

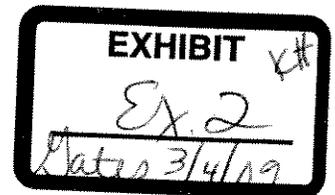
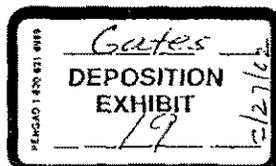
PX 51

Group 1 Topic

- How can Chicago increase sales of MS Apps?
- How can Chicago increase market share of MS Apps?
- How can MS Apps increase sales and market share given expected high penetration by Chicago?

Team Members: MikeMap, JonL, TandyT, SteveM, DavidCol, ChrisGr, EdF, NathanM

Note: Client-Attorney Privileged Material



MS-PCA 2535283

CONFIDENTIAL

How we think of the Problem

- How can we motivate upgrades in a semi-stagnant user base?
- How can we leverage the systems OEM channel? (more effectively than bundles!)
- How can Chicago and MS Apps cooperate to differentiate MS Apps on Chicago?
 - Design for App differentiation, prioritize development
 - »Chicago and Apps
 - Product Packaging / Marketing / Business Approach
 - »Think of Chicago as a channel for apps product and marketing
- Before and after MS Chicago App versions release

MS-PCA 2535284

CONFIDENTIAL

Upgrading

- **Letting people know about upgrade -- and motivating *ACTION!***

- Chicago billboards - picture the drive into town...
 - Check system for MS apps - teasers and mini-demos
- Leverage online service - instant electronic order for ras apps
- MS version of apps on Chicago upgrade teaser

- **Cost/Difficulty of Upgrade**

- price discount
- "you win" - maybe even really!
- locked cd-rom

MS-PCA 2535285

CONFIDENTIAL

Leveraging the OEM Channel

- **First-time OEM ad for OEM sales (mini-demo) on disk**
 - re-labeled by oem
 - oem resells app
 - locked cd-rom again
- **Online registration, follow-up with promo / locked cd-rom**
 - option to print the card

MS-PCA 2535286

CONFIDENTIAL

Design Wins

- **Make MS apps look great in shell**
 - What can we do before Chicago App versions are released?
 - What can we do in the Chicago release of our Applications?

Change is the key...

MS-PCA 2535287

--

CONFIDENTIAL

Pre-Chicago App Release What can we do in Chicago?

- **Make our apps look different - like the shell**
 - change the shell design to make this a real, not-for-free change!
- **Mini-upgrade (SDM DLL) bundled with Chicago**
- **Make applettes “ad’s” for our apps**
 - consistent UI
 - consistent Visuals
- **Visual changes - for a change!**
 - icon style, colors, size, etc
 - window adornments (e.g. scroll bars, status bar, etc)

MS-PCA 2535288

CONFIDENTIAL

Chicago Apps and “The Points of Light”

- **Do all high visibility features that support the shell**
 - include some in “mini upgrade”
 - continuous reinforcement of the MS Family of Products
- **Do all app features that increase consistency, leverage with the shell**
...and of course, continue work on MS Apps Interop

MS-PCA 2535289

CONFIDENTIAL

Some ideas...

- **Custom view of MS install diskettes**
 - show package for ms apps
- **Applette ole servers consistent with MS Apps**
- **“Chicago Cool” certification stickers**
 - “kinda cool”, “cool”, “way cool”
 - “optimized” requirements for “way cool”
- **Plug’N’Play differentiation**

MS-PCA 2535290

CONFIDENTIAL

Product Definition and Packaging A General Approach

Assumption: Chicago *will* sell big even with only its core feature set

Fact: An Office sale is a big MS Win

- **We need to trade off features towards motivating office upgrades and new sales**
 - in preference to overloading chicago feature set
- **Basically, prioritize feature allocation as:**
 - Office
 - Chicago
 - MS Apps
- **Consider OS enhancements as part of Office**

