

1 (10:20 a.m.)

2 THE COURT: Okay. Let's get the jury in.

3 (Whereupon, the jury returned to the courtroom.)

4 Q. (By Mr. Holley) Mr. Belfiore, one more question
5 about Page 11. If an ISV provided in its application the
6 ability to password protect files, could that functionality
7 be exposed in the Microsoft's common file open dialog?

8 A. Yes.

9 Q. How would you do that?

10 A. Is this on? Oh here. There we go. The same
11 mechanisms that I described earlier. So if an ISV wanted to
12 have password protection on their files, they would
13 implement that as part of their application. So when the
14 user double clicked the file, their application would run,
15 their application code would know that that file was
16 password protected, then they display a dialog box where
17 they issue a password. And they could enable that by
18 creating a shell extension, adding something onto the
19 context menu for a file.

20 So in this case if I wanted to password protect that
21 letter to Nancy, as a user I would right click, get this
22 context menu, the software developer could add an item
23 called password protect. When the user clicks it, our shell
24 will say hey, application, somebody clicked the password
25 protect menu, and then the application would run and the

1 password protect dialog would be displaced and the user
2 could do it.

3 Q. Thank you. Now, directing your attention to the
4 page internally numbered 19 which bears the control number
5 734389 entitled shell extensibility, can you tell us what
6 you were talking about in the fourth item there which is
7 entitled Explorer UI Integration?

8 A. Sure. What I am describing here is this notion,
9 as we talked about earlier, that third-party ISVs could
10 create a custom container, so imagine a folder or a
11 briefcase or an e-mail server, and that custom container
12 would display hierarchy in the tree view and then its
13 contents in the content menu.

14 Q. Now you say in the -- in the sub bullet, "if you
15 have an application that displays a collection of file-like
16 objects," what did you mean by that?

17 A. That is the same point that I was talking about
18 earlier where the kinds of applications that make sense to
19 use this facility are applications that want to display
20 collections of icons. And so here I'm saying, with
21 emphasis, if you have an application that does that, as
22 opposed to an application that does something entirely
23 differently, then you can use this facility and it would
24 make sense.

25 Q. Let's look at the very next page of the document

1 which is entitled explorer integration details. And the
2 first bullet is not for most applications, exclamation mark.

3 Why were you emphasizing the point that the
4 NameSpace extension API mechanism was not for most
5 applications?

6 A. Because I am having this -- I'm in a presentation
7 room presenting to a bunch of software developers. And this
8 is all new to them. And I'm trying to explain the
9 rationale, the reasoning behind when we think it makes sense
10 for software developers to use this mechanism which you see
11 illustrated here. So here is the example. In the red box I
12 am saying a software developer might want to create their
13 own little custom container. And then in this case your
14 custom folder has two sub folders inside of it.

15 And what this is illustrating is that inside of that
16 red box, that is, you know, data, e-mail as an example, that
17 your application understands, but our shell doesn't
18 understand. And so you have added this. And when the user,
19 see it is shown in blue, when the user clicks on that, then
20 these other red rectangles show what your application is
21 able to modify in this window. So that the user gets a
22 generally consistent experience as they would if they had
23 selected printers or control panel or my documents.

24 So in this case, the user has selected your custom
25 folder which you might imagine is the e-mail inbox. So they

1 have selected this and over here (indicating), this red
2 rectangle is showing well now your application software is
3 now being asked to run. You can draw whatever you would
4 like in this red rectangle here to illustrate the items that
5 your software knows are inside your custom folder. And in
6 this example, my picture shows the software developer using
7 one of our other lego building blocks awaiting display a
8 list of icons. But that is entirely up to the software
9 developer. If they want to, you know, write cursive text on
10 the screen here, they can do that. They can do whatever
11 they want in it to display their idea of the contents of
12 your custom folder. So that you can draw in this rectangle,
13 they can change the text that is shown in the status bar
14 down here (indicating). And the reason for that is in this
15 example the user has selected your custom folder. And as a
16 matter of consistency, generally, we would display status
17 for the thing the user clicked on.

18 So we don't know anything about what that thing is.
19 All the application has done is tell us I have a thing
20 called inbox, we don't know what is in it, but the
21 application is invited to describe to display some
22 description of it, so we give them that ability. And then
23 over here (indicating) the right hand side of the tool bar,
24 not the entire tool bar but the right hand side of the tool
25 bar, is available for the software -- the software developer

1 to add their own buttons. And the rationale here is that
2 this window over all is still a general Windows Explorer
3 that is useful to look at a hard drive or to look at the
4 contents of the computer on the network. And so that the
5 facilities for navigating around, like going up a level or,
6 you know, this case turning the tree pane on or off, they
7 need to be displayed always whether it is our code or
8 third-party software developers code. But these buttons
9 over here can change to match whatever makes sense for what
10 the software developer has written. Then up here on the
11 top, inside of the menu bar, the user clicks on a file.
12 When the software developer's custom container is selected,
13 they get to merge in some items in the file manager. And so
14 what is effectively happening is some of our software is
15 running and displaying this window and that part of the tool
16 bar and the menu bar, and it is commingled with software
17 that we're asking the third-party to run which is drawing
18 this part of the tree and this red rectangle and these
19 buttons on the tool.

20 Q. Now, I would like to flip back to demonstrative
21 Exhibit 200, if we can do that technically. How, if at all,
22 is the custom container you just described different from,
23 for example, the my files folder shown in this picture?

24 A. Sure. In this picture it is worth sort of taking
25 it quite literally. So in this case, this is a real world

1 computer, not just an illustration. So here is my computer
2 and you have a hard drive called C colon which is very
3 common, almost all computers have that. And that is -- that
4 is intended to represent the actual hard disk. And inside
5 of the hard disk, we're now displaying the contents of that
6 hard disk and that is something that Windows knows how to do
7 very well. It is a basic facility of Windows that is useful
8 to end users and useful to developers. And so in this case,
9 if the user clicked on my files, which is an actual folder
10 or directory on the actual hard drive, then our software
11 would be used to display all of the documents or files that
12 are inside of the file system folder called my files. So
13 for the case of general files, word processing documents or
14 spreadsheets or images or music files or the case of things
15 that are general files, we have already written all of the
16 software that enables the icons to be displayed and drag and
17 drop and different views and all those kinds of things.

18 Q. Are the NameSpace extension APIs necessary to add
19 a folder to the file system like my files?

20 A. No.

21 Q. If we could go back, Mr. Goldberg, to PX-113.
22 The next -- there are two sub bullets. I think we have
23 talked about the first one, but just so it is clear, why did
24 you say only should be used if your application displays a
25 pseudo-folder colon electronic mail, document management, et

1 cetera.

2 A. The entire rationale of this mechanism goes back
3 to that original thing that I said which is when a user is
4 going to look at the things like files or printers or things
5 that are like that, then it makes sense for there to be a
6 common metaphor. But when they're going to do something
7 that is quite different, like edit a document, we think it
8 does not make sense for the task of document editing to
9 happen in this window. And I wanted to be really clear when
10 I made this presentation that the purpose of the NameSpace
11 extensions was explicitly for the situation where your
12 application is storing file like things in a way that we
13 can't display, and so there is a facility for you to make
14 that consistent. And I explicitly wrote users should not
15 edit documents via an Explorer extension here because in my
16 opinion that would have been a bad user interface.

17 In practice, if you think about PCs of that era, the
18 screens were pretty small, there wasn't that much
19 resolution, and your word processing, for example, would be
20 limited to this red rectangle. You couldn't even put
21 buttons across the entire toolbar, you would get a fraction
22 of the screen, and you get a user experience that in my
23 opinion would have been confusing. So I was discouraging
24 developers from choosing to do that.

