

From: Eric Rudder
Sent: Monday, December 07, 1998 8:04 AM
To: Bill Gates
Subject: RE Office rendering

my point abt DAV has always been that we messed up in not making DAV do all the things that we want in PROTOCOL+. you can always defend it along lines of reasoning, but we didn't do a general job in ways that we should have. the standards thing is another issue, but for me, it's not really the primary one.

-eric

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

From: Bill Gates
Sent: Monday, December 07, 1998 8:02 AM
To: Eric Rudder
Subject: FW: Office rendering

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

From: Bob Muglia (Exchange)
Sent: Monday, December 07, 1998 7:20 AM
To: Bill Gates
Subject: RE: Office rendering

Plaintiffs' Exhibit
2991_B

Comes v. Microsoft

I think the concerns about DAV are a red herring.

First, in order to support the Internet, we needed to expose the full richness of Exchange on a stateless protocol. MAPI is based on synchronous RPC and this protocol is inappropriate on unreliable networks. 99% of Outlook hangs are caused because Outlook is blocked waiting for an RPC response which will never come because a packet was dropped. While we could change timeouts, etc., a stateful, synchronous protocol will never be appropriate for the Internet.

Once you accept the need for a stateless protocol, HTTP is the obvious choice. It allows unification of messaging and database applications onto a common protocol. It goes through firewalls and of course is ubiquitously supported in the network infrastructure.

The DAV verbs provide a small set of file system capacities on top of HTTP - directory enumeration, basic query, and snapshot/putback of files. DAV is very minimal and with the exception of basic versioning, represents a subset of the technology we supported in SMB in 1988. By itself, it certainly can't come close to replacing the MAPI protocol. Instead, we leverage the small subset of DAV capabilities and add many proprietary extensions - advanced query, cursoring, replication, link fixup, advanced versioning, checkin/out. .

The advantage of using the DAV subset is that it allows our client apps and tools to interoperably support other platforms while still providing considerable advantage to the user when used with our servers. For example, Front Page will move to supporting DAV+MS Extensions as their primary protocol in their next release. This means that Front Page will be able to support Unix servers in a basic way (less functional than today's FP extensions) without any code running on the server. On the other hand, they will provide huge advantages when running against Platinum/PKM (link fixup, versioning, property promotion and indexing, workflow process, document vocabulary, checkin/out, etc.) DAV will make it easy for Unix-based ISPs to offer minimal support for FrontPage and at the same time, the MS proprietary extensions will provide huge benefit to moving to an NT server.

On a more fundamental level, I believe that our core assets and intellectual property reside primarily in the IMPLEMENTATIONS of our apps/services, not in proprietary APIs and protocols.

It is certainly true that we need APIs/protocols which are technically appropriate to provide the full richness of capability. We also need complete freedom to innovate.

But I believe that the long-term strength of the assets we develop are based on the depth of the underlying semantics of our implementation. When an ISV builds an app on one of our services, their dependency on MS is completely

MS01 0094992
HIGHLY CONFIDENTIAL

proportional to their reliance on the semantics of our service. The API (or protocol) is just a reflection of these underlying semantics. Yes the API needs to be good, yes it needs to be unique. But whether it is rooted in a standard or not is secondary.

A good example of this is HTML. The investment we've put into HTML has created very unique semantics. We are now to the point where an ISV who uses the XML/XSL capabilities in IE 5 have a deep reliance on MS. Clearly, at least one ISV (Office) is deeply taking advantage of these capabilities. Now obviously, many ISVs still use HTML 3.2 because of its ubiquity, but this is not because HTML is rooted in a standard - they just want the reach.

There are lots of good examples where a proprietary API didn't provide any long-term advantage. Netscape's object model and scripting language were proprietary. However, because their investment in Nav 2's object model and livescript was a handful a person-years, it was relatively straight-forward for us to independently engineer this and create a compatible implementation in IE 3. In this case, there was insufficient time for Netscape to develop the complex semantics needed in their implementation to create a long-term asset.