MS-PCA 2535291

CONFIDENTIAL

The Radical Extreme: The Office Shell

Basic Approach

- **Hold extensible shell for Office**
 - Differentiation feature: shell integration. WOW!
 - Ship fully extensible shell, including OLE 3 with Office
- **Chicago shell is non-extensible**
 - "we couldn't get it done in time..."
 - differentiates both Office and Cairo
 - keeps Chicago focused on memory goals
 - nips this whole damn ole extensibility thing in the bud
 - limited OLE 2 integration - app interop category only
- **LEVERAGE an OLE 3 release**
 - license for app ship
 - we're first
- **Cairo, next version of Chicago delivery (OLE model)**

MS-PCA 2535292

CONFIDENTIAL

The Office Shell - Ideas...

- **Combined upgrade and install - single SKU for Office with Windows**
 - Office can require chicago!
- **Programmable (IDispatch enabled) shell**
- **Workbook integration**
- **Show internal doc structure in Office Explorer**
 - leverage outlining
- **Macro recorder that works with ms apps**
 - bundle automation scripts that leverage office apps
 - vbasic/vb only comes with office, no vb in chicago (?)
- **Extreme shell/common dialog integration**
- **Lazy unload / quick pre-load**
- **Post-it, net capable**

MS-PCA 2535293

CONFIDENTIAL

OLE 3

- Continue the OLE 1, 2.0 app pre-distribution model
- Shell integration focused feature set
 - new MDI / projects / workbooks / folios
 - new window styles, e.g. tabs
 - extended drag and drop
 - more automation/macro recording
 - explorer integration

MS-PCA 2535294

CONFIDENTIAL

Summary

- Apps: Do leveraged Chicago integration work
- Chicago: Do specialized support for MS Apps at install time for immediate MS Apps benefit, deliver mini-upgrade for apps
- Leverage Chicago as a "channel" for Apps marketing
- Leverage Electronic Order, Demo
- Consider the Office Shell
- Stay *extremely* aggressive on OLE releases, MS Apps support

MS-PCA 2535295

CONFIDENTIAL

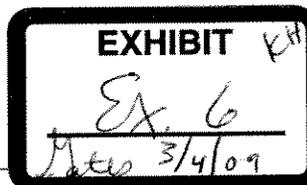
PX 52

Teresa Jennings

From: Dennis Adler
To: bradsi; davidcol; johnlu
Subject: My notes from retreat
Date: Monday, June 14, 1993 10:32AM

are pretty sparse, except for the Chicago talk. But here they are; note they are set up to be printed dbl sided and put into a Junior-size time systems notebook. For std paper you would want to readjust the margins to normal stuff.

<<File Attachment: RETREAT.DOC>>



Subject: Hood Canel Exec Retreat 6/10 - 6/12/93

Page 1

Rick Rashid (Rashid) 6/10 - ATBD research

David Heckerman (Heckerma) working on Baysian (?) decision tree for expert systems. Use for printer/hdw diagnosis. Or to determine best action in a UI at a given point. Not adaptive, though could be in the future. Right now requires rules be changed to adapt behavior.

Charles Simonyi (Charless) working on Intentional Programming (IP). Very cool dev environment that takes program tokenization to a new level, carrying a logical representation of a program around in a morphable form. Morph the code w/ agents/enzymes (transforms). Can change output style to C or Pascal Or LISP or... Or grab a common code section and turn it into a subfn. Or subclass a subfn to be called w/ specific args and have now unneeded code drop out. When combined w/ pointer/list work from Dweise, can get really powerful. GET A DEMO!

Billg, 6/11 - Thoughts

If we win market share from Lotus in apps and Novell on net, we will be in a great position vs. our other competitors as well. Saturation of our markets will become an issue, wich means *upgrades* will become a big business. So the more share we have by then, the bigger our upgrade business. Major product updates (x+1 release) should come every two years, marketing sizzle releases every year (x.1 release). Ship out to customer base on 1 CD; customers phone in w/ credit card number, get key, and unlock software (? put demo or ltd access versions onto the CD ?).

