

From: Eric Lockard (Exchange)
Sent: Friday, December 18, 1998 10:42 PM
To: Bill Gates
Cc: Kurt DelBene (Exchange); Jeff Teper (Exchange); Gordon Mangione (Exchange); Mike Tuchen (Exchange); Nat Ballou (Exchange); Darren Shakib (Exchange); Chris Larson (Exchange); Alex Hopmann (Exchange); Malcolm Pearson (Exchange); Eric Rudder; Jim Allchin (Exchange); Bob Muglia (Exchange); David Vaskevitch; Paul Flessner; Brian MacDonald; Paul Maritz; Jon DeVaan; Russell Stockdale (Exchange); Dave Malcolm (Exchange)
Subject: Storage for Office

Summary

With the Platinum store, we are on the verge of achieving an integrated storage and application development environment, client and server, in 1999, which would allow us to compete effectively with Notes and Domino as well as bring tremendous advantages to Office and Windows users and developers. I also believe we could totally screw this up, lose momentum and lose years if we make the wrong decision at this crossroads.

Outlook, PKM, NetDocs, VSS and other groups have done the analysis - Platinum is fastest, safest, best way to get the right storage for Office and beat Notes which insures the value proposition of Office/Windows over thin clients. We can make this happen in 1999. The Storage+ alternative has merits but it has the same risks our previous strategic storage efforts faced. Lets bet on the thing that we know has been optimized for the scenarios we must address and do a great job of development consistency and SQL integration.

Focusing on the right requirements

As a company, we need to make sure we focus on competing with Notes/Domino. Too often in the storage debate, we get off track on what are the most important and immediate goals for storage unification. Running SAP on the same storage system as that used by Office or provided natively in Windows, while certainly important long-term, is not helping and potentially hindering competing with Lotus and providing a server-transparent, caching local storage system which seamlessly unifies the file system, Outlook storage, Office documents, Access, Outlook Express and IE's cache. (If integration with SAP is the high order bit, we should be running hard at enabling Lotus's DECS style integration by creating Exchange-BAPI, Peoplesoft and Baan replicators. This provides the end user integration without requiring a wholesale conversion of the underlying store.)

I would argue that reducing overlapping efforts within the company is not the prime motivator for storage unification. Competing with Notes, making Office and Windows the premier application platform, providing a unified storage paradigm for the Windows user for all their data, these are what we should stay focused on.

Notes/Domino R5 is very scary. We all saw the demo. Exchange has worked with teams around the company to put together a very detailed analysis of the R5 betas and the hints they've exposed on their future direction. We now have a fairly good understanding of what they are doing and where they're going. Lotus's goal is nothing less than to become the new OS - to provide the environment applications are built upon and users live in without ever needing to leave. They threaten not only the Exchange/Outlook business but all of Office and eventually, Windows. BobMu points this out very eloquently in this recent mail:

IBM threat to
Knowledge Worker...

We are neck and neck in some areas, ahead in a few, and behind in a number of others. We have to execute crisply over the next several years to address this threat - any mis-steps we make will likely make us lose the battle for the foreseeable future. Aside from dramatically bolstering our tools support, and integrating it with our messaging application platform, I would argue that the most critical need is to achieve storage unification of files, email, calendar items, tasks, contacts, documents, pages and other "file like objects" such as source code modules and provide a symmetric application environment client and server for collaborative applications that build on these data types. Not having this store be the same one used by high-end ERP systems is okay in the near term. We have a competitive product in SQL, but Oracle still has the upper hand on the high end. While one could argue that BI changes the rules, in the battle against Unix and Oracle, I would have the SQL team focus primarily on scale at the high end for ERP systems with clustering, etc. for at least another turn or two of the crank.

Lotus's focus is on the same data model and applications as Platinum, but their store is actually relatively weak in comparison. The richness of their queries or the flexibility of their relations are not their strength. Rather, they are providing the object oriented storage constructs and application model appropriate to the space and they are winning! They are not letting themselves be distracted by deep unification with DB2 (yet) but are going squarely after sweet spot of so-called semi-structured data. Even if they do end up leveraging DB2, they realize that the vast majority of developers prefer to program in an environment that reflects how end users interact with the environment; rather than focusing on a model that requires advanced knowledge of the data types, full canonicalization of the metadata and normalized tables with restrictive schema definitions. Sure there are applications that can fit in this model but they are not the mainstream ISVs nor are they the people developing apps on Win32, the Web or Notes.

The Data Model

The issue we face is not one of simply choosing which of two essentially equivalent technologies to advance. If this was truly the heart of the issue, the decision would not be a difficult one. As stock holders, we would simply choose and move on.

