

Microsoft's Internet Strategy

Transcript of Paul Maritz's presentation at the Professional Developers Conference.

March 12, 1996

Paul Maritz

Microsoft Corporation

Download the Microsoft Word (.doc) format of this document (zipped, xxK).

Download the PowerPoint presentation (zipped, xxK).

Editor's Note: In a few cases, we have added in words that were apparently dropped in the transcription process. They are set off by brackets: [].

PAUL MARITZ: Good morning. Thanks a lot. Feel like Miss America coming down those stairs there. It's a great pleasure to be here. My name is Paul Maritz, and I'm responsible for all our systems software efforts at Microsoft. This is the fifth of our major professional developers conferences that we've had over the years. I [hope] that there are many of you here today who are with us that were back here, for instance, in 1992 where in this same building, we took the time with you to introduce really the next—a key step in our Windows® family evolution. It's an introduction of the 30 cubic windows with Windows NT™ family and with Windows 95. And we've used these conferences to lay out major steps along the road, and today's conference is indeed laying out a huge step where we're bringing together two major phenomenons in our industry, personal computers and communications. And we don't think this is a zero-sum game. We think, in fact, this is an incredible opportunity for us to work together and really open up the next leg of growth in terms of making computing relevant to hundreds of millions of people. As I said we're undergoing a revolution.

Our industry was built on the back of the microprocessor revolution. We're now seeing the unfolding of the communications revolution. There was an underlying technological change. In the case of the PC revolution, it was the fact that the microprocessor made CPU cycles a commodity. Computing became by historical standards very cheap, almost free. And we're seeing similar changes in the communication space in our making communications a commodity, making it possible to reach large numbers of people without regard to distance. Now, each of these revolutions need this underlying technological change, but they need a series of standards to make them relevant.

In the case of the personal computer industry, we had MS-DOS® and other standards that put down a foundation on which people could build software that made that underlying technological change relevant to many people. It allowed us to build applications like spreadsheets that made the microprocessor an indispensable part of many people's lives and made the underlying technological change relevant to them. In the communications space, we see the same phenomenon happening. We're seeing laid down the basic underlying infrastructure, the interoperability standards like TCP/IP that allow us to have interoperability between many people, and we're seeing the applications being built on top of that that make that underlying technological change attractive and relevant to large numbers of people, and the Web is the first great example of that, and I'm sure there will be other examples as we go forward.

Now, all successful revolutions rapidly turn into evolutions. In other words, there's an incentive now that the basic infrastructure has been laid out, the basic framework has been set up, to start innovating within that framework. And certainly we saw that in the PC revolution where basically we were innovating on top of the underlying Intel microprocessor architecture and in and around the original DOS system to the point where we have something on a point-to-point comparison basis is quite unrecognizable as to what we started off with 15 years ago. But there is a basic theme of evolution there as people built within that framework, replacing pieces bit by bit, extending and bringing compatibility forward.

So we've seen it go from a largely stand-alone environment to a more graphical environment. We've seen in the revolution from monolithic applications to componentized applications, the evolution of standards for how we package our software, and allow a more flexible environment. We've seen declined server computers so we have behavior on the client and on the server and interaction between those two sets of behavior. We've seen the people and 3-D computing to give a rich interactive experience. We're seeing communication tools, video conferencing, data conferencing, other forms of person-to-person communication. And what we're seeing essentially as personal computing goes forward towards these technologies, is the emergence of a distributed computing environment with rich interactive multimedia presentation of information. Now, the Internet is starting to undergo the same evolution—sets of evolutionary pressures. People are innovating within the basic framework that is being laid down.

Today the Internet is really for the most part about static pages. People bring up the static representation of an HTML page, but rapidly people are trying to figure out how to extend within that framework, how to make those pages more graphical, how to introduce behavior, both client and server side, package up behaviors, components and share those sets of software. We're seeing people hook the behavior on the client to the behavior on the server in the interest of client/server computing. We're seeing people introduce multimedia and 3-D elements on the pages. We're seeing people using the infrastructure to do conferencing and other forms of communication. In other words, many of the same pressures that are working on the PC as it evolves forward in the local area networks space are now operating on the Internet as it evolves forward from static pages towards a world of rich distributed multimedia computing. So the obvious question is are there, in fact, going to be two [worlds] out there? Are we going to reinvent everything? Are we going to have two ways of doing everything? One way to develop applications on your personal computer, another way to develop applications on the Web. Two ways of communicating. Are we going to have two forms of client/server computing, are we going to have one set of structure inside the company and another set of infrastructure for use on the public area network? And we think that that is obviously a very wasteful exercise. It's not what customers want and there's no need to do that. This is, as I said earlier, a revolution that does not have to replace the previous revolution. In fact, these two revolutions need each other, if we're to really reach the real potential of these technologies.

So we are hard at work in terms of realizing a common way of doing computing, both on the personal computer and on the Internet. We believe that both of these technologies need to evolve towards a world of rich interactive distributed computing. We need a common paradigm for doing client/server computing inside the company and outside the company. Ultimately the client of your client/server application will run outside of your organization, perhaps in a home, and you'll be reaching many, many more people with your client/server applications.

We believe that communication can become fundamentally richer if we harness the power of the personal computer with the ubiquity and the power of the Internet communications; that we can make a fundamental change in the way people communicate with each other, giving rich interactive multimedia dimension to personal communications.

So at Microsoft, we believe that we can contribute along with—by working with partners in the industry to bring about this synthesis, this active wave of the Internet that represents the best of the personal computer and the best of the Web. And we believe we need to do the synthesis in several dimensions. First of all, from the infrastructure that developed the CD, then from the respect of the user, the paradigms that we see of how they interact with communication, then from the perspective of how they lay out the infrastructure inside their companies, the intranet and then how they reach beyond their companies between companies with Web sites. And we believe that if we work together as an industry in an open way to achieve this synthesis, we will really open up the next big leg of growth for all of us, and we can really take personal computers from being irrelevant tools for tens of millions of people to have it being a relevant tool for hundreds [of] millions of people. Because communications is an essential part of almost everyone's life.

Now, we at Microsoft realize that the world has changed with the coming of the communications revolution. This is actually something that we have been thinking about for many years, and originally had thought that this may evolve in the broad interactive television space. So Microsoft, along with several other companies, made some very early investments here and kind of went in a different direction while at the same time the Internet was growing like grass under our feet.

And about two years ago we looked down and realized that the grass was, in fact, pretty deep. As a result, we have really turned our company around and we realize things have changed and we need to change with it. We realize that things are going to happen much more rapidly, that we can't work on two-to-three-year development cycles, that we have to, in this interactive distributed world, deliver technologies far more rapidly. We realize that this is a synthesis that reaches beyond just Windows; although we think that Windows will be central to this phenomenon, it's not the only platform out there, and we need to provide the synthesizing technology that brings together Windows and other platforms as well.

So as a result, we've really refocused our company over the last 18 months, culminating in a major reset of our internal organization to give us renewed focus, and we now have in our systems software area world at Microsoft three major groups, or three major divisions. A group focused on our PC desktop and server business—that's the Business Desktop and Business Systems [Division] under Jim Allchin—and very importantly now, a whole division focused on this new paradigm of doing client/server computing,

of providing the tools, of providing the rich interactive communications abilities—and that's our Internet Platform and Tools Division under Brad Silverberg.

