

**PX 1**

**From:** Bill Gates  
**To:** Bill Bliss; Bob Muglia; Brad Silverberg; Brad Struss; Brian MacDonald; Chris Guzak; Chris Peters; Darryl Rubin; Doug Henrich; Erik Gavriluk; Jim Allchin; Joe Belfiore; Kurt Eckhardt; Leif Pederson; Mike Koss; Paul Maritz; Russell Siegelman; Satoshi Nakajima; Steve Madigan; Tom Evslin  
**Cc:** Brian Fleming  
**Subject:** Shell plans - iShellBrowser  
**Date:** Monday, October 03, 1994 5:18PM

Its time for a decision on iShellBrowser.

This is a tough decision. The Chicago team has done some great work in developing a user interface that will be a big step forward for millions of people. The explorer is an important part of this because it provides a neat paradigm for finding interesting information. The shell group did a good job defining extensibility interfaces. It is also very late in the day to making changes to Chicago and Capone.

it is hard to know how much actual market benefit iShellBrowser integration would bring. I believe Chicago will be very successful either way. Unfortunately I don't think the integration will have a marked effect in terms of Capone competing with cc:Mail, so that battle will have to be won on other grounds. This is not to say that there was anything wrong with the extensions - on the contrary they are a very nice piece of work.

On the other hand, we are in a real struggle vs. Notes and the Office/REN team needs to move as quickly as they can to deliver really rich, unified views of information and to provide and exploit storage unification as systems makes that possible, and we need as clear a path as possible to allow them to do that. The Ren team has a lot of challenges and compatibility would be an extra effort for them of at least 5 man years. If we felt we could expand this team easily to help Office, beat Notes, be a source of future shell technology and be compatible then I would say the extensions are ok. However the Ren team will find it tough to deliver on all of these even without compatibility.

I have decided that we should not publish these extensions. We should wait until we have a way to do a high level of integration that will be harder for likes of Notes, Wordperfect to achieve, and which will give Office a real advantage. This means that Capone and Marvel can still live in the top level of the Explorer namespace, but will run separately. We can continue to use the iShellBrowser APIs for MS provided views such as control panel, and can use them for other MS-provided views that don't create a large compatibility or ISV issue.

i would also like to add a few words about the recent Shell re-organization. We have gone from three centers of UI innovation to two. There is a lot of pain in doing this. All 3 groups were doing excellent work and I hope the Cairo shell and Ren can come together to provide the best of both. I think there will be real benefits to be reaped. Having the Office team really think through the information intensive scenarios, and be a demanding client of systems is absolutely critical to our future success. We can't compete with Lotus and Wordperfect/Novell without this. Our goal is to have Office'96 sell better because of the shell integration work, and to have the Ren/Office effort yield technology that can be an integral part of the shell in Windows'97. I look forward to the Office team getting excited about using Component Forms, OLE automation, OFS, etc. in the future - and pushing systems much harder than before.

The Personal Systems team has many challenges ahead of it - they need to remain focussed on overall systems ease of use, and on being the conscience of the individual/home user - on thinking through integration of new opportunities opened by the Internet, by CD-ROM titles, etc. This means that we are still going to have to work together and deal with tensions as they arise, but we can't give up on either market, and there is a huge amount of creative work to be done. We need to allow for innovation in both Office and Windows, even if this makes the line between them hard to draw.

EXHIBIT # 24  
 WITNESS Gates  
 DATE 5-19-09  
 ZOYA SPENCER (206)382-9895

FL AG 0042508  
 CONFIDENTIAL

MX 9030733  
 CONFIDENTIAL

FL AG 0087459  
 CONFIDENTIAL

**PX 61**

**HIGHLY  
CONFIDENTIAL**

## Office Shell Ideas and Issues

*The latest version of this document can always be found on \\design\public\chrisgr\offshell.doc.  
Please open it as READ ONLY.*

### Summary

This paper investigates a proposal that the next major version of Office after Chicago should consist of a Windows shell and applications optimized to work together. The proposal originated at a senior technical retreat at Hood Canal in June/93.

**Recommendation:** We should follow the "Aggressive" version of the plan outlined below.

### Proposed Plan

- Bundle an enhanced Windows shell with the next major version of Office to ship after Chicago.
- The Office shell would be functionally a superset of the Chicago shell, designed for maximum synergy with Office..
- Enhancements to the shell could include minor modifications to the shell UI for optimal interaction with Office apps; increasing the extensibility of components such as the Explorer, the Desktop and the Tray; the provision of app-specific extensions to take advantage of them; and additional applets, file viewers, OLE servers and other tools.
- Apps in the Office shell release would include Excel 6, Word 7, PowerPoint 5 and Access 3.
- The Office shell would define the next standard Windows UI after Chicago. At an appropriate time after Office+Shell ships, the enhanced shell would become the next standard Windows shell for both Chicago and Cairo.

### Schedule

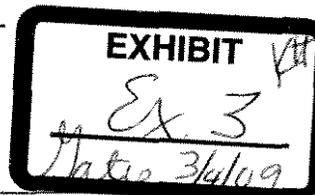
- |                             |  |
|-----------------------------|--|
| Q2/94                       | <ul style="list-style-type: none"> <li>- Chicago ships</li> <li>- Shell has limited extensibility. (See below for details.)</li> </ul>   |
| Chicago + 6 months          | <ul style="list-style-type: none"> <li>- Office ships with optimized shell.</li> <li>- Shell adds features for optimal support of Office requirements. (See below for details)</li> <li>- Office includes many features that exploit the new shell.</li> <li>- New shell not initially available with Windows itself</li> <li>- The Office shell should be approximately a superset of Chicago shell features (although some components, such as the Tray, may be replaced.)</li> <li>- Note that the Office shell date may not be strictly dependent on when Chicago ships. If the Office shell used a different code base, then a slip of Chicago could reduce the delta to less than 6 months.</li> </ul> |
| Sometime after Office ships | <ul style="list-style-type: none"> <li>- Cairo ships with a shell that is a superset of the Office shell</li> <li>- Excludes any components that we choose to keep only for Microsoft Apps.</li> <li>- Extended to use special features of Cairo</li> </ul>  |
| When Cairo ships            | <ul style="list-style-type: none"> <li>- Enhanced shell added to Chicago</li> <li>- An alternative would be to add the Office shell back into Chicago when Office ships. This should still give Microsoft Apps a significant development lead.</li> </ul>  |

Plaintiff's Exhibit

7377

Comes V. Microsoft

- Page 1 -

OFFSHELL.DOC  
Microsoft Confidential  
02/03/93 11:10 AMMS 0097121  
CONFIDENTIAL

**HIGHLY  
CONFIDENTIAL**

## Pros

- Chicago team can concentrate on shipping within their memory targets and schedule because they would have to add less OLE support, and would not have to provide as much extensibility.
- Office gets a shell optimized for its use.
- Office gets a big jump on competitors in creating apps optimized for the new shell.
- Since the new shell is bundled with Office, we don't have to assume that it needs to run on Win 3.1. (Issue: Actually, this would require bundling all of Chicago with Office.)
- Assuming the Office shell is upward compatible to the Cairo shell, then Office apps will be automatically much more optimized for Cairo.
- Simplifies the cross-group interaction necessary to produce synergistic versions of apps and the shell.

## Cons

- Risk of ISV retaliation.
- Negative impact on corporate image.
- Would probably delay release of Excel 6, Word 7 and other Office apps to do work necessary to leverage shell. This would probably mean we would not get Chicago-optimized releases within 3 months of when Chicago ships, as originally planned.
- Might require some extra work by Chicago to provide enhancements or hooks needed for eventual use by the Office shell. (We don't want to have to ship new versions of GDI and User in the Office time frame.)
- Increases the pressure to ship major apps, and adds the shell as another component to ship.

## Product Vision

There are two possible plans we might follow:

- 1) Conservative plan: We develop enhancements to the shell and modifications to apps that are relatively well understood, and don't change current designs too much. The emphasis would be on creating an Office shell that has considerably higher value added than the shell in Chicago, both by limiting what we provide for free in Chicago, and by adding features in the Office shell. We would also add features to applications to leverage the currently planned shell features.

### Advantages:

- This plan has less impact on current designs and schedules. For example, we originally wanted minor upgrades of major apps to ship as soon as possible after Chicago to optimize them for Chicago, and to showcase Chicago features.

### Disadvantages:

- We may not be taking full advantage of this opportunity.
  - Assuming we do intend to eventually do the changes described for the aggressive plan, it would come later, and might have to be done in parallel with the Office shell work.
- 2) Aggressive plan: We use this opportunity to bring about a major improvement to the model of how users interact with the shell and applications. This could include changes as large as switching

**HIGHLY  
CONFIDENTIAL**

apps to SDI, and the necessary changes to the shell to optimize it as an environment for SDI or document-centric apps, and to make progress on the problem of factoring functionality between apps and the system.

**Advantages:**

- We could gain a much bigger advantage from the Office shell. We could pull off the "UI Paradigm Shift" to document centricity possibly two years sooner than if we did not follow this plan. Major breakthroughs in app usability may be possible. This would give us a very significant lead over our competitors, and make our competitors' products look "old".

**Disadvantages:**

- It would certainly take longer to ship the Office shell and related apps because the design issues are less well understood and the development work would be greater.
- It could delay the minor Chicago-optimized releases of apps. We could still ship minor app upgrades soon after Chicago. However this may cause too many upgrades too close together. This would dilute design, development and testing resources, and could delay the release of the Office shell. We would have to resist the temptation to add too many features to these minor releases.
- Implications for Mac core-code/core-doc strategy are not well understood. The aggressive plan would cause us to confront these issues sooner.
- Implications for the ability to run on Win 3.1 are not well understood. We probably could produce a version that would install and run in a limited way on Win 3.1, but it would take more work.
- In the past, people have assumed that developing next-generation apps ("Cairo apps") should include major architectural changes in addition to user model changes. However, the proposed aggressive plan puts more emphasis on the user model, although it does include some less extensive architectural work such as enhancements to OLE, improved OLE support, and enhanced programmability. Deeper architectural changes, as appropriate, would come in subsequent versions.

The following is a list of possible features in the Chicago shell, the Office shell and the Cairo shell. These specific features are largely orthogonal to whether we pursue the conservative vs. aggressive plans described above.

**Chicago Shell Includes**

- Most of the features currently planned for Chicago, including:
  - Combined program manager and file manager
  - New visuals
  - Context menus, drag/drop, NDD, etc.
  - Interoperability enabling. i.e. Supports drag/drop compatible with OLE
  - OLE 2.0
  - Simple Idispatch enabling of shell and applets. (So Excel 5 VBA can get the benefit of being the best language that can program the shell.)
  - Probably supports extensibility of document property sets and commands.
  - Assuming there is a "simple shell", it is upward compatible to the Office shell.
  - If there is a tray, it is not extensible, and not replaceable
- But not including:

**HIGHLY  
CONFIDENTIAL**

- Extensibility e.g. Explorer not extensible (Capone hard coded into explorer)
- Vbasic
- Full-featured document viewing. Maybe only allow viewing thumbnails with the shell. A full set of document viewers would only ship with the Office shell.
- Some of the features of Mail. Can we limit the feature set of the Mail that is included with the system, to leave more value-add for Office?
- Other changes, TBD, to shell for optimal interaction with Office apps.

**Office Shell Adds**

(I assume that only some of these things could be done in the time available.)

- Moving apps to SDI. I'm optimistic that we could make SDI work very well given the opportunity to design apps and the shell together to make the shell an optimal environment for SDI app windows to reside.
- (Note that doing SDI would require following the "aggressive plan" described above.)
- VBA, including ability to automate cross-app scenarios that include the shell.
  - Explorer extensions to browse into app document types: OLE Objects in Docfiles, Excel workbooks, Clipart files, etc.
  - New tray designed for maximum benefit to cross-app requirements of Office
  - OLE-based workbook
  - OLE extensible Explorer
  - OLE extensible desktop
  - OLE extensible tray
  - LoisO's document library as a low end document library solution for Chicago. Would be supported on desktop and in File Open, etc. Cairo doc mgmt should be upward compatible.
  - Enhanced cmdmg.dll, and cmdmg code sharing with apps in shell/office bundle
  - Investigate feasibility of adding multiple, switchable desktops
  - Useful objects that could be placed on the OLE-container enabled desktop:
    - Information displays such as Post It Notes, data fields, tables.
    - Controls like buttons or sliders, that could activate VBA scripts.
    - Graphical indicators like warning or status lights, gauges, or even charts.
    - Special purpose information containers such as "document piles", "parts bins", etc.
    - Communication devices or devices that interact with the "Microsoft At Work" office
    - Decorations, such as clip art, pictures of one's family, etc.
  - Enhanced mail: Add back what we took out of Chicago mail. Also add features for synergy with Office apps.
  - The Office shell would be the target platform for Ren.
  - Can some support for smart folders and/or project folders be added at this point? VBA programming of smart folders.
  - Toolbar code sharing with apps in shell/office bundle?