On the OEM side, the max royalty we can reasonably get is \$40-50 per unit. OEM has been so successful in selling that we've cannabilized our retail channel for Windows.

Apps prices are also under pressure. Projected overall revenue growth is 20-25% in FY94, from \$3.7B to \$4.4B. Unit sales growth in Europe is higher than this but price pressures + strong dollar keep profits in this area. Growth higher in some (most?) other overseas markets, but they are not large enough to impact the bottom line significantly.

Synergy - huge opportunity for apps in use of Chicago (and later Cairo). Shell, Viewers, Docfiles, taking over right pane of explorer. Also *applets*. Apps should move UI features to applets and use this to cement the features as standards; then people buy MS apps and leverage what they learn in the applets. Word's new word selection model *and more* should be in Writepad for example. Leverage shared code in system (e.g. OLE, comdlg, etc) to keep working set smaller vs. our competition. VB for apps (VBA, formerly OB) is

MS7089439
CONFIDENTIAL

Subject: Hood Canel Exec Retreat 6/10 - 6/12/93

Page 2

something to leverage between our apps, but not to give away in the system. Too valuable as a competitive advantage right now.

All apps should bet on Chicago; any user who does not buy Chicago is not a good candidate to buy an app upgrade either. Chicago will be pervasive; it is ok to require it for your next app version. Level the field for apps by cross platform licensing (to Unix and other non-Mac platforms) so you can take advantage of Chicago features in your apps and not worry. You must expect that there is some functionality in Chicago that will not make it to the Mac or any other platforms though, and account for that. *My observation:* Bill is saying two things here... that it is ok to use Chicago features because we will license to our partners on other sys platforms and that some features will not make it. Never a clue what features will/won't make it... do not know what apps are supposed to do!

Chicago presentation, 6/11

- *Bill says the new transfer model is in Chicago*
- *Davidcol signed me up to justify "every .INI setting"*
- *Paulma has signed up to create a plan to test "portable" DLLs from Chicago on NT (currently only TAPI, Unimodem)*
- *Bill still has a hot button about Unicode support and the poor story for this on Chicago, He asked about Murry Sergeant's idea and I stated Davidw examined and still says it leads us down the "slippery slope"*
- *AtWork (Karenh/Efax) team has a rendering engine that we should ship in Chicago. based on the WPS printing "language" that is something like but not a metafile... used to send fax info. There will be machines w/ this engine out there and Chicago should ship it!*
- *PrP networking - hot topic on what to do when an app loses the net without warning and a file is open. Excel might survive this; Winword will die.*
- *Performance Issue - One database guy asked (and Word/Excel concurred) that we make it cheaper to create new windows. Very expensive now (this is one of the enhancements apps has in SDM). Right now he says they do a lot of work to avoid creating windows because it is so expensive.*
- *Bill asks why can't SDM call the underlying cmodlg code; easier than ripping SDM out of the apps in the near term (though that should happen longer term). This should get rid of a few hundred K of code.*
- *Scheme for portable 16/32 bit metafiles currently calls for embedding a full 32-bit metafile as a single, huge record in a 16-bit metafile. This is painful but ok if you are smart about not duplicating sharable pieces like bitmaps. We should examine what can/cannot be shared and not be needless memory pigs.*

MS7089440
CONFIDENTIAL

Subject: Hood Canel Exec Retreat 6/10 - 6/12/93

Page 3

- Bill wonders if we (MS) are doing any work to improve the compressability of app data files and/or to give the system a way to tell the app that its file is on a compressed volume, so don't bother trying to compress it (my thought: app probably should tell us when a file is not compressible, e.g. encrypted mail file).
- Use object basic subset in Chicago (*my thoughts*: seems like one of the things Bill wants to leverage to make it a standard?)
- *Along with Word's new word selection method, Bill wants us to leverage other app features in Writepad. One additional example he gave me was right button popups should be the same.*
- When someone asked if we should put ODBC into Chicago, Bill shook his head no.

