

# Internet Development Environment

---

*Transcript of Tod Nielsen's presentation at the Professional Developers Conference.*

March 12 1996

Tod Nielsen  
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: [ ].*

**TOD NIELSEN:** Hi there. I'm not Bob Muglia. In fact, I'm Tod Nielsen, and the reason I'm here today is yesterday I was sitting in my office in Redmond and I got what I thought was an obscene phone call in the morning. I answered the phone and I hear, "Tod. Read your e-mail from Bob." And it hung up.

I thought that was kind of an odd thing to do because while Bob is my boss, so I try to read the e-mail he sends me, and then I found out that he has laryngitis and I'm doing this talk for him.

So it's been kind of a crazy 24 hours, as you could imagine, for me, and we've changed the talk just a little bit. We're still going to talk about tools but what's going to happen is I'm going to talk about our tools strategy and some of our Internet products, and then I'm going to bring Denis Gilbert down and he'll talk about our Visual Languages business unit. Denis and I spent some time over the last 24 hours looking over Bob's slides and decided to shorten them a little bit so they don't directly match what you have in your handouts.

We did that so we can get more time for demos.

The other thing I wanted to do is I have a habit or tradition of starting any presentation I do with the Tod Top Ten list, so I thought it was appropriate for the PDC to have Tod's Top Ten Cheesy Sites on the Internet. So, starting off number ten, the [Cyrano Love Letter home page](#).

For those of you who haven't seen this, this is the Cyrano server. I don't know if that's an ISAPI application yet but we'll talk to them. You can see it says if you're shy or unimaginative but want to tell someone how you feel, Cyrano is here to write the letter for you. So he or she composes the letter and sends it off to your loved one. He helps on both sides of the transaction, so he's a container and a server because he helps you write the letter and then helps you dump someone electronically.

(Laughter)

**TOD NIELSEN:** Okay. Number nine, the [Mr. Potatohead home page](#). Some of you have seen the Mr. Potatohead home page. What you can do with the Mr. Potatohead home page—let's maximize this—is you can click on your various facial parts and go ahead and Mr. Potatohead will activate it for you. So kind of cheesy.

Number eight, the [Perfect Present Picker home page](#). This is an interesting page that has some list boxes, and I'll scroll this up, and you can go ahead and choose the occasion, you can choose the profession of the person you want to buy a gift for, you can choose their interests,

you can choose their life, the personality, all the good stuff, and then go ahead and say let's see. Now the reason this is actually pretty cool, the reason it's somewhat cheesy is the same products show up no matter what you choose. You're going to get—

(Laughter)

**TOD NIELSEN:** So it's kind of creative marketing there.

Okay. Number seven on our top ten cheesy sites. The Brady Bunch home page. Now, the thing I like about this page is this is the unofficial Brady Bunch home page. I haven't found the official one, but all that Brady trivia as far as information and good stuff about the Brady Bunch can be found on this home page.

Number six, Captain James T. Kirk's Sing-along home page. Now, unfortunately, with the sound we've been having, I can't go ahead and sing along with Captain Kirk, but you can see this is an official home page on the Internet where you see our bud Captain Kirk and you can go ahead and click and he sings these various songs and you can sing with him and it's quite a site. You can see Scottie and all different information to sing along with Star Trek.

Okay. Number five, the Gumby home page. The thing I like about the Gumby home page is a couple things. One, if you have a heart, then Gumby is a part of you. Oh, that's nice. The other thing that's interesting is this particular person doesn't have an official Gumby license. Now, the Microsoft Developer Network doesn't have any information on how to obtain a Gumby license, but if you'd like to know, we'll try to find out for you.

Number four, the Disco Web home page. This is a great page in that—

(Laughter)

**TOD NIELSEN:** -- everything you want to know about disco can be found on this page, including you can download a virtual disco ball and I love the tagline here, it says 470K impact file. Could there be a bigger waste of bandwidth.

Number three, the Tod and Allison home page. I'm getting married this May and of course I work on Internet stuff; I had to make my own home page. We're getting married in Hawaii and the original intent of our home page was to give people information about coming to the wedding and hotel information and all of that stuff. The scary thing I'll warn all of you about is when you put a wedding home page on the Internet, we have right now 131,000 people that think they're coming to our wedding.

(Laughter)

**TOD NIELSEN:** Number two, the Top Ten Ways to Tell if You Have a Sucky Home Page home page. This is something all of us that are creating home pages should take a look at. They have a list of all the things you should do. For example, if your header picture is over 50K, you have a sucky home page. Or number nine I like, if you have a 400K picture of yourself but you only appear in the lower 20K, that qualifies as sucky.

And finally, the top [of the top] ten cheesy sites on the Internet, according to Tod, the How to Toilet Train Your Cat home page. Now, this could be actually pretty useful. The thing I like about this page is there are included pictures.

(Laughter)

**TOD NIELSEN:** And the other great thing about this page is that at the very end, you see he's successful and you go, "Oh, great." But at the very end the guy goes "hire me." I mean, okay; I'll hire you.

So anyway, that's our top ten list.

(Applause)

**TOD NIELSEN:** Now I want to talk about our development tools strategy. And the first thing I want to say is we have a tool strategy, and that tool strategy is to build incredible tools so that people can build active applications. Our tools are going to let developers span the desktop, the server and the Internet. We believe that these tools are going to empower or make the active platform become dominant and become everywhere on the Internet and on PCs.

Second thing is we believe in an open environment for third parties, and this means two things. Number one, since we want our development tools to be best of breed, we want every other tool vendor out there to build even better tools. We want you to kick our butts and build development tools that support the active platform so that the active platform can become everywhere. And number two, we want to provide in our development tools, in our environment, a way for you to plug in or put designers to integrate into our development environment. So if you have a particular designer that edits photos or does multimedia or some interesting design aspect, it will be—as part of a full integrated tool, we want to provide the interfaces for you to do that and the active interfaces and active platform will allow you to do that.

