From: Steven Sinofsky
Sent: Sunday, January 24, 1999 11:15 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

AndySchu is the place to start.

I'd also suggest talking with grantg after we ship. There are a number of office scenarios we should begin to think about including using Exchange as the repository for a network install of office, templates, linked workbooks, etc. All of these are very basic file operations we should look at.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Sunday, January 24, 1999 10:42 PM
To: Steven Sinofsky
Subject: RE: checking something

I spent sometime talking to folks about this today. Apparently it may be easier than I expected as FrontPage adds very few new primitives to the namespace that is published to end users. This is where all of my concerns were. Most of the work is done thru the WEC provider DLL thru post commands in a separate administration virtual root. I'm having some developers run tests tomorrow to see how feasible it is. I should know more tomorrow.

Who can my IFS test lead, Adam Green, sync with on the FrontPage team to get their set of tests to verify our support ? Ideally, I'd like to run the same tests on our IFS that the FrontPage team runs on FAT or NTFS. I'm assuming this test contact could also give us the full list of server extension DLLs so we can enumerate all of the file system calls FrontPage requires. With this list I can bump the priority on any calls FrontPage makes that we don't currently support.

-----Original Message-----
From: Steven Sinofsky
Sent: Sunday, January 24, 1999 12:03 AM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

I need to learn more to understand the last paragraph. I'm just not understanding it. We'll talk more.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 6:30 PM
To: Steven Sinofsky
Subject: RE: checking something

In many ways I believe we are saying exactly the same thing. I am doing everything in my power to make Platinum look and act like just the file system and exposing this functionality over Microsoft's existing SMB file system protocols. At the same time, I'm adding functionality on top of this file system to support attributes, views, column level security, events, rules, workflow, submit mail messages, links, nested search folders... and allowing apps to use OLEDB to access this new functionality. In fact apps can OLEDB queries to search the data and then call CreateFile on the results to open the file. Believe me, I get the religion that people shouldn't have to change their apps to simply work. Today I have to simultaneously support POP3, NNTP, SMTP, IMAP, MAPI, ccMail, msMail, Notes, x.400, groupwise, profs, Netscape, SMB and HTTP access to exactly the same data. In this area I think we're totally in sync.

With the local store, I can bring Platinum extended functionality down to the client and synchronize with multiple servers over various protocols and expose all of these server's data through existing interfaces - including Win32. Yes maybe I'll need to do some hacks on win9x to really pull it off but I am determined to make it painless as possible to get existing apps working against this store. I would nuts to pursue any other plan and I believe its the biggest advantage we'll have over Notes which requires people to write to the Notes api to even view their data.

Best I can tell the only area of disagreement we have is whether I should have exposed Exchange services natively over the web for HTML, IE5 and OLEDB clients or relied exclusively on FrontPage as my web remoting agent and all of the restrictions that implies. Unfortunately, the FrontPage server extensions do not include the additional functionality Platinum supports today and I

needed a way to remote Platinum functionality over the wire to clients. I couldn't extend SMB and its corresponding client interfaces, FrontPage was working on another release, MAPI is barely moving forward, OLEDB was the direction I was told to charge and I knew I needed to support native web access to Exchange. Realizing that HTTP was our pure superset remoting protocol, we decided to build a local store that synchronized over this protocol - and others for back level server support - which exposed all of this data through our 3 most important interfaces: Win32, OLEDB and MAPI. The main goal was to support existing applications but still allow the additional functionality and semantics to flow over the wire. FrontPage extensions couldn't help me and there is no way for Exchange to extend the set of HTTP primitives supported on FrontPage virtual root. Unfortunately this means until we converge plans, we'll have two virtual roots exposing the same hierarchy when you run FrontPage server extensions on Platinum machine. Personally I don't think its the end of the world - but it definitely points out where we can do a better job going forward.

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 5:19 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

Progress...

FrontPage namespace == file system namespace. That's all. The web is also the file system namespace. Now some clever folks came along and made web pages appear out of thin air (dynamically generated) and those have their own namespace, but since they came out of thin air you can't really browse them or enumerate them so really it is unfair to say they are part of a namespace. A namespace is something that can be enumerated. FrontPage is a namespace because it just uses the file system. When you're in the frontpage explorer you're just looking at the file system except that there are some special files FP knows about that are interpreted. Word is the same way--you don't have to know about all those temporary files but they are in the plain old file system namesapce and word knows how to find them and interpret them.

I'm a simple person. I understand the file system and everything on planet earth does.

So for me, for Platinum to be insanely cool it has to just be the file system. If it does that, then I totally understand how to add value to it. Let's just say the platinum file system is a normal file system in all respects (i.e. file save/open, frontpage server extensions, and most importantly when I install Platinum on my fully functional NT4/IIS/FP2000 server my normal wwwroot just keeps working but now my machine just turned into an awesome mail server too!). No new protocols, no new APIs. Things just work. File system APIs are something we understand incredibly well in our products. The semantics have been well-established for 15 years and the code to manage them in Office is baked in to the core. Changing that is mind-boggling hard.

