# Developers and TechEd: IT Forum 2007 5 November 2007

## Keynote Speech

### S Somasegar

#### Corporate Vice President, Microsoft Developer Division

##### Introduction

Good afternoon. I want to thank you all for coming here. It is a great pleasure for me to be here this afternoon and to get a chance to talk to you here at TechEd. One of the things I like about coming to such a forum is that I get the chance to interact with at least some of you and as I look at you all I know that many of you have played a crucial role in working with us hand in hand in delivering the next version of Visual Studio, the next version of .NET Framework. When I say that, some of you have probably given us feedback on our designs, some of you have taken a look at our early buzz in terms of Community Technology Previews and the like. Some of you have given us bugs that we had to fix, some of you have given us feedback on the kinds of functionality and scenarios that you wanted us to work on. Some of you have gone one step further and taken a bet on our products and technology and are building applications that you can deploy in production environments even before we ship the product, so that we know that by the time we ship the product it is truly ready for the demands of our customers and developers around the world. To each and every one of you who have played a role in helping us shape and deliver a great set of products, I want to take this opportunity to say a big thank you. Over the next few days, we have planned a lot of interesting and informative sessions for you all. I hope, as you walk through those sessions, you will share at least some of my enthusiasm for the kinds of innovation and innovative work that we are delivering today, in the coming months and over the next to 12 to 18 to 24 months. There is a lot of interesting stuff coming out here and so I hope you have a great time here at TechEd.

This is the first time I have come to this wonderful city of Barcelona. Some of you may have been here before and some of you may live here, but I have heard so much about this city, so much about the culture, the architecture, the people; it is nice for me to get a chance to experience it. As soon as I landed here on Saturday evening I found out that there was a dinner engagement that some of my colleagues had planned and was not sure whether I needed to take the time out to go for dinner, because I had just crossed many time zones and it was a long flight. I was not sure whether or not to do it. I finally decided that since my colleagues had planned it I should do it. They took me to a restaurant called ‘Moo’ in a hotel called ‘Ohm’; it is an interesting name and I thought I should try it out. As is typical of the local custom here, the dinner was set for 10pm. We went to that place and little did I realise going in that it was going to turn out to be one of the most fantastic dining experiences that I have had in a long time. It was a long dinner of three hours and by the time dinner was over it was about 1am and I was really tired, so as soon as politely possible I left the dinner and hopped into a cab. I was driving back to my hotel and reflecting upon how tired I was and not really excited about going in yet I went anyway and why did it turn out to be such a fantastic experience for me? Two things jumped out at me: one, obviously, the chef and whoever else was involved in preparing the food did a fantastic job putting together a great dinner. Equally, going hand in hand with that was the ambience, the décor and the way the food was presented. Together, it was a fantastic experience.

Being the guy from the Developer Division of Microsoft, it quickly hit home with me an analogy that I have been talking about for a couple of years now. In this world of mechanical applications and distributed applications and applications that run the next generation businesses, differentiated user experience is absolutely key for the success of the business of the developer of the applications. I have been saying for a while now that it is very important for developers and designers to come together to work hand in hand in a collaborative fashion using a set of tools that enables seamless workflow between the two sets of people. If we can enable that then I feel good about it, because the designer and developer together can build fantastic user experiences for the next generation applications. The products that we are working on, such as Expression Studio, Visual Studio, .NET Framework, Silverlight most recently, all of these products, technologies, tools platforms, are geared towards enabling the designers and developers to work together in a collaborative fashion to enable our developers and designers to build great applications.

##### Visual Studio 2005

If you go back in history and remember what happened two years ago this month, it was November 2005 when we launched Visual Studio 2005 and .NET Framework 2.0. I thought I would take a moment to step back and reflect upon what kinds of things we have been seeing with these products in the marketplace for the developer customers. As of today, we have over a million professional developers from around the world taking a bet on Visual Studio 2005 and .NET Framework 2.0 for the applications that they are building and for the successes they are seeing with their customers. No matter which way you look at it, that is a phenomenal number. Visual Studio 2005 is by far the most successful toolset, not just from Microsoft, but broadly from pretty much anybody else out there. I feel really good about the progress that we have made there.

With Visual Studio 2005 we also took this as an opportunity to expand the set of developers whom we were going to target with this set of tools and this framework. At one end of this spectrum, therefore, we introduced a new set of products called the Visual Studio Express Edition, which was geared towards the beginner developers, people who were just starting with development who may not have been professionals or proficient. We wanted to provide a way that was easy for people to get started, to learn, to start figuring out what it means to be developing applications and software easily. If you look at the response that we have seen for Visual Studio Express, it has been mind‑boggling. We have seen over 17 million downloads of Visual Studio Express Edition so far. Clearly, I feel we have been very successful in terms of broadening the reach of developers with our toolset using Visual Studio Express.

At the other end of the spectrum, we also wanted to use Visual Studio 2005 as our foray into the application lifecycle management tool space with the introduction of Visual Studio Team System and Team Foundation Server. With these products, we said that historically we have always been focused on individual developer productivity. With every version of Visual Studio we have said that we wanted to provide a set of tools that made you – the individual developer – highly productive. With Visual Studio Team System and Team Foundation Server, we wanted to take the opportunity to expand our focus from the individual developer and their productivity to think about the entire development team and make the team more productive and enable it to work together in a collaborative fashion. As of today, over a quarter of our developer base, about 25‑30%, are using the Visual Studio Team System tools and really seeing the benefit and power of those tools in an integrated fashion, working hand in hand with other members of the team against a common repository at the back end with Team Foundation Server. We have made a good start in this space and with Visual Studio 2008 and everything else that is coming down the road we feel that we are set to be even more successful for our customers in this space.

##### MSDN

As much as I am excited about the technologies and the products and the cool stuff that we do at Microsoft, I have always believed in having a vibrant and self‑sufficient community to go hand in hand with our products to enable the kinds of successes that we all care about enabling for others and for our customers. Let us take a look at the work we have done in MSDN. About a year or 18 months ago, if you had come to our MSDN forum and posted a query or a question or you wanted some information, we were running in the 30‑40% response rate. In other words, there was a 30‑40% chance that, if you had a question, it would have been answered in a timely manner. That was an unacceptable state for me. My vision is this: if you are a developer taking a bet on our stack, no matter which part of the world you are in, if you have a question, a query, need some information, are blocked and need some help, you need to have the confidence that you can post a query and within a few hours, within a day, maybe at the most within two days, you know that somebody out there is going to help you answer your question so that you can go on and continue being productive. That is the state that I want us to reach. Therefore, we decided a couple of years ago that we needed to invest in this space, because as much as we can say that the community is there for the sake of the community, we felt that we needed to jump‑start the community. So, working hand in hand with some of the community leaders, we decided to do something about this and, as a result, today we are running a little over 80% response rate on our MSDN forum. Eighty percent is not 100%, so we still have a little way to go, but clearly we have come a long way in terms of making sure that each and every one of you can help each other as much as we can help you in the process.

