

Internet Platform

Transcript of John Ludwig's presentation at the Professional Developers Conference.

March 12, 1996

John Ludwig
Microsoft Corporation

Download the Microsoft Word (.doc) format of this document (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 :[].

JOHN LUDWIG: Thank you. (Applause)

JOHN LUDWIG: So Paul did a pretty good job of introducing the active platform that we're investing in, the ActiveX™ Technologies. And what the role of my presentation is today is [to] try to give you a little bit more insight into the active platform, what it is, and why we're doing it the way we're doing it.

So I'm going to dive into that. As Paul said, our intent is really to put the best of the PC and best of the Web together, to come together in this active wave. We've got the PC evolution with all its technologies and the Internet evolution. We want to get to a common side, ways of interactive common: Applications over the Web space. And with rich interactive multimedia which is exciting content, very compelling content. That leverages the best of the PC hardware architecture and gains architecture and combines it with the best of the communications infrastructure of the Web. So that's what we're doing, I think as Paul fairly clearly articulated.

Where are we starting from? We're starting from, most of us, a very large base of computing that's already been established, a huge base of users and tools and applications that have already been established by all of us working together. And it's been a very successful model. We've built up a huge industry together. We've got 150 million-plus users of Windows today. We've got 10 million-plus users of office productivity tools, documents, we're all very familiar with that. We've got five million developers already working on developing great solutions in the personal computing space. Three million of them using Visual Basic®, Visual Basic for applications products, but just a tremendous number of people. I've heard estimates where for the V[isual]B[asic] programmers there's a billion line of codes per year being written by these folks. So that's an incredible amount of investment and training we already have. We already have a thousand-plus suppliers of software components that today have control of those OCXes, a very successful market today, very active market in component software. And that's kind of our starting point. This huge industry we've all worked together to create that's created a lot of value and a lot of products and a lot of great technologies.

Now, our goal is to get from that starting point, that launch point, to this active Web, this active wave. And our key challenges that we all face, we face these in developing our own products, you face these in developing your products, are pretty, you know, simple things. We all want to know how fast can we get there, how can we move from where we are today, the skills we have today, the tools we have today, the technology we have today, to this new active space, these cool active technologies, and communicating, interactive, immersible applications on the Internet on your PC, on the Web. We want to know the lowest-cost ways to get there. How can we get there without spending an incredible amount of money on development, on training, on new tools and software. How can we get there as fast as we can, in the lowest cost there, and then once we've gotten there how can we get the most bang for our buck? We've—how can we

all make money off of that? How can we work together to make money off of that? That's what I want to talk about in my presentation today.

One of the different paths to our goal; I want to talk about some straw-man paths for our goal that we certainly looked at ourselves. In the past year and a half or so as we've increasingly focused on the Internet we've looked at a lot of different ways for how we can get to this active wave. And certainly one way is to create an entirely new Internet focused language and development environment. And just reinvent the entire industry around the Internet.

This would mean having new languages with brand-new, you know, syntax, brand-new tools to support those languages, all Internet-optimized. Brand-new object models that don't necessarily leverage [w]hat we already have and know today. Brand-new libraries, brand-new applications making use of all this. And as we reinvent this, make it completely OS dependent, don't have any dependencies on the operating systems of today. This is certainly a path we could follow. And we call this kind of the complete break path.

As Paul said, there are a lot of shortcomings in this path. And we don't actually recommend this. We think this is fundamentally a broken solution that we don't want to have the industry follow. We think to turn our back on that huge group of users and developments, software market today around OCXes is just a foolhardy thing to do. It's an incredible waste of all that expertise and talent and energy we've built up around that environment.

At the other end of the spectrum, you can try the far alternative approach. We can just take what we have today and shoot whatever technologies we have on the Internet. That's all we do, just focus on that. And we just use existing tools. We don't worry about inventing new tools. We just take what we have today and try to make them a little bit better. Existing languages are all we need. We don't try to adopt any of the new interesting Internet technologies in this space like Java. We just say if you want to have an object model, you just take the full OLE support that we have today and make that everywhere the way you do it with all the complete set of capabilities that are in OLE, everyone if you want to have it run on a broad number of platforms, you say it's Windows everywhere and we make that as real as you can and push Windows, Windows, Windows everywhere.

This is certainly another model we could follow. This is kind of the more of the same model. And we don't actually recommend this either. And I think a lot of people tried to cast us in the industry as saying this is what Microsoft recommends that the industry do. This is not what we recommend. We think this is kind of a tired, not wired, approach. This is a solution that worked great for us all. We developed a lot of products on this path. But ultimately, going down this path all by itself, it didn't anticipate all the Internet explosion and all the new Internet technologies. And so we don't think we can stay on this path.

We think, obviously, what we need to do, working with you, and to combine the best of the PC development environment and the best of the Web development environment and bring those together. We want to have both existing languages, all of the investment we've made today in our existing languages and Basic and Visual C or other C products, we want to leverage all that and have new tools and languages like Java and integrate that into our approach. We want to make sure that we integrate new Internet technologies like HTML and HTTP and Java and all other kinds of innovations in the Internet that are happening today, right back into our platform in a very fundamental way.

