:36:55 -0700  
**From:** bradsi@microsoft.com  
**Message-Id:** <9310010436.AA09074@netmail.microsoft.com>  
**To:** 76320.302@CompuServe.COM  
**Subject:** RE: WfW, DDJ  
**Date:** Thu Sep 30 21:36:29 1993

I didn't mean to convey that Taligent, NeXT, Go were my only competitors. Hardly. I also have IBM, Novell, Apple, Sun, HP (Unix), Novell Unix, SCO, ... These are very determined companies that each have strengths, both technically and marketing wise. [Not the least with IBM the will to lose hundreds of millions of dollars in the process.] I have to keep improving Windows as quickly as I can to stave these guys. OS/2 has made strong inroads. Apple is 15% or more of the market. Novell is coming at us from every direction possible and has a very concerted, multilayer attack strategy, including using the political system. I would include other companies who are intent on building "middleware", which is just system software by another name. In fact, Jim Manzi just said today that by 1995 Windows will just be a graphical C: prompt; users will instead live in Notes.

I think many people -- yourself included -- vastly underestimate the competitive situation for operating systems. We have to invest our meager resources in building a better product. And it's not easy with a big installed base either. The other guys have the luxury of not being tied down with compatibility and can have the opportunity to move quicker. A major part of the win 3.1 and chicago efforts is dedicated just to staying compatible with existing dos and windows apps and drivers. Do you know how many people work in total on windows and ms-dos?

Including development, testing, marketing, documentation, it's just over 300. That's all! And that includes marketing.

So you can see, we have to work hard and smart to stay ahead. Each competitor can exploit a weakness, and they are getting better everyday. They've learned a lot from our "playbook" over the last two years and have become much tougher competitors. We naturally have to take bigger and bigger risks, push harder to avoid falling behind. Chicago and Cairo are ambitious projects but necessary to compete with ... IBM, Novell, Apple, Sun, Taligent, NeXT, Go, ...

PX 90

**From:** luist  
**Sent:** Thursday, October 07, 1993 11:14 PM  
**To:** bobkr; bradsi; denisg; johnlu; jonro; robp; scotra; thomasre  
**Subject:** Appware at Brain Share

This is a summary of Novell's Appware products and strategy, as presented at Brain Share Dallas early this week.

: summary

Appware is a Novell product strategy that includes, for now, a cross platform (Win, Mac, UNIX) development foundation (Appware Foundation), a visual Appbuilder, an object model (Appware Bus) and a distributed system services strategy (based on NW 4.x Directory Services). From the "Novell Distributed Appware Services" white paper, "Novell's strategy is to provide a set of enabling services to facilitate the t between the troops (Novell VPs from newly aquired pieces on down) and the company's (?)strategists (King, Edwards). IMHO Novell could be touting this as Cairo today. Given the market confusion of what Cairo is or it is not, they could easily claim they have all the pieces integrated and ready to go...

I have a lot more information in slides, notes and whitepapers if you are interested.

: announcements

: BORLAND: the next version of Borland's foundation library will be built on top of Appware Foundation, they will be sold as one product.

: Word Perfect: all future versions of Word Perfect will be built on top of foundation. The 60 WP engineers currently in charge of platform independence are now helping Novell with the founfation.

It is unclear wheather Appware Foundation, BUS and Visual Builder are ready or in their final beta cycle.

An uninterested party using the product confided that it is probably 6 months away.

However, Novell is already selling it, one time flat fee ( i will get a free copy, if anyone is interested):

Foundation: \$595/platform

Visual Appbuilder : \$495/plat

Appware Bus, ALM SDK: \$295/platform

: Appware Foundation, speaker Doug Donzelli, STI ex-owner VP Appware sys group (under Edwards)

Though his first slide is: "Goal: Stimulate growth of network applications by hiding complexity of net and standardizing access to services" (a scary angle coming from Novell and if you include o/s APIs as

services), the rest of the presentation was as if it came from STI. Doug seems very smart, good

n, Mac, UNIX and, maybe GeoWorks on DOS 7 later:

- great performance: all code is native to plat. most APIs are very thin layers

- modularity: pick and chose layers, and thickness, to include

- extensibility: coexists with native code. uses platform's compilers: MS,

Borland, Think C, MPW, etc.

, UNICODE,

BiDi text...

- claim transport independence, but have very little specific functionality to back up
- NO OLE support: part of Apple's Open Doc, "will support OLE in as much as it is subsumed by AOD"
- NO threads
- NO file abstractions
- NO RPC support - but will do a superset of DCE-RPC
- NO NW 4.0 DS support

: Visual Appbuilder and BUS

This is the Serious technology acquisition. The Visual Appbuilder looks like a cross between VB and Object Vision. It sits on top of the AW foundation and is a way to create application from ALMs. ALMs are lightweight objects managed by the Appware Bus("first commercially available sw engine for managing intelligent sw components"). They are similar to VB's installable controls, but seem much easier to implement. There is no compound document technology yet, but will have OLE support "soon".

It is being marketed by Novell as a vertical app builder for now, but with potential for full-scale apps in the future.

akes it invisible

For now, the above is bullshit, since they have no mechanisms in place to accomplish any distributed execution and synchronization of ALMs.

There are "hundreds" of ALMs available today, but the ones demoed by Novell are mostly in the category of early low end or specialized VB controls, unusable for any large-scale commercial development.

: Luis's editorial

Why is Appware dangerous? What sets it apart from other 4-5GIs and cross-platform foundation tools?

1.- quality: might be first viable platform for commercial cross-platform development.

=> could in the long run blur the o/s API line and squeeze us into the camp of BIOS builders.

2.- backing by Novell brings:

- credibility, ability to harness the Network (NW 4.x DS, RPC, Messaging, etc)

- can be positioned as a Notes alternative:

=> might be nice if it fractures the mkt.

3.- Backing by Borland:

- an immediate boost of credibility and installed base of serious developers

4.- A vehicle for Novell and its networking strategy to be on developer's faces and manipulate

direction...they move from being the pipeline into the limelight.

5.- It can be touted as a Notes competitor on the way to Cairo. But, unlike Cairo, it is real and available

today (in fact I believe they can claim this IS Cairo)

6.- Might be channel to unleash 4.x features.

7.- It is NOT, for now:

- a Novell revenue source: they've spent mega bucks and have a lot of work to do, it is costing them much more than \$595 a pop!

- ready: they have a way to go

- focused:

:why in HELL invest so much in UNIX. they should promise UNIX as "it will come soon" and forget it

: their distributed system and networking integration stories are weak. they need to teach STI and Serious that Novell rules the Nets and nothing else.

8.- someone in our languages/tools group should be interested in this stuff and be prepared to kill it, or at least understand why it is, or is not dangerous to our strategy.

PX 91

**Erik Stevenson**

---

**From:** luist  
**To:** bradsi; johnlu; robp  
**Subject:** call me paranoid...  
**Date:** Saturday, October 09, 1993 9:16AM

rob mentioned that a few of lotus' technical big shots will be in town soon to talk about chicago. after manzi's big public push to make notes the c-prompt for network users, it worries me that we open our kimono and show them what chicago will bring, its UI and net features. it seems to me that notes can be as dangerous middleware as a well thought out appware strategy...

i may be just paranoid after seeing red all over networkd...what think?

PX 94

PLAINTIFF'S  
EXHIBIT

1841

Comes v. Microsoft

Amy Johnstone

From: Paul Maritz  
To: Bill Gates  
Subject: strategy  
Date: Wednesday, October 13, 1993 4:59PM

I am sorry if I have not done good at communicating status to you (althought the thinking of using of providing the OLE2/Component Forms based Shell environment on Chicago by doing a second release of Chicago with OFS, was communicated via the email, attached at end of this message, which went to "all concerned" - see "to" list), but the issues and tradeoffs ARE complicated. I would appreciate being able to go thru this face to face - is there possibility of meeting Thurs/Fri/Sat?

I have been doing following work to get a handle on things:

- getting bradsi and jimali in sync (more below),
  - met 1-1 with Peteh (week before last),
  - communicating with other divisions (had meeting with key folks from DAD, WGA, DDT last Friday pm),
  - met on Monday with REN folk (following your mail).
- I have been trying to achieve following:

1. Recognizing that Chicago is THE next ISV target, then:

(i) Ensure that the API's exposed by Chicago are as close as we can make to the OLE direction we want to go - mainly can we get shell extension API's to be OLE (answer: yes, we can by using lighter weight OLE implementation for just those scenario's that Shell uses - ie. not for general in place editing, etc.). This will mean that any Chicago UI exploitive apps would work decently on Cairo - ie. no need for ISVs to do different work to run on Cairo.

(ii) Contain Cairo to meet following criteria:

- be proper superset of Chicago (all apps, including Shell extensions, must run)
- be available within six months of Chicago - this means that Cairo team has to have high quality beta that we can broadly give out when Chicago ships.

To do this, we have to figure out how much can/should be done in Win NT 3.11 - should we try to do more to sync up with Chicago API's, or do less and try to ensure prompt shipment of Cairo (defined to meet criteria above). I have discussed this with Jimall and he is currently working through the issues with his team, and deciding what is on critical path and can/has to be cut.

This does raise the issue of whether we should simply declare Cairo to be "sync up with Chicago" release, or whether we should imposed constraints above and let Jimall decide how much he can afford to ship. I am currently working through the latter path.

2. Understand role that REN plays. I was actually impressed by the meeting with Brainmac & Koss, I think it is great move to have them be part of office. However, it was somewhat frustrating to see the overlap in what they are doing and what the Cairo shell team is doing - both are producing a Capone replacement, both are producing a set of standard views for looking at heterogeneous objects, etc. In a perfect world, the following would be true:

- REN would be build using Component forms (can be done if REN will accept Q1'95 ship date - we took action to try to convince them that it would be OK to take this dependency)
  - REN currently wraps the current non-OLE Chicago extensions, with their own OLE based extensibility mechanism - this should be the same as that officially used in Chicago/Cairo (action taken to see if this can happen),
  - REN would use DAO wrapper for LMS, meaning it would work without change over OFS (need to get DAO closed for this).
  - REN would be "vend" some views to Cairo - why both do the job of providing standard views? This provoked fairly strong reaction from Brianmac that his charter is to "help Office, not Systems" - but I still think we should look harder at saving effort.
- I regret not having met before with the REN guys. This needs more thinking and understanding.

3. Meet with Peteh and co. to understand what Apps / Systems Plan interaction should be. I gave them almost identical feedback to the mail you sent out, namely that I think it is crucial that they get "reasonable" Chicago based releases out within 90 days of Chicago, with:

- 32bit
- reasonable Chicago UI exploitation (I told them that we guarantee that those apps run on NT)
- some way to claim that they exploit multi-threading

The hard issues then comes in when there next MAJOR release should be - should they do it in mid'95, or

Page 1



MS7048981  
CONFIDENTIAL

MS7048981

should they wait for the full Component Forms UI & OFS to be Chicago in late '95/early '96?  
This issue is complicated - I don't think I can have MAJOR ISV CONTENT releases much closer together than every 18 months, i.e. if Chicago is major ISV content release (Win32, UI, m'threading, MAPI, etc.) in Q3 '94 - then the next such event can't be much before late '95/early '96.

So, as best I can wrap my mind around things now, our systems/apps strategy could be:  
Q3 '95 - Chicago: MAJOR EVENT for Systems, we do as much as we can to make it a major event for Apps  
Q1 '95 - Cairo: sync NT up with Chicago  
Q2 '95 - REN & Component forms based Office release? Major feature of this would be REN.  
If we can sync up the issues listed above, this could also work well on Cairo as well as Chicago - it could even be billed as first OFS exploitive app when on Cairo, and be billed as providing OFS like function on Chicago (Apps leads way)  
Q4 '96/Q1 '96: MAJOR EVENT for BOTH SYSTEMS And APPS - OFS exploitive on volume platform, etc.

4. Get rational plan for EMS on Cairo. Given above constraints it means basically that we should not try to replace EMS server in first release of Chicago, but make sure that the Cairo UI environment which replaces Capone, can work well with a EMS backend (a necessity anyway since EMS store customers will not upgrade quickly) and handle EMS special message types correctly. EMS would be fully sync'd up with Cairo (OFS, DS, etc.) in late '95/early '96 (9 months after Cairo initial release). This raises issue of EMS not initially using OFS based directory.

Here is the memo on systems release priorities that I am working on. I have sent it to Brad and Jim all so far to get their reactions.

<<File Attachment: OBJS1093.DOC>>  
-----

From: Paul Maritz  
To: Bill Gates; Mike Maples; Pete Higgins; Brad Silverberg; Jim Allchin; Darryl Rubin; Roger Heinen; Tom Evslin; Bob Muglia; Christopher Graham; Chris Peters  
Date: Wednesday, September 29, 1993 3:21PM

From conversations with Billg and others, it is clear we are somewhat out of sync, on the interaction of systems plan and desktop apps plan - we need a meeting to update each which I will try to get set up for next week. But here is quick status on some things from "systems" end:

1. OLE-based Forms:

Jim all is prep'ing some mail to send on this, but the OLE-based Forms package can be made independent of release of Cairo, since all it really requires is OLE. Thus usage of this forms package is something we should be considering for products that ship in early '95.

2. OLE-2 Based Shell and environment on top of Chicago - codename "London":

The major issue here had been the way in which OFS storage services which the Cairo environment presumes, would be provided on Chicago. After analyzing the alternatives (doing nothing, improving FAT, using some form of LMS, porting OFS), it was determined that the best alternative would be to port OFS - since it offers maximum benefit, and could be done with a reasonable effort (<5 man-years to provide a complete environment - i.e. moving CairOLE to Chicago, moving DFS to Chicago, enhancing Chicago IFS to allow simple port of OFS), and working set would be reasonable (worst case would increase working set by <3 MB for CairOLE + DFS + OFS + Cairo Shell) and would allow a reasonable 8MB solution.

The issue with the above would be that it would require a "new" release of Chicago, which would mean that we would probably do other cleanup, enhancements at the same time. If we wait until Chicago ships before starting on this, it means a delay of 18 months or so after Chicago v1 ships. This means that we have to start the effort before Chicago ships. Fortunately it is fairly well confined/defined work. The Cairo shell pieces can and are being engineered to be well-behaved "Win32/OLE" applications which means that it should be a port. We will look to try start the Chicago scaffolding work (IFS hooks, etc) soon.

However, presuming we can come up with a reasonable staffing plan, this would mean that the systems plan would look something like:

Adding a quarter to everything for pessimism's sake - all internal dates are a quarter

earlier)

Q2'94 Win NT 1.0a (includes sufficient of Chicago Win32 API enhancements to allow ISV to target both Chicago and Win NT in one effort)

Q3'94 Chicago

Q2'95 Cairo

Q3'95 London

Note: the above plan is not yet "committed" - we have to figure out staffing implications, so please do not forward this mail to all and sundry.

3. DAO

Dave Fulton has chartered a task force to drive the DAO interface to the point where it could be sufficiently defined as our general container access API, and Rogerh has declared it high priority. Cairo people are participating, and I have asked them to get darryr in review loop. We should get update on this next week.

4. EMS

There is proposal on a "middle cost" road that we will have to take on providing the coexistence/migration facilities for EMS and "CMS" (the "Cairo Mail System"). We should have the costs by next week.

**Systems Release Strategy - Draft: 10/10/93**

This memo deals with the current issue, and the options, we face in our priorities for systems releases.

Note: Others have seen these issues earlier and more clearly than I have - jimall, bradsi, and jonl all have the right to say "I told you so".

**Background "Facts":**

1. The current center of gravity in the installed base market is 4MB, and the center of gravity in the new machine market is beginning to approach 8MB. It will not be until 1995 before 8MB is center of gravity of installed base, and 16MB the center of gravity of new machines (and laptops will lag this trend, just as they currently do).
2. Thus in the market place at the current moment, we face the following two major problems: (i) Windows NT does not run in 8MB, and (ii) Windows 3.1 (which does run in 8MB) does not multi-task well. These two facts are what are currently undermining our ability to say that Windows family covers most of our customers computing needs, and is what is leaving OS/2 a major opportunity to become entrenched (as an "8MB desktop OS that does multitask"). We should not panic, but we should be very focused on this as a very real threat.
3. As a consequence of not running well in 8MB, NT sales will be anemic in CY'93 and CY'94 - we will not likely sell more than 250,000 units worldwide in FY'94 (counting the 80K PDK sales). While NT 3.11 (forecasted for May'CY94) will improve this situation (NT 3.11 will most likely get NT to run acceptably in 12MB, and well in 16MB), it will not alter it materially.
4. Cairo will not lower NT's resources requirements, and may in fact increase them.
5. As a consequence of being outside of the "mainstream" in terms of machine resource requirements in CY'93, CY'94, and probably even CY'95, and thus being low volume in these years, it will be impossible to get broad ISV attention to function that is unique to the NT platform.
6. For this reason, we will have to realize that we have only two major "ISV events":
  - (i) Chicago in Q3'94 (when machine requirement for high volume is: runs in 4MB, runs really well in 8MB)
  - (ii) Successor to Chicago in late '95/early'96 (when machine requirement for high volume is: runs in 8MB, runs really well in 16MB).

This means that NT/Cairo will have to derive its broad ISV support from the above two platforms, and consequently the priority of running Chicago and "successor to Chicago" apps must be very high for the Chicago and NT/Cairo groups. Beyond the server, there will be few other NT/Cairo unique apps.