Number three is we want to support the Internet wave. Of course we want all the users of our tools to be able to take advantage of the great things happening in the Internet. We also want to protect your investments. We don't want to say well, 1995 and before, the investment you made as developers in our tools, forget that. It's now the new world. We want to take the investment, the millions and millions lines of code that you put using Microsoft development tools and bring that forward so you can take advantage of the active platform.

Next we want to provide a scalable solution so you can start off building a local application on the desktop, and that will scale to a full client/server application and multitier application so if you want to build a three-tier application, all aspects, the client, the middle tier and the client code will be able to use our tools. Finally we want to integrate to get you to some of the legacy environments or data that may be in your solution.

Another aspect of our tool strategy is MSDN or arming developers with information and products and betas and all the stuff you need in training on how to use and be successful using Microsoft development tools. So today I want to announce the ActiveX™ Development Kit that's going to be sent to all MSDN Level 2 and Level 3 [members] after the conference, and what you're going to get in this kit is the Internet [Information] Server, the ActiveX SDX, the Explorer and all the stuff listed on this slide and this is something we'll continue to update and enhance over time, so there's more software and training material is available we'll send it out to MSDN Level 2 and 3 [members], as well as of course you can get it on the Internet.

So Microsoft development tools, we have a full spectrum of tools for supporting the active platform. On one end we have products like Office and FrontPage for supporting a more document-centric development environment where you're just building documents or building pages and there's not much coding involved all the way up to our programming tools and Visual C++ for building full-blown applications.

So this is a slide we've seen and I'm going to talk through now just how our tools support the active client. Obviously, FrontPage is an HTML editor in Web document tool that I'll talk about

in a few minutes and with FrontPage in this scenario, you can see I've HTML content running in our Internet Explorer.

You then with Internet Studio and Jakarta—Jakarta is our code name for our Java technology at Microsoft, Visual Basic and Visual C++®—you can create Active[X] Controls and these Active[X] Controls can be hosted in the Explorer like we talked about. Then of course with Visual Basic® Scripting [Edition] or VBScript you can do Active Scripting to do visual automation and integrate these controls.

And then finally, with our Office product or any particular tool that creates Active Documents, you can host these in the explorer to build full-blown applications with any document format, be it XLS or whatever.

On the server side, we want to provide the tools so you can take advantage of our server platform that "J" Allard talked about. Starting off on this we have FrontPage and Internet Studio, and in this case it's storing those particular documents or those Web pages in Web pages in a wage file storer. Products will help you create CGI applications or you can do the in-process high performance ISAPI applications or you can build things with a database connector which will allow you to get to any ODBC database and all tools, Visual C++ and Visual Basic will allow you to do this. And then throwing Jakarta into the mix or our Java technology, you can put your business logic or tools into your server application. Then, just like on the client side, on the server side we're going to support Active Scripts so you'll be able to script and integrate all this together.

I want to do something on cross platforms since people think Microsoft just does Windows tools. We support Windows 3.1, Windows NT; FrontPage (we're going to take a look at in just a second) runs on Windows 95 and Windows NT as well as a future version will be in the Macintosh.

As far as Internet Studio goes, I'm going to talk about that in a few minutes; but it's hosted on Windows 95 and Windows NT but it will be able to run everywhere that the Internet Explorer supports.

Visual Basic and Jakarta, they're hosted on Windows 95 and Windows NT. Jakarta and V[isual] C++ is hosing an Windows 95 and NT and targets Macintosh.

One of the things at Macintosh I get called a lot is I'm also responsible for Access, and I'm sort of the "what is the difference" guy. By that I mean I answer the questions: What is the difference between Access and FoxPro, what's the difference between Access and VP and SQL server? So I get asked this question a lot, so I should answer the question, "What is the difference between FrontPage and Internet Studio?" And if you look at [today, it's] the wedding page and how to toilet train your cat page, all the way up to the Dodge database that Steve Jobs showed you this morning. And what FrontPage does is look at the Web as a series of linked documents and makes it easy to look at these documents and link them together doing stuff on the client side as well as stuff on the server. And it's sort of a complement to the Office technology. So Office has documents like Excel spreadsheets or Word documents and it's treating a Web as a collection of documents.

Now what I'm going to do is have George Bing come up, who is the program manager on FrontPage, and give us a quick demo of FrontPage.

**GEORGE BING:** What I'm going to do is give you a brief demonstration of FrontPage which we think is the fastest and easiest way to create Web sites. Last night in thinking about marketing this product, I did an informal survey and we really have some formidable competition, in particular one which I think you know what I'm talking about.

**TOD NIELSEN:** That would be the Visual Notepad.

**GEORGE BING:** That's right. It seems to be a very favored choice of editors. What we have with FrontPage I think is a great alternative for creating Web sites and so what we're taking a look at right now is one of the key elements of FrontPage which is explorer. It comes through in a combination of HTML as well as visual site management. And we think it's really the combination of the two that allow FrontPage to really deliver a great solution for creating Web sites.

So we're taking a look at, as I mentioned, is the FrontPage explorer and this provides a top view of the contents of this Web site. So you can see it's a traditional explorer view, and I'm expanding it out, taking a look at the hierarchy of the Web site and then on the right-hand side I have a view that is showing me graphically how pages are linked to one another and how the links are going from one page backwards and forwards.

I can switch the view to a summary view, and this gives me literally an inventory of the different elements that are contained within my Web site. So if I want to change a filename, I can do that within this view, and then it will automatically update and allow me to sort of reverify and connect the links.

Now, in order to really boot-strap people's creation of Web sites—

**TOD NIELSEN:** The magic—we didn't wave over the machine there.

**GEORGE BING:** We will try this one more time.

Now, as I was mentioning before, in order to boot-strap people's creations of Web sites, FrontPage provides a series of Wizards or templates that people can use. So if I want to create a new Web, I simply choose New File, New Web and I have here a series of options that are available to me for the normal Web, corporate presence wizard, customer support Web, and these are easily created with using Visual Basic or Visual C++, and you can create them yourself to be included with FrontPage if you have an infrastructure that you want to share with others.