**HIGHLY  
CONFIDENTIAL****Cairo Shell Adds**

- Query based Explorer into OFS and summary catalogs
- Smart folders
- Project folders
- Other features necessary to work with OFS/DFS, security, etc.
- Infobooks

**Assumptions**

- The Office shell would start with the Cairo shell code base, but would be subsetted and adapted to run on Chicago, and shipped in time for Office.
- The office "shell infrastructure" would still be developed and productized by Systems. However, the Integrated Office group would in parallel develop extensions. The Systems base code and the Office extensions would ship simultaneously and appear as a seamless part Office. Some of the app extensions might eventually become part of Cairo and Chicago 2.
- Since Chicago shell does not need all the bells and whistles, it should now be easier to meet its memory goals and schedule.
- We will be able to make OLE fast enough, and reduce the working set enough to support the desired scenarios.
- We would have a little more time to design apps synergy features into Office shell
- Changing apps to SDI would be more feasible because of the opportunity to optimize the shell itself as the working environment for Office.
- "Integrated Office 1" would be redefined as Office Shell + Office Apps.
- Participants in the Office ISV program would be brought into the plan soon enough to announce support when the Office shell ships.
- Ren would probably require the advanced shell since it relies on Explorer extensibility.

**Issues**

- Need to determine ASAP any features needed in Chicago to support enhanced shell. e.g. What to we need in USER to support planned features?
- Would need to ensure compatibility of enhanced shell with 3rd party apps.
- What staffing would be required? How to organize?
- Keeping in sync with Chicago and Cairo versions. There's no way we can support three separate shell code bases. We'd need to divide the responsibilities clearly.
- Code base for Office shell? Probably the Cairo shell code.
- Do we also include the shell with the non-office versions of apps?
- If apps rely on shell extensions for important functionality, then to be cross-platform, we would have to duplicate these things on the Mac. For example, the Mac desktop isn't an OLE container.
- Can the Office, including the new shell require more than 4 meg of RAM? (I think the answer is probably yes, assuming the late 1994 time frame, but preferably basic functionality would still work in 4 meg.)
- Is the above schedule too tight? If so, is there a way we can scale back the plan, or stretch out the schedule?

- Are the Office apps of this generation only available as 32 bit?
- Does the Office shell use win32 OLE with LRPC as IPC?
- What kind of 16/32 interop work is required?

**HIGHLY  
CONFIDENTIAL**

**PX 63**

Trip Report  
**Chicago User Interface Design Preview**  
July 8th and 9th, 1993  
Rich Hume and Grant Skousen

**Note:** The information obtained at this preview was obtained under non-disclosure. Tim Satalich, the Windows Systems User Interface Evangelist, took the time at the beginning of the first day to warn everyone that if this information (especially the visuals) appears in PCWeek or any other magazine then Microsoft will attempt to find out who leaked the information and the guilty company will not be invited to future previews like this.

### Summary

Chicago is definitely real, but it is at least a year away from release. The Group Program Manager for Chicago said that they are planning three betas of three months each. The first beta is planned to begin late this month<sup>1</sup>. That is when we should receive the PDK (preliminary development kit) and runtime that will include the new shell. When the PDK is received the authors would like to give a demo that will overview the UI changes in Chicago.

There are still a number of areas where there are disagreements within Microsoft on how certain features should be implemented. The presenters seemed most interested in input in those areas, presumably to gather ammunition for their own opinions on how the feature(s) should be implemented. Other presenters gave the impression that they had already decided how things are going to be and that they were not very open to input.

When released, Chicago will boot directly into the Windows GUI interface. All functionality provided by DOS 6.0 utilities today will be available from Windows. Most (if not all) of these utilities will have new graphical interfaces. There will be significant improvements to winoldap (i.e., the DOS box) including a toolbar and the use of TrueType fonts that will auto scale as window size is changed.

Brad Silverberg (VP over system development) emphasized that Microsoft is attempting to grow the market. Today, most machines sold are either upgrades or second machines. To help get more new customers, MS is working with a number of other vendors to get plug and play capabilities in the open PC hardware world much like Apple already enjoys in their proprietary hardware. Based on the things said, it appears that Chicago has two overriding goals. This first is ease of use. Brad Silverberg stated that Microsoft intends to make the Windows platform the easiest computer to use, bar none. Toward that end, when alternative designs are considered, the approach that appears to benefit novice users is given priority. The second goal is to make the upgrade from Windows 3.1 to Chicago a no-brainer.

---

<sup>1</sup>It is interesting to note that the first beta is also the second release, the first release (which we have) was an alpha release. The first beta (or second release) was also referred to as M5, or milestone 5. One would presume that M4 was the alpha release. No mention was made as to what the first three milestones were.

The following quote from the Preliminary Chicago User Interface Design Guide points out some of the challenges WordPerfect Corp. will face as we prepare our applications for the next release of Windows:

"Windows (Chicago) is the advent of the evolution of GUI to OOUI; that is, a more data-centric/object-centric, rather than application-centric, interface; a continuation of the design direction portended by OLE. As a result, application developers/designers may need to rethink their application in terms of what are their basic components and the respective operations and properties that apply to those objects. This is particularly important since long term, from the user's perspective, the application is destined to become as transparent to the user as the code which manages menus. The user's data/information (and tasks associated with that data) will become the major focus on the user interface."

On the second day of this conference, Joe Belfiore, the program manager for the Chicago Shell/UI gave these 14 items as the UI Designer's "Points Of Light" (Note OLE 2.0 capability including Doc Files was assumed throughout the conference):

- 1)Use 32 bit APIs. (Note: this is the one non-UI item that was mentioned several times)
- 2)Support long filenames
- 3)Support Universal Naming Convention (UNC) path names
- 4)Make sure that documents/data files are accurately displayed and used in the shell
- 5)Support drag and drop and other transfers consistently and extensively
- 6)Use the common dialogs, especially File Open
- 7)Be careful about multiple instances of application being started too easily
- 8)Maintain a consistent user interface and object paradigm between the application and the shell.
- 9)Extend the shell's ability to provide general information about application files (more OLE 2.0)
- 10)Support pen input for pen notebooks and desktop tablets
- 11)Support Chicago-Style help
- 12)Size / color scalability
- 13)Avoid drawing custom title bars
- 14)Don't draw into minimized windows

These points are expanded upon in the detailed notes that follow, but they suggest that creating great Chicago applications may require significant new design. The change to an Object Oriented UI (OOUI) may require a paradigm shift greater than that we experienced when going from a character based UI to a Graphical UI. We feel that it is of critical importance that we get the foundation (primarily 32bit and OLE 2.0) in place soon. If we do not, it is unlikely that we'll have a prayer of releasing great Chicago applications in a timely manner after the system is available. You can be sure that Microsoft will expend some effort in making Chicago successfull, especially if NT sales are lackluster over the next year.

## **Overview**

**Tim Satalich, Microsoft Windows System User Interface Evangelist**

### Day 1 Agenda

Introductions

Chicago Product Overview

Chicago Shell Overview

New Controls, Dialogs, and Shell integration

Mobile UI

Breakout Sessions (Transfer model, Explorer / browser, menu behavior)

### Day 2 Agenda

Visual Style Guidelines

Application Style Guidelines

User Assistance

How to be a Great Chicago App

Cairo User Interface Directions

Breakout Sessions (User Assistance, UI and Visuals, MDI)

Microsoft will be setting up a private CompuServe forum to discuss Chicago UI issues.

Microsoft is planning more UI design reviews in other system areas (no dates set yet). It was interesting to note that the Cairo Program Manager stated that he felt they were ready for a Cairo design review right now.

The following sections attempt to present information of interest that came out of each presentation. The slides and the information contained on them are not presented verbatim. The authors each have a bound copy of the slides that is available if you are interested. Tim was unwilling to give us an electronic copy of the slides.

### **Chicago Product Overview**

#### **Dennis Adler, Group Program Manager**

Chicago delivers features tuned for personal needs while Cairo delivers Chicago features plus features tuned for corporate needs. The typical migration path for Windows 3.1 and Windows for Workgroups 3.1 is to Chicago. The move to Chicago is intended to be a no-brainer for users with a 386 and 4 meg. The typical migration path for Windows NT 3.1 is to Cairo. Cairo will "really shine" on the network.

On a 4 meg 386, Chicago should meet or beat Windows 3.1 in performance. According to Brad Silverberg, with 8 meg Chicago will blow away Windows 3.1.

Chicago components will form the core of MS-DOS 7.

Microsoft is working with many hardware vendors as well as BIOS vendors to establish standards that will provide a "plug and play" environment for consumers. The goal is a "no-compromise standard for PCs".

With Chicago, Microsoft will be providing a new setup program. The system as well as Microsoft applications will be providing an optional 'slim' installation to reduce disk requirements.

The new Chicago shell will provide a unified manager and desktop. There will be no more separate program manager and file manager. Drag and drop will be ubiquitous within the shell (and within the system if apps cooperate). The shell is OLE 2.0 enabled. Long file

names will be supported<sup>2</sup>. The long file name support is being added to the FAT file system using some undocumented bits. No details were given because a patent is being applied for. The new shell will also likely include file viewers (others may be added in an extensible way). The pen gestures have been simplified and additional support is being provided to make pen support easier to add to applications. There is an improved printing interface that offers much better background printing and the DOS application interface has been improved.

Chicago provides an extensible file system architecture and will ship with a 32 bit version of the FAT file system. The file system includes a last accessed date field. Chicago also includes a 32 bit CD-ROM file system. A protected mode implementation of double space will be provided.

Chicago is supposed to provide significantly improved communications capabilities. It was stated that most of today's products do NOT use the Windows communications driver.

Chicago provides a preemptive multi-tasking kernel. The system boots in real mode to maintain compatibility. After autoexec.bat has run, the system switches to protected mode operation. Each MS-DOS application and Win32c application executes in its own address space. All Win16 applications operate within a single virtual machine.

Chicago will provide a 32 bit multimedia subsystem including audio, video, and image compression services. The multimedia utilities will be OLE 2.0 enabled. The default system palette in Chicago contains 256 colors.

Chicago was presented as the enterprise client

According to Brad Silverberg, Chicago is "on track for a mid-'94 release". An update to Chicago is planned to coincide with the release of Cairo to ensure that Chicago operates as a good client for Cairo servers.

It was stated that it will likely be difficult to replace components of the shell.

A couple of the participants expressed strong concerns that Chicago perform better than Windows 3.1. One even said that their product was losing accounts because of the slowness of Windows itself.

## **Chicago Shell Overview**

### **Joe Belfiore, Program Manager, Chicago Shell/UI**

The work on the shell to date has been targeted primarily at existing Windows 3.1 users. However, the team's focus is shifting to server novice users as well. The stated goals for the Chicago shell include: providing a single object oriented paradigm (no more Program manager / File manager duality); begin moving towards document centric view; lower the entry point for novices (as compared to Win 3.1); and share a UI with Cairo (Chicago provides a subset of Cairo's functionality).

---

<sup>2</sup>It is interesting to note that long file names in Chicago will be limited to 254 characters while NT supports 255. The Group Program Manager for Chicago said that they estimated a 15 day development time in Chicago to get 255 characters so he decided to go with 254.

Most of this presentation consisted of a demonstration of the Chicago shell. Since we should have a copy of Chicago with the shell towards the end of the month, no attempt is made here to fully describe it. Only the most notable highlights are mentioned.