25 Q. Now, I apologize if I have asked you this

1 already, but were word processing applications and
2 spreadsheet applications and presentation graphics
3 applications among the applications that you thought were
4 good candidates for using the NameSpace extension APIs?

5 A. No.

6 THE COURT: I'm sorry, I just didn't -- this is a new
7 concept to me. If you edit a document, you couldn't click
8 on the thing at the top right hand to make it display on the
9 full screen?

10 THE WITNESS: Well, the -- in the way that you
11 described it, if you added a file to the file system, then
12 our software is the only software that is going to run. We
13 haven't -- we don't need to ask a third-party software to do
14 anything, we'll display all those icons. When a user double
15 clicks one, we know they want to launch an application. So
16 we go run the application and say hey application, please
17 launch, and this is the file that the user wants, go open
18 that one. And the NameSpace extensions are only for a
19 situation where what we have done is inadequate. And there
20 are not very many situations like that.

21 THE COURT: Go ahead. I'm sorry.

22 Q. (By Mr. Holley) Thank you, Your Honor.

23 Mr. Belfiore, directing your attention, sir, to the page of
24 this document numbered 24 entitled 10 keys for making a
25 great windows Chicago application UI, and then I guess

1 actually we should probably look at Page 23, that is the
2 first of your 10 keys. If you look at the -- the 10 which
3 are in front of you, is adding a NameSpace extension to the
4 Windows Explorer one of your keys for making great Windows
5 application?

6 A. No.

7 Q. Why not?

8 A. It wasn't one of the most important things that I
9 felt the software developer should do and it only applied in
10 very limited cases.

11 Q. Now, directing your attention to item number
12 four, use the comm DLG file open, just pausing there, what
13 does that mean, comm DLG?

14 A. The commdlg is the actual software code letters
15 that are used for the common dialog. So what I am saying
16 here is so that users get a very consistent experience for
17 how all these icons are displayed and what a right click
18 does, I'm recommending as the number four most important
19 thing, that in your application you use the common dialog
20 file open dialog. Or if you choose not to do that, that in
21 the version that you build, to please create that NameSpace
22 in a way that is accurate and consistent with the rest of
23 the shell so that users get that learning benefit as they
24 switch from one application to another.

25 Q. If an ISV chose to do the second thing which is

1 not hold the common open file dialog in Windows 95 but
2 instead write its own, how could it recreate the Windows 95
3 NameSpace accurately?

4 A. The easiest way to do that would be to use our
5 tree control lego block, which I showed earlier, to draw the
6 icons in the little lines and the plusses and stuff, and we
7 would have done that work for you, and to match that with
8 there was an API that we created called IShell folder. And
9 you could call IShell folder and it would start from the
10 desktop and then it would do the work of knowing how many
11 hard drives there were, or if there were floppy drives and
12 what the items in the network were. And you could drop the
13 tree control into your dialog box and then say I'm going to
14 fill it with the things that IShell folder says to fill it
15 with and it would be quite straightforward to implement your
16 own version of it.

17 Q. Now, using the IShell folder interface, did ISVs
18 have any flexibility about what portions of the Windows 95
19 NameSpace they wanted to display inside of their own file
20 open dialog?

21 A. Yes.

22 Q. What flexibility did they have?

23 A. They could have decided to do anything. They
24 could have called IShell folder and said well, I'm not
25 interested in showing the desktop, so I'm going down a level

1 and maybe I'll only show a hard drive. Their software code
2 would be able to see what IShell folder is suggesting is the
3 next thing to go on the list. And in realtime, their
4 software could decide to accept it, add to it, eliminate it,
5 and so on.

6 Q. What about the ability, excuse me, of ISVs to
7 supplement the Windows 95 NameSpace? Did your team provide
8 any functionality that would enable an ISV to do that?

9 A. I'm not sure I know what you mean by supplement.

10 Q. Sure. Well add their own information stores to
11 their file open dialog?

12 A. Yes, there are a number of things they could have
13 done.

14 Q. Okay. And can you explain at a high level what
15 their choices were in that regard?

16 A. They -- if you assumed that they're creating
17 their own file open dialog, they could have used our tree
18 control and filled it with anything they wanted to. They
19 would have had complete control over that. They could have
20 chosen to use our IShell folder or not. There is a wide
21 range of choices that they would have had.

22 Q. Okay. Now, continuing on with this, with the
23 Windows 95 common file open dialog, what benefits, if any,
24 did ISVs get by using that common file open dialog?

25 A. If an ISV chose to use the Windows file open

1 dialog, as an ISV I might say that the main benefit that you
2 get is it is very efficient. You just don't have to do that
3 same work. You literally write in your application. The
4 user has clicked an item in the menu, you write in your
5 application hey windows, please go handle the job of
6 displaying on the screen a dialog box to let the user, in a
7 sophisticated interesting way, choose the file they want and
8 then windows says back at the end user could have done a
9 whole bunch of things in there, they might have right
10 clicked on things, looked at viewers, made a copy of the
11 file, could have done drag and drop, at the end when the
12 user clicks okay, the windows says back to the application
13 here is the file they wanted. But the main benefit to the
14 ISV is that it just doesn't take very much work.

15 There is a second very significant benefit to the
16 ISV which is that their application will seem very good and
17 full featured and consistent. Because the experience in the
18 common dialog box is a bunch of work that we did, it will be
19 consistent from one application to another. So the user
20 when they see it will find something that is familiar, they
21 will know how to use it, and they will get a bunch of
22 benefit out of that. And it is also very feature rich. So
23 a secondary benefit to the ISV the is application will seem
24 good.

25 Q. And maybe you have touched on this to some

1 extent, but what benefit, if any, did end users get if an
2 ISV chose to use the Windows 95 common file open dialog?

3 A. Yeah, the end user will get the benefit of
4 consistency. They will see the same thing from one place to
5 another, they won't have to learn something new, they will
6 get the benefit of functionality and power. They will have
7 learned in one place that a right click works or that a drag
8 and drop works, and all of that will be available so there
9 is a lot of features in there. And then they'll get the
10 benefit of a well-tested, high performance quality piece of
11 software.

12 Q. Mr. Belfiore, are you aware of any applications
13 written for Windows 95 that did not use the Windows 95
14 common file open dialog?

15 A. Yes.

16 Q. Can you give us an example?

17 A. The Microsoft Office applications Word and Excel
18 didn't use the built in common file dialog.

19 Q. What did they use?

20 A. They created their own file open dialog using
21 some of those lego blocks that we provided.

22 Q. Did that mean that the file open dialogs in the
23 Microsoft Office for Windows 95 were visually inconsistent
24 with the user interface of the operating system?

25 A. I would say no. I think that they made a

1 decision to include many elements that were identical, but
2 then add additional elements that provided other features.

3 Q. The jury has heard about the M6 beta version of
4 Chicago which was provided to ISVs in June of 1994. At the
5 time that that first beta version of Chicago went out to
6 software developers, was there any debate within Microsoft
7 about whether the NameSpace extension APIs should be
8 published?

9 A. I don't remember if we were debating it
10 specifically at that point in time, although we did have
11 some debate inside Microsoft about how much of the shell
12 extensions to publish.

13 Q. And can you tell us what the nature of that
14 debate was?

15 A. Sure. The Chicago team or the Windows 95 team
16 was building our shell and our APIs aimed at a lower end set
17 of machines. The Cairo team was building a visually similar
18 shell aiming at higher users and a higher end set of
19 machine. And when you're designing these user experience
20 and APIs, there are tradeoffs that you make for things like
21 performance and reliability and flexibility that you might
22 make differently based on whether you're trying to aim at a
23 lower powered machine versus a higher powered machine. We
24 had debates mostly centered around that issue.

25 Q. What criticism, if any, did the Cairo team have

1 for the NameSpace extension APIs in Chicago?

