

PLAINTIFF'S
EXHIBIT
2566
Comes v. Microsoft

From: Bill Gates
Sent: Thursday, October 10, 1996 11:06 AM
To: Brian MacDonald (Exchange); Pete Higgins; Nathan Myhrvold; APPS Comm/Richard Fade's Direct Reports; Peter Pathe; Antoine Leblond; Ralf Harteneck; Jon Reingold; Duane Campbell; Jon DeVaan; Ben Waldman; Steven Sinofsky; Naveen Garg; Aaron Contorer; Bill Bliss (Exchange); Mike Koss (Exchange); Rao Remala; Tod Nielsen; Larry Engel; Paul Maritz; Robert (Robbie) Bach; Ron Souza; Andrew Kwatinetz; Chris Peters; Michael Mathieu; Alex Loeb; David Goodhand (Exchange); Doug Stumberger (Exchange); Alex Barke Rector (Exchange); Grant George; Robert (Robbie) Bach's Direct Reports; Lani Ota Carpenter; Tim Lebel; Reed Koch; Eric Michelman; Dean Hachamovitch; Richard Wolf; Heikki Kanerva; Yuval Neeman; Richard McAniff; Darryl Rubin; Butler Lampson; Jim Alchin (Exchange); Jeanne Sheldon; Ross Hunter; Sam Hobson
Subject: RE: RE: Proposed: New Office Products - The Office Content and Office Document Servers

I think there are a lot of great ideas in here.

However I think we should extend the Exchange server to play this role. If we can integrate the Exchange server store with the file system in a better way ad extending it for the scenarios described below then it becomes an "advanced file system".

My idealistic view is eventually have one rich store that includes files and has the replication, rules, properties and indexing described below. One path to get there is to extend the Exchange store. I think an element of this is storing messages as files in the file system while having the richer behavior that Exchange can provide.

We CANNOT have the file system, exchange server and the Office document server be 3 different things. We won't be able to get it done and its too confusing. This means we have to merge the file system and exchange server or merge exchange and your vision for the Office document server.

I doubt our Office competitors will be aggressive in taking advantage of this "advanced file system". Corel because they won't focus their R&D on it and Lotus because it will conflict with the Notes message. With this approach we can make it part of the platform pitch and still get a competitive advantage for Office.

We should find a forum to discuss this further.

I agree with the Office Content comments except I am not sure we get to sell ads to Office customers. Its like seeing ads at the theatre. I think we should rename the Web Office group at some point so less people get confused about what they do.

-----Original Message-----

From: Brian MacDonald (Exchange)
Sent: Sunday, October 06, 1996 3:22 AM
To: Bill Gates; Pete Higgins; Nathan Myhrvold; APPS Comm/Richard Fade's Direct Reports; Peter Pathe; Antoine Leblond; Ralf Harteneck; Jon Reingold; Duane Campbell; Jon DeVaan; Ben Waldman; Steven Sinofsky; Naveen Garg; Aaron Contorer; Brian MacDonald (Exchange); Bill Bliss (Exchange); Mike Koss (Exchange); Rao Remala; Tod Nielsen; Larry Engel; Paul Maritz; Robert (Robbie) Bach; Ron Souza; Andrew Kwatinetz; Chris Peters; Michael Mathieu; Alex Loeb; David Goodhand (Exchange); Doug Stumberger (Exchange); Alex Barke Rector (Exchange); Grant George; Robert (Robbie) Bach's Direct Reports; Lani Ota Carpenter; Tim Lebel; Reed Koch; Eric Michelman; Dean Hachamovitch; Richard Wolf; Heikki Kanerva; Yuval Neeman; Richard McAniff; Jeanne Sheldon; Ross Hunter; Sam Hobson
Subject: RE: Proposed: New Office Products - The Office Content and Office Document Servers

Microsoft memorandum

to: LIST
from: Brian MacDonald
subject: NEW OFFICE PRODUCTS: THE OFFICE CONTENT SERVER AND THE OFFICE DOCUMENT SERVER
DATE: 10/6/96