Hence across the technology, with the options, the momentum, the kinds of structures that our customers are seeing, the vibrancy in the community, I feel very good about the progress that we have made with both Visual Studio 2005 and .NET Framework 2.0 in the last couple of years.

##### Our Mission

How do I describe our mission? We have always been focused on making sure that we take care of our developer customers. That is what the focus has been for our division and for a lot of the products and technologies that we have been working on. If you go back in history and look at 1975, which is when Microsoft came into existence, ever since then we have always been a platform company in our hearts. Today, if you look at our businesses, we are in the operating system business, we are in the client server devices space, we have something in the games area, we have something in online services. We really are working on a set of products and technologies that span the breadth of the kinds of problems that you want to see solved using software. When you are working across this breadth, how do I think of us as a platform company?

Pretty much every piece of software that we do we always think about as having a platform component and having an end‑user application or experience component. Now, I will be the first to tell you that being a platform company you can never scale up to meet the demands of every customer out there, so the best thing that we can do is ensure that we have a great platform, which is accessible, modular. We will probably be building a set of applications and experiences on top of the platform, but I truly expect the rest of the developer community around the world to be able to build additional applications, additional experiences, additional services on top of this platform and be able to deliver great experiences for our customers.

##### Variety of Developers

If I look at things from that perspective, therefore, I really want to be in the business of delivering a set of tools, runtimes and platforms that provide the broadest pipeline to the developer community around the world. In addition to developing great technologies and great products, we want to make sure that we have the strongest partner ecosystem and the most vibrant community that go hand in hand with these technologies to deliver the right set of things for our developers around the world to be successful in what they are doing.

If I drill down one more step into what this really means, on the one hand, there are many kinds of developers in the world today. Take my 11‑year‑old, for example, she does not know programming, she does not want to know programming – at least today, maybe some day in the future she may want to learn programming – but she spends a fair amount of time online. She has an account on Facebook and she wants to get online, customise her home page and be able to share the things that she is creating with her friends and family. I think about her as a developer; although she is not writing code, she wants to create an online experience. That is one kind of developer. As another example, my wife is an accountant by profession, she does not program for a living, but occasionally she dabbles a little in programming to automate some tasks that she otherwise would have to do manually. That is one end of the spectrum. At the other end of the spectrum, there are developers who code for a living, day in and day out; developers who live within a business environment working with a team of people building the next generation mission critical applications; and you have everything in between.

Thus, there is a variety of developers with different skill sets wanting to build different kinds of applications, different kinds of experiences, and they want to target different kinds of platforms. I may want to build an application that I want to deliver on the web. I may want to build a rich client application, an application that runs on top of Office, a server application. I may want to build a service or I may want to service‑enable my application. There are different kinds of things that I want to do. Therefore, on one hand, I have different kinds of developers with different skill sets having different expectations of what they want to do. On the other hand, I have a set of platforms that people can target. Our mission, therefore, is to be able to deliver a set of platform technologies and a set of tools to be able to bridge these two worlds, so that as a developer, no matter what application you want to build, no matter what platform you want to target, we have a consistent programming model in .NET and a toolset that you are very familiar with in Visual Studio and Expression Studio to be able to build and deliver the kinds of experiences that you want to do for your customers and for yourselves.

##### Current Platform and Tool Highlights

###### .NET Framework 3.5

I now want to drill down into the kinds of advances we are making on the platform and tools sides, and then show you some of our products in action.

With regard to our platform, as we speak here, the product team back at Microsoft is putting the finishing touches to .NET Framework 3.5. .If you have been following the evolution of .NET, NET Framework 3.5 is the next big step in terms of delivering higher levels of productivity for our developers. With .NET we also have a principle in mind where we want it to be a consistent programming model that has the broadest applicability. You may be building a device application using Compact Framework; you should be able to use your .NET programming skills. You may be building an application for the client; you have the full‑fledged .NET Framework. Most recently, we announced with Silverlight we want to bring the power of .NET programming into the Silverlight runtime so that you can reuse your skills, both in terms of programming model and in terms of the programming language, and be able to deliver rich internet applications over the web. We really wanted to have a broad applicability for .NET Framework and we are heading in that direction with .NET Framework 3.5 as well.

With .NET Framework 3.0 on Windows Vista we delivered significant new advances in terms of Windows Presentation Foundation, Windows Communication Foundation and the Workflow Foundation. We are enhancing those frameworks and providing additional functionality in .NET 3.5. For example, we are adding a whole bunch of new controls for Windows Presentation Foundation; we are adding support for a set of protocols for Windows Communication Foundation to be able to build connected applications in an easy way; we are also taking the workflow designer that we had and integrating it into the product. Thus, we are doing a variety of things.

The one thing that I am very excited about in .NET Framework 3.5 is support for language‑integrated query or what we call LINQ. If you think about how programming has evolved over the years, you have the data world and you have the programming world and they are two different silos. Pretty much everybody out there builds data‑enabled applications but you as a developer need to understand the programming language, the programming constructs, the data language and the data constructs and you have to figure out how to bring those two semantics together to build an application. We wanted to make that easier for our developers and so we said that from the same programming language that you are already used to – it could be VB or C# – using the same language constructs, if we can make querying of data – it could be relational data, it could be xml data, it could be objects – a first‑class citizen then we feel as if we have moved the bar considerably in terms of productivity for our developers, so that they can understand one set of constructs and be able to build data‑enabled applications. That is the work we are doing with LINQ and that we are on a path to delivering with .NET Framework 3.5.

###### Silverlight

Silverlight is the new entry from a platform perspective from Microsoft. We shipped the first version of Silverlight earlier this year. Think about Silverlight as a cross‑platform, cross‑browser runtime that enables you to deliver a richer set of media experiences and internet applications over the web. Today, you can take the version of Silverlight that is out in the market, use JavaScript as a programming language and be able to build the whole thing. With the next version of Silverlight that we are currently working on, which we will deliver some time next year, we are bringing in the power of .NET programming to the Silverlight platform so that you can reuse your .NET programming skills, the language that you are already used to and familiar with, and be able to create great experiences with Silverlight.