We want to have the best of today's object model, just the lightest-weight part of OLE that you need to support Objects COM, which is what's really supported by a huge number of OLE Controls today and all huge range of software. We just want to focus on that and just leverage that small piece of the object model. And we want to do all this and make sure we have great cross-platform support. We want to invest in a tremendous amount, we know that's required in

today's market to make sure all our platform works cross-platform on every popular computing platform so that people can deploy solutions that target the broadest number of users and eyeballs out there. At the same time we want to make full use of these platform capabilities. This is what we call the ActiveX platform, the ActiveX initiative. It's a set of technologies designed to meet these goals, to support existing new tools and languages, to work cross-platform, to integrate Internet technologies in a very fundamental way into the COM programming models that we already are familiar with today.

So what I want to do is talk a little bit about how you merge those Web and PC technologies. And I'm going to expand upon some slides that Paul had. And I'm going to show how we take today's Web clients and meld them together with these other technologies to create the best of the PC, best of the Web in this active environment.

Now, of course, where we are today is your basic Internet browser that does a great job of displaying very nice-looking text and graphics and has a very easy, simple-to-use navigation model, simple user interface, something that users are comfortable with, been a very successful class of software over the past year or so. What an Internet browser today, which I think we all know it displays HTML content, very text[-heavy], images static, inside a frame and a simple set of protocols to get that data from the Internet, HTTP. Small, simple protocol, that's why it's been adopted so broadly, because it is simple. But it's also very limited. And you can't do a lot of rich, interactive things, because of the limitations of the design today.

So what's the first step that we want to do to make this really exciting and really active and much more compelling? The first step is you need to start to add behavior into these pages, something that's more than just static text and images that's been precomputed. And so that's what these Active[X] Controls are. Active[X] Controls are the standard way to add behavior to a Web page or to any document, which I'll talk about. But it lets you just simply insert any kind of active control into a page, whether it be—and I have some examples here of some multimedia controls, an ActiveMovie control that does movie screening and play back in a page, or ActiveVRML for 3-D worlds rendering or interactive worlds or Direct3D for a really high-performance 3-D rendering and game support.

All of these things can be simply plugged in using an open-standard interface into any client; in this case, the Web browser. And that's the key first step to adding a level of richness to today's Web environment. It also leverages every investment that we've made already today in the PC development environment, because Active[X] Controls, our OLE Controls, and they just add more and more capability over time. But they leverage the already-existing investment of thousands of OLE Controls, which can now be embedded into Web pages without any additional development work by the OLE Control community.

Now, once you've done that, that's great, you can have point elements on your page, really cool animation or really cool behavior. And that's pretty neat. But you want to go more than that. You want to be able to make these things talk to each other and have some interaction. You want to be able to have one control that's getting data from a database application and feeding that up to some multimedia control that plays back data in a certain way.

So the next step is to add Active Scripting to a page, this allows you to embed, now, simple behavior right in the page in a script form. That script can interact with all the Active[X] Controls, it can get data from those controls, it can get events coming up from those controls, it can ask those controls to do things. And all in a standard open way, a standard interface for how you talk to any given control that might be embedded on a page. And the interface for how you plug in a script interpreter is also an open interface, so that any script interpreter can be plugged in.

And we'll certainly talk a lot today about some script interpreters we have today like VBScript and also about JavaScript compatible support. It's an open script. You can plug in any architecture or language. You can add any because it's an open interface for plugging in scripts that can interact with the rest of the page.

Once you've done that, you have what essentially is our Internet Explorer 3.0 product. And that's what we're going demo a lot today [and] over the next couple days. This is the heart of the technology for the Internet Explorer 3.0. Adding Active Scripting, Active[X] Controls in a page, and then define the object model within the page, the methods and events that everybody agrees to support within the page so that they can all be easily scripted and accessed by scripts and by these Active[X] Controls.

That is the heart of what Internet Explorer 3.0 is. And you're going to see extensive demos today and the next couple of days about how to make best use of that. That's great. And it's given us these really active pages now. But it hasn't done the full job of integrating in the rest of the PC space. And so the next step in the evolution is to say, all right. Not only do I—can I see Web documents, HTML documents, in this frame, I should be able to see any kind of document in this frame.

And any kind of document should be able to be an Active Document that shows up in the frame that can participate in the menus and the toolbars, the frame, or provide new things to put on that—on those menus and toolbars, and can sit in-frame and have the same navigation model that a Web page has. So now you can go navigate simply between Web pages and other document types, whether it be a Microsoft Word document or a Visio document, or gosh knows what. You can view them all right in-page and go hyperlinking between them and have them all support the same navigation methods.