Likewise, Novell's IPX and NCP are fully proprietary. NCP isn't even published. Yet here again, we were able to create an independent implementation. Why? Using a sniffer is not difficult so creating the documentation was a doable job. And the underlying semantics of the Netware file server are generally similar to what we implemented in NT. So hooking up their proprietary protocols to our implementation was easily within our reach. Novell's mistake is that they didn't build more sophisticated functions into their product and create an ISV reliance on this. If, for example, they had built a sophisticated file system with property indexing into their server and gotten apps to use this, we still wouldn't be able to have an answer to this.

In contrast, there are some proprietary implementations which haven't been cloned. Sun failed in their Windows clone. Why? Not because the Windows API is proprietary. Creating their own windows.h file from our documentation is the least of their problems. IP may have been an issue but they failed on their own. I submit that Sun failed because the semantics of the Windows implementation (particularly user) are unbelievable complex. Creating an independent implementation of the APIs and messages is within reach; what is hard is matching the underlying semantics in ways which ISVs rely on. Also, the breadth of Windows services (including COM and OLE), make this a huge, huge task. The key to the Windows asset is the thousands of person years we've invested in the implementation.

Likewise, I submit that it would be very difficult to build an independent Notes implementation. Again, it is not because their protocol is proprietary. The same sniffer which worked for Novell will work for Notes as well. The key asset which Lotus has created is a deep set of semantics which applications rely upon. That implementation has been built over a 10 year period and undoubtedly took well over a thousand person years to establish. Anybody who expects to create an independent version of Notes better be willing to spend hundreds of person years to match the semantics that Notes apps rely on. I believe strongly in interoperability with Notes and have obviously made significant investments in this, but that is different from creating a fully independent implementation.

So the bottom line is that I don't believe DAV is big exposure for us. First, it is a relatively minimal set of capabilities. It hardly represents the full set of Platinum semantics. Second, I believe that our real asset with Exchange/Platinum exists in the implementation. The store, all the clever work that has been done to get great performance, the events, all the specific semantics of the implementation. Of course, the importance of this is based on getting apps to rely on these.

Fortunately, I think that the success of Exchange and the excitement we can generate from Platinum will generate lots of apps.

bob

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

From: Bill Gates
Sent: Saturday, December 05, 1998 5:09 PM
To: Steven Sinofsky, Bob Muglia (Exchange); Jon DeVaan
Cc: Paul Maritz; Eric Rudder
Subject: RE: Office rendering

I think the current support we have is just right for both technical and business reasons.

MS01 0094993
HIGHLY CONFIDENTIAL

Its right for technical reasons because the team worked hard to support old browsers as much as they could.

Its right for business reasons because it supports competitive browsers but with a clear benefit for people who use our browser (particularly IE 5).

What I trying to say is that looking forward we should not do heroic things like add new capabilities to the

standards to help Office

We should look at even patenting the things that we do add to help Office.

I need to learn more about this whole DAV thing.

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

From: Steven Sinofsky
Sent: Saturday, December 05, 1998 4:39 PM
To: Bill Gates; Bob Muglia (Exchange), Jon DeVaan
Cc: Paul Maritz
Subject: RE: Office rendering

Office does not love DAV. In fact we, I, didn't want to support it at all, but the Exchange team delivered our abstraction layer (the derivative of OLEDB that works against FrontPage). It was not something we needed, and several times pushed back since it made the FrontPage case we cared most about more complex and inefficient. I personally think this is an area that has been oversold as a benefit and in terms of interoperability. In essence, this is a proprietary protocol for us anyway since we are re-building MAPI on top of it. Nevertheless, Office 2000 will be able to save/load against FTP, FrontPage, SMB, and the Exchange/IIS DAV server. But DAV servers (to the extent they really exist) do not support any of the richness we have with FrontPage 2000's server extensions such as link fix up, checkin/checkout, page themes, site statistics, etc.

For me, DAV is a case where Microsoft is out there leading with the newly proposed (by Microsoft) but yet to be implemented "open" standard. In contrast, HTML is a case where we are dealing with an installed base and standard that already existed and our conflicts are how to work within that environment.