Therefore, across the board between the full .NET Framework, Silverlight and Compact Framework we feel like we are continuing to move the dial in terms of both functionality, security, reliability and the like as well as coming through on our principle of providing broader applicability for the framework. That is the set of advances that we are coming out on the .NET Framework side.

###### Visual Studio 2008

Going hand in hand with that, if you think about what is happening on the tools front, as I mentioned before, as we are putting the finishing touches on .NET Framework 3.5 we are also putting the finishing touches on Visual Studio 2008. There are a couple of things that I want to point out here. First, as with every other previous version of Visual Studio, we are moving the bar considerably higher on enabling developer productivity. I talked about how LINQ is a great of how we are moving productivity higher for our developers. There are a couple of other examples that I want to make. One, for the first time in the history of Visual Studio, we are supporting multi‑targeting. Previously, I would have told you that you have to use a particular version of Visual Studio with a particular version of .NET Framework to be able to build and target that framework or for an application to target that framework. For the first time with Visual Studio 2008, you can have one toolset and be able to build an application, the target, .NET Framework 2.0, .NET Framework 3.0 *and* .NET Framework 3.5. You do not have to have multiple toolsets, which we think is going to be a huge step forward and, looking at the feedback from developers, people are very happy that we are now heading down the multi‑targeting path.

The other thing we are doing with Visual Studio is making sure that that is the toolset you can use to take advantage of the latest and greatest platform features. If you are building an application for the web, a DAT file application, we want to add ASP.NET extension integrated into Visual Studio 2008. If you want to take advantage of the latest and greatest features in Windows Vista, be it Windows Presentation Foundation or Communication Foundation or Workflow, we have designs built for those things that make it easy for you to develop applications. We are also making it easy for you to target applications that are going to be running on Office 2007. In the past, I would have said if you want to use Office as a development platform, go buy something called Visual Studio Tools for Office. That is a separate product, use it and then you can get all the tools for Office. We had feedback from many of you that said it was more complicated than it needed to be and why did we not make it easy for you to get access to the Visual Studio Tools for Office. We have taken you feedback and integrated Visual Studio Tools for Office into the Visual Studio Professional product, so each of you who has access to Visual Studio 2008 Professional will automatically receive all the tools for Office functionality so that you can use Office as a smart client platform and be able to customise and build applications on top of Office.

Therefore, between the servers, between the clients, between Office and devices, we have a good story in terms of Visual Studio being the toolset you can use to take advantage of the latest and various platform features.

###### ALM Tools

We are also taking the next step forward in terms of our ALM tools journey with Visual Studio Team System and Team Foundation Server. There are a couple of things that I want to highlight here. We have done a bunch of work in terms of scalability and performance on Team Foundation Server so that if you have a large development team and you have dozens of people wanting to work together in parallel, you can expect the right level of performance that makes you productive and gets you to do the things that you want to do. That is one big advance we have made.

The other thing we have done is we took all of our database tools and integrated them with the rest of Visual Studio Team System so that database application development is truly a first‑class citizen and is a part of our application lifecycle management process.

These are some of the things you are going to see with Visual Studio 2008 that is going to come out later. We also released Expression Studio earlier this year. The goal for us between Expression Studio and Visual Studio 2008 is to enable a seamless workflow between these two applications or these two toolsets. They both use the same language. You can open up a project from Visual Studio and be able to open the project again in Expression Studio and know that you can work back and forth. This enables a seamless workflow between the designers and developers that we hope will enable people to deliver the right level of user experience in their applications.

These are some of the advances that you are going to see in the near future here.

##### Other Developments

###### Prescriptive Guidance

Going hand in hand with platform and tools advances, we also want to continue delivering more prescriptive guidance on content that enables our developers to be successful. As part of our patterns and practices work, we are going to continue giving you prescriptive guidance for the latest platform technologies. There is one new thing that we are starting to talk about now, which is what we call the software‑plus‑services blueprint. Think about this as a way to get you started easily to build a set of software‑plus‑services solutions right out of the box. There is going to be some prescriptive guidance, some tools, some frameworks with access to source code, so that it is very easy for you to take your application and make it service‑enabled. That is some work that we are very excited about.

###### MSDN

If you think about the work that we are doing on MSDN, historically, MSDN has been a publishing platform for us. Microsoft has put out a bunch of great information, source code, samples, bits, content, but it is a one‑way publishing environment where we give you some information and then the rest of the world consumes that information. That is a good start, but that is not where my vision ends for MSDN. I want MSDN to be a community platform where as much as we can publish information I really want to enable the rest of you to publish information on MSDN and be able to share your knowledge and expertise with one another. If you think back to my earlier comments about wanting to make the community both vibrant and self‑sufficient, I need to provide a forum where the community can come together and share their knowledge with each other. I really want to take MSDN in that direction.

There are a couple of things that we have already started doing. We have something called MSDN Wiki now, where we took all of our product documentation, published it on MSDN and said you as a community member outside Microsoft can come and take a look at it. You can annotate the content selection that you want; you can add some more content to it so that it can be shared with the rest of the world. I always say that only about 10% of the content comes from Microsoft for any product. The rest of the content is in the community’s hands, except there is no easy way for the community to be able to share it with the rest of the world. I want to change that. I want to make it easy for the community to be able to participate in this.

###### Localising Visual Studio

Likewise, we are localising Visual Studio into nine different languages. The world has a lot more languages and there is a scale issue that we run to when we try to do everything for everybody. Therefore, using machine translation as a base technology that comes out of Microsoft Research, I want to be able to give you a first‑cut translation into a new language and then enable the community to be able to do sanity tests and build on top of that. Currently, we are working on a pilot project with a few community members in Brazil to get them to translate our product documentation into Portuguese. That effort is going well and we want to do more from a technology and tools perspective to enable communities around the world to be able to scale up how we provide a broader reach to our tools and products to a set of people who want to access these products in their own language.

###### MSDN Code Gallery

There is something called MSDN Code Gallery we are working on currently. Today, I know how to put out some sample code on MSDN that can be consumed by you all, but there is no easy way for me to say, ‘You have a great sample code, do you want to share it with the rest of the world? Come publish it here.’ MSDN Code Gallery is something that will let you do that. We are looking at releasing a beta of MSDN Code Gallery some time in the next month or two and then, I hope in the beginning of next year, we will make this live so that each of you who has some great samples that you want to share with the rest of the world will be able to do that when you come to MSDN. I really want to transition MSDN from a one‑way publishing platform to a collaborative community platform and we are well on the path to getting there.