2 A. I think that their biggest problem or issue with
3 the approach we took was that we made a decision in that
4 window when a NameSpace extension is displayed, that the
5 third-party software code that actually draws those icons
6 and those rectangles, is the software code that is
7 commingled in the same process as the entire Windows 95
8 shell. A process is an operating system concept like a
9 capsule that code is run in. And an operating system
10 ensures that one process can't make another process crash.
11 It is a reliability mechanism that makes your computer more
12 reliable. And because Windows 95 was aimed at inexpensive
13 PCs, we wanted it to be available to a large number of
14 people at a low cost. To keep PCs inexpensive, they had to
15 have less memory. So we had a key objective in the Win 95
16 team to ensure that Windows 95 ran very well and only four
17 megabytes of memory. And so in order to achieve that high
18 performance, we had to make our code not take up that much
19 memory. If you have your code taking up too much memory, it
20 is all getting in the way of each other, runs more slowly,
21 becomes unusual eventually. So we made a decision that in
22 that case where there is a third-party software that is
23 interacting with our software, that they be together in one
24 process. Because having only one process is a lot more
25 efficient in terms of memory than having more than one.

1 The positive of that tradeoff is that we take up less
2 memory, we get a less expensive computer that will run
3 faster. The negative of that tradeoff is that if that
4 third-party software is bad, or has bugs, if it crashes,
5 then because the entire Windows 95 shell is run in a single
6 process, then that Explorer window and the start menu and
7 taskbar and the desktop would all disappear. So the Cairo
8 team, because it was focused on higher end PCs and was more
9 oriented around reliability and work station like behavior,
10 found that shortcoming highly objectionable.

11 Q. Now, you said that if a NameSpace extension
12 malfunctions that the Windows user interface would
13 disappear. Did you literally mean that?

14 A. Pretty much. The way that the shell was
15 architected, the start menu and the task bar and the
16 desktop, and any of those Explorer windows, are all part of
17 the single process because that was efficient. If that
18 process crashed, then the process would stop running on the
19 computer and all of those elements of the user interface
20 would not be shown any more. And so the user would be
21 accustomed to seeing their start menu and task bar, and if
22 this third-party code which was commingled with our code
23 caused a crash, then all of those elements would vanish from
24 the screen.

25 Q. What did the user have to do at that point once

1 the user interface had vanished from the screen?

2 A. Without us adding a different facility to handle
3 that case, which we eventually did, I can't remember which
4 version we did that in, a user would have to press
5 control-alt-delete and get the task manager to automatically
6 run if you press control-alt-delete, there is a file menu
7 with a run command that you can select and then type
8 Explorer dot exe and that will relaunch our shell process
9 and those items would reappear.

10 THE COURT: The more accurate answer is curse and call
11 the IT department.

12 THE WITNESS: Or restart your computer. Shut down and
13 restart your computer.

14 Q. (By Mr. Holley) Why did the Cairo team care what
15 the Chicago team was doing in terms of this tradeoff that
16 you described?

17 MR. JOHNSON: Objection to what the -- why Cairo
18 cared.

19 THE COURT: Rephrase the question.

20 Q. (By Mr. Holley) What was your understanding of
21 the reasons why the Cairo team was concerned about the
22 reliability issues you have just described?

23 A. Part of the intent in building these two versions
24 of the operating system was an assumption, because it is
25 highly valuable, that that software that was written to

1 Windows 95 would also run on Cairo. That it was
2 Cairo-compatible with Windows 95. And the Cairo team cared
3 what -- what our software did because the way we built it,
4 in order to have compatibility, they would have to accept
5 and run as well. If they -- if they had built their
6 software differently, then that third-party ISV software
7 would not have run on Cairo.

8 Q. Now, how did the NameSpace extension mechanism
9 that you have described relate through, you know, in
10 quantitative terms to shell extensibility and Windows 95 as
11 a whole?

12 A. I would say that the NameSpace extension was one
13 of 10 or 12 or 15 different facilities that we had created
14 for there to be shell extensibility.

15 Q. Did you regard it as the most important of those
16 shell extension mechanisms?

17 A. No.

18 Q. Where would you -- where did you rank it?

19 A. Towards the bottom.

20 Q. Why was that?

21 A. Because the other mechanisms would add value to
22 the user experience in a much higher percentage of
23 situations. For example, I described earlier the notion of
24 when you right click on a file and you go to properties, we
25 have displayed property dialog, and the third-party can add

1 a tab to that properties dialog.

2 And in practice, that has proven to be quite useful
3 for many different types of files. You know as an example
4 when MP3 files for music became popular later on, that was a
5 mechanism for third-parties to describe characteristics of
6 that music which we couldn't have anticipated in the timing
7 of Windows 95. So that is a benefit that is pretty
8 significant and might affect a lot of users a lot of the
9 time. The NameSpace extensions, as I have described on the
10 slide, I viewed as only useful for a narrow case of
11 applications that are trying to display their own hierarchy
12 of containers.

13 Q. Now, these other shell extension mechanisms that
14 you have just described, did they also run on the same
15 process as the rest of the Windows 95 shell?

16 A. Yes.

17 Q. Did those other shell extension mechanisms pose
18 more or less of a risk of crashing if an ISV wrote bad code
19 than the NameSpace extension APIs?

20 A. I would say it would be less likely to happen
21 because those things are -- they tend to be simpler, they're
22 not as involved, they're pretty quick and not complex.

23 Q. Now, I would like to show you, Mr. Belfiore, what
24 has been marked as Defendant's Exhibit 90. First of all,
25 Mr. Belfiore, looking at the second page of this document,

1 it is an e-mail which the jury has seen quite a number of
2 times from Mr. Gates to various people dated October 3rd,
3 1994, entitled shell plans IShell browser. Did you receive
4 a copy of this e-mail at the time?

5 A. Yes.

6 Q. And then directing your attention to the first
7 page, there is an e-mail from you the next day, October 4th,
8 1994, to an e-mail alias called SYSwin program management.
9 Can you tell us who, not by name, but who are the people in
10 that e-mail alias?

11 A. Yes. That alias refers to the team of people who
12 are program managers on the Windows 95 team. That was about
13 12 or 15 people.

14 Q. Now, you say in the first -- in the first
15 paragraph, "FYI - I wanted y'all to see this mail so you
16 could answer people's inevitable questions accurately." Why
17 did you say that? Why did you forward Mr. Gates' e-mail to
18 this group?

19 A. The -- at the time there was -- there was an
20 understanding within our teams that there was this debate
21 that had been happening between the earlier on the Cairo
22 team and our own Windows 95 team about whether this facility
23 would be available to ISVs or not. And this e-mail thread
24 Bill asserts a decision and I wanted my peers to know so
25 they could answer questions for the people on their team.

1 Q. Now, in the next paragraph you say basically this
2 decision. Are you referring to the decision that Mr. Gates
3 made there?

4 A. Yes.

5 Q. You say, "this decision means that we won't be
6 publishing the specific shell extension mechanisms that make
7 it possible for people to extend the NameSpace you see in
8 the Explorer. We will still be publishing all the other
9 shell extensions, an interface you can call to enumerate the
10 NameSpace for use in your commdlg, et cetera."

11 What do you mean when you say we will still be
12 publishing all of the other shell extensions?

13 A. So as I described, there were, you know, 10 or 12
14 things that you would characterize as shell extensions that
15 let third-party software developers add value throughout the
16 shell. We were still going to publish those and makes those
17 proactively available and evangelize them so ISVs could take
18 advantage of them.

19 Q. And the parenthetical there, I just want to make
20 sure I understand what you're saying, those shell extensions
21 included an interface that you can call to enumerate the
22 NameSpace for use in your common dialog. What does it mean
23 to enumerate the NameSpace?