Rather, much of the strategic decision should hinge on which of the two fundamentally different storage models and approaches are more in line with the semantics and base constructs we want to surface for applications, users and the OS. There are very fundamental differences between the two approaches which will have huge impact on how people build apps and deal with storage. For example, the Platinum approach is, for lack of a better term, more "object centric" in that schema is generally associated at the item/row level (as opposed to the table level) and there is a space in which with items can be moved, copied, etc. while the SQL approach leverages the strengths of a more relational model. While there is certainly overlap and either team can make inroads over time towards exposing more of the strengths of the other's approach - in effect providing the best of both worlds - these differences will surface to application builders and will have fundamental impact on such things as the user's and application developer's perception of the system - the mental model they work within. We are, in essence, debating the fundamentals of the future computing paradigm.

I believe that a storage model which surfaces hierarchy as a fundamental concept and provides for item-associated schema is the more appropriate model for Office and for the OS. It is a natural extension of the file system, web and MAPI paradigms and it is clear how things such as multiple storage volumes, removable media and distributed data are handled. This is not say that there should not be homogenous views or table associated schema, or multiple hierarchies or perhaps for some apps, no hierarchy at all. It is simply to say that an object-oriented paradigm is the more appropriate one to build upon for collaborative applications and documents. We will fail vs. Notes if applications development and schema definition are even as complicated as Access, much less SQL Server. Anything more complicated than a simple object-hierarchy model, while more versatile, will be too complex to complete effectively.

XML

The plan presented to host Outlook on MSDE focused on recreating the base Outlook scenarios on MSDE. This is a small subset of what it takes to support heterogeneous data in a hierarchical, metadata-driven, adaptive model that premeates not only the folder structures but also extends into the documents themselves. Attempting to model these constructs on a system that in its most base form requires tables to be homogenous will not scale and will force developers to understand all possible data types and all potential metadata before the application is released.

Why does Oracle 8i claim to host 146 different types of files ? Why only 146 ? The answer is simple. The developers at Oracle must understand each file type and potentially relevant metadata before 8i ships because corresponding restrictive table schemas must be created for each of these file types. For more concrete examples closer to home, try to adding a new field in a RAID database, or provide a URL link to associated source files or email discussions, or attach the written review document to a record in MS Review or describe a complex HTML page in a restrictive schema.

Platinum at its most base level assumes that any row in any table can have arbitrary metadata associated with it so developers can extend the system in ways that are obvious and straightforward and thus model real world scenarios. Furthermore, these same concepts extend past the folder hierarchy and directly into the structured documents or links to associated documents. This is precisely where Platinum's model and XML are twins separated at birth.

In the relational world, XML is viewed as an interchange or serialization format to exchange data. On the Web, XML is viewed as a self describing structured document format that can model real-world, heterogeneous, complex documents and hierarchies. These documents are queryable, support rich navigation models, links and support arbitrary metadata at any level in the document. These are radically different concepts and best illustrate the difference between the two models. With Platinum, navigation from the folder hierarchy through these documents is transparent and dynamic; where the contents of the document are only examined if necessary. With an underlying relational model, the entire document must be fully canonicalized and exploded into third normal form tables before the QP can query across the contents of the document. One of the key tenets of Platinum was to move to deferred conversion to improve our performance, fidelity and our handling of signed documents. Columns can be manifested on the fly if and only if required. The combination of dynamic indexes coupled with heterogeneous rows and virtual columns ensures that the system adapts to how end users interact with the system instead of requiring advanced knowledge at development time of how the system will be used.

In addition the data model must be part of a larger system that can unify data from multiple servers, across multiple protocols, in different data formats into a consolidated, cachable, unified namespace that can be exposed through our existing and evolved sets of interfaces to support the widest variety of applications. It cannot be only about exposing OLEDB access to a limited set of data sources. To make Windows truly indispensable the local store must be a container for all of my hierarchical data.

Tools

It is pretty clear that OLEDB 2.5 (along with ADO and data and binary schemas for common classes) is the interface we want to evangelize and build applications upon in the near term. Office 2000 uses it today for accessing HTTP, DAV and OSE servers, Outlook is moving to it, PKM leverages it, Access, and so on. OLEDB 2.5 exposes the appropriate row-level schema and semantics necessary for handling heterogeneous rowsets and object-level relationships.

Platinum certainly has the richest and most mature OLEDB 2.5 implementation today. We've worked through the hard problems of mapping namespace between OLEDB 2.5, DAV, and even Win32 and MAPI. In addition to its own implementation, Exchange has driven requirements to the DAG team, has pushed a compatible set of OLEDB 2.5 proxies into Windows 2000, and has worked with the PKM team to get a symmetric remote implementation on the client. The important thing to realize here is that OLEDB 2.5 is a very different model than OLEDB 2.0 or 2.1. The tools teams have significant work to do to surface this data model to

developers and this work is actually the same for them to support either SQL or Exchange.