*More background: Objectives:*

As a division we have the following objectives (in priority):

1. Above all, maintain desktop market share.
2. Increase revenue and profit by raising the per PC revenue that we get from \$35 (on average) to over \$50 (on average) over the next 3 years.
3. Keep the ISVs busy implementing new Windows functions, and the cloners on a treadmill.
4. Grow our market share on the server to over 30% over next 3 years.
5. Establish new functionality to allow us to accomplish above goals, and prevent "middleware" from reducing our OS's to "graphical C> prompts":
  - support for workgroup computing
  - distributed systems capability
  - object oriented programming paradigm
6. Make things simpler for the end-user:
  - make the PC an appliance

**Implications of the above objectives:**

1. **Market Share:**

Since our most immediate issue is a possible loss of market share to OS/2 on the desktop, we should focus on the most immediate way of addressing it - i.e. get Chicago shipping as soon as possible. Bradsi and I must be willing to lie down on the tracks over this one. Beyond its currently committed set of functionality, and subject to any major performance/size/useability issues, there is nothing worth delaying Chicago over. This will become a major issue internally as we seek to make Chicago our next "silver bullet for all ills". It means we need to be prepared to decide how to deliver things like Russ's Online client & Paulo's Multimedia stuff in an update release, or through alternate means.

What does it mean for Chicago to stop OS/2 dead? I believe we have to do following:

- (i) Provide a smooth multitasking environment - this does have customer value, and IS what is currently selling OS/2. To do this we need to get the focus moved to 32bit applications. This means that it is critical that the MS applications group (at a minimum) have a 32bit version of Office ready to go with 60 days of Chicago.
- (ii) Perform very well on 8MB (better than OS/2), and OK for existing scenario's on 4MB.

- (iii) Have a "cool" image. I believe that our new Shell, Plug n'Play, and Remote operations capabilities will do this.

## 2. Increase Revenue and Profit:

The most immediate opportunity to do this will come from increasing net per PC revenue on Chicago. The basic strategy here will be to:

- (i) Offer OEMs following options:
- a base ("silver") version of Chicago at same price as DOS/Windows today (approx. \$35)
  - offer them a Premium ("gold") version at a significantly higher royalty rate.
  - offer them pieces of the premium versions for extra royalties (over the base rate), so that OEMs can selectively enable hardware features.
- (ii) Offer end users an upgrade package that upgrades any version of Windows (past or present) to Premium ("gold") level. The net retail revenue of this package would be equal to or greater than the royalty we would receive from an OEM.

Figuring out this packaging such that it will be acceptable customers and OEMs in terms of value, will not cause us to lose market share, will not be used against us by our competitors, and will incent ISVs, is a large and key challenge for us.

### *What strategy to pursue for NT? First yet more background:*

The following are all priorities for NT/Cairo today, we need to decide which ones to optimize for, in what time frame. I list them out first, before giving some recommendations:

1. Compete for market share vs. Novell:  
Since this is a priority for us, and since this is something that clearly Windows 3.1 and even Chicago cannot address, this should clearly be a high priority for the NT effort. We should probably say that until 16MB becomes "normal" (late CY'95/96), should we say that this should be the #1 priority for the NT group? If so what about Notes (see next objective)?

However, competing with Novell is complicated in that, as Jimall and I have previously articulated, the way to win market share on the server, is to win it by changing the relationship between the client and server - and have the client increase the functionality and integration it expects from the server - only then can we will start to really turn the tide. This means that until we can get OFS function onto a high volume client (late'95/96), we are going to have slow progress vs. Novell.

2. Compete for market share vs. Notes.

Again this is inextricably tied up with the client. I do believe that "OFS" strategy of:

- (i) "consolidating the all the incompatible, hard to administer, workgroup databases into the file system", and
  - (ii) "enabling the Shell to be the basic browsing/query/forms composition tool for the workgroup database") is a very powerful one, but it will not work until we have this function (OFS and the OLE based Shell environment) on the high volume client.
3. Defend against OS/2:  
The goal here was to use NT's "real OS" capabilities vs. OS/2. Owing to memory requirements as noted above - we will be able to use NT only in limited situations.
  4. Defend against UNIX on the high-end desktop and the server.  
This remains a viable goal for the server but it will mean having to do a lot of the server apps ourselves (at least initially) to overcome momentum and hostility from existing UNIX server ISV (see NT as commoditizing their market). On the client we probably have to pay more attention to "workstation" technology (e.g. 3D) - but even if we are successful, the numbers will be small.
  5. Pioneer and introduce new software technology:  
e.g. Win32, DFS, DS, OFS, CairOLE, Components Forms, etc. The issue here as noted above, this will not matter to ISVs, or affect the Novell/Notes war until they are exploited on a volume client.
  6. Support new types of hardware:  
Windows NT was engineered to support new hardware: SMP and RISC. SMP is important today on the server, and may become important on the client in '95/'96 (Intel is positioning P54C - 0.6 micron Pentium - to be used in pairs, to provide an upgrade path for customers).

RISC could become important, but not until the next generation of RISC. I.e. to be brutally honest, in the current contest between R4400, HPPA, PPC 601 and Pentium - it is safe to say that 486/Pentium/Intel clones will be the winners on the PC desktop. To the extent that RISC is important in near term, it will be on the server. The next opportunity for RISC at the desktop will be in the timeframe of the P6 vs. MIPS R10K (a.k.a. T5)/IBM PPC 620 battle (CY'96). To win, they have to open up a > 2x advantage.

The above points paint a fairly bleak picture of our investment in NT hosted technology being two/three years ahead of its time in terms of giving us a decisive market advantage. We face triple whammy of (i) not deriving decisive competitive advantage from it during this period, (ii) having to fund it (550 people in systems), (iii) having to explain and market it.

**So what should we do with respect to NT/Cairo?**

First, I think we should consider what our medium/long term (CY'96) endpoint should be. I believe that this should be a Windows client and server that do the following:

- support the functions needed for us to compete long terms (new file system, and UI and programming environment that exploits it),
- is truly scalable across hardware (i.e. we have one client SKU that covers range of interesting hardware).

In this timeframe, the concepts of NT, Chicago, Cairo disappear for all customers and ISVs (exception perhaps of device driver writers). We are back to "standard" and "enhanced" mode. Customers get one SKU for client, one SKU for server, and maybe decide to enable some extra functions by paying more.

When can we realistically achieve this? If we say that the scalable client is based exclusively on NT ("plan of record"), then realistically this going to be when a Pentium class machine with 16-24 MB of memory will be both high percentage (> 60%) of new machines, and greater than 30% of installed base - this will not be until late CY'96/97 (i.e. 3+ years from now). If we want it before then, it means that we have to have an "8MB client" (with OFS, and associated UI/programming environment) - and I believe this means that we cannot rely solely on the NT base to deliver this, and must host the next set of key client functions on Chicago as well (latest code-named for this release is "Memphis", but has also been called "ChiCairo" and "London").

**Key features of NT/Cairo:**

1. NT base functions (portable, secure, high-performance server, SMP):  
We probably do not need these for the client until CY'96 - and then only high-end clients. It is OK to leave these "16MB" unique.  
These functions are crucial for the server, and we should invest to keep them competitive vs. Netware, UNIX, Workplace OS.

**2. API's:**

Given discussion above, we will have the following two generations of API's:

- (i) Q3'CY94: "Chicago":  
The Chicago generation of Win32, and the EMS generation of "MAPI" & "Capone message types" (I don't totally understand the latter). We have to make sure that there is a release of NT (call it "Cairo") within 6 months of Chicago that definitely runs the Chicago generation of Win32/MAPI/Capone message apps - all of them. Given this six month constraint, we need to decide how many other objectives we can accommodate:
  - how much supersetting of rest of Chicago can or should be done (e.g. Plug and Play),
  - what can be done to improve combination of EMS/NT combination
  - how much next generation (see next point) technology can we introduce on NT (DFS, OFS, DS), and how much of this technology could be exposed to the Chicago in this timeframe (e.g. could we at least release DFS for Chicago at same time as Cairo),

- how much can we allow into NT 3.11 without jeopardizing above.

- (ii) Q1'CY96: "OO Environment on Chicago" (or "Memphis")  
This is when we take functionality currently in Cairo, and put it on the 8MB client - i.e. OFS, Cairo OLE, Component Forms based UI/Shell, Component Forms based end-user development environment ("CDE") which competes with Notes's end-user environment. This combination of function replaces to a large extent the MAPI/Capone messages environment.

We can make two decisions about this "OO Environment" function:

- (i) Subject to getting Cairo out within six months of Chicago, Jimall can decide how much of this function to put into Cairo as get it done early, allowing us to have it tested for volume deployment in CY'96. But until then we should be under no illusions as to how this would help vs. Novell/Notes.
- (ii) It is confusing to even have to speak about it until we have it on the high volume client - until then (Q1'CY96) we will just have to take our lumps and use NT's base function to compete with Novell, and EMS/MAPI/Capone to compete with Notes.

Options for public positioning NT/Cairo before Q1'Cy'96:

Given above options for NT/Cairo priorities, our public position can be:

Option A:

- MS's UNIX for the Server (scalable, secure, SMP, etc.)
- A secure version of Chicago for the client (which requires 16MB).
- Technology introduction platform for new API's and distributed systems features.

Option B:

- MS's UNIX for the Server (scalable, secure, SMP, etc.)
- A secure version of Chicago for the client (which requires 16MB).

**Recommendations:**

1. Ship Chicago ASAP to hold our market share - don't hold it back for other objectives.
2. MS must ship 32bit Chicago apps within 60 days of Chicago to support Chicago. We need the applications group to re-align around this, and consider implications on this like component forms.
3. Adopt decision (I) and positioning A on NT - but do not let priorities get confused. It means we must close all remaining issues on API that are exposed in Chicago, and which have to be supported in Cairo.

4. Ship Cairo within 6 months of Chicago (i.e. no later than Q1'CY95) with 100% superset of API ad UI content- ensure that we have 95% confidence plan to do that.
5. Articulate clearly where we are going: scalable client and server in Q1'CY'96.

Paulma: rev 0, 10/8/93, page 7.

**CONFIDENTIAL - draft**

**MS7048990  
CONFIDENTIAL**

---

MS7048990

## SECTIONS NOT COMPLETE

### *Interaction with MS Applications Release Plan*

Current MS apps plan is:

- do minimal update releases for Chicago with 60 days of Chicago.
- do a major release in mid'CY95 based on Chicago API set and Component Forms.

### *Interaction withn MS Tools Release Plan:*

### *The Cross-platform Challenge*

Both Novell ad Lotus are mounting an explicit cross-platform challenge to us. How should deal with this?

### *Organization/Morale:*

Are we organized optimally to do all of the above? What are morale implications for Systems (esp. NT team)?

**PX 102**

**Erik Stevenson**

---

**From:** Steven Sinofsky  
**To:** billg; bradsi; davidcol; dennisad; jimall; jonl; paulma  
**Subject:** FW: Novell AppWare -- first impressions  
**Date:** Friday, November 05, 1993 10:52AM

This is very interesting. I was looking over the books and stuff and was on the one hand impressed (they got all the right stuff for MIS people--style guides, methodology, etc.) and on the other hand it is scary since it is just another windowing API, and a fairly complete one. This is direct competition to Windows, not VB or VC++ . Not what I was expecting based on their white paper (which seemed to focus on using the recently acquired Serius Developer development tool, with some C code for customization).

We're not sure what Borland is going to do with this, but they are making a lot of noise about it.

-----  
**From:** Scott Randell  
**To:** SYS Denis Gilbert's direct reports; Garth Hitchens; Luis Talavera; Richard Tong; Steven Sinofsky  
**Cc:** Scott Randell  
**Subject:** Novell AppWare -- first impressions  
**Date:** Thursday, November 04, 1993 9:04PM

re: Novell AppWare Foundation SDK

=====  
=====  
Well I've taken a quick look at the "NAWF" SDK. Here is my overview and my impressions->

What is it ?

It is just the SDK for the NAWF API. It does not include any visual tools or compilers or anything else of interest.

It is / it contains:

- A set of headers for the NAWF API
- A set of libraries and DLLs for the NAWF layer on Windows.
- Some sample code, an online API help file.
- A few funky tools for dealing with their strange resource scripts and other AWKfull things.

It is not / it does not contain:

- No visual tools
- No compiler
- No extensive examples (ok they have Draw and Clock)

How done is it ?

It is a very complete product for an API -- with a lot of documentation. It comes with many books covering the API reference, overviews, a study guide, coding conventions, and so on. They also have a training course for people to learn about it.

They've done a good job, as SteveSi put it, targeting what MIS people like -- thick books, spelled out conventions, very structured taxonomies, courses for learning, hand holding and customer support. The documentation is well written -- but still a little raw

(including review comments to the writers).

The next release of NAWF will be available in February.  
The SDK is only for evaluation purposes -- in the future  
you will need to get a licence to incorporate the libraries.

What is the NAWF API:

It is a Windows-like C API that abstracts out platform  
specific differences. They claim they currently run  
under Windows (Win 3.x, not NT), the Mac and Unix (X Windows).  
There is no OS/2 version yet. The SDK Kit I got from LuisT  
is just for Windows development.

The API is split into three major categories:

- Operating System (base features)
- Network Connectivity
- User Interface (gui)

It is really just a C API that looks a heck of a lot like Windows.  
It is not really object oriented in any way beyond what WndProcs  
and switch statements in DialogProcs give you.  
The flavor of the APIs are quite different with different conventions  
(naming conventions in things like passing access objects around)  
and especially the way objects are instantiated,  
There are lots of public and unprotected data structures --  
not unlike many C API or portability toolkits I have seen  
in the past.

There are a ton of APIs -- they revel in their complexity.  
Over 750 new APIs (in addition to the platform specific APIs  
that you may choose to use). Over 150 data structures  
(with all public data fields, remember this is C),  
and over 3500 #defines for constants (and I thought  
Win32 was complicated !-).

They mention a C++ framework (NAWFFW Class Library) layered  
on top of the NAWF API. It was not part of the SDK.

How do you use it.

For Windows:

You develop and compile code using your favorite PC development  
tools (MS C, VC++, BC), edit your resources using your favorite  
tools (DlgEdit, App Studio, Resource Workshop) and just compile  
and link with the NAWF headers and libs.

They have a bunch of funky command line tools to process  
resource files, not unlike Bedrock.  
The resulting application has the proper native look and feel.

For the Mac, you use MPW (not enough stuff was provided  
with the Windows SDK to do that of course).

=====

MORE DETAIL:

The Operating System "Series":

This consists of general purpose functionality, abstraction  
of the operating system and hardware. This is very  
similar to the services provided by Windows KERNEL, SYSTEM and GDI.  
Includes the following "components" or services:

- Character (including locale)
- Data (including collections, formatting, currency, ...)
- Error (for exceptions -- but using GetLastError and

- error hooks)
- File (including aliases, directories, ...)
- Font (platform independent font selection)
- Graphics (platform independent graphics -- hmm, looks rather GDI-like)
- Instances (this is the unique notion of an object created in a IMHO strange way, usually from some resource information (i.e. a super LoadResource -- not unlike MacApp).
- Keyboard (access to key state)
- Memory (abstracted memory management -- yes with moveable blocks)
- Messages (callback procs, postable messages -- when used it looks a lot like Windows MSG)
- Module (abstraction of HMODULE and other app global things)
- Pointer (cursor on the screen)
- Preferences ( = = preferences)
- Print (device independent printing, GDI like)
- Records (for storing in collections, simple data structure with a hell of a lot of complexity)
- Resource (resource management ala Windows or Mac)
- System (misc system services and general machine info)

The Network Connectivity Series:

This is one of the unique things they have that they play up a lot -- they have a model where they are more network aware than say Windows.

Includes the following "components" or services:

- Clipboard (similar to Windows clipboard)
- Pipes (a simple pipe model with a ton of options I don't understand -- with network capabilities of course, this is Novell remember)
- Tasks (platform independent API to get at multi-tasking info -- doesn't do anything to the 2/3 of the platforms currently supported).

The User Interface Series:

This is the main USER level user interface functionality. They have no "high-level abstractions" (save perhaps Table) like toolbars or status bars. They are designed around a very simple UI, so there aren't a lot of UI Widgets (like spin buttons).

Includes the following "components" or services:

- App (misc global app data structure, also provides a Windows specific one too)
- Box (~ = group box)
- Button (~ = Windows BUTTON control)
- Dialog (souped up dialog manager)
- Display (~ = Windows STATIC control)
- Edit Text (EDIT control, but abstracted to cover Mac TextEdit)
- Help (simple help hook, but like many components they have a lot of optional bits -- some of which may actually do something).
- Item (screen drawable object -- good for static graphics).
- List (LISTBOX more or less)
- Menu (~ = menu)
- Slider (a slider control)
- Standard Package (~ = COMMDLG)
- Table (two dimensional grid control, really ugly)
- Void (terrible name, but this is a place holder)

for custom output to be drawn on a dialog.  
If you can't layout a dialog with the canned  
Buttons (BUTTON), Display (STATIC),  
List (LISTBOX) or Items (static graphics),  
then you plop down your own custom "void"  
component.

Window (very HWND-like as you probably could have  
guessed).

For all the above when I say "similar to Windows XXX" that means it  
looks functionally very similar, and even the flavor of the API  
is similar. Of course the worlds are very very different when  
it comes to looking at what code you write. It would require  
a complete rewrite to convert a Windows app to NAWF or vice-versa  
because there are so many basic differences.

For your amusement, here is the C code to their Hello World  
sample application ->  
<<File Attachment: SAMPLE.C>>

IMHO->

I personally wouldn't want to program to this API, so if it turns out  
to be popular, then something like OWL or the NAWFFW  
class library on top of it would be really valuable to  
programmers who want the portability feature of NAWF  
without having to deal with the crazy C API.  
Windows programmers won't like it since it is not Windows,  
Mac programmers won't like it since it is too Windows-like  
(too un-Mac-like). Don't ask me about Unix programmers.

=====  
=====

What this means to us (both VC++ and MS as a whole) ?

I'm not really sure.  
This is a competitive API, competitive to the Windows and Win32  
APIs. It is positioned as a portability toolkit.

Our Mac/Unix story is very similar -- instead of the NAWF  
API, we have the Win32x API.

This isn't a Bedrock or a Tallent -- it isn't an object  
oriented system API, it is a strange portability toolkit  
that isn't functionally all that much better than Windows.  
And IMHO it is pretty bad from an API point of view. This  
won't get acceptance as a better API than Windows -- except  
that it has a "portability story".  
Bottom line NAWF is not a direct competitor to MFC.