And we also have our Consumer Platforms Division, because although it may surprise you, we actually are not religious about other form factors in the PC and we have set-top box systems, all of which will connect to the Internet. In fact Bill Gates will discuss this very briefly tomorrow morning and I think you'll be surprised to find out how far along we are in this area. Certainly a lot more real than some of the other claims being made in the industry.

So I'd like to return back to this theme of embracing and extending. How do we achieve this synthesis between the personal computer and the Internet, the Web, and the communications technologies that are evolving?

I want to look at it first from the developer's perspective and today we're unfolding an architecture to achieve this synthesis and we call this the ActiveX™ set of technologies [ActiveX™ Technologies]. It's a set of technologies for the next wave of the Internet. The active, interactive, client/server, rich communicating wave of the Internet where we move beyond static pages. And what we've laid down is a series of technologies. First of all, we're laying down basically a general-purpose, high-speed, lightweight object model that is language-independent, platform-neutral and we can use as a basic factoring technology for software. That's our COM model. That interacts with the basic operating services on the particular platform that you're working with, Win32® in the case of Windows, the Mac API in the case of Macintosh, and it works in conjunction with a new set of high-speed, highly accelerated multimedia services that we call our DirectX Services; and as you'll see later on we're working very closely with the hardware community to achieve a new level of acceleration for these technologies.

But on top of that basic framework, we lay down a way of packaging up software in forms of objects or controls, orchestrating the behavior of those objects and controls, and introducing high-level services for media. And we call that our ActiveX layer, and I'll go through that in just a second. And those are the basic technologies that we can use to build applications that exploit the best of the PC, the best of the Web in an open, cross-platform, language-neutral way; and we're committed to working through all these technologies with the industry in an open fashion, taking the best input from whoever has the input to give and synthesizing it into this architecture.

So let's look at things from a client point of view. Today, as I said, the world of the Web is largely that of static pages. Essentially, you code up a series of HTML tags, and that's used to display a formatted page which is essentially a static document. Where we want to go next is to add behavior to that page, put some code down on the page, and what we're putting forward is a technology that we call Active Controls [ActiveX™ Controls]. And we're introducing Active[X] Controls today and providing you with the information and the OLE software to start working with this technology.

The key thing about Active[X] Controls is they are a way of packaging up software objects. They're upward compatible with what we've had before in the form of OLE Controls or OCXes. They are language-independent. In other words, we don't require you to write our controls in any one particular language. You can use whatever language you like to write these software objects. They are cross-platform. They are not inherently tied to Windows in any way. We've structured the interfaces so you can factor them and make these objects lightweight and fast; and as a result, you can use these technologies, drop these controls of these software objects into HTML pages, you can drop them into existing PC applications, personal productivity applications, client/server applications, so we provide the broadest possible way to reuse this technology across both PC applications and HTML pages, and leveraging all of the support that exists today for tools in the OLE world. And we take this forward into the Internet in the form of Active[X] Controls.

The obvious next thing to do once you've got behavior onto—chunks of behavior onto your page or application is to start scripting them, to start coordinating their interaction with each other and with the page. And we've defined an open scripting architecture here which allows you to again—independent of the particular language that you've used to write your control—to interact with those controls or software objects, and you can use scripting languages such as VBScript, JavaScript or for that matter any other scripting language that comes along. It's a somewhat long stretch of the imagination to believe that there won't be other important scripting languages in the world. Telescript, whatever it is can be used in this environment to control the behavior of your controls and to coordinate the behavior of the page or application as a whole. And again this will work in the context of Web pages or Web applications as well as in the context of stand-alone or networked or client/server PC applications.

Now, this question has given the amount of discussion that's gone on in the industry is, "How does Java play into this picture?" because Java is put forward as a way of packaging up software—software into objects. I'm very happy to announce that we now do actually have a signed contract with Sun Microsystems for Java. (Applause)

PAUL MARITZ: And I must extend my thanks to Allen Baratz and other folks at Sun who have worked with us so cooperatively over the last several weeks to conclude this contract which is basically a win-win for both Microsoft and JavaSoft, and, we think, for everyone in the industry, because we really will be able to take the best of these technologies and bring them together.

And as an example of that we've already been working on our Java implementation. And what we've done within the connection of the active architecture is extended the underlying Java virtual machine or the interpretive, it interprets Java by codes, to automatically expose Java classes or applets as Active[X] Controls and vice versa. In other words, if you write a Java applet, you will be able to script that applet using any Active Control scripting language, Visual Basic®, JavaScript, Telescript, whatever it is you happen to be using. It will be exposed to the outside world automatically as an Active[X] Control. So that you can just write Java applets and they will be participating automatically in this broader, more encompassing ActiveX architecture.

Again, a great example of what you can do when you take the point of view of wanting to synthesize rather than have an either/or world.

The other key component was the extensions for multimedia. And this has built upon [the] DirectX layer that basically provides the hardware acceleration that you need if you're going to provide a really interactive, satisfying experience for the end user. The first element there they're announcing today is the Direct3D, which is a set of technologies to provide very high performance accelerated 3-D capabilities directly in the context of PC applications or Web pages. Those are exposed, again, to the world as a set of COM objects and these services are wrapped as Active[X] Controls and exposed out to the world as Active[X] Controls.

ActiveMovie is our new technology that we've laid down to provide synchronization services for multimedia. The existing synchronization services such as we have in Windows or QuickTime in the form of the Macintosh basically aren't capable of dealing, for instance, with the MPEG-2 streaming world. And we need to upgrade our technology to allow to us [to] real-stream these new interactive multimedia formats that are coming along.

So we've put forward ActiveMovie, which is basically a set of object-based synchronization services that we'll be building into the ActiveX architecture to allow you to have the synchronization services for audio, video, conferencing, et cetera, in the context of the Web, in the context of the PC. ActiveVRML is a standard that we're putting forward to really lift things up to the next level of abstraction, provide people with a high-level way of describing the interaction between objects. Ultimately opening the door to even another level of hardware acceleration than what we're seeing over the next year or two.

So at this point in time, I've described to you ActiveX from the client perspective. ActiveX controls our general software work on the server side as well so we can script behavior on the server side to make the Web truly a client/server paradigm in use in the context of an internal company environment as well as on the Web at large. And to describe that for you, I'm going to invite up one of the early proponents of object technology. And now that he's over 40 years old, I don't think he'll be insulted if I call him one of the fathers of our computer industry. So I'd like to welcome Steve Jobs. (Applause.)

STEVE JOBS: This is really weird, isn't it? (Laughter.)

STEVE JOBS: It's weird for three reasons. Reason one is we're all here together. Reason two is that Microsoft's vision of where the Web is going is exactly the same as mine and my colleagues at NeXT, which is the Web is about to evolve from static publishing to active Web sites as Microsoft calls them, or dynamic services as we call them.

And the third very strange thing, I must tell you, is that we're an independent software company, and we deal with all the people in this industry. And Microsoft is treating us, as an independent software developer, just like you might have expected Netscape to treat an independent software developer. And Netscape is treating us exactly like you might have expected Microsoft to treat us. (Laughter.)