- The desktop contains icons representing active tasks as well as icons that do not represent active tasks (e.g., documents and directories can be dragged and dropped on the desktop)
- The desktop contains a tray that can be positioned at any edge of the screen. The tray can be thought of as the system toolbar. It contains three buttons, one for shutting down the system, one for searching for objects, and one for accessing help. The tray also contains an area much like the shelf in Office 4.0. Anything that can be dragged to the desktop can be dragged to the tray for easy access. This area of the tray uses 16x16 icons which will be created if the application does not define any. The tray also contains the current time of day. They realize that the tray will need to be moved out of the way at times. One implementation they have looked at is a tray that pops up when the mouse is moved to the edge of the screen. Dennis Adler said that he would veto this implementation though. The other possibility that was discussed was that you could grab the edge of the tray opposite the edge of the screen and drag it off the screen (leaving probably a three pixel handle for grabbing and dragging back onto the screen).
- The look of open windows has changed quite a bit. The caption will likely have a 16x16 document icon (as defined by the app) at the left edge. This icon will provide access to the system menu on the left button up. The icon will also provide a quick menu on right button up. The text in the caption is left justified rather than centered. The text will consist of the document name followed by the application name. The minimize and maximize/restore button graphics have changed. There is still discussion going on as to whether separate close and/or restore buttons are needed.
- Minimized windows are simply truncated captions. At least this is the most likely implementation (and it is definitely the implementation currently planned for Cairo). This was an issue that was brought up many times as an issue that they were seeking input on. The current alpha copy that we have shows a 32x32 icon with text to its right. The more appealing implementation (and the one preferred by most if not all participants) has a small icon with part of the text from the caption to its right. (Note: the width of the minimized window is under user control. Also, this implementation will break applications that currently write into the minimized window). All minimized windows are of the same size because they make better bricks.
- By default, the desktop includes three icons: the filing cabinet, the world, and a programs folder. The programs folder contains links that pretty much correspond to program manager icons today. The filing cabinet opens the explorer in a manner very similar to today's file manager showing all local drives including mapped drives. The world icon views the network using UNC pathnames.
- New menu behavior. Menus will track with the mouse button up. Cascaded menus automatically unfold when mouse moved over it. (Note: this felt very natural and should not cause any problems for existing users because the press drag release model also still works.) The menus are now gray with a 3D etched look for disabled menu items. There is no support planned for menu items with an indeterminate state (e.g., Bold when text that contains some bold and some not bold is selected).
- Links. There were many participants who disliked the word link. Links are very much like Program Manager icons in Windows 3.1 in that they have command line arguments, open window state settings, hot keys, and working directories associated with them.

Links are implemented as entries in a per-directory database stored in a DESKTOP.INI hidden file in each directory. (Note: support for displaying these links is built into the open and save as common dialogs. This will be an issue for us if we choose not to use the common dialogs because the file system API calls won't return or handle the links stored in a directory). Links can have a different icon and parameters that the original object—this is the difference between links on Windows and aliases on the Mac.

- Transfer model. This is an area that we need to look at very closely. It was interesting to note that the Excel guy who was participating was quite vocal about his disagreement with the currently planned transfer model. The interesting part is that the planned model is almost exactly what Excel already does, with a slight name change. A complete description of the transfer model is beyond the scope of this report (the style guide has about seven pages of description) however, I will attempt to provide a simplistic overview so that you can get the flavor<sup>3</sup>. 'Cut' is no longer a required or recommended action, and 'Paste' is replaced by 'Put'. At the source you make a selection and then pick either 'Move', 'Copy', or 'Link'. None of these options make the selection go away. At the destination, you choose 'Put <object> Here' or possibly 'Put <object> Here As...'. If you had chosen 'Copy' then the menu would likely read 'Put Copy of <object> Here'. The same is true for link. If 'Move' had been chosen, then selecting 'Put <object> Here' would cause the original selection to be removed and placed at the destination (much like cut and paste). The new transfer model still uses the clipboard so that you can say 'Move', 'Put', 'Put', ... 'Cut' is an optional command that may still be provided. The proposed accelerator for 'Move' is Ctrl+M with Ctrl+X as a suggested alias. A right click drag and drop displays a menu at the destination that includes 'Move Here', 'Copy Here', and 'Link Here'. Because the right mouse button is used in this way, quick menus are not displayed until the mouse up. This is a tradeoff because you can no longer press, drag, release to make a quick menu selection. However, Microsoft's usability testing has shown that many (if not most) users click the right mouse button anyway and do not use the press, drag, release method of making selections. Doing a left mouse drag and drop performs the default operation (which is most often a 'Move').
- Scraps. Applications will be expected to support the user selecting a chunk of data and dropping it on the desktop (or tray). It appears there as a scrap and may be later picked up and dropped into a document or container. (supported via OLE)
- The shell is currently using a chevron '»' to denote a link. There was universal distaste amongst the participants regarding this. The design team will likely try to come up with an alternative, possibly graphic, representation for a link.
- The user will be given significantly greater control over the visual look of the environment. Specifically sizes (like scroll bar and caption sizes) and colors can be set.
- Scroll bars now have a proportionally sized thumb.
- The control panel dialogs that were demonstrated had large graphical previews.

The shell properties are not extensible. It will likely be difficult to replace shell components (this is not something that is being designed for).

The shell ties into and used the OLE registration database extensively.

---

<sup>3</sup>As soon as we get an electronic version of the style guide it will be easy to share this information. In the meantime, you can contact either of the authors for a hardcopy.

There is a strong move to eliminate user INI file settings. Brad Silverberg stated that every entry left in the system.ini file has to be justified to him and the group program manager.

Brad noted that a number of the changes coming in Chicago are already available in SnowBall (the latest Windows for Workgroups beta).

### **New Controls, Dialogs, and Shell Integration**

#### **Joe Belfiore, Program Manager, Chicago Shell/UI**

The following new controls will be provided in Chicago:

##### **Toolbar**

Will support button wrap so it can be made to float or dock

##### **Status Bar**

The menu font should be used in the status bar. The status bar appears to have an optional(?) visual at the right end that is used as a larger handle for sizing a window (much as is done on the Mac).

##### **Column Heading**

Supports resizing columns by dragging as well as sorting by clicking (exactly as is done in MS Mail today). A request was made that the control be made to support reordering of columns via dragging as well as giving the ability to hook the right mouse button for quick menus.

##### **Slider**

##### **Spin Buttons**

Much like our counter control

##### **Progress indicator**

Can be used in status bar for long processes

##### **Tabs**

Supports multiple banks of tabs as well as horizontal scrolling. Bitmaps can be placed on tabs. A request was made that the tabs allow visually showing a 'dirty' state for use in property sheets when one page has been changed but the changes have not yet been committed.

##### **Property Sheet**

Pages are added as dialog templates. Much like property sheets in OS/2.

##### **List View**

Supports large icon, small icon (these are both positional), list, and details views. Uses an image list icon cache that may be made available for use elsewhere.

##### **Tree View**

Used to display hierarchies - containers, outlines, etc. The application can add its own icons as well as hiding or showing lines and expand/collapse visuals.

##### **Rich Text Edit**

Supports more than 64k of text. Provides OLE 2.0 client support. Provides left, right, and center align on a per paragraph basis. Supports definable left tabs. Multiple font support as well as simple bulleting, find and replace, and superscript, subscript, and strike through.

##### **Common Dialogs**

The 'File Open' and 'Save As' dialogs are changing substantially from their current form and functionality. They will provide most of the functionality of the explorer. It will also be possible to right click on items (files, directories, etc.) in the dialog to get a context menu for

the item. These dialogs will display and handle links properly. These dialogs will also likely provide a file preview using OLE 2.0 thumbnails as well as included file viewers.

The Print dialogs generated a lot of discussion. MS is currently suggesting three dialogs: a 'Print' dialog (much like today's); a 'Choose Printer' dialog (much like today's Print Setup dialog); and a new 'Page Setup' dialog. (Note: in one of the breakout sessions, the Excel person was rather vocal about this not being a very good approach, while the Word person seemed to like it a lot)

The 'Find' and 'Replace' dialogs as well as the 'Fonts' dialog are very similar to the Windows 3.1 versions.

Once again they are planning a color dialog. The design is "still underway".

The OLE 2.0 dialogs will be 'Chicagoized' (e.g., Put Here As instead of Paste Special).

Once again in this presentation the support of OLE 2.0 by applications was emphasized. In particular, the use of compound file summary properties was noted as something that will be displayed by the shell for all files that have it. Also, in the support of 'scraps'.

Joe solicited input as to whether and which common dialogs should be made available for NT as well as Windows 3.1 (where there are differences).

### **The Pen in Chicago**

#### **Michael Van Kleeck, Group Program Manager, Mobile Services**

MS appears to have not given up on the pen yet. The Chicago shell will be pen aware. There are several new types of edit control specifically for pen support. The gesture set has been simplified and reduced to five basic gestures. Standard controls like lists and check boxes directly support the pen.

A couple of things that will be provided in Chicago to make pen support easier are a transparent control that supports gestures and ink and a system API - DoDefaultPenInput (not discussed in detail).

### **Chicago Remote Network Access (RNA)**

#### **Wassef Haroun, Microsoft Windows Systems Program Manager**

The RNA component of Chicago is intended to make remote access to the network very easy. A Chicago desktop can act as a private host or as a network gateway. Chicago will support merging of a remote file system with a host at a large granularity level (i.e., newer files will overwrite older ones).

From usability testing, MS found that it is important to minimize the number of steps required to establish a connection. Connection information resides in objects that appear in a special 'Remote Networks' folder. Opening one of these objects initiates a connection sequence.

The presenter solicited input as to what kind of information we believe would be useful to display in the RNA Connection Window (e.g., time connected, bytes transferred, packets transferred, errors, etc.).

**Break Out Session - Transfer Model**Grant

A new name for the term link was discussed. Suggestions included shadow, pointer, reference, substitute, stand-in, view, and picture. There was also a discussion of how to represent a link better than the double chevron. Ideas mainly centered around using a different attribute for a link title. The Microsoft representatives didn't like this idea because the user can select the font and attributes of the title.

Rich

The Excel guy had strong opinions about transfer model. He was pretty vocal about the planned change being the wrong thing to do. The proposal is actually very close to what Excel does today. Apparently Excel is getting beaten up because 'Cut' doesn't behave the way most every other app works. It behaves more like the new 'Move'. Of course the simple change of name might be enough to eliminate the problem their customers are currently having.

Everyone in the group (but me) expressed opposition to the name link. Personally, I have not heard any suggestions that I like better. Besides, Office already uses the term link. :-)

**Break Out Session - Menu Behavior**Grant

Everyone liked the new tracking. There was discussion on whether or not clicking on a popout menu should execute a default action. I felt that it should not execute a default action but that it should require a double click to execute a default action like the main menu.

Rich

Everyone liked the new tracking. There was quite a bit of skepticism regarding executing default menu items on cascaded menus. Input was solicited on the desirability of tear off and or scrollable menus. There were a few people who expressed an interest in each, but no overwhelming desire for either.

**Break Out Session - Explorer/Browser**Grant

Our group was a little unorganized on this session. We talked about Cairo and SDI vs MDI.

Rich

During this session, we ended up talking mostly about MDI. The Excel guy felt pretty strongly that it would be a mistake to get rid of MDI. Steve Madigan (program manager for Cairo) disagreed very strongly. He is concerned that there is no model of containment today that a user can relate to. His example was two browser windows on the screen, each with a Word and an Excel document in it. Opening the four documents in any order always produces two additional windows, one containing both Word documents and the other containing the Excel documents. That really makes no sense at all. He indicated that there is work underway for Cairo that will likely allow association based on project or task, not based on the application that happened to create the object.

**Chicago UI Design from a Visual Perspective**

**Renee Marceau, Microsoft Windows Systems Visual Interface Designer**

Microsoft's visual interface designers now say that the black outlining of icons (that they had previously strongly recommended) cause visual clutter and unnecessary complexity. In one of the breakout sessions the designers were taken to task on this point. Their response was that "they discovered that the icons work without the black outline". They now recommend using black on lower and right edges while using a dark color on the upper and left edges. This helps provide a 3D effect for the icons.

In Chicago they intend to address the areas of the environment that are not visually consistent and intuitive (e.g., the system menu visual vs. the other caption visuals). They also intend to allow users to set all colors used by the system. It will also be possible to alter the scaling between physical and logical pixels. All system visuals are system drawn and scalable.

When released, Chicago should appear more refined and softer than today's Windows interface. They are removing a lot of unnecessary clutter in areas like resizable frames.

The difference between combo boxes where the user can enter text and those where the user can't will no longer be distinguished by a separation between the edit control and the button. Instead, if the user cannot type into the edit control, its background color will be gray (by default) while controls that can be typed into will have a white background (by default).

In Chicago, applications will want to provide both document and application icons in sizes of 32x32 and 16x16.