Then what can I do on top of that is only limited by my imaginitive uses of the features that Platinum adds. Right now I know of 3 big features:

attributes -- Wow how cool that I could come along and add attributes to my files. These attributes can be secure (see next item) and by the coolness of MAPI [sic.] I can make cool views on those from Outlook 2000. Now in Outlook10 and all the Office10 applications we can make it very easy to add attributes while I'm creating messages and or documents. Heck, in the next frontpage we could even do that as well. Older frontpage just keeps working and blindly ignores all the new goo (sure some stuff might break, but it still works). Once I have attributes, I know how clever people can be about creating views on those, and writing code to muck with the attributes and do special things when the values change and events are fired. I understand how the lame File Properties tab in Office can be made much cooler and how people can write all sorts of solutions based on tweaking those attributes or counting on them being certain values. Whole libraries exist because attributes are assigned by a third party to books that were printed 100 years ago!

security -- platinum adds item security on these things. In some sense this is catchup to NTFS, but still we don't have this really exposed in Office today or in FrontPage, and certainly not in plain old threaded discussion in Outlook. Most excellent and easy to expose assuming we don't have to rewrite the entire transport to do so [sic].

workflow -- I really have to claim complete ignorance of this area. But I can just assume that through a combination of attributes, security, and code along with events on the server neat things can happen. Well we can set up those neat things from within Office10. Again, I could have a normal frontpage 2000 web and with the new frontpage things just continue to work. But if I wanted I could add some neato workflow things and somehow fun things happen. My simple brain can imagine a dialog box in frontpage that allows me to say "hey when this document changes, send a message to x" or a dialog in excel that says "do not display this sheet until my boss comes along and changes the approve property to yes".

btw--some other things that will just continue to work if you're a plain old file system:

*   embeddings in Office application (sound, video, etc. in powerpoint)

*   linked workbooks -- excel power users have workbooks connected to each other and these are done through plain old file system semantics
*   document templates -- word's templates are used by tons of ISVs and all they are are just other word documents we get to via a UNC

We have investigated getting to these things via URLs and boy it is insanely hard. URLs don't have the semantics of an "open file" so it is impossible to make these work. Simply downloading them locally and then running really doesn't cut it.

I realize there are enormous challenges to making things "just work" but what I am suggesting that to the degree that the focus is on making things work seamlessly with what is out there, there is more opportunity for us to do new things. If the work on your application vendor's part is all around just redoing infrastructure it will take all the development time just to do that and we'll have a tiny number of new features for our collective customers.

I'm pleading with you to get the religion of not doing new things that require major changes on the part of existing apps. If you can't be the file system, then don't do it halfway. If you can't make OLEDB work for Access 2000, then don't do it halfway. Halfway is just as much work as inventing a whole new thing if you're an ISV :-)

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 4:59 PM
To: Steven Sinofsky
Subject: RE: checking something

I hear you on getting FrontPage to work out of the box. I will certainly investigate what it takes to make sure FrontPage is one of the server apps that must work on top of platinum before we ship. Based on both of our descriptions it should just work but I will explicitly add it to the list of scenarios we'll actively test. I think both of us would agree that this would have some goodness for our customers but it would also lead to the inevitable question of why can't my other applications view the same metadata that FrontPage stores ? How can a customer add value on top of FrontPage ? Can MAPI clients also see this metadata ? Can Access Data Pages build forms on this same data ? All good questions for us to address in Office 10 to make these applications better than the sum of the parts.

Just so we're clear, in the area of namespace this would mean that there are 2 HTTP URL namespaces exposed from the platinum server. One would expose URL hierarchy supporting FrontPage extensions as FrontPage registers itself for a namespace and processes all HTTP requests under that namespace. As a result there was no way for us to share the FrontPage namespace and add more value to it. There would also be a second parallel namespace that would expose Outlook Web Access, Access Data Pages support, PKM scenarios and all of the OLEDB, search and metadata primitives needed to support these applications. Certainly not perfect but at least the underlying file streams in each parallel stream would be the same. Again this is clearly something that the two teams working together can make better in a future release.

Gord...

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:29 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

I'm going to sound like a broken record...

There is infinitely more leverage to be had if you just make FrontPage 2000 work out of the box. The server extensions run just fine on Unix which means all they do is some basic file i/o. On NT they do muck with the ACLs and stuff, but you should find a way to support that since presumably the security support you are doing is at least a proper subset of NTFS. But even that is pretty low tech.

It is always a tricky conversation for me to have when it gets in to "is XXX a store that manifests a namespace". FrontPage just reads and writes a bunch of files and interprets them. THat's the value it adds on top of the file system. Yes, one of those files is a list of all the files, but docfiles do that and so does every other application that works with more than one file. For example, Word does the same thing (it has over 20 different types of files it might create at a given time while editing one .doc -- files for multi-user locking, files for tempory storeage of macros, images, etc....etc.) Is that manifesting a namespace? Not really sure.

This is just like the data page work I encouraged :-)

Customers want integration with our products, but if you have to update everything at once it is a non-starter.