So the other element of FrontPage is the editor. And so what I brought up here is one of those pages that would be set up by one of the Wizards, and here as you can see it's a pretty basic page. So I want to add some flash to this, and let me show you how FrontPage automates and really makes easy some of the common things that you want to do with Web pages. So what I want to do right now is put a background here so I bring up this dialogue, I'm going to browse for an image, include this from a file, and so on, include that image, and it's not quite the disco background image, but I think it will do here. Let me show you how I would insert a bitmap, and so I want to insert an image, and again I'm going to go from File and I want to look for BMP. So in this case I just want to enter this. Now I brought in this image but I have this green color which is obscuring the background so what I want to do is make that transparent and I can easily do that by selecting this tool bar feature and then make that transparent.

And so now I can see into the background. So as I mentioned to you before, one of the things we're really trying to do is take some of the common tasks that people have with creating Web sites and Web pages and make them very easy. So one of the things we've done is really automated and made it easy to create image maps so I'm going to select a rectangular tool here and I'm going to make this an active spot on this image.

And so what happens is it automatically brings up this dialogue and I want to associate it with an HTML page, and so I know which one I want, so let's see, I'm going to scroll down here. And then quickly, I very easily created a hot spot within this image. For those of you who are

really into sort of figuring out the coordinates, this may not be the thing for you, but we think it's a really great alternative for making that process very easy.

One of the things that we've done also is enabled users to incorporate functionality within their Web pages which normally would require some CGI programming on the server side so what we have is something called Web box and what Web boxes are, they're dropped-in functionality and provide me with a variety of things I can drop in a search Web box which allows me to do a full text search across my Web site and if I'm hard core about it and I want to incorporate some HTML into it for something for FrontPage doesn't support, I can also use Web box to include some HTML directly into my page.

So let me go ahead and incorporate a search Web box, and what I want to do is set the parameters to return to me document size and then I'll click okay here and so, boom, I've incorporated the search which frees me from having to figure out what to do on the server side and client side as well.

Let me show you what happens on the overall, I'll bring up Internet Explorer here, I'll bring up Internet image and this is a hot spot, so I'll go out to What's New, and then just to show how you the searching mechanism will work, search for a specific coffee blend, Arabica is one, and then what happens is it returns to me a table with both the title as well as the file size. So if I go ahead and click on that it will bring me directly to that page. So that's just a brief look at FrontPage.

**TOD NIELSEN:** That's great. Thank you, George.

(Applause)

**TOD NIELSEN:** Now I want to talk briefly about Internet Studio. The reason we're talking about Internet Studio is I'm not announcing the product today and we're not going to go into a bunch of detail but there's been a lot of attention on the product and I want to give people an update on where things stand and what our vision is with the product. Internet Studio is going to be for developers who want to build active applications. We're going to support professional layout; some of the things this tool supports was demoed in the sessions this morning when Chris Jones and Thomas Reardon showed the browser supporting multiple frames and cool layout designed with HTML that can be designed with Internet Studio. As Steve jobs called the active HTML or supporting the active server platform is going to be a key platform. So it will design not only the client side technology but also stuff on the server. And then of course integrated scripting. There will be an environment for you to write all your scripting code. In fact, Internet Studio is going to be hosted in the Visual C++ shell so in that tools environment it's going to be a set of components that fit into that shell and will be integrated so you'll be able to have the same way you integrated with MSDN and V[isual] C++ and that stuff.

We'll have the ability to build active certify servers and easy drag and drop, support for site design, all the database integration and E-commerce. And we realize in many cases there is more than one developer building these Web sites and we'll support the ability for multiple people to check in and check out the pages and code, et cetera.

Nine years ago I was the product manager of a product code named Omega which eventually became Microsoft Access, so I have a unique experience base of working with products that refocus and transition. And so I want to briefly talk about [Internet Studio] Blackbird and what now has become Internet Studio and what changes we've made. First, and probably foremost is our focus on HTML. We are now fully supporting HTML and we're working aggressively with the W3C to provide leading age HTML and we'll demo some of that in a second.

Second is we're targeting the active Internet platform. If there's one thing you're going to take away today it's active, active, active. Our tools are going to support it, we'll take advantage of that platform. One thing we are keeping and going to bring forward from Blackbird is the application model, the model we had for project management and creating pages in Blackbird. We think that was a good model for building these apps and we're going to bring that into Internet Studio as well as maintain some of the high-end design, the ability to have overlapping controls and text on top of each other, we're going to maintain that and do [it] with HTML using the CSS style sheets.

I talked briefly about some of the Windows options and some of the frame options that were demoed in Internet Explorer. Again, Internet Studio will support the development of that.

So what I want to do now is have John Shewchuk come up with give us a brief demo of Internet Studio.

**JOHN SHEWCHUK:** So it looks like we're on the wrong monitor? Okay. That's right. What we're looking at is very, very early code that's going to be incorporated into Internet Studio and in fact during the last six weeks we have been working to get this up and running as a result of refocusing. So I'm going to show you some basic stuff which is that I can type into this, do things like set bold and italic. So we're looking at a basic HTML editor. But where things get really interesting is when we go and we open up the tool bar, and we can start doing things like dragging in controls.

Now, again, this is pretty early code so for the most part this seems to be working but every now and then we'll get a bug.

So these are controls. We can look at the properties and we can see that there are things like button characteristics, transparency. And when we're using Internet Studio, a lot of times what we're going to be doing with it is building complicated applications, which is a little bit different model than what we have with FrontPage where the focus has been on doing things like documents.

So this scroll bar could, for example, fire events when the thumb is moved and so on. Similarly, we've got a text box here, and we can set properties like the font type, so if we go down and say the font should be larger, we can say, okay, and then go into this and here in WYSIWYG mode, this is a task test.

So what we've shown is plain old HTML editing with but with the addition of in-place Active[X] Controls. Make sense?

**TOD NIELSEN:** Great.

**JOHN SHEWCHUK:** So what we're going to do now is take a look at what we saw earlier today which is when we start doing some of the more high-end development: Wants to defrag my disk. Okay.