24 A. That is in the scenario that we talked about
25 earlier where if a software developer wants to display their

1 own version of the file open dialog, they want to create it
2 themselves, they can use our tree control. And then this
3 interface, which I described I named it IShell folder, to
4 ask our shell hey, here is the desktop, what comes next? Oh
5 my computer, the shell would say. So what is underneath my
6 computer? The hard drives. It is the facility that makes
7 it straightforward for a software developer to build their
8 own version of that.

9 Q. Okay. Then in the next paragraph you say, "The
10 most visible result of this decision is that Capone and
11 Marvel will no longer be shell extensions you can browse
12 into with your explorer." Now the jury has heard this
13 before, but Capone was what?

14 A. Capone was a code name for the exchange e-mail
15 client that we were building as part of Windows 95.

16 Q. And what was Marvel the code name for?

17 A. For the MSN experience that we were building as
18 part of Windows 95.

19 Q. What does it mean to say that Capone and Marvel
20 will no longer be shell extensions you can browse into with
21 your explorer?

22 A. I would say that the most important part of that
23 phrase is the part where I say that you can browse into with
24 your Explorer. And what I -- what I was indicating is that
25 if you run the Windows Explorer window, the same one you

1 have seen with the desktop on my computer, those will no
2 longer be displayed literally in that tree. That instead of
3 them being displayed literally in that tree, there will be
4 an icon that you click and it will open in a separate
5 window.

6 Q. How, if at all, did opening in a separate window
7 affect the design goal, the original design goal, of the
8 NameSpace extension APIs?

9 A. I am not sure I -- I'm not sure I know what you
10 mean by the design goal of the NameSpace APIs?

11 Q. Was the original intention that when you clicked
12 on a custom container that a separate window would open?

13 A. No, that was not the original intention.

14 Q. So opening Capone and Marvel as separate windows,
15 how, if at all, did that relate to the original idea behind
16 the NameSpace extension mechanisms?

17 A. The reason we created the NameSpace extension was
18 to solve for the situations where we wanted the user to be
19 able to think about all of these things stored together in
20 one tree. And at the time we had an opinion that your
21 e-mail or MSN made sense to be in that same tree. And this
22 decision meant that they couldn't be displayed in that same
23 tree. So I would have said well, we are not achieving that
24 part of our goal as a result of that decision.

25 Q. What was your understanding of the reason why

1 Capone and Marvel were changed to open in separate windows?

2 A. In this debate between our team and the Cairo
3 team, because the primary thing that we valued and thought
4 was critically important, the primary thing that we valued
5 was if as an end user you get -- you pay money to buy an
6 application that you run on Windows 95, and then later on
7 you get a new PC that has Cairo on it, or you buy a Windows
8 box that upgrades your Windows 95 PC to Cairo, you would
9 have an expectation that the application you paid money for
10 would still work. This is a huge thing for us, this notion
11 of compatibility.

12 And the Cairo team felt that the architecture of
13 having a third-party application that can bring down the
14 entire shell was unacceptable for their goals of
15 reliability. And in this case, Bill was making a decision
16 to say compatibility and reliability in the Cairo product
17 are more important than the incremental user benefit of
18 seeing that tree within the exact same tree as my computer
19 and the network.

20 Q. Now looking at the fourth paragraph of your
21 e-mail it reads, "as mentioned below." Is that referring to
22 Mr. Gates e-mail when you say as mentioned below?

23 A. Yes.

24 Q. "The main reason these aren't going to be
25 published is because it will add work to the Ren group,

1 whose goal will be to ship a replacement explorer in office
2 96." What did you mean by that?

3 A. This e-mail happened at a point in time after the
4 Cairo shell people had been merged into the Ren team. The
5 Ren team -- the project that was Ren eventually became
6 Microsoft Outlook, although it changed a lot before that
7 happened. And what I'm saying here is that the Ren group
8 has an intention to still ship the notion of that Cairo
9 shell, except instead of doing it in an operating system at
10 least they would ship an explorer that lets you view your
11 mail and your calendar and your documents as part of Office
12 96.

13 And because there is a fundamental premise that I am
14 assuming that the people on my team, and this team would
15 have taken as a basic assumption, that if a software
16 developer writes one of these extensions, it needs to keep
17 working in the future. That being able to do that,
18 simultaneous with having Cairo liability, would have been a
19 lot of work for the Ren group. And Bill has written in his
20 e-mail that he is choosing one benefit for the other.

21 Q. Did the office team or the Ren group in the
22 office team create a replacement for the Windows Explorer at
23 any time to your knowledge?

24 A. No.

25 Q. Did the versions of Microsoft Office that were

1 released in 1995 and 1997 use the NameSpace extension APIs?

2 A. No.

3 Q. How do you know that?

4 A. I installed them and used them and looked. You
5 could see. If you used them, there was a NameSpace
6 extension you would see it in the tree.

7 Q. Well, just -- I just want to being clear about
8 this. If office, Microsoft Office in the version released
9 in 1995 or 1997 used the NameSpace extension APIs, where
10 would you see it?

11 A. Well, the whole point of the NameSpace extension
12 APIs is that they add new stuff into the tree that a user
13 can click on and expand and see the contents of it. And I
14 didn't see that. I knew they didn't do that.

15 Q. To your knowledge, did the Capone exchange inbox
16 use the NameSpace extension APIs in the final version of
17 Windows 95?

18 A. It did not.

19 Q. Are you familiar with an e-mail client code name
20 Athena that was developed for subsequent versions of Windows
21 95?

22 A. Yes.

23 Q. Did it use the NameSpace extension APIs?

24 A. No, it didn't.

25 Q. How do you know that?

1 A. I had familiarity with that project as it was
2 being done, and the people who were working on it were in a
3 group that was close to mine. And the Athena project was
4 trying to be competitive with NetScape Communicator as part
5 of an internet sweep. And the Athena team, as part of that
6 competition with NetScape, viewed performance being really
7 fast as an incredibly critical thing because NetScapes
8 communicator software was very fast and they made a decision
9 that the performance was more important than trying to merge
10 into the NameSpace and so they didn't use the NameSpace
11 extensions.

12 Q. Now, directing your attention to the period of
13 time October 1994 which is when you sent the e-mail to your
14 team about Mr. Gates' decision to withdraw support for the
15 NameSpace extension APIs, did you ever hear anyone say that
16 a rationale for that decision, even a partial rationale for
17 that decision, was a desire to hurt Novell's office
18 productivity application?

19 MR. JOHNSON: Objection, leading.

20 THE COURT: Overruled.

21 THE WITNESS: No, I never heard that.

22 Q. (By Mr. Holley) Did you ever hear anyone say in
23 1994, 1995, or 1996 that Microsoft's decision to withdraw
24 support for the NameSpace extension APIs was intended to
25 delay the release of WordPerfect, Quattro Pro or

1 PerfectOffice for Windows 95?

2 THE COURT: Again, you're breaking the rules. Say
3 what if anything did you hear?

4 Q. (By Mr. Holley) What if anything, Mr. Belfiore,
5 did you hear in 1994, 1995 or 1996 about whether the
6 motivation to withdraw support for the NameSpace extension
7 APIs related to those Novell office productivity
8 applications?

9 A. Nothing.

10 Q. Now, you talked earlier about this debate between
11 the Chicago team and the Cairo team about whether the
12 NameSpace extension APIs should be supported in future
13 versions of Windows. Did you yourself communicate the
14 existence of that debate to independent software vendors?

15 A. No, not that I recall doing.

16 Q. Why not?

17 A. It would have been a weird thing to do. When we
18 talked to software developers, we would give them
19 explanation about what is coming in the products and how
20 they work and there would have been no reason to describe
21 that to a software developer.

22 Q. Well, don't you think it would have been
23 important to software developers to know that the NameSpace
24 extension APIs might change in the future?