And because the interfaces for how you add Active[X] Controls and Active Scripting are all documented and open, you can add those into any other document format. Anyone can go and host, as many applications do today, Active[X] Controls so they can add behavior right inside their application. And many applications do this today. Many applications are increasing containers of Active[X] Controls, so, like, Lotus Notes for example has a container for Active[X] Controls. So over time you can add Active[X] Controls to any application so they can take on these Active Document capabilities. And they can also add support for Active Scripting so any application can easily add scripting support in just by supporting these interfaces. And they can do it in a language-independent way.

Now, beyond that, the next step is to—we've enriched the client area a lot by adding behavior and all other kinds of capabilities. But now we also want to enrich the client/server communications paradigm so you can do lots more interesting things between the client and the server. We want to broaden beyond just HTTP so you can have arbitrarily complex protocols between clients and servers and people can write much more innovative things in this space. That's where we add Windows Sockets underneath the browser as a communication paradigm. You can still get to HTTP as well from the browser. But now you can get there and have two-way direct communication between the client and the server. Now your control on a page can have any arbitrary conversation with a VCR and get any kind of bit streams back and any kind of data streams back that they want by using Windows Sockets or distributed COM for the development of the client/server communication.

So that's the next step, even beyond just the active page is just to have that greater level of activity between the client and the server. You saw this picture from Paul Maritz. This kind of—in a different cut—is the architecture of our client, our active client. And at the top, you have applications, tools, and content that are going to make use of at the next level down, the ActiveX interfaces, the Active Documents, Active Scripts, Active[X] Controls, active multimedia things like ActiveMovie or ActiveVRML. All these services will be made available and

applications to make use of those, as they wish. Those are then hosted in turn on COM—the COM object model and the COM interfaces for binding to objects, for getting the services to objects. And then some standard services. URLs and hyperlinking for actually resolving names and getting to data and the direct technologies for getting to fast video playback for fast audio playback for great graphics and sound support on the platforms. Additionally, these active libraries may use system services from the Win32 system services or the Mac toolbox to be implemented and we'll go straight to those as appropriate to get the best performance or features.

The key thing is the active layer, the pink layer, URL, DirectX, will all be made to be completely cross-platform and work on the widest variety of client platforms possible. Work with any operating system, Windows NT, the Mac, Windows 3.1, Unix. We want to target the broadest set of platforms that people like you want to target with your applications. That is the big development that we're making to take this cross-platform. And actually want to demo some of our work for cross-platform, focusing on the Macintosh platform. So let's go to the demo station. And I have with me here today, Don Bradford, who runs our Macintosh development team. So thanks, Don, for coming on stage. What we're going to do is do a quick demo. Some of our capabilities on Windows and the Mac. And I'll kind of kick it off here on the right-hand side over here with Internet Explorer 3.0. This is, of course, or Internet Explorer browser. It provides, of course, a lot of great HTML support. We have some interesting innovations, HTML, like fonts, that we support, so you can have better-looking text on the page. We also support background sounds, in-line movies, and I'll see if any of that is working here.

Let's see. Microsoft Corp. MSN home page. I didn't customize the page. But we can see the marquees here that we have on the Windows side, the scrolling text. And let me see if I can bring up the volcano site. Where did we hide that? There it is. No, that's not it. Well, you have the volcano site, don't you?

DON BRADFORD: I think I do.

JOHN LUDWIG: We have a pretty good HTML browser here. I think our Mac product is kind of a secret, Don, because I don't think a lot of people know it. What do we have?

DON BRADFORD: It's a tough ask. If you look at the volcano page, you've got in-line video, sound, and you've basically stuck me on an eight-meg system here. So we'll give it a shot. But....

JOHN LUDWIG: Wow. You have video online just line—the text support, the font support, I guess we would have sound support, I'm not sure if we have the speakers cranked up on this thing. But we have all that on the Mac today, huh?

DON BRADFORD: Uh-huh.

JOHN LUDWIG: Now, this must be a pretty traditional Microsoft Mac app, this probably takes, what, 16 meg, 20 meg?

DON BRADFORD: Could if we wanted it to. We're running along fine at four or five right now.

JOHN LUDWIG: That's pretty good. Wow, that seems like about the smallest browser available on the Mac. Pretty much that's the case. What's your charter, is that to make this the best Mac browser period, or, like, are you constrained?

DON BRADFORD: I wouldn't say we're constrained. We're more kind of encouraged and invigorated. We want to make the best possible browser for the Macintosh, bar none.

JOHN LUDWIG: Why don't you show me some things. Because I want to make this a good Mac browser.

DON BRADFORD: A lot of nice features in Windows 95 that are hard to duplicate on the Mac. One of them is favorites. There's kind of an existing file system. Of course, we've leveraged that a bit. We've actually taken things out; at that point, we have complete interoperability with the—we actually support a lot of Mac-specific technologies. These are pretty critical to both client and [server] developers for the Macintosh. If you notice down here, we support Internet configure, configure. That basically lets us work seamlessly with the Macintosh.

JOHN LUDWIG: I don't know what that is. That's not on the Windows, is it?

DON BRADFORD: With the registry and other services that are on Windows 95, but with the Mac a lot of this is provided by third parties. We've been aggressive in listening to our ISVPs and adopting things that are really important.