So as we saw before, this is the HTML editor that can support the more professional layout capabilities such as overlapping and so on. So what we see here is a—is an object that's sitting on the screen and I can do things like resize this object and you'll notice that the transparent text in the background picks things up, or, for example, I can bring this over and move it to the front of the "Z" order, and it's gone and it's put the transparent text in front of it and so on. So we can support those more advanced layout capabilities that are coming in the CSS style sheets.

As we showed earlier today, we can take elements and we can drop them on the toolbar, and when we drop elements up onto the toolbar they become templates. So if I now wanted to go drop a whole bunch of these things on the page, I could go down here, for example, and add the Volcano button, and let's save that out, and look at it inside of—oops—inside of the Microsoft's Internet Explorer, and let's refresh that page. And we can see the new objects. Now, you'll notice that as I go around and change things, these buttons are all active, but the button that I just added isn't active. So what I need to do is I'd need to go make some changes. So let's go take a look at the source code and we're using everybody's favorite editor here, Visual—Visual Notepad.

**TOD NIELSEN:** Yes.

**JOHN SHEWCHUK:** And let's do some things like change this to be a VC icon and we'll change this to be the BC button and we're going to go up and change one more thing, which is the idea of this thing, the VC button.

**TOD NIELSEN:** You need the Visual Notepad debugger there, I see.

**JOHN SHEWCHUK:** Which is exactly what I wanted to talk about. I hope I get this right. Sometimes I make mistakes. Let's reload that and as you can see we're going to support some simple kinds of debugging in this but where things get really interesting is where we pull this together with our other tools, so for example, we're going to be incorporating Visual Basic Scripting [Edition] inside the development environment and Internet Studio will be brought into that same environment so that we have a way to take all of our languages, all of our development tools, debuggers and put them all together. So that's pretty much it.

**TOD NIELSEN:** Thanks, John.

**JOHN SHEWCHUK:** Okay. Thanks.

(Applause.)

**TOD NIELSEN:** Okay. We talked about FrontPage and Internet Studio. And now I want to bring Denis Gilbert down to talk about our visual languages.

**DENIS GILBERT:** High five. Thanks a lot, Tod.

My plan this afternoon is to blast through a few slides, give you some overview of what to expect over the course of the next year in our visual languages products.

Now, some really exciting demos we put together incorporating all of our language technologies and the Internet. And just to give you an idea of what kind of exciting sort of captivating things you can do with tools like Visual Basic and VBScript and our Java technology and our [Visual] C++ technology.

Let's start with Visual Basic. You know, we've never actually showcased Visual Basic at this PDC. So just for my own edification, can I just get a show of hands for who use V[isual] B[asic] in your sort of day-to-day application development, please. Looks like about ten, 15 percent. So hopefully next year we'll get that up a little bit more.

One little known fact about V[isual] B[asic] 4 today is that you can already build ISAPI services with V[isual] B[asic]. The Internet Server technology kit includes a piece of technology that lets you take a V[isual] B[asic] 4 remote automation OLE server and actually run it off the back end of an HTTP server using ISAPI. This means that you can basically start treating the Internet server as your application server for these three-tier apps that you've been building with



VBS[cript]. That's really nice that you can preserve that logic and code you've already written against OLE.

Another thing that VB is also very famous for is a wide variety of third-party controls and components that are on the market. And we're actually pleased to announce this week an open beta on the World Wide Web for a new set of 32-bit Internet controls specifically for V[isual] B[asic] and all of the other 32-bit tools that support V[isual] B[asic] controls. The Web site is up here on your screen this. Went live last night. All of the documentation. The product is out there, the reference material, the sample code. Everything is up there. So it's a wide-open beta for these new controls.

There's an HTTP control, an HTML control, there's an SMTP and a pop control for doing mail-type things. An FTP for file transfer, WinSocks control, and finally there's an NNTP for doing news group management from inside V[isual] B[asic]. You can take your existing application, plopp this control in and you can take advantage of a lot of things out on the Web and start building in some pretty sophisticated functionality.

We're working on a new version of V[isual] B[asic] called V[isual] B[asic] 5 and we expect to have it out before the end of the calendar year. And two of the very related sort of ActiveX Technologies that we will be supporting is the creation of document servers, Active Documents with V[isual] B[asic] and the creation of Active[X] Controls with V[isual] B[asic]. And there's no magic involved. It's going to be about as straightforward as saying save as to take any form-based app and actually turn it into an active object, an active document running inside an active object—or active document container like the office binder, like Internet Explorer, and in fact I'll show you an example of that very thing in a couple minutes. And I'll also show you some examples of V[isual] B[asic] 5-built OLE Controls.

Visual C++ just released to manufacturing at the end of February a new version; units are being [released] today in batches of 25,000. It's the first subscription update to Visual C++ 4.0. This version is called 4.1. It includes new classes in MFC to easily create with the use of wizards and the like. ISAPI servers, ISAPI filters. There's some great sample code there that you can use. In fact, that's another example of a piece of technology we'll be demoing in a second. And, of course, Visual C++ is fully compatible with that same Internet control pack that we're starting the beta of this week.

Visual C++, over the course of the subscription releases, will include gradually more and more support for the active platform. And that'll be in the form of being able to create Active Controls and Active[X] Control containers and the ability to create Active Documents, both the actual servers and containers for those documents, and I'll show you how that's going to be phased in over the course of next year.

We'll also have full MFC support for all of the Sweeper APIs. Going to be derived off—it's going to be natural to use if you're an MFC programmer.

If you're already creating OLE compound documents with MFC, it's like a couple of lines of code and you're well on your way there. Not only that, but MFC version 4.1, the version available today, includes a version of the scribble signal, has all of the MFC code you need to create Active Document servers today with your MFC app. That library of code which is provided as a sample is going to be built into the next version of MFC as sort of a core part of the library, so you can just sort of click on a button and create one of these Active Documents right out of the box.

The Web site that outlines all of the new 4.1 features is going to go live tomorrow, Wednesday. Should be up there around noon. The address is—[microsoft.com/visualc](http://microsoft.com/visualc). The address again for the control pack is [microsoft.com/icp](http://microsoft.com/icp). And that's Visual C++.

