



Amy Johnstone

---

From: Paul Maritz  
To: Bill Gates; Mike Maples  
Subject: Memo's  
Date: Thursday, July 01, 1993 12:18PM

I took some time out this week to "put some thoughts down on paper". Please don't forward them as they raise lots of issues that I don't fully understand and/or have answers to, but I think it is critical that we start sorting through them now. It may be worthwhile getting the three of us, plus steveb, peteh, rogerh together for a couple of hours to talk this thru - and see if there is even concensus as to whether our problems require this kind of thinking/approach, and if so, what to do about it.

The first memo is an admittedly weak attempt to try to provide some framework to get more order and efficiency into our mainstream product line. The org. recommendations in particular are there purely as a strawman.

The second one is some thinking I was doing wrt. the "Office Shell" and what it would take to make it "compelling" enough from a customer point of view (compelling enough to move more "apps"). The conclusion is that to get something compelling one needs more than just a "Shell", we basically need "Cairo on Chicago". This lead me to try to lay out a larger framework, hence the first.

<<File Attachment: FRAMEW.DOC>>

<<File Attachment: OFFICE.DOC>>

MS 0153730  
CONFIDENTIAL

## SYNERGY AND EFFICIENCY IN THE (mainstream) MS PRODUCT LINE

This memo is intended to provoke constructive debate.

### Problems:

As we all know, we have at least the following major problems:

1. Declining ability to differentiate our applications from competition, and the possibility of severe price pressure.
2. Too many products and in particular too many overlapping products. This is not only a big resource drain (development, marketing, localization, etc.), but it makes it harder to sell them, and will cause real customer problems down the road - maintaining all these things, explaining how to administer them, how they do/do not interoperate, etc. It also causes high frustration levels in terms of internal relationships within the company.
3. We don't have credible products to counter Notes and Novell.
4. In systems, we have overlap between Chicago and Cairo, and difficulty selling NT.
5. In many ways Cairo is the answer to reducing our product line and competing with Notes and Netware - but NT/Cairo are not credible inside the company - which leads to people try to build "interim" solutions, and causes product proliferation - compounding the problem.
6. Our cost structures and efficiency are way out of line - we have too many people. Our numerous business units and management hierarchy are causing us to duplicate and proliferate.

So... this is an admittedly very "simplistic" effort to try to firstly articulate a framework for what our product line should be (in say H1'95), but secondly, and more importantly, to try to think through the really hard part: how to get there - i.e. what should happen to current projects, and what should happen wrt. organization. I know that are a TON of issues that are not addressed here, but we have to start thinking this through.

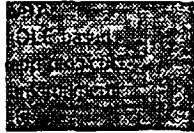
### Product Framework:

In H1'95, the company should be selling the "products" diagrammed below. Note:

- it is necessary to read the notes,
- the color shaded groupings could indicate packaging, i.e. our basic product line COULD be reduced as indicated - of course there other ways to package things,
- the framework is not intended to be exhaustive - there will be other products - but these would form the "anchor" products.

MS 0153731  
CONFIDENTIAL

"CUST TOOL"



Word  
[11]

"OFFICE" [10]

Excel  
[12]

PIM  
[13]

Query/  
Report  
Tool  
[14]

"ENHANCED CLIENT PACK"

Explorer [11]	Client Record Store [7]
------------------	----------------------------------

Server Record Store [4]
----------------------------------

"CLIENT"

Base Shell [8]	Client File System [6]
----------------------	------------------------------

Server File System [3]
------------------------------

Client OS [5]
------------------

Server OS [2]
------------------

"SERVER"

Notes:

1. The Explorer is an OLE extensible shell/browsing tool. It is capable of invoking app. supplied extensions that allow it to "see" structure inside a document (e.g. double clicking on Word doc would show outline view of the document, etc.). One particular type of document that it can browse is the "record store" (client and server - see below). As noted in the shading this Explorer would be available only in the "enhanced client package" (requires OEM to pay extra, or customer buys as add-on). The Explorer also has the UI to do document library functions - check in/out etc. This tool also acts as the "mail client/bulletin board browser".
2. The Server OS provides network wide services such as multi-domain security, distributed file system, system and network admin, etc. It requires 16MB system and is scalable to very larger systems.
3. The Server File System is a service of the server OS, and provides a general purpose document container, providing the following capabilities:
  - Replication
  - Event notification/Action invocation
  - Extensible properties
  - Queries over properties
  - Per user properties (e.g. read/unread)
  - Content Indexing
  - Store and Forward (including gateways, etc.)