OWL on AppWare would be a direct competitor to MFC on Windows.  
This of course depends a lot on what Borland does with its  
OWL2 and OWL2/Appware products.  
The mapping from OWL2 to NAWF would be very weird regardless.

I personally think we can turn this to our advantage, something like:  
"Geez Borland just screwed over OWL 1 users with OWL 2 on Windows,  
now they are abandoning Windows for that AppWare platform. What's  
AppWare? -- it's a platform/API that Novell is in the process  
of designing that looks a lot like Windows, but isn't Windows  
-- isn't object oriented -- and by the way requires a runtime  
licence you will have to pay to Novell..."

Since AppWare is not vaporware (at least with this Early

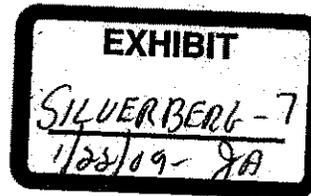
Adopter version) -- we can turn this to our advantage.  
OWL/AppWare is vaporware (even a TeamB guy  
on CIS said that) so we should play up the fact that  
OWL/Appware is a vaporware product on a very rough  
infrastructure. The only danger I see is if "OWL/AppWare"  
remains vapor or pseudo-vapor like Bedrock, Taligent,  
OpenDoc,... -- you know how bad we are at competing  
with vapor.  
Since AppWare is concrete - if developers look at it I don't  
think those developers will want it.

=====

all for now....

...scott

**PX 105**



Erik Stevenson

From: Brad Silverberg  
 To: jimall; 'smtp:bobmu'; paulma  
 Subject: FW: WP vist  
 Date: Thursday, November 18, 1993 9:38AM

fyi

From: davidcol  
 To: bradsi; bradstr; dennisad; georgem; jefft; joeb  
 Cc: davidcol; doughe; johnlu; marionho; robp  
 Subject: WP vist  
 Date: Monday, November 15, 1993 5:48AM

Jefft, Bradstr, and I went to WordPerfect last Thursday to talk to them about what we thought a good Chicago app was and what barriers they would have to doing one close to the time Chicago shipped. Overall, the visit was good. There were around 10 WP guys, (VP dev lead types) sitting around a table so it was much more intimate than the Borland visit. They weren't nasty at all, in fact had good feedback and decent questions.

These guys will bet on Chicago, they've never had any doubts about that. They want to create a single binary that exploits Chicago and supports NT. (exactly what we want) However, they need to fo think about this before they can concretely indentify any barriers. The kind of help the KNOW they need is help with sample code, help with style guide issues, good docs, etc. They thought a private forum on compuserve might be good, email contacts would be great too. We need to decide how to support these guys. They will have a separate win16 version which they'll keep on the market until they don't need to anymore. It was interesting to see how enthusiastic WP was about Chicago, much in contrast with the ho-hum attitude of our own apps group.

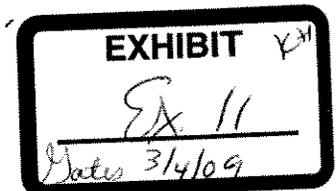
Bradstr was going to get email names of all the guys and who does what. I think it would be good for us to establish decent relations with these guys.

They were very happy about us deciding to document the shell extentions. I explained conceptually how the extensibility would work and what controls they'd have. Since they just aquired a document management system (I forget from who) I assume they will want to plug that in, plus WP mail and other part of WP office too. I'm sure they will also supply shell property sheets for their docs too.

They use the char versions of the MS tools, so the M5 PDK is exactly what they need to get started and they will. There were lots and lots of other areas that they liked a bunch; the shell, move/copy for the transfer model (I said it was still open), new help features, (especially jumping from help into an app), but I won't get into it all here. I anticipate that WP will have a very exploitive Chicago app ready close to when Chicago ships.

Here are some other notes and action items I wrote down: (most of the nits came from Tom Crux(?) who is their shared code dev lead, worth getting to know for lots of reasons, not the least of which he would be the one to directly support Appware if that was going to happen for WP)

- interested in shipping a good viewer with Chicago. I said we would be very interested if it was 50-80 kb. They wanted to know if there'd be a good viewer for word in the box. we need to followup on this in a few weeks.



- would like to see a global key/local key scheme for the registry. global keys are set centrally by a net admin. Apparently they have done this sort of thing privately for wordperfect. They would be willing to share their requirements and how they went about doing it since this is all shipping stuff. Robb or Johnlu should talk with them. Bradstr, please give rob and John the name of the correct WP guy to talk with.
- mentioned multithreaded MFC, or rather lack of.
- thought we should allow property browsing in the shell. IE lock down a property window, then select random docs and see it's properties. I think we are going to do this as part of the viewer UI.
- they call the winnet apis directly since they have their own fileman and will probably continue to do so. they want to make sure we document these at least in the ddk. would like to have docs written which describe better how apps call winnet, but not a requirement.
- complained that help only allowed for max index size of 9000. marionho?
- thought help should allow OLE embeddings. great idea, no time to get it done..
- shell needs to allow extending the find command. at minimum allow apps to extend the menu to search other stores. We should also allow a simple global that just returned results in separate results windows for each type of thing being searched. I think this would be pretty easy. Joe, we should think about this some more. I bet we can do something really cool that simple. What does the mac do for searching across mail and files?
- wanted different sprites. partially transparent thing that moved around on the window. they have a "coach" thing which moves around and points things out. apparently it's really really hard. they thought they heard davidw say he did this for Chicago. george?
- they want to set a global search path in Windows and not force the system to reboot. An app search path would be great, but they thought we didn't need that much. Basically just add to the global search path from in Chicago would be great. Perhaps we should have a WindowsPath= and it was just appended to the dos path when we searched. george?

**PX 110**

CAL

**Shiba's Breakthrough Color Subnotebook!**

**Incorporating PC Sources**

December 1993

# PC Computing

# Annual Awards

**100 Best Products of the Year**

**WE PICK THE TOP PRODUCTS IN EVERY MAJOR BUSINESS CATEGORY**



**PLUS**

**WordPerfect 6.0 for Windows  
Excel 5.0 • QMS Color Laser  
Borland's New Paradoxes**



# New! Software

LOOKING FOR THE ULTIMATE WINDOWS PROCESSOR?

# WordPerfect 6.0 for Windows

**BUCKING THE TREND**  
It's a constant debate at WordPerfect—whether to toe the Windows line in the interest of a common user interface or to create a program that's more usable than the standard Windows look and feel (top icon). In most cases, according to Dave LeFevre, product manager for WordPerfect for Windows, the programmers didn't worry very much about



Microsoft's recommendations—such as with the "cut" icon—with mixed results (bottom icon).



On one hand, innovations such as managing files from the Open and Save dialog boxes are great.

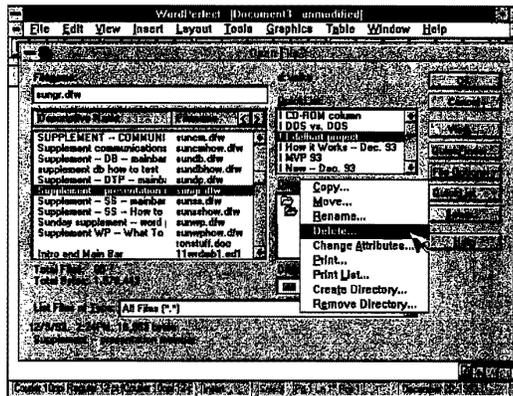
And WordPerfect's QuickList—which lets you assign directories and files to a quick-access list and avoid thrashing through umpteen layers of subdirectories—is pure genius. On the other hand, some innovations make WordPerfect inconsistent with itself. The Browse function in a typical dialog box is sometimes the Windows-standard button, but in WordPerfect's Play Macro dialog it's a mysterious file folder icon.

"Microsoft would have us use common icons for open, printing, and saving," LeFevre says, "but we had no reservations about varying from the standard if we had something better."

**W**ORDPERFECT 6.0 for Windows isn't just another software program. It is a computer user's garden of unearthly delights. An MVP Award winner, this latest version of WordPerfect makes major gains in the word-processing features war, adding functions available in its chief rival, Microsoft Word for Windows, and then some. By including a full spreadsheet, drawing program, hypertext, multimedia sound and video, charting, and support for fax and e-mail, you could easily use WordPerfect alone to replace half a dozen other programs. That's good, but what really shine in the new WordPerfect are its usability tools.

Take toolbars. If you like the convenience of a row of buttons that you can click to perform common tasks, you'll love WordPerfect's Power Bar, Button Bars, and Feature Bars. The Power Bar is a collection of buttons that you'll find on most decent Windows programs. It's filled with common functions for opening and saving files, formatting text and pages, and so on. WordPerfect comes with 12 versatile Button Bars, and you can create as many others as you wish, packing them with buttons that launch menu selections, macros, or even other programs. Twenty Feature Bars filled with the corresponding buttons pop up automatically when you activate features such as graphics, hypertext, indexing, footnoting, and outlining.

But here's the really good part: After supplying all these wonderful toolbars, the program lets you get along easily without them. Clicking the right-mouse button brings up a floating QuickMenu that includes all the functions you're most likely to need for whatever you're doing at that moment. Whether graphic, table, or plain-and-simple text, you get the appropriate menu options. We spent hours performing complicated tasks with WordPerfect without



Unlike Word or any other word processor, WordPerfect's Open File dialog box includes long filenames, and makes file-maintenance functions available in a pop-up menu.

ever using any of the normal menus or toolbars. Everything we needed to do was instantly accessible with a button click.

QuickFormat is another ease-of-use feature that sets this program apart. It lets you select a section of text with formatting that you want to apply to other text. Then you use a paint-roller icon to spread the formatting around wherever you want it.

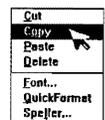
QuickMenus and QuickFormat offer meat-and-potatoes usability. When you're ready for dessert, check out the dialog boxes. Any Open or Save dialog box includes terrifically helpful aids that you won't find in any other word processor. QuickLists let you access commonly used directories or files quickly and easily. Gone are the days of traipsing through a series of convoluted subdirectories to find the one you want. Simply assign any directory or specific file to the QuickList and give it an intelligible name up to 30 characters long. Changing to that directory or opening that file requires only a double-click of the mouse.

### File Management Done Right

WordPerfect expands the tools found under the dialog box's File Options button in WordPerfect 5.2 for Windows. Without leaving WordPerfect, the options let you delete, move, rename, print, copy,

or change the delete directly to save a file create a new Customiz

Guid  
Tired of  
of its man  
creating



Quick  
Power Bar

# Windows Does It All

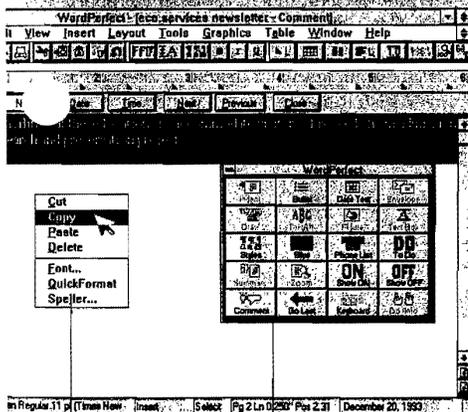
or change the attributes of files, as well as create or delete directories. (How many times have you gone to save a file only to discover that you first need to create a new directory for it?)

Customization has been a tradition in WordPerfect and

continues in Version 6.0 for Windows. Menus and toolbars are totally customizable. If you think WordPerfect made a mistake—as we do—by making Macros a submenu under the Tools menu, you can move it back to being a menu item. Or if you'd rather have your Button Bar on the left side

## Guided Tour: WordPerfect 6.0 for Windows Gets Friendly

Tired of digging through WordPerfect's complicated menu structure to find the right command? Using one of its many toolbars, the most-used commands are no more than the click of a mouse button away. For creating more complicated documents, WordPerfect steps you through the necessary tasks with Coaches.

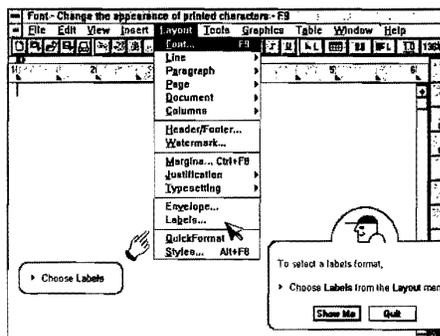


1. WordPerfect's features are available at the touch of a button from the Power Bar at the top of the screen below the menus; from context-sensitive QuickMenus that pop up when you press the right-mouse button; and from Button Bars, shown configured as a floating palette of buttons.

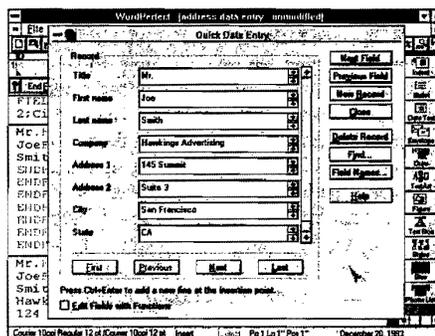


2. For amazing text effects, WordPerfect 6.0 includes Bitstream's TextArt. Mold type into attention-getting shapes, then paste it into ads, brochures, etc.

ower Bar



3. Coaches are animated training aids that teach you how to use features such as label creation by pointing to each step along the way.

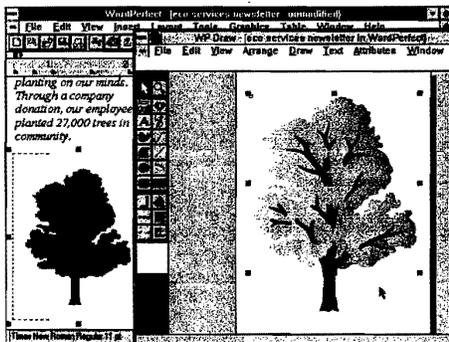


4. Mail merge gets needed improvements with forms to create data files and boilerplate letters. The merge language is still available for sophisticated tasks.

## New! Software

### CHART YOUR COURSE

Let the critics be silent! In previous versions, WordPerfect for Windows lagged behind its major competitors—Microsoft Word for Windows and Lotus Ami Pro—particularly in the drawing and charting arenas. But no longer. Version 6.0 includes WordPerfect Draw, the



some drawing and charting module found in WordPerfect Presentations. Want to change the color of your company's logo? Simply double-click on the image and make the change from within WordPerfect Draw.

In addition to supporting the usual selection of tools, lines, fills, and Bézier curves, WordPerfect Draw has complete charting functions like those in any presentation package. You can create line, bar, area, and 3-D charts, as well as combination charts. And because WordPerfect 6.0 supports OLE 1.0, editing the chart is as easy as a double-click.

of your screen, simply drag it over there.

WordPerfect also includes interactive tutorials, or Coaches, that take you step-by-step through a process, such as inserting a graphic or creating a footnote.

The functions found in Coaches are similar to WordPerfect's Express Docs. Found under the File menu's Template option, Express Docs take templates to a new level. Template suggests that you merely retrieve a form and replace a bunch of placeholders with the information you really want. Express Docs automates the process for dozens of preformatted and professionally designed documents

from fax cover sheets to legal billings by taking information, such as your name, from a database to fill in parts of the template and by prompting you for the rest of the information needed for the form. WordPerfect ships with more than 45 templates, and as with Coaches, you can create your own.

What helps to make Express Docs so powerful are the spreadsheet capabilities built into

WordPerfect's tables. No one is pretending this is a replacement for Borland Quattro Pro or Microsoft Excel, but with data fills, named ranges, and more than 100 formulas, it's more than adequate for many of your spreadsheet needs. Documents such as invoices can use spreadsheet table functions to automatically calculate totals and sales taxes. And *floating cells* can be placed in the middle of a text paragraph and show the results of a calculation from a spreadsheet table on another page.

Even with all the spreadsheet functions built into the program, WordPerfect still maintains its ability to link to spreadsheets created by Lotus 1-2-3, Excel, and Quattro Pro, in addition to OLE and DDE links (but not support for OLE 2.0—yet). Using WordPerfect's own linking abilities means you don't need to have access to the programs that created the spreadsheets. Similarly, WordPerfect has the ability to directly read and query database files created by dBASE, Paradox, Fox, DB2, Oracle, SQL servers, and others. When a program can be used as a

front-end for so many spreadsheets and databases, it's inaccurate to call it a word processor. It's a *work* processor.

### Improved Mail Merge

WordPerfect is late to the party with some other new features, however, having just added drawing and charting modules like Word for Windows and Ami Pro. And WordPerfect is playing catch-up by improving the interface on its mail-merge function, which has scored low on *PC/Computing's* usability tests in the past. The new mail merge prompts the user through the creation of mailing lists, and a Feature Bar lets a simple click insert merge fields in a letter.

So what's wrong with WordPerfect 6.0 for Windows? Not much. It doesn't support OLE 2.0, although that's planned for an upgrade release. The grammar checker is still dumb, but what grammar checker isn't? We wish it had Word for Windows' AutoSelect, which takes care of those pesky extra and missing spaces when you delete or move text, and AutoCorrect, which fixes typos on the fly. And we wish the QuickMenus could be customized. Then again, we need something to live for, don't we?

More important, the great usability features in the main part of WordPerfect don't carry over to the modules that have been tacked onto the program. QuickFinder, for example, lacks the file-management tools found in most other file-related dialog boxes. And little thought has been given to using the program without a mouse. WordPerfect's programmers should be forced to travel on a coast-to-coast flight—with a notebook and no pointing device—and use all of WordPerfect's features. Maybe it would make them believers.

Is there any serious reason not to get WordPerfect now? No way. Version 6.0 strikes a perfect balance between Windows interface standards and WordPerfect functionality. Are there serious reasons to buy it? Only if you want a word processor that takes the brakes off what you can do with a document—from word processing to spreadsheet computations to drawing—and makes the process so easy that it's more fun than work.