STEVE JOBS: It's really weird. So, because of all these strange occurrences, I'm here today and I'd love to tell you what I think is reinforcing Microsoft's messages of active Web sites. We introduced this product called WebObjects about six weeks ago. And we've seen enormous success with it so far. And it's doing pretty much active Web sites in a slightly different way that is totally complementary [to] what Microsoft is proposing here today. We all know about static Web sites. This is one of my favorite ones of

course, where the Web started off with a static publishing metaphor; you make a Web page by hand using whatever tools you like, and you publish it and then you and 300,000 of your closest friends can look at it. And it's amazingly cool. And it's taken us this far. But it's not going to take us the next distance.

And a lot of emphasis goes on the browser. And we see a lot of work to expand the user interface with the browser with languages, scripting languages, with Active[X] Controls, and that's all wonderful. But the next evolution is also going to be involving the server much more. Now, what's an example of this next evolution to dynamic services? Well, what happens if the page that you want doesn't exist? Right? What happens if you, as a user, request something that doesn't exist?

And the first time we saw this was this Federal Express package-tracking Web site. You can go to Federal Express's Web site, type in your package number and then their Web site goes and looks in four mainframes, gets all the data as to where your package is; it's got all this data. What do they do? Do they give it to 10,000 gnomes to make up a Web page real fast? No. They build a Web page on the fly with computers and send that custom-built Web page just for you down the wire to your Web browser. And it comes back and it looks, you know, something like this, whatever your package is. And what's amazingly cool about this is this Web page has been made just for you, to answer your question. Something that has not been prevalent in the Web today. Now, unfortunately, it took them four or five months to build this app and we've got to be able to build these apps in a day or two. And that's what WebObjects and this new active—group of Active[X] Technologies lets you do. So what I'd like to do very quickly is show you an example of a dynamic Web site that we built in about a day to give you a flavor for what's going to be happening on the Web.

Now, one of our customers is Chrysler. So we used—we used Dodge as an example. And so this might be an example, you might log onto the Web and you might see Chrysler's Web site for Dodge. And on this thing has got all their cars they make. And I don't know if you can read this. But here's Neons all the way down to Stratuses and here's the price ranges, you know, from \$8,000, I think, to—yeah, let me grab them all—to \$90,000. And they make models of two-door cars all the way to trucks. And I'm going to display these things by model number. And I'll just say display the cars.

This'll go to a database, grab all the components, and build these Web pages on the fly, send them down to my Microsoft Explorer browser, and here I go. And here's all their cars sorted by models. So I've got Neons and Intrepids. And Ram trucks and all sorts of other things. And I can go back there and say that's really cool. But what I really want to do is sort them by price. And it'll take all that same basic data and build a new Web page for me sorted by price. And it'll come back to me, and now the Neons are the cheapest and there's a cheap truck or two and et cetera, et cetera.

Now, let's say that I want—I'm interested in Neons, so I'm just going to go take a look at the Neons, show me all the Neon cars, because I want to spend 10,000 bucks on a car. Let's go for the cheapest one here. And I'll select that one. And it will go get all the data about the Neon, the latest and greatest prices, et cetera, et cetera, and it'll build a Web page for me. So here it is. And let's see, I want to see what that car looks like in red. I click the page again. It'll go get the picture for the red car, and stick it in there, lime green, et cetera. I can say, I want an option package. Let's go ahead and pick this option package, B. It will go ahead and add this to the total so I'm up to about \$12,000 as you see. Check off the page. Again, these Web pages are all being built on the fly, because every permutation, you certainly wouldn't want to build all of these permutations manually and update them every time something changes. And I can go to leasing and say I might want to lease this thing.

And it'll go ahead and throw in the cost, again, dynamically, I can say I want to spend \$5,000 on a down payment and I want it over 60 months, calculate the amount. And it'll calculate the amounts, et cetera, et cetera. And I think that you will agree that this is probably, you know, this Web site is probably two—order of magnitude, let's say, more dynamic than we typically see on the Web today. Wouldn't you agree with that? This was built in a day. (Applause.)

STEVE JOBS: Using the class of dynamic tools that you're hearing about today from Microsoft and, of course, we used our WebObjects tools as well. We've had some fun experiences with this stuff. It works with all Web browsers. So it works with Microsoft's Internet Explorer and it works with all HTTP servers, in terms of Microsoft's Internet HTTP server, which works quite well, by the way. And in terms of scripting languages, we are announcing the day that we're introducing VBScript, along with JavaScript. And we tried a new pricing model. (Laughter.)

STEVE JOBS: We decided this stuff was so important that we've got a version of this that we are giving away so that when you get your free Internet Explorer—your Internet Information Server from Microsoft,

you can just go to [NeXT's] Web site and get free WebObjects to layer right on top of it, working with NT.

And this is the actual page, I grabbed this a few days ago. We've had this on our Web site for a little over a month. We've had over 12,000 downloads and well over half of them are for NT. So we're extremely pleased with this. If you'd like to learn more about WebObjects and how it works intimately with the Microsoft Active Technologies, we're giving a special session tonight at 7:00 o'clock in Hall B, which I believe is just the back end of this hall. And we're having a technical breakout tomorrow at 4:00 p.m. in room 250 and 262. So we'd love for you to be with us tonight if you'd like to learn more about this and thank you for allowing us to present and to support Microsoft's Active[X] Technologies. Thank you. (Applause.)

PAUL MARITZ: Thanks, Steve, it's a great example of what can happen when people start to agree on basic underlying infrastructure, agree on a common object model, and start to build tools that can really complement each other and allow us to build for satisfying applications. Now, the other—the second aspect of achieving in-use synthesis is in terms of what we can do for the end user of the Internet, of the PC. And there are several important aspects here where we think we can make major progress. The first idea is that the Web has introduced a really intuitive, simple paradigm for navigating and finding information; the page and hyperlink paradigm is a very powerful paradigm. Now, many of our customers tell us that this is—a browser is one of the first applications they're actually comfortable putting on their CEO's desk because it's so simple to learn and use and millions of people have testified to that effect.

Today it works only with information stored in one format, HTML. There are lots of other documents that are very hard to shoehorn into an HTML format, CAD drawings, spreadsheets, et cetera. But all those documents have to live in a different UI paradigm; do they have to live in a different way of navigating and dealing with links? And we know that the answer is no, they do not have to. And one of the key steps that we'll be taking with our ActiveX Internet Technologies is allowing you within the context of a common browser, to display not only information stored in HTML format, but directly integrate basically documents that store their information in any format, so you can get cooperation between the browser and the code that's actually displaying the underlying document format. And this is what happens when you componentize your browser, as you will see that we are doing.

Basically, we can now extend this common navigational paradigm to all types of information. The second key integration for the users—from the user's perspective, is to look at the fact that on the average desktop today, you have two basic tools for finding and displaying and dealing with information. You have one tool for dealing with your local and your LAN-based data and you have another tool for dealing with information coming in from the Internet or your online sources. These are two separate tools or two different UI schemes, two different looks and feels. They're two full different paradigms for navigating and finding that information. So the first step in synthesis—in the synthesis, then, is to actually get to one tool. And towards the end of my talk, I'm actually going to demonstrate this to you with live running code, so we take today on the Windows environment where we have a Windows Explorer and Internet Explorer and we bring those together into a single Explorer tool. And then the second step of synthesis is to provide the ability to have one paradigm for dealing with the information, wherever that information is coming from.

So, for instance, you will see in a few minutes how we essentially allow you [to] look at all information as a series of linked Web pages, whether it be information coming in off the Internet or information coming off your hard drive or control panel. Basically, we can provide a single integrated experience of navigating and viewing information.