Suitability for Outlook / Office 10

On the specific issue of a local store for Outlook 10, clearly Platinum is the more functional and more mature choice. Platinum gives Outlook a symmetric storage platform client and server which supports the same data model, event infrastructure, schema, same functionality as the server. But more importantly, with a Platinum store, Outlook 10 could take advantage of Platinum functionality and the unification of storage between Outlook and the rest of Office, PKM and the file system to provide capabilities above and beyond their current requirements and vision. For example, Outlook could provide for rich views onto properties prooted from Office documents and the file system. Outlook could leverage many other Platinum features such as versioning for the items they own such as appointments and tasks. Additionally, because the Platinum store supports a full MAPI interface, Outlook would able to provide richer backward compatibility for existing MAPI applications and Outlook extensions which call MAPI.

In short, there are massive advantages for Outlook using Platinum in terms of functionality and integration with the rest of Office as well as it being the more conservative option from a development risk perspective. Updated dynamic views on normalized tables is not the model of either the file system, platinum, CODA or other hierarchical systems and as a result an object model view on a relational system will NOT be symmetrical. Hierarchy into documents requires deferred conversion, method based columns and preservation of the underlying documents for both transmission, full fidelity and signed documents.

Platinum is long-term storage technology

We have a lot of very smart people in Exchange. We have been working on providing an application storage platform for many years. I would argue we have a very deep understanding of the application data model necessary for collaborative applications and structured documents and understand the appropriate balance and unification between homogenous rowsets and heterogeneous objects storage unification requires. We have built a storage system which provides a well-architected superset of the functionality that many application groups, internal and external, have been forced to cobble together themselves. Clearly, given the current momentum of Microsoft applications groups focusing on Platinum storage, we are addressing a storage need not currently addressed by any single store.

I would also argue that we have many of the best storage people in the company and that our technology is as elegant, scalable, and well-designed as any in the company and in fact, more advanced in many cases. It is also general purpose. I am disappointed when I hear people claim that our store is email or MAPI specific. Email has driven many of our storage decisions because it exhibits many of the properties (no pun intended) of the storage needs of a unified storage system, and while we do have lots of technology focused on dealing with email specific formats and protocols, the storage technology itself is very suitable for a general purpose store. Platinum has real tables. Rows can just be rows if that is the appropriate model. Exchange is not just about messages in folders anymore.

We have many hundreds of man years invested in the Exchange store. While it is somewhat meaningless to get into a pissing contest over estimates of work which would need to be done, it is clear than any SQL solution would be far less functional (in most ways) and be delivered on a far less optimistic schedule than would be an Exchange solution.

Going Forward

The nirvana of grand storage unification has been attempted in the past. I think we must be sensitive to the fact that we do not want to screw up the Exchange, SQL, Office or Windows businesses by being too aggressive with a strategic initiative. However, neither should we shy away from moving towards unification. I believe we must therefore make decisions, but pragmatic ones which cover the bases and leave our options open. The cost of duplication of effort is minimal compared to the risk of a grand strategy which fails or falls short.

How do I think the relationship between SQL and Platinum should progress? I do not have all the answers. However, I believe the right approach is three pronged: Unify many stores, client and server, around Platinum today, strengthen the ties between Platinum and SQL as a two-store storage platform within BackOffice in the near term and work towards ultimate unification in the long term.

Today, we don't really have just two or three stores, we have dozens - FrontPage, VSS, IE, the MAPI PST and OST, Jet Red, OSE, Office (Rosebud), the IIS metabase, etc. Going forward with a Platinum store plan for Office 10 leverages the momentum we have to unify all of these stores on a conservative plan in the near term and provides tremendous product and end-user benefits. Platinum is the most functional, most expeditious way to compete with Notes and Domino.

There are many things we can and should do to make application development on top of Back Office, using both SQL and Platinum feasible and desirable e.g. making sure the SQL DQP and OLAP works against Platinum efficiently, getting DTS to work with Platinum data and so on. We should do these things and more in under any outcome.

We have tremendous momentum behind the Platinum store. Many parts of the company are signed up to use Platinum based on their decision criteria. It's the right thing to compete with Notes and the right store for Office and the OS going forward.

-E

On the Windows client front, we have spent a lot of time discussing this. It is clear that providing a compelling developer message, a great mobile experience, and simplifying the installation and management of our software can solve many of our problems. This is tough but I think we all agree we need to do this.

Anyway, I didn't think all of this was summarized yesterday so I thought it useful to pull it together.