##### Microsoft Sync Framework

I want to take a moment here and announce a couple of things that we are going to make available today and in the near future. The first is we are making available a Community Technology Preview for Microsoft Sync Framework. If you have played around with Visual Studio 2008 you will know that we already have some functionality for offline synchronisation built into the product. Now take this Microsoft Sync Framework and think about taking data synchronisation to a whole new level. Sync Framework is basically a framework that lets you easily sync‑enable your applications for two scenarios: to provide richer offline synchronisation capability and to enable peer‑to‑peer collaboration capability. Ultimately, as an application developer, the thing that you care about the most is enabling the data to follow your customers no matter where they are, no matter what application they are using and no matter what device they are accessing that application from. Microsoft Sync Framework is a good step forward in terms of delivering on our data vision to be able to synchronise data irrespective of the protocols you are using, irrespective of the data store and irrespective of the data life, so that you can always ensure the data follows your customer. That CTP is available today. I would encourage you to download it, take a look at it and give us your feedback, because as we have tried to make this a robust framework that spans multiple devices and protocols, we need your help in making sure that we are designing the framework and the [inaudible] in the right way.

##### Add-Ins to Outlook 2007

Note that this is going to be the first in a series of blueprints that targets different scenarios in different problem domain spaces. The first one that we are making available today is something that is going to let you take Outlook 2007 and enable you to quickly do an add-in that both exposes data from Outlook and enables you to interact with third-party services. Here is a standard scenario: I am a big user of Outlook 2007, I am sometimes absent minded when it comes to personal things, and my wedding anniversary is in a few months and I often do not do a good job in planning for it in advance, buying a gift for my wife and doing all the right things. Would it not be nice if Outlook 2007 was a little bit smarter than it is today, and there was an add-in that could understand when my anniversary date is coming up, knows that my wife loves to shop, and can automatically know the amount of money that I am planning on spending and the kinds of gifts that I have bought in the last five years, and has a web server that exposes the things that I could buy in terms of clothes, jewellery, and based on the budget and past shopping experiences such as Nordstrom, could go out and buy something, and automatically schedules it for delivery to my house on our wedding anniversary. That would be phenomenal, and more than anything else it will put me in my wife’s good books.

That is the kind of scenario that I want to enable. There are some uses for personal reasons, and a whole lot of scenarios that you can enable in a business environment, but being able to do add-ins to Outlook 2007 in a simple, easy way, that lets you connect Outlook 2007 to external servers, both in terms of exporting data and in terms of importing data, is a huge step forward. This blueprint contains a set of frameworks with core scope, a set of tools that work in the [inaudible] context, and a set of prescriptive guidance that enables people to service enable Outlook 2007. Over the next few months you are going to see us deliver more of these blueprints that enable you to deliver software plus services in a straightforward and easy way.

##### Launch of Visual Studio 2008 and .NET Framework 3.5

This one I am really excited about. These products are coming out of my division. So there is a little bit more unbiased excitement from my perspective. We have been working on Visual Studio 2008 and .NET Framework 3.5 for a couple of years now, ever since we registry in 2005. As we sit here today the team is getting the final build out, we are going to go through one more round of regulation testing to make sure that everything is fine, and before the end of this month, we will make .NET Framework 3.5 and Visual Studio 2008 available for our MSDN subscribers. Soon after that we will make it available in retail for our customers. We are still planning on an official business and marketing launch for Visual Studio 2008, along with Windows Server 2008 and Secret Server 2008 at the end of February 2008. As of this month, we will be done with these two products, Visual Studio 2008 and .NET Framework 3.5, and these products will be available for you to start using. I am really excited about this milestone.

##### Conclusion

As I was getting ready and thinking about what I should say in the rest of the keynote, I had a couple of choices. One was that I could spend a lot of time, maybe one, two, three, four or five hours, talking about all the great features of Visual Studio 2008 and .NET Framework 3.5. Then vanity prevailed, and I said that I did not want to waste people’s time by boring them with a monolog, but rather show you some of the cool features of these products, which will enable you to build great applications. I will pass over to my colleague, Tony Goodhew, who works in the product marketing team, who will show you these products.

## Product Demonstration

### Tony Goodhew

#### Product Manager

##### Preamble

I am going to show you how developers could be more productive building web applications using Visual Studio 2008. I have a website for a fictional supermarket chain called Contoso Foods. Contoso foods have been entering the online food business and building their website using ASP.NET 2.0. I have opened up their existing ASP.NET 2.0 application inside Visual Studio 2008, and I can now work against this project without needing to deploy anything more to my service.

##### Visual Studio

This is the website, it has a data grid, some list views, some drop downs, let us see what that looks like inside Visual Studio. This is my shopping list page, where you can see it is designed so that the editor has the colour and text highlighting that you have come to expect inside Visual Studio 2008 for building web applications. One of the things we wanted to do was make developers far more productive inside the web environment, so we have not only the editor and design view that you are familiar with; we also have a new split view. The split view enables you to see both the html code and the design view at the same time, and will keep the two windows synchronised. So by clicking on the control we will jump to the right part of the window, if I click down here we will jump our source around. If I did the same thing in the source view, we jump the design window around as well.

We have also added a great new feature called our breadcrumb list. If you look at the bottom of the screen you can see our breadcrumb list, which shows where the asset is that you have selected and how it exists in the html hierarchy. You can also hover over the item and see the actual declaration in the html page. This is a great way of seeing where in the rendering part a control will be when you start working with it on the design or source views.

##### Cascading Style Sheets

We have also done extensive work around cascading style sheets, which was one of the big customer requests for Visual Studio 2008. We now have full cascading style sheet support, will link the pages to external style sheets, and will correctly put styles you create into those external style sheets. So by selecting an item, I can go in and view my style sheet, and you will see down here that I have a list of my styles, and when I hover over them I can get a tool tip that shows what the style definition is. We also have a great new feature that will help you understand what is happening when you apply a style. If I click on view and select CSS properties, we now have the CSS properties window. The properties window enables me to see the order of styles applied. All

the time an HTML developer will look at an asset and go ‘why is that not rendering correctly?’ With this property grid you can go in and see where you are, what rule are being applied, and in this particular case I could scroll down to position and see down here that I am applying a few items to that style in the position grouping, with top width and making it relative.

##### Upgrading

Not only have we done all of that work, we have also made it easy for you to upgrade. When you move from one version of Visual Studio to another, you usually will move on a framework as well. So we have an ability to upgrade. I am going to select properties and click our target framework drop down. One of the great new features in Visual Studio 2008 is what we called multi‑targeting, which enables you to use Visual Studio and select the version of the framework that you want your project to target, whether it is 2.0, 3.0 or 3.5. I am going to select 3.5 because I want to upgrade my project. I am going to click on yes, and I have now upgraded my website from 2.0 to 3.5.