The other aspect of embracing and extending and achieving the synthesis is to look at the intranet environment and the Web site environment, basically, the corporate environments for internal and external use. You look at things from an intranet perspective, the obvious requirements are, first of all, to get a common navigational and information viewing paradigm. You want, in the Internet environment, to be able to deal in the same way with information in whatever format it's stored, wherever it's stored. You don't want to have to have multiple tools, multiple paradigms for dealing with information so you want to have a common navigational and viewing platform that works with all the important stores of information in your environment. You want to work with your traditional files that are accessed by traditional file-sharing. You want to access files that are storing information in HTML or Web formats. You want to deal with your messages that are stored in your message and workgroup stores. You want to deal with information that is being synthesized out of a database. You want to deal with information that's

potentially being fetched from a mainframe. You want a common navigational paradigm that works across all these important sources of information. You want it to work with your existing infrastructure. You don't want to have to rip and replace and learn new security schemes and new ways of administering things, and you want an environment that is proven and scalable and easy to extend from where you are today. And what we are laying out with, essentially, Microsoft Windows and BackOffice is a framework that meets all those requirements.

As you'll see us demonstrate in a minute, we are integrating the navigational and viewing paradigm into a common tool for looking at information, wherever that information comes from. We're making sure that that tool can work across all the important stores of information. Your existing files and documents, your Web pages, your messages, your structured data coming out of databases, whether it be local or fetched out of a mainframe. Doing it in the [model] of an integrated world from an administration point of view, from a security point of view, building all of these things in a scalable, complete, and extensi[ble] fashion. And then when you take that basic infrastructure, you can then add to it because we've laid down the basic scalability infrastructural requirements, you can extend it very easily outside of your organizational boundaries as well. So we put in there the proxy services to allow you to reach in and out of your environment in a safe way. We put in there the merchant services that basically allow you to conduct commerce over the Internet.

And we have drill-downs over the course of the next couple of days on all of these aspects; basically, how you go through the common navigational experience, how the browser gets componentized, how you use ActiveX to build client/server applications, how you work against these back-end services for both intranet and Internet purposes.

So the whole of the next several days is going through all of these concepts in a great deal more depth to try and give you the information so that we can really achieve the solutions that customers want.

So I'm going to go through now, very rapidly, a roadmap of what we'll be doing in terms of actual products to lay the foundation for these concepts, the synthesis that is the best of the PC, the best of the Web. First of all, from an Internet perspective, we've been laying down an Internet Explorer browser line, it is a cross-platform set of technologies. We support it on the Win32 platforms. On the Macintosh and Windows 3.1 we're working on very rapid releases. We have introduced the first version of our Internet Explorer back in August, the second version came in [the] November/December time frame. And we're now providing to you today the first release, the early release, of IE 3.0 that really carries these major architectural improvements that we're making for this new world of rich interactive client/server computing. It is a componentized architecture; we've taken the browser and broken it apart into a series of services that will allow to you tap in at every level so you can extend that experience for your users, customizing it, giving it the right look and feel and behavior. Our browser will no longer be a monolithic entity that you can only come into at one level. It'll have full support for the active architecture, controls, scripting, multimedia, et cetera.

And then beyond that, after Internet Explorer 3.0, we take the next step in the synthesis of really integrating it into the Windows environment so we can get this single tool, this single, common navigational and UI paradigm for accessing information of whatever type, be it coming off your local hard disk or off your LAN or off the Internet.

From a server point of view, we have the Internet Information Server built into Windows NT Server and BackOffice today. It turns out that we've labored long and hard at the business of networking and file sharing. We laid down with Windows NT Server a great general-purpose platform for distributed computing. We did that four years ago when we introduced it to you in 1992.

Since then, we've been sweating blood, competing with the aging champion of no-holds-barred file sharing, which is Netware, and getting Windows NT Server, which is a general-purpose extensible platform, to have the same level of performance that Netware can deliver in a much more restricted architecture. So we've built into Windows NT all of the services that you need to do very high-performance networking and file sharing. And it turns out that we can just reuse all of that technology for the Internet world. It is just, after all, another way of pushing bits down the pipe and Windows NT server is extremely efficient at doing that. So we can deliver, what is without any question today, the highest performance Internet server. We're making it extensible so you can tie in client/server applications that we have the connectors right now to hook up to your databases. They can synthesize information on the fly. And we'll be extending that during the course of the remainder of the year with additional services, additional connectors; you'll be able to connect up to information not only in the database, for instance, through ODBC of SQL Server or Oracle or whatever, but also the messaging databases, which is another

store of information. So you'll be able to look at messages held in an Exchange database, for instance, using a Web browser. We will integrate this into a common Internet paradigm of accessing information. We'll be adding in content and site management, searching services, versioning control, et cetera, proxy services, and merchant services. And we'll be going through these in more detail during the correlation of this conference.

Now, one of the key themes that we've been talking about is the emergence of the Internet changing from a static world to an active world. It is changing from a world of static pages to a world of applications. What we really see unfolding here is a common way of doing client/server computing, both for the local area network and the public area network, or the Web.

So knowing that the key to client/server computing is a high-quality family of tools, just as we've made many changes in other areas of Microsoft, in the last nine months, we've also made major changes in our tools thrust. And we've taken all of the major investments that we've made over the years in our Visual Tools family, Visual C++®, Visual Basic, Access, et cetera, and applying those assets firmly and squarely to this problem of allowing you to develop high-quality interactive client/server applications for the PC and the Internet. That means working within the paradigm of standards like HTML that have evolved. So as our tools go forward, we will be using the browser basically as the run-time services for all our tools.

So our tools will allow you to generate automatically both Internet-ready applications and PC-ready applications. You won't have to decide whether you're building a client/server application for the local area network or building a client/server application for the Internet. So we'll have a common family of tools. We're integrating them in terms of common run-time services, but also into a common integrated development environment. So all of the great services that you've seen, for instance, in the Visual C++ environment will now be applied across the spectrum of our tools, Visual Basic, Access, and including the Java investments that we're making. So we have a common family of tools, sharing services that span the entire spectrum of need. And that need extends from people who don't think of themselves as developing applications, people who use office applications like Word and Excel. People who use FrontPage as a publishing tool, Internet Studio as a more sophisticated application development and publishing tool, Visual Basic; Jakarta is our code name for our Visual Java product that we'll be talking more about in more detail later on today. Visual Basic, Visual C. All of these will form a common family of Internet-ready, LAN-ready tools that help you develop these applications that exploit the best of the PC and the best of the Web.

Now, to illustrate this, I've asked John Shewchuk, who is a program manager at Microsoft in our tools area, to join me and actually show us Internet Studio, which is a great example of bringing together the best of what's on the PC and in the Web. Hi, John.

JOHN SHEWCHUK: How are you doing? Actually, we're going to start off by showing some of the kinds of things we can build in Internet Studio. And what you see on the screen is Microsoft's Internet Explorer. We've enhanced it with some new capabilities to do things like professional layout. Let's take a look. There should be sound. There we go. What we see on this page is a couple of interesting things. The first thing that you'll notice, although this looks like an image map; in fact, it's actually constructed with a set of objects that are placed on the screen.

PAUL MARITZ: This is not a bitmap that we've scanned out and laid out here.