25 A. I think as a general matter when I communicated

1 with software developers I gave them information about the
2 fact that these APIs might change as we go through the
3 development process. And the notion that one might change
4 as a result of a debate inside of the company, the fact that
5 it is as a result of a debate is kind of immaterial, in my
6 opinion.

7 Q. Now, after Mr. Gates made his decision, excuse
8 me, in October of 1994 to withdraw support for the NameSpace
9 extension APIs, did you yourself receive any complaints from
10 independent software vendors?

11 A. No.

12 Q. As the lead program manager in charge of the
13 Windows 95 user interface, did you believe that Mr. Gates'
14 decision had an adverse effect on the operating system?

15 MR. JOHNSON: Objection, leading.

16 THE COURT: Rephrase it.

17 Q. (By Mr. Holley) What effect, if any, did you as
18 the lead program manager for the Windows 95 user interface
19 believe that Mr. Gates' decision to withdraw support for the
20 NameSpace extension APIs had on the operating system?

21 A. I would say as a person who had worked on the
22 shell and the design and that stuff, I was disappointed and
23 perhaps annoyed at the decision. But I didn't conclude that
24 it was going to have a material effect on the operating
25 system.

1 Q. Why not?

2 A. It is a relatively minor capability in the grand
3 scheme of trying to build an operating system that delivers
4 a lot of very significant benefit to end users and ISVs.

5 Q. Now, I would like to have you look, Mr. Belfiore,
6 if you would, at what has been marked as Defendant's
7 Exhibit 131-A.

8 Now Mr. Belfiore, directing your attention, there
9 are internal numbers on this document, but the page that
10 reads Page 95 of 98, could you identify what the document,
11 what this e-mail message is that begins toward the bottom of
12 that page?

13 A. Sure. So this guy, Andrew Schulman, who was like
14 a writer and influential posting on internet forums at the
15 time, sent me this e-mail asking me if I could tell him
16 where he can find documentation on how to create these
17 custom container NameSpace extensions.

18 Q. Okay. And it has been highlighted up on the
19 screen, but this occurred when?

20 A. Oh, yeah, he sent me the mail on March 13, 1996.

21 Q. Okay. And is the e-mail that begins on the first
22 page of the Defendant's Exhibit 131-A your response to
23 Mr. Schulman's inquiry?

24 A. Yes.

25 Q. Now you say in the second paragraph we have a

1 preliminary doc that we have been giving to people who ask
2 for it, and we're getting this cleaned up and putting
3 together sample code to post on the web and ship in the
4 April MSDN. The jury has heard this before, but just so
5 we're clear, what does MSDN stand for?

6 A. Microsoft developer network.

7 Q. And what did you mean when you said we have a
8 preliminary doc that we have been giving to people who ask
9 for it?

10 A. We had created a document which wasn't final and
11 polished and cleaned up and sort of formal that explained
12 how, in this case people are software developers, could add
13 these NameSpace extensions if they were interested and asked
14 us and we had a document that we had been giving to people.

15 Q. And in the next paragraph you say to Mr. Schulman
16 here's the doc itself. Is that a reference to the
17 preliminary documentation that you referred to in the first
18 paragraph?

19 A. Yes.

20 Q. Here is the doc itself. I'll pass along the
21 samples next week when they're ready. What did you mean by
22 samples?

23 A. Samples would be software that we wrote as if we
24 were a third-party ISV illustrating exactly how the
25 NameSpace extension would work. The sample is a common

1 thing and it is -- you create a sample to give to a software
2 developer so that they either use it directly or copy and
3 paste it into their app or modify as they see fit.

4 Q. Now, you have a PS here. It says, "the letter of
5 explanation below has been going out with the doc." And
6 just so we're clear, the reference to the doc there, is that
7 to the same preliminary documentation that you referred to
8 above?

9 A. Yes.

10 Q. This gives the background as to why these have
11 been B-list in the past, we want to reduce the possibility
12 of shell extensions getting written that run in process and
13 take down the shell and other shell extensions. Now, what
14 does it mean, just initially, to talk about something that
15 has been B-list in the past?

16 A. The idea of a B-list API, when we create APIs for
17 software developers and we publish them and evangelize them,
18 we're out saying to the software developers please use
19 these, these will make your application better, this will
20 make the system and the applications all better experience
21 for the user. There is an implicit promise that we will
22 support those APIs from one version to another. So if you
23 write your application to these APIs, then when we create a
24 new version of Windows, the same APIs will exist and will
25 work in the same way, thus your application will run on a

1 new version of Windows without needing to be changed and the
2 end user gets the benefit. They bought the application for
3 \$100, they get a new version of Windows, the application
4 will still run. That is a promise of future compatibility.

5 The B-list is saying this is an API that we're not
6 making a promise of future compatibility. You can ask for
7 this documentation, but we're not proactively out saying
8 please use this. And so in particular cases where a
9 software developer thinks it is in the best interest of
10 their customers or their company, they can ask for it and
11 use it, but it is with an understanding that we are not
12 making a commitment to support any future versions.

13 Q. Were the NameSpace extension APIs on the B-list
14 after Mr. Gates decided to withdraw support for them in
15 October of 1994?

16 A. Yes.

17 Q. Did anyone from Novell WordPerfect, to your
18 knowledge, ever ask Microsoft for the preliminary B-list
19 documentation being provided to ISVs?

20 A. Not --

21 MR. JOHNSON: Your Honor, I don't understand how he
22 can speak for all of Microsoft.

23 THE COURT: He can't. He can only speak to his own
24 understanding which I think is the question.

25 Q. (By Mr. Holley) That is what I intended, Your

1 Honor.

2 A. Not that I'm aware of.

3 Q. Now, you go on to say in this paragraph, "we want
4 to reduce the possibility of shell extensions getting
5 written that run in process and take down the shell and
6 other shell extensions." Is this the same reliability point
7 that you have been discussing previously?

8 A. Yes.

9 Q. Now, there is below your postscripts to your
10 e-mail there is text that begins, "dear ISV." What is that?

11 A. As I mentioned in my mail to Andrew Schulman,
12 that is the opening to the letter that was an explanation
13 for the ISV who had asked for the documentation on using
14 these NameSpace extensions.

15 Q. Okay. And it says, the letter says, "Dear ISV.
16 It has come to Microsoft's attention that a number of ISVs
17 have been trying to reverse engineer the Windows 95 shell
18 extension mechanism in order to integrate their application
19 with the Windows Explorer." What does it mean to reverse
20 engineer the shell extension mechanism?

21 A. Reverse engineering means that a software
22 developer has gotten a -- they bought a copy of Windows and
23 they're figuring out on their own how to make this happen
24 without being given documentation from us that proactively
25 describes how to do it.

1 Q. How can they do that if they have no
2 documentation from Microsoft?

3 A. Well, there is a bunch of techniques for poking
4 around in the code and figuring out what you can do with it.

5 Q. Okay. And then the next sentence you say, "due
6 to some architectural limitations of the current design,
7 Microsoft originally chose not to publish these mechanisms
8 until the design could be changed to work robustly on both
9 Windows 95 and Windows NT."

10 What does it mean when you talk about something
11 working robustly in this context?

12 A. Typically when we talk about something being
13 robust, we mean it is insulated or protected from being --
14 from crashing.

15 Q. Now, you go on to say, "with the upcoming beta
16 release of Windows NT, these limitations have been addressed
17 and the extension mechanism will be published." Did that
18 occur? Did Microsoft, after March of 1996, publish the
19 NameSpace extension APIs?

20 A. Yes.

21 Q. And then you say, "The following document
22 discusses the limitations and the solutions available to
23 ISVs both today and with the upcoming release of Windows
24 NT." And then there is a heading, "limitations with the
25 current implementation." And I think we have discussed

1 this, but that is the first paragraph that talks about an
2 application failing and bringing down the entire shell. How
3 does that relate to your earlier testimony about reliability
4 concerns with the NameSpace extension APIs?