We can certainly think of new features in Office10 (FrontPage too) that work better, but if the stuff we have works out of the box then we have a great starting place. Otherwise I feel like we'll spend all of our effort to just run in place -- it is hard to sell "upgrade because it works" to customers or to the development team.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 12:08 PM
To: Steven Sinofsky
Subject: RE: checking something

We've yet to test FrontPage on the server against the IFS but it theory it would be possible especially if FP isn't making any NTFS specific or esoteric volume level calls; although we support enough of these to support the shell, common dialogs.... We've explicitly not tried to support Norton utilities and other low level formatting and utility programs. At a high level our goals for the IFS have been support FAT applications and NT ACLs. This was the same set of goals for NT's own DFS file system.

Where this scenario starts to break down is shared metadata across other apps and the unified namespace. One of the key goals of our release was to support the same namespace across Win32, OLEDB, HTTP and MAPI. If FP were to mainfest its own namespace it would break this goal. Where there is definitely low hanging fruit is getting FP's bots to run on top of Platinum but to alter them slightly to store the metadata back into platinum instead of look aside files. These bots could be triggered either off file system notifications or Platinum's server events and could read the documents using vanilla Win32 calls. Ideally we'd use the PLatinum server events because we could ensure synchronicity between the PUT/POST operations and the promotion of metadata; but file system notifcations are still avaialable as a fall back mechanism. We'd then get all the benefits of FP's features but also get transacted operations and metadata that follows the documents whether they are moved, copied or renamed and the metadata would also be visible to MAPI, OLEDB and Web clients.

Julie Larson is working with Jeff Teper's team on this approach and alexhop is presenting platinum to the FrontPage team this week. Hopefully this will spark some ideas on how to bring together the benefits of both programs as quickly as possible. Alex and I will also be doing the same demo for jonde's staff on next friday.

----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:16 AM
To: Gordon Mangione (Exchange)
Subject: checking something

So at the retreat yesterday there was a lot of talk about the IFS for Exchange. One thing I'm curious about is that if Exchange has an IFS and supports IIS, then it would seem that FrontPage should just work out of the box if installed on a Platinum server?

Is that really true? That would be a pretty interesting scenario if it were true.

If it isn't, what could we do to make that so?

I worry that if we have to do things twice in frontpage we're going to have to do things twice in everything. If we can have things just work in frontpage, my guess is that all the other scenarios like templates and linked workbooks will work as well.

As it stands, the current plan is that you can sort of have a frontpage virtual root that has nothing to do with the platinum server. The issue is also one of HTTP v. DAV v. SMB which just gets confusing to me. I am sort of assuming that the reality of things is far from what is being said below since there are probably assumptions that clients are updating in all the thinking. Much like the OLEDB stuff and Access--oh yeah it is Access 2000, well with some changes everywhere.

So I'm not sure what we would actually do differently. What gord is suggesting is that when we're in Exchange "mode" FrontPage stores web pages the same way, but it keeps all of the information about those pages in a completely different format—instead of the silly _vti files we invent a new way that attaches properties to those files or does whatever they think will work on platinum. That's a disaster in terms of doing things all over again. I think with the current plan it should not be a problem being able to edit a plain old HTML file without bots or server side parsing since they should be able to have the basics working (though since there is only one release of FP, it would be a strange hit/miss use of a server that runs with platinum and FPSE).

I think they try to make this very hard—I don't think the high order bit is getting the full value of the platinum store (transactions, logging, server-server replication) as much as just being able to continue to use things that work. In a sense if there is a place to do things twice it is in the server implementation.

Sort of a classic thing in terms of the API--"none of the ones out there did what we needed, so we invented an entirely new one". Everything API evolves like that.

PS: I must be doing something wrong with this whole car buying experience. I showed up with a carpoint printout and asked "would you order this for me" and "Wade" just stared at me. Sigh.
-----Original Message-----
From: Jon DeVaan
Sent: Sunday, January 24, 1999 2:49 PM
To: Steven Sinofsky
Subject: RE: checking something

Two things don't seem quite right yet.
1)   Your point about the access from every possible mechanism. It would be interesting to see what the file tree looks like in vanilla exchange, then adding in a merged set for FPSE. If we just teach FP to treat the exchange folders specially, then we have pretty good integration. Now suppose there is one folder called "forms." If FP can edit those, then we have even better integration.
   That's just one dumb example. I don't' know enough about how exchange forms work to know if it makes sense. What would be cool is if each mail message was its own form and forms weren't some different thing (you'd need them only for creating a new thing).
2)   Access to everything via MAPI. I'm still a fan of extending MAPI to get complete access so Outlook doesn't change. He seems to concede that it's an impossibility.

Jon
Created with Word and Outlook 2000
Sent by Exchange Server 5.5

-----Original Message-----
From: Steven Sinofsky
Sent: Sunday, January 24, 1999 12:16 AM
To: Jon DeVaan
Subject: FW: checking something

I think this is progress.

I don't quite get the third paragraph in the last reply though, and I worry that is where all the issues are. How can HTML, IE5, and OLEDB all be clients (a file format, a browser, and an API spec)?