JOHN SHEWCHUK: For example, the blue background that you see here is sitting behind some transparent text up front. We see transparent metafiles and all of these things are layered one on top of each other. This is something you can't do on the Web today. We also have things like these transparent objects being hooked together with our Active Scripting technologies. And when I click on these objects, what's happening is rather than going up to the Web site, we're actually just changing the order of these things and bringing them to the front.

PAUL MARITZ: John, clearly you can't do this in the context of an HTML page today, but this is the Microsoft Internet Explorer that's running here, this is not a proprietary client, so what's the magic? There's actually two things. The—

JOHN SHEWCHUK: There's actually two things. The first thing we're looking at are Active[X] Controls and the OC96 Specification includes like my transparency, like windowless controls and the ability to layer these objects with double buffering and things like that. But the second key thing that we're showing is CSS style sheets. So the W3C committee has released a new specification for HTML and if we take a look in here, if we actually view the source, we can see that this is just standard HTML. So what we're working with the W3C on these style sheet extensions and what we're seeing here is in fact

the next generation beyond that which is CSS layout. Let's take a look at a page, so this is an example of...

PAUL MARITZ: This is an example of where we trickle out of the concepts and techniques that have previously been tried out in our proprietary environment, but are reworking those to fit into the standard Internet HTML environment so we can develop these high-quality pages and applications.

JOHN SHEWCHUK: That's absolutely right. In fact, a lot of the things that you see here were first developed in Blackbird [Internet Studio], but now that we're working with W3C, we've taken all those cool Blackbird technologies and moved them directly into HTML. So let's switch gears now and take a look at the operating environment.

PAUL MARITZ: Good.

JOHN SHEWCHUK: What we see here is a partially constructed page that we had just seen before. And this is a very early version of Internet Studio. We're not done with Internet Studio, but we wanted to preview it today to let people see where we're going, so what I'm going to do is take a control out and drag it onto the page. One of the great things about controls is that they make an excellent run-time component. But what's perhaps far [more] interesting is that they're very good for authoring. So what we're showing here is not just any kind of authoring, but this is WYSIWYG, in-place authoring. I can't spell volcano. There we go. And not only do we have WYSIWYG offering, but for those of you familiar with DB—

PAUL MARITZ: This is taking a lot of the concepts that we developed in terms of our Visual Tools in termination of drag drop, property pages, et cetera, applying them to the problem of developing these Internet applications.

JOHN SHEWCHUK: That's correct. So we also have some things in here that are really interesting for designers. So all these elements on the page, we can take them and we can actually drag them up to the toolbox, where they become a template. So by now, if I drag this back down to the page, what we do is we pick up all those properties and we've essentially cloned that control so that we can now reuse it all over the place and here's that in-place editing again.

PAUL MARITZ: There's a lot of very sophisticated stuff that you can actually do without having to become a C++ coding expert in this environment.

JOHN SHEWCHUK: That's right. This [was] designed to be super easy, drag-and-drop HTML editing. Let's see that. And we'll go into Internet Explorer again, go back to our home page again. And here's that constructed page. And again, this is all just standard HTML. So we've got one more quick thing to show you, which is once we've taken all that stuff and put it together, it's interesting to produce—you can use these technologies to produce titles that look more like consumer titles. (Music playing.)

JOHN SHEWCHUK: So things like sounds over objects. And let's go into the Miss Megabyte area here. And again, we're using that exact placement, the Z ordering. So let's look at a quick example. We're going to use Active Scripting, the Z ordering and exact placement capabilities to do a simple behavior. And this all runs on the local client. So I'm going to go over to the light switch and click on it. And what we did was brought a large black rectangle that was hiding in the back all the way to the front. It then occluded the page as the light switch goes out. And then when I click on it, it comes back. If we look at the source for that, and we go down to the bottom here, we can see that we're just using script tags. This is W3C standard HTML. And we're saying that when the click event occurs, what we're going to do is play an audio file and then bring the blackout object to the front of the Z order.

PAUL MARITZ: Great. It's very simple.

JOHN SHEWCHUK: So always kinds of other things. And let's go to one more page. (Music playing.)

JOHN SHEWCHUK: I have to warn you, my developers have been working on this, and they say that there's some sort of new bug here. So I'm not sure exactly what we're in store for. So these are things like transparent objects. And we can change when their rendering order comes in so that you can create a navigational framework right up front. And if we go over here. (Computer:) "Yeah!" Sounds like we found the bug. And a couple more Z orders.

PAUL MARITZ: Sounds like the Sun guys got into your files last night.

JOHN SHEWCHUK: I suspect.

PAUL MARITZ: This actually looks like something we would have developed for a CD-ROM title that my kids might be using to play with at home.

JOHN SHEWCHUK: Exactly. All standard HTML, so we can now take the best of the capabilities that we have on the PC for doing CD-ROM-type titles and integrate them with those with the Internet technologies.

PAUL MARITZ: So people developing titles now won't have to think am I developing this for the PC CD world, am I developing this for the intranet; it's a common development paradigm, common set of technologies.

JOHN SHEWCHUK: Yep.

PAUL MARITZ: Great. Thanks very much, John. (Applause.)

PAUL MARITZ: I wanted to also comment today on the work that we're doing to cooperate with other members of our industry to really make sure that the underlying infrastructure of the Internet really is robust enough for us to build these interactive distributed applications and enable these rich communication scenarios. And we've become very active in the industry working through these standards committees, working in partnership with other companies, to really play our role in enhancing that infrastructure.

For instance, we announced yesterday that Microsoft and Intel, along with 100 other companies who are assisting us and signed on to contribute, are—will be implementing a set of key standards that are really needed to take the Internet to the next step in terms of real-time communications. So we've committed to do high-quality implementations in an open, standard way, and build these into our products. RTP, RSVP, 2.120, et cetera. We agreed to percentage on a common set of conferencing APIs that will get video and data conferencing interoperability between products and between platforms.

We're leading an open digital signing initiative, which basically is a way to allow people to achieve the equivalent of shrink-wrap code on the Internet. If you walk into a store today and you pick up a box that is shrink-wrapped and it says, Lotus Corporation or WordPerfect or whatever on it you have some degree of confidence, not absolute; but a high degree of confidence that code infected with viruses, it isn't going to do bad things to your system.

And we need to achieve the same level of consumer confidence on the Internet as well, where people are going to be downloading and accessing applications. So we need a way of digitally signing code that is the logical equivalent of shrink-wrapping code up on the Internet that says this code came from a reputable vendor. They warrant it that they've tested it, they warrant that they've checked it for viruses. And you can have a reasonable degree of confidence. The same that you would have in going and buying an application out of a computer store, and that is an open initiative we're working on with the industry. We're developing parental control mechanisms; we're involved, as Jobs said, in these HTML extensions to allow us to build high-quality titles that are good enough to be CD titles or PC titles and not have to make people choose between doing things one way for the Internet and another way for the PC environment. We're involved in security initiatives to get standard interoperable protocols for secure transport mechanisms and secure security mechanisms on the Internet. All of this being done, as I said, in an open, standards-compliant way.