RON WHITE

**WORDPERFECT 6.0 FOR WINDOWS** • \$495 (\$129 UPGRADE FOR WORDPERFECT USERS; \$149 UPGRADE FOR USERS OF ANY OTHER WORD PROCESSOR WITH A LIST PRICE OF AT LEAST \$395) • WORDPERFECT CORP. • (800) 451-5131  
Circle No. 610 on Reader Service Card

**New!**

## Software Saleswire

OS/2 2.1 and Windows NT: Two 32-bit operating systems make the top 10.

### DOS APP

#### DOMINATES

The DOS-based WordPerfect 6.0 is the No. 1 seller for the third month in a row, while the competing Word for Windows has climbed out of the sales doldrums and into the top ten for the first time since March.

BUSINESS APPLICATIONS	VENDOR	PRICE	COMMENTS	SEP	OCT
WordPerfect 6.0	WordPerfect	\$495	A GUI-like DOS word processor.	1	1
Lotus 1-2-3 4.0 for Windows	Lotus	\$495	Designed to ease workgroup computing.	7	2
Excel 4.0 for Windows	Microsoft	\$495	Crunches numbers and makes 3-D charts.	5	3
Office 3.0 for Windows	Microsoft	\$750	Word, Excel, PowerPoint, Mail, and Access.	3	4
WordPerfect 5.2 for Windows	WordPerfect	\$495	Drag-and-drop editing and grammar check.	4	5
Word 2.0 for Windows	Microsoft	\$495	Wysiwyg word processing.	-	6
Access 1.1 for Windows	Microsoft	\$495	A relational database with a GUI.	2	7
WinFax Pro 3.0 for Windows	Delrina	\$129	Send and manage faxes from your PC.	8	8
Publisher 2.0 for Windows	Microsoft	\$199	Microsoft's answer to low-end DTP.	9	9
Procomm Plus 2.01	Datastorm	\$129	Dial up data with this comm software.	10	10

### 32-BIT BATTLE

Windows NT, with its multithreading and multiprocessing capabilities, leaps onto the charts only a few months after its release, ready to slug it out with OS/2 2.1.

UTILITIES	VENDOR	PRICE	COMMENTS	SEP	OCT
Windows 3.1	Microsoft	\$149	Point-and-click, drag-and-drop computing.	1	1
MS-DOS 6	Microsoft	\$129	New version sports many useful utilities.	2	2
QEMM-386 7.0	Quarterdeck	\$100	Frees upper memory for your applications.	4	3
OS/2 2.1	IBM	\$249	Supports 32-bit graphics and multimedia.	3	4
Norton Utilities 7.0	Symantec	\$179	Analyzes, repairs, and optimizes disks.	5	5
After Dark 2.0 for Windows	Berkeley Systems	\$50	Save your screen with flying toasters.	6	6
Windows NT	Microsoft	\$495	Long-awaited 32-bit operating system.	-	7
Saber LAN Workstations	Saber Software	\$179	Management software for Novell networks.	10	8
More After Dark 2.0 for Windows	Berkeley Systems	\$40	Flying toasters with sound.	8	9
Visual Basic 3.0 for Windows	Microsoft	\$199	Develop your own Windows apps.	-	10

### PREPARE FOR TAKEOFF

If software sales are any indication, PC users are plugging into cyberspace with a passion. The CompuServe Windows Membership Kit has been a best-seller for four months in a row.

PERSONAL PRODUCTIVITY	VENDOR	PRICE	COMMENTS	SEP	OCT
Quicken 6.0	Intuit	\$60	DOS-based money management.	2	1
Print Shop Deluxe 1.2	Broderbund	\$80	Graphics for banners, newsletters, etc.	3	2
Money 2.0 for Windows	Microsoft	\$70	Onscreen checkbook shows your balance.	4	3
Quicken 2.0 for Windows	Intuit	\$70	Track checks, credit cards, and expenses.	1	4
CompuServe Windows Kit	CompuServe	\$50	A gateway to online services and forums.	7	5
QuickBooks 1.01	Intuit	\$140	Accounting for small businesses.	5	6
Print Shop	Broderbund	\$80	Nuts-and-bolts graphics software.	-	7
Calendar Creator Plus 5.0	Spinnaker	\$70	Never be late with this scheduling software.	10	8
QuickPay 2.0	Intuit	\$70	Payroll automated for small businesses.	8	9
Quick and Easy for the SF-171	DataTech	\$50	Canvas the federal job market.	-	10

These software charts are compiled monthly by G. Meier, Inc., which gathers ranked-order sales information from the following software industry sources: 800-Software; ABCO Distributors; American Software; Babbages; Baker & Taylor; Software; CompuAdd; CompUSA; Computer Discount Warehouse; Dustin Discount Software; Electronics Boutique; Gate/FA; Ingram-Micro; Kenfil; Merisel; Micro Warehouse; Multiple Zones; Navarre; PC Connection; Software Resource; Software Spectrum; Software, Etc.; and Waldensoftware. The rankings reflect sales of more than \$4 million of software through tens of thousands of resellers. A weighting system is applied to each entry based on three factors: the entry's position in the individual chart, the power of the contributing source based on annual gross sales, and the relative strength of a source in a given category of software. All prices are suggested retail and are subject to change.

# The Gene Mon

ViewSonic's New Ge performance monit

The ViewSonic 15, remarkably crisp, sl enhanced vibrant o innovations include drop-down digital c screen coatings, the system, and resolut non-interlaced.

The biggest surpris less than you'd expe astounding quality.

Call ViewSonic at 8 information on our

Welcome to the I

FEATURE	
ViewSonic	15"
Screen Size	15"
Dot Pitch (mm)	0.27
Horizontal Frequency (KHz)	30-60
1600 x 1280 Non-Interlaced	
Maximum Refresh Rate (Hz) at 1280 x 1024 Hi	60
Flat Square Screen	■
ViewMatch™ Color Control	■
Low Radiation (MPRI)	■
Suggested List	\$549

Circle 150 on

# ViewSonic

20480 Business Pkwy W  
Bellevue, WA 98008  
Tel. (909) 869-7976 Fax

ViewSonic 15" screen image by SHUTTER  
15" Geometric Model 2 and 15" Geometric Model 3  
ViewSonic temperature select 1500K or 6500K  
All products and brand names are registered trademarks.

# CHOOSING THE MVPs

Nineteen ninety-three was a good year for shrink-wrap. More than 5,300 new or updated pieces of hardware hit the streets. In the same period, nearly 6,000 software packages rolled off the production lines. And the person who invented heat-sealing plastic made a fortune. If you're like most people, you probably don't have time to look at more than

11,000 products. That's OK. That's our job.

The editors of *PC/Computing* tore the shrink-wrap off truckloads of these products to evaluate them and let you know which are hype, which are vapor, and which will change your life. This process starts the path to the Most Valuable Product Awards.

The first step is to set our selection criteria. For us, that's easy. We look for technological innovation and usability above all else. Products must not only be the best in their categories but also have innovations that revolutionize the way we think about computing. On top of these innovations, the products must help users be more productive.

After inno  
weigh facto  
performanc  
factors cha  
category. F  
desktop sys  
price rises  
get the bes

Using th  
experts wi

## SYSTEMS

192

### DESKTOP SYSTEMS

- Gateway P5-60
- ALR Evolution V
- Compaq Deskpro XE 466
- Dell Dimension XPS 466V

### ENTRY-LEVEL DESKTOP SYSTEMS

- AST Bravo LP Green
- Gateway 4DX2-66V
- IBM ValuePoint 466DX2

### NOTEBOOKS & PORTABLE PCs

- IBM ThinkPad 750C
- NEC UltraLite Versa 33C
- Texas Instruments TravelMate 4000E WinDX2/50
- Toshiba T4700CT

### SUBNOTEBOOKS

- IBM ThinkPad 500
- Hewlett-Packard OmniBook 425
- Toshiba Portégé T3400CT

### FILE SERVERS

- Compaq ProLiant 2000 Pentium
- Hewlett-Packard NetServer 5/60 LM
- NetFrame NF450C/S



Winner

## PERIPHERALS

206

### MONITORS

- Nanao FlexScan F760iW
- Nanao FlexScan F560iW
- NEC MultiSync 5FGp

### VIDEO ADAPTERS

- Number Nine GXE
- Matrox MGA Impression
- SuperMac Thunder/24 for Windows

### PRINTERS

- Hewlett-Packard LaserJet 4Si MX
- Hewlett-Packard LaserJet 4L
- Lexmark IBM LaserPrinter 4039 16L

### COLOR PRINTERS

- Tektronix Phaser 200i
- Canon BJC-600
- Hewlett-Packard DeskJet 1200C
- QMS ColorScript Laser 1000

### STORAGE DEVICES

- Maxtor MXL-105-III
- NEC MultiSpin 3XE CD-ROM
- Seagate Barracuda 2

### INPUT DEVICES

- Logitech MouseMan Cordless
- Pinnacle Micro Paperless 1
- Nisca Niscan Page

## COMMUNICATIONS & NETWORKING

222

### COMMUNICATIONS SOFTWARE

- Symantec Norton pcAnywhere For Windows
- Delrina WinFax Pro for Networks
- Traveling Software CommWorks For Windows

### NETWORK SOFTWARE

- Novell NetWare 4.0
- Microsoft Windows NT Advanced Server
- WordPerfect Office 4.0

### COMMUNICATIONS HARDWARE

- Intel SatisFaction
- Apple Newton
- Megahertz 14,400 PGMCIA Data/Fax Modem with X Jack

### NETWORK HARDWARE

- Telebit NetBlazer PN
- Intel StorageExpress
- Synergetics LANplex 6012

## BUSINESS

### APPLICATIONS

232

### DATABASES

- Borland Paradox 4.5 for Windows, Workgroup Edition
- Borland Paradox 4.5 for DOS
- Lotus Approach 2.1

### SPREAD

- Borland (Winc)
- Lotus 1-2-3
- Microsoft

### WORD P

- WordPer
- Microsof
- WordPer

### FINANC ACCOU

- Intuit Q
- Intuit Q
- Reality (By)

### BUSIN

- Folio Vi
- Arabesc
- Jensen
- Waterm

### PRESEN

- Softwa
- Gr
- Asymet
- Microg
- WordP

### DESKI

- Micros
- Aldus
- Frame

620Pr.

After innovation and usability, we weigh factors such as price and performance. The order of these factors changes, depending on the category. For an entry-level desktop system, the importance of price rises because the idea is to get the best system cheap.

Using these criteria, our experts winnow the list of

products to a dozen or so contenders per category by comparing and analyzing them. We present each product to the MVP panel, a collection of our top product experts who narrow each category's list to three or four finalists. This all takes more than 100 hours of meetings. In other words, a typical week.

The final stage of the MVP cycle is the selection of the winner. In a daylong meeting, our experts select the product that we believe offers the most innovative technology for its category.

Maybe next year we'll offer our services to the Academy of Motion Picture Arts and Sciences.

#### SPREADSHEETS

- ✦ Borland Quattro Pro 5.0 for Windows, Workgroup Edition
- Lotus 1-2-3 4.0 for Windows
- Microsoft Excel 5.0 for Windows

#### WORD PROCESSORS

- ✦ WordPerfect 6.0 for Windows
- Microsoft Word 6.0 for Windows
- WordPerfect 6.0 for DOS

#### FINANCIAL & ACCOUNTING SOFTWARE

- ✦ Intuit QuickBooks 2 for Windows
- Intuit Quicken 3 for Windows
- Reality Smart Investor Network
- By Money Magazine 2.0

#### BUSINESS UTILITIES

- ✦ Folio Views 3.0
- Arabesque Ecco Professional 1.1
- Jensen-Jones Commence 2.0
- Watermark Discovery Edition

#### PRESENTATION GRAPHICS

- ✦ Software Publishing Harvard Graphics 2.0 for Windows
- Asymetrix Compel 1.0a
- Micrografx Charisma 4.0
- WordPerfect Presentations 2.0

#### DESKTOP PUBLISHING

- ✦ Microsoft Publisher 2.0
- ✦ Aldus PageMaker 5.0 for Windows
- Frame Technology FrameMaker 4

#### SYSTEM

#### SOFTWARE 246

#### OPERATING SYSTEMS & ENVIRONMENTS

- ✦ IBM OS/2 2.1
- ✦ Microsoft Windows NT 3.1
- Novell DOS 7.0

#### SYSTEM UTILITIES

- ✦ Hewlett-Packard Dashboard 2.0
- For Windows
- Central Point Software PC Tools
- For Windows
- Symantec Norton Desktop 3.0
- For Windows

#### APPLICATION DEVELOPMENT

- ✦ Microsoft Visual C++
- ✦ Borland C++ 4.0
- Microsoft Visual Basic 3.0

#### MULTIMEDIA & GRAPHICS 256

#### MULTIMEDIA SOFTWARE

- ✦ Adobe Premiere
- ✦ IBM OS/2 2.1 Multimedia Presentation Manager/2
- Macromedia Authorware Professional 2.0.1

#### MULTIMEDIA HARDWARE

- ✦ Pioneer CD-ROM Jukebox DRM-604X
- Pinnacle Micro RCD-202 PC
- Sigma Designs ReelMagic

#### PAINT & DRAW SOFTWARE

- ✦ Virtus Walkthrough
- For Windows 1.1.1
- Adobe Photoshop 2.5
- CorelDraw 4
- Micrografx Picture Publisher 4.0

#### EDUCATION & ENTERTAINMENT

- ✦ Virgin Games *The 7th Guest*
- ✦ Broderbund *Arthur's Teacher Trouble*
- Microsoft Musical Instruments*

#### INNOVATION 268

#### HARDWARE INNOVATION

- ✦ Intel Pentium Processor
- Cyrix Cx486SRx<sup>2</sup>
- Intel Energy Star PC

#### SOFTWARE INNOVATION

- ✦ Microsoft OLE 2.0
- Lotus Smart Suite 2.1
- SunSelect Wabi

# BUSINESS APPLICATIONS

sued only programmers. Well, all that's changed: This year we saw software vendors introduce more than a dozen powerful but easy-to-use Windows databases. In fact, many of them are as easy

to use as word processors. The notable omission was a Windows version of dBASE from Borland.

Borland's Paradox and Microsoft's Access strove to

outdo each other. While Microsoft was introducing Access and its set of Wizards, Borland was tuning Paradox's InterBase engine. That accounts for the blazingly fast

perfor  
for l  
Win  
har  
our  
Win  
mas  
pow  
Def

SP  
If y  
dat  
load  
in F  
for  
Exc  
ahe  
con  
no:  
tra  
a g  
few  
tha  
the  
to r

Er  
job  
the  
spr  
tha  
righ

## WORD PROCESSORS



### WordPerfect 6.0 for DOS

WordPerfect 6.0 for DOS might single-handedly keep DOS alive for the next few years. With most of the usability advantages of Windows (Wysiwyg layout, scalable fonts, and a graphics interface), it still runs briskly on the underpowered PCs that haunt most offices. If this were Windows, we'd be impressed. In DOS, we're floored.

WORDPERFECT 6.0 FOR DOS • \$495 • WORDPERFECT CORP. • 1555 N. TECHNOLOGY WAY, OREM, UT 84057 • (800) 451-5151

CIRCLE NO. 740 ON READER SERVICE CARD

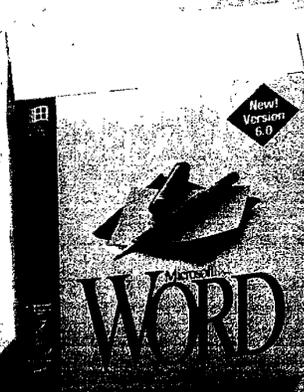


### WordPerfect 6.0 for Windows

WordPerfect 6.0 for Windows is the Carnac of word processing. With the trophy. Overcoming usability problems, this WordPerfect has the Windows look. With right-mouse-click menus, ingenious file management, animated help, and Express Docs for automatically filling in forms, WordPerfect is power and ease of use together.

WORDPERFECT 6.0 FOR WINDOWS • \$495 • WORDPERFECT CORP. • 1555 N. TECHNOLOGY WAY., OREM, UT 84057 • (800) 451-5151

CIRCLE NO. 741 ON READER SERVICE CARD



### Microsoft Word 6.0 for Windows

Microsoft Word 6.0 for Windows is the Carnac the Magnificent of word processing: It reads your mind. It doesn't have to ask how a document should be formatted. You don't have to keep telling it that *which* really means *which*. It automatically formats your text and corrects your typos. Add 100 levels of undo, animated help, and workgroup revision marking, and you've got one of the most intelligent word processors you can find.

WORD 6.0 FOR WINDOWS • \$495 • MICROSOFT CORP. • 1 MICROSOFT WAY, REDMOND, WA 98073 • (800) 485-2048 • (206) 882-8080

CIRCLE NO. 745 ON READER SERVICE CARD

performance of Paradox 4.5 for DOS and Paradox 4.5 for Windows. It's not like they're hard to use, either. In particular, our usability tests of Paradox for Windows found it to be easier to master than Access. Who says power must equal complexity? Definitely not us.

#### **SPREADSHEETS**

If you thought that the new databases were savvy, get a load of these. Build a chart in Borland's Quattro Pro 5.0 for Windows or Microsoft's Excel 5.0 for Windows. Go ahead—give it a try. Like the commercial says: No muss, no fuss. Wizards and Experts transform your boring data into a graph extraordinaire with a few mouse strokes. Really, it's that simple. All you do is tell the spreadsheet that you want to make a chart.

**E**nhancements that do the job for you are only part of the improvement. Today's spreadsheets employ metaphors that mimic how you work. That's right: You don't have to think

like a spreadsheet—the spreadsheet will think like you for a change.

**A**ll three finalists in this bunch adhere to the familiar workbook metaphor, where your spreadsheet is designed like a stack of papers that you'd use for budget tracking or analysis. You can easily maneuver between worksheets with tabs along the top or bottom of your screen—just like flipping through the pages of a ledger. Excel and Quattro Pro include powerful and easy-to-use scenario managers that let you save different "what if" configurations of data as you perform your analysis.