During the presentation, one of the Lotus representatives related a usability study they did in which they were addressing a confusion their users expressed over not knowing which buttons would dismiss a dialog and which wouldn't. They found that placing the buttons that would dismiss the dialog in a separate raised area at the bottom of the dialog produced dramatically improved user understanding.

**Chicago UI Design Guidelines****Tandy Trower, Acting Program Manager, UI Design Guide**

Tandy emphasized that the new style guide is still preliminary. We need to review it and get out input back to them. He also emphasized that the style guide presents guidelines, not rules. I suspect that he is trying to head off some of the debate that took place during the discussions on the first style guide.

The UI is changing to treat everything as objects. These objects are, or have, components (object are often composed of other objects), properties, and relationships.

The following types of relationships between objects were presented:

- Collection - a set of objects
- Constraints - a set where a change to an object affects another (e.g., layout)
- Composite - a set where the aggregation is a unique object (e.g., grouped objects)
- Containment - a set where one of the members is related, but independent, a place where a set "lives" (e.g., folder, document)

Mouse selection will change somewhat. You can use the right mouse button to do a drag select. On the mouse up, a context menu will appear for the selection. A Shift+click always

extends the last selection. It is no longer necessary to use both the Shift and Ctrl keys to extend a disjoint selection. There is currently no definition for what a double right mouse click should do. They are discussing possibilities including help and properties. Right clicking on a selection should not collapse the selection (this allows for a context menu on the mouse up).

They are evaluating having this same behavior on the left mouse button. They are also thinking about swallowing a right click that is used to dismiss a popup menu so that the selection does not change.

Tandy said that they have observed a problem where many users will open the same document multiple times because the previous window will become obscured. They are recommending that if the user attempts to open a document that is already open, the previously opened document should be surfaced rather than opening it a second time. There would obviously need to be some non-standard way of opening the same document twice for those rare cases where that is actually what you want to do. If a user attempts to open an application that is already running, they should be given the option to surface the running app or to open a second instance.

Property sheets were felt by Tandy to be a significant new aspect of the UI. Property sheets provide pages of information and tabs as the common form of navigation. In the presentation, Tandy was proposing that 'Apply Now' be a standard/common button on property sheets. There was a strong vocal reaction that the button should be 'Apply' and not 'Apply Now'. Tandy said that they are evaluating (and soliciting input on) whether there should be recommendation made as to when changes made to a page are committed. From the discussions in the breakout session, it seemed apparent that no recommendation could realistically be made. Some apps will commit changes on a page change and others will commit on the dialog's dismissal. Given this, there was a request made that the tab control be enhanced to support a visual clue that a page is 'dirty' (i.e., changes have been made that have not yet been committed).

It was stated during this discussion that Cairo will support extensible properties on objects and that Cairo will address per user views of objects. This information will be stored externally and will be accessed via a new OLE protocol.

Chicago will likely not provide property inspectors, but an application could do this on their own and Cairo will likely address the capability in the system. A property inspector differs from a property sheet in that it is like a modeless dialog within which you could display the properties of various objects. The Lotus Notes guy made some comments that gave the impression that they are planning this type of capability.

A number of window design changes are still being debated within Microsoft. Their questionnaire specifically requested the participants to choose amongst several possible title bar alternatives. They are trying to decide whether to include close and or restore buttons on the window title bar. The title bar will include an icon that represents the object (much like Office 4.0 does today). They were also soliciting input on what a maximized MDI child window should look like. The title bar caption is being reversed. It should contain the name of the document optionally followed by the name of the application. A minimized window will most probably look just like the left portion of the title bar (the width is under user control, but all minimized windows will have the same width).

MDI was a very popular topic. I got the impression that most participants want to see MDI go away. Several (including Lotus) specifically asked that Microsoft make a strong statement encouraging vendors to move away from MDI. Based on what Tandy and other Microsoft people said, no such statement will be made in the Chicago time frame. Chicago will definitely support MDI. The Cairo program manager talked quite a bit in the break out session about his desire to have Cairo support an evolution of MDI where windows from multiple applications will either be contained in some minimal workspace window, or will simply be related to one another in some visual way with common window management functions (e.g., put this project away, arrange the windows of this project). He even made the statement that he was surprised no ISVs have done something like this with their own application suite already.

### **Windows User Assistance**

#### **Marion Hogan, Chicago User Education**

Microsoft's usability studies and focus groups have shown that people have trouble finding the help they need. To attempt to address this, Chicago will make help more visible by adding a persistent access point to help on the desktop. There was some discussion in a break out session about providing a mechanism for applications to tie into that access point so that help for the active app/object could be found there. They intend to emphasize search and build much better keyword lists. They may even ship a way to fully index the help files, but the system help won't include this because of disk space limitations.

Microsoft has also found that the search dialog was hard to figure out. Users do not typically learn the most efficient way to find topics. In Chicago, the search dialog will be simplified and will be modeled after the standard Chicago property sheet.

The system help for Windows 3.1 has been found to be too lengthy and frustrating to read through. In Chicago, each control on dialogs will have a quick menu that includes a help menu item (they mentioned that it might be called 'Quick Info' or 'Quick Help', since we already call our context menus quick menus, we should change help to quick help or quick info). Selecting the help menu item displays a popup help window describing the control. The help writers at Microsoft are also taking a much more minimalistic approach to writing help. There will not be a lot to read through, and there will often be shortcut buttons in the help that take the user directly into the application dialog being described. The help will also focus more on tasks.

Microsoft has also observed problems when a user is reading help and they switch back to the application. This typically obscures the help window and causes frustration. To address this, they plan to use more secondary windows and give them the 'always on top' style. These windows will either be typically small, or dynamically sized so as not to obscure too much of the application window.

The performance of help has been a problem. They have reduced the startup time for help by 50%.

They have also found that users often have a hard time getting out of help. To address this, they have enabled the escape key to dismiss a help window and they have added a close button to the button bar.

The new help system will support interactive procedures by allowing help to establish a one to one link with an application so that help authors can write trackable procedures. This will be much like que cards (not like wizards or coaches).

Chicago will include a tour that teaches Chicago skills, a tips and tricks help for advanced users (they have had a lot of feedback that says users like this kind of stuff), and a User's Guide that includes conceptual information and troubleshooting information.

The new help compiler will have its dependency on MS-DOS conventional memory removed so that larger help files can be compiled without resorting to using OS/2. A problem with code page problems for non-english versions has been fixed. Support for PCX and other graphic formats has been added. Compression for graphics will be better.

### **How to be a great app in the Chicago Shell** **Joe Belfiore, Program Manager, Chicago Shell/UI**

The items presented here relate primarily to user interface issues and the shell. Other system related Chicago issues were not explicitly discussed.

- 1)Use 32 bit APIs. (Note: this is the one non-UI item that was mentioned several times)
- 2)Support long filenames
  - Open and save files with long names
  - Display them in title bars
- 3)Support UNC pathnames
  - Open and save files with UNC paths to enable direct network browsing
  - Display the "friendly form" in the title bar (My File on SERVER1)
  - Use correct sharing modes so multi-user scenarios work well
- 4)Make sure that documents/data files are accurately displayed and used in the shell
  - Put icons for document types in EXE and register them in the OLE database so they are displayed correctly in the shell
  - Add data-specific commands to the database for document types (e.g., "Play" for a sound). These will then be displayed in menus throughout the shell.
  - Make sure a document-window title bar displays the object a user double clicks (SDI parent or MDI child, as appropriate)
  - Support "print-to" in the registration database to enable drag/drop printing to specific printers in the shell.
  - Long extensions. (This sounded like a half-baked idea of registering long extensions for use in file type identification, and then only showing users the first three characters of the extension.)
- 5)Support drag and drop and other transfers consistently and extensively
  - Support OLE 2.0 drag and drop to enable users to move data to and from the desktop, tray or explorer and other applications. Shell transfers are interesting and important! (mail someone a link to a document, etc.)
  - Support non-default drag with mouse button 2, displaying a menu
  - Revise menu-based transfer model to work well with the one in the shell. Change "Paste" to "Put <object> Here", and consider replacing "Cut" with "Move". (Note:

there was much discussion on this point. Further elaboration can be found below. This is an area that we need to think about and provide input to Microsoft soon.)

- 6) Use the common dialogs, especially File Open
  - To be consistent with other applications and with the explorer itself, making the system easier to learn overall. (in breakout session, many said they would likely not use the common dialogs although they seemed concerned for same reasons that I am)
  - Get lots of functionality for free if File Open and Save As common dialogs are used
    - links (both enumeration and resolution)
    - direct browsing of the network
    - shell operations like drag and drop of files, creation of new folders, etc. via quick menus
- 7) Be careful about multiple instances of application being started too easily
  - Restore windows that are open rather than creating additional windows on the same document by default
  - Put up a list of currently open application instances if a user double-clicks the application icon a second time -- MS has seen people run out of memory by accidentally starting the same app over and over
  - Follow the Style Guide for details (Note: we have a hard copy of the new style guide in its current form, we do not have an on-line copy)
- 8) Maintain a consistent user interface and object paradigm between the application and the shell.
  - Use quick menus on mouse button 2 like the shell does (Note: the menu appears on button up to allow right button drag and drop)
  - Replace complicated "settings" dialogs in applications with property sheets where appropriate (a new control is provided to assist in this)
  - Take advantage of new system-supplied controls like the Toolbar, Status Bar, Spin Buttons, etc. (Note: 'Toolbar' is no longer trade marked by Microsoft)
  - If a delete function is provided by the application, the app should call the new Wastebasket API so that deletions to the user's Wastebasket instead of disappearing completely.
- 9) Extend the shell's ability to provide general information about application files (more OLE 2.0)
  - Use compound files as datatype and store the OLE 2.0 general properties like author, thumbnail, etc. in them so they will be displayed by the shell
- 10) Support pen input for pen notebooks and desktop tablets
  - Use pen APIs to activate the pen so that users can edit documents using gestures and input text using handwriting recognition
  - Use ink-edit controls to allow users to enter scribbled notes, drawings, and signatures
  - Enable "ink" annotation of documents using the OLE 2.0 annotation server
  - Add other natural pen-oriented features and gestures(Note: Chicago will make it possible to add a lot of pen functionality with little investment, or so it appears, and so they say)
- 11) Support Chicago-Style help
  - Use context-sensitive help popups in dialogs

- Add "Help" items to the quick menus
- Revise Help menu according to the Style Guide

12)Size / color scalability

- Custom controls need to get 3D colors and size metrics from system
- Test application when logical/physical scaling is altered. Using a TrueType font for dialogs can help a lot here (Arial 8 is the preliminary suggestion)

13)Avoid drawing custom title bars

14)Don't draw into minimized windows

### **Cairo User Interface Directions**

#### **Steve Madigan, Group Program Manager, Advanced Systems User Interface**

Steve stated that Cairo is being designed as a framework for extensibility. He emphasized that Cairo is intended to address cooperative computing issues at the system level.

Today, most personal computers are used by information producers. Cairo is being designed for information consumers. Cairo will incorporate technology advances that won't initially be seen in Chicago.

In Cairo, OLE is the model for extensible services, period. OLE is even more extensively used than it is in Chicago. Improvements to OLE include moving links with an object automatically, distributing OLE across the network, making programming interfaces extensible, and storing OLE naively in the object file system. The desktop is an OLE container.

Some things added by Cairo that Chicago will not have are, query support (everything in the file system is indexed), distributed processing, and tools.

In Cairo, a property set can have an associated behavior. The example of e-mail was used. An e-mail message is any object that has the e-mail property set. That property set might be added to an object by dropping the object in an Out Box.

### **Break Out Session - User Assistance**

#### Grant

I ask that they consider including an application with the PDK release the shows the help style they are recommending.

#### Rich

There was discussion, but no real answers, regarding how online material relates to printed material. Both the Aldus representative and the Symantec representative expressed a lot of interest in this.

Given their proposed model for the Help window using property sheets, I asked if they had thought of compressing the help menuitems to one, and if so, then is there any need for a help menu at all or could active applications tie into the persistent help icon on the tray. They seemed interested in this idea, but still uncertain as to what their help menu recommendations are going to be.

### **Break Out Session - UI and Visuals**

#### Grant

Tandy emphasized that Microsoft would like feedback on the style guide. There was general agreement that the new visuals are an improvement. Dialog button placement will still follow the guidelines in the 3.1 style guide except that for tabbed dialogs bottom button placement seems to work the best.

