

BRUNEL: Welcome to another edition of Getting the Most out of IBM U2. I'm Kenny Brunel, and I'll be your host for today's topic. Today, we'll take a look at IBM U2's implementation of Microsoft's .NET framework.

To get us started, we should learn a little more about what .NET is. To help us with this today, I'd like to welcome a couple of guest speakers.

My first guest has been immersed in the U2 way of life for over 10 years, Ray Else is part of the U2 engineering team and he's been involved most recently with the U2 .NET projects. Ray joins us on the phone today from Dallas, Texas. Ray, we've got a lot to cover today, so I'm going to jump right in.

I'm sure our listeners know that .NET is a component of the Windows operating system, but can you give us a little more detail?

ELSE: Sure, Kenny. Two development camps exist in the business programming world today. You've got the open systems development and Microsoft Tools development.

With open system development, you're talking about development for all the OS platforms which includes Windows and you're talking about languages such as Java and PHP and

you're talking about tool sets of servers such as IBM's Rational App developer or IBM's WebSphere App server and the Eclipse framework.

Microsoft Tools development is typically intended for Windows only, and .NET is Microsoft's latest state of the art development and application server framework for Windows.

It rides on Windows, closely integrated, but the good thing is that it isolates itself enough from Windows so you typically won't see your black screen of death when your .NET flashes as we used to when you had strictly Windows tightly integrated programs running and crashing.

The languages used in .NET are typically C# or VB .NET both of which turn into a common byte code that the .NET framework works with. And the nice thing again that the .NET framework manages the memory requirements of your programs keeping your programs from stepping where they shouldn't.

When it comes to .NET apps typically using Microsoft's Visual Studio toolset, a very rich environment which allows add ins and supports APIs that U2 takes advantage of with its new tools and plug ins.

Generally speaking, the .NET development is very programmer friendly because you have an environment which is intuitive, it's very quick and easy to publish apps to Windows since it's written specifically for Windows. From a maintenance point of view, you really end up having a very cost effective solution with the .NET application.

BRUNEL: Okay, thank you, Ray. It sounds like there are a lot of different options for .NET and likely more than one implementation that our partners could take advantage of.

I'd like to turn now to my next guest, Michael Burn.

Michael has been involved with U2 for over 10 years and he currently works as an IT specialist for IBM.

Michael, thanks for calling in today. Would you mind telling our listeners about the different options available for U2 developers when it comes to .NET.

BYRNE: Sure, Kenny thanks for having me today. Just as a side note I guess, all the technologies that we're going to talk about today will work with any of the .NET supported languages. So we're talking about VB .NET or C# and any of the topics that we kind of talk on could either be just as well used as Web or Windows development.

So I guess I'll talk about three of our APIs and tools and

data providers that we've got today both being IBM .NET, U2 .NET and uniobjects .NET mostly is what we'll focus on.

So IBM .NET is actually called the IBM Data Server Provider for .NET and we call that IBM .NET for short. What this is is simply a managed [ADO] .NET data provider. So applications can use the ADO .NET provider to connect to universe and uni data and retrieve and manipulate and update the data seamlessly.

So ADO .NET also supports world connect as a disconnected access. So that's nice for as far as the scalability of applications in a multi tier environment you can deploy that disconnected access easily.

A second part of that toolset for IBM .NET is that actual addins for Visual Studio. The addins actually seamlessly integrate that UNIX data source into your Visual Studio environment. So you can build Windows, Web or Web services applications without running a single line of code if you wanted to.

[INAUDIBLE] allows you to execute the U2 basic subroutines right in the server explorer. You can actually view your U2 tables and views in the server explorer in Visual Studio so that you really have a seamless integration with the Visual Studio from itself.

Secondly, I guess we've got U2 .NET. U2 .NET is, also provides a tight Visual Studio integration without that whole rapid application development environment. And it's pretty similar to IBM .NET in that it's a data provider that you would actually just ties your data access to Visual Studio but this one is more multi [INAUDIBLE] aware.

So in that you don't have to do any seamless flagging steps and it's a little bit more multi value centric. So you can view, again, do the tables and subroutines and do drag and drop and all those same type things. So it's a similar thing, but it's a little bit more [INAUDIBLE] value centric where ADO .NET is a little bit more standards based with the SQL in relational databases.

Third one I guess I'll discuss today is the [ginni objects] .NET. now this one is actually a API that we've got. So it's a set of .NET classes that provides you the functionality to retrieve and update the U2 data servers.

This is a native interface into uni data or universe. So you're getting a really fast transaction here. You get things as far as reading and writing files, doing locking, while you're doing that working with dynamic arrays, executing subroutines and TCL commands and some other nice things that the uni XML objects which works really well with

data mining and do data grids and things like that.

So a lot of objects, .NET native objects, these are 100 percent .NET managed code that you're integrating with there which is compared to the older com uni objects that some of our listeners might be familiar with.

That actually does bring up another point. .NET actually does support com objects through its interop interface. So that also means that you can use Web development environment.