Gates
DEPOSITION
EXHIBIT
36
2/28/01

Overview

The goal of this memo is to build awareness of and hopefully momentum behind two new products that I believe should be built

for the next major release of Office: The Office Content Server and the Office Document Server. I believe that these products represent some of the lowest hanging fruit we have for generating incremental revenue for DAD and excitement about Office for our customers. This memo is a collection of my thoughts and those of many people around DAD, (including Richard Wolf, Mike Koss, Ross Hunter, Jon DeVaan, Stevesi and others) that have been bantered about in recent conversations, meetings, and retreats.

Background: Office Ain't Office Anymore

How we should define Office has changed. In the past the motto for Office was "The best tools for creating documents". This motto served us very well for years as practically the only thing business users did with their personal computers was create documents. That is changing.

Clearly the momentum of the Internet is huge factor but it is not the only factor. As the penetration of desktop computers increases per capita throughout the world we are finding our new customers to increasingly be what I call more "marginal" knowledge workers. Our document creation features are less important to their jobs (modulo email). For the first time, we have a class of users whose managers didn't plop computers on their desks for them to make complex spreadsheets with. Some only used a dumb terminal before strictly to access PROFS. Some only expect to use their PC to review database reports. Some only to read memos. Information browsing is more common an activity than information creation.

For these reasons, we now need a broader motto: "The tools you use during the course of your day". More important than a motto is an attitude. Where do we draw the line on the products that Office (DAD) creates? Before, any document processor of a sufficiently horizontal nature was our turf (e.g. PowerPoint, Publisher YES; AutoCad, Pagemaker NO). Now our goal should not be tool-creation oriented. Rather it should be to try to capture with DAD products as much of the time that each business user spends behind their computer as possible - the "market share" of business computer use.

There is a lot we need to do to dominate in business computer "use" share the way we have in tools use share - broader email back-end support, adding value to web browsing, etc. I believe that these two proposed products will be key elements in obtaining that share.

The Office Document Server

What is it? The Office Document Server is a server-based store of documents with rich features for viewing and access control and an object model api for accessing them from multiple clients. It necessitates new client (i.e. Word, Outlook, Excel, PPT) support for both posting and viewing.

Publish is the default. One of the unfortunate things about our current products is that you very rarely bang out a long Word document or a complex Excel spreadsheet just for yourself but yet that is precisely our default - the doc is saved to your hard-drive where generally only you can see it. The Office Document Server is a necessary ingredient in this "Publish is default" mentality that Stevesi has been evangelizing.

Rather than thinking about a hard location for your document on your hard drive (which most people don't really do anyway - they just end up with a pile of stuff in My Documents), you mainly think about the audience the document is intended for. You "stamp" the document with a tag for the intended audience. This stamp is replicated along with the document and all its other properties up to the server and is used automatically to control access to the document. The default stamp is probably to allow access by everyone in the domain that can access the server. Stamp it for just your direct reports, your product group, your peers in Testing, whatever.

Background replication of documents from local to server store and vice versa. Almost like an extension of AutoSave, your documents get replicated in the background up to the server. You don't worry about losing your documents when your hard drive goes wacky. You can always access your documents from a co-worker's machine. You can always access them from home, even if your office PC is shut off.

As in synchronizing interesting email and email folders for off-line use, you can also determine criteria for documents posted to the server by others that you always want replicated to your own hard drive. Never again will you be forced to watch a cool movie on a flight across the country - your laptop will be full of specs and proposals and other docs you haven't read even if you are all caught up with your email!

Is this just Saros / FileNet? I've been asked this several times. Although the existence of an Office Document Server can form the basis of a complete document management system, the motivating features of this product do not include those mostly associated with a high-end document management system. The key features are the automatic replication and making documents easier for people to access. The capacity to track a large number of documents and support of content indexing is certainly there. Versioning, check-in / check-out, etc. is a like not a must. Other more vertical features of document management systems, e.g. tight integration with document imaging systems like Watermark is almost certainly a no.