Again, Visual C++ is on this subscription release. So it ships roughly in the February-March time frame, roughly in the June-July and roughly in the October-November time frame. So today at 4.1, we have an exciting bunch of new controls, MFC support for server side objects. For Microsoft's Internet [Information] Server we have some sample code that lets you get started creating Active Documents today.

And in the second release, the 4.2 release if you like, we'll have full MFC support for all of the new sweepers and APIs as well as the ability with basically two lines of code to turn any MFC-built OLE Control into an Active[X] Control.

And likewise we'll have incorporated the tools you need inside MFC to create those automatic document servers. The third release of the year, the one that's slated for sort of October-November time frame will have class library support for creating document containers, Active Document containers, in your own application.

I'm also pleased to say that given that we've just released 4.1, we also cut an alpha release of 4.2. So the MFC version that has all of the advanced support for Sweeper is available to show attendees. And we have CDs at the Visual C++ booth on the trade show floor here this week for anybody who's interested in picking up sort of a leading-edge version of MFC, notwithstanding the fact that 4.1 has recently been shipped.

One thing I think that everybody at this conference is anxious to hear about is what's happening with VBScript. Take note of the address at the bottom of your screen. This site went live last night at about 10:00 p.m. Everything you need to know about VBScript is up on the World Wide Web, including the ability to download VBScript. The reference manual, the samples, the tutorials, the code is all up there as we speak.

There's a lot of nice features about VBScript. It's small. The actual language run time is only about 45K. It's safe because it doesn't have any built-in intrinsically damaging intrinsics. It's a proper subset of the Visual Basic and Visual Basic for applications language. So you effectively have close to three million trained programmers out there who already are proficient in VBScript. And very exciting for us is it's going to be built into Internet Explorer 3.0 using this new Active Scripting API that probably was outlined to you earlier today. Again, VBScript will be showcased in a demo in a couple of minutes. It's going to be built into all the clients, all the servers. We're doing the Windows platforms ourselves.

And also, importantly, did I—I probably forgot to mention that this is free. It's not like a 90-day trial period, then you pay. It's not a \$2 a box royalty. It's not pay in a year. It's absolutely free. There's no cost involved. It's exciting technology. You can incorporate it into your apps. It doesn't have to be an Internet browser type of solution.

Then a great thing about VBScript is you can script any COM object so later on you'll see that we're actually scripting today Java applets running inside the Explorer. We can script COM objects, Active Controls. So that's incredible power. The way that we keep the sort of safety thing protected is that we only have access to objects that are exposed by the container in which VBScript is running. So it's sort of like Internet Explorer decides what is safe for VBScript to access. And using code signing or a number of other technologies, it lets us basically have the best of both worlds. It runs on the client, runs on the server.

And also, the source for VBScript is going to be available by the end of the month on that same location that I just gave you. So not only is the run time free, the source code is going to be up there.

So there you go. That's VBScript. And we'll see that again in a second.

Today, we hinted at an internal project inside Microsoft, [a] sort of umbrella name for the project is Jakarta. And it—for those of you who are geography majors, Jakarta is the largest city on the island of Java. It's the capital of Indonesia.

And the code name, it's not a product name, it's just sort of an umbrella name, represents the work we're doing on the development tools and the work that we're doing in the run time, sort of the virtual machine. We've incorporated great, great development tool support for the Java language technology inside that developer studio IDE that we share with Visual C++ and test and Internet Studio and all of the others. So all of the features you've come to like: The great visual resource editing tools, source code control, syntax coloring, visual project management, integrated bills, all of those things are already in there.

Great visual debugger based on the Visual C++ debugger. The other thing that's really exciting is we have our own dot Java compiler. Don't confuse this with the JIT compiler. This is the compiler that—supported inside the development tool. Today, on a sort of a high-end Pentium, say 120 megahertz, this compiler compiles at a rate of about a million lines a minute. So it's pretty fast.

We have our class view technology for browsing, to help you get quickly started writing your first app, a standalone Java applet or inside a Java container.

Switch over to the Internet Explorer, where we've incorporated the support for Java run time and interpretation; we're using the active scripting interfaces, so it's ported the same way that JavaScript and VBScript are supported.

And we've added two very, very powerful features to Java. One, any Java applet can access any COM interface, and it looks just like a Java class. You don't have to do wrappers. You don't have to do any new upper classes to get at that stuff. So anything that's sort of a COM object, an Active[X] Control, another sort of object is immediately available to a Java applet, running on our reference implementation. And it looks just like another Java class. You invoke methods, you set properties. I'll be showing you an example of that in a second as well.

And vice versa,—sorry, Java classes themselves are exposed as COM objects automatically by a virtual machine. This means that VBScript and JavaScript today can script Java applets running inside the Internet Explorer. This is unique to Microsoft. The only other related technology I've ever seen that shows how to do that is in white papers. And today we'll actually show it to you running.

We have a built-in JIT compiler inside the virtual machine. The rate of compilation of this JIT compiler, this is the one that takes the byte codes and converts to native, is close to two million lines a minute. I'll just give you an idea, we are doing some real serious tools here.

So let's drop the slides. I'm going to go over here and show you the way-amazing radio station Web site.

How are we doing on time? Got about 20 minutes. If you don't mind, I'm going to run about five minutes late.

I just want to show you this site. And this is sort of [a] pretty cool site. Way amazing, 96.1 on your [dial], plays traditional rock, '70s, '80s, some '90s and they have their own Web site; this is using HTML. What I'm going to show you after we browse through this Web site is this same Web site created using all of these new language technologies that I've just described on slides. So VBScript and Java technology and Visual C++ and Visual Basic and the like.

So the first of the four places you go in the site is a list of concert listings. But it's kind of dull. First of all, it's not interactive. I can't ask for all the concerts in Montreal and I can't see what Chicago is playing during the month of December and it's not very interactive. So we have ways to fix that up later on.

The next location is the top 1,000. This is the top 1,000 requested songs that the radio station has played; because this is displayed as HTML, the back end server has to actually construct this page, you know, 1,000 lines into this HTML page and send it over to my client. So it takes a long time to actually create a page that size and download it.