#### Rich

The graphics designers denied my accusation that they must have received a NeXT system for Christmas. They definitely seem to be struggling with the idea of putting a close button on the caption. Tandy mentioned that they are the only significant GUI that does not do this.

### **Break Out Session - MDI**

#### Grant

SDI is the model for Cairo. The Microsoft representatives were asking what obstacles we saw in getting our applications to SDI. We also talked a little about the Visual Basic model - this does not appear to be a model they are pursuing for general use.

#### Rich

This was my last session on Friday and it was after 5:00. There were only three of us left and there were three of them. We mostly just talked about Cairo and the Cairo demo they gave that day.

### **Further Information**

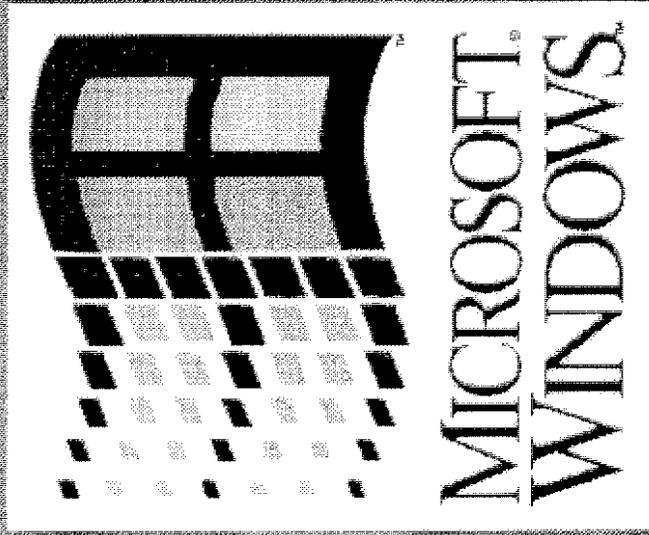
Rich Hume and Grant Skousen have the following documents and software. If you need more information than is provided in this document something in this list may be of help to you.

- *Slides from Chicago User Interface Design Preview*, July 8-9. This was the basis for this document. It contains visuals of most of the new UI.
- *User Interface Design Guide for Microsoft Windows (Chicago)—Preliminary Draft*. We need to get comments on this back to Microsoft as soon as possible
- *Slides from Chicago Overview* presented at WordPerfect by Glenn Thompson of Microsoft, July 2. This is a high level overview of Chicago.
- *Application Compatibility in Future Versions of Windows*, dated March 24, 1993. This document points out what must be done to insure current applications are compatible with Chicago. This document was passed out at the Chicago Overview.
- *List of Win32c APIs*. This is a soft copy of a Excel spreadsheet listing all of the Win32c APIs. This was also passed out at the Chicago Overview
- Alpha Chicago CD. This copy is missing the new shell and many other features of Chicago. Installation on at least one machine has caused the loss of the FAT. It may be best to wait till the PDK release to spend time looking at Chicago.

**PX 113**

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

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



Microsoft Confidential

CONFIDENTIAL

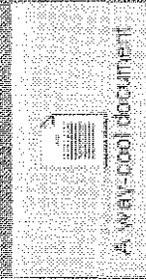
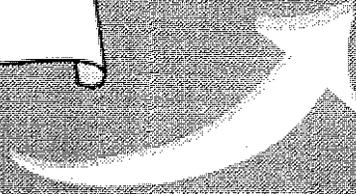
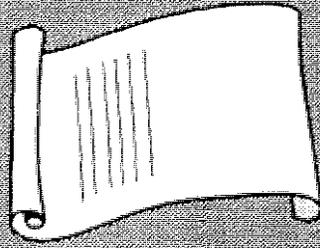
EXHIBIT # 19  
WITNESS Gates  
DATE 5-19-04  
ZOYA SPENCER (206)382-9495

NOV 00734371

For Baltimore BE-93197 PDC 1203 2

# 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



Microsoft Confidential

Job: Bellicore-BS395197 PDC 12/06/03

# 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

Microsoft Confidential

Loc: Bellcore-BE093107-PDC-1293-4

# New Controls

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

Microsoft Confidential

Job: Belfiore-BL293197 PDC 12/21/5

# 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	Attr's
255color.bmp	6K	Paintbrush Picture	10/01/92 03:11AM	10/01/92 03:11AM	A

Microsoft Confidential

Los Angeles, BE-59-197-PDC 1295-6

# New Controls - 2

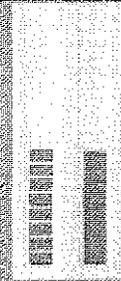
Slider



Spin buttons



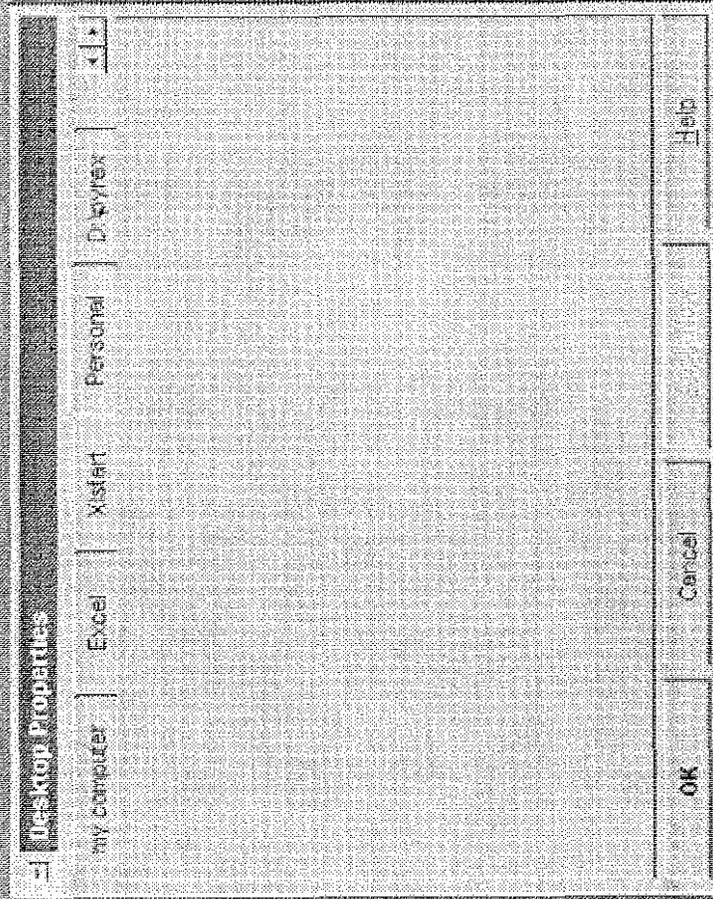
Progress indicator



Microsoft Confidential

# 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**

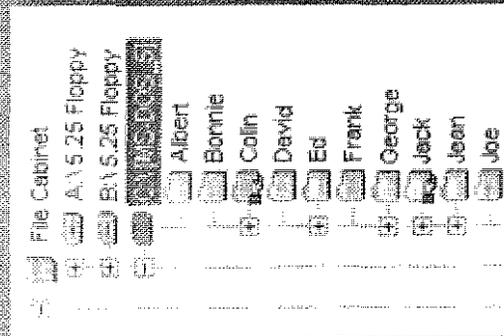


Microsoft Confidential

Inv. Bellcore BE193197 PDC 12/93 S

# New Controls - 4

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



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

Microsoft Confidential

Joe Bellon: B1399197 PDC 12/03/09

# 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

Microsoft Confidential

Joe Bellion BEL98197 PDC 12/93 10

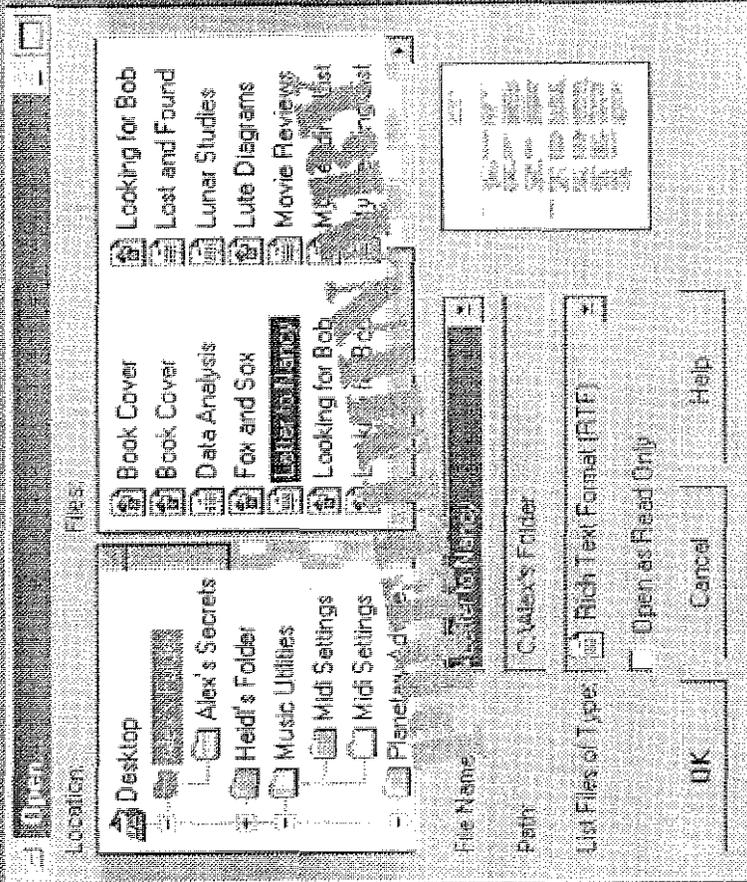
# Common Dialogs

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

Microsoft Confidential

Joe Bellmore BE593197 PDC 12/93/11

# File Open/Save As

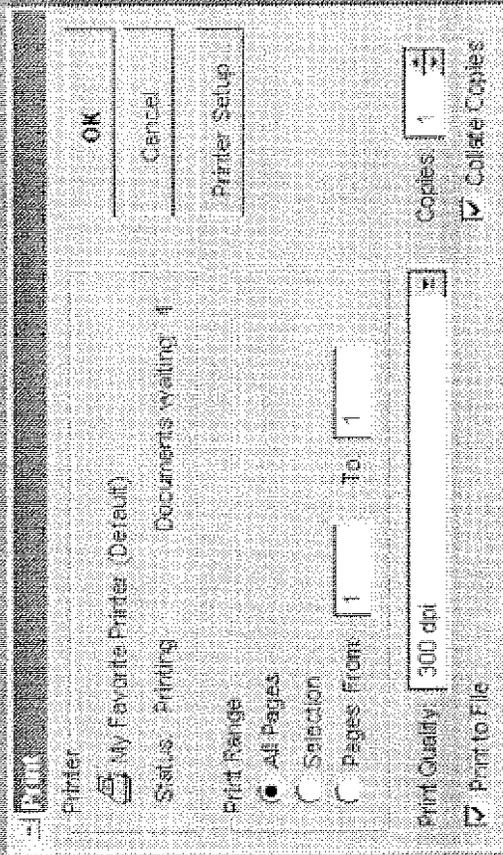


- ◆ Explorer functionality
- ◆ Direct browsing of the network
- ◆ Displays and handles links properly
- ◆ Context menus available on files and background, allowing fs operations
- ◆ Extensibility similar to Windows 3.1