Ideally we would position the Office Document Server as a platform on which the document management vendors would build their systems and add value. Our spin to them is that while it does encroach on their turf, it generates awareness and demand on a broader basis for high-end document management solutions.

Besides componentizing the server to allow piecemeal replacement, this means writing clients and viewers to a standard interface

MS-PCA 1391909
CONFIDENTIAL

such as OLE/DB and exposing a good object model. Vendors should be able to completely replace the data store or add client or server features.

Cool views on interesting properties. Key to leveraging the Document Server is that there is a rich set of properties (that aren't blank!) which accompany each document. Then the interesting views can come - e.g. "Show me all the documents written by Excel or PowerPoint product planners that have to do with drawing created in the last 3 months". Properties such as "Intended Audience", "Valid until", etc. are now useful to ad.

People have ignored filling out the document properties in the past. Hopefully this will be overcome when they know that its useful to fill them out, but we will have to expend energy on features that automatically fill-in the properties. The classic and cool example that comes to everyone's mind is Word AutoSummary as a property. There are others that are interesting once a doc is in a server repository like "Number of accesses". You want to be sure you are reading the memos everyone else is so you don't fall behind.

Platform for a different set of "Auto" features. Clearly Office's bread has been buttered in recent years from upgraders who saw the new Auto features and considered them must haves. Rummaging out of those to add in the future is a bit scary, not only for losing the upgrade business but because while generally very compelling, they are not incredibly difficult for our competitors to duplicate. The Office Document Server provides a basis to implement a new wave of Auto functionality - AutoPublish, AutoReplicate, AutoArchive, etc. AutoSuggest could look at what you have been reading, perhaps also utilizing feedback from you on which docs you have valuable found or not, and suggests other documents that you probably also want to read.

A different paradigm for sharing. The user interaction model for documents on the Office Document Server is really the same model as Exchange or Notes public folders. This really is a superior model for document sharing in many respects. One of the problems with Email is the implied assumption on the part of the sender that the recipient will read it. The sender tends to hold back on who is on to To: line and recipients feel bad about deleting something without reading it. The recipient problem is made worse with Outlook where it is easy for the sender to be notified regarding whether or not you've read the document!

An example. There are many memos and documents that I write now and send out in Email that, for instance, I do not send to Grant George. It is not that I don't want Grant to read them - it's that I know that if I send them to Grant he *expects* that I expect he would read them. Grant is a busy guy with a lot of responsibilities that don't overlap with mine so I don't want to necessarily saddle him with having to read something he would not care about. But very few of these documents would I actually want to restrict Grant from reading. Suppose Grant decides that he was in the mood to read some documents that tend to be long-winded and oversimplify a set of problems. He can go to a view on the server, filter documents authored by me, and access a bunch that he could probably not get to from his Inbox. It's the same key benefit of an Intranet - increasing the easy access to and availability of information.

Is it just an Exchange public folder then? There could be a time in the future when it would be but it isn't possible today. For one, Exchange does not have built-in content indexing. Also, the replication isn't completely symmetrical with respect to client and server. Object and field level security is a must. Even if the datastore was the Exchange MDB there still would be a need for more code to provide a higher level "document server" api for ISVs and code for the HTML rendering, etc. A different store is required but integration with Exchange is important.

Integrate rules and documents. Once we have a central server repository for documents and a server process, it is easier to integrate Email-style rules to allow customer configuration. The scenarios become more compelling and the server is necessary to really do rules right (you need rules to execute immediately upon certain events, such as the posting of a document, and this should not require that your client machine be turned on and connected to the net at precisely that time).

Notification integration with the inbox is a great example of leveraging rules on a server store of documents. For example, I want to create a rule that always sends a message to me whenever Nathan posts a new memo. I could also leverage the full feature set of Outlook rules, specifically exceptions, so that I could get notified whenever Nathan posted a memo EXCEPT when he posts one of his memos about how to make a presentation (just kidding Nathan).