From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 6:30 PM
To: Steven Sinofsky
Subject: RE: checking something

In many ways I believe we are saying exactly the same thing. I am doing everything in my power to make Platinum look and act like just the file system and exposing this functionality over Microsoft's existing SMB file system protocols. At the same time, I'm adding functionality on top of this file system to support attributes, views, column level security, events, rules, workflow, submit mail messages, links, nested search folders... and allowing apps to use OLEDB to access this new functionality. In fact apps can OLEDB queries to search the data and then call CreateFile on the results to open the file. Believe me, I get the religion that people shouldn't have to change their apps to simply work. Today I have to simultaneously support POP3, NNTP, SMTP, IMAP, MAPI, ccMail, msMail, Notes, x.400, groupwise, profs, Netscape, SMB and HTTP access to exactly the same data. In this area I think we're totally in sync.

With the local store, I can bring Platinum extended functionality down to the client and synchronize with multiple servers over various protocols and expose all of these server's data through existing interfaces - including Win32. Yes maybe I'll need to do some hacks on win9x to really pull it off but I am determined to make it painless as possible to get existing apps working against this store. I would nuts to pursue any other plan and I believe its the biggest advantage we'll have over Notes which requires people to write to the Notes api to even view their data.

Best I can tell the only area of disagreement we have is whether I should have exposed Exchange services natively over the web for HTML, IE5 and OLEDB clients or relied exclusively on FrontPage as my web remoting agent and all of the restrictions that implies. Unfortunately, the FrontPage server extensions do not include the additional functionality Platinum supports today and I needed a way to remote Platinum functionality over the wire to clients. I couldn't extend SMB and its corresponding client interfaces, FrontPage was working on another release, MAPI is barely moving forward, OLEDB was the direction I was told to charge and I knew I needed to support native web access to Exchange. Realizing that HTTP was our pure superset remoting protocol, we decided to build a local store that synchronized over this protocol - and others for back level server support - which exposed all of this data through our 3 most important interfaces: Win32, OLEDB and MAPI. The main goal was to support existing applications but still allow the additional functionality and semantics to flow over the wire. FrontPage extensions couldn't help me and there is no way for Exchange to extend the set of HTTP primitives supported on FrontPage virtual root. Unfortunately this means until we converge plans, we'll have two virtual roots exposing the same hierarchy when you run FrontPage server extensions on Platinum machine. Personally I don't think its the end of the world - but it definitely points out where we can do a better job going forward.

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 5:19 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something
Progress...

FrontPage namespace == file system namespace. That's all. The web is also the file system namespace. Now some clever folks came along and made web pages appear out of thin air (dynamically generated) and those have their own namespace, but since they came out of thin air you can't really browse them or enumerate them so really it is unfair to say they are part of a namespace. A namespace is something that can be enumerated. FrontPage is a namespace because it just uses the file system. When you're in the frontpage explorer you're just looking at the file system except that there are some special files FP knows about that are interpreted. Word is the same way--you don't have to know about all those temporary files but they are in the plain old file system namesapce and word knows how to find them and interpret them.

I'm a simple person. I understand the file system and everything on planet earth does.

So for me, for Platinum to be insanely cool it has to just be the file system. If it does that, then I totally understand how to add value to it. Let's just say the platinum file system is a normal file system in all respects (i.e. file save/open, frontpage server extensions, and most importantly when I install Platinum on my fully functional NT4/IIS/FP2000 server my normal wwwroot just keeps working but now my machine just turned into an awesome mail server too!). No new protocols, no new APIs. Things just work. File system APIs are something we understand incredibly well in our products. The semantics have been well-established for 15 years and the code to manage them in Office is baked in to the core. Changing that is mind-boggling hard.

Then what can I do on top of that is only limited by my imaginitive uses of the features that Platinum adds. Right now I know of 3 big features:

attributes -- Wow how cool that I could come along and add attributes to my files. These attributes can be secure (see next item) and by the coolness of MAPI [sic.] I can make cool views on those from Outlook 2000. Now in Outlook10 and all the Office10 applications we can make it very easy to add attributes while I'm creating messages and or documents. Heck, in the next frontpage we could even do that as well. Older frontpage just keeps working and blindly ignores all the new goo (sure some stuff

might break, but it still works). Once I have attributes, I know how clever people can be about creating views on those, and writing code to muck with the attributes and do special things when the values change and events are fired. I understand how the lame File Properties tab in Office can be made much cooler and how people can write all sorts of solutions based on tweaking those attributes or counting on them being certain values. Whole libraries exist because attributes are assigned by a third party to books that were printed 100 years ago!

security -- platinum adds item security on these things. In some sense this is catchup to NTFS, but still we don't have this really exposed in Office today or in FrontPage, and certainly not in plain old threaded discussion in Outlook. Most excellent and easy to expose assuming we don't have to rewrite the entire transport to do so [sic].