Microsoft Confidential

# Print

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



Microsoft Confidential

Job: Balfanz BE393197 PDC 12/28/11

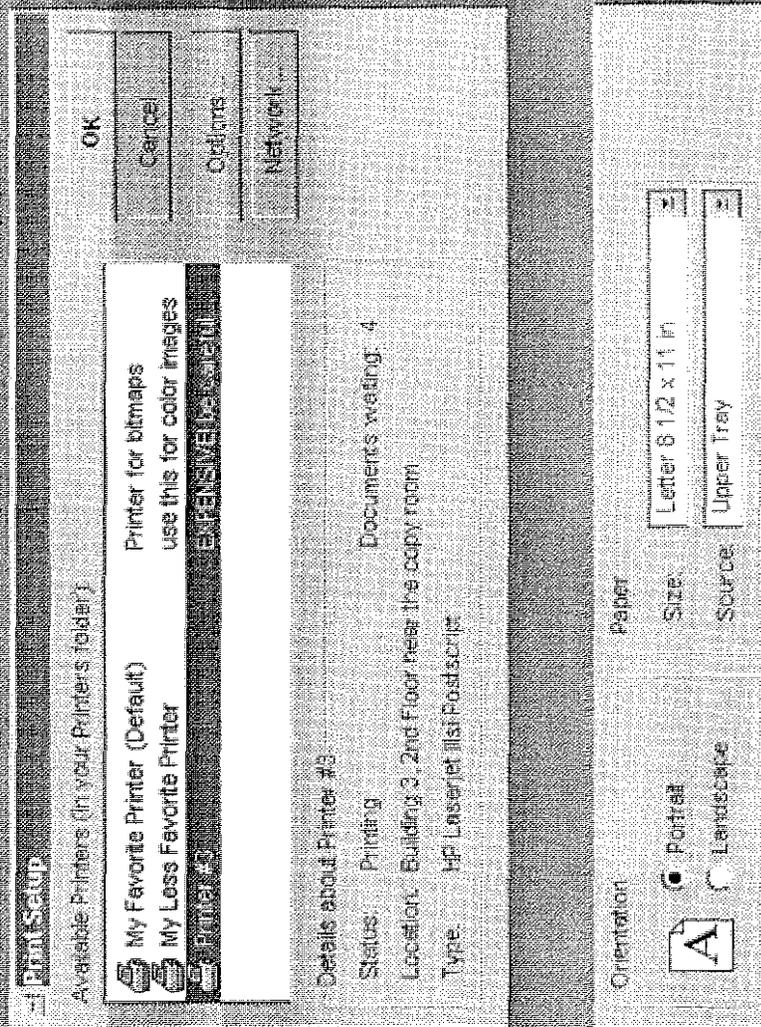
# Choose Printer (Print Setup)

## ◆ 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



Microsoft Confidential

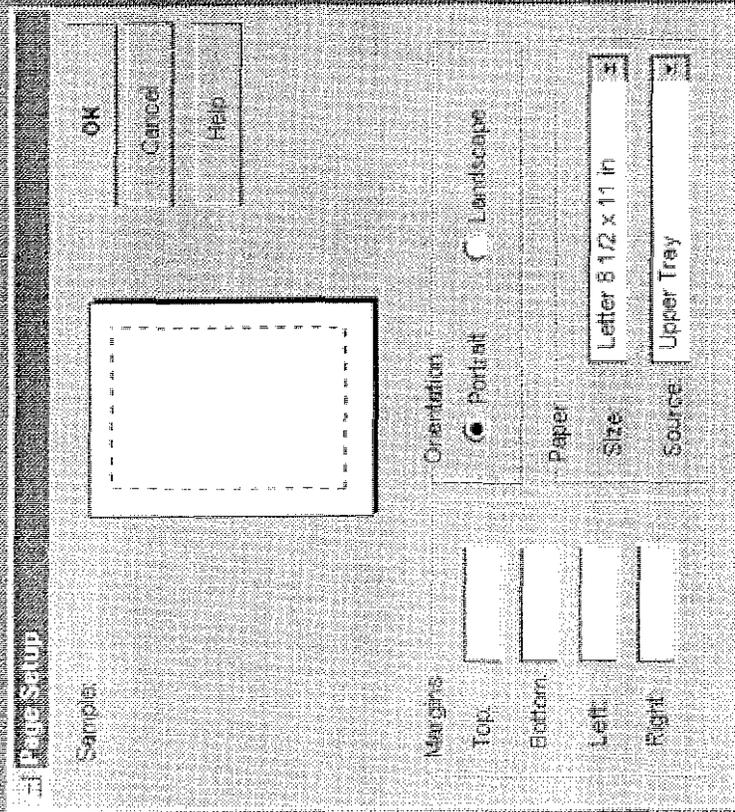
File: BelProc: B1395197 PDC: 12/02/14



# 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

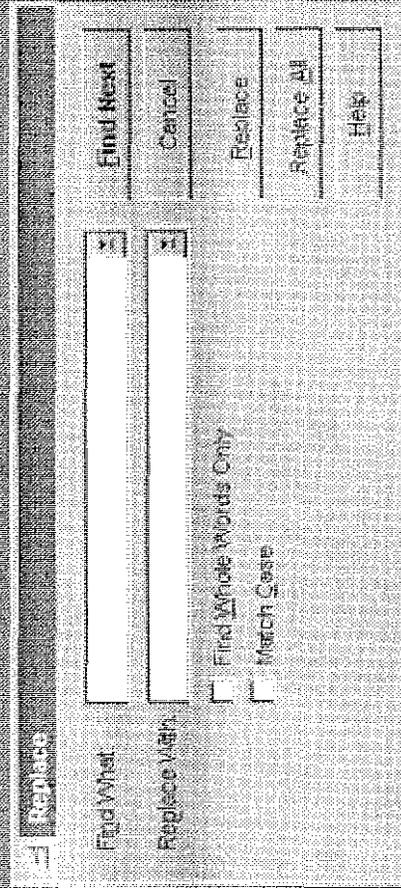
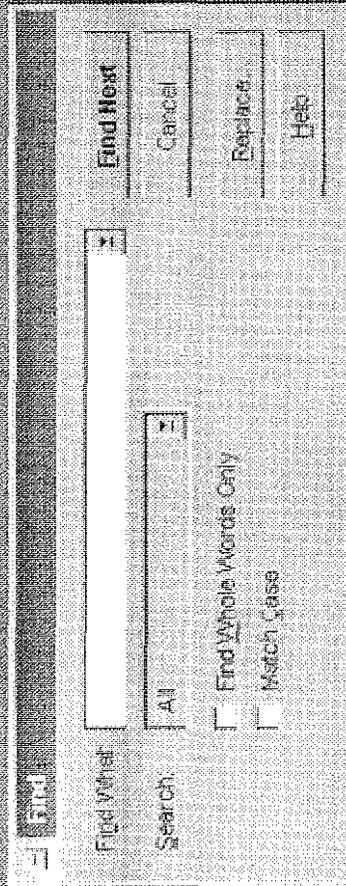


Microsoft Confidential

Joe Balfiore BE593197 PRC 12/05/15

# Find/Replace

◆ Very similar to Windows 3.1

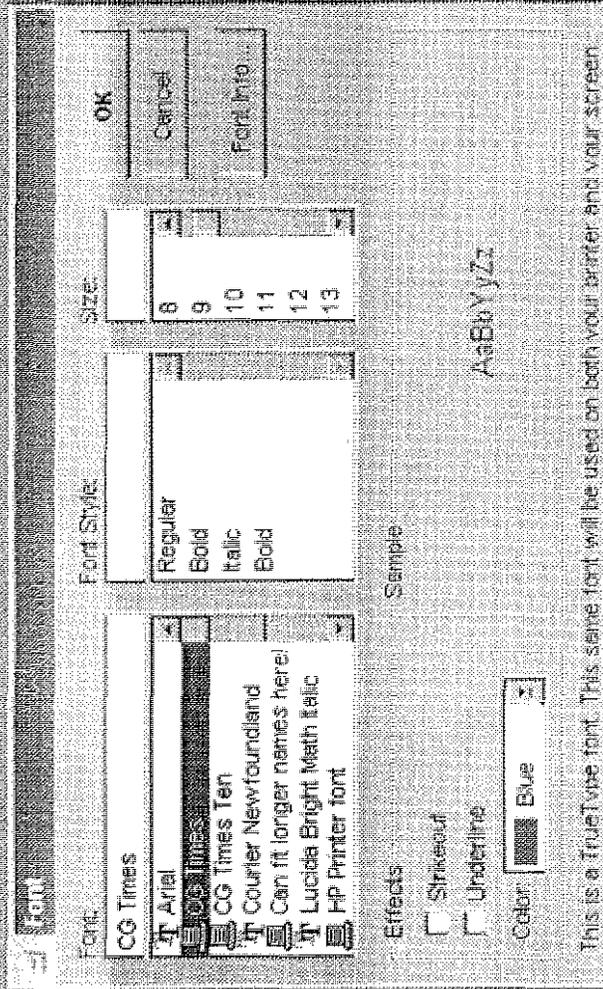


Microsoft Confidential

log: Bellcore-BE-394197-PDC-12/93-116

# Fonts

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

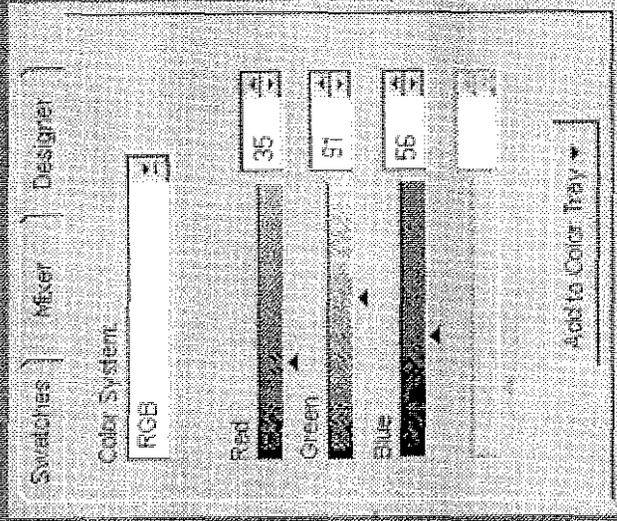
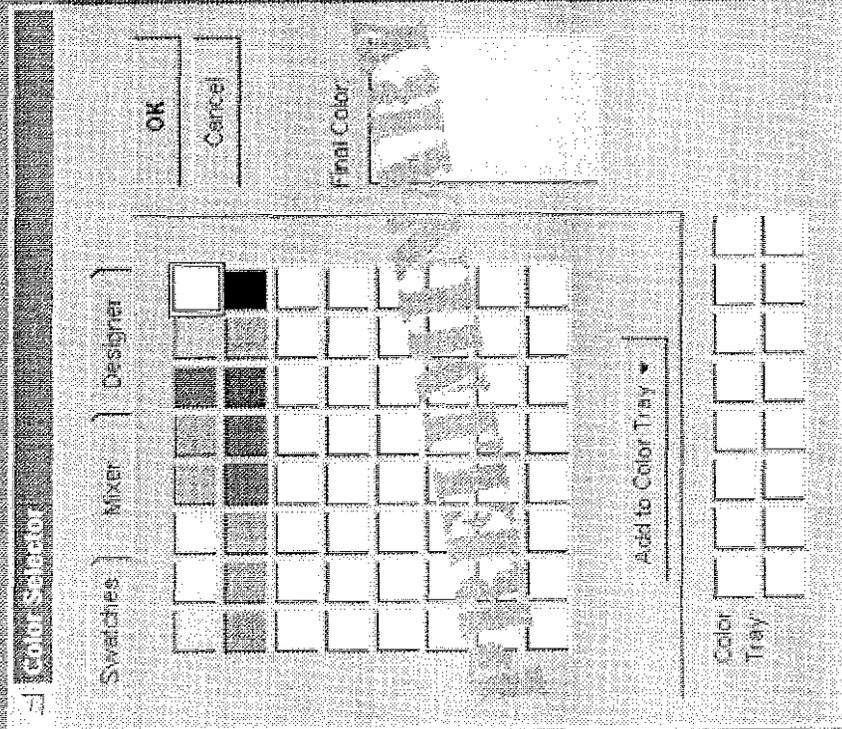