I want to give you the road map for Windows now, having spoken about our Internet products, our tools products, and clearly, Windows is the foundation [on] which most of the computers on the Internet run. We've been building networking into Windows for many years now. [So that today,] Windows has by far and away the best built-in infrastructure for Internet computing. We have great implementations for TCP/IP, PPP dial-up, WinSoft, et cetera, and that's an investment that we started making many years ago and work on and continue to work on. And as we announced just earlier with Intel, we'll be taking those sets of services to the next level in terms of real-time built-in communication services. We're working very closely with the hardware community to have the necessary hardware acceleration without which you can't get a satisfactory interactive multimedia experience. And we're exposing all of the basic services of Windows so the people who have existing Windows applications can extend them forward, utilizing the services of our componentized browser, utilizing the services of the ActiveX Controls and [Active] Scripting to take your existing Windows applications and extend them into the Internet space. This is not an either/or zero-sum world. And we think there's a tremendous opportunity for people to take their assets and extend them forward.

Now, back in 1992, I actually put this diagram up for the first time. It was our Windows family diagram. It laid out a strategy of having three complementary members of the Windows family, Windows for the server, Windows for the Workstation, Windows for the general desktop. And layering on top of those three Windows members a common user interface and a common API. And that in fact remains our strategy. We'll be working in the—this framework for the next several years.

In terms of status, Windows 95 received a huge amount of hype from the industry. In terms of reality, it's actually been quite successful if you measure it by any objective standard of a product. And in spite of what anyone may tell you, this is by far and away the most widespread, highly-used software product in

the industry. We have over 20 million users using it today. And a conservative estimate is that by the end of this year, calendar year 1996, the installed base of Windows 95 will be over 50 million. Windows NT workstation, Windows NT Server have really lived up to the great technological promise that they had over the last year. And we've seen many major corporations now standardizing on Windows NT Workstation. We know of over 50 major accounts that will be deploying more than 10,000 copies of Windows NT as their standard workstation environment over the next 12 months. Windows NT Server is taking basically all of the market share growth in the server space. And this year, calendar year '96, we expect Windows NT Server to ship more copies on the server than all copies of Unix servers combined. So all Sun shipments, all SGI shipments, et cetera by far and away [are] becoming the dominant application server platform.

In terms of Windows releases, as I said, we're working in the context of that common Windows family framework. We have releases coming up. In the middle of the year we'll be releasing the Windows NT 4.0 which is now in beta test which puts into Windows NT the graphical user interface and restores that common API, common UI theme across our Windows family. We'll be refreshing Windows 95 for the OEM community to put new hardware on there as it comes online, and also to include Internet Explorer 3.0, which will be ready around the first of the year and will be free to the industry; you'll be able to download it and include it in all the refreshes of our product. We'll also have Nashville, which will give us this integrated explorer, and then next year, calendar year '97, we'll have major releases of Windows 95 and Windows NT which we call Cairo and Memphis, respectively. So at this time I'd like to take some time and look at how we're achieving the Windows interface and paradigm, particularly in that release I called Nashville on the previous slide. So with me I have Joe Belfiore. Those of you with us in 1992 know Joe was my demo partner there. What are you going to show us? Multiuser solitaire?

JOE BELFIORE: No, what I have is the early version of Nashville; this is early so we'll see how this goes. Hopefully won't have any bugs.

PAUL MARITZ: This is real code here.

JOE BELFIORE: [The] real thing. What I want to show is how it integrates with browsing and makes it more easy-to-use and functional for people. I'll jump in here. It looks like what people are used to, but you'll see as soon as I open my computer it looks richer and more engaging. What you'll see is what we call a Web view on hypertext view on my computer.

PAUL MARITZ: This is the standard Explorer. There aren't two Explorers here.

JOE BELFIORE: No. Now it supports this additional view called hypertext view, and I can use an older view if I wanted, but you can see it's obviously a lot nicer looking and I have all the things like my items over here and you'll notice as I mouse-over here I get a pointer.

PAUL MARITZ: You're synthesizing a Web page that shows the contents of that directory or hard disk.

JOE BELFIORE: Right. And this over here, where you see these shell items, is an Active Control; it's a pocket of a shell. So if I right-click on a hard disk you'll see I get the same old context menu I get in Windows 95, and I'll bring up the style log and all the functionality is right there, available in this Web page. Furthermore, we can integrate things like content. I can supply information about what's new at Microsoft and have hypertext links and changing content.

PAUL MARITZ: So you can customize these pages to put specific information on so an MIS organization can push information out through this mechanism.

JOE BELFIORE: You can have "agenda for the day" or whatever you want showing up in these templates. Let me show you how it's easy to use and customizable. If I click on the hard disk, one of the interactions we can take from the Web, I get a view of my hard disk and you can see all the files and folders of things in my hard disk and I can navigate around those in a single quick way, and this is ostensible in [that] what this is an example of is an active object showing me the amount of free-use space in high hard disk. They're fully customizable, and people can integrate in the shell in a better way than before. All the navigation paradigms they're used to on the Web now work on the shell. I also have a forward button and a backward button like I would on a Web.

PAUL MARITZ: People don't have to think I'm on the local Web so I can do this, I'm on my local hard drive so I can do, et cetera.

JOE BELFIORE: Exactly. Let me open the control panel here and I'll show you how we can integrate even online content on the shell. We have Microsoft support on line so if I'm having trouble with something in my PC configuration I can choose something, I can type something, do a search and get an answer back. All this is a closely integrated part of the shell. Another thing that's actually very interesting and worth talking about here is the fact that this window not only shows you local content; I should point

out that I can also use it to browse through the Internet. So the Internet now becomes a part of the shell's name space and if I click on the Internet I actually go out there and load [pages] all in the same window. I can browse around from links to the local content to the Web. It just doesn't matter, it's all exactly the same paradigm. Nothing new to learn. So that gives you an example of how this unification makes it easier for people and less to learn. Another great example of a system service and a new idea we can take from the Web is an idea of a favorites list. So here, as part of the shell, we get a list of favorite items which I have in a folder and you can see that now this becomes a much richer kind of thing. I can keep things like my hard disk in my favorite—

PAUL MARITZ: You've got different and unique icons here. Where are those coming from?

JOE BELFIORE: This is an ordinary Internet shortcut but if you look at the globe and mail and the University of Waterloo newspaper at the bottom, those have a gleam on them because it includes a background service that is checking these Web sites. So if I want to see what's new on the University of Waterloo, I can type in and see what's new on the Web site to make browsing easier for me. Also, Web authors can customize their own Internet shortcut icons. Let me show you a little more about how some of this stuff works. Another neat thing is we have these precreated category folders and these are smart and can go out onto the Internet or corporate Internet and get sites for you and you have a pick-of-the-week list. So I might have the Internet provider giving me picks of the week; and I can click a button and drag them up here and now it's permanent.

PAUL MARITZ: So we have a place where people can integrate into the environment, use it to display information.

JOE BELFIORE: Exactly. And it's all done through HTML. It's very straightforward. One thing people talk about when talking about Internet browsing and making it easier for people and safer, parents are often concerned as well about the content that's out there on the Internet and how they can make sure that their kids won't run into anything that will be offensive or they're not ready to teach them about yet. So I'll give you a quick example. I'll bring up the property style log in the Internet, and our Internet browsers now will support W3C picks committee compatible ratings. So I'm show[ing] you quickly how this works. Pretending I'm the parent I will type the supervisor password in here which will let me get to the Internet ratings dialogue, and what you see here is a simple example of Internet ratings and this has been provided by the recreational software advisory counsel who we're working with and I can decide for language I only want this machine or my kids to see a certain level of stuff and they've defined these four categories and I won't go into this in detail now. You can all play with this on your own later.