workflow -- I really have to claim complete ignorance of this area. But I can just assume that through a combination of attributes, security, and code along with events on the server neat things can happen. Well we can set up those neat things from within Office10. Again, I could have a normal frontpage 2000 web and with the new frontpage things just continue to work. But if I wanted I could add some neato workflow things and somehow fun things happen. My simple brain can imagine a dialog box in frontpage that allows me to say "hey when this document changes, send a message to x" or a dialog in excel that says "do not display this sheet until my boss comes along and changes the approve property to yes".

btw--some other things that will just continue to work if you're a plain old file system:

[remainder of page redacted]

I need to learn more to understand the last paragraph. I'm just not understanding it. We'll talk more.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 6:30 PM
To: Steven Sinofsky
Subject: RE: checking something

In many ways I believe we are saying exactly the same thing. I am doing everything in my power to make Platinum look and act like just the file system and exposing this functionality over Microsoft's existing SMB file system protocols. At the same time, I'm adding functionality on top of this file system to support attributes, views, column level security, events, rules, workflow, submit mail messages, links, nested search folders... and allowing apps to use OLEDB to access this new functionality. In fact apps can OLEDB queries to search the data and then call CreateFile on the results to open the file. Believe me, I get the religion that people shouldn't have to change their apps to simply work. Today I have to simultaneously support POP3, NNTP, SMTP, IMAP, MAPI, ccMail, msMail, Notes, x.400, groupwise, profs, Netscape, SMB and HTTP access to exactly the same data. In this area I think we're totally in sync.

With the local store, I can bring Platinum extended functionality down to the client and synchronize with multiple servers over various protocols and expose all of these server's data through existing interfaces - including Win32. Yes maybe I'll need to do some hacks on win9x to really pull it off but I am determined to make it painless as possible to get existing apps working against this store. I would nuts to pursue any other plan and I believe its the biggest advantage we'll have over Notes which requires people to write to the Notes api to even view their data.

Best I can tell the only area of disagreement we have is whether I should have exposed Exchange services natively over the web for HTML, IE5 and OLEDB clients or relied exclusively on FrontPage as my web remoting agent and all of the restrictions that implies. Unfortunately, the FrontPage server extensions do not include the additional functionality Platinum supports today and I needed a way to remote Platinum functionality over the wire to clients. I couldn't extend SMB and its corresponding client interfaces, FrontPage was working on another release, MAPI is barely moving forward, OLEDB was the direction I was told to charge and I knew I needed to support native web access to Exchange. Realizing that HTTP was our pure superset remoting protocol, we decided to build a local store that synchronized over this protocol - and others for back level server support - which exposed all of this data through our 3 most important interfaces: Win32, OLEDB and MAPI. The main goal was to support existing applications but still allow the additional functionality and semantics to flow over the wire. FrontPage extensions couldn't help me and there is no way for Exchange to extend the set of HTTP primitives supported on FrontPage virtual root. Unfortunately this means until we converge plans, we'll have two virtual roots exposing the same hierarchy when you run FrontPage server extensions on Platinum machine. Personally I don't think its the end of the world - but it definitely points out where we can do a better job going forward.

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 5:19 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

Progress...

FrontPage namespace == file system namespace. That's all. The web is also the file system namespace. Now some clever folks came along and made web pages appear out of thin air (dynamically generated) and those have their own namespace, but since they came out of thin air you can't really browse them or enumerate them so really it is unfair to say they are part of a namespace. A namespace is something that can be enumerated. FrontPage is a namespace because it just uses the file system. When you're in the frontpage explorer you're just looking at the file system except that there are some special files FP knows about that are interpreted. Word is the same way--you don't have to know about all those temporary files but they are in the plain old file system namesapce and word knows how to find them and interpret them.

I'm a simple person. I understand the file system and everything on planet earth does.

So for me, for Platinum to be insanely cool it has to just be the file system. If it does that, then I totally understand how to add value to it. Let's just say the platinum file system is a normal file system in all respects (i.e. file save/open, frontpage server extensions, and most importantly when I install Platinum on my fully functional NT4/IIS/FP2000 server my normal wwwroot just keeps working but now my machine just turned into an awesome mail server too!). No new protocols, no new APIs. Things just work. File system APIs are something we understand incredibly well in our products. The semantics have been well-established for 15 years and the code to manage them in Office is baked in to the core. Changing that is mind-boggling hard.

Then what can I do on top of that is only limited by my imaginitive uses of the features that Platinum adds. Right now I know of 3 big features:

attributes -- Wow how cool that I could come along and add attributes to my files. These attributes can be secure (see next item) and by the coolness of MAPI [sic.] I can make cool views on those from Outlook 2000. Now in Outlook10 and all the Office10 applications we can make it very easy to add attributes while I'm creating messages and or documents. Heck, in the next frontpage we could even do that as well. Older frontpage just keeps working and blindly ignores all the new goo (sure some stuff might break, but it still works). Once I have attributes, I know how clever people can be about creating views on those, and writing code to muck with the attributes and do special things when the values change and events are fired. I understand how the lame File Properties tab in Office can be made much cooler and how people can write all sorts of solutions based on tweaking those attributes or counting on them being certain values. Whole libraries exist because attributes are assigned by a third party to books that were printed 100 years ago!