Comments are welcome.

bob

MS-PCA 1411811
HIGHLY CONFIDENTIAL

From: Bob Muglia (Exchange)
Sent: Thursday, December 17, 1998 4:57 PM
To: Bob Muglia's Direct Reports (Exchange DL); Kurt DelBene (Exchange); Steven Sinofsky; Brian Arbogast (Exchange); Richard McAniff; Craig Fiebig (Exchange); Russell Stockdale (Exchange); Betsy Johnson (Exchange); Andrew Kwatnetz; Gordon Mangione (Exchange)
Cc: Bill Gates; Steve Ballmer; Paul Maritz; Jim Allchin (Exchange); David Vaskevitch
Subject: IBM threat to Knowledge Workers

After yesterday's IBM competitive review, I thought it would be useful to send some mail which summarizes a set of concerns I have been considering. I believe that if you look at everything IBM is doing, it is fair to consider that they are executing on a comprehensive strategy to take knowledge workers away from MS. Their strategy:

1. Capture the knowledge worker data in Domino.
2. Replace the Office desktop with Notes.
3. Create the opportunity for an IBM OS on the business desktop which replaces Windows.

You can argue that for Lotus's 30M customers, we've already lost their data to Domino. As Bill pointed out yesterday, this is gone and we need to do something to get it back.

For the desktop apps, the situation is better but tenuous. Most customers who use Notes today also use Office. But if you look at the R5 Notes UI, they are clearly focused on minimizing the reasons for a user to run Office. We saw a glimpse of that with the Java HTML editor in R5. Lotus has 540 people building desktop apps and we know that most of these guys aren't working on Smartsuite. I think it reasonable to believe that these people are building Java editors which can fit into the Notes desktop and effectively replace Office.

This is very scary because if the knowledge worker only needs Notes and Domino, then we have truly lost that user. This impacts more than Office because the knowledge worker desktop is the foundation upon which we want to build for business ops and commerce. Put another way, how hard would it be to sign an EA with a customer who isn't using Office?

Lastly, although from where we sit today it might seem distant, I think we must assume that IBM's ultimate goal is to use Notes to enable the replacement of Windows on the business desktop. If a user's data is in Domino and their UI/app is Notes, then Windows is basically irrelevant for that user. We know IBM is building an NC OS but I think we could be confused about this being just a terminal replacement. With Domino as your filesystem, there is no reason why this OS couldn't support mobile scenarios. When you add a Java-based runtime environment to the picture, the Notes/Domino/NC could be wrapped up in a crisp package as a replacement for the Windows desktop and Office. IBM account control could make this proposition attractive for many companies.

So how can we win?

1. Keep the data we have. Get back the data we've lost to Domino.
2. Make Notes unnecessary.
3. Provide unique value to Windows desktops.

In the medium term, we need to enrich the filesystem to once again make this an attractive place to store documents. But for now, to keep the data we have we need to use Platinum as a rich store with the modern features people have come to expect from solutions like Domino. We need to get Platinum/PKM deployed on servers with support for symmetric client-side caching and offline capabilities ASAP. This is required just to stay in the game.

Yesterday at the IBM review, Bill and I had a "discussion" about what it will take to get back Domino customers. To start, we need parity with the Notes/Domino feature set. We also obviously need additional features beyond Notes that causes customers to choose our solution.

To help them with migration, we need to continue the work currently underway to move all the data and run Notes apps on our solution. While I don't think the way to do this is to independently develop full Domino compatibility, there is a lot we can do which we aren't yet doing. For example, we will probably need to add additional semantics to the Outlook/CDO object model to enable easy conversion of Notes apps onto our solution. I like Steve's idea to prioritize this work based on customer input from companies like E & Y. We have already done some of this but I'm not clear if we've yet done this based on the Outlook 2000 and Platinum feature set.

In the desktop space, while most Lotus customers today also run Office, we must assume that this can't last. The question comes down to: Which will they choose in the future? Office or Notes?

For Office to win, the combination of Office+BackOffice needs to easily enable the full set of solutions customers solve today with Notes. A lot of very positive things are happening around this as a part of the ATG planning process. It is critical that these conversations turn into product plans which we can effectively bring to market.

Jon and I have talked about doing a short cycle on the next release of Office to quickly get the KM benefits of Office+BackOffice in the Platinum timeframe. This also came up during today's customer feedback review. We need to decide what to do here. Another related issue which yesterday's Notes client demo highlighted is the need for us to pull together a cohesive set of developer/design tools which simplify the creation of common scenarios. It may be hard to do this in a short cycle.

MS-PCA 1411812
HIGHLY CONFIDENTIAL

On the Windows client front, we have spent a lot of time discussing this. It is clear that providing a compelling developer message, a great mobile experience, and simplifying the installation and management of our software can solve many of our problems. This is tough but I think we all agree we need to do this.

Anyway, I didn't think all of this was summarized yesterday so I thought it useful to pull it together.

Comments are welcome.

bob

MS-PCA 1411813
HIGHLY CONFIDENTIAL