JOHN LUDWIG: So what other kinds of things can you show me on the Mac browser? What about multimedia features or—like maybe VRML sites.

DON BRADFORD: I think we can show you VRML in a couple of features. We chose to do very lightweight VRML. Let me just pick one out here. We're actually live.

JOHN LUDWIG: Live on the Net, so we're downloading a VRML page now, which is probably not small.

DON BRADFORD: Uh-huh. And we've got the connection. And here we go. Those of you familiar with it, this is a teapot.

JOHN LUDWIG: You have VRML support already on the Mac?

DON BRADFORD: That's right. At this point we're not using any Active[X] Controls or plug-in or anything, it's all native.

JOHN LUDWIG: That's pretty cool. I mean, we have VRML as well on the—on Windows, but I didn't realize that we have it going on the Mac yet. I can go to the same page, I guess; which one did you load, the teapot?

DON BRADFORD: Actually, the teapot. We'll download that. It's taking its time.

JOHN LUDWIG: Oh, look at that. Well, you know it's real time. So I guess we won't show that. Let me exit this. Let's go back to your demo, Don, since the Mac product seems to be a lot more stable than the Windows product. (Laughter.)

JOHN LUDWIG: I guess we put our best and brightest on the Mac. We'll just stay over there from now on. (Applause.)

JOHN LUDWIG: So why don't you show me -- I know you've done the VRML; what are our other multimedia investments on the Mac? What about the Direct3D or DirectDraw?

DON BRADFORD: Let me show that [to] you, you've seen the teapot using some of our standard technologies. Now let's show what you we can do with some of the DirectX stuff. This is the teapot; a little bit different than the one we showed you before.

JOHN LUDWIG: That's a pretty cool-looking teapot. That's built out of what, the Direct3D work that we're doing.

DON BRADFORD: Yeah, this is running on the Mac with no hardware acceleration at this point.

JOHN LUDWIG: I think that's kind of early alpha shape now, but that's our Direct3D technology that we're going to make available cross-platform on the Mac and on Windows, and this is the early alpha, the Mac running, we do have that running hopefully a little more stably on the Windows machine as well. I could show the same picture, but I think I'll show one that's a little different, just cause I think this one's pretty exciting, too. Here we've got Direct3D running on the Windows platform and probably the key difference right now is we have the hardware acceleration working on the Windows platform, so you can see a lot more detail and higher frame rate, and a little bit better performance. We're going to do the same kind of work on the Macintosh and have the same kind of capability on both platforms. We are committed to bring the best of the direct technologies to both platforms, so there's a common way to get high performance, multimedia capability on both platforms.

I think another cool thing to show, Don, and maybe I'll start on the Windows side here, is going back to this page. We're making a lot of investment in scripting, obviously. I've talked about our Active Scripting supports and how important that is so I've got a simple demo page that's got VBScript behind it, a little pizza order form. You can pick a style, Hawaiian or Chicago, and you can see that the toppings vary on the other side, as I pick a style of pizza based on what you're going to get. And you can pick one, say order, and it'll tell you your order is placed and we're going to charge you \$14. You can right-click on this and view the source. And you can see that it's a little hard to read, probably, but script language equals VBS, that's VBScript. And the script is right in line and interpreted on the fly by our browser. And this little app is run within the browser. Now, that, of course, we leveraged a lot of our great Windows technologies and our investment in V[isual] B[asic] to get that working in the Windows browser. What are we doing on the Mac on that side, Don?

DON BRADFORD: Nice that you would ask. Actually, we're running pretty close. Again, we're running at a very small footprint here. But barring any platform differences, it's very similar. So we can go through it.

JOHN LUDWIG: So you can do pretty much the same thing.

DON BRADFORD: Same thing. We'll give you—since you're from Redmond, give you a little salmon there. Place the order.

JOHN LUDWIG: This is the same VBScript, same page running on both, VBScript in it as well.

DON BRADFORD: It looks very similar. It's identical.

JOHN LUDWIG: So we have VBScript running on both platforms right now. That's great, that's pretty cool. And this is I guess an alpha version of VBScript on the Mac.

DON BRADFORD: Absolutely.

JOHN LUDWIG: And we'll be shipping that later this year, you think?

DON BRADFORD: Yeah. Basically, everything but VBScript is part of our offering that's been out for the last couple weeks.

JOHN LUDWIG: Cool. We've got scripting on both, and we're committed to bringing the full scripting features to both platforms and making sure it's a full interface to both platforms and it can be seen on all platforms.

Now, the next area that I think we'd like to talk about cross-platform is control support. To go with scripting, the other key element of getting active content is to be able to get controls on pages. And so that's the next thing we want to show is controls on both platforms. We're actually going to have a little bit of help here. I'm going to ask Norm Meyrowitz from Macromedia to come out and help us do this demo. I think he's hanging out somewhere right behind the stage.

So. Thank you, Norm.

NORM MEYROWITZ: Thanks.

JOHN LUDWIG: He is going to take a few minutes to show us shock waves on both platforms. You got this machine here and that machine there.