Soma Somasegar

That is fantastic, but if I wanted to Visual Studio 2008 and still target 2.0, I could still have done that? I do not need to have necessarily upgraded?

Tony Goodhew

That is correct, Soma. I can open up my existing website, I can work against them, and by selecting the target framework as being 2.0 I will have a constrained view of my tool, and be able to develop against that platform.

##### LINQ

I am going to go in and add a new item to my website. One of the great new innovative features we have made inside Visual Studio is LINQ (language integrated query), which enables developers to easily access objects, XML, and relational data in a consistent way, and with a concise, straightforward set of language taps.

I am going to select the LINQ to sequel class, and we will call that Contoso Inventory, and then I am going to click add. What we will do now is add the LINQ to sequel file, and it will bring up our object relational design. When I click on my server explorer, I can select my categories, order, order detail, products and suppliers, and drag those over to the design service. Right now we will look at the server, extract the schema to those tables, and work out the key relationships and present this information to you. Here you can see how my table layouts are related to each other.

Now I have done that, I have created a thing called the data context, and I can go and use that just like I would use data sources and data from previous versions of Visual Studio. If I open my shopping list again, look at the design view, click on the smart pack of my drop down button and choose data source. We will create a new data source, click on OK, select LINQ, we have already auto‑filled the context that I am using, and I want to have a look at my products table. So I will click finish, and I want to show the product name in that drop down, so I click OK. What I have done there is very simply show you how to create a LINQ entry that takes your databases and surfaces that data, and how easily you can consume that in an ASP.NET component.

##### ASP.NET AJAX Controls

Just simply LINQ was not all we did in moving forward to the .NET Framework 3.5; we also integrated the ASP.NET AJAX controls. Now when you get Visual Studio 2008, you have that that AJAX ability inside the toolbox and inside the environment. I want to go in and AJAX-enable my page. Currently when I change pages my shopping list will do a round trip to the server, and I want to get rid of that. So I change back to source, scroll down to my grid view and drop in a piece of ASP.NET code, and that creates a script manager, which is the base AJAX class, and an update panel. The update panel is what we use wrap a control so that it gets AJAX enabled. I am going to put my grid view inside that control and, now I have my AJAX control and my grid view enabled.

The code looks a little bit messy here and I want to clean that up. There is a cool feature in Visual Studio, control-K-D, which auto formats your code, so that makes it look like you have actually thought about how to lay stuff out. Now that I have done that, we can go back to the design surface, and we can see that I have my link data source, I have my AJAX script manager there, and I have now enabled this page. As I am on 3.5, I can also take advantage of integrating things such as Silverlight into the application.

##### Virtual Aisle Browsing

My designers have been looking at how people shop. They have discovered that a lot of people like the experience in the shop of walking up and down an aisle, and seeing the different items. They have found that that experience does not exist in an online store. In order to replicate that they have built a Silverlight control, that does the virtual browsing of an aisle. I am going to go over to my aisle browser page, and drop this control into my function. I have a JavaScript function that will launch that control, and JavaScript is now a first‑class member of that environment. So I have colour syntax highlighting, if I were the type I would have IntelliSense, I have drop down support, I have code completion, and so on. I do not need to do that, because I have a piece of code already written. Here is a call that will go and instantiate my Silverlight object.

I talked about JavaScript being a first‑class member. A full class member is not just having the support and the IDE, but also having full support in the debugger. So I can set a break point against this item. Now if I run my application, what we see is my shopping list is rendered, I have my drop‑down linked to my product names and, if I look through my shopping cart, you will see that I do a jump back and forwards, because the shopping cart starter grid is now AJAX enabled. I can click on virtual browsing, and we hit the break point. This works in IE and it will also work in Firefox. We will run our page, we will load our Silverlight object, and I can now zoom up and down my aisle and select the item that I am after.

I just showed you how you can be much more productive as a web developer using Visual Studio 2008, integrating things such as LINQ, AJAX, and Silverlight into your web applications.

##### Automating Processes

Soma Somasegar

Tony, you did a good job in showing what we can do within the framework, but really you focused on the consumer‑facing side of things, presentation, some ASP.NET, binding the links. That is all great stuff, but what about some of the features in Visual Studio 2008 and .NET Framework 3.5? What can you show us in relation to the back office, things like workflow? Can you show us anything there?

Tony Goodhew

Absolutely, Soma. I can show you some of the client and server features of Visual Studio 2008. Contoso Foods uses Microsoft SharePoint to track their business. They keep a track of a number of key performance indicators. One of the key performance indicators that they keep track of is how many times customers go into a store and ask for a particular product. If I look at the business side, what I can see is that in my product selection index I have a warning status. So I am going to click on that to see what the issue is. What I can see here is that we have a backlog of products in the request for quince paste. My business guys have looked at this and have discovered that our vendor approval process is too slow. So they came though to the development group and asked us whether we could speed that up. One way that we can speed up the process is to automate it, by making use of automated workflow.

Let me go and show you the workflow that we built. Here is our workflow designer, which is showing the workflow that a vendor approval will go through. The first thing that we will do is have a look at the application, do some reviews around credit checking, and then we will ask our procurement people to approve the process, and then tell our store managers that they have a new product and need to do something with it. I am going to go back to Contoso Foods and start the workflow process. Let us have a look at the vendor pipeline, I am going to edit this item, select in progress, and click okay. You will see we are now back in my workflow, and we have broken on my break point in my workflow process. We have integrated the bugging in Visual Studio for workflows, whether they are built on workflow foundation, on Windows SharePoint services, or on Microsoft Office SharePoint server. So you will be able to do development against all those three workflow platforms.

##### Workflow Design Tools

The workflow design tools are available in every version of Visual Studio, so you do not need to get anything additional to be able to build this type of workflow-enabled application. If I run this, and we can see that my workflow will run through and complete, and return me to my homepage for Contoso Foods. Now that my workflow is going on it has sent an email to our procurement people, and it has asked them to review the vendor application. Typically this is the vendor application that they would have to review, it is a Word document, with information about the company, the business, the products and so on. The procurement person would look at this document, before going off to query some online services, looking at what customers think of this particular product and this particular company. They would get some prices, and have a look at what an actual product looked like. It would be great if we could place that information into Word, so that my procurement people do not need to do a context switch from one application to another. They could stay in the application that makes sense for what they are doing, the business process, and actually execute the business process from inside that application. To do that, I have built an Office custom Task Pane. This enables me to stick that inside Word, and when those documents are loaded, this Task Pane will come up.