security -- platinum adds item security on these things. In some sense this is catchup to NTFS, but still we don't have this really exposed in Office today or in FrontPage, and certainly not in plain old threaded discussion in Outlook. Most excellent and easy to expose assuming we don't have to rewrite the entire transport to do so [sic].

workflow -- I really have to claim complete ignorance of this area. But I can just assume that through a combination of attributes, security, and code along with events on the server neat things can happen. Well we can set up those neat things from within Office10. Again, I could have a normal frontpage 2000 web and with the new frontpage things just continue to work. But if I wanted I could add some neato workflow things and somehow fun things happen. My simple brain can imagine a dialog box in frontpage that allows me to say "hey when this document changes, send a message to x" or a dialog in excel that says "do not display this sheet until my boss comes along and changes the approve property to yes".

btw--some other things that will just continue to work if you're a plain old file system:

*   embeddings in Office application (sound, video, etc. in powerpoint)
*   linked workbooks -- excel power users have workbooks connected to each other and these are done through plain old file system semantics
*   document templates -- word's templates are used by tons of ISVs and all they are are just other word documents we get to via a UNC

We have investigated getting to these things via URLs and boy it is insanely hard. URLs don't have the semantics of an "open file" so it is impossible to make these work. Simply downloading them locally and then running really doesn't cut it.

I realize there are enormous challenges to making things "just work" but what I am suggesting that to the degree that the focus is on making things work seamlessly with what is out there, there is more opportunity for us to do new things. If the work on your application vendor's part is all around just redoing infrastructure it will take all the development time just to do that and we'll have a tiny number of new features for our collective customers.

I'm pleading with you to get the religion of not doing new things that require major changes on the part of existing apps. If you can't be the file system, then don't do it halfway. If you can't make OLEDB work for Access 2000, then don't do it halfway. Halfway is just as much work as inventing a whole new thing if you're an ISV :-)

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 4:59 PM
To: Steven Sinofsky
Subject: RE: checking something

I hear you on getting FrontPage to work out of the box. I will certainly investigate what it takes to make sure FrontPage is one of the server apps that must work on top of platinum before we ship. Based on both of our descriptions it should just work but I will explicitly add it to the list of scenarios we'll actively test. I think both of us would agree that this would have some goodness for our customers but it would also lead to the inevitable question of why can't my other applications view the same metadata that FrontPage stores ? How can a customer add value on top of FrontPage ? Can MAPI clients also see this metadata ? Can Access Data Pages build forms on this same data ? All good questions for us to address in Office 10 to make these applications

better than the sum of the parts.

Just so we're clear, in the area of namespace this would mean that there are 2 HTTP URL namespaces exposed from the platinum server. One would expose URL hierarchy supporting FrontPage extensions as FrontPage registers itself for a namespace and processes all HTTP requests under that namespace. As a result there was no way for us to share the FrontPage namespace and add more value to it. There would also be a second parallel namespace that would expose Outlook Web Access, Access Data Pages support, PKM scenarios and all of the OLEDB, search and metadata primitives needed to support these applications. Certainly not perfect but at least the underlying file streams in each parallel stream would be the same. Again this is clearly something that the two teams working together can make better in a future release.

Gord...

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:29 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

I'm going to sound like a broken record...

There is infinitely more leverage to be had if you just make FrontPage 2000 work out of the box. The server extensions run just fine on Unix which means all they do is some basic file i/o. On NT they do muck with the ACLs and stuff, but you should find a way to support that since presumably the security support you are doing is at least a proper subset of NTFS. But even that is pretty low tech.

It is always a tricky conversation for me to have when it gets in to "is XXX a store that manifests a namespace". FrontPage just reads and writes a bunch of files and interprets them. THat's the value it adds on top of the file system. Yes, one of those files is a list of all the files, but docfiles do that and so does every other application that works with more than one file. For example, Word does the same thing (it has over 20 different types of files it might create at a given time while editing one .doc -- files for multi-user locking, files for tempory storeage of macros, images, etc....etc.) Is that manifesting a namespace? Not really sure.

This is just like the data page work I encouraged :-)

Customers want integration with our products, but if you have to update everything at once it is a non-starter.

We can certainly think of new features in Office10 (FrontPage too) that work better, but if the stuff we have works out of the box then we have a great starting place. Otherwise I feel like we'll spend all of our effort to just run in place -- it is hard to sell "upgrade because it works" to customers or to the development team.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 12:08 PM
To: Steven Sinofsky
Subject: RE: checking something

We've yet to test FrontPage on the server against the IFS but it theory it would be possible especially if FP isn't making any NTFS specific or esoteric volume level calls; although we support enough of these to support the shell, common dialogs.... We've explicitly not tried to support Norton utilities and other low level formatting and utility programs. At a high level our goals for the IFS have been support FAT applications and NT ACLs. This was the same set of goals for NT's own DFS file system.

