

## 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.

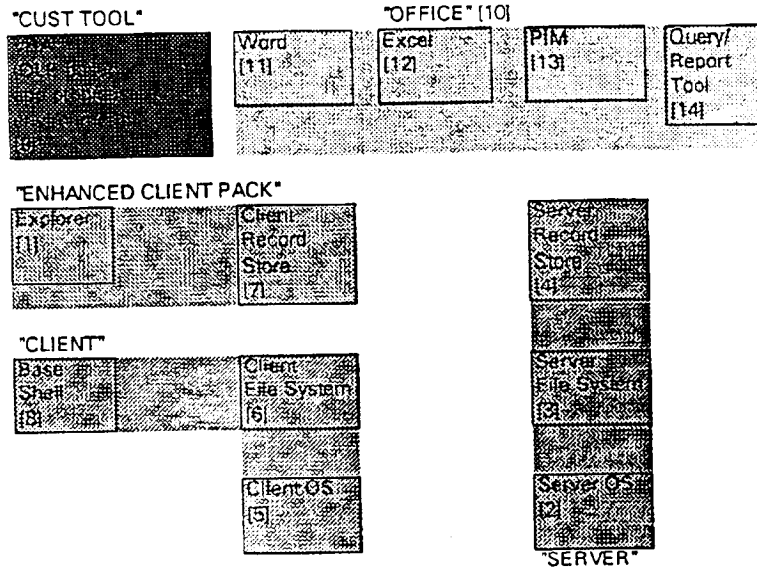
So... this is an admittedly very "simplistic" effort to try to firstly articulate a framework for what our product line should be (in say H1'95), but secondly, and more importantly, to try to think through the really hard part: how to get there - i.e. what should happen to current projects, and what 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.

*Line of business  
- Design - for*



**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.)

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 5048492  
CONFIDENTIAL

NT-chicago Phase 1 RISC

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.



**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"**

Bradsj.

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.

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.

*Pete ?*  
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.