For all practical purposes. Office 2000 requires Windows and IE. We started the project trying to be *great on all browsers, and even greater on Internet Explorer* (from our vision and presentation we did for you), but the momentum inside the company essentially prevents that message from making it through development. Only the most basic rendering works in other browsers--IE is required for:

- PowerPoint (the default output is IE only, and that is essentially IE5)
- Access Data Pages (IE5)
- Web Components (IE5)
- Reasonable performance in Excel (due to big tables and the IE5 support for a predefined table width)
- Word and PowerPoint output tons of stuff that only looks good in IE due to the shared line layout code and bugs in other browsers implementation of CSS (which is essentially an IE-specific feature)
- HTML email essentially requires Outlook Express or Outlook
- Vector Graphics (VML which renders using vectors rather than GIFs) requires IE

to name a few. I think these are enough to convince people that Office requires IE in a proprietary way and that if you want to exchange documents, the odds are your recipients won't be happy with anything but IE.

On top of that, we have dozens of features in the product that require IE4 and many that require IE5 -- this is in order for them to run at document creation time.

I totally understand where you're coming from, but in trying to decide what to do it isn't that black and white for me based on the experiences I've had personally with people. We have talked about this a lot and I really do need your help. If Office documents can only be rendered in it is a complete non-starter with customers. This is not a religious issue, but just a practical one.

If Office documents only render in IE then there is zero chance that anyone will be able to use Office to create documents that will be shared outside an environment with the standardized Window browsers (intranet perhaps, but only perhaps given the time to migrate and the minority of Win 3.1, etc.) Personally I put pictures of a trip out on sinofsky.com that were made with PowerPoint 2000 and got a dozen messages from friends and family (including a webtv person) saying they could not see the pictures. Everything I've posted here at the business school has been "recalled" by me because students were not able to read it (all sorts of combinations of OS/browsers).

No area of the product has received more skepticism and push back than our HTML output--from reviewers, analysts, and beta customers. The other night I attended a 500 person Office 2000 event in Boston (the "Team Web Tour"). The whole presentation was in IE and every time the browser was shown hands went up to ask "what about non-IE browsers?". Finally the demonstration showed powerpoint 2000 in IE which is **awesome** output--then showed the non-IE output and it was just ugly (didn't scale, fixed size slides, no slide show view, no DHTML, etc.). I thought the audience was either going to get up and walk out in disgust or rush the stage in protest.

MS01 0094994
HIGHLY CONFIDENTIAL

Again, I really understand the business issues and strategic issues. I think we're just faced with the reality that if we require IE for rendering as an explicit choice (that is when you load a page it just says "You're not running IE") then we are just saying that Office's HTML is a demo feature and not for practical use. If we didn't have HTML support in Office 2000, then I'm still convinced we would have been working on a release that customers would have viewed as utterly irrelevant--creating web documents is what people need/want to do: with Office or without Office. That's the catch-22 I feel we're in. Unless things change a lot, my feeling is that an upgrade to Office 2000 is already in jeopardy with customers that do not use IE and any higher level of requirements will drive our upgrade changes way down.

I think this knob will continue to turn **even more** towards IE over time as Windows/IE continues to achieve success. I suspect that each release of Office will continue to require more and more of IE. But in order to even be in the consideration set we will have to have some amount of downlevel support that customers will tolerate if they want to exchange information in a professional manner.

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

From: Bill Gates
Sent: Saturday, December 05, 1998 12:44 PM
To: Bob Muglia (Exchange); Jon DeVaan; Steven Sinofsky
Cc: Paul Maritz
Subject: Office rendering

One thing we have got to change in our strategy - allowing Office documents to be rendered very well by other peoples browsers is one of the most destructive things we could do to the company.

We have to stop putting any effort into this and make sure that Office documents very well depends on PROPRIETARY IE capabilities.

Anything else is suicide for our platform. This is a case where Office has to avoid doing something to destory Windows.

I would be glad to explain at greater length.

Likewise this love of DAV in Office/Exchange is a huge problem. I would also like to make sure people understand this as well.