**O**ne more thing: Usability wasn't the only breakthrough this year. Perhaps best of all, just when we thought that prices had dropped to their all-time lows, Borland blew our minds by reducing the price of Quattro Pro to \$49.95. Talk about practically giving away software—that's around one-tenth the list price of its competitors.

#### **WORD PROCESSORS**

And speaking of ease of use, word processors were among the first applications to enter the salon for a usability make-over. No surprise: Think about how many hours you spend using a word processor—if anything needs more usability, these products do.

**P**ower through accessibility is the theme behind Microsoft Word 6.0 for Windows, WordPerfect 6.0 for Windows, and WordPerfect 6.0 for DOS. Both Windows word processors have context-sensitive right-mouse support for easy formatting. But that's only the start. Don't like the way your word processor looks? Change it: Customizing your work space has never been easier. All three finalists offer a host of options for configuring toolbars.

**M**icrosoft in particular sets the direction with products that do the dirty work for you. Look at Word 6.0's AutoCorrect, which catches and corrects typos and misspellings on the fly, and AutoFormat, which takes the ugliest e-mail, kisses it, and transforms it from a frog

**PX 113**



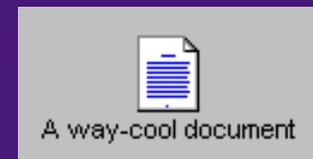
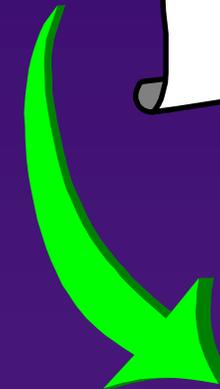
# New Windows™ “Chicago” UI: What It Means For Your Application...

**Joe Belfiore**  
**Program Manager**  
**“Chicago” Shell/UI**  
**Microsoft Corporation**

---

# A Change In Model...

- ◆ **Document-centric and object-oriented**
  - Users should begin to deal directly with objects (files) that your applications create, using common actions
- ◆ **Greater consistency between applications**
- ◆ **Greater integration among applications and the shell**
- ◆ **Simpler to learn, easier to use**



# How Do I Get There From Here?

- ◆ **We're providing new controls you can (and should) use**
- ◆ **We're enhancing common dialogs**
- ◆ **We're making it possible for you to extend the shell**
- ◆ **Guidelines about what you should do first**

# New Controls

- ◆ **Toolbar**
- ◆ **Status bar**
- ◆ **Column heading**
- ◆ **Slider**
- ◆ **Spin buttons**
- ◆ **Progress indicator (gas gauge)**
- ◆ **Tabs**
- ◆ **Property sheet**
- ◆ **Rich text control**

# New Controls - 1

## Toolbar

- ◆ Will support button wrap, so you can make it float, dock, etc.



## Status bar



## Column heading

- ◆ Resize columns by dragging
- ◆ Sort by clicking the desired column

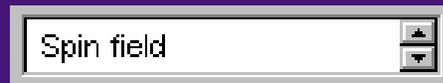
Name	Size	Type	Modified	Created	Attrs
 256color.bmp	6K	Paintbrush Picture	10/01/92 03:11AM	10/01/92 03:11AM	A

# New Controls - 2

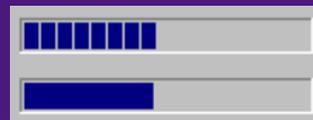
## Slider



## Spin buttons

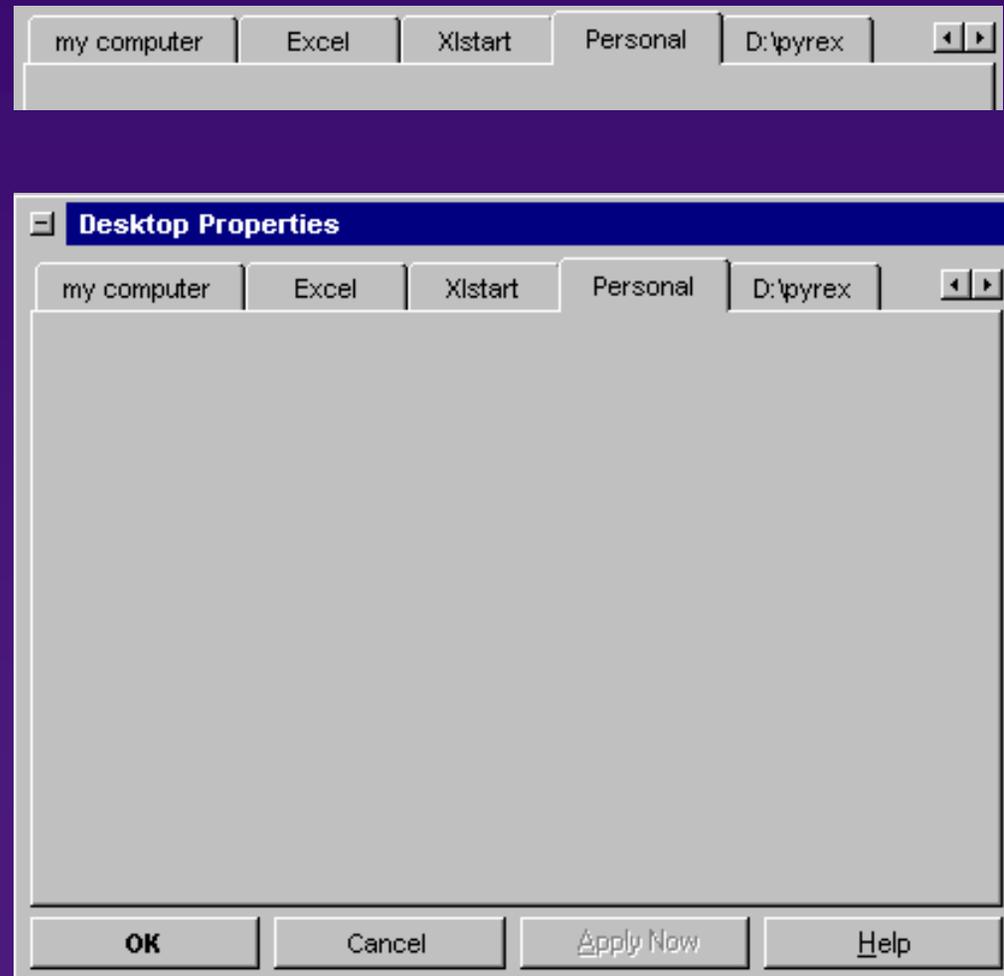


## Progress indicator



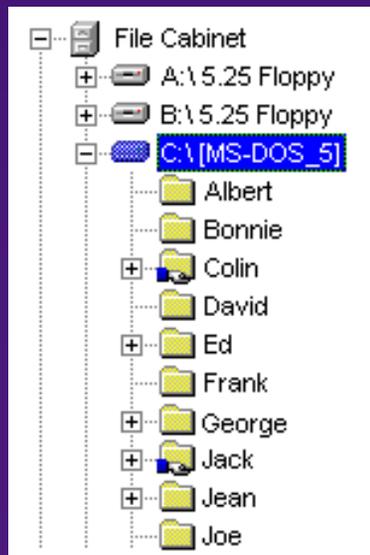
# New Controls - 3

- ◆ **Tabs**
- ◆ **Can set the maximum number of rows to be displayed**
- ◆ **Tabs can scroll horizontally**
- ◆ **Property sheet**
- ◆ **You add pages as dialog templates**
- ◆ **Can add/change the buttons displayed across the bottom**



# New Controls - 4

- ◆ **ListView**
- ◆ **Supports large icon , small icon (these are positional), list, details views**
- ◆ **Useful for “custom container” implementations**



- ◆ **TreeView**
- ◆ **Use to display hierarchies - containers, outlines, etc.**
- ◆ **Can add your own icons, configure expand/collapse**

# New Controls - 5

- ◆ **Rich text edit**
  - **More than 64K of text**
  - **OLE 2.0 client support**
  - **Left, right, center alignment on a per-paragraph basis**
  - **Definable left tabs**
  - **Fonts (bold, italic, single underline, color)**
  - **Simple bulleting**
  - **Find and replace**
  - **Superscript, subscript, strikethrough**

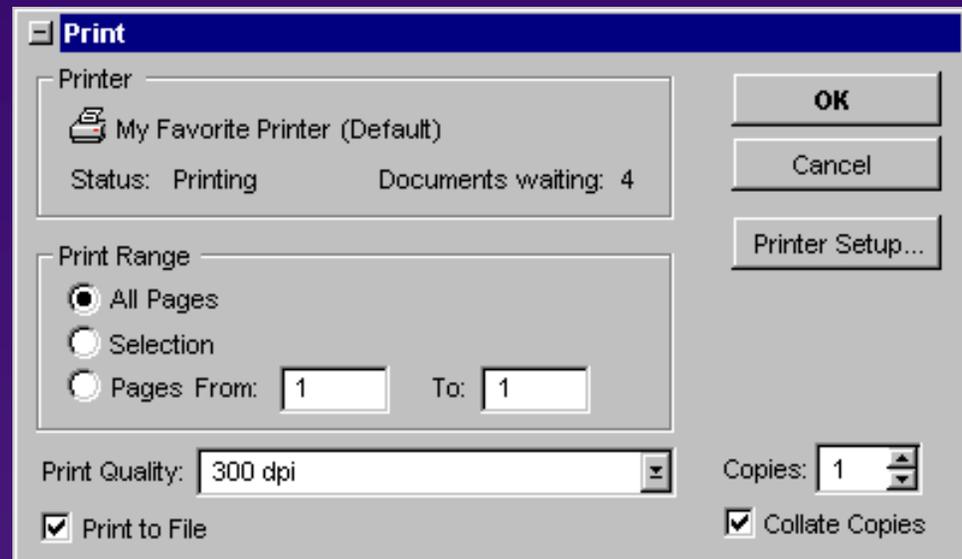
# Common Dialogs

- ◆ **File open and save as**
- ◆ **Print**
- ◆ **Print setup (choose printer)**
- ◆ **Page setup**
- ◆ **Find and replace**
- ◆ **Font**
- ◆ **Color**
- ◆ **OLE 2.0 dialogs**



# Print

- ◆ **Very similar to Windows 3.1**
- ◆ **Includes printer status information**
- ◆ **Hookable, as in Windows 3.1**



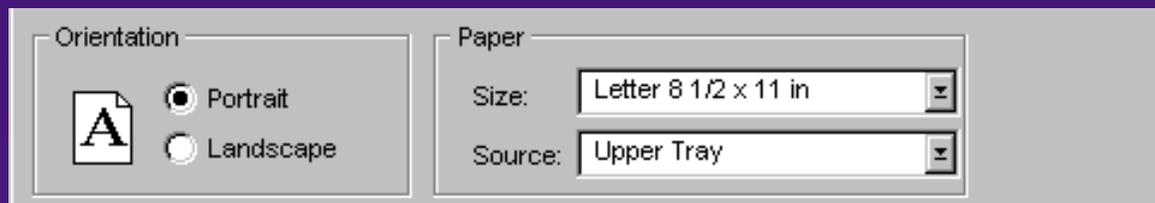
# Choose Printer (Print Setup)

## ◆ Use for printer selection UI

- Choices are all the printers in your printers folder
- Status info for each printer is shown

## ◆ Compatible with today's applications

- Landscape/portrait and paper tray choices are displayed for old applications (bottom section shown above)
- New applications can set these using the page setup dialog

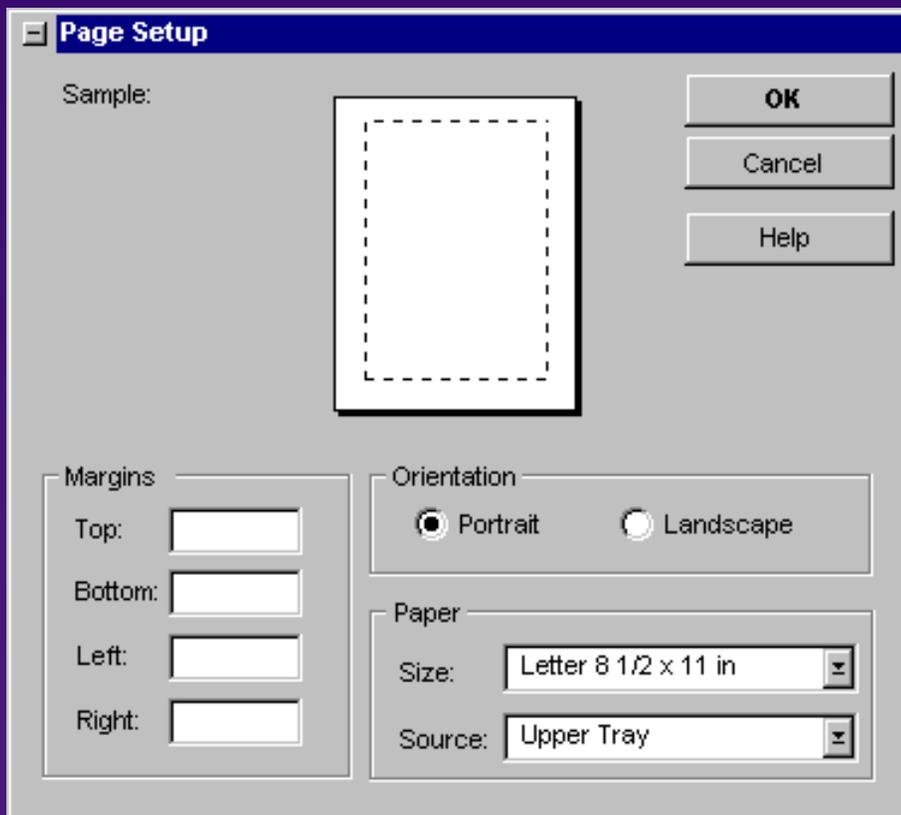




# Page Setup

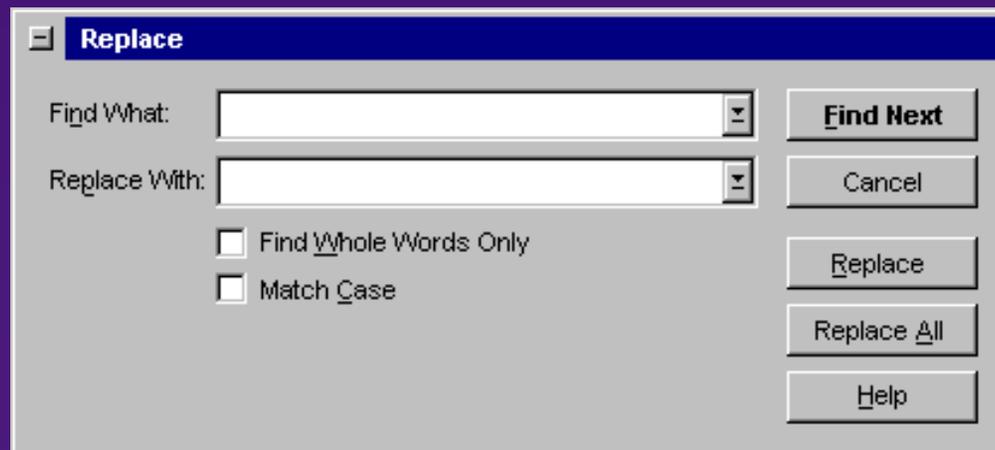
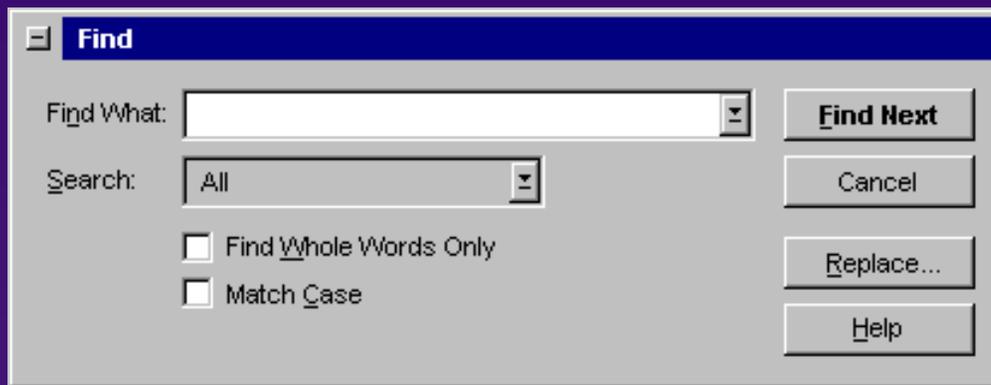
## ◆ General UI to set page layout

- Includes: orientation, paper settings, and margins
- Separates the old print setup dialog into printer selection and page setup components
- Available in property sheet mode