##### Using Windows Presentation Foundation

I want to provide not only information about the product reviews and the pricing, which we have done through a set of controls built around Windows Communication Foundation and Windows Live Services, they will go up to Live services extracted information and then present that to us. I also want to give them that view of the product. So I can make use of Windows Presentation Foundation (WPF), to go and get an image of the product, present that image to me, and enable me to look at that. I have a control already written in WFP that does that, what I need to do is integrate it into my Windows Forms application. To do that, I go to my toolbox and drag over a control called Element Host. Element Host is a control that we have built that enables interoperability between Windows Forms and WPF control.

Soma Somasegar

Is it one‑way interoperability or two‑way interoperability?

Tony Goodhew

It is two-way interoperability. This means that you can go into a Windows Forms application and drop in Element Host. You can go into a WPF application and drop in the corresponding host control, and host the right technology in the right place, so you can evolve your existing applications with Windows Forms application with WPF controls, without needing to make a jump straight to a new technology. I am going to dock this in my parent container, and we will select my vendor products control, and I have a default representation, which is the quince past item.

Now I have that, because I have built this as a custom task pane, if we go back and have a look at my documentation, I will see my vendor application and custom task pane. Here you can see the product reviews, the product pricing, and if I click down the drop‑down, I am going to have a look at quince paste, and here is my WPF control rendering. So I can have a look at the product, I can move it around, if I had good eyesight I could look at the nutritional facts.

So now as a procurement officer I can get all the information I need to make my decision. For the purposes of this demo, I am going to click on accept. If you remember our workflow, one of the things that we wanted to do was go in and alert the store managers that there is a new item. The procurement officer has approved it. We will go through the process of ordering it, and now we are going to make use of SharePoint Server’s ability to directly create Outlook entities. So we have directly created an Outlook task for our store manager, to alert them to the new product.

One of the cool things that we want to do with a new product, when we know there is demand for it, is to be able to place it in our store in a location that would generate a lot of interest and people would buy it. So I have had my developers build me a control that shows a representation of my store, but I want to take this representation and extend it, and do some cool things with it. As the representation is built in WPF, it means that developers and designers can work together on the same project and we can integrate the workflow without them getting in each other’s way. So while my developers were building this control, my designers were going and using Expression Blend to work with that control themselves, so do some great 3D rendering in it. So I can go in, zoom, have a look at it, I can move it around. My designers have created a timeline that will take this representation and turn it 90 degrees for me. So all I need to do is go in here and add my zoom task, and now I am going to go in and save my documents. If I go back to Visual Studio we are told that the designer has modified the asset outside of the space so I can click on yes, I can ask it to re‑render the control, and now I have that, my Outlook add‑in and my Visual Studio Tools for Office, take over control of the task. This control is now added to that.

##### Conclusion

Let us have a look at what that would look like inside Outlook. We see the task, when I click on that we see the task, the representation, and if I click on the aisle, we zoom in, look at the aisle, and I can click on the product, and I can pull an item from that product. I can go in and stick quince paste in this location, knowing as the store manager that this is a highly trafficked area. What I have shown you is how you can build workflow enabled applications, you can debug and integrate with SharePoint Server, you can build Office applications easily with Visual Studio, and you can integrate the designer and developer workflow across all these different asset types inside an corporation.

Soma Somasegar

Tony has just spent the last few minutes showing us some of the different things you can do with Visual Studio and .NET Framework 3.5. I am hoping that a lot of you have already had a chance to play around with Visual Studio 2008, if you have, great; if you have not, now is the time to pick it up, give it a try, and see what kind of things this toolset and framework enables you to do

So far I have spent a little bit of time talking about Visual Studio 2008 and .NET Framework 3.5, and we have had Tony show you some of the things you can do with this product. Now I thought it would be interesting to take you behind the scenes and show you the things that we did in the making of Visual Studio 2008 and .NET Framework 3.5

[Video presentation]

###### Progress

Although there is some humour, there is some truth to that, which I want to spend some time talking about. For the last nine months, starting around January/February of this year, we have been self‑posted on the Team Foundation Server, which we are delivering with Visual Studio 2008 and our entire development process has been running on top of that. For me personally, one of the things I found value from was getting insight and transparency to knowing where we are with the project at any given point in time. That has always been a pain point for us.

###### Visibility

If you look back to Visual Studio 2005, it took a little longer in terms of when we wanted to deliver. Part of that was getting transparency into the project. When you have a complex and large project, sometimes that can get tricky. In hindsight, thinking and looking at it now, you would say that it looks like we did a good job delivering this project with the right kind of features to a good quality and in a timely manner but, along the way, there were a number of road humps we had to ride over. The only reason we were able to do what we did is because we had good visibility into the project at all stages, knowing where things were, where they needed help and where we needed to apply more resources and do some scoping. Doing those corrections along the way was a huge part of us being able to deliver Visual Studio 2008 and .NET Framework 3.5 on time. In some sense, my philosophy is that, when we are building tools, we want to be using what we ship to our customers. What we ship is what we want to use internally for Microsoft product development.

##### Partner Ecosystem

###### Collaboration

As much as we have great products and technologies, as part of being a platform company we always have to think about how to have the broadest partner ecosystem working hand in hand with us. We deliver a platform, a set of offerings, but I fully expect our partners from around the world to be able to build additional functionality so that, between what we have and what they have, we have the most comprehensive offering for our customers. As of today, we have over 200 partners delivering more than 2,000 products that are add‑ins, plug‑ins and work on top of Visual Studio to provide a broad set of functionality. To ensure that our partner ecosystem continues to be strong, effective and successful, we are making a couple of changes to our partner programme.

###### Licence De‑restriction

In the past, we used to have a licence restriction that told our partners, if they were using Visual Studio, to use that ID and toolset to build applications that target the Microsoft platform. We have heard loud and clear from some of our partners that they have genuine business needs to target additional platforms in addition to the Microsoft platform. I am excited to announce today that we are going to remove that licence restriction from Visual Studio so that you use Visual Studio and Visual Studio Tools to be able to build an application that targets not just a Microsoft platform, but any other platform of your choice. That is a huge step forward for our partner community.