Microsoft Confidential

In: Bellion-BL2003197-PDC 12/95-17

# Color

◆ Design still underway



Microsoft Confidential

Doc. Ref: Belfone BE307197 PDC 1295 18

# 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!!

Microsoft Confidential

Doc RefId: BE505197 PDC 1203 19

# Shell Extensibility

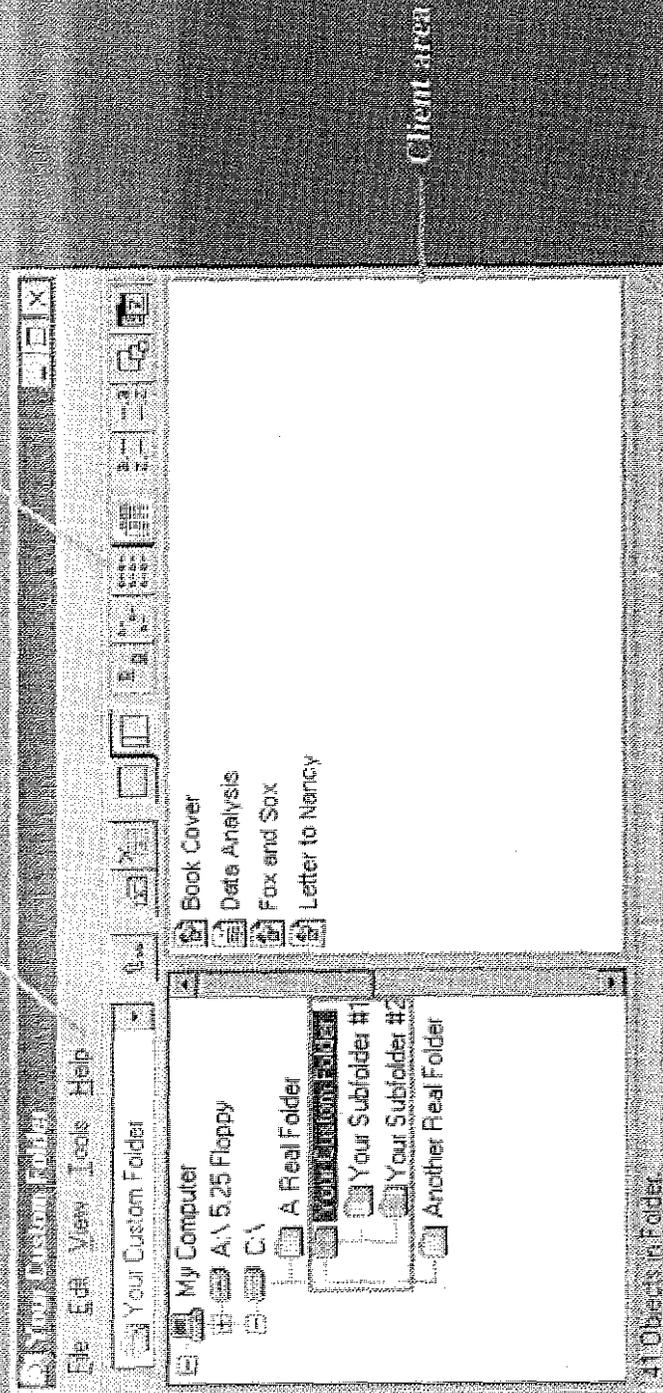
- ◆ OLE 2.0 integration
  - > Drag-and-Drop feature
  - > Summary properties shown by default
  - > Verbs you add to the registry will be displayed on object menus
- ◆ Property sheet extensibility
  - > Add your own property sheet pages to objects on a per-class basis
- ◆ Viewers
  - > Hook your own 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*

Microsoft Confidential

Doc. Bellcore BE293197 PDC 12/93/20

# Explorer Integration Details

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



Microsoft Confidential

Joe Balfiore BE993197 PDC 129521

# What About MIDI?

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

Microsoft Confidential

Doc Belfiore BE-93197 PDC 12/03/12

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

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

Microsoft Confidential

Doc RefId: B539197 PDC 12932

# 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 commandg “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

Microsoft Confidential

Jac Bellone BES93197 EDC 12/05/24

# 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)

Microsoft Confidential

**PX 297**

# WORLD REPRINT

► Product Reviews: Office suite

# PerfectOffice nearly lives up to its name

■ Novell's revamp of Borland Office isn't perfect, but it's the best of the high-end office suites

By Patrick Marshall

**P**ERFECTION, the Hopi Indians of the 18th century believed, would be an offense to the gods. For that reason, they intentionally wove an error into each blanket they created. To read its ads, Novell Inc. isn't worried about offending anyone with its claims that PerfectOffice Professional 3.0 ("the perfect place to work") is flawless. To hear Novell tell it, PerfectOffice — Novell's revamp of Borland Office, acquired from Borland International Inc. — is the best set of integrated applications to come down the pike yet.

It's true. PerfectOffice is a terrific assortment of programs that offers more integration than we've seen so far in any high-end office suite. It not only does an admirable job of supporting OLE 2.0, but it offers prebuilt macros for use across applications. Overall, it's far superior to Borland Office, and it's a better-integrated program than either Lotus Development Corp.'s Lotus SmartSuite, Release 3.0 (reviewed Nov. 21, 1994, page 106) or Microsoft Corp.'s Microsoft Office Professional, Version 4.3, the highest-scoring office suite we've evaluated so far. (See our comparison of Windows office suites, June 27, 1994, page 104.)

But perfect? No. PerfectOffice is almost as big a system resource hog as Microsoft Office, and there are still a few data sharing kinks to work out. Other flaws, although relatively minor, are so obvious that Novell seems to have left them in intentionally. The gods should be pleased.

For this review, we followed the test plan developed for the June 27 comparison.

### FEATURES

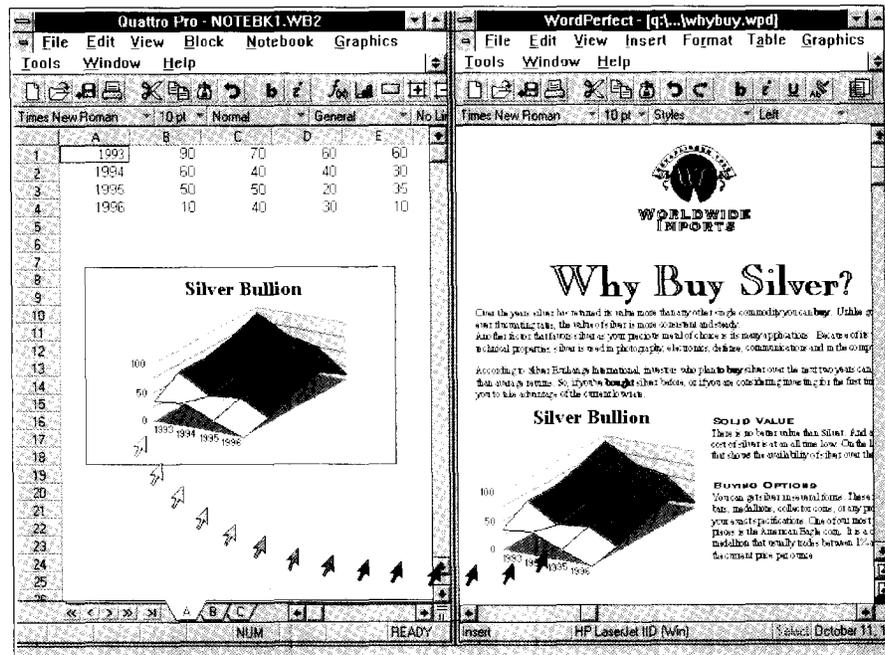
PerfectOffice Professional 3.0 bundles a broad array of first-rate applications, including the newly revitalized WordPerfect 6.1 for Windows, Quattro Pro 6.0a for Windows, and Paradox 5.0 for Windows. In addition, PerfectOffice features Presentations 3.0 for Windows, a presentation graphics application; InfoCentral 1.1 for Windows, a personal information manager; Envoy 1.0a for Windows, a file viewer; and a client license and demonstration version of GroupWise 4.1a, an E-mail, calendaring, and group-scheduling product. PerfectOffice also includes AppWare 1.1, a visual development tool that allows you to create and customize applications. (The Standard version of PerfectOf-

*PerfectOffice is a terrific assortment of programs that offers more integration than we've seen so far in any high-end office suite.*

fice lacks AppWare and Paradox.)

All of the suite's major applications share a strikingly common interface, and all support OLE 2.0. On the workgroup front, PerfectOffice retains Borland Office's built-in transport, Object Exchange (OBEX), for sharing embedded data among users, a facility that competing suites can't match.

PerfectOffice is also the only suite to offer a scripting language for recording macros that work across applications. Novell bundles a batch of prerecorded macros — dubbed QuickTasks — for accomplishing a variety of chores, such as creating a slide show from a WordPerfect outline or moving a list of names and numbers into InfoCentral's phone book. In addition, PerfectOffice saves disk space and ensures a consistent interface by sharing more tools across applications than any of the other suites do.



**WE HAD NO TROUBLE PASTING SPREADSHEET DATA** into a WordPerfect document.

PerfectOffice's Paste Special command let us selectively copy cells and charts as linked tables and charts, or as images in Windows metafile, .DIB, or .BMP formats. With only a few exceptions, we found that WordPerfect could resize and otherwise modify the data without distortion or mishap.

## PRODUCT REVIEWS

We encountered a few irregularities, particularly in Paradox's interface. For instance, the File/Open dialog boxes are identical in all the applications — including WordPerfect, Quattro Pro, and InfoCentral — except Paradox. And instead of providing pop-up cues to icon functions, Paradox offers explanations in the status line at the bottom of the screen. Paradox is also the only application in the suite that doesn't offer GroupWise as an option under the File menu.

Minor inconsistencies among other applications exist as well, such as the different Format/Font dialog boxes in WordPerfect and Presentations. Quattro Pro has no Format menu at all. The Print dialog boxes also vary considerably from one application to another.

All told, PerfectOffice's interface is not seamless. But it achieves a level of harmony and integration equal to that of Lotus SmartSuite, Release 3.0. We rate common interface very good.

### Data sharing

To test the suite's data-sharing capabilities, we performed a variety of operations. First, we tried to simply copy and paste data from one application to another via the Clipboard. Next, we used the special pasting commands to insert data in various ways (for example, OLE embedding, OLE linking, pasting material as a bit map, and so on) in each of the suite's applications.

PerfectOffice owes its improved showing in this category primarily to its QuickTask macros and support for OLE 2.0. You can use OLE to exchange data among all five of PerfectOffice's major applications — WordPerfect, Quattro Pro, Paradox, Presentation, and InfoCentral — a capability no other office suite can offer. Moreover, PerfectOffice's implementation of OLE 2.0 is relatively free of glitches, especially when compared to Borland Office's implementation. One caveat: When we ran more than two applications, the suite's heavy demands on resources caused frequent crashes resulting from low memory.

PerfectOffice makes simple copy-and-paste operations easy by anticipating the most likely format in which you want the data pasted. For example, the program automatically pastes a Quattro Pro cell into WordPerfect as a table. (Unfortunately, it fails to retain font formatting, with awkward results.)

The handy Data Sharing DAD has icons for cutting, copying, and pasting data, as well as for copying to WordPerfect data currently selected in Presentations, Paradox, or Quattro Pro. You can edit this toolbar to include other icons as well.

The suite made short work of our spreadsheet-to-word-processor OLE data-sharing tasks. The Paste Special command allowed us to copy the cells and charts selectively as linked tables and charts, or as images in Windows Metafile, DIB, or BMP formats. With only a few exceptions, we found that WordPerfect could resize and otherwise modify the data without distortion or mishap.



**GOOD OL' DAD.** PerfectOffice's floating command/application toolbar, the Data Application Director (DAD), is the easiest way to hop from one suite application to another. You can create your own, customized DADs for different types of work.

We had real difficulties, however, moving data from Paradox to WordPerfect using the Paste Special command. First, Novell does not provide any linking options for use in updating data automatically from Paradox, unless you select the entire table. Our only options were to paste the data in as unformatted text or in the Windows Metafile, DIB, or BMP formats. The Paste Special dialog box even indicated that it couldn't identify the source of the data. When we used the image file options, embedded images appeared distorted and awkwardly sized.

Fortunately, a QuickTask macro made it very easy to bring Paradox data in as an unlinked table. WordPerfect's Insert/Database option also allows you to bring in Paradox files (as well as a wide variety of other database and spreadsheet files) as linked or unlinked tables. Unfortunately, using Insert/Database doesn't format the tables to fit the document (cells run into the margins and the spreadsheet's fonts won't match your document's.) On the plus side, the simple fact that you can edit Paradox tables in place in WordPerfect documents gives PerfectOffice a big advantage over the competition.

WordPerfect's mail-merge utility makes it easy to grab data from Paradox (or other databases) to create form letters or other docu-

ments. When you activate the utility, the program pops up a dialog box for specifying the data file and the document file. Then, with the document file on the screen, you can use the merge toolbar to insert fields in the document — selecting from a list of all available fields in the data file — and perform the merge.

PerfectOffice handled the first of our spreadsheet-to-presentation tasks — that of inserting a graph from the spreadsheet and modifying it in the presentation package — without a hitch. When we inserted the graph as a Quattro Pro graph, we automatically had access to all of Presentations' editing tools. But we were unable to paste a selection of Quattro Pro cells into Presentations and then turn them into a chart.

Given the problems that all office suites still have in sharing data, none of them earn a top score in this category. But PerfectOffice sets a new standard. We rate it very good.

### Workgroup capabilities

Not only do office suites allow users to share data among applications on a stand-alone machine, but the most progressive ones provide tools — such as E-mail enabling — for sharing information among multiple users.