5 A. I am basically giving the ISV the exact same
6 explanation I just gave here.

7 Q. And then in the next paragraph you say, "there is
8 a solution to this in Windows 95, and that is to allow
9 applications that are written as shell extensions to run as
10 rooted," in quotation marks, "i.e. each one runs in its own
11 process." Can you tell us what that means?

12 A. Sure. In order to accomplish the visualization
13 of one explorer window that has its tree over here, and then
14 a custom container being part of that list literally inside
15 of it, in that case, we ran it all as one process for the
16 purposes of efficiency. But when I say run as rooted, what
17 I mean is that the user will see an explorer and when they
18 see one of these icons for a custom container, if they
19 double click it, a second new window will open on top of the
20 windows explorer. And that custom container will be in the
21 upper left at the root of the -- of the namespace hierarchy.
22 And the reason why we would open a second window with the --
23 with that custom container at the root, is to put it in its
24 own distinct process.

25 So if you would imagine before that is done there is

1 one process that has the shell and all of our stuff, and
2 when the user clicks the custom container a new window is
3 opened and placed in new process so that if that code
4 crashes, only that window and that process will disappear
5 and the original one will still be there.

6 Q. Now, let's go down the page. There is a heading
7 entitled, "solution to the current limitations" and it talks
8 about rearchitecting the process so that the desktop task
9 bar process is separate from the rest of the Explorer
10 extensions that live in the shell NameSpace. Can you
11 explain to us in as nontechnical terms as possible what this
12 is saying?

13 A. If you will recall, when we were first building
14 Windows 95 we had a very significant goal to run on
15 inexpensive machines. And our target was machines with only
16 four megabytes of memory. So we put all of the shell code
17 in one process because a creating process takes a
18 significant amount of additional memory. What I am saying
19 here is that now time has elapsed, we have figured out a way
20 to balance the desire to be on these low cost machines,
21 machines are getting more memory as time is going by with
22 this -- with this issue of the entire shell disappearing.
23 So we decided to take just the part which is the task bar
24 and the desktop and put that in one process, and then
25 whenever somebody clicks my computer or windows explorer, we

1 put that in a new second process. So we have gone from one
2 to two for the basic case of clicked my computer, well now
3 there are going to be two processes for which there is one
4 for the windows that operated it. The benefit is if a shell
5 extension in one of those windows crashes, only that window
6 will disappear but the start menu and the task bar and
7 desktop will stay there. That is much better from a
8 usability perspective because your computer hadn't gone back
9 to just a blank screen. It is not that -- it is a bummer,
10 but it is not that big of a deal to go back and relaunch the
11 window.

12 Q. And is that what you mean in this last sentence
13 of that paragraph which says, "now if a shell process goes
14 down, the desktop and task bar will still be active and you
15 can relaunch the other applications easily."

16 A. Yes.

17 Q. Now, in the next section entitled, "what does
18 this mean for an ISV," the letter says an ISV still has
19 three choices. The first one is write their own application
20 using the sample code provided. The second choice is run
21 rooted. And then the third choice is integrate fully into
22 the primary explorer. And I would like to talk about those
23 three choices one at a time.

24 The first one, "write your own application," what does
25 that mean?

1 A. That means rather than take advantage of the
2 shell NameSpace extension at all, just write your own
3 application as a stand-alone. But if you want it visually
4 to look like it matches the Windows Explorer, then you can
5 use these lego building blocks, the controls that we already
6 talked about. And what I'm saying here is we are providing
7 you with an example of an application that we wrote that is
8 distinct from the Windows Explorer and uses the tree view
9 and the list view controls so that you're starting from a
10 thing that looks very much like the Windows Explorer, but
11 then you go modify it however you would like to use as your
12 own application.

13 Q. And just -- just to be clear about terminology,
14 the first sentence after write your own, it says sample code
15 has been provided with this document. Does that refer to
16 the letter itself?

17 A. Yes.

18 Q. And in the SDK, what does that mean?

19 A. The SDK is the Software Development Kit. It is
20 the documentation and samples that an ISV gets when they
21 want to write software for windows.

22 Q. And then the next full sentence says, "ISVs are
23 strongly encouraged to use this solution where ever
24 possible." Why was Microsoft giving ISVs that advice in
25 March of 1996?

1 A. Because in this case the ISV has complete
2 isolation from the rest of the shell. So a crash in one
3 place doesn't affect a crash anywhere else. It gives them
4 complete control of the user experience, but still a high
5 degree of commonality with the shell.

6 Q. Now, if we go over to the next page, excuse me,
7 the next page of the carry over of this paragraph which is
8 number 95 of 98 at the top, it says, "you also never have to
9 worry about another application bringing down your
10 application." What does that mean?

11 A. When you write your own application and it is
12 launched, it opens into a window. It is its own process.
13 And again, a process is that capsule that protects the code
14 from dying or being killed if there is a crash. Because you
15 wrote your own code and your own application and your own
16 capsule, you're safe from whatever anybody else does.

17 And you have to keep in mind that if a lot of
18 people were creating these shell extensions, and if they
19 were all running in one window in the same process, then if
20 any one of them crashed, they all crashed, they all
21 disappeared together. So what I'm referring to is if you
22 are separately in your own process, then you don't ever have
23 to deal with the -- with the problems that might be in
24 someone else's code that you don't know anything about.

25 Q. Now, let's look at the second choice of run

1 rooted. It says, "In the case that you don't want to do
2 your own, but you do not need to appear in the NameSpace
3 itself, then you should run rooted."

4 Can you explain to us, in a slightly less
5 technical jargon, what this means?

6 A. I sort of described this idea of running rooted
7 already. This is -- you still want the code to be a mix of
8 our explorer code and your code as we saw in the picture
9 with the red rectangles. But in this case, your -- you want
10 our code to be there drawing the tree and drawing the tool
11 bar and showing, you know, those parts of the user
12 experience. But you don't want to be subject to other
13 people's code crashing. So rather than your NameSpace
14 extension being shown in the midst of the list along with my
15 computer and network neighborhood, when somebody double
16 clicks you, it opens into a new window and your NameSpace
17 extension or custom container is shown with the -- your top
18 level container at the top and that is all a user will see
19 in that window.

20 Q. Now, let's move to the third of the three choices
21 which says, "run as part of the NameSpace. If your
22 application absolutely cannot run rooted, you are willing to
23 risk being taken down at any time by another application,
24 and you are willing to be extra careful testing your
25 application to make sure you are not going to take anyone

1 else down, then go ahead and run as part of the NameSpace."

2 Can you explain to us what this third choice is?

3 A. Yeah. This is the choice that is essentially the
4 picture that I showed or that you showed, we showed
5 originally, where someone has decided that they want to have
6 the whole tree with the desktop and my computer and
7 everything, but they want to appear within that list. And
8 what I'm saying here is that you have to understand that if
9 you make the choice to do that, then you're accepting the
10 risk that other people might be in there with you and you
11 might all crash together. And I am sort of imploring that
12 if you choose to do this, please be really careful because
13 our code and other third-party code is dependent on the
14 quality of your code.

15 THE COURT: And then they would get the information to
16 do that off of the B-list? I'm not sure how you do that?

17 THE WITNESS: Earlier in time, yes. Then later in
18 time it was published as part of the regular documentation.

19 THE COURT: Okay.

20 Q. (By Mr. Holley) Now Mr. Belfiore, the letter
21 goes on to say, "In this situation, it is imperative that
22 you are not dependent on the fact that you are running in
23 the same process as the rest of the shell, especially in
24 terms of data access." Why was that?