Infrastructure for "Domino for Docs". Domino is Lotus' product / strategy for bringing a rich viewing and interaction experience with Lotus Notes apps and data from just the Web browser. The beauty of Domino is that no special-purpose client code has to be installed on the user's machine yet they can experience a very rich subset of the full Notes functionality. This supports the roaming user scenario very well, among other benefits such as cross-platform ubiquity.

The Exchange team (WebConnector) and Outlook team (WebLook) both have products under development to compete with Domino for email / scheduling / PIM / groupware data. Both products are based on the same backend technology - Denali (VBScript for the IIS server) talking through OLE automation interfaces to Exchange server store data. WebLook's goal is to provide as complete a facsimile of the Outlook user interface as possible within a web browser.

The Office Document Server could serve the same role as Exchange does for WebLook in forming the basis for document collection browsing and document viewing. WebLook will utilize a version of Outlook on the server for the Denali scripts to call into and help with the high-fidelity rendering. Server-side code rendering Office docs into HTML would be needed - that is one of the key differences with the Domino for Docs idea and the Office 97 web features. For maximum ubiquity documents are not viewed using ActiveDocument on the client but are strictly served up as HTML documents.

The Complete Document History. Once on the server, it's more practical to think of storing a complete history of what has happened to a document including all of its' interactions with humans. Versioning is the classic and most often implemented example but certainly not the only interesting one. What about tracking everyone whom has read the document? Who printed it / faxed it / emailed it; when and where? Who was it sent to? What web sites has it been posted to? By whom? What other

documents link to it? What comments have people made about the ideas or data presented in the document?

Telephone interface to server; fax back documents. Imagine being able to call into the Office Document server and, through an annoying push button interface or better yet speech recognition, be able to interact with all the documents in your organization. You could have a document read back to you through text to speech or faxed back to a number you specify. You can "browse" through a collection of documents before you select one by having their properties or summary read to you, or have a listing of documents based on some criteria you specify faxed to you to enable you to pinpoint the one you are interested in.

This is probably a separate product that leverages Office Document Server, rather than being just extra functionality alongside it. It gets more interesting when combined with Outlook so you can have email, your schedule, a contact's phone number, etc. read back to you. Lotus (for Notes) and Novell (for GroupWise) have had products with this functionality for several years. Their sales have been hampered by limited and expensive hardware support. Through our telephony and SIPC initiatives the barriers are likely lessened.

Automatic links with the directory service. Once you have data stored in a database, it's easy to have server-side processes that link that data to other databases. One of the most interesting databases with which to link Office document data would be a directory service. From the server doc, you can get at the relevant directory service entries - personal properties of authors, readers, etc. (like double clicking on a resolved email address today). From a directory service entry, you can see the set of documents that a particular person has created.

Easier Workflow. By having the documents in a central repository, industrial-strength workflow applications become much easier to create. This includes not only relatively simple workflow scenarios that many users with a little VB knowledge could create but by leveraging Viper (the Distributed Transaction Coordinator) and the Office Document Server, I think we can also give corporations what they need to build true industrial strength enterprise solutions. Modifications, additions and deletions to the Document Server could be wrapped up in transactions with other operations (e.g. db posts, delivery of a piece of mail) which makes programming easier because they rolled back together easily on failure, etc., even modifications across multiple servers.

The Office Content Server

What is it? I see the Office Content server as an extension of the mission of the Web Product Unit, fusing it with an infrastructure and a delivery vehicle for more ISV add-on products, broadcast news content, and internal corporate news content.

Note: There has been some confusion about the role Web Product Unit because of its name, so just a explanation that this is *not* the group charged with implementing the new web-based evolution of Office ("Watchtower") that Chrisp, Stevesi, and others have been championing.

The current WPU mission. My understanding of the first mission of the Web Product Unit is to take the Microsoft Office Web site and make it first class. Next step is to get it and its associated content: help, tips, templates, product information, etc. along with all the product bits we produce and allow it to be brought into an organization in their own proxy of the Office web site.