Bobmu - Cairo

To really take advantage of the cool features in Cairo on the desktop you need Cairo on your server(s). I pointed out that this is a double-edged sword. Bobmu noted that he sees this as a way to get Cairo in on new servers, not to displace old ones. Cairo will work fine w/ other (e.g. Novell) servers. But to really leverage Cairo you'll want Cairo on all your new servers.

Animated controls in Cairo... Someone told me that these things are on the Newton and thinks Apple may be trying to patent this sort of thing... Cairo needs to look into this!!

Notes from Breakout presentations & wrap-up 6/12/93

- Setup idea: When Setup sees MS apps on the system, check the version number. And tell user about upgrades that are available (or will soon be?). Little ads.
- When Cairo first ships, Chicago will need to play catch-up on some features (esp. browsing OFS servers... be great Cairo clients), just like NT needs to catch Chicago today. There will likely be 3 Chicago releases, then the Chicago technology base goes away. Still a hi-end and low-end package to sell (one w/out all the full query/browsing capabilities perhaps), not unlike the 3 SKUs Bradsi sees for Chicago.
- Corp developer ideas for Chicago: OLE automation in the shell for our apps. More controls for corporate developers (e.g. routing slip -- *note that we should probably expose routing info as a docFile property and have Capone pay attention to it!! Winword & Excel do their own different routings now*). High-level access to MAPI in C and VB (make it happen w/ a class wrapper to objectize the interfaces). Replication (to workgroup enable) for a data store (?)... see Group #3 slides for more ideas.
- **Ship extensible shell in Office!!! Wire the features we need for Chicago into the Explorer (e.g. mail integration, printman, CPanel, Fonts, etc.). Bill sez do it! My note: probably want to keep extensibility stuff as-is, but**

MS7089441
CONFIDENTIAL

Subject: Hood Canel Exec Retreat 6/10 - 6/12/93

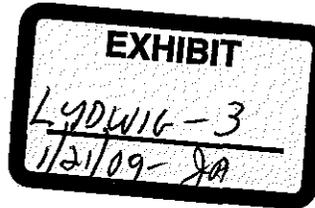
Page 4

require preloading things from some internal table...? Just to make implementation easier.

- Need overall UI closure from Apps/Cario/Chicago.
- Bill will write the Technical Strategy Document called for in the Synergy action items.
- We will do another retreat like this in October to follow up on progress.

MS7089442
CONFIDENTIAL

PX 54



Donna Poreda

From: johnlu
 To: bradsi; davidcol; paulma
 Cc: johnlu; stevesl
 Subject: RE: shell in office
 Date: Tuesday, June 15, 1993 10:28AM

i am feeling pretty queasy about it too. our worst nightmare is novell/lotus being successful at establishing their "middleware" as a standard. ours ought to be ubiquitously available to forestall this. our huge advantage vis-a-vis novell is our end-user franchise, we shouldn't cast aside this advantage.

From: bradsi
 To: davidcol; paulma
 Cc: johnlu; stevesl
 Subject: RE: shell in office
 Date: Tuesday, June 15, 1993 10:17AM

i agree with all this.

in addition, i fear that if we are in Shell Wars, it will lead to the view that the shell is just an app, not an integral part of the system. Thus Shell Wars will only serve to delay cairo adoption. the shell is such a central part of the whole notion of cairo's information access; we need to be in control of the shell or else cairo will just be yet another (radical)

variant and won't usher in a new age of component software.

further, if the shell is now viewed as an app rather than as part of the system, then it devalues windows itself and makes it easy to clone.

now, one can argue that lotus et al are going to do it anyway. my counter

is that it will be viewed more as some random activity, a layer on top of windows rather than a full replacement, and won't get third party support.

our competitors are going to do everything they can to fragment windows. they will build their own middleware to claim api ownership. they will do

clones. they will try to diverge at the ui level. we should not take actions which encourage and reinforce those efforts.