So this could be quite handily knitted into a .NET framework usually and we've got developerWorks articles and things that you can actually look into Google-ing [red bag] with .NET and find developerWorks articles for that.

So the development environment is actually more of a tool than any of the other ones that we talked about which are the kind of plug ins and APIs that actually will allow you create Web pages and it generates the .NET Web pages for you and [wrapper] classes for the actual [red bag] objects and all that so that you can work natively in .NET.

BRUNEL: Michael, so of these different tools that you mentioned, I know that uni objects for .NET and U2 Web development environment have both been available for a

while. When will, if they're not already available, U2 .NET and IBM .NET be available?

BYRNE: As you mentioned, obviously uni objects for .NET is available now and listeners can find that on the client CD as a free download but you want to make sure that you've got uni data 6.1 or universe 10.1 or greater to do those.

Some of the other providers, ibm.net is available now as of late 2007 so we're in early 2008 right now and it is available now. It's actually under the DB2 downloads.

So the easiest way to find this is there's a link in the developerWorks article that I think will give some directions to where to get some of these developerWorks articles entitled Access IBM U2 Data Server From Your .NET Applications Part One.

And there's a link in there right to get the actual IBM .NET data provider in that. And you just want to make sure you're on uni data 7.1.11 or universe 10.2 or greater if you're going to use those technologies. U2 .NET, the multi value centric form is actually not out yet; its version 1.1 is expected to be released in 2008. So look for that soon.

And again, you're looking at uni data 7.1 or universe 10.2

or greater. So those are just some concerns, prerequisites that you'll need before that.

BRUNEL: Well, it sounds like again there are a lot of different options becoming available for .NET on U2. now I'd like to take it back over to you, Ray, I've heard you tell of some demos and tutorials that work for .NET and U2 on .NET. What can you tell us about that?

ELSE: Now, Kenny, I've actually been involved with putting together a couple of tutorials and articles myself working with the engineers.

We've realized that developers really need quick start, code samples, tutorials that walk them through how to really get started doing the basic building blocks of an application using these toolsets, working in the Visual Studio environment. And so we've really made a push with these products to get those sorts of things out there.

Personally, I've worked on a quick start tutorial for ibm.net. it should be on the developerWorks site sometime this year. I know that [Rayjon] has already got one of the articles out that Michael mentioned. And he's working on part two of that same series and he's got several other parts that he's planned.

I've worked with him on a couple, one in fact covering data sources. And in terms of U2 .NET too I can say that [Gopa] talking with him, he's our lead programmer working on that project and he's been really pushing to get some tutorials and articles together as well.

Again, our priority is to not only deliver these tools and we don't want to just throw them over the fence. We want to throw them over the fence along with some excellent tutorials so that people can really get started quickly.

BRUNEL: That's great. So it sounds...I mean, you guys have gone over the number of different tools we have available and also tools and tutorials and articles and so on that are going to be available for our users to get up to speed.

Now I'd like to look into a little bit more deeply and Ray I'm going to come back to you on this on what advantages .NET is going to have over other development frameworks.

ELSE: Well, really what you're looking at of course is development frameworks for Windows. That's the key here. And it was written by Microsoft specifically to run on its Windows platforms. It's going to be really a premiere choice here for developing applications.

I like to think of it as kind of programming by the masters, for the masters in the sense that with these toolsets that you have it's very quick and easy to develop applications using drag and drop, wizards, data binding of data from your database directly into grids and drop downs and the like.

So it's quick to throw together applications, good for applications that are world class and of course, your target audience is this Windows desktop environment typically where there's millions of them out there and also you have a lot of servers running Windows operating systems today.

Finally, you've got this very rich support network because you have so many .NET programmers out there, many of who created their own FAQ support sites. You've got also the Microsoft MSDN site to go to when you have questions. And just lots of application examples and answers to issues out there.

So when you're talking .NET, you're talking about a way to extend your applications and your technologies to create powerful, cost effective solutions for Windows.

BRUNEL: Okay. Well, my next question I think could actually be addressed by either of you but I'm going to send this one back over to Michael. Michael, what else could developers do to help take their .NET applications to the

next level?

BYRNE: Yes, that's a good question, Kenny. Actually, you know, [U2] you get to so far to be able to find out a little bit about how to connect to the data and how easily it is to actually interact with the DB2 servers. But what you really want to look at to make it a world class business application is what do I do to get to the next level.

My suggestion would be I'd go out to the ASP .NET development site at www.asp.net. So out there as Ray kind of alluded to before I mean there's tons of white papers and actual like videos of people doing specific things on practical type tasks like data binding and how to build Web pages and parts of it and do that.

So you can use other techniques but they do a very...since we've provided these APIs and data provider tools, you can simply plug ours in there and start working with that data in the same way. So I think an example would be like how Ajax enables your Web site. Our technologies definitely would work with that because they're simply .NET managed code. So you're working with it just like any other code.