Where this scenario starts to break down is shared metadata across other apps and the unified namespace. One of the key goals of our release was to support the same namespace across Win32, OLEDB, HTTP and MAPI. If FP were to mainfest its own namespace it would break this goal. Where there is definitely low hanging fruit is getting FP's bots to run on top of Platinum but to alter them slightly to store the metadata back into platinum instead of look aside files. These bots could be triggered either off file system notifications or Platinum's server events and could read the documents using vanilla Win32 calls. Ideally we'd use the PLatinum server events because we could ensure synchronicity between the PUT/POST operations and the promotion of metadata; but file system notifcations are still avaialable as a fall back mechanism, We'd then get all the benefits of FP's features but also get transacted operations and metadata that follows the documents whether they are moved, copied or renamed and the metadata would also be visible to MAPI, OLEDB and Web clients.

Julie Larson is working with Jeff Teper's team on this approach and alexhop is presenting platinum to the FrontPage team this week. Hopefully this will spark some ideas on how to bring together the benefits of both programs as quickly as possible. Alex

and I will also be doing the same demo for jonde's staff on next friday.

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:16 AM
To: Gordon Mangione (Exchange)
Subject: checking something

So at the retreat yesterday there was a lot of talk about the IFS for Exchange. One thing I'm curious about is that if Exchange has an IFS and supports IIS, then it would seem that FrontPage should just work out of the box if installed on a Platinum server?

Is that really true? That would be a pretty interesting scenario if it were true.

If it isn't, what could we do to make that so?

Progress...

FrontPage namespace == file system namespace. That's all. The web is also the file system namespace. Now some clever folks came along and made web pages appear out of thin air (dynamically generated) and those have their own namespace, but since they came out of thin air you can't really browse them or enumerate them so really it is unfair to say they are part of a namespace. A namespace is something that can be enumerated. FrontPage is a namespace because it just uses the file system. When you're in the frontpage explorer you're just looking at the file system except that there are some special files FP knows about that are interpreted. Word is the same way--you don't have to know about all those temporary files but they are in the plain old file system namesapce and word knows how to find them and interpret them.

I'm a simple person. I understand the file system and everything on planet earth does.

So for me, for Platinum to be insanely cool it has to just be the file system. If it does that, then I totally understand how to add value to it. Let's just say the platinum file system is a normal file system in all respects (i.e. file save/open, frontpage server extensions, and most importantly when I install Platinum on my fully functional NT4/IIS/FP2000 server my normal wwwroot just keeps working but now my machine just turned into an awesome mail server too!). No new protocols, no new APIs. Things just work. File system APIs are something we understand incredibly well in our products. The semantics have been well-established for 15 years and the code to manage them in Office is baked in to the core. Changing that is mind-boggling hard.

Then what can I do on top of that is only limited by my imaginitive uses of the features that Platinum adds. Right now I know of 3 big features:

attributes -- Wow how cool that I could come along and add attributes to my files. These attributes can be secure (see next item) and by the coolness of MAPI [sic.] I can make cool views on those from Outlook 2000. Now in Outlook10 and all the Office10 applications we can make it very easy to add attributes while I'm creating messages and or documents. Heck, in the next frontpage we could even do that as well. Older frontpage just keeps working and blindly ignores all the new goo (sure some stuff might break, but it still works). Once I have attributes, I know how clever people can be about creating views on those, and writing code to muck with the attributes and do special things when the values change and events are fired. I understand how the lame File Properties tab in Office can be made much cooler and how people can write all sorts of solutions based on tweaking those attributes or counting on them being certain values. Whole libraries exist because attributes are assigned by a third party to books that were printed 100 years ago!

security -- platinum adds item security on these things. In some sense this is catchup to NTFS, but still we don't have this really exposed in Office today or in FrontPage, and certainly not in plain old threaded discussion in Outlook. Most excellent and easy to expose assuming we don't have to rewrite the entire transport to do so [sic].

workflow -- I really have to claim complete ignorance of this area. But I can just assume that through a combination of attributes, security, and code along with events on the server neat things can happen. Well we can set up those neat things from within Office10. Again, I could have a normal frontpage 2000 web and with the new frontpage things just continue to work. But if I wanted I could add some neato workflow things and somehow fun things happen. My simple brain can imagine a dialog box in frontpage that allows me to say "hey when this document changes, send a message to x" or a dialog in excel that says "do not display this sheet until my boss comes along and changes the approve property to yes".

btw--some other things that will just continue to work if you're a plain old file system:

*   embeddings in Office application (sound, video, etc. in powerpoint)
*   linked workbooks -- excel power users have workbooks connected to each other and these are done through plain old file system semantics
*   document templates -- word's templates are used by tons of ISVs and all they are are just other word documents we get to via a UNC

We have investigated getting to these things via URLs and boy it is insanely hard. URLs don't have the semantics of an "open file" so it is impossible to make these work. Simply downloading them locally and then running really doesn't cut it.

I realize there are enormous challenges to making things "just work" but what I am suggesting that to the degree that the focus is on making things work seamlessly with what is out there, there is more opportunity for us to do new things. If the work on your application vendor's part is all around just redoing infrastructure it will take all the development time just to do that and we'll have a

tiny number of new features for our collective customers.