Like Borland Office, PerfectOffice earns a high score in this category thanks mostly to its inclusion of OBEX, a workgroup tool that surpasses anything available in Lotus SmartSuite or Microsoft Office.

It's not the smoothest workgroup utility we can imagine, nor the easiest to set up, but OBEX works. Like Apple Computer Inc.'s Macintosh Publish and Subscribe feature, OBEX allows a server application to "publish" an object — which must be either a WordPerfect document, a Quattro Pro notebook page, or a Paradox query — and other users to "subscribe" to the object. The subscriber can specify whether subsequent updates should be automatically incorporated or whether the program should wait for manual updates. In addition, the publisher of data can specify the number of different versions that subscribers can maintain and access.

In workgroup scenarios, OBEX has an advantage over OLE in that it does not require that data objects and their source applications reside on the same drive. For example, a subscriber need not have Quattro Pro installed in order to view and use locally a published Quattro Pro object. You can't perform spreadsheet operations on the object, but you can, for example, insert the object as a table in WordPerfect and edit it.

You can configure OBEX to distribute data objects over a wide variety of media, including network transports, E-mail, and MCI Mail. You can also stipulate different methods of data transport for different users.

PerfectOffice includes one client license and demonstration version of GroupWise, Novell's powerful E-mail and group-scheduling program. You can directly use all of GroupWise's functions — which includes using your inbox, your calendar, and your address book — from

# PRODUCT REVIEWS

## PERFORMANCE

### Installation and configuration

PerfectOffice is much easier to install than its predecessor, Borland Office. We encountered no error messages or insurmountable complications. The setup routine made it easy for us to select installation drives by letting us view the amount of space required and the amount available on each drive. PerfectOffice also afforded a high degree of control over which applications or parts of applications we installed, a good thing because even a minimum installation consumes 68MB of hard disk space. A full installation needs 138MB of space, the most of any office suite yet. For those reasons you'll also appreciate PerfectOffice's new uninstall utility, which will remove applications — including files located in Windows directories — that you no longer want on your drive.

We have only a couple of gripes. We were a tad put off not to get an on-screen explanation of what the modules do. (Fortunately, we were able to quickly summon an explanation by hitting the Help button.) A more serious complaint is that a standard installation does not install the suite's Paradox application or, hence, the OBEX files — to do that you'll have to choose the custom installation option and install a separate set of Paradox disks. Then PerfectOffice will select the appropriate OBEX files for installation. We were not able to find any reference to this in the suite's installation documents or its on-line help.

Once you have installed OBEX, you can create OBEX address accounts with relative ease if you have a GroupWise, Messaging API (MAPI), Message Handling Service, or cc:Mail system established.

When installed on a network, PerfectOffice grants an unusual degree of control to network

## Product summary

### PerfectOffice Professional

Version 3.0

Novell Inc.

Orem, Utah

(800) 451-5151 or (800) 321-4566;

fax: (801) 222-5077

PerfectOffice has leaptfrogged Microsoft Office Professional and Lotus SmartSuite to become the most tightly integrated office suite on the market. In the increasingly critical area of workgroup support, PerfectOffice's Object Exchange (OBEX) data sharing technology makes the suite an especially attractive choice.

**List price:** \$859 for Professional version; \$659 for Standard version, which does not include Paradox or AppWare. An upgrade to the Standard version costs \$259; an upgrade to the Professional version costs \$359.

**Requires:** 386 or higher processor (486 recommended); 8MB of RAM; 68MB of disk space for a minimum installation of the Professional version; Windows 3.1 or later.

**Pros:** Cross-application scripting; OBEX data exchange technology; homogenized interface; support for OLE 2.0 in all major applications.

**Cons:** Resource hungry; Paradox not fully integrated.

administrators. The administrator can allow users to define their own environments. Alternatively, to enforce a standard interface, the "Corporate model" installation procedure allows the administrator to control the configuration of each workstation installation, including the setup of toolbars, templates, and printers. In addition, savvy administrators and users can use AppWare to create custom applications or customize included applications.

In either single or networked installations, you can add applications and cross-application macros to PerfectOffice's floating command/application toolbar, the Desktop Application Director (DAD), and you can create multiple DAD toolbars for different types of work. For example, Novell supplies a prefabricated Data Sharing DAD that helps you move and share data among applications. You can also automatically create a new DAD toolbar from any program group.

PerfectOffice is generally easy to install and uninstall, and it earns high marks for customizability. The confusing Paradox and OBEX installation requirement, however, holds its score to good.

### Common interface

Novell has made great strides in providing a standard interface across PerfectOffice's applications, correcting many of Borland Office's old incongruities. When you move from one application to another, you will generally find identical icons, as well as familiar dialog boxes, menu structures, and options.

You can use the intuitive icons on PerfectOffice's flexible, customizable DAD toolbar to move quickly from one application to another. One DAD icon lets you swiftly tile all open PerfectOffice applications, a nice touch. Another icon, for QuickTasks, gives a list of available macros for performing operations across applications. You can also create new macros.

We were a bit disappointed to find that you still can't float the DAD toolbar into the title bar of the current application. If you float it at the top, it obscures the title bar and control boxes of a full-screen application. You'll want to float it as a bar along the bottom or either side of the screen, or as a box, which can go nearly anywhere. You can also use the Auto Hide feature, which lets you specify where on the screen you wish the DAD to appear; when you move your cursor to that area, the DAD pops up. It disappears when you move the cursor off that area.

## Report Card

### Office suites

Criteria	Weighting	PerfectOffice Professional Version 3.0		Microsoft Office Professional Version 4.3		Lotus SmartSuite Release 3.0	
<b>Performance</b>							
Installation and configuration	100	Good	62.50	Very Good	75.00	Very Good	75.00
Common interface	150	Very Good	112.50	Good	93.75	Very Good	112.50
Data sharing	150	Very Good	112.50	Good	93.75	Very Good	112.50
Workgroup capabilities	150	Very Good	112.50	Good	93.75	Satisfactory	75.00
Application sharing	75	Very Good	56.25	Satisfactory	37.50	Very Good	56.25
Applications	75	Very Good	56.25	Very Good	56.25	Very Good	56.25
Resource utilization	50	Satisfactory	25.00	Very Good	37.50	Satisfactory	25.00
<b>Support and value</b>							
Documentation	75	Good	46.87	Good	46.87	Good	46.87
Support policies	25	Very Good	18.75	Very Good	18.75	Very Good	18.75
Technical support	50	Good	31.25	Satisfactory	25.00	Good	31.25
Value	100	Very Good	75.00	Good	62.50	Very Good	75.00
<b>Final score</b>		<b>7.0</b>		<b>6.4</b>		<b>6.8</b>	

\* Lotus SmartSuite 3.0 reviewed Nov. 21, 1994, page 106. Microsoft Office Professional 4.3 reviewed June 27, 1994, page 104.

WordPerfect, Quattro Pro, and Presentations, as well as from the DAD toolbar. You can update your calendar information and send messages in InfoCentral via GroupWise, but you cannot receive messages or files from within InfoCentral. As already mentioned, Paradox doesn't work with GroupWise at all.

PerfectOffice's support for traditional E-mail is not quite so strong. WordPerfect, Quattro Pro, InfoCentral, and Presentations are the only applications in the suite that provide direct access to VIM-compatible or MAPI-compatible E-mail.

PerfectOffice gets a few extra points for including Envoy, a distributed file viewer designed to allow users to see files in their original format even if they do not have the source application. Envoy does a poor job of reproducing fonts (see our review of Envoy, Oct. 10, 1994, page 110), but it's fairly useful for distributing documents.

PerfectOffice's workgroup capabilities still leave room for improvement, but no other suite provides so much built-in workgroup functionality. We rate workgroup capabilities very good.

#### Application sharing

The more applications and utilities a suite's applications share, the fewer resources are required and the more consistent is the interface.

PerfectOffice offers significantly more shared applications than Borland Office did. In addition to both a standard spelling checker and a QuickCorrect on-the-fly spelling checker (which checks your spelling as you type), PerfectOffice provides a thesaurus, the Grammatik 6 grammar checker, a drawing module, and file management tools.

All the suite's major applications except InfoCentral and Paradox can share the thesaurus, grammar checker, and drawing module. You can access the spelling checkers and file management utilities from anywhere within the suite by using the DAD toolbar.

The QuickOpen file management utility consists of a simple list of your most recently opened files, providing quick access to them. QuickFiles not only lets you locate and open files, it also includes copying, deleting, printing, and viewing tools. (With the exception of Paradox, each application's Open dialog box also offers all of QuickFiles' capabilities.)

QuickFiles also features a QuickFinder option that lets you index all your document files and search for terms in them using Boolean operators. This slick and easy-to-use utility makes it a snap to locate files.

We rate application sharing very good.

#### Applications

In this category, we judged the strength of the suite's individual applications relative to those offered by the competition.

The PerfectOffice application bundle is strong overall, although only one program —

## PerfectOffice has leapfrogged Microsoft Office Professional and Lotus SmartSuite to become the most tightly integrated office suite on the market

Paradox for Windows 5.0, which won our March 20 comparison of Windows databases — has earned the highest *InfoWorld* score in its product category. WordPerfect for Windows 6.1 and Quattro Pro for Windows 6.0a are second only to their Microsoft counterparts (Word for Windows 6.0 and Excel 5.0). Presentations 3.0 is a fairly impressive presentation graphics program.

InfoCentral 1.1 is a low-end but easy-to-use personal information manager. GroupWise 4.1, an E-mail program we've ranked just below cc:Mail, is not especially easy to set up, but it provides group scheduling and several other unique features.

Of the most doubtful value is Envoy 1.0, a "portable document" package that enables file sharing but doesn't reproduce fonts very well. Envoy still scored a respectable 6.0 in our Oct. 10 review, however, due to the tools it provides for distributing and manipulating documents. AppWare 1.1 is an innovative graphical application development environment best suited to programming novices.

On balance, PerfectOffice offers very good applications.

#### Resource utilization

To measure how effectively the suite uses Windows resources, we performed a common set of tasks to mirror a typical user's experience. The tests showed performance degradation (if any) over the course of a typical daily encounter with the suite's applications.

Our resource utilization tests knocked PerfectOffice off its pedestal. The package used all but 36 percent of our Graphics Device Interface (GDI) resources, a level that can result in errors when performing memory-intensive OLE operations. That's almost as bad as Microsoft Office, which allowed resources to drop even farther — to 32 percent. By comparison, Lotus SmartSuite 3.0 never let GDI resources fall below 62 percent.

PerfectOffice is also quite demanding of disk space, requiring 68MB for a minimum installa-

tion of the Professional version and a whopping 138MB for a full installation.

We rate resource utilization satisfactory.

#### SUPPORT AND VALUE

##### Documentation

Novell provides an 83-page Up and Running Guide that covers the suite's special features, including DAD and OBEX. The manual is generally well written, but it doesn't offer enough details on certain topics, most notably OBEX. Instead, it expects the user to pick that information out of the documentation provided for individual applications. The suite wins extra points, however, for strong on-line help, pop-up cues, and a very useful QuickTour that familiarizes users with features.

On balance, we rate documentation good.

##### Support policies

Novell backs PerfectOffice with a 90-day money-back guarantee and 180 days of free — and toll-free — telephone support, 7 a.m. to 6 p.m. Mountain time, starting with the first call. After the 180-day period, users can choose from a variety of paid support plans.

Novell also offers free support via its own BBS, CompuServe, SpaceWorks (a national BBS), and a fax-back service. Novell's support policies earn a score of very good.

##### Technical support

We never had to wait long to reach the technical support staff, and we found them knowledgeable and eager to help. They answered all of our questions accurately and without delay. We rate technical support good.

##### Value

For its revamp of Borland Office into PerfectOffice, Novell raised the list price of the suite from \$595 to \$859. That places PerfectOffice Professional smack between Microsoft Office Professional (\$899) and Lotus SmartSuite (\$795), neither of which has changed in price. The standard version of PerfectOffice, which lacks Paradox and AppWare, lists for \$659.

PerfectOffice is more than worth the price increase. It costs a bit more than Lotus SmartSuite, but PerfectOffice is a better choice in several important ways. It does a better job of sharing information across applications or a workgroup and has a more consistent interface. PerfectOffice costs a little less than Microsoft Office and performs as well as or better in the same areas, particularly in sharing data among multiple users. All the suites have a strong set of applications, but PerfectOffice's are integrated the best.

We rate value very good.