And they have a whole section on Ajax on there and how to use the Ajax toolkit that Microsoft's got and actually works pretty well to be able to leverage that thing.

Another thing I think I would mention would be taking a look at Web services, the up and coming common language that allows a lot for interoperability. So some of these legacy systems and needing to maybe integrate with a couple of other things in your enterprise.

U2 itself, we've got our own Web services developer tool which is a great drag and drop tool that creates Web services from your TCL queries or basic [INAUDIBLE] or even that XML DB files. So you can actually look at all your files or sub routines on a thing, drag and drop it over onto a Web service and it's going to create that interface for you and allow you to deploy that simply.

On top of that even if we're talking about the [dot] network you can actually use the Microsoft Web services wizards to actually create these Web services for you and then you'd go into the code behind pages and then just simply do the coding in there and actually leverage it that way so that you would, you don't need for a separate Web services server running and you can do it all in [AIS].

So there's plenty of opportunities to be able to extend I think the basic concepts that we've gotten once you've got that down in the connectivity down.

BRUNEL: You make it sound pretty easy, Michael. So, you know, I'd like to give Ray a chance to answer this question as well. Ray, maybe expand a little bit more on how our users can get the most out of .NET with U2.

ELSE: Yes, I've looked at two new things that we've been pushing with our uni data and universe database servers. One is XML support. We've added some functionality there where you can select data out and get XML, we've added some [birds] into the basic language as well and that's just a natural fit for .NET environment.

Along with that, we now have service side connection pooling which you're going to need to have a Web application hitting your uni data and universe server. So XML, connection pooling, those are two things to look at.

Finally, you have to make a decision of course of which of these technologies you're going to use. Are you going to use this IBM .NET, the U2 .NET, the [INAUDIBLE] .NET, et cetera. And it's a challenge because one technology might be great for one company and another technology might make better sense for another. And I think I'll let Michael tackle helping us make that decision.

BRUNEL: Okay, well we are running short on time now but Michael let's bring it back over to you and maybe you can

give us some insight on how companies would best choose which U2 .NET technology is best for their applications.

BYRNE: Sure. I'll just wrap it up with that. I think that as Ray alluded to it's not the perfect, what you really need to do is look internally and see your development staff in your company and what's important to that. I'll just address a couple of other things that you might want to look at when you're looking at some of these tools.

So if you're looking at uni objects .NET, it's really the APIs. So it's fast and it's easy to use but it's definitely native access. You're actually working with the actual files and calling server teams directly and it's not a drag and drop thing.

This is more a hands on, those sort of people that want to get in there and actually do the coding. So you've got a lot of control, but you're definitely going to have to get in there and code. That code doesn't have to be difficult.

It could be as simple as the sub routine calls, you have a couple of lines of code and most of your business logic is still in your basic sub routines.

The nice thing about this ATI is there's really no configuration. It's simply a [DLL] you would include in your project to deploy whenever you go. So you're up and

running in no time.

Conversely, I guess when we're looking at [L] group, ibm.net and U2.net together as the data server, data providers, these are more graphical tools. There's some installation required with each of those to get things set up. But then these data servers are more for those that want kind of the drag and drop tool, they're going to work with Digital Studio and integrate directly in there and you can actually get a visual representation of your database and drag and drop and make queries directly from there.

So it's a little bit easier to use and if you want you can deploy things without ever having to do code but you're probably going to want to extend that at some point.

If we're looking at the ADO .NET, you know, its very standards based. When you're looking at the ibm.net ado provider. So its standards based so it goes along with some of the, I guess, any of the other development environments that are going on and all the other data providers.

If you're looking at ibm.net, there is schema set of requires for the accounts if you're going to use SQL commands. But on the other hand, you could also just call basic programs from there and get back XML screens and things like that. So it depends on how much you want to

bite off.

And then U2 .NET you actually don't have to do any schema flattening as I mentioned before because it's so MD centric.

So if you're still want to work in the multi value world and familiar there, you know, U2 .NET might be for you.

If you want to be more standards based and relational database, IBM .NET might be for you. So just a couple of things you might want to take a look at when you're deciding on the development.

BRUNEL: Well, that's a lot of great information. And gentlemen, I'd like to thank you both. Michael Burn, thanks for calling in today. And Ray Else, appreciate your time as well.

I would also like to take the time to thank our guests for listening in today. We hope that you found today's podcast useful. Be sure to send us e-mail with your feedback, any comments or suggestions you may have for future episodes.

Our e-mail address is U2askus, that's u-2-a-s-k-u-s at us.ibm.com. You can also provide feedback directly on this podcast from where you downloaded it. And transcripts of today's podcast which include all of the relevant links are available on our Web site by following the U2 podcast link

from the main page.

You can get to our main page by connecting to www.ibm.com/software/U2. And I'd also like to point out that getting to developerWorks and finding the downloads that Michael mentioned earlier you can get directly to developerWorks from the main U2 Web site.

Be sure to stay tuned in for future episodes of How to Get the Most out of IBM U2.

[END OF SEGMENT]