From: davidcol
 To: paulma
 Cc: bradsi; stevesl
 Subject: shell in office
 Date: Tuesday, June 15, 1993 9:22AM

I've been thinking about this a bit since the retreat. It's an incredibly dangerous thing that we should not do.

The danger is clear. Shipping a shell in office forces the other vendors to

ship shells as well. Either the competition teams up and uses the same

shell, or they all have their own. In either case the Windows UI world is completely fragmented and we lose control. In fact, since shells are usually the things that break from release to release, we'll be cornered into either breaking other shells and thus office suites and making customers unhappy; or restricting what we do in each rev. Either choice means Windows stagnates; users won't upgrade if it's not compatible and they

won't upgrade if it's not compelling. I think the danger of major vendors shipping a Windows shell even if we don't in office is low, they want to

be

part of the Windows church. Us kicking them out of the church is the only thing that will really force them to make a church of their own.

The whole notion of people buying more office because it adds value to the system contradicts my view of Windows being just an appliance. People care

about apps, they buy apps. It's not clear to me that a significant number of additional people would buy office if it had some features which enhanced

the system or if it had a shell of their own. I also doubt users will buy

the MS Office just because it had a shell that was more like the Windows church shell than a break off shell. People make buying decisions based on

apps and how those apps interoperate.

Forcing multiple Windows churches is bad.

PX 55

Mike Maples

From: Tom Evslin
 To: Daniel Petre; Laura Jennings
 Cc: Darryl Rubin; Mike Maples
 Subject: system/apps retreat
 Date: Wednesday, June 16, 1993 5:46AM

Pls don't forward.

A few things came up of importance to wgd.

1. A very interesting plan was developed and tentatively adopted to bundle the extensibility of Chicago shell and some of the shell sizzle with Office rather than release with Chicago itself. This makes these features a compelling reason to buy Office rather than icing on the cake of an OS we can't make as much profit on. Implication is that the mail client in Chicago might not want to be as full featured as the mail client in Office or Mail itself. Perhaps this is even the 16 bit client if Explorer is not extensible in Chicago itself. This was not discussed in any detail but we should be thinking about it. Pls give me your thoughts so I can write followon mail. I think we should push for full MAPI to be in Chicago but that may be controversial.

It is too soon to pass this on to the client teams or other people in wgd who I know will welcome another change like a hole in the head but it is likely to become the Chicago POR around midJuly.

2. DavidV presented a far more developed plan for unification of database strategy including Cairo, EMS, and a to be developed engine. Although he presented it as immediate replacing all efforts, it is much more realistic two or three years out. I spoke to PaulMa, RogerH, DavidFu, and MikeMap after and all agreed that ems must go forward as planned (Paul the weakest on this) and that we will review the db strategy as it develops to make sure we have a convergence plan. I especially asked these four to make sure this does not spread as rumor of the demise of ems and all agreed. It is very clear that EMS' value is strongly linked to it delivery date. It's all we have in that timeframe to compete with NOTES and it's key that it stay in that timeframe.

3. Darryl presented EMS to general interest but some skepticism that we can get it all done. There was no objection when he presented exposing MAPI in the 16bit client altho I'm not sure everybody understood the implication of this.

4. There will be greater emphasis on synergy including a billg memo that there is no Chinese wall nor any legal or ethical reason for having one. All groups including us need to be more proactive in planning and working together. A couple of specifics - we need to plan with JimAll's people how the directories will eventually converge; we need to see if it is possible to get NT 3.1 to include support for Netware 4.0 directories and then have EMS inherit this from NT rather than support Netware directly. I'll get the right contacts in systems to work on this but we are going to have to spare some time of valuable people.