###### Access to Source Code

The other thing I want to announce today is that, for our Visual Studio Industry Partner (VSIP) Premier partners, we want to make access to the source code available so that, as you are designing and building an add‑in for Visual Studio, you will have the choice to be able to understand how Visual Studio works and, more importantly, can debug your add‑in in the most effective way. To the extent that we can give you source code to help you be more successful building on top of Visual Studio, we want to enable that for you all.

I want to take this opportunity to invite Dan Fernandez, who is a lead product manager in the Developer Division, to show you some of the power of the Visual Studio experience as we take this toolset and enable developers to target additional platforms.

## Product Demonstration: Visual Studio

### Dan Fernandez

#### Lead Product Manager, Microsoft Developer Division

As we talked about, the Visual Studio Shell is the ability to bring the Visual Studio experience to any platform. One of the examples we have taken is an open source project, which will be on CodePlex when Visual Studio ships, called Add‑on Studio for World of Warcraft. World of Warcraft is a game built by Blizzard Entertainment. We have brought the Visual Studio experience to World of Warcraft. Today, this is the experience that add‑on builders use to build add‑ons to World of Warcraft to play the game: Notepad. I think we can do better. Add‑on Studio for World of Warcraft has custom images, a design surface, project systems, a properties window and a toolbox, which is the experience we are used to as developers. We are going to build an add‑on called the opponent analyser.

Soma Somasegar

How long did it take for us to put together this Add‑on Studio?

Dan Fernandez

This was built by one of our partners, EPAM, in just a couple of weeks by two developers. It is because we are using the foundations that you do not have rewrite all these different features. Project templates, item templates and all those things are built, then you can just customise it to your specific domain.

The opponent analyser add‑in that we are going to build, as I change targets, will let me know whether it is something I can kill or something that is going to kill me. I am also a Halo fan. As we kill characters, we are going to play some MP3 files from Halo. Let me show you it in action.

Notice we get a visual designer and SnapLine. This texture is going to represent the image we change. We can set properties like the image filename. Notice it dynamically pulls in whatever images are there, so we get a drop‑down to choose images. If I click ‘view code’, I can show you that, underneath the covers, that design surface is still building that same XML file. In fact, we now have IntelliSense on that XML file, but it still makes it that much better than Notepad. We also have smart tag. We can make our frame moveable so that, in the game, people can easily move it. You can build this on any of the different toolbox controls.

We can double click to go into the code. There are a couple of different things we do when the frame loads. We run an event for when we change the target to say whether we can defeat the person or not, and then we also listen to the hostile player death event. The function we are going to show is this one called ‘set texture and sound’. Depending on how many people we killed, we have a count. If we kill two, we are going to change that texture we saw to a double kill image, and then play the double kill MP3 file.

I will add a bit of code here. As I click, you see we get IntelliSense for a totally different language. I will say if the count equals three, choose that texture, set it to triple kill and also play a different sound file. In this case, we are going to choose the sound file ‘Kilimanjaro’. Let me save our file. I am going to jump live into World of Warcraft live to show you this add‑on. Our add‑on has loaded. Here we are in the game. I am going to target something at the same level as me. You can see a question mark show up there in our opponent analyser. We can pick a couple of targets. This one shows it is a low enough level, so we should be able to kill it easily. We can kill this one too; there we heard the double kill audio. Let me try one more here to see if we can get the triple kill. There we go. I think it is time for our victory dance.

We can build additional and even more complex add‑ons. There is an RSS reader built into the game so I can now read your blog, Soma, directly from World of Warcraft. Once we have email in here, I will never have to leave. This is all uses .NET code that converts into a Lua data structure. These projects will be available as open source projects as well. This project, Add‑on Studio for World of Warcraft, will be free open source and gives you everything. Whether you are a World of Warcraft player or an independent software vendor (ISV), it gives you an intelligence language service and everything you need to build your own custom integrated development environment (IDE) using Visual Studio.

## Popfly Explorer

##### Broad Appeal

Soma Somasegar

As I mentioned earlier, with the Visual Studio Express Editions, we wanted to broaden our appeal for Visual Studio to a set of beginner developers, hobbyists and enthusiasts. The other thing we have done in the last year is to come up with a new tool called Popfly, which enables you to pull together mash‑ups and websites, and be able to customise your application in a simple way, using a visual set of tools without having to get into programming. We made Popfly available by open beta last month, 17 October. In the last few weeks since then, about 100,000 people have signed up for the beta program and started to build some creative stuff, giving us feedback about Popfly, which we can use to make it more useful and effective for our customers. Today I want to announce the availability of Popfly Explorer.

Think about this as a new add‑in to Visual Studio that lets you do two things. The first is that you can use Popfly to create a Silverlight gadget, which you can use on your web page. Popfly Explorer helps you to do that in a very easy way. The second thing it does is it lets you take the web page you have created and then publish it through Popfly, so you can share it with your friends and family. That is a great add‑in to Visual Studio, which leverages a part of Visual Studio and still allows you to do the kinds of things you want to do. More importantly, if you have ever been interested in making your website look cooler, that has become a lot easier with Popfly Explorer and the integration between Popfly Explorer and Visual Studio. Dan will show us what Popfly Explorer can do.

##### Product Demonstration

Dan Fernandez

We now have Visual Web Developer Express open, a custom HTML page, and are using what is called Popfly Explorer Beta, which was made available to download today. What it does is enable me to add any Silverlight gadget on Popfly to my website, as well as be able to publish HTML, CSS, AJAX‑enabled applications to Popfly. I want to show you Popfly, then we are going to build a couple of Silverlight gadgets to decorate my web page.

Let me create my mash‑up. I am going to use live image search. For those of you not familiar with Popfly, it is an online web‑based tool, built in Silverlight, which enables you to build applications, gadgets and so forth. I am going to add live image search on to a designer, will zoom into the block, add ‘Warcraft’ as our search term, then zoom out. I am going to click this light bulb, which shows me image display blocks that can connect to this. To be able to build my custom Silverlight application, I do not need to write Silverlight; I can use these pre‑built templates and then connect the results of the live image search web service directly to that Silverlight gadget. Let us see what that looks like. There we are retrieving images dynamically from the web. As this is a Silverlight gadget, it does dynamic resizing as well. I will save this as ‘my new gadget’, go to my project page and share it. We are going to go back into Visual Web Developer Express and then add this to my page. Back here, we click refresh and I see my new gadget. For me to add this here, I can just drag, drop and resize the container. It is automatically marked as a transparency, even though I have a complex HTML layout. The Visual Studio designer understands it and I can embed these Silverlight iFrames.