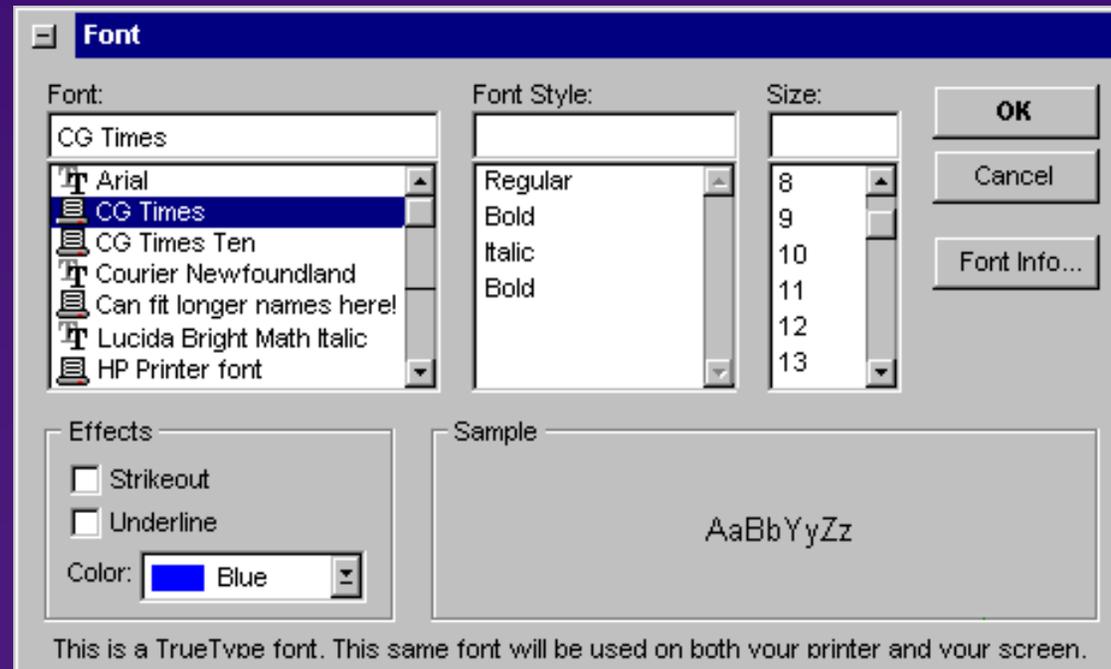
# Find/Replace

◆ Very similar to Windows 3.1



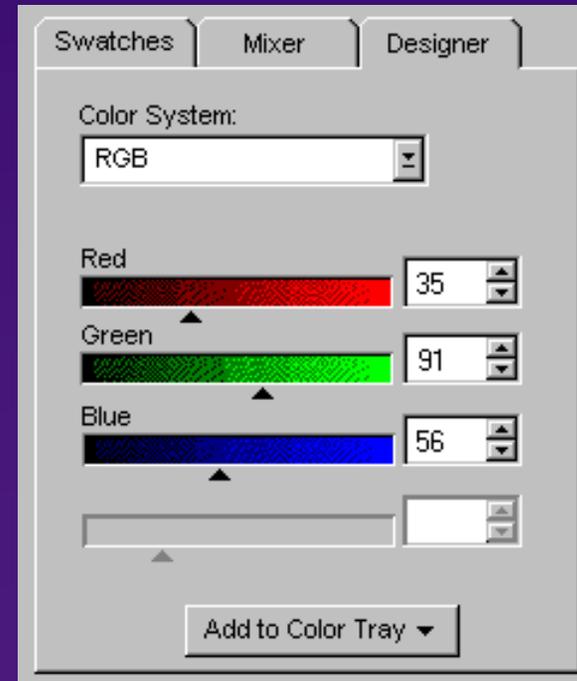
# Fonts

- ◆ **Very similar to Windows 3.1**
  - Adds “font info”
  - Also available in “property-sheet mode”



# Color

## ◆ Design still underway



# OLE 2.0 Dialogs

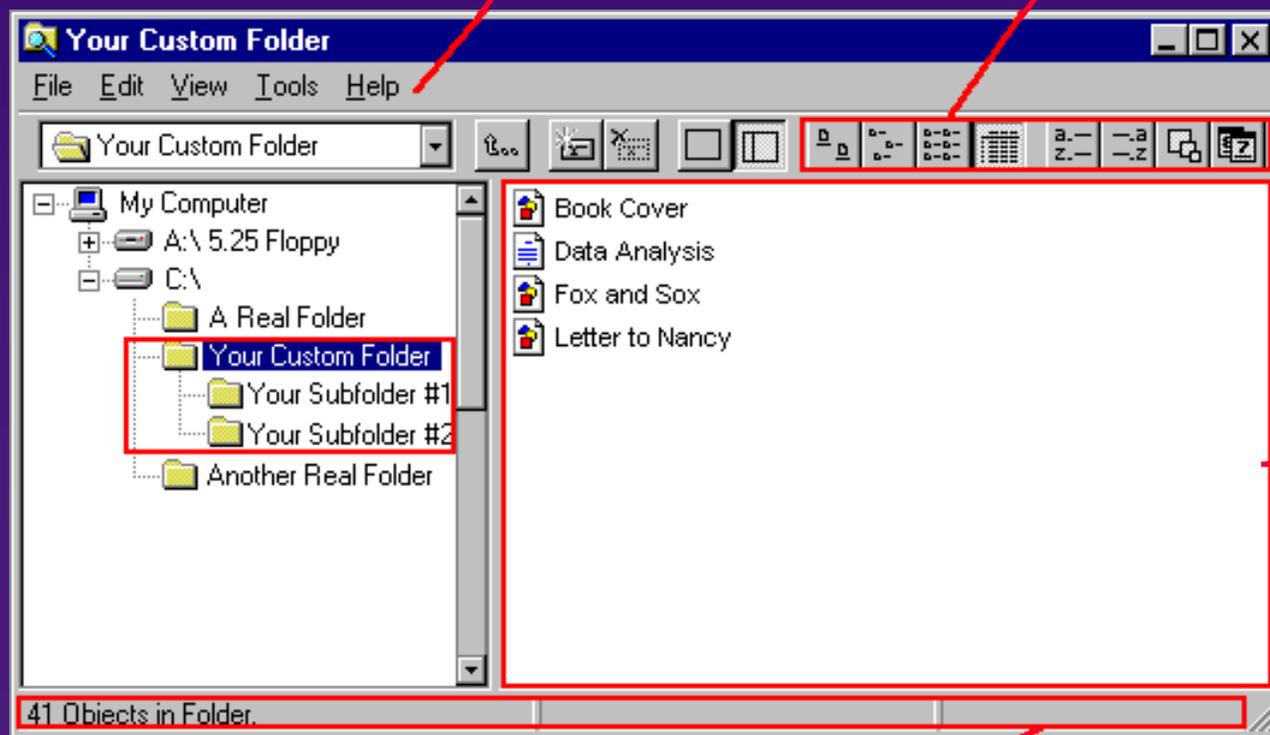
- ◆ We'll include all the OLE 2.0 dialogs, revised for look and feel of "Chicago":
  - Insert object
  - Change icon
  - Put here as/paste special
  - Convert
  - Links (link properties)
  - Object properties 
- ◆ Attend Randy Kerr's OLE 2.0 UI talk for more details!!

# Shell Extensibility

- ◆ **OLE 2.0 integration**
  - Drag-and-Drop feature
  - Summary properties shown by default
  - Verbs you add to the registry will be displayed on object menus
- ◆ **Property sheet extensibility**
  - Add your own property sheet pages to objects on a per-class basis
- ◆ **Viewers**
  - Hook your own file viewers directly in to the shell
- ◆ **Explorer UI integration**
  - *If you have an application that displays a collection of file-like objects, you can create your own “custom container” displayed in the folder/explorer hierarchy*

# Explorer Integration Details

- ◆ **Not for most applications!**
  - Only should be used if your application displays a pseudo-folder: electronic mail, document management, etc.
  - Users should **NOT** edit documents with an explorer extension!
- ◆ **How it works:**



Customize status bar

Microsoft Confidential

# What About MDI?

- ◆ **MDI is ok**
  - **Make MDI behavior as consistent and non-mysterious as possible in the shell**
  - **Think about evolving away from MDI - read the style guide!**

# Summary: How To Build A Great Windows “Chicago” Application UI

- ◆ Win32
- ◆ OLE 2.0
- ◆ “Chicago” UI style
- ◆ Plug and Play
- ◆ Shell integration

# 10 Keys For Making A Great Windows “Chicago” Application UI

1. Support long filenames
2. Support UNC path
3. Make sure your documents/data files are accurately displayed and used in the shell, with multisize icons, verbs, and no visible “.3” extensions
4. Use the commdlg “File Open”, or recreate its namespace accurately, including network browsing and links
5. Support OLE 2.0 Drag-and-Drop feature and the system menu-based transfer model consistently and extensively

# 10 Keys For Making A Great Windows “Chicago” Application UI

6. Be careful about multiple instances of your application being started *too easily* at the same time
7. Maintain a consistent user interface and object paradigm between your application and the shell. (Use right-click menus, prop sheets, our controls, etc.)
8. Support pen input for pen notebooks and desktop tablets
9. Support “Chicago”-style Help. (WinHelp authoring style, context-sensitive Help)
10. Make sure your visuals don’t break in “Chicago”! (3-D color, window metric changes, maximize to the desktop toolbar, not behind it)

**PX 114**

**Erik Stevenson**

---

**From:** Brad Silverberg  
**To:** davidcol; dennisad  
**Subject:** RE: mid month status report  
**Date:** Thursday, December 02, 1993 1:32PM

-----  
**From:** davidcol  
**To:** bradsi; dennisad  
**Subject:** RE: mid month status report  
**Date:** Monday, November 29, 1993 5:07PM

- Where do we stand in the great shell extension debate,  
on coming up with api's we and cairo can live with?

= > Satoshi has been busy for the past month working on new interfaces that  
cairo will support. Total he'll spend about 6-8 weeks on this stuff, he's  
about 1/2 way done now.

good

- Will File Search go into document stores? There really  
should be an api so searching can continue into a document  
store. By document store, I mean things like a mapi store,  
where someone registers to do right pane drawing for its  
contents. We want such document stores to be as seamless  
as possible, and that includes passing down file searches,  
file opens, etc.

= > talked with the shell guys about this. we might be able to have a  
generic search criteria for common fields (like filename/message title,  
filecontents/message contents) so westart one search but return results in  
multiple Windows, one for mail one for files. We don't really have the  
time to do things in a generic way in Chico v1. We have no storage or  
search API, I think it's too late to tackle these. Let's talk when you get  
back.

thx  
||

- You should review the tape api and tape format issues  
with Bobmu and others in Cairo/NT (loup, jimall, etc).  
If the api's and format are as limited as johnpa's mail  
made it seem, this is a problem that needs to get fixed!

= > I still contend the tape APIs are bogus. They don't support formats  
other than MTF, yes the API is tied to the format. Also, adding a tape api  
isn't enough, you have to add all the lower level layers to make access to  
different device seamless for the app. There is little end user benefit to  
doing this stuff, let's not. Let's just get a great backup app rather than  
a backup operating system.

do the nt dudes know they screwed up tape so badly? i don't want tape api's so limited or tied to the  
format either. maybe we should work with them on tape api 2.0 that works with the impt drives and  
formats.

- What file format will Writepad use? Is it compatible with Word? Write? How will old write documents be viewed?

= > I meet with the Word guys on Thursday. The format will be T3, some people still think this is a debate, but it's not. WritePad will read old write files.

excellent

- I really want us to make sure our file formats for drivers, dils, vxds, have all the info we want and are designed with the unknown future in mind, ie, so that additional info can be added in an upward and downward compatible way. This includes things like mfg, product that installed, name, version, description, comment, etc etc etc. I'm sick of haivng all these stupid files on my disk and having no idea where they came from. I should be able to just view it and it will tell me, getting the info from the file (and not requiring some database and code that divines such stuff).

= > Yes, I agree. We don't have a good plan here yet. Dennisad and I will drive this forward and get an owner.

good. we have a real opportunity here to make life immeasurably better with just a little effort. we shouldn't let it slide. please let me know who will own.

**PX 117**

# WordPerfect for Windows "Eliot"

## Marketing Requirements Document

January 4, 1994



**Signoff review sheet**

## Corporate

President/CEO	Ad Rietveld
Chief Technology Officer	Dave Moon

## Marketing:

V.P. Product Marketing	Mark Calkins
Director WPWin Marketing	Todd Titensor
WPWin Technical PM	Dave LeFevre

## Development:

V.P. Word Processing	Bruce Brereton
Director WPWin Development	Steve Weitzel
Manager WPWin Development	Dave Hallmeyer
Manager WPWin Development	Tom Freeman
Director Engine Development	Gary Gibb
Shared Code	Tom Creighton

## Other:

Coaches, Tutorial	Dennis Wilkins?
Templates, Macros	Bevan Wulfenstein?
WPDRAW	Shaun ?
Writing Tools	Dan Rask?
QuickFinder	Rodney Smith
Conversions	Lynn Merrill
Printed Documentation	Robyn Openshaw Pay
Online Help and Documentation	Susan Zeller

## Executive Summary

WordPerfect 6.0 for Windows has done many things to help establish WPCorp as a leader in Windows word processing. The product has been correctly positioned, advertised, and reviewed as the most comprehensive word processing program in the market. WPWin 6.1 (Eliot) will be release to the market soon and will have improved on the WordPerfect 6.0 release in funtionality, many features and processes have renovated. The bug file counts for Eliot are much less than when 6.0 was being readied for release and customers are more please with it reliability.

To continue this leadership trend and to respond to key efforts being made by the competition, the next version of the program, code named "Eliot," needs to focus on eight key objectives:

- **Improved speed and reliability:** WPWin 6.0 is universally praised as having great functionality, but was considered by the press and many users (in its initial release) as too slow for their current hardware and as compared to the competition and containing too many bugs to be considered sufficiently stable. WPWin 6.1 October 94 will improve on reliability but performance issues are althought have been midigated are not to the point where anyalist and user will praise us for that effort.
- **Ease of Use—"PerfectSense" Technology:** We need to focus on the common tasks of writing and editing and make them as easy and automatic as possible in Eliot
- **Great Chicago Application (Windows 95):** As a partner in Microsoft's First Wave Chicago program we have committed to be a well behaved Chicago application in several area's.
- **Best Network Integrated Application:** We should seek ways to leverage our Networking Roots and allow Eliot32 to be more that Newtork aware or enabled but extend ourselves to be Network Integrated.
- **User-responsive Interface:** Eliot should always provide appropriate and useful menu and button choices for the task at hand and needs to permit total customization options.
- **Comprehensive:** The labeling of WPWin as the most comprehensive word processor needs to continue into a positioning as the document processor of choice, containing word processing, spreadsheet and database abilities and support, scanning, e-mail, and more.
- **Modularity:** WPWin 6.0 cannot be run by many users because it requires too much memory and too much disk space. We need to midigate minimum install requirement
- **Integration:** Support for OLE 2.0 is critical, with OpenDoc and OBEX (in this still important) close behind, as well as suite issues such as a cross-product macro language. Visual Appware and other umbrella technologies.
- **New Technologies:** Grammatik 6, new equation editor, add-ins for speech control, dictation, and OCR, and support for network APIs and emerging distribution standards are all important elements in keeping and improving the image of WordPerfect as the technology leader. Open Doc, SGML, ALM Aware, ODMA, Product management help cost of onwership (possible from Novell like MS Hermes solution)
- **Transition and Learning Tools:** No audience is more critical to the future of WPWin and WPCorp than our large DOS user base, and Eliot must provide flawless conversions of files, macros, and other items, as well as for competing products; also, it is important to provide unmatched learning tools in the form of Tutorials, Coaches, and online feature demonstrations. Our competitors offer some better tranistions tools from WordPerfect 5.1 DOS to their Windows

applications. We need to be the undisputed transition choice and provide the tools to back that up.

- **Closer alignment with Quattro Pro:** The one two punch of MS has always been Excel and Word. We have to find ourselves closely aligned with QP. As we move to 32-bit hopefully our 32bit development in Shared Code and other areas QP can leverage in there 32-bit Chicago release.

The above objectives are all important directional priorities for the next release of WordPerfect for Windows and should be used in prioritizing resources during the development of the product and in product positioning for marketing, ads, and public relations.

## I. Introduction

WordPerfect 6.0 for Windows is the most comprehensive word processor available today. It moves WordPerfect past the definition of a word processor into document processing. It offers a great depth of functionality in its core code, as well as a host of modules for things such as drawing, charting, and text special effects. It supports advanced writing tools, data integration for both spreadsheet and database, and strong learning tools with an excellent help system.

The need for a new version of WordPerfect for Windows lies in the twin goals of matching and surpassing popular features from competitive products and continuing WordPerfect's role as the leader in word processing with new and innovative technologies. The next version of WordPerfect for Windows, code named "Eliot," needs to build on the document processing foundation and move the product forward in several key areas, including improved speed, reliability, modularity, comprehensiveness, ease of use, customizability and context-sensitivity, integration, new technologies, and transition and learning tools. Making our software smarter or smart software

- Eliminate redundancy
- Crystalize process
- Anticipate the user

After speed and reliability, the most important issue for Eliot is ease of use. While there are many possible changes to the product that will enhance ease of use, the cornerstone of this concept lies in an umbrella technology tentatively called "PerfectSense."

This document outlines the marketing requirements for Eliot, likely numbered WordPerfect 6.1 for Windows.

## II. New Release Objectives

There are eight major objectives for Eliot, each of which encompasses many potential product changes as discussed in the Product Description section below:

- Improved speed and reliability
- Ease of use—"PerfectSense" technology
- User-responsive interface
- Comprehensive
- Modularity
- Integration
- New technologies
- Transition and learning tools

These major objectives mirror some of the objectives for 16-bit Eliot but other are specifically targeted to be able to provide us the greatest pop and sizzle in our changing market.

- Improve speed and reliability (increase start-up time by 30%)

- A great Chicago Application (32-bitness, threading, explorer, user Interface specifications etc.
- Network Intergrated application.
- A great PerfectOffice component.
- Renovation and refinements to application processes.
- User responsive interface
- OpenDoc Integration
- Best Transition for WordPerfect for DOS users.
- New technologies
- Modularity
- Closer Alignment with Quattro Pro 32-bit
- Don't take away unnessarily features that DOS WordPerfect user may be acustom. (be careful)

Of these objectives, the most important internal goal is improved speed and reliability; without which the product can have the most wonderful functionality in the world and still fail. The most important public goal (as used in positioning, advertising, and other promotions) is ease of use, with the principle story encompassed in an umbrella technology called "PerfectSense," or the ability to streamline and automate the task of writing and formatting documents.

#### *Market Share Objectives*

>>Todd<<

#### *Revenue and Shipment Forecast*

For the fourth quarter of 1994, we are projecting unit shipments of 1,010,000. We anticipate fourth quarter revenue to total \$86,630,000 (based on an average ASP of \$86).