5. The respect people have for wgd has increased greatly in the last year thanks to 3.0. But there is not general knowledge that we now lead our category or that we have released anything since 3.0. Has there been anything in MicroNews about 3.2, remote, new gateways, eforms designer - I don't remember? Anyway, we need to be more IYF.

PX 56

Erik Stevenson

From: Tandy Trower
To: Bill Gates; Brad Silverberg; Mike Maples; Paul Maritz
Cc: Darryl Rubin
Subject: extensions
Date: Wednesday, June 23, 1993 10:43AM

I have been thinking about the discussion we had about where we package "shell" extensions and I am increasingly convinced that this is a bad idea.

1. The extensions are NOT that significant. After talking with JoeB they can be summarized as adding commands to icons, adding property pages and adding views to the "explorer". Perhaps there may others we haven't defined yet, but the significance of the "extensions" seems to be overblown. I don't believe these result in any significant benefit in the marketability of Office.
2. If there IS any benefit, it just penalizes our own customers who only buy individual products from us.
3. This significantly raises the stakes of competition. It incents ISVs to directly compete with us on the "shell". You might argue that this may happen anyway, but we maintain a stronger position by ensuring that the common base is of high quality. The more we deliver only a weak shell, the more we open the opportunity for someone to displace the basic UI and operation of the system. If shells just become a commodity of a software suite, then it can no longer be considered the core of the system. Minor ISVs have to figure out which shell to integrate with. In short, this strategy potentially makes it harder to maintain a sense of standard.
4. This strategy signals a sign of weakness. This stinks of "proprietary-ness", something that we have been critical of others for embracing. We should be able to compete effectively by doing better OLE apps, thinking hard about how to evolve from GUI to OOUI apps. There is a significant transition in both the architecture and in UI to be made. (I've seen very little evidence that we are really thinking of the implications and moving forward here.) We are better focused on making this transition, like we did from character to GUI, than trying to leverage some weak extensions. If we really wanted to leverage a technological advantage we should have kept OLE 2 as a proprietary set of APIs.

It just doesn't appear to me to be a smart strategy. It seems contrived and the possible repercussions not worth the risk. In the twelve years, I have been here, I've always taken pride in the fact that we excelled by doing things better than our competition, not by withholding some functionality that we might uniquely leverage. That doesn't mean that I think we should just give away all our technology. I just don't think this particular proposal is a good one and doesn't fit our character.

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"> - Chicago ships - Shell has limited extensibility. (See below for details.) |
| Chicago + 6 months | <ul style="list-style-type: none"> - Office ships with optimized shell. - Shell adds features for optimal support of Office requirements. (See below for details) - Office includes many features that exploit the new shell. - New shell not initially available with Windows itself - The Office shell should be approximately a superset of Chicago shell features (although some components, such as the Tray, may be replaced.) - 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. |
| Sometime after Office ships | <ul style="list-style-type: none"> - Cairo ships with a shell that is a superset of the Office shell - Excludes any components that we choose to keep only for Microsoft Apps. - Extended to use special features of Cairo |
| When Cairo ships | <ul style="list-style-type: none"> - Enhanced shell added to Chicago - 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. |

Plaintiff's Exhibit

7377

Comes V. Microsoft

- Page 1 -

EXHIBIT

Ex. 3

Date 3/6/09

OFFSHELL.DOC
Microsoft Confidential
07/02/93 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 (Cafone 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 62

EXHIBIT # 9
 WITNESS Muglia
 DATE 2-8-09
 ZOYA SPENCER (206)382-9695

From: Bob Muglia
To: Christopher Graham
Subject: FW: Office Shell
Date: Saturday, July 03, 1993 7:21AM

I don't know what happened, it was sent to you...

bob

From: Bob Muglia
To: Paul Maritz
Cc: Christopher Graham; Jim Allchin
Subject: Office Shell
Date: Thursday, July 01, 1993 6:13PM

I've talked to both Chris and Jim in detail about this. I think there are three options:

1. Status quo

We leave the org and objectives pretty much as is. Chicago continues along their path, exposing shell extensibility. Where possible, we convert the shell extensibility mechanism to OLE. The apps group continues enhancing Word/Excel.