I can also choose any projects from my friends on Popfly. Here is one of my friend’s projects. You can see the properties for these projects. I can do things like rate them. I will just add it to the design surface, then click ‘save’ and view it in the browser. We will maximise this. It is making a dynamic web request, and there we have our Silverlight gadgets on the same page. We can reuse JavaScript and CSS. I have already published this on my website so, if you are looking to publish your web pages’ AJAX applications, including Silverlight, you can do this easily and use a project running live on the web.

Soma Somasegar

Popfly Explorer is available today in beta for you to pick it up and try it out. If you go back and think about our mission and how we want to provide a comprehensive set of offerings, as well as the other sets of products, technologies, runtimes, tools, platforms and applications we have for you as developers to be able to build the kinds of applications you want to build, this is an amazing story.

No matter what kind of developer you are, what your skill-set it, what kind of application you want to build, what kind of platform you want to target, we have really got a broad, rich array of offerings that you can use today.

This is exciting because it is already available. If I take a look forward and look at what is coming down the pipe in the next month, next six months, next 12 months, next 24 months, I am even more excited about what we have coming in the pipeline. The amount of innovation, the kind of products that we have got underway is just mind-boggling for me. I am excited to be part of Microsoft and the development division at this point in time.

As I mentioned before, we are putting the finishing touches on Visual Studio 2008 and .NET Framework 3.5 and will ship that later this month. Moving forward we are already starting to work on the next version of Visual Studio Team System, which we have code named ‘Rosario’. We have delivered a CTP of that and we will deliver a few more CTPs in the coming month, but we are well underway in terms of starting to design and code that product. We are also working in parallel on the next version of Expression Studio, which is the first time that you will be able to take that toolset and be able to target Silverlight, and build in both media as well rich internet application scenarios.

The next version of Silverlight that we are working on is where we are going to bring .NET programming into these platforms that you can reuse your skills, your programming knowledge and expertise to this platform. If we look at all these things, plus all the server products between SQL Server and Windows Server and the web server, we really have a great set of products targeted at people who care about the application platform. People who want to build in a great application for their personal use, to enhance their productivity, to be able to run the next generation business. We have a fantastic slew of offerings coming down the pipe in the next few months and the next couple of years.

I want to spend a minute on Rosario, the next version of Visual Studio system. This is going to be the next big step on our journey to providing an end‑to‑end set of tools, that truly makes the whole team, and in some cases the whole IT organisation productive. If I had to summarise the key themes or key pillars of Rosario, there are two things that come to my mind. One, I really want to enable you to build the right thing. When I say build the right thing this is what I mean. The development group that I run, very much like any development group around the world, I have a finite amount of resources and a seemingly infinite set of tasks. I always struggle with having my focus on having my people work on the right priorities. That is always a challenge that any software group goes through.

Being able to have a rich way to prioritise your features, requests and things that you want to work on, understanding which resources are working on which priorities and being able to track that during the duration of the project, is something we have heard loud and clear from our customers. We are going to provide a deep level of integration between Project Server and Visual Studio Team System and Team Foundation Server in Rosario. We have already started doing some down payment on this, with the work that we have done, both on Visual Studio 2008 as well as some tools that we have provided in our ops cycle, which enable certain levels of integration between Project and Team Foundation Server. Some of that work is already available on CodePlex along with the source code for the community to start up, take a look at it and play with it. That is a big investment area for us in Rosario.

Once I help you decide to build the right thing, I also want to help you figure out how to build it the right way. I want to provide a much more comprehensive set of testing tools than what we have today. I can see a lot of things that we can do here, we can provide you with a way to do test case management, a way to track manual testing, a way to do rich client functionality testing, a way to do stress testing, a way to load testing, a way to do digression testing, a way to do code coverage analysis, a way to do test case prioritisation. I can see a whole slew of comprehensive test tools that we can do, that enable you to build things in the right way.

Those are the tools that are big areas of investment that we are thinking about doing in the Rosario timeframe. If there is one thing that I want you to remember about Rosario it is that the toolset is that we want to help you build the right thing and build it the right way. I want to quickly summarise the set of things that I talked about earlier. The big thing is that Visual Studio 2008 and .NET Framework 3.5 is going to be available to you later this month. We are doing a number of things to enable our partners to be even more successful, both in terms of removing the licence restriction so that you can use Visual Studio to target Microsoft platforms as well as other platforms of your choice.

As well as giving our VSIP premier partners access to the source code for Visual Studio ID so that you can do a better job of being able to design and develop your add into Visual Studio. We have got a variety of additional tools and offerings that we are making available, either in beta or a community technology preview to help you use those technologies and build in a cooler, richer functionality into your application. That is a summary and you can go to *microsoft.com/vstudio* and be able to look at this information and pick up the bits that you care about.

I do want to close my thoughts with one slide. I think about my job as delivering a set of technologies, a set of products, a set of platforms to enable developers to be successful. That is the bread and butter of what I do. Then I think as part of doing that, there are a couple of core things that are near and dear to me, that I really think as my commitment on behalf of Microsoft to each and every one of you.

When I think about it that way, there are a couple of things that come to my mind. First and foremost, I am a big believer in being transparent and open with you all. The more transparent I can be, the more I can get you engaged and involved with the development processes right from day one. I can share with you my designs, I can share with you my early code, and get feedback from you in a timely manner, and I can react to that feedback. I feel that that is the best way for us to be able to have a partnership and relationship and know that we are building the right products for our customers.

We will continue pushing the bar on this. I need your help and participation in making sure that we are doing the right thing here. As part of that being transparent and having the two-dialogue, I want to make sure that there is a robust mechanism, in which you as a customer or a community member can give us feedback that flows directly into the product team, and know that the product team can reach our and respond to you. You can talk to the product team through blogs, you can go to the feedback system of the connect system and be able to give feedback and know that the person who is building that feature of the particular technology gets your feedback directly, and hopefully they will turn around and have a conversation with you. Enabling a two‑way dialogue, and a good feedback mechanism is something that I am committed to.

I have already talked about my feelings on having the broadest partner ecosystem and a vibrant and self‑sufficient community. These are a set of commitments that I want to take on behalf of Microsoft, which each and every one of you. I do need your help, your cooperation and your partnership in delivering on these commitments. If I have to sum up these commitments, I think about these as a way for us to learn together, for us to grow together and for us to be successful together. With that, I do want to close this. I do want to thank you again for giving me the opportunity to get a change to talk to you this afternoon. I do wish you a great next two days here at TechEd and hopefully you will see some of the good things the city has to offer. Have fun, and thank you.

**This Full Transcript was produced by Ubiqus (+44 (0)20 7269 0370)**