(Laughter) But I will show if I try to use the browser to jump to a page that has adult content in it, you may want to close your eyes, but you don't need to because I have restrictions on and the browser will stop and say sorry, Internet ratings won't let you see this site. We also have a feature called "ask mommy edit control" where they can ask and mommy comes and types in the password and they can see the site.

PAUL MARITZ: Spawn a whole new generation of teenage code crackers here.

JOE BELFIORE: That may happen. (Applause)

JOE BELFIORE: Another really cool thing that we can do to enhance Web browsing and to show really close integration between the shell and the Web is to take advantage of some of the browsing metaphors we've introduced in Windows 95. The Explorer is great for high-end users who want to get an overview of things. So I'll show how this will work with Web browsing. I have the Microsoft home page. As you know, one of the larger and perhaps more complex sites on the Web. And when I bring it up in the Explorer, the Internet actually becomes part of my shell name space. So I can expand the Microsoft page here and what I'm looking at is the contents of a site map which is the site [the] author has created. It's a simple text file and it gives people a way to navigate around the site in a hierarchical way which is—

PAUL MARITZ: So it's going the other way of taking some concepts that are useful on the PC applying it back to the Web.

JOE BELFIORE: Exactly, exactly. (Applause)

JOE BELFIORE: That's another quick example. Actually, [the] key thing about Internet and Nashville is enabling communication scenarios. I'll show you we're trying to address that as well, and what I have as a subpart of the Internet is a messages folder which shows me e-mail messages, which will provide SNTM pop messaging services here, and I can click on messages down here on a list and double-click them and get a new window, and [they] become hyperlinks for me and so on. We're also providing a news reader that works in the same way.

PAUL MARITZ: Integrated into the browsing experience.

JOE BELFIORE: Totally integrated. Nothing new to learn. You mentioned customized Internet before. I'll show you more closely how people and companies can customize their public folders and places on their Internet to give access to data. So actually what I have here is my own personal public folder. I switch to it—

PAUL MARITZ: So this is a directory that you can be sharing out from a server or your local machine using networking services.

JOE BELFIORE: Exactly.

PAUL MARITZ: Now you can make it look like a page.

JOE BELFIORE: Right. What I can do is make my public folder personalized and show a little bit about me so when people connect to my server on my page on the network, they can see what I look like and put some customized text on here.

PAUL MARITZ: Nice photo. Get it from the DMV?

JOE BELFIORE: Actually, that's the passport photo, so pretty close there. This is totally customized. All I have to do is create an HTML document and save it in this directory.

PAUL MARITZ: One of the key points I was making earlier, Joe, was this idea of dealing with information that's stored in HTML and extended HTML but also being able to provide this common navigational paradigm across existing documents which is important to the Internet world.

JOE BELFIORE: Good point. Let me show you quickly how that works. What you're saying is lots of people have lots of content they've created in their organizations and they want to publish in a Web-like way and have this be as integrated as HTML or anything else. What I have here is a MicroGrafX designer document which you might imagine our design team has created and wants to publish on our Internet. And this is the shipping MicroGraphX designer now and what happens when I bring this up from my favorites folder, is you see the designer document showing right here in [a] pane in the Windows Explorer. What I have is a floating toolbar here.

PAUL MARITZ: So from the user's perspective they'd be using standard navigational techniques, using the links, et cetera. They happen to navigate into a page which isn't stored in HTML but they don't have to worry about that.

JOE BELFIORE: Doesn't matter. They never need to know how it's implemented.

PAUL MARITZ: Any application, and this is just using an extension of existing OLE technology, so people who have done compound document work don't have to do a lot of new work. It will just fit in. And they're integrating in the toolbars in the top there.

JOE BELFIORE: You'll see these menus belong to designer and some of these menus, the Explorer menu and the Favorites menu, belong to the shell so I can still navigate using the menus and the same is true of the toolbar. This top one belongs to the shell. Here is my Back button and Favorites, and this is Designer.

PAUL MARITZ: This is an open standard we're encouraging everyone to do. We're encouraging applications vendors to do view-only versions of their apps and give those away on the Internet so people can have a rich navigational experience without having to go and buy the apps necessarily to see the documents.

JOE BELFIORE: Right.

PAUL MARITZ: And people can also use our componentized browser services to make links.

JOE BELFIORE: Exactly. People can browse forward and back. It doesn't matter. It's all integrated.

PAUL MARITZ: This is real live code of what you'll be shipping in the Nashville release. What we're looking at now is the next idea we have which is a prototype to extend it into a more integrated environment where you really are using the fact that you've got online and local information closely integrated.

JOE BELFIORE: All the Active services make it possible to make the whole environment really [im]mersive, rich and filled with content. So what you're looking at here is a prototype of enhanced themes that will be part of Nashville, where anyone can create a theme just like in Plus 95, but they're active and live. So instead of being a background bitmap it's a background HTML page. And this is the power records theme and, for example, right here they've downloaded a set of album picks of the week and if I want to look at Fujihara's album, I can get a track list, click on a track and hear the sound being downloaded through RealAudio or some other technology, and the experience doesn't stop there. Since all the page views, the hypertext views, are created by HTML, when I open my computer I can get an album pick of the day or an album pick of the week. When I open my "C" drive I might see another one and if I want to get more information about this album—

PAUL MARITZ: This would be a great thing to do in a home environment where you want recreational information, but you could just as easily do this in a work environment where you can use the same information to populate them with work information, latest information on companies or company movements or whatever.

JOE BELFIORE: The possibilities are just vast, and it's as easy as customizing your HTML.

PAUL MARITZ: Thanks a lot, Joe. Really appreciate it.

JOE BELFIORE: Thank you.

PAUL MARITZ: In closing, I'd like to summarize for you, then, we're making several major announcements today. In fact, a lot of major announcements today around these initiatives to really push forward into the Active wave of the Internet, and really achieve this synthesis that we've been describing to you.

First of all, we're rolling out today and providing developers with materials they need to really understand and digest this Active platform. We have major announcements, as I referred to earlier, with Intel in terms of support of over a hundred other companies to really lay down the infrastructure for this rich, interactive world. We're working with progressive networks in the context of our ActiveMovie synchronization services to take the next step in Internet distributed audio. We're working with SAP and will have an announcement with them actually on Thursday on C-bit in Germany on how we're going to the next step in terms of building business applications infrastructure in an open way for the Internet. We'll be integrating the CITRIX content so you can use existing Windows a.m. unmodified on the Internet. We're handing out today the pre-release Internet 3.0. We're describing the various initial steps in terms of controls, digital signing; we've announced we've signed our license for the Java technology which allows [us] to integrate that in and get a great synthesis there, as well as taking these technologies to other platforms. So there are a whole series of announcements I'd like to draw your attention to.

In closing then, Microsoft is really committed to growing this market, and we believe the best way to grow the market is not to have two separate worlds, [it] is to achieve the synthesis which is the best of the PC and best of the Web and do it in an open, highly functional rich way. We will be encouraging people to use these technologies as widely as they possibly can be used. We will distribute our Internet Explorer broadly, we'll build it into Windows, we'll allow other people to license it freely and distribute it.