And, again, you know, even though we use tables and it looks nice, it's not very interactive. And it's certainly not dynamic in its appearance.

Now, the third section in the way-amazing radio site is the video library. And you've seen these before. And I'll show you how we can use COM technology and Java and VBScript to make this incredibly exciting. And of course there's the traditional job opportunities. And the people who work in the personnel department at way-amazing have a little problem, because they usually use Word to do their documents. And they're trying to switch over and they're not sure if they're going to use FrontPage or Hot Dog, or Notepad and they'd really just like to continue using Word because that's what they do their job opportunities page as.

So let's bring up the new way-amazing radio station home page.

Now it's over here.

Now the first thing that you'll notice is that we have an OLE Control, an Active[X] Control in the middle of the page. This is a control written with Visual C++ that shows you what's happening at the radio station. Right now they're playing commercials. I expect in a second or two they'll start playing a song and it's going to scroll up. Let's go to the job opportunities page again.

Click down here.

Now, what happened? The personnel department is still using Word. They didn't have to get retrained with HTML. This is a Word document. In fact, it's running as an Active Document. So it does tool bar negotiation. So I have to tear off toolbars and all of the features that you'd expect out of Word, but I haven't left the Internet Explorer. In fact, the About box is about Microsoft Word. But if I look at the tool bar, I can navigate back to the first page and then forward again to my Word document. So I've completely integrated my Office document into my Web site, but I haven't had to retrain the people to do so. So that's a really nice feature of these Active Document servers.

Now, that said, it would be nice if you could use maybe some of your development tools to create your own document servers like that. So what I'm going to do is go to the new Talk 1,000. And let's see what we have running over there. I don't know if I clicked that. Is that running?

Here we go.

Now, what you see here is obviously it's a lot more hip than the last site. But this is actually a V[isual] B[asic] application built as an Active Document hyperlinked from that—that home page, and this is actually a V[isual] B[asic] application, I can list the top 1,000 songs. And I've got, like, a lot more sophisticated things, 'cause I have all those V[isual] B[asic] controls that I can use.

I can select things. I can, you know, move borders around. I can ask for a band by name. In fact, I can—let's see, I can go up here, maybe search for, say, the word string held inside one of the bands. And just to make the point that this is really V[isual] B[asic] running, I've set a break point on that search button and I've broken right into V[isual] B[asic] 5 development environment. And you can see that on click, I'm constructing a sequel statement and I'm passing that to remote data object and I'm getting the results of that query. And then displaying it inside my grid control.

So, again, we've used an office document. We've incorporated that into our Web site. It's really nicely integrated. And we've taken a V[isual] B[asic] form-based application, we've done the same thing with the V[isual] B[asic] form.

Let's go to upcoming concerts. And we'll show you an example of OLE Controls or Active[X] Controls built with V[isual] B[asic] 5.

Now if you remember what this looked like, this was just sort of a list of bands and where they were playing by date.

And now I've got a calendar control, which is built with V[isual] B[asic] 5, and I can just click on a date on the calendar control and it'll give me a list of all the bands that are playing that day. And I can double-click on this control over here and I can change that to get a different view on the concerts for that event.

Now, again, this is an HTML page, unlike the document server we saw in the previous demonstration. But this one is actually running these controls that were built with V[isual] B[asic] 5. And I'm using VBScript to coordinate between the controls to script the controls so I get the date out of the calendar control and I pass it to the concerts by date control to get the list. And I've also built in some support for doing business-related things. So let's see, I'd like to order a ticket for that concert.

I could fill in the tab control over here. Let me type in an address. And then I think we do zip validation, so I need a valid zip. Okay. Let me go to my credit card. Now, fortunately, this demo doesn't accept American Express. So I have to go to Visa. I'm serious. Somebody actually hard-coded that into the VBScript.

And put in an expiration date. Again, VBScript is handling all of the validation of fields, the verification. It puts up an out of—out of frame message box here, confirming that my credit card's been validated. And let's say I want to order two tickets. And I place the order. Now, what that's doing is actually sending the order to an Internet Information Server, and what I've done on the Information Server is I've hit a break point inside that ISAPI DLL, which is handling the business transaction and I've just hit that break point now and I have Visual C++ running on the server.

And this is the break point inside the method that's handling the do order command from the client. You can see that we're using some of our new HTML classes to handle the order. This is very straightforward. If you scroll down, it creates the command, does a co-create instance, buys the ticket you know, formats the results, expands the template for my screen, and then just goes on.

So let's just execute this, go past the break point, and switch back to our client machine.

And we're unable to process the order for the following reason: The system is hosed.

If the demo had worked that -- I would have gotten a map of the concert stadium, telling me where my seat was. At least the people in the back of the room have a sense of humor.

All right. We've seen how you can take off documents, link them into a Web site, we've seen how you can take form-based applications with [a] tool like Visual Basic and actually have them be an integral part of a Web site. Let's go back and see a couple more things that you can do. Let's see. So that was the upcoming concert. Let's go to the video library.

Okay. Now, this is an interesting page. We're going to spend a little bit of time on this page. Again, this is an HTML page. The control up at the top left [is an] OLE Control built with Visual C++ and with MSC. Down at the bottom there's actually a COM object that plays videos and it's actually set up to be scripted via Java. Let's play ZZ Top. Can we get sound on this, guys? Maybe we don't want sound on ZZ Top. We're playing AV movies directly on my HTML page. Let's have a look at some of the codes behind the scenes making this happen.

What I'm going to do is switch over to my Jakarta development tool. And what we have here is the HTML for the page that you just saw. When you click on a play button, the code that's highlighted in yellow is the code that's executed.

The parameter is a property out of the OLE Control. It's the text selection for the band name. And then the method here is actually a method inside a Java applet. And so from VBScript we're extracting the OLE Control property that is the band to play and we're passing that to the Java applet. Again, no blue code required. It's just a one-liner, and then we call another Java class that does—that instructs it to actually start playing the video.