NORM MEYROWITZ: So, we're excited to be here, Macromedia and Microsoft are working together to use the Macromedia architectures and Macromedia Director and work with Microsoft on making sure that the ActiveX Control architecture and the multimedia architecture that we've developed really work well together. So the first thing that I'm showing you, the first fruits of this partnership, is the Active[X] Control for Shock Wave running inside of Microsoft Internet Explorer. So here's the Shock Wave movie running in the Internet Explorer on Windows 95. Let me show you some more exciting ones. Here's real corporate presentation done by multimedia advertisers out in the industry, this is for Intel's home page and this is what is on it today. So you notice that you have all of this animation, and that this is really a dynamic Web page that when I move the mouse on top of these buttons, you get rollover. It changes the information over here, and if I click on one of these buttons it will take me to an Intel page. So we're using the ActiveX client architecture to actually call out to the Windows operating system to change the page to hyperlink to a different Internet Explorer page.

If I go back, here's an example of something on Sony's music site. This is from the group Deep Forest, and if we turn up the audio on this, I think there should be audio on this, you can actually click here and you'll get music from the Deep Forest group. I'll sing if the audio doesn't come up. (Laughter)

NORM MEYROWITZ: Which would be a problem. We'll go back to that in a second. A final thing we'll have here is something done by another set of multimedia artists and this is a WebFrog game done with a Shock game control. If I start the game -- I'm actually terrible about this. I've already gotten killed. (Laughter)

NORM MEYROWITZ: Here I am. So here, what we've seen today, Shock Wave is pretty much the de facto standard for multimedia out on the Internet today. We've had a million downloads. We have 10,000 sites using Shock Wave and we've been able to wrap that technology in the Active[X] Control architecture really, really easily and it allows us to get to the scriptability that Active[X] Controls bring you, it allows us to get to the active client architecture to do more sophisticated Internet access. Let me switch over here to the Macintosh and show you Shock Wave running in Macintosh. Here's -- I'll show you Deep Forest here and hopefully there's audio—audio?

JOHN LUDWIG: No. Sing.

NORM MEYROWITZ: I'm going to sing. So if there were audio here I'd be moved to a mixer page that would allow me to mix all the music here. Let me go back and show you a couple of other interesting pages.

JOHN LUDWIG: Oh, Don. Don's been talking to our Windows team too much.

DON BRADFORD: We're one for one, John.

NORM MEYROWITZ: So why don't we move on, and -- (Laughter)

JOHN LUDWIG: Well, so I appreciate you coming out here, Norm. I think the key thing is showing that we're committed to bringing a control architecture to both platforms that provides the same level of performance capabilities on both. Is there anything else we should try to show on this side or should we just call it a day?

NORM MEYROWITZ: There's a lot of stuff we should show but we should probably call it a day and we'll show some more—

JOHN LUDWIG: Okay. Why don't I --

NORM MEYROWITZ: Chris is going to be showing some more in his talk and we can show some more there.

JOHN LUDWIG: Thanks for coming out, Norm.

NORM MEYROWITZ: Thanks a lot, John. (Applause)

JOHN LUDWIG: So that's a quick run-through of where we were in platforms today. I'm sorry some of the demos didn't work as well as they should. The other things we're doing, we're going to take a broad set of things across platforms, all the W HTML-like style sheets we've been working on, that will bring cross platforms. The COM, and Distributed COM architectures for the standard object services across platform, we'll be taking that all cross-platform, we'll have the full Active[X] Control architecture so you can host Active[X] Controls the same way in both platforms, the full complement of lower level active services, multimedia, database support, so an application will be written, run cross-platform and access any of these.

And then we'll also have a virtual machine that runs cross-platform, and Paul talked a little bit about our license with Sun Microsystems. We will have [a] virtual machine on both platforms that lets you access the full active features, lets Java applets be written and lets them behave as full Active[X] Controls and we'll make that available cross-platform as well on our browsers.

Now, the one thing I want to say about this architecture is there's really no drawbacks or limitations to it, but the investments we're making are going to allow code and controls to work cross-platform on all the interesting client platforms. We're also totally committed to making these things work cross-language so that any control or scripts can be written in any language that supports the open interfaces. So you can write in [Visual] Basic or C++ or Java or any language or VBScript or JavaScript or whatever language you want. We want our investment to work cross application so any investment you make in a control or a scripting language will work not only in a browser, but will also work in a whole set of applications.

All our Microsoft Office applications and other applications that support these standard interfaces. And it's also going to work cross-browser. We're going to make sure that the active architecture works inside of other browsers like Netscape Navigator so any investment you make in content and code work with any browser regardless of what browser the customer chooses to run. So I wanted to walk over here and do a few demos of that as well.