25 A. What I'm trying to explain here is that if you

1 assume that this third-party code is now running commingled
2 in that same process capsule, there are things that software
3 developers can do that are sort of short cuts where instead
4 of formally using all of the APIs in the way they're
5 formally documented they can cheat a little bit and try to
6 do things that are easier or quicker but not the formalized
7 methodology. And so there are techniques that a software
8 developer can use to look at memory and copy a thing from
9 one place to another. And those techniques will work as
10 long as the architecture doesn't change. But if there is a
11 new version of Windows where the architecture changes, then
12 those techniques will not work in the future. And so what
13 I'm saying here is that I am asking people if they choose to
14 do this third choice, then please don't cheat and make the
15 assumption in your code that you are in the same process as
16 the shell code because some day we want to make -- make it
17 so that even in this case where things are in the same
18 windows, we could have your code in a different process and
19 then everyone is protected from each other. And I want your
20 code to still work some day in the future if we do that.

21 Q. Now, if Novell had chosen to use the NameSpace
22 extension APIs in its products, would that have made it
23 easier or harder for those products to run on operating
24 systems other than Windows 95?

25 MR. JOHNSON: Objection, Your Honor, he doesn't have

1 knowledge of that.

2 MR. HOLLEY: Your Honor, this man knows all about the
3 architecture.

4 MR. JOHNSON: He is not the developer.

5 THE COURT: He can answer. Go ahead.

6 THE WITNESS: I think it would make it harder.

7 Q. (By Mr. Holley) Why?

8 A. Because the way that you would write your
9 NameSpace extension, you're going to literally type into
10 your software code a lot of places where you're going to ask
11 Windows to do things on your behalf. I did the simple
12 example of the file open dialog where an application says
13 hey windows, in a very specific and literal way, please go
14 handle this situation of the user finding a file and then
15 tell me the app what it was. And that is an intricate
16 communication between your application software and Windows
17 and in a way that is very specifically defined.

18 So if you are writing a shell extension, if you
19 even go to back to the visual picture that I drew, then
20 you're making an assumption that we have drawn the tool bar
21 on the left side of the screen. And on Windows we have
22 drawn that and you're making an assumption that we have
23 drawn the menu bar with all those menus and on Windows we
24 did draw that. But on another operating system, presumably
25 some other company didn't do that. And unless they did, you

1 would have to recreate all of that work yourself.

2 MR. HOLLEY: Your Honor, I pass the witness.

3 THE COURT: Mr. Johnson?

4 MR. JOHNSON: Thank you, Your Honor.

5 CROSS-EXAMINATION

6 BY MR. JOHNSON:

7 Q. Mr. Belfiore, good morning. We haven't met
8 before. My name is Jeff Johnson.

9 A. Hello.

10 Q. I represent Novell. First I would just like to
11 clear up, you are not a software developer, correct?

12 A. Not in my job, no.

13 Q. Okay. And Mr. Nakajima, he was the developer of
14 the NameSpace extensions, right?

15 A. Yes, one of them.

16 Q. And he was actually the inventor of the NameSpace
17 extensions, correct?

18 A. There are many people who worked on them. He
19 invented the specific software code.

20 Q. And certainly if anybody would know about the
21 uses that can be made of the NameSpace extensions, it would
22 be Mr. Nakajima, right?

23 A. He certainly is an expert, yes.

24 Q. And if anybody would know what products that
25 Microsoft used the NameSpace extensions, it would be

1 Mr. Nakajima, right?

2 A. I am not sure that he would necessarily have the
3 specialized knowledge of that, but he might.

4 Q. So as I understand it, from 1992 until through
5 August of 1995, you were the program manager that was
6 responsible for the Chicago shell and user interface, right?

7 A. Yes.

8 Q. And your responsibilities were to generally drive
9 or shepherd the design of what that operating system product
10 would look like and work like and how end users would
11 interact with it, right?

12 A. I would say primarily, but it was all aspects of
13 the shell and that included things like how OEMs might work
14 with it and how ISVs worked with it and so on.

15 Q. And as the program manager for the Chicago shell,
16 it would be your job to define the way ISV's, Windows
17 applications might look and how they might be consistent
18 with the rest of the operating system, right?

19 A. Generally, yes.

20 Q. And so you attended conferences to discuss with
21 ISVs the different ways that their applications could easily
22 use or take advantage of the capabilities in the operating
23 system, right?

24 A. Yes.

25 Q. Is it fair to say that during the Windows 95

1 development process you were very aware of what
2 extensibility was being offered to the ISVs, right?

3 A. Yes.

4 Q. And you would agree with me that in order to
5 produce a great application for Windows 95, ISVs needed to
6 provide end users with a consistent and predictable
7 experience, correct?

8 A. I would definitely say that is an important
9 benefit.

10 Q. And therefore it was understandable and expected
11 that throughout the development of Windows 95, ISVs wanted
12 very badly to be able to extend the shell in the same ways
13 that Microsoft was able to, correct?

14 A. Well, that is a very broad sweeping thing to say.
15 I would say ISVs wanted to be able to take advantage of the
16 shell or make their apps be really consistent with the shell
17 so I think that is certainly true.

18 Q. And part of the ability to extend the shell
19 included the ability to add custom folders into the
20 explorer; correct?

21 A. It could include that, yes.

22 Q. Let me show you now what has been marked as
23 Plaintiff's Exhibit 69. Now this contains some e-mails that
24 you wrote in July of 1993 to Brad Silverberg, David Cole and
25 Dennis Adler; right?

1 A. Hang on, I'm just looking at it. Let me catch up
2 with you here.

3 Q. Actually, you can see where I'm talking about
4 right on the screen in front of you, if you would like.

5 A. Sorry, can you ask that again? I was reading
6 while you were asking.

7 Q. Sure. This exhibit contains e-mails that you
8 wrote in July of 1993 to Brad Silverberg, David Cole and
9 Dennis Adler?

10 A. Yes.

11 Q. And Mr. Silverberg was in charge of Chicago;
12 right?

13 A. Yes.

14 Q. And David Cole was the group manager of Chicago;
15 right?

16 A. Yes.

17 Q. And Dennis Adler was group program manager for
18 Chicago; right?

19 A. Yes.

20 Q. And you, in fact, reported to Mr. Adler?

21 A. That is correct.

22 Q. And your e-mail is entitled first comments on
23 UIDP. Do you see that there?

24 A. I do see it, yes.

25 Q. And do you recall a Chicago user interface design

1 preview in July of 1993?

2 A. I recall that we had design previews, although I
3 don't remember one that was specifically on that date. But
4 we certainly did have one.

5 Q. And does this refresh your recollection that you
6 attended that design preview on behalf of Microsoft?

7 A. If I could just take a minute to read it, it will
8 probably help me recall what it was or what happened that
9 day.

10 Q. Maybe I can just direct your attention down to
11 about the middle of the first page where you asked all of
12 the main presenters in the break out session, the moderators
13 to send me, I guess that means you, their comments, and here
14 is what you collected so far?

15 A. I did write that, yes.

16 Q. And looking at number two there it says, "They
17 want us exposing ways to hook in. Every single opportunity
18 they saw, they wanted hooks for. You already know about
19 explorer folders and prop sheets, but they also want to add
20 their own finders to the find menu, and they want to add
21 their stuff to the help menu on the tray and to the single
22 help entry point dialog." Do you see that, sir?

23 A. I do.

24 Q. Now, so ISVs wanted -- this is very early in the
25 process back in July of 1993, they wanted Microsoft to

1 document the way that they could integrate into the shell;
2 correct?

3 A. Well, you use the phrase document, and I wouldn't
4 have used that phrase. I would have said that what was
5 happening at the time is that software developers saw the
6 experience we were building, and then they would identify
7 places in it where they thought it would be meaningful for
8 them to add additional functionality. And so what I wrote
9 is they want to expose ways, that is asking for features and
10 work on our part. What I would have said is they're asking
11 for us to create APIs for additional ways that they can hook
12 in new value or add in parts of their experiences.