So let's take a look at that Java code behind the scenes. I'll use the class browser over here, class view, and I'll look for the select band implementation. And again, because it's public, it's exposed as a COM object through scripting tools like VBScript. If you look at this, it looks like a declaration of a Java class. Actually, cobandrecord is a COM object, and that becomes available in my name space over here with this import command. Now the syntax for this might change, might be something like, you know, import com.ms.cocreate band. But what it takes is takes anything that's registered itself in the database and makes it instantly available to your Java applets as a Java class. So subsequently, when you go down here, you can just declare an instance of that class, you can call all the methods inside it set the properties and it works. That video player we saw was a COM object built with V[isual] C++ and we were just firing it off using the Java source.

Let me show you how easy it is to create your own Java applications. Let's create a new one from scratch. Say file New, Project Work Space.

Now, this particular developer's studio has both Visual C++ and the Jakarta tools installed. I don't necessarily have to have both together. Let's create a project called Denis—there's already a Denis, but that's all right. It's polite as usual. It asks you if you want, yes, please, no thank you. It asks if you want to run as a stand-alone application, source code comments, do you want your app to be multithreaded, do you want it to run single-thread, what size would you like the default window in your HTML page to be for that applet, would you like a sample HTML file generated for you so you can see how to invoke the Java applet from inside the tool like the Explorer. You bring in appropriate classes and definitions and interfaces from AWT. Does your application react to any of the events that are fired from AWT?.

And this is a really nice one, this lets you set up the parameters that you want to pass from HTML to your Java applet. It's kind of tortuous to remember because the applet tag is not a variable-length tag. You have to actually insert all these param tags. So I could have one called Denis, another one called Fred, and I could just add those as—to my HTML applet invocation string.

And then we see finished. And the apps built—it wouldn't let me save it into the same directory. Let's go back and pick the one we just had a second ago, and just to show you the speed of

compilation, I'm going to build this file. It's done. Point nine. Let's try that again. Build. Point two. Let me do a rebuild all because it is kind of fast.

Oh, sorry. It was at the bottom, it took five seconds to rebuild the thing. And these are all the files that make up this particular application. So as you can see, it's really a very rapid application development tool using this built-in native Java compiler.

That kind of summarizes where we're going with VBScript with Visual Basic, with integration of, you know, great Internet features using tools that you have today, tools that are already on [the] horizon like the Internet control pack, and forthcoming versions of Visual C++ and as this active platform starts materializing, how the investments you're making today in our tools will just give you immediate payback so you can have that competitive advantage.

So enough on the visual languages. Back [to] Tod.

**TOD NIELSEN:** Thanks. I just want to summarize what we were talking about today and what the call-to-action points would be. First and foremost is everyone create ActiveX applications. We want active applications all over the world, and our development tools, FrontPage and Internet Studio, and V[isual] C++ are going to help you create that application but we want everyone using the tools to build active applications. You should build Active[X] Controls. You can see we've exciting ways to use these controls and how to build them and we encourage everyone to do so. Create Active Documents. These can then be used within the Internet Explorer and you can include them to build full-fledged applications for the user.

Second is activate your existing applications with VBScript. You should start now to look at VBScript and realize how you can automate and do the integration as well as extending the application with VBScript.

Finally for tools vendors what we want to do is target ActiveX. We want you guys as we said earlier to kick our butts. We want you to build better tools than we're going to build. We're committing to building best-of-breed tools. We want tool vendors to build awesome tools for the platform.

Also we want you to work with us to extend our environment so we can plug in your designers and integrate them into our environment, so we can provide a solution to developers that has a common environment with Microsoft components as well as third-party components.

And to emphasize our commitment to openness and working with tool vendors what I want to do now is introduce Mark Benioff, who is the senior vice president of Oracle Corporation, to come and talk to us about Oracle and its tools.

(Applause)

**MARK BENIOFF:** Okay. I guess I could relate a little bit to Steve Jobs this morning when he was saying this is a little strange. Usually Oracle and Microsoft are in very competitive environments, but there's a lot of change in our industry, and we're working very closely to support that change.

Today what we'd like to do is show you some of the things that we're doing with the ActiveX Technolog[ies], VBS[cript] and some of these new Microsoft technologies, so if you'll join me over here and give applause to Jeremy Burton and Jon McCormack from Oracle.

So Jeremy, it looks like you're running one of our tools here, Oracle power objects, which is a client/server application tool, runs cross-platform, and it's got the basic scripting language behind there; is that right?

**JEREMY BURTON:** That's right. We've heard a lot about the Internet today and I'm a client/server guy really, so I thought we'd start with something I'm interested in, something which a lot of people out there are interested in and probably what they're doing today. So this is borrowing old client/server. And what I've been doing—

**MARK BENIOFF:** These are legacy systems.

**JEREMY BURTON:** Some people may argue they've put a lot of time and money in these so they wouldn't refer to them as that.

What I've been building here is a standard application, it's a database application. And what I want to do is just finish this off before we move on. I'm going to just add a control in the same certain way you would in a lot of tools these days. And what I want to do is put some information from a database on there.

**MARK BENIOFF:** Oh. Live software. Why don't we come back to you, Jeremy. How does that sound.

**JEREMY BURTON:** Sounds pretty good.

**MARK BENIOFF:** Why don't we come over to Jon. Jon, how is it running?

**JON McCORMACK:** I think better than that. You're running Power Browser which also has the basic scripting language in it [and] can be a container for ActiveX Controls; is that right?

**JON McCORMACK:** That's right. And what we're going to see now is—everybody knows pretty much what a browser does and everybody knows what a server does. What we're going to show you is how you can use the Internet and the Internet type technology within your organizations to help solve some of your data management problems. So what we see here is Power Browser. It works like Netscape, it works like Internet Explorer, it does browsing. But a problem that I've got in my organization at the moment is I've got this Excel file sitting on my PC which contains a list of product specialists; okay? So there are people who know about how to use Power Browser and how to use power objects, and I find it really hard to publish this because every time somebody wants to get information out of this file, they've got to either call me on the phone or they've got to physically log onto my PC using file sharing and bring it up in Excel, which raises a problem. What if they don't have Excel, what if I want to move it from Excel to a database?