The first demo I want to show has to do with multi-language development and support for that within our Active[X] Control architecture. So I want to bring up a simple page here that is a page with a JavaScript and a Java control. Actually, let me maximize this because it will help us a little bit, and reload that. So here I have a page that happens to have a JavaScript in it which is automating a Java language control. And you can hit the play button, and a little video

comes up, and there would be sound with this as well. But apparently sound isn't working. So you can see this is a, you know, pretty classic, very simple application showing the interaction of JavaScript and the Java language. You can view source and you can see here that its script language is JavaScript, and all the JavaScript is in here, and then there's an object embedded down here that's the actual applet that shows the video. So that's a pretty standard scenario and kind of what people expect to work in the Web environment to some degree.

Now we've got another page here which is that same page, JavaScript now, but going against a control that's been written in Visual C++ and it's basically the same thing. It looks similar, you can write controls in any language you want to. You can write them in Visual C++, in Java, and you can write ActiveX Controls even in Visual Basic so I'll show that as well. And again this is the mix of JavaScript now driving a Visual Basic control. So JavaScript, Visual Basic control that does the video here.

Now, that shows how we have an open architecture that supports writing controls in any language that you choose, any tool that you choose as well. Not only do we support that, we also support any stripping language you choose. We can have the same page now; now it's the Java applet again. Now it happens to be VBScript that's driving the Java applet down here. And again, it's exactly the same, I can view the source and now you can see the script language is VBScript and the script language that is embedded is all VBScript code, the same applet that's embedded but it's all VBScript driving it. And you can see I have equivalent pages for driving a Visual Basic written control, VBScript driving a Visual C++ written control.

So you have the same exact page behavior regardless of what language and tools you use to write your code in. The active architecture supports mix and match of languages, mix and match of tools. That's a key thing we wanted to achieve with our investments. Now, another thing I want to show you is how active—ActiveX Controls can sit in many different applications so that your investment in ActiveX Controls goes beyond just browsers.

So first I want to go to a page here. It's called the model demo page. This is going to load in some controls, some ActiveX Controls, that will display, put some content upon the page. This is Seattle Modeling Guild. And you can see here that there's a bunch of controls. I can go down here, page through their book of clients and pick one out and that -- the fact that I picked it down here makes the rest of the page change. It makes pictures show up here and it also puts the picture up here on the spinning cube.

So this is ActiveX Controls with Active Scripting written inside of the browser, the Microsoft Internet Explorer 3.0. The same controls, though, can be used in other places besides just that browser. For instance, another place where you can see them in use, we have a—where's that VBScript demo? Let me go to the Netscape Navigator first. We have this on Netscape Navigator as well, the same content right in inside Netscape Navigator. It's the same ActiveX Control, same content, except now it's viewable within here. So I'll go to the bookmarks, bring up Seattle Modeling Guild again, you see the same basic page, again implemented with ActiveX Controls. I can go down now, pick on the same control down here, pick on the same element here. Again, it updates the controls on the page. This is using our support that we developed with—encompassed to get ActiveX Controls working inside the frame of the Netscape product. So that you can make your investment in ActiveX Controls and know that they'll work in any environment.

The last area I want to show that in is in Visual Basic. We have Visual Basic project here that—there it is. ActiveX Controls and Visual Basic. Those same controls are usable within Visual Basic. So here we have a little bit different look and feel because we've laid out the form and the page using the Visual Basic form layout tools rather than using an HTML page. But it's the same controls in that page. So you can see the same controls here showing up—and just integrated now in a Visual Basic application. So one investment in ActiveX Controls will

work in our browser, will work in Netscape Navigator, will work with V[isual] B[asic], will work with any other development tool that supports Active[X] Controls, and there's a huge range of them today from Microsoft but also from other third parties, products from Oracle, products from Sybase, [that] work with applications like notes that supports Active[X] Controls. So the investment that you make in Active[X] Controls can carry on into a lot of different environments.

Okay. The last items I want to talk about now are—I've talked a little bit about the technology in a little more detail. I want to talk now about the business opportunity around us. We've talked a little bit about how we think we're going to encourage development to happen in the Internet space, the tools and some of the methodology that we're going to encourage there and how we think we can get the fastest development space. I want to talk about the opportunity we're going to create with you so we can all benefit from this development work.

The first thing that we're going to do to make sure that we can all benefit from this work is we're going to seed the industry aggressively with this technology. We want to make sure it's widely available, widely usable; that everyone feels like it's very simple to adopt this technology in their products. And so we're going to seed the industry broadly with technology, we're going to seed it by making Internet Explorer free so anyone can make use of these technologies through Internet Explorer. We think that's the right thing to do to help grow the market overall as fast as possible for all of us. Free Internet Information Server so you can serve up this content and make it available and make it very easy to deploy rich sites on your sites and Internet.

The multimedia run times, the ActiveX run times, we'll allow those to be freely distributed so with your applications you can redistribute these as well so that your application can take full advantage of these services in any way it sees fit. We're going to provide free source for some of the major initiatives we have that we think will make the Web a more active space. So our VBScript will be available. We'll make our ActiveVRML available for people to use in their products; we're making sure there's no real obstacles to adopting these technologies and people can use these.

And finally, I think Paul said we're very much involved in the standards community so there's an open environment for innovation, and the innovations we support are supported broadly in a wide range of products by making sure they become standards and are freely implementable by everyone involved in the industry.