We're integrating this whole paradigm which is particularly important in the intranet environment where you really can get a common view on information, your existing information as well as new information in the common infrastructure. And we believe this will—this synthesis will yield by far and away the best return for developers. And this is the way to go forward in terms of growing the market for all of us. I'd like to draw your attention to an announcement that in fact is being made in real time right now in New York, and it has to do with us really encouraging the spread of the Internet Explorer 3.0, [and] these ActiveX Technologies. As we speak, [Microsoft and AOL] are announcing a broad partnership between Microsoft and AOL to agree on a common browser framework around the Internet Explorer 3.0 and really promote widespread use of these technologies. So this will be a great step forward in making these technologies broadly available. (Applause)

PAUL MARITZ: And both Microsoft and AOL will get firmly and squarely behind making these extensions freely available to millions of users and to using these extensions in our own content, because both Microsoft and AOL are content developers as well, and we will fully exploit these extensions. So there's a great promise there that if you exploit these extensions you will have a large market to reap the return on. So there I will stop and take a few questions and then actually members of the press are welcome to join us, and we'll go back and listen to a live feed from the Bill Gates/Steve Case conference that's going on in New York at the current time.

If you want to ask a question, at this point I'm also going to ask John Ludwig who is our next speaker to come up and join me so that we can take questions together. If somebody wants to ask a question, please go up to a mike and I'll do my best to answer the question or get somebody to answer it if I don't know the answer. Yes, sir.

FROM THE FLOOR: Josh from Forester Research. I'll wondering how your AOL/Internet Explorer announcement relates to the AOL/Netscape announcement, gentlemen.

PAUL MARITZ: As I understand it, the Netscape announcement pertained to one particular service they have, which is their AOL GNN Internet point-of-presence service. The announcement we have pertains to AOL center, the five million users as compared to the small GNN surface.

FROM THE FLOOR: So the next AOL compliant will include the Internet Explorer browser in it?

PAUL MARITZ: Correct.

PAUL MARITZ: On the other side, sir.

FROM THE FLOOR: Can you hear me?

PAUL MARITZ: Yes, I can.

FROM THE FLOOR: You mentioned multiple platform support, but I see no explicit mention of a Unix anywhere, particularly RISC-based Unix systems for the browser.

PAUL MARITZ: We ourselves are developing with our own development resources for Win32, Win[dows] 3.1 and the Mac[Intosh]; we are working with other companies like Spyglass to make sure there will be Unix implementation.

JOHN LUDWIG: We've worked with Spyglass to make sure they take all we do on browser to their Unix platform and to all the OEMs who use that today. So broad distribution of these capabilities on the Unix platform.

PAUL MARITZ: On my right here.

FROM THE FLOOR: Hi, Paul. You talked about the delivery schedules and they used to take two and three years. Will you be improving the delivery and when things will actually come out now so they actually match some of the things you're putting on the table? For example, before second half, after second half, and actually start meeting those even if they are slightly less feature rich? Is that the intent?

PAUL MARITZ: This is kind of the reformed alcoholics argument here. Yes, our intent is—is to try and be a lot more rigorous about giving commitments that we can live up to, which is why I'm not giving you a specific date, but I'm giving you a range that we're comfortable with meeting.

FROM THE FLOOR: It's difficult from a corporate side to champion some of these technologies when things have a tendency to move nine months, 12 months past when we're expecting them.

PAUL MARITZ: That's a good point and that's why we're trying to break things up into smaller buckets so we don't have these large releases where we get ourselves all constipated.

FROM THE FLOOR: Thank you.

FROM THE FLOOR: Two questions, the first one being you mentioned a lot of different things that will be contained in Windows NT 4.0. You didn't mention a network directory service. Is that going to be contained in [Windows] NT 4.0 or does that come in Cairo '97?

Second question, you also mentioned that you wanted to be able to have access to all different types of structured data. This was in an earlier part of your presentation. What are you doing in terms of getting data off of all the legacy systems to address that?

PAUL MARITZ: I'll try to take those in the order you asked them. The Windows NT 4.0—I should have mentioned it in passing—will include distributed object services. In fact, those are being handed out in early code form to the developers today. That will be put out in 4.0. In terms of taking the next step in directory services above and beyond the domain services that we have in Windows NT today and the DNS services, that will come in the '97 release of Windows NT where we're doing the work to integrate both directory services as you need them for an internal corporate multi-site corporate environment but integrate them with the Internet domain services so again we don't have this bifurcated world. So that's where we've gone back and done re-thinking of our strategy to recognize the reality of the Internet as well. The second question was—remind me.

FROM THE FLOOR: The second question was in regard to structured data types.

PAUL MARITZ: What are we doing to get stuff off the mainframe.

FROM THE FLOOR: Because all you did was talk about your SQL server. How about the other stuff?

PAUL MARITZ: Sorry if I came across that way. First of all we have the database connection that we support, supports ODBCs and the way we get the SQL server is through ODBC so it is a technology that works across anyone that supports that. We'll be—one of the announcements is we've acquired technology from a company called Aspect which will allow us to improve that release as well, and when you combine the ODBC technology [that's] built into BackOffice, [which] is OSF communications we can reach up to legacy data stores.

FROM THE FLOOR: Terrific. Thank you.

PAUL MARITZ: Yes, sir.

FROM THE FLOOR: A lot of the Web enhancements I've seen to date have been cool but really slow. What kind of bandwidth assumptions are you making on the client and server side and are you bifurcating your assumptions for intranet and Internet applications?

PAUL MARITZ: In general, we're trying to work with assumptions of most Internet users in terms of the general Internet will be working on 28.8 links. We're not assuming that by somehow, by some magic we're all going to get cable modems tomorrow. On the other hand, we are trying to engineer all of the

work that we're doing to be scalable because there will come a day when we do have higher bandwidth links. So the media services we're doing all anticipate being able to scale from a 14.4, 28.8 world right up to megabits per second, and you have to really think hard how to do that, and that's one of the key thoughts that we fortunately have from the investment that we did in interactive television. We now have actually in running trials over ATM in Tokyo, so we know what it takes to go up to a fully video-enabled ATM world. We've been able to work backwards to 14.4, and [see] how do we lay things out so we can scale forward in the future. One more question, and then I'm going to need to stop unfortunately.

FROM THE FLOOR: Thank you, Paul. Lee Freedman. First, I want to applaud you on the ActiveX concept. It looks very promising. My concern with that technology, however, is in the demos that you showed, you have lots of active objects that are hidden behind others. All of this, "A," increases the bandwidth needed to get that page or that information, that HTML object, downloaded for somebody to use, and, "B," increases the disk space requirements on those. How or what kind of a strategy do you have in order to do just dumb things like compress this stuff better so it transmits and stores better, rather than trying to put some fancy disk compression algorithm on the disk itself to double space and stuff, which ends up being inherently dangerous to the system? We're sending around mostly ASCII text and it's pretty inefficient. So there are ways of being able to make that work better. Is Microsoft looking at dealing with some of those problems to make these things more efficient, to store and move around?

PAUL MARITZ: John, you want to comment on this.

JOHN LUDWIG: Yeah. A lot of investments we're making in the multimedia space in particular are very much focused on trying to do things in a bandwidth-efficient way. So we are—rather than sending out a lot of bitmaps and that kind of data—you can send out much smaller description files. So we are sensitive to that. We're investing a lot in caching strategies so you don't have to reload from the net very often. We're also working with the industry on how to do simple compression of things such as HTML as well, in a way that we can support so we can all move to technologies there.

PAUL MARITZ: So we do already have on-the-wire compression, so when it goes over the wire it's compressing. We need to shut up shop here so I'll be happy to take this topic up further with you. Thank you very much, and I'd like to introduce John Ludwig.