**MARK BENIOFF:** So this is like a small database that is running on your PC, [it] can even be a server environment and now we can show we can publish it.

**JON McCORMACK:** That's right. Because one thing we all know is people going through downsizing and upsizing of [the] last decade and start of this decade; there's lots of data around the organization, none of it connected and now we're going to show you how to do it. So Power Browser [is] unlike any other currently available; [it] is also a server. So what we see here is we've got the Power Browser server and the Power Browser server manager and this lets you put your PC onto the Internet as a network peer and lets you publish real live documents from your PC onto your internal Internet. I'm going to show you how to do that.

So if I want to connect a database or an Excel spreadsheet to the Web, I just say New Database Query, and I'll select Product Specialists which [is] an ODBC data source.

**MARK BENIOFF:** So are you going out through standard data access or—



**JON McCORMACK:** We also support Oracle SQL. We'll just hit okay and now it goes out and it picks that up and it comes in and says we've got one table or you've got one book here which is specialists. So we'll choose Specialists. Then it says, well, okay, this is a dynamic document. You want people to be able to search. You don't want something static. So let's do a search criteria.

So let's Search Product. And now that we've searched on product, what fields do we want to come out of this. Well, we want the product specialist, we want the region, and we want the e-mail so that we can send the mail about, okay, I've got Power Browser and I want to know how to use it.

**MARK BENIOFF:** So very simple. Here's my table, here are my columns and now what do we do? How do we actually publish this information?

**JON McCORMACK:** The first thing we do is give it a name so we'll call it product specialists and this will generate a couple of things for me. One thing is an input form which is an HTML form over the Web, so we need to give that a heading, so we'll call this "find specialist" and a heading for the result sheet, "The specialists are," and we'll just hit OK here. And now this has gone and generated for me, if I just refresh my list, it's gone and generated three files. One of them is an HTML file which is the input form, another is a query definition, and another is the client-side scripting stub that I can go in and edit to add validation.

So if we just run this to show how it works, I come up and here's a generated form. I haven't got into typing any code yet, and because we've got the browser and server completely integrated I can actually run and test this form from directly within the server manager. So I want to learn how to use OPO so I type it in, hit submit. Now this has gone back to the database and it's come back and said, okay, Jeremy Burton is the guy to talk to in [the] U.S.A. and I can click here to send him e-mail.

Now, what we have done is taken what used to be a data island, we've exposed it on the internal Internet and anybody, whether they're using Mac, Windows, Windows 3.1 or Unix can access that as an HTML form.

**MARK BENIOFF:** We're using ODBC. What's the status right now?

**JON McCORMACK:** This is currently in developer's release two and is going into production on April 1st.

**MARK BENIOFF:** Thank you very much.

Jeremy, how are you doing over here?

**JEREMY BURTON:** We've overcome a few traditional problems there. So I wanted to demonstrate what everybody is doing today, and we've done that. Okay.

**MARK BENIOFF:** Sounds good.

**JEREMY BURTON:** What I want to do now is move on and just take on where I left off before.

What I've gotten is, again, my client/server application running here. This is what everybody has seen and everybody is familiar with. We've got customer information coming out of SQL information and order information coming out of Oracle and you may notice we have a number of different fields in that orders table including the waybill number, and the company that we're using to send the package.

**MARK BENIOFF:** Right. Now it seems like with the Internet some of this waybill information is more valuable. In the past if I had this application, if I was an order entry operator I'd have to call Fed Ex or DHL and let them know what the waybill number was but I can use Power Browser or something like that and make it happen; right?

**JEREMY BURTON:** That's right. And we've invested a lot of money in client/servers these days and we want to extend that architecture to access the information on the Internet. So what I want to do is drop out of the run-time environment here. What I've got, you might have noticed, I've added a couple of tabs to the top of the form there, and what I'm going to do is just drag and drop in an OCX which is actually the Power Browser in OCX form.

**MARK BENIOFF:** So everything you saw over there in the complete Power Browser environment which is all the scripting environments, VBS and Java and JavaScript and all the features and functionality of Power Browser is also now available as an ActiveX Control. So you can embed it inside ActiveX containers like power objects.

**JEREMY BURTON:** So here we have the same application. We can look at our orders again. And you notice here the first order has been sent by FedEx and we've got the waybill number there so what I can actually do is go onto the package tracking screen and pass the waybill number down to the home page and give me some useful information back from the Internet. I think the Internet is a great thing, there's a lot of great things but in terms of getting useful information back—

**MARK BENIOFF:** You can embed a browser inside your client/server application.

**JEREMY BURTON:** That's right. So we're back at the FedEx home page.

**MARK BENIOFF:** What if you're interested some of the information Jon had over there in Power Browser and he just published that product reps list. Can you access that?

**JEREMY BURTON:** Absolutely. That's just one more piece of information, one outside of the fire wall and one inside. So we go in and we've got the table Jon published earlier.

**MARK BENIOFF:** It's publishing the information here. Great. Glad to see that running. Great job. Thank you very much.

(Applause)

**MARK BENIOFF:** Okay. So what we tried to show you is what Oracle is doing in this area. We're excited to work with Microsoft. We'd like to thank them for having us here. This is an unusual situation, they're being good sports about it, and enjoy the rest of the conference. Thank you very much.

```
<!--HtmlStart--><P><IMG SRC="/intdev/gifs/uparrow.GIF" WIDTH="14" HEIGHT="10"
BORDER="0" ALT="Up"><A HREF="/intdev/pdc/pdcgen.htm">Back to table of contents for
General Sessions</A><P><IMG SRC="/intdev/gifs/uparrow.GIF" WIDTH="14" HEIGHT="10"
BORDER="0" ALT="Up"><IMG SRC="/intdev/gifs/uparrow.GIF" WIDTH="14" HEIGHT="10"
BORDER="0" ALT="Up"><A HREF="/intdev/pdc/pdcpres.htm">Back to table of
contents</A><P><!--HtmlEnd-->
```