MS 0153732  
CONFIDENTIAL

4. The Server Record Store is a storage subsystem on the server that provides high-performance, reliable, multi-user record access. It can be replicated, leveraging replication mechanisms of the Server File Store. It is administered in same ways as the server OS.
5. The Client OS in its base form requires only a 386/4MB, but in its enhanced form requires 8MB of memory. It does not provide local security, and is not scalable (near term). Applications running on an enhanced client can access (remotely) all the functions of server based services, esp. the Server File System.
6. The Client File System provides a subset of the server file systems functions. It can handle simple events/actions, it can correctly tag documents as to their class, it allows browsing by the Explorer "into" local documents (maybe with less performance than for documents stored on the server). It can "sync" documents with the Server File System.
7. The Client Record Store - is a lighter weight version of the Server record store, with compatible API. Records in it can be "sync'd" up with a Server record store.
8. The Base Shell is subset compatible with the Explorer but is not extensible (i.e. cannot drill down into documents).
9. VBA - this is the toolset needed to customize/program solutions using the other components as building blocks. It should have a common forms model that which is used within the other components, and it should allow OLE objects as controls.
10. "Office" - this is the suite of MS "apps". Each of these apps can:
  - expose their structure to the browser, and allow the user to specify what/how to expose things,
  - be externally programmed from VB, and thus be extended in VB,
  - raise/respond to events,
  - reconcile differing documents of their type,
  - and obviously conform to OLE compound document spec's.
11. Word - this is the tool that allows one to view/edit/present textual documents. It serves as the word-processor/presentation/drawing package.
12. Excel - this is the spreadsheet tool.
13. PIM - this is the tool for viewing todo-lists and calendars - which are stored in the client or server record store. It allows events/actions to be associated with items in the list/calendars (as do all the other Office tools).
14. Query Tool - this allows one to quickly generate queries against the record stores, and produce reports. It also works with maybe lower efficiency against items stored in the server file system, and maybe even lower efficiency against the client file system.

#### Current Projects:

In order to implement the above framework, current projects would have to be redirected as follows:

#### Chicago:

For the years 1994 and 1995, Chicago would be the principal client OS.

MS 0153733  
CONFIDENTIAL

Chicago should be delivered in two phases. Phase 1 (mid'94) should provide Chicago as currently defined, plus the Base Shell. This would mean that Chicago would not have an integrated mail client - this is not crucial to either the initial success of Chicago (as a PnP vehicle, and successor to Win3.1), nor will it greatly alter the dynamics of the mail business in the near term. Phase 2 would be sync'd with the Explorer and Cairo (A.k.a. Server OS) in Q1'95. Phase 2 would include the necessary pieces to allow client apps to access Cairo based services - esp. OFS/DS. An issue is whether Phase 2 would also require extensions to the Chicago File System to support the Explorer.

**NT/Cairo:**

NT would undergo a 1.0a release in H1'94, and then the focus would be on providing the Server OS, and its Server FS - to serve clients. We would also sell NT/Cairo as a client during this period - for those customers who want security/reliability and for RISC systems. We would continue to work on improving NT for client operation, with a view to be able to position NT technology to replace Chicago technology in 1996 time frame. This is also important to get the server resource requirement down, as our preferred configuration for a customers, small and large, in H1'95 will be Chicago clients with Cairo server. The positioning should be "to get the most out of your Chicago clients, install a Cairo server". Specifically the Cairo server would provide to Chicago clients: efficient queries over documents stored in server file system, distributed file service, directory service, multi-domain security.

We would probably charge same for NT client as for "Enhanced Chicago Client" - and would thus bundle the Explorer.

**Cairo Shell:**

This becomes the Explorer. Goal of shipping in Q1'95 with Chicago Phase 2 and Cairo.

**Cairo Development Environment:**

This is the OLE2 based Forms/Control development environment and gets merged into the VB "TOOL".

**Visual Basic:**

The "TOOL" - focused around delivering VBA with support for OLE Forms/Controls. Ships same time as Explorer, and runs on Chicago and Cairo.

**Access:**

Becomes set of extensions to the Explorer/Tool.

**Word:**

Continues as main word-processing tool - but given responsibility for producing all "word" oriented components. They do the work necessary to integrated into the Explorer environment, to integrated with the "Tool", respond/raise events, etc. In particular, they also do the work to merge textual entities (in response to events). Subsumes Powerpoint with time.

**Excel:**

Ditto for grids/figures.

**Powerpoint:**

Subsumed into Word.

**REN:**

Part is subsumed into Explorer, rest becomes the "PIM" (Todo List/Calendar) tool.

MS 0153734  
CONFIDENTIAL

**Capone:**

Shipped in mid'94 along with stripped down EMS (see below). Shipped as separate entity from Chicago. Replaced in Q1'95 by the Explorer.