I'm pleading with you to get the religion of not doing new things that require major changes on the part of existing apps. If you can't be the file system, then don't do it halfway. If you can't make OLEDB work for Access 2000, then don't do it halfway. Halfway is just as much work as inventing a whole new thing if you're an ISV :-)

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 4:59 PM
To: Steven Sinofsky
Subject: RE: checking something

I hear you on getting FrontPage to work out of the box. I will certainly investigate what it takes to make sure FrontPage is one of the server apps that must work on top of platinum before we ship. Based on both of our descriptions it should just work but I will explicitly add it to the list of scenarios we'll actively test. I think both of us would agree that this would have some goodness for our customers but it would also lead to the inevitable question of why can't my other applications view the same metadata that FrontPage stores ? How can a customer add value on top of FrontPage ? Can MAPI clients also see this metadata ? Can Access Data Pages build forms on this same data ? All good questions for us to address in Office 10 to make these applications better than the sum of the parts.

Just so we're clear, in the area of namespace this would mean that there are 2 HTTP URL namespaces exposed from the platinum server. One would expose URL hierarchy supporting FrontPage extensions as FrontPage registers itself for a namespace and processes all HTTP requests under that namespace. As a result there was no way for us to share the FrontPage namespace and add more value to it. There would also be a second parallel namespace that would expose Outlook Web Access, Access Data Pages support, PKM scenarios and all of the OLEDB, search and metadata primitives needed to support these applications. Certainly not perfect but at least the underlying file streams in each parallel stream would be the same. Again this is clearly something that the two teams working together can make better in a future release.

Gord...

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:29 PM
To: Gordon Mangione (Exchange)
Subject: RE: checking something

I'm going to sound like a broken record...

There is infinitely more leverage to be had if you just make FrontPage 2000 work out of the box. The server extensions run just fine on Unix which means all they do is some basic file i/o. On NT they do muck with the ACLs and stuff, but you should find a way to support that since presumably the security support you are doing is at least a proper subset of NTFS. But even that is pretty low tech.

It is always a tricky conversation for me to have when it gets in to "is XXX a store that manifests a namespace". FrontPage just reads and writes a bunch of files and interprets them. THat's the value it adds on top of the file system. Yes, one of those files is a list of all the files, but docfiles do that and so does every other application that works with more than one file. For example, Word does the same thing (it has over 20 different types of files it might create at a given time while editing one .doc -- files for multi-user locking, files for tempory storeage of macros, images, etc....etc.) Is that manifesting a namespace? Not really sure.

This is just like the data page work I encouraged :-)

Customers want integration with our products, but if you have to update everything at once it is a non-starter.

We can certainly think of new features in Office10 (FrontPage too) that work better, but if the stuff we have works out of the box then we have a great starting place. Otherwise I feel like we'll spend all of our effort to just run in place -- it is hard to sell "upgrade because it works" to customers or to the development team.

-----Original Message-----
From: Gordon Mangione (Exchange)
Sent: Saturday, January 23, 1999 12:08 PM
To: Steven Sinofsky
Subject: RE: checking something

We've yet to test FrontPage on the server against the IFS but it theory it would be possible especially if FP isn't making any NTFS specific or esoteric volume level calls; although we support enough of these to support the shell, common dialogs.... We've explicitly not tried to support Norton utilities and other low level formatting and utility programs. At a high level our goals for the IFS have been support FAT applications and NT ACLs. This was the same set of goals for NT's own DFS file system.

Where this scenario starts to break down is shared metadata across other apps and the unified namespace. One of the key goals of our release was to support the same namespace across Win32, OLEDB, HTTP and MAPI. If FP were to mainfest its own namespace it would break this goal. Where there is definitely low hanging fruit is getting FP's bots to run on top of Platinum but to alter them slightly to store the metadata back into platinum instead of look aside files. These bots could be triggered either off file system notifications or Platinum's server events and could read the documents using vanilla Win32 calls. Ideally we'd use the PLatinum server events because we could ensure synchronicity between the PUT/POST operations and the promotion of metadata; but file system notifcations are still avaialable as a fall back mechanism, We'd then get all the benefits of FP's features but also get transacted operations and metadata that follows the documents whether they are moved, copied or renamed and the metadata would also be visible to MAPI, OLEDB and Web clients.

Julie Larson is working with Jeff Teper's team on this approach and alexhop is presenting platinum to the FrontPage team this week. Hopefully this will spark some ideas on how to bring together the benefits of both programs as quickly as possible. Alex and I will also be doing the same demo for jonde's staff on next friday.

-----Original Message-----
From: Steven Sinofsky
Sent: Saturday, January 23, 1999 12:16 AM
To: Gordon Mangione (Exchange)
Subject: checking something

So at the retreat yesterday there was a lot of talk about the IFS for Exchange. One thing I'm curious about is that if Exchange has an IFS and supports IIS, then it would seem that FrontPage should just work out of the box if installed on a Platinum server?

Is that really true? That would be a pretty interesting scenario if it were true.

If it isn't, what could we do to make that so?