### **III. Market Perspective**

#### *Market Conditions*

>>Todd<<

#### *WPCorp Market Position*

>>Todd<<

#### *Competition*

There are two principle competitors in the Windows environment—Microsoft Word for Windows and Lotus Ami Pro for Windows. Of the two, the most important competition is Microsoft Word for Windows, currently at version 6.0. This product focuses on IntelliSense™, a collection of functions that help automate routine word processing tasks. The most significant of these tools from a technology

viewpoint is probably AutoFormat, which takes an existing document, analyzes it, and reformats it based on pre-defined rules. Included is a styles gallery that allows the user to select from any template and apply the styles from that template to the current document. Other aspects of IntelliSense include AutoCorrect to correct common spelling errors as you type, as well as, fix capitalization problems and replace straight quotes with typeset-quality Smart Quotes. AutoSelect is the third piece, allowing Word to automatically select entire words when a multi-word group is desired.

Lotus is currently shipping version 3.01 of Ami Pro with a new version planned for early in 1994. Other than an emphasis on group editing, little is known at this writing about the product. For more information on the competition, readers are referred to the detailed Product Comparison available from the WPWin Marketing team.

### *Industry Trends and Observations*

Windows 3.1 has been immensely successful, selling over 60 million copies. The next version of Windows, code named Chicago (Windows 95) is due to ship in the second quarter of 1995 and will combine the functions of DOS, Windows, and Windows for Workgroups, making Chicago a true operating system not a thing on a thing. It is anticipated that this upgrade will be extremely popular, adding 32-bit processing, a new file system, built-in networking, e-mail and scheduling, and a new interface built on OLE 2.0 and taking concepts from the Macintosh, OS/2, Motif, HP OpenLook and others. It is critical that WordPerfect Corp. have a version of WPWin that is coded for the 32-bit Chicago release within no more than a two to three months of Chicago's ship date. Although many of the initial market research indicated that upwards of 50% of Windows users would upgrade to Chicago within the first year. These estimates have been since been severely scaled back to around 10 or 20% within the first year after release. Windows 3.1 16-bit WordPerfect will still provide us the predominate revenue during 1995 and into 1996. We are contractually committed to release a Chicago release of WordPerfect 90 days after Chicago ships.

The other important trend is towards suites of products. Microsoft sells Office, currently at version 4.0, which includes Word, Excel, PowerPoint, and a license for MS Mail. Another version, called Office Professional, adds their Access database. Lotus has their SmartSuite, which contains Ami Pro, Lotus 1-2-3, Freelance, Approach, and Organizer. WordPerfect currently ships in a suite called the Borland Office 2.0, containing WPWin, Quattro Pro, and Paradox.

Microsoft claims that 60% of their application sales come from the Office suite and are putting their marketing efforts behind Office rather than the individual applications themselves. Some analysts maintain that Microsoft is trying to make applications merely features of the suite, relegating them to simple checkoffs in a total data processing solution. Certainly the inclusion of networking, e-mail, scheduling, fax support, and other abilities in the current Windows for Workgroups and the upcoming Chicago are indications of the trend in this direction.

The two things WordPerfect must do in this area are: 1) Have a strong suite message and product offering, building on what the Borland Office has started, with WPWin as the keystone in that suite of software programs; and, 2) Position WPWin as a serious and essential stand-alone application, providing the best tools for all document processing needs. These two positions are not contradictory, but complementary, because we know that word processing is the most important application in the suite and

is used by more people for more of their work than any other piece of software, and that many suite purchases are made on the condition of best-of-breed as well as price and interoperability. Positioning the best document processor at the core of a powerful suite gives users inclined to purchase in that manner a strong motivation to go with WordPerfect's suite, while positioning and promoting WPWin as the best stand-alone product can keep Microsoft and anyone else from relegating word processing to a simple checkoff item.

Finally, the most important trend in product development is the move away from features and towards usability and ease of use. This is not to say that powerful features are not expected from future word processors, but rather that buyers perceive that today's programs are all on a virtual parity in terms of their functionality, so they are looking for differentiation in ease of use, automation, and other interface improvements. In order for WPWin to be recognized as a market leader and to improve its sales, it is critical that this ease of use focus be the primary factor in determining all improvements and changes made to Eliot32. WordPerfect 6.1 16-bit has delivered in this area sustainably there are many more areas that can be improved for Eliot32.

Since Chicago is Microsoft's major thrust for its future growth and revenue it is quite likely that any Microsoft applications for Chicago will try to embrace feature and capabilities to create pull to the new operating system. Microsoft released Word 6.0 for Windows 3.1 in November 1993 and Word 6.0 for Windows NT 3.5 will likely ship November 1994 and includes no new features except for threading in printing. If Chicago ships in May of 1995 it is quite probable that within one and a half years that Microsoft will have had since the initial release of Word 6.0 they will likely have a release of Word for Chicago with a clump of new capabilities that will create pull to the new operating system.

#### **IV. Target Markets**

##### *Target Users*

The primary target for WPWin is the large installed base of WordPerfect for DOS users, approximately 9 million in number. A large percentage of them are quickly moving to Windows, and it is critical to the success of WPCorp that they continue to choose WordPerfect as their word processor as they make that move. Secondly, we have to target WordPerfect 5.1/5.2 for Windows users for the upgrade to 6.x. Users of competitive products are a tertiary target in the U.S., but the primary target in many other countries where Microsoft Word is the dominant product. Finally, we must target new Windows and new computer users, especially through OEM bundling and other emerging distribution methods. Virtually every new machine sold today comes pre-loaded with Windows and many hardware manufacturers also include application software. It is important that WordPerfect at least be a choice in this arena, if not the preferred choice. Additionally, new and emerging distribution options, such as multi-product, key-locked CD-ROM and PCMCIA should be fully exploited, with WordPerfect at least as a choice if not the primary option.

### *Target Systems*

The target system is broadly stated as any Windows-capable machine. However, because of the large number of users running Windows on older, slower machines and their need to use them for the life of the depreciation schedules, it is critical that Eliot exhibit a performance improvement over WPWin 6.0 on low-end 386 systems with 4MB of RAM.

Chicago although initially purported to run equally or more efficiently in 4 megabytes than Windows 3.1 is now being positioned to not be capable of any applications work with anything less than 8 megabytes

### *Target Channels*

>>Jeff<<

## **V. Product Description**

### *Product Requirements*

The eight major objectives are listed below with some details about changes required in Eliot to meet each objective. This list is not meant to be comprehensive, but rather to highlight major points of emphasis and give direction and priority. Detailed feature lists and design reviews will flesh out the details and determine the final implementation for the ideas presented here.

#### **Improved Speed and Reliability**

WordPerfect 6.0 for Windows made some improvements in core product speed over 5.2, but the program also grew in size and complexity, making most of the speed gains minimal, or even offsetting them. WPWin 6.0 is usable on high-end machines with adequate memory (6MB or more), but our own benchmarks, the general feeling of users, and feedback from the press and analysts is that the product needs to be faster on existing 386 hardware.

Improving the speed of WordPerfect for Windows needs to be the major development goal in the most commonly used parts of the program, such as file open and save, scrolling and other navigation, using documents with columns, tables, graphics, printing, macro compilation and playing, and startup time. This is obviously a focus that should be kept in mind at all times when developing and testing the product, but should also be something that receives significant and specific attention and resources in the product development cycle.

The target hardware for our speed benchmarks and improvements needs to be a low-end machine, e.g., a 386 with 4MB of RAM. While the benchmarks should also tell us how we perform on mid-range (386-33 with 6MB) and high-end (486-50 with 8MB or more) computers, the low-end is the most critical target to satisfy as many users have these machines and plan on keeping them for the life of the depreciation schedules, another one to three years. Every effort should be made to improve Eliot's performance in the 4MB environment.

Even more important than the speed of the product, early feedback on WPWin 6.0 was that the product was too unstable to use reliably for many of our users. This has been brought out in the press and through our customer feedback via Support lines and CompuServe. This judgment has been made publicly and

privately by enough people that many of our customers have determined to wait until the first interim ships (due out next February) before upgrading.

We cannot afford to have Eliot go out in a similar state. Every effort must be made through thorough manual and automated testing to identify, properly prioritize, and fix bugs that will impact product perception and performance in this way.

### **Ease of Use—"PerfectSense" Technology**

WordPerfect 6.0 for Windows made great improvements in ease of use with things such as preview windows, Coaches, and Feature Bars. Usability and ease of use are the most important trends in software development today, and must be continued with the highest priority in Eliot.

The principle focus of ease of use in Eliot must be on common tasks, the things that most users do most of the time in their word processor, such as entering, editing, and formatting text, opening and saving files, printing, creating bulleted lists and tables, headers and footers, and spell checking. This priority on ease of use must be kept in mind, even to the point of dropping work on features or enhancements that appeal to a narrow range of users in favor of automating and streamlining more common document creation and editing tasks, if need be.

While Coaches should continue as the industry's leading instructional tool (see Transition and Learning Tools below), as a critical part of our product's ease of use strategy, Eliot should include Advisor technology based on "fuzzy" logic—the ability to select from a range of options rather than a binary yes or no choice—with the program asking minimal questions and then automatically completing the task for the user. This concept should be used to not only automate filling in specified fields in templates, but to help select document settings with templates (how many columns in your newsletter?; what should your closing say?; do you want a header and a footer?), perform formatting tasks on existing documents and tables, and even add document content using content-building tools (brainstorming and expert suggestions) and boilerplate information where appropriate for standard letters and forms. Creating custom automated templates and editing existing ones should be a strong focus, with Eliot providing simple and automated tools to speed and greatly simplify the process.

The main mechanism of ease of use in Eliot lies in the implementation of several separate (but somewhat overlapping) technologies falling under the possible umbrella term of "PerfectSense." The concept behind "PerfectSense" is to streamline and automate the task of writing and formatting documents. There are at least eight aspects to this strategy, as follows:

- **QuickCorrect**—On-the-fly spelling and formatting corrections. QuickCorrect checks for misspelled words while typing after the user inserts a word boundary character (space, comma, period, hyphen, colon, semicolon, etc.) from a replacement list. This should provide the option to either automatically replace the word(s) or offer a replacement list from which to choose, as the WPWin spell checker does now. The replacement list should be closely integrated with spell and grammar checking, perhaps even making it an additional supplemental dictionary recognized by those programs so that the user could achieve the same results with QuickCorrect or by spell checking. Other items QuickCorrect could monitor and change include irregular capitalization, all LRS terms for the selected language, and single and double quotation marks (SmartQuotes).

- QuickSelect—As an extension of Autocode Placement, QuickSelect could intelligently apply attributes, such as bold or italic, and other functions, such as index marking, to words when no text is selected. Also included would be drag-and-select functionality that automatically selects entire words once the selection crosses a word boundary.
- AutoGenerate—Automatic creation of table of contents and index through analyzing the document for likely phrases and words. This uses logic from the Format Advisor (see below) and linguistics technology from our Writing Tools division for analyzing and determining key words and phrases in paragraphs.
- Format Advisor—Analyzes the current document and applies styles for basic formatting improvements, as well as applying SmartQuotes, replacing two hard returns with paragraph spacing, replacing multiple space characters with tabs, adding bullets to lists of items, and more. The Format Advisor would also include a Styles Gallery that would allow the user to select from a list of available templates and apply the styles in those templates to the current document. This should be interactive formatting with the document itself using a Feature Bar as the gallery tool, but with the option to view a “dummy” document or bitmap image, displaying the available styles for any given template in case the current document does not have sufficient text for a useful preview. As with all advisors, a fuzzy logic interface should be provided to narrow the list of available templates and styles based on user criterion, and include the ability to save the selected settings by name for future use. Also included with the Format Advisor is shrink-to-fit capability, to change the font size and/or margins of a document to shrink it to the last full page.
- Table Advisor—Allows the user to select from pre-defined formats to improve the look of tables. The Table Advisor would include a Tables Gallery that would display a list of available table styles as well as a fuzzy logic interface to narrow the list and save the selected settings by name. The format changes would be interactive in the current table, but with the option to view a sample as needed.
- Template Advisor—Allows the user to select from pre-defined formats when creating a new document. The Template Advisor would include a Template Gallery that would display a WYSIWYG view of all templates including an option for “dummy” text to show them in completed format as well as a fuzzy logic interface to narrow the list and save the selected settings by name. This view could also be in the form of pre-defined bitmaps for the templates, though this option offers many challenges in terms of editing existing templates, displaying custom templates, etc. The Template Advisor should be closely tied to the Format Advisor and Styles Gallery. The final piece of the Template Advisor would be for it to help users actually write some of their documents by asking a series of questions about audience, purpose, tone, main points, etc., then creating boilerplate text to match the information provided. This text could be in the form of an outline or as full body text, depending on the template and the complexity of the information. Initially, this would probably apply only to letters and memos, but should later include other document types such as reports, proposals, etc. This is a key area where Eliot could move ease of use well past the competition.
- Task Advisor—This Advisor would monitor a user’s actions and offer advice on how to do it easier and/or faster. Advice offered could include formatting options, such as tabs instead of

spaces or paragraph format instead of two hard returns, as well as step-saving help, such as promoting QuickMenus or Button Bars. The interface should include a non-obtrusive optional means of alerting the user when a tip is available, such as a button on the Power Bar that changes appearance (light bulb lighting up) when an action has been performed for which there is a tip. There should be a very close tie-in with Help and Coaches for more information on the presented hint.

- **Advanced Morphological Tools**—Though this needs a better name, it includes functionality such as an intelligent search and replace that checks for all verb forms, improved word searching using QuickFinder, and the new Grammatik 6 tools that actually correct your writing, not just offer information about what might be wrong. Grammatik and spell checking need to be closely tied so both operations can be performed in one pass through the document, and both should be linked in a simple manner with QuickCorrect.

This ease of use strategy requires that Eliot make some basic assumptions about what the user is doing and act accordingly. These assumptions need to be based on usability data and include a mechanism to easily and obviously override them, if desired. By having WordPerfect perform in this way, the product can satisfy most people's needs most of the time. The overriding design principle is anticipating what the user desires and acting accordingly.

Ease of use also includes consolidation and consistency. For example, WordPerfect 6.0 for Windows has at least three different ways to create and use mailing lists, none of which are compatible with each other. This should be consolidated into one method. Also, WPWin 6.0 is not entirely consistent in the presentation of options, such as prompting for a filename. In some dialogs, a list of files is presented, while in other dialogs the users must first click on an icon to get to the list. Consolidation and consistency should also include an examination of all the bars (Button, Feature, Power, Status, etc.) used in WPWin. It might be desirable to have one consistent interface (such as the Button Bar with the added ability to use combo boxes and pop-out lists and improved context-sensitivity) for all buttons, with the ability to display multiple bars at once, or at least to consolidate some of those elements (Button Bar and Power Bar, for example).

While common tasks need to be the main focus of ease of use, simplifying complex tasks should remain an important second priority. Complex tasks should be defined as those items that are important to the format and content of the document but are not employed by the user on a regular basis, and those tasks that offer so many options and levels of complexity that they are frequently avoided by users who could actually benefit from their use. This list could include mail merge, complex tables, lists and indices, keep text together, typesetting capabilities, text and graphic styles, outlining, and setting WPWin defaults (something that should probably be a simple task, but is quite complex because of the many options, locations, and ways these defaults can be changed).

Coaches and Advisors should probably be the principle means of accomplishing these goals, though the developers of these program areas should do usability to simplify the interface wherever possible. For example, merge was greatly improved in WPWin 6.0, but could be made even easier by providing more feedback in the merge dialog (have I selected a data file yet and what is it?) and by walking the user through the process even more (for example, when I close the data entry dialog, prompt to have me create or select my form file).

Finally, the interface itself should be reviewed for ease of use issues. This could include:

- WYSIWYG preview of changes using actual document text wherever possible
- Explanatory text and indicative icons/bitmaps in dialogs, such as explaining what each option will produce in the Paste Special dialog
- Direct manipulation of all document objects (equations, layered graphics, envelopes, bookmarks, more)
- Eliminate the need for Reveal Codes with options to display non-formatting functions (bookmarks, index and ToC marks, etc.) in the document text itself
- Short help prompts that appear directly over buttons and other interface elements
- Remembering the last-used choices and offering them at the top of lists; we do this for files now, but should extend it to fonts, macros, sort, find/replace, templates, and any other place the user can select from a list of choices

### **User-responsive Interface**

WordPerfect 6.0 for Windows' interface received praise as a huge improvement over our previous version as well as the competition in many aspects. Button Bars with text and icons, the Power Bar for common tasks, QuickMenus—all of this and more contributed to the impression of a Windows-like and user-responsive interface. In addition, some capabilities were included but not fully exploited, such as context-sensitive Button Bars and Feature Bars.

Eliot needs to complete the thought begun with 6.0's interface. Consolidating interface elements, as discussed above under Ease of Use, is a part of it. More important, however, is taking advantage of our context-sensitive abilities and decidedly modifying the interface for the task at hand. Eliot should anticipate what the user would like to do next and provide appropriate choices based on that context. Invalid items should be removed from menus and Button Bars, and a greater depth of choices offered where appropriate. For example, if the user is not editing a table, most of the Table menu choices are grayed out in WPWin 6.0. Why not remove them until a table is entered or created, then display them. We should take full advantage of context-sensitivity in Button Bars, especially in relation to specific templates, and provide a straight-forward interface for setting up those associations.

In WPWin 6.0, virtually every interface element can be edited. The exception, however, is QuickMenus. In Eliot, these should be editable, perhaps using the same interface as editing the pull-down menus. Also, the options on the Button Bar and menus should all have "Properties" dialogs to allow editing of the action of the buttons. This is especially important for macros.

Probably the most important thing to do in Eliot along these lines is to take full advantage of Feature Bars. These dialogs-disguised-as-bars can display with a host of features replacing many dialogs that we use today, allowing for more direct manipulation in documents and a great means of making the right tools available when they're needed. Eliot should be able to display multiple Feature Bars at once, probably with the option of some limited editing. Features to take advantage of this could possibly include Font, Drop Cap, Master Document, Paragraph Format, Find/Replace, Power Fields, Sound, Border and Fill options, and Kerning.