**Schedule+:**

Replaced by PIM tool.

**EMS:**

Shipped in Mid'94 as MTA/Gateway engine only. I.e. MAPI's "store" API gets gutted (leaving the "transport" and "directory" APIs of MAPI). This store function is picked up in Q1'95 by the Server File System. The EMS group gets focused on providing the MTA/Gateway functions for objects in the Server File System. They have to write the conversion utilities to move directory information from Blue to Server File System (OFS), etc.

**Red/Fox:**

Either become, or are replaced by, the Client Record Store (which has come access and navigation API with the Server Record Store, Server File System, and maybe Client File System).

**Blue:**

Lives briefly as part of EMS Phase 1, then dies.

**SQL Server:**

We get deal with Sybase that allows us to use it as Server Record Store, and enhance it.

**"Workgroup" Database (the Adamb/DavidV project):**

Explorer part of it gets subsumed into the Explorer. The replication part of it gets redefined to be layered on top of "Cairo" (server OS) replication - i.e. replication can work if there is Cairo server around.

**ORGANIZATION AND PEOPLE:**

This is the hard part. The only way that people will give up local goals and local structures is if it part of a major change. There are other ways to do things, but here is a proposal. It would propose four development teams and two marketing teams. It is my belief that this should require dramatically reduced staffing - in fact what do with people will be a major issue.

**Development Teams (dev/test/program management):**

**"Base Client OS"**

Brads.

Current Chicago team, plus headcount to do client side pieces to access the server.

**"Explorer" and "Tool"**

Roger Heinen

We form explorer group under Steve Madigan, using significant resource from Cairo.

We unite CDE and VB groups under a strong manager (who?).

Move in the Capone team.

**"Client Record Store and Server Pieces"**

Jim Allchin.

MS 0153735  
CONFIDENTIAL

NT/Cairo (OFS, DS, DFS, etc.). We move the EMS team now, the database engine pieces (Red, Blue, Fox).

"Office"

Chris Peters(?)

Gets Word, Excel, PowerPoint, REN, Access.

#### Marketing Teams:

"Platform Marketing Team"

Rich Tong(?)

This team would market: Client, Server, Client Enhancement Layer, Tool.

"Office Marketing Team"

Lewis Levin(?)

This team would market the Office components: Word, Excel, PIM, Query.

The whole thing could/should report to one manager, and would have a small architecture/program management staff (not to design, but to ensure things were not falling through cracks).

#### Other Products/Projects:

Obviously there are other products/projects that would continue: MSDOS, AtWork, C Compiler, Mouse, Consumer, etc. They should continue but should probably not be part of the above organization. ?? It does lead to the "how many companies should MS be" question.

#### Sacrifice:

Such a large change as outlined here, would come at a large cost:

- we would have to forego competitive actions in the near term in certain categories (e.g. presentations?, database tools?) in return for a "paradigm shift" product line in H1'95.
- we would have to live without an compelling answer to Netware/Notes until H1'95.
- a lot of pieces have to come together on the same schedule in H1'95.
- a lot of people would resent being part of the larger group, and not in the own "business unit" - the inherent hostility by our current organization will be very high. How to make this come about is a crucial issue.

MS 0153736  
CONFIDENTIAL

## Office Shell:

I was reading thru the outline document that ChrisGr gave me on Friday. Here are my thoughts on what it would take to make the Office Shell a "compelling" offering for a user, as well as other issues. This memo does not address packaging issues.

### A. Key Features needed to create enough customer benefit (other features may make sense, but I would think one would need at least the following):

#### 1. Browsing "into" Office documents:

This would allow the shell explorer to see into Office documents (Word, Excel, PPT, Access DB's, and Mail documents). Eg. double clicking on Word document would bring up chapter/title headings, doing so on a PPT file would bring up the outline view of a presentation, etc. I.e. the distinctions between a folder, and a document would be become less rigid. Note that including Access in the Office apps means that the Office Shell should be able to browse Access-accessible databases - in this sense, the Office Shell would subsume the explorer portions of what Adamb/davidv have been proposing.

#### ISSUES:

- a. Performance - could it be done efficiently enough to allow browsing over large collection of document. Would it be that to do this efficiently, would this require the equivalent of Cairo summary catalogues.
- b. Changes to applications - significant work would need be done in the component apps so that they would generate these views needed.
- c. To what extent would content indexing and access of documents also be needed?

#### 2. Document Library functions:

By this, I basically mean, be able to associate events and actions with updates to documents in folders, and to be able to track check-out/check-in of documents.