Corporations would create their own "typical" install and would be able to maintain an Office Setup homepage that includes the add-ons, templates, etc. that are blessed by the IT organization. QFE fixes and other drops are synchronized with Microsoft whenever the customer is ready. Corporations tailor help and add their own FAQ documents. The goal would be to help customers reduce TCO.

Next on the current goals list would be to gather more traditional Office content - templates, add-ons etc. and add them onto the "site". There would be enough stuff there that you have a very enjoyable first visit and want to come back later for more. A constant stream of new stuff would be up there for repeat visitors.

Overtime, the infrastructure for charging for accessing this content would be developed. Only members would get the good stuff and the best treatment.

Leveraging that infrastructure. OK. So far so good. In terms of helping corporations with TCO, we have a very compelling product. In the minds of the average user and how it impacts them though, it's probably a single not a home run. The good news is that the product and infrastructure we have created is applicable to bringing in other content that will make a difference.

Increasing exposure and revenue opportunities for ISVs. I think it's safe to say that the minds of the ISV community are not very focused on Office. Solution providers yes, ISVs no. There are many reasons for this, including product ones that are being addressed, but exposure for their products is a key factor. There just aren't many products that are going to get shelf space at Egghead or Computer City. Links from our web site to theirs aren't enough. The Office compatible program isn't a big asset anymore.

We need a serious showcase for ISV products via the content server. We need to allow corporations to pick and choose from the add-ons and bring them onto their Office.com proxy as integrated pieces of our configuration. We need to pay ISVs part of the "membership" dues we collect in return for having access to their products as content. We need to create "virtual SKUs" of ISV add-on bundles that are easier for LOGs to justify purchasing large numbers of licenses for, and we need to have the infrastructure for doling out appropriate shares of the pie that is collected from the purchase of the SKUs. We need the big carrot to dangle so that we can recruit and engage ISVs to focus on adding value to our products rather than Netscape's.

OfficeCast. Our relationship with so many millions of the world's most sophisticated workers is a huge asset. Our main product has been so successful and the opportunities with it so large that we haven't been able yet to really pay much attention to affinity

MS-PCA 1391911
CONFIDENTIAL

revenue potential.

OfficeCast is what I call our integrated Pointcast competitor. We have the opportunity to make a much more interesting and integrated product than Pointcast but even if we did nothing but a "me-too" it's an opportunity. Pointcast is doing very well. They are receiving over 4 cents per 1k gross impressions and are sold out to the end of the year. Their ShockWave ads are fun and, to me, not overly annoying. Unlike television I can tune them out when I want to concentrate on the real content.

I think its possible over time to generate around 100 million gross impressions a day from Office/Outlook users. This would generate about \$400,000 a day. Times 22.3 business days per month times 12 months we'd have about an extra \$100 million a year in revenue.

We can surpass Pointcast in UI presentation (look at the Pandora beta), integration with general purpose offline downloading (leveraging Nashville's WebCheck and our deal with FirstFloor), email notification, tabular info as Excel docs, etc. It also could be a boost for MSNBC.

Corporate News. The tools for distributing and viewing Internet content can also be leveraged for the Intranet. At first this might seem unnecessary - corporations are already publishing Intranet content at a dizzying rate. That is true but the opportunity lies in making that content easier to consume. The web is still oriented toward the model where you go fetch the info rather than having it come to you, and also so much of what you fetch is discarded. A Pointcast UI for corporate news data is very appealing. The personalization features can be applied so that you filter and tailor your personal edition of the corporate news.

Outlook PIM content. The content server is also a great vehicle for collecting and distributing PIM objects, especially locale-specific ones. For example, school schedules, College sports schedules, restaurant phone numbers, etc. Much of this content is probably already in mind to be generated for Cityscape.