The next thing we're going to do besides just seeding the industry is we are going to, as Paul said, evolve rapidly in this industry. We're going to try to do a lot to not only plug the holes that are there in architecture today and add more capability over time, but also to do a lot of, you know, exploration work and try to make new things happen and new opportunities and basically expand the industry as fast as possible so we all have new opportunities and new ways to enjoy the fruits of our labor. In our first 100 days we've done a lot. We count 100 days from our December 7th strategy day, Internet strategy day, announcements. We've done a lot since then. We've gone into beta with our Mac, Windows Explorer, Windows NT, we've released an Explorer in 22 languages, Internet information server release, Vermeer acquisition, we've put these pages on msn.com to personalize their view of the Internet. We're working on partnerships in virtual, private networks. We've formed a new division to go focus on the Internet. We've done a lot of things. And we've been very aggressive about trying to make a lot of things happen in the space.

So there is more and more opportunity for all of us all the time. And the next 100 days we're accelerating. We are announcing a ton of stuff today that I'm not going through in all great detail but there's a ton of things going on today and a ton of things coming down the pike that we're trying to make hope, work in an open standards way in the next three months. These are

some of the lists of some of the partnerships and some of the initiatives we have underway today. There will be a lot more. But we're going to work very hard to continue the aggressive pace of Internet development and to continue to open up new frontiers for all of us to innovate and make more money. And from that base of growth and kind of the seed providing a technology, we really want to work with you to activate your business. We want to help you really achieve the benefits of working with us and working with these technologies. So there's a lot of initiatives we're going to do to help provide you opportunity. Certainly one thing we want to do and we are going to be very aggressive about is partnering with companies to cause innovation. Some of our partners will be more on-stage here today and later this week. We have a lot of partnering announcements we're doing today, and we're open to all kinds of innovative ways to work together with everyone in the industry to encourage quickest development and quickest adoption of new technologies.

We're very much focused on letting you have branding opportunities as you use our technologies. The reason—one of the reasons we've componentized all of our work so much and broken it up into controls and scripts and our HTML control and all the different various elements of our platform, is to allow you to make use of the technologies as you see fit and to get your branding on top of the technology. So you can make use of our active run times for multimedia or for Web browsing or for network access. And you can control the brand that the end user sees. You can put the frame around our browser and replace our browser frame very simply and control the branding that the end user sees.

We very much want to create the opportunity for you to create solutions that, you know, carry your brand and your image to your end users using our technologies. And we want to make sure that our platform allows that to happen. We very much want to help you distribute interesting client work that you do. I know many people in the Internet industry are creating client run times to view particular content types or to view particular types of transactions on the Internet, and a key issue for those client run times is distribution. So we want to work together with partners that are interested in this service to make sure that we can help them get their client distributed. For instance, we'll be working with Macromedia to help them get Shock Wave distributed with the Shock Wave player with Internet Explorer very broadly. We'll be working with many vendors like that to help them get their players distributed very, very broadly. And we want to work with our partners to make sure that we can leverage some of our distribution channel and some of our strengths and help you get opportunity by helping you to get your clients distributed very broadly in the Internet space.

We also want to help to promote interesting work that people do with active content on their sites, and if people get involved and do really interesting, compelling active sites making full use of these Active[X] Technologies, we certainly want to help promote your sites and give those sites some recognition for the work they've done. We have some vehicles to do that today, like our own site which is very, very high traffic site and which we can use to send traffic towards other sites that make full use of these technologies. And so we'll be working out the programs to let us do that as well. And we'll want to work with all of you to make sure that as you invest in creating great active content that we can help you get eyeballs towards that active content. We can help you channel traffic towards those sites and let them see that content.

Now, we're also going to do a ton of programs and initiatives to help you in a lot of other ways. And the basic place you're going to go for information from us is the Web site www.microsoft.com/intdev/. That's where we're going to keep all our Internet developer information. And it's going to keep all the developer services and programs and partnering activities up there. It's going to be updated continuously and it's going to have a whole huge set of information available up there. It will be [an] active site. So we encourage you to go there to see what are the active platforms or business opportunities that we might have. So going forward, I think our key messages are we really want to work with you to help activate your application. That's really what this professional developers conference is all about, is giving you

the information to start that process and getting the feedback from you as well about the technologies that we're making available.

I'm really interested in seeing you activate your site. Again, there's a lot of information at this PDC about how to activate your site and build active content. We'll be doing things throughout the spring about activating your content and one would be a Microsoft at the Movies event in May, in which we'll go very broad and make information available about how to author sites that make best use of active content. And then we really want you—we really want to help you activate your business. And you can go to our Web site for more information. Or if you're interested in talking to us about particular relationships, about particular technologies or business opportunities, you can send us e-mail at ipartner@microsoft.com, and it will give you a chance to take full advantage of our technologies and make opportunities for yourself.

That's the end of my presentation and I'll be happy to take questions if anyone has them. Yes.