#### ISSUES:

Mainly ones of deciding how far to go on a Chicago only platform:

- a. Does one try to provide some level of security, how does one handle notion of a "user".
- b. What events does one detect and how, how are actions specified by the user, etc.
- c. etc.

#### 3. Visual Basic Programmability:

This is somewhat divorced from the notion of the Shell, but:

MS 0153737  
CONFIDENTIAL



- (i) it should be that the Shell and its function are customized in exact same ways the applications - eg. actions for doc lib events are customized/specified in VBA,
- (ii) unified forms composition tool - one way of creating form that can invoke OLE2 components and VBA code.

**ISSUES:**

All the usual ones:

- a. new common OLE forms model,
- b. apps need to present common programming model.

**4. PIM:**

Office should include a schedule and list browser, and the schedule package should work in a group environment. The schedule/list should be integrate with the event/action model - allow certain events to be defined and set off on time events.

**5. Mail:**

The mail package should integrate in, and expose "properties" as the other apps do. It should also (at a minimum) allow for bulletin board functions when used with a suitable server.

**B. GENERAL ISSUES WITH THE ABOVE:**

When looking at the above list, one can see a large overlap (not surprisingly) with Cairo, and in many ways one can think of this Office Shell (as defined above) as being "Cairo-Lite on Chicago". This raises several issues that would need to be delt with - in no particular order.

**1. Timing:**

Given the scope of the above work, and the fact that it would require a major rev. to the applications, we are probably looking at something that could be delivered no earlier than Q4'94. This is probably OK, but we should not allow this effort to delay Chicago - we have to get Chicago into the market no later than mid'94 (we need the upgrade revenue). It would mean that we would have to be very disciplined in ensuring that the Chicago base Shell and the Office Shell are aligned.

**2. Relationship to Cairo:**

What does Cairo now become? In many ways Cairo (as currently defined) consists of two pieces:

- a. NT + more plumbing (OFS, DFS, DS)
- b. An OLE2 enabled Shell and the CDE (tools to make VBA programming easier including common OLE2 based-forms package & controls).

The Office Shell proposal (if you accept argument that above functionality is minimum necessary) essentially puts a large amount of b. (the end-user visible functionality) onto

MS 0153738  
CONFIDENTIAL

Chicago. What end-user visible stuff would Cairo retain if this were to happen?

Probably:

- Queries over arbitrary properties (although depending on how one did browsing, one might be able to also do limited queries on non-Cairo system).
- External user extensible properties on certain documents (those managed by 32bit apps),
- More general purpose event action/model ("smarter folders"),
- Replication
- Object level security

If we were to go so far as to do the Office Shell proposal as defined here, it would also make sense to expose most of the previous list of remaining Cairo functions via a Chicago client piece (ie. these functions would be operative when a Chicago client was looking at Cairo server). This would have the effect of basically constraining NT to a niche role on the desktop - high security / reliability, and RISC would be only real motivations to use NT on desktop for next 2-3 years. This would prolong two code base issue for systems to have to deal with (two driver models, etc.).

On the other hand, this may be reality anyway, and this revision of what we are doing may make "Cairo" functions would credible/compelling within MS and maybe also outside of it. It may also make folk more willing to bet on the server side functions of Cairo and resist the tendency to put a lot of server functions (replication, security) into the apps (making for easier admin model for our customers).

### 3. Relationship to REN

REN's role in this model would be to be the PIM extensions for the Shell. ie. REN project becomes the same as the Office Shell in many ways - ie. REN becomes greatly "simplified" or "unified".

[Needs further understanding by me - but we should force the unification between REN's data access model and that of either Access (Jet) or OFS - ie. we should have only three kinds of API's for access stored data - (i) Win32 File IO, (ii) Jet (or whatever DB group defines as its successor), and (iii) DocFile/OFS access.]

### 4. Relationship to DB products

As noted above, the Office Shell should become the "explorer" as defined in recent DB group proposals. It is second question as to what we should be doing about unifying the storage and administrative mechanisms.

### 5. Relationship to CDE

The CDE (as I understand it??) is mainly a collection of tools to help customize the shell, and to provide infrastructure (forms and components) for OLE2. In this sense it may be considered the "Office Shell" development kit - targetted at VBA level programmers.

### 6. Cross-platform

MS 0153739  
CONFIDENTIAL

Under this scheme, the Office Apps would become fairly tightly integrated in with this "mini-Cairo" layer. Supporting the Mac would mean placing this layer on the Mac, Ditto UNIX.

#### **7. Organization**

Not subject of this note, but obviously a key topic.

MS 0153740  
CONFIDENTIAL