Build annuity revenue; reinforce other revenue. Maintenance revenue today is our annuity revenue and it is sort of like the original Office when it wasn't much more than the sum of its parts. The Office Content Server with the associated content and capabilities discussed gives us the ability to deliver some real annual content other than our product bits, justifying the annuity revenue stream we want to take in. The Office Content Server gives us a lot of different revenue streams that reinforce each other: the membership fee, the Content Server SKU price, the Win NT Server price, the ad revenue, the CAL license (see below), and the client upgrade revenue.

The Crude and Handwavy Business Case

One could argue that the servers have sufficient merit to warrant investment in developing simply out of current business preservation concerns, but I would like to see if we can do better than that.

We think a lot with Office about our revenue per PC or per copy of Win95 sold. We need to start thinking about how we grab revenue from as many copies of NT Server sold as possible. If we could reach 10,000 copies a month of both servers and net \$500 for each of them then we have a business approaching \$125 million. (This volume would necessitate penetration into MORGs and SMORGs where people per server would be small; i.e. 15-20) The fun money comes though when we take into account that we would not actually give customers anything for their \$500. To really use it, they must purchase a Client Access License for each workstation.

At a price of \$50 each for the client access licenses, revenue adds up fast. If we create a compelling enough product and can bring in 300,000 users a month that means an extra \$360 million a year, or a total business approaching half a billion dollars a year. Like Project, the server price point yields a very attractive profit margin and the CALs, net of an increased support burden, are basically found money since they don't involve any extra bits.

Corporations would likely need fewer Office Content Servers than Document Servers, especially in smaller organizations. This could possibly be adjusted for by a higher price for the Content Server, and justified by its impact on TCO among other things.

Other Benefits

Increase switching costs. Obviously the more we can do to have our customers depend on other products of ours the better. Having customers depend on the Content Server and Document Server (especially once its involved in mission-critical customer apps) would make it much harder to scrap Office in favor of a collection of Java applets.

Up the ante on what a Suite is. Its getting harder for us to add more commands to our base apps and leverage that into an upgrade justification. It's a boon to our competitors who aren't really able to survive a checklist war. As alternatives, Corel is trying kitchen sink product bundles with one hand and a less is more strategy with the other (JavaOffice). Lotus just emphasizes Notes. While its hard to pile on the client app features, we can't at the same time be static. New server products and the clients to go with them will help progress the definition of a state-of-the-art suite and keep us on comfortable battle turf.

Reducing the rationale for non-concurrent use. Currently, DAD loses a very large amount of money due to our licensing provisions that allow corporations to buy fewer copies of Office because they can claim a lack of concurrent use (e.g. "Of these 12 secretaries, only 2 run Excel at any one time"). Whatever we can do to get every Office user executing our bits all throughout the day will take away the potential for our customers to claim that they have many users but not much simultaneous use.

MS-PCA 1391912
CONFIDENTIAL

This is big money - our best estimate based on a study done for DAD by the Boston Consulting Group pegged the loss at 17% of revenues. Besides what we are doing already with Outlook and increasing PIM and email usage, getting our customers "hooked" on frequent content accesses (internal or external) will be important for reducing the concurrency argument.

Can we afford it?

I'm suggesting we start serious efforts on these products at a time when there are many other needs for headcount within the Division. Certainly the Watchtower project deserves a decent number of people. Consumer Productivity has many good ideas that should be pursued. Word and Excel are more threatened, at least in publicity rhetoric, than they have been in many years. Outlook hasn't shipped v1 and analysts are attacking it for poor Internet standards support (RTF rather than HTML, over-reliance on Exchange, etc.) Web-centric RAD tools threaten Access. Can we really start two new ventures?

I don't think we can afford to bypass these opportunities. While it has been prudent in the past to keep an eye on revenue growth and limit headcount growth to a comparable percentage, I don't think that is now a prudence we can afford. We know that the death from saturation is less looming than we once worried about, so that helps.

As our business matures, I think we have to take the opportunities we have to increase profit, even if it means doing so at a smaller margin. Would we rather make an extra \$400 million and net \$200 or an extra \$800 and net \$350 - I suggest the latter, especially since there is a degree of zero-summing with our competitors that helps avoid future battles.