### **Comprehensive**

WPWin 6.0 was positioned, advertised, and rightly perceived by press and users as the most comprehensive word processor available today. This focus and positioning needs to continue into a position as the document processor of choice, making WPWin the ideal container for all your data and document publishing needs.

This comprehensive positioning is supported by all the current functionality of WPWin 6.0 (e-mail, spreadsheet capabilities in tables, spreadsheet and database import and link, etc.). Eliot can supplement the emphasis here with TWAIN support directly from WPWin, spreadsheet export, and possible improved database publishing abilities, such as using styles to format fields as they're imported, combined query and sort, conditional record import, and more.

### **Modularity**

WordPerfect 6.0 for Windows takes over 32MB of hard disk space for a full install. Perhaps more significant, it requires over 10MB for a minimal working version. In addition, it demands 6MB or more of RAM for adequate performance.

For many users, these requirements exceed the capabilities of their available hardware. Some of our large accounts have told us that they must consider the competition for a large number of their older machines that will not be replaced for several years. There is also a growing number of subnotebook computers that fit into the minimal machine category in hard disk space and/or RAM, though they may have adequate CPU speed.

To help address this problem, Eliot needs to have as many parts broken out of the main EXE as possible and put into loadable modules. While this is a challenging task, given the design of 6.0 and the planned time frame, any effort we can make in this area will help us from losing potential customers. Development will have to ultimately provide the list of what can and cannot be broken out, but possibilities could perhaps include equation editing, generate, spreadsheet functionality in tables, and document compare.

The two goals for modularization are:

- Reduce the memory footprint of WPWin so it will run better in a 4MB environment
- Reduce the required disk size for a minimal installation to a target of 5-7MB and provide a standard installation with the most useful document tools using under 20MB

The Setup program should obviously be changed to reflect these changes. However, Setup could possibly be enhanced to provide common preset installation options, such as laptop/notebook, simple documents, and desktop publishing.

### **Integration**

Windows by its design encourages integration. Surveys show that Windows users have an average of seven applications they use consistently, many of them open at the same time. It's important that Eliot enhance WordPerfect's current strong abilities to work well with data from a variety of sources and that Eliot clearly establishes itself as the container of choice for multiple data types.

The most important integration focus must be with the other applications that will be in the WordPerfect Windows suite. It is critical that WordPerfect work with and look like the other suite applications, and all efforts must be coordinated with other development groups and with Borland to make this happen.

There are currently three important technologies for sharing of information: OLE 2.0, OpenDoc, and OBEX. The first, OLE 2.0, must be supported in Eliot. OpenDoc is important as WordPerfect's competing answer to OLE, and should be supported as soon as possible. Borland's OBEX, both publish and subscribe, should also be supported for suite compatibility.

ODBC support should also be included in Eliot for better integration with programs, principally databases, that follow that standard. This allows quicker support for new databases or versions of products without having to write our own conversion drivers each time.

Other areas of integration include spreadsheet and database export and improved import capabilities, perhaps with a front-end that allows for formatting data from spreadsheets and databases as it is imported using conditional styles and other formatting, as well as enhancing the current query operation with consolidated, real-time sorting.

Support for TrueType font embedding is also an important element of integrating with other users and applications. This should be optionally implemented for better transport and exchange of documents.

As part of integration, we should rework the Paste Special dialog and capabilities to fully support all the data types, provide thorough descriptions of the data coming in (DDE, OLE 1.0, OLE 2.0, linked, embedded, etc.), and preview the data itself. Full page OLE object support is also desired.

Work should also be done to support Netware Navigator and Netware License Server APIs from Novell. WordPerfect has announced support for both of these items in conjunction with Borland and we need to follow through on that announcement.

## **New Technologies**

Having WordPerfect for Windows demonstrate product leadership with new technologies is important to the future success of the product. Though it must match key features in competitive products, Eliot cannot be perceived as a "catch-up" product only, but must demonstrate WPCorp's ability to adapt, respond, and move forward in progressive and exciting technologies. Key points of this strategy include:

- Workgroup tools, such as multi-user, interactive editing of documents, tracking and viewing revisions by user, document routing, and document locking by section or object
- Document management tools with QuickFinder improvements and SoftSolutions technology, including a document log and a save workspace capability that allows for multiple named workspaces/projects
- Support for the Chicago interface, including tabbed diaogs, and file system
- Speech technology, including command-and-control and dictation capabilities (this would be an add-on but should be thoroughly tested with Eliot for compatibility)
- Linked text boxes for more desktop publishing abilities
- Support for pen computing with ink layer
- Installation routines for CD-ROM and network software distribution

### Transition and Learning Tools

Given that our principle U.S. market consists of previous users of WordPerfect products, it is critical that we provide a smooth transition to Eliot. This transition involves at least the following:

- The best file conversion for WP 5.x files anywhere
- An excellent macro conversion from WPDOS 5.1 and WPWin 5.x with clear debugging tools for those macros that require editing after conversion
- Conversion of keyboard layouts, PRS files, SET files, Button Bars and other interface elements, templates, abbreviations, address books, and any other customizable aspect of WPDOS and WPWin
- Transition kit to train the user on Windows in general and Eliot specifically
- Online help, including a Tutorial and Coaches specifically for WPDOS users

A key to this could be a tutorial that covers a host of product features in a task-oriented training approach, going well beyond what the current Tutorial does in 6.0 to include intermediate and advanced tasks.

### *International Requirements*

>>Randy<<

### *Target Release Date*

The following table outlines a potential 1994 deliverables schedule:

Product	Anticipated Release (Street) Date
WPWin 6.0a interim release	February 21
Eliot, 16-bit version or WPWin 6.0b interim release	August 5
Eliot, 32-bit version	October 14
WPNT 6.1	October 14

Note that the 16-bit version of Eliot due in August is only done for suite needs, with the principle goal being to implement OLE 2.0. Because of the time frame, it is not likely that the August release could have all or even most of the functionality outlined herein, thus it may end up being a 6.0b release supporting OLE 2.0 with little additional capabilities in order to not conflict with the goal of the full product as described above being released for Chicago in October.

### *Documentation Requirements*

Online documentation is an important consideration for Eliot, as it will contain the bulk of the product information. Due to the shrinking ASPs, our cost of goods must be significantly reduced; therefore, Eliot packages may only include one small printed manual (150-250 pages) that will be task-oriented.

Help files, coaches, agents, a guided tour, and a product tutorial are all part of online documentation.

### *Technical Support Requirements*

Eliot will demand extensive technical support, similar to what we have seen with WPWin 6.0. We must ensure that we are capable of handling calls in a timely manner, without turning people away with busy signals.

### *Positioning Strategy*

>>Todd<<

### *Critical Success Factors*

>>Todd<<

## **VI. Relationship to Other WPCorp Products**

### *Product Interdependencies*

>>Todd/Scott/PRWin/Office/InForms/Shared Code<<

## **VII. Feature List**

### **Improved Speed and Reliability**

- Focus on speed improvements in key areas, such as file I/O, scrolling and navigation, columns, tables, graphics, printing, and startup.
- Ship a completely reliable and usable product without the need for an interim release to address major problems soon as shipment.

### **A great Chicago Application (32-bitness, threading, explorer, user Interface specifications etc.**

- 32-bit executable fully optimized (leverging our 32-bit work with MAC,UNIX, VAX, AS/400)
- Chicago User Interface Guidelines
- Long files names
- UNC support
- OLE container support and Drag and Drop
- Simple mail enabling
- Must run successfully on Windows NT /Intel. Must degrade gracefully.
- Threading
- Explorer
- Tab dialogs

**Network Intergrated application.**

- Directory Services?? aware
- Information Highway Enabled (Parritt)
- UNC full path support.....If mappings change you can find the document and its parts
- 

**A Great PerfectOffice Component.**

- Leverage the share code effort
- Minimize the difficulties of the Shared BIF (recovery)
- Cross-application Experts (Wizards)
- Common Interface
- ALM
- Visual Appware
- Network Intergrated Suite

**Ease of Use—"PerfectSense" Technology**

- Focus on common tasks, such as entering and editing text, bulleted lists, and printing
- Support fuzzy logic for ranges of options rather than binary yes or no
- Support "PerfectSense" technology to streamline and automate the task of writing and formatting documents:
  - QuickCorrect for on-the-fly spelling and formatting corrections
  - QuickSelect for extending Autocode Placement to words and automatically selecting words when dragging the mouse
  - AutoGenerate to automatically create tables of contents and indexes
  - Format Advisor to automatically format documents, supply a styles gallery for selecting formatting, and shrink-to-fit functionality
  - Table Advisor to automatically format tables and supply a tables gallery for selecting formatting
  - Template Advisor to help select pre-defined documents layouts and formats and assist in building content
  - Task Advisor to monitor a user's work habits and offer suggestions for shortcuts and improvements
  - Advanced morphological tools for intelligent search and replace
- Grammatik 6 for advanced grammar checking and sentence rewriting
- Integrated grammar and spell checking
- Base interface on what the user is doing and is likely to need next
- Consolidate similar features and functionality
- Improve consistency between similar functions
- Combine button options to simplify choices

- Secondary focus on complex tasks using Coaches or Advisors and step-by-step choices in dialogs
- Work in the document itself rather than in dialogs and preview windows wherever possible
- Use detailed explanatory text and instructive icons and images in dialogs
- Direct manipulation of all document objects
- Eliminate the need for Reveal Codes
- Use short help prompts over buttons and other interface elements
- Supply last-used choices options wherever lists of items are presented

### **User-responsive Interface**

- Take full advantage of context sensitivity in Button Bars and menus
- Editable QuickMenus
- Properties dialogs to edit contents and actions of buttons
- Exploit Feature Bars

### **Comprehensive**

- TWAIN support from WPWin File menu
- Spreadsheet export
- Database publishing capabilities
- Multiple-level undo/redo
- Power fields

### **Modularity**

- Reduce the memory footprint of WPWin so it will run better in a 4MB environment
- Reduce the required disk size for a minimal installation to a target of 5-7MB and provide a standard installation with the most useful document tools using under 20MB

### **Integration**

- Support OLE 2.0, OpenDoc, and OBEX
- Spreadsheet and database export
- Spreadsheet and database publishing abilities
- ODBC support
- TrueType font embedding
- Rework Paste Special dialog
- Support full-page OLE objects
- Work with Netware Navigator and Netware License Server APIs

### **New Technologies**

- Workgroup tools for multi-user editing and more

- Document management capabilities with QuickFinder improvements and SoftSolutions technology
- Support for Chicago
- Speech technology
- Linked text boxes
- Ink layer support for pen computing and ink editing comments
- Installation routines for CD-ROM and network distribution

### Transition and Learning Tools

- 5.x file conversion
- Macro conversion and debugging tools
- Keyboard, PRS, etc. conversions
- Transition kit for WPDOS and other users
- Online help, Tutorial, and Coaches for WPDOS users
- 
- 
- 
- 

These major objectives mirror some of the objectives for 16-bit Eliot but other are specifically targeted to be able to provide us the greatest pop and sizzle in our changing market.

Improve speed and reliability (increase start-up time by 30%)

- 
- 

- Renovation and refinements to application processes.
- User responsive interface
- OpenDoc Integration
- Best Transition for WordPerfect for DOS users.
- New technologies
- Modularity
- Closer Alignment with Quattro Pro 32-bit
- Don't take away unnecessarily features that DOS WordPerfect user may be accustomed. (be careful)

- **Great Chicago Application (Windows 95):** As a partner in Microsoft's First Wave Chicago program we have committed to be a well behaved Chicago application in several areas.
- **Best Network Integrated Application:** We should seek ways to leverage our Networking Roots and allow Eliot32 to be more than Network aware or enabled but extend ourselves to be Network

**New Wizards work across suites**

(WINDOWS MAGAZINE, 10/94, p. 29)

Most application aids in the software suites have been limited to helping users accomplish common tasks. One minor exception may be the newest generation of WP Experts from **WordPerfect**. These aids not only help users perform basic tasks, but also query users about where information needs to go and how it is supposed to get there. In addition to this feature, the current generation of **WordPerfect** Office includes facilities for the creation of user-defined application aids. - Jim Forbes

**Ami Pro trades features for speed**

(WINDOWS MAGAZINE, 10/94, p. 46)

Amid user complaints that the two leading word processor have become slow and feature-heavy, Lotus is touting Ami Pro as a resource-savvy application that outperforms both Microsoft Word 6.0a and **WordPerfect** 6.0a. Recent Lotus ads emphasize that Ami Pro 3.01 requires 4MB of memory and 18MB of hard disk space. Ami Pro recently completed a National Software Testing Laboratories benchmarks 49 percent faster than Microsoft Word 6.0a and 103 percent faster than **WordPerfect** 6.0a. While admitting that Ami Pro may require fewer resources, **WordPerfect** counters that by not releasing a major Ami Pro update in more than two years, Lotus has lost ground in the word processing market. - Philip Albinus

**ISVs mimic office suites**

(WINDOWS SOURCES, 9/94, p. 34)

The Microsoft Office Compatible program is a certification process designed to help third-party software developers imitate the look and feel of Microsoft Office applications. Does Microsoft's interface initiative mean that ISVs will focus their development resources on Microsoft Office, leaving rival suite users with fewer applications to choose from? The terms of certification for Microsoft's look-and-f

**Novell to bridge OpenDoc, OLE gap**

(CRN, 9/19/94, pp. 1, 253)

**Novell** plans to bridge the gap between the OpenDoc and Object Linking and Embedding 2.0 object standards with technology that will allow resellers and developers to build applications to run in both object environments. The new vision for Visual AppBuilder, if successful, would greatly simplify development and make it easier to sell object-oriented programming to customers, said integrators and developers. - Ed Sperling & Shawn Willett

**OpenDoc ripe to charge OLE**

(WINDOWS SOURCES, 9/94, p. 28)

OpenDoc is an open standard that is being supported by several important vendors on a variety of different computing platforms including Windows, OS/2, Unix, Macintosh, NetWare, and ultimately IBM mainframes and minicomputers.

David Moon, a chief developer of **WordPerfect** believes that OpenDoc support will give his products an advantage in the market by making it simpler for customers to use **WordPerfect** in combination with other programs. Jed Harris, President of Component Integration Labs, is enthusiastic in recommending OpenDoc, "All Windows developers should consider OpenDoc for three reasons: First, OLE 2.0 support is built in so you don't have to choose.

Second it's much easier to code to OpenDoc than OLE - ask anyone who's done both. Third, the OpenDoc implementation of OLE delivers more OLE features than most programmers could handle on their own." - Frederic E. Davis

**OpenDoc: better OLE than OLE**

(WINDOWS SOURCES, 9/94, p. 65)

Don't look now, but OLE - the shining star of Chicago - is about to be eclipsed by OpenDoc. In case you haven't heard, and I'm sure Microsoft hopes you haven't, OpenDoc offers more flexibility and better integration than OLE. In short, it's a better OLE than OLE. Software developers will find OpenDoc easier to work with than OLE because the consistent OpenDoc API is more powerful and straightforward than the complex group of APIs needed to support OLE 2.0. - Frederic E. Davis

### **OpenDoc, OLE squaring off**

(MACWEEK, 9/19/94, pp. 1, 72)

OpenDoc proponents landed a series of body blows that sent Microsoft's OLE 2.0 efforts reeling last week. Adobe joined Component Integration Labs, the nonprofit association promoting OpenDoc technology, and **Novell** adamantly rebutted a rumor published elsewhere that it planned to defect from CI Labs. Microsoft responded by pushing vendors to bypass OpenDoc and create applications exclusively for OLE, or Object Linking and Embedding 2.0. While Adobe and **Novell** threw their support behind OpenDoc, a handful of developers claim Microsoft has been coaxing them to develop only for OLE. - Jon Swartz

### **Editorial: CI Labs suffers from mixed messages**

(MACWEEK, 9/19/94, p. 24)

The handful of dedicated folks at Component Integration Labs must be confused. Key companies that set up CI Labs to develop, deploy and defend OpenDoc - Apple; IBM Corp.; and **WordPerfect**, the **Novell** Applications Group - have been subtly undermining the organization's goals. IBM has said that OpenDoc is fine for developers that want to migrate existing applications into OpenDoc parts. But if you're developing new programs, Big Blue thinks you should skip OpenDoc and jump on the Taligent bandwagon. It's also been reported that in cozying up to Microsoft, **WordPerfect** has agreed to play down its OpenDoc enthusiasm in favor of OLE 2.0. **WordPerfect** has always been a supporter of OLE, but its concurrent backing of OpenDoc is said to be faltering

### **OpenDoc standard touted as solution for documents of mixed lineage**

(GLOBE AND MAIL, Canada, 8/30/94)

Can a unified PC industry make Microsoft flinch? It certainly will give it a good shot with something called OpenDoc. OpenDoc promises to deliver true interoperability among different brands of software. For its part, **WordPerfect** is in the midst of modifying the alpha 6.0 version of OpenDoc for the Windows operating system. Bill Kesselring, **WordPerfect**'s Manager of OpenDoc technology, believes OpenDoc will be widely accepted by developers. - Andrew Tausz

### **Microsoft debuts low-cost NT 3.5**

(CRN, 9/19/94, pp. 1, 253)

Microsoft this week plans to unveil bold new pricing for Windows NT 3.5 and related server applications, and is rolling out a strategy to push the product through **Novell** resellers. At press time, Microsoft planned to offer Windows NT Server 3.5 for \$699, plus \$39.95 per client. It also planned to offer a bundle of the product with SQL Server, SNA Server, and Systems Management Server and electronic mail - a suite known internally as Back Office - for \$2,199 for the server, plus \$309 per client license. Microsoft also is targeting NetWare accounts, offering free NT training to VARs, seed units and 50 percent discounts on client licenses for NetWare users. - Barbara Darrow & Ed Sperling