Although this is the current plan, it seems like a bad option no matter how you view it. Chicago is in a catch-22 situation - their extensibility mechanism isn't compatible with anything, yet it's not at all clear that they can meet their size goals using OLE. In any case, whatever they do is the standard which apps will be written to (and what Cairo must be compatible with).

Word and Excel are forced to battle against their competitors on even turf. Given that Lotus and Wordperfect have largely caught up, they almost certainly lose ground - if not in market share, then in margins.

Chicago, of course, sells in high volume. But we don't reap the benefits of this.

2. Office Shell

Although there are conflicting opinions about whether this would work organizationally (Jim has valid concerns), it would be possible to solve the problems I outlined in the "status quo" approach by forming a team in apps which focused on shipping an integrated office product, including the shell.

Organizationally, this would likely take the form of a single design team in applications (with stevem playing a significant role), and two development teams - Mack over here and another one in apps. In order for this to work, the design team would need to consider the requirements of both office and Cairo. From a development perspective, you'd need to have one team which owns the core code, with the other team making modifications appropriate for their product. It makes sense to start with the Cairo code (the Chicago shell would NOT meet the needs for office). Thus, it probably makes sense to have Mack drive the code design, with the office development team working from that code base.

The one thing which must be understood about this plan is that it is NOT a plan which pulls Chicago features into Office for the benefits of our apps. As currently planned, the features one could pull from Chicago ARE NOT compelling enough to do the Office shell. What this plan does is pull features from Cairo into Office, and thus onto the MS-DOS/Chicago platform. If you look at what Chris is proposing, the vast majority of the Office features are things we are either planning to do in Cairo, or are beyond the current Cairo plans.

As far as timeframes go, the integrated office shell will ship in the same timeframe as Cairo (18-24 months from now).

The net effect of this plan is that our applications move forward significantly. We probably pull-in by two years or more some key features which would differentiate our apps. On the other hand, we do this by

MX 1389851
CONFIDENTIAL

FL AG 0019001
CONFIDENTIAL

pulling features which previously differentiated Cairo from Chicago. Do not kid yourself about the net effect on Cairo - it becomes much less compelling. The net result of this is that it is much harder for us to move corporations to NT on the desktop, thus delaying our ability to achieve a single OS kernel.

3. Bet on Cairo

At one point long ago, I think this was the plan... ;)

Everything you might anticipate doing with in the office shell could be done on Cairo, it could be done in a more architecturally correct way, and it could be done with less effort.

Given that the office shell borrows ideas from Cairo and is likely based on Cairo code, it's pretty clear that doing all this work on one platform would be less effort than doing it on two.

The issue is of course, that nobody outside building 5 believes Cairo will sell in volume in an acceptable timeframe.

However, this really is the plan which makes the most sense. Organizationally, it requires the least perturbations. It keeps a UI design center in Systems (rather than having all UI expertise in apps). It moves the apps forward and allows them to differentiate themselves from their competitors.

The Cairo product spec would have to change to meet the needs of apps, but this could actually happen if there was focus on building apps for Cairo. We would succeed in moving to a single OS kernel.

From a Chicago perspective, you'd do whatever makes sense to get the shell extensibility in-sync with Cairo - that probably means a combination of removing interfaces and changing some to be compatible with Cairo.

You'd certainly continue to fund update releases of apps product for Chicago - but you'd invest heavily in Office for Cairo.

Although I know it's a stretch for people to consider this, I think we're doing a disservice to ourselves by ruling it out without consideration. At a minimum, I think we should think thru what Cairo would need to do (in terms of size, speed, features, distribution, etc.) in order for us to "bet on Cairo".

bob