FROM THE FLOOR: Yesterday, the Wall Street Journal chose to highlight one of the bullet points that you had on an earlier slide called "get ISDN" on their Internet home page. That's particularly intriguing to me, this get ISDN program, and evidently the Wall Street Journal thought it was important too because it was a headline on their Internet home page. Could you share with us just a few more details about the ISDN program?

JOHN LUDWIG: It is public on our Web site today. You can get information on it. It's a partnership to make it easier to get ISDN configured and ordered for your users. We've worked with this with every significant network operator in the country and we're working internationally as well. It's a true partnership. The business benefits from people signing up, accrued to the partners we're working with as well as to us for helping to promote that. And we hope it's going to help people deploy faster Web content sooner. We know speed and bandwidth is a key issue as people want to do deeper and richer sites so we hope by speeding up ISDN adoption we'll get to that faster. Yes.

FROM THE FLOOR: Yes. I was interested in finding out how your COM and Distributed COM initiative is working with the OMG and the CORBA work.

JOHN LUDWIG: The question is how is our COM and DCOM work, working with OMG and CORBA. I'm not the best person to answer that. I don't know if any of our COM friends are hanging on the side there. I guess I'd like to hook you up with a couple of our program managers over here who are involved in the CORBA relationship and they can tell you the current status of that. Why don't you catch me off line and I'll help you find the right people.

FROM THE FLOOR: Thank you.

FROM THE FLOOR: How does the Netscape plug-in architecture interact with ActiveX?

JOHN LUDWIG: How does the Netscape plug-in architecture interact with ActiveX? You saw that we demoed ActiveX Controls working with the Netscape Navigator and the way we can do that is taking advantage of their plug-in interface to allow controls to work inside Navigator. So that's one way. We use that so we can provide rich content within their navigator. In general, we believe the Active[X] Control interfaces are far, far richer than the plug-in implementation. They provide events, methods, property information about the control that you can't get through the plug-in interface today. And so the combination of this feature benefits and the fact that Active[X] Controls work in many many applications, there's already thousands of controls written in this interface, we think that's a better solution but we'll be happy to work within the navigator frame and using the plug-in interface.

FROM THE FLOOR: But plug-ins won't work in Explorer?

JOHN LUDWIG: No, they'll work in that too. We'll support them in Explorer but there will be feature benefits for doing a full Active[X] Control.

FROM THE FLOOR: My question is about the same. What time frame will plug-ins be supported in navigator?

JOHN LUDWIG: In Internet Explorer? We support plug-ins today with the current Macintosh beta. You can download that today and support that. On the Windows side we're not far behind. We've got the code working. It wasn't ready to release today but will be ready very soon for us to make available.

FROM THE FLOOR: We've heard about DirectX. We're going to here more about Direct3D. What happened to open GL? Is that chopped liver?

JOHN LUDWIG: No, open GL is not chopped liver. I think at our multimedia lead breakout session either later today or tomorrow we display our multimedia architecture and basically the way it works is DirectX is the lowest interface for highest performance access to the hardware, but then we'll put services like open GL so we'll have that layer on top of DirectX. People can write GL if they think that's appropriate. Other people might go to higher level services.

FROM THE FLOOR: But you are providing that means for providing open GL over the Web?

JOHN LUDWIG: Yes, we'll provide the Web to host open GL on DirectX and we'll support that.

FROM THE FLOOR: Thank you.

FROM THE FLOOR: The free seating strategy that you're going to do with the server and the—that developers can distribute the run times for the direct stuff is a great thing to get market penetration. It worked really well for Netscape. As developers, how long can we count on being able to distribute those run-time components with our applications?

JOHN LUDWIG: Well, our stated goal, I think, pretty clearly is to build all this active technology into our operating system products. And so today I make those run times available for free if you distribute because I can't drop operating systems products tomorrow. But that will be in the operating systems products and so it will still be free to you on operating system. Obviously I don't have that same flexibility on, say, the Macintosh platform or other platforms, but we'll figure out over time how to make that work. But certainly on the Windows platform, it will continue to be freely available, either as a redistributable or as part of the operating system. Yes?

FROM THE FLOOR: That was a very impressive demonstration of the cross-language support you had in Internet Explorer. In the demonstration version that you had here, were you actually running Java?

JOHN LUDWIG: Yes, that was a Java applet.

FROM THE FLOOR: So what you're handing out at the show, at the conference, the Internet Explorer has Java support already built in?

JOHN LUDWIG: I don't believe the thing we're handing out at the conference has Java support already built in since we just recently closed that agreement with Sun. So I think we're able to make it available on the disks today. But it will be there very soon. Now that we have our deal concluded with Sun Microsystems, we'll have it available very, very soon.

FROM THE FLOOR: So what you were demonstrating up there was slightly different than what you're handing out here?

JOHN LUDWIG: Yeah, the Java applet support is not going to get handed out here. Everything else is available that you saw up there.

FROM THE FLOOR: Thanks.

JOHN LUDWIG: Okay. No more? Okay. Tammy says no more so I'm going to move on and let her take control. Thank you.