13 Q. And so every -- every opportunity, every single
14 thing that Microsoft has demonstrated, ISVs wanted APIs in
15 order to do the same thing, correct?

16 A. No, I wouldn't say that. You literally said
17 every single thing we were demonstrating and that is not
18 what they were asking for. They were looking at our user
19 experience and picking out in this, you know, in this
20 example there is I think three things that they're asking
21 for. And I think that what was really going on is they saw
22 our design and they could imagine interesting things they
23 might like to do. So they gave us feedback on places that
24 they would have liked for us to add features that would
25 enable them to extend the shell.

1 Q. One of those things you wrote about here was
2 explorer folders, right?

3 A. Yes.

4 Q. And you were referring to here the custom folders
5 that could be added to the Windows Explorer; correct?

6 A. Not necessarily.

7 Q. Could very well be though, right, sir?

8 A. Well, I think what the ISVs were mostly asking
9 about were adding functionality to the views that we
10 provided for Explorer folders. So the things that they
11 mostly gave us feedback on was wanting to be able to run
12 some code to alter icons as an example. You know if you
13 have created a word processing document and you have a value
14 added piece of software that will sync it with your laptop,
15 they wanted to be able to modify the icon to show whether
16 the current version of the file was in-sync. They wanted to
17 be able to change what drag and drop would do from one
18 folder to another. So what I'm saying is I am acknowledging
19 as the people on this mail that read already know, ISVs are
20 interested in extending Explorer folders generally, and
21 extending prop sheet pages generally, and I'm naming three
22 other things that in this particular meeting they expressed
23 an interest in doing.

24 Q. Let's try it this way. In order to create a
25 custom explorer folder, an ISV would have needed the

1 NameSpace extensions, right?

2 A. Well, that is true, but that is not the same as
3 what I'm referring to specifically here.

4 Q. Let's go down to number six, please. If you look
5 at that number six there it says, in the second sentence it
6 starts, "Lotus and WP guys," I should assume that is
7 WordPerfect, right?

8 A. Yes.

9 Q. "All seem to want to push the document-centric
10 envelope. Everyone in general just wants us to tell them
11 clearly what to do." So what does document-centric mean?

12 A. This point, as it is introduced there says, it is
13 referring to the question of MDI versus SDI. My entire
14 point six is about that question. And I had talked about
15 this at some of the developer conferences that was in the
16 talk that I gave that was shown early.

17 THE COURT: I'm sorry, remind me what MDI and SDI is
18 again?

19 THE WITNESS: MDI -- the acronym means multiple
20 document interface and single document interface. And here
21 is the specific idea they refer to. In Windows 3 --

22 THE COURT: You can refresh our recollection.

23 THE WITNESS: In Windows 3, a lot of applications use
24 this mechanism called MDI, and almost none use it today. So
25 I'll describe it, if you don't remember this, it might seem

1 like a very strange thing. In MDI when you launch, let's
2 take a word processing example, you run Microsoft Word or
3 WordPerfect or any other word processor and you have this
4 window. And inside of that window let's say you start
5 editing the first document, Document 1, okay, you're editing
6 Document 1 but now you also want to edit document number
7 two. There is an interesting question in software design
8 when the user is editing document number two, should it be
9 constrained to this window that has Document 1 in it, or
10 should it be a second separate window? And today, these are
11 almost all separate second windows. And when you minimize
12 one, it goes down to the task bar and they show up
13 separately or get grouped together, but they're not living
14 inside of the same window.

15 In Windows 3.1, actually one of the problems with
16 the program manager and file manager that made them hard for
17 people to use was they were built as MDI. And what that
18 meant is that if you opened a window, it was contained
19 inside of another window. So imagine a parent window that
20 is this big and has three little windows inside of it. And
21 if you minimize the parent window, they all go away. This
22 is a sort of powerful concept but it is also really
23 confusing. Because if you have this big window and then
24 this little window, this little window inside of that big
25 window, is going to behave very differently than this

1 identical looking little window that is in a different
2 application on the front. So what would happen in usability
3 testing, if this is an MDI child window, and this is a
4 parent window of another app sitting in front, if the user
5 minimizes this window up here (indicating) which contains
6 this window, this window disappears and this window does
7 not. And that was very complicated for people to
8 understand.

9 And at the time of Windows 3.1, a lot of applications
10 took advantage of the feature in Windows that let them
11 create these MDI experiences. And as a UI designer, and a
12 person who sat in on a lot of usability tests, I knew this
13 was confusing. And so the notion of document-centricity,
14 which is what I'm talking about here, is creating one window
15 for every document. And we made the task bar, I mean it was
16 intentionally designed to make it really easy for people to
17 switch between windows. So what I'm saying here is the
18 Lotus and WordPerfect people who, you know, they created the
19 word processors and spreadsheets, they wanted to go far in
20 this document-centricity and get their documents each in
21 separate windows which really meant not using MDI. For a
22 developer that has already got a Windows 3.1 app that is
23 written to MDI, you have to do a bunch of work to change
24 that MDI into an SDI thing. So we were being sensitive to
25 the fact that software developers had already invested MDI.

1 We thought it would be better for them to move to SDI and we
2 wanted to say, in fact this is what I wrote, it is okay if
3 you use MDI in Chicago, but even if you do that, you have
4 three child windows inside of this MDI window, we wanted
5 there to be three buttons on the task bar so the user could
6 figure out where they went if they minimized.

7 And so on the question of MDI versus SDI, what I'm
8 saying is the Lotus and WordPerfect guys wanted to move
9 forward with this notion of document-centricity, and they
10 wanted us to say hey, we really think you shouldn't use MDI.
11 And eventually, I mean I think MDI has all but disappeared,
12 I don't think you can find it. Mac uses it today.

13 Q. (By Mr. Johnson) Mr. Belfiore, the next
14 sentence, "everyone in general just wants us to tell them
15 clearly what to do." The word clearly has asterisks on
16 either side of it.

17 A. Yeah.

18 Q. Was that a general custom in that time period to
19 do that with words that you wanted to particularly emphasize
20 in the document?

21 A. Yes.

22 Q. It was kind of like the bold of that day?

23 A. Kind of, yeah.

24 Q. And you would agree with me that ISVs were
25 reliant on Microsoft's cooperation and information in

1 creating a great application for Windows 95; right?

2 A. Sure, yes.

3 Q. And that ISVs needed clear direction from
4 Microsoft in order to efficiently and effectively develop a
5 great application for Windows 95; right?

6 A. Are you asking me a general question or as it
7 relates to this document?

8 Q. A general question.

9 A. Okay. Could you ask it again?

10 Q. Sure. ISVs needed clear direction from Microsoft
11 in order to efficiently and effectively develop a great
12 application for Windows 95?

13 A. As you stated it I don't think I would agree with
14 you. I think that some ISVs can figure a lot of things out
15 on their own and they don't really need Microsoft telling
16 them please do this, please do this, please do that. There
17 are some ISVs who benefit from that and would like to hear
18 clear direction from us, but there are other ISVs that are
19 deeply talented at software design and development and can
20 figure it out for themselves.

21 Q. Okay. So you would agree with me that Lotus and
22 WordPerfect were deeply talented in software design;
23 correct?

24 A. Sure.

25 Q. And here you're saying that these two deeply

1 talented software developers wanted clear direction from you
2 with respect to MDI versus SDI, right?

3 A. Yes, because they want to know when we design the
4 system with regard to the question of whether their windows
5 should have child windows inside of it, is it going to work
6 well with the taskbar, are people going to be able to figure
7 it out and will it be well supported.

8 MR. JOHNSON: Your Honor --

9 THE COURT: Do you want to break for lunch? Let's
10 break for lunch and pick up in about 20 minutes.

11 (Whereupon, the jury left the courtroom.)

12 (Recess.)

13

14

15

16

17

18

19

20

21

22

23

24

25