

BRUNEL: Welcome to another episode of Getting the Most Out of IBM U2. I'm Kenny Brunel, and I'm your host for today's episode, and today we're going to talk about IBM U2's latest technology, U2.NET.

First of all, what is .NET? So I'm going to turn my attention, first of all, to a guest that I have in the studio with me today in Denver, Dave Peters. Dave is the product manager for the U2 data servers and the client tools, and Dave has been with IBM for over 12 years.

What can you tell us about U2.NET? But first of all could you give us a brief rundown or a description of what .NET is itself.

PETERS: Sure, Kenny. .NET is really a development framework architected by Microsoft. This is the framework that Microsoft hopes that will be the development choice for any new development on Windows. It consists of user-defined interfaces, data access, database connectivity, cryptography, Web application development and communications.

IBM has three options for U2 developers that want to access the U2 data servers using .NET. The first I'll mention is [UniObjects] for .NET or UO.NET. UO.NET is a MultiValue API for use in the .NET applications. It's very familiar within

the constructs and was introduced to help Basic programmers get to .NET very easily.

And one good thing is it's simple to get. UO .NET is available on the client CD that comes with either UniVerse or UniData.

The second option is, the long name is IBM database add-ins for Visual Studio or as we refer to it as IBM .NET. IBM .NET is an ADO .NET provider that supports four of IBM data servers including UniData and UniVerse. And it is available at no charge, it can be downloaded from the IBM Web site.

The third option and the focus of our discussion today for .NET developers is U2.NET. U2.NET is the result of our agreement with BlueFinity to leverage the MV.NET technology.

IBM has optimized this product for use with the U2 data servers.

As with IBM .NET, it is not an application development tool but can be leveraged from Visual Studio. Like UO .NET it understands MV constructs such as accounts, files, dictionaries and MultiValue fields.

U2.NET is available for a reasonable charge and you can talk to your software partner or your IBM representative for details about that.

BRUNEL: All right. Thanks for that introduction, Dave.

And I do have one point of clarification. I'm wondering if each of these are ADO .NET providers. And I'm actually going to turn this question to my next guest, who is [Gopa Padmanabhan]. Gopa is the lead developer for U2.NET and he's been developing in U2 and on the Windows platform for over seven years. So Gopa, first of all, what's the appropriate application for each of these .NET tools that Dave mentioned?

PADMANABHAN: Yes, Kenny. If UO.NET or UniObjects .NET is a pure MultiValue type API, it's not an ADO.NET provider. It is just a MultiValue API, you can download MultiValue type applications, you have access to MultiValue features like accounts, files, dictionaries.

The ADO.NET provider by U2 is the IBM .NET product which is a standard SQL provider. And U2.NET also has a component which is ADO .NET provider but it is a lot more than that. It has primarily a lot of MultiValue functionality which allows you to do MultiValue style application development using APIs or MultiValue style drag and drop through Visual Studio like application downloads.

BRUNEL: Okay. So Dave had mentioned that we had actually taken this code from another product and we have

enhanced it, we've added a lot to it. So can you, Gopa, tell us a little bit about what you've done with that code?

PADMANABHAN: The original product, the MV .NET supported many MultiValue databases, and UniData and UniVerse were among them. So we identified by looking at this product a lot of scope for improvement mainly in the areas of performance, simplification of the usage of the product, also the implementation of the product and also on consumability, how fast you can get up to speed developing using U2.NET.

On the simplification front, we now support UniObjects .NET as the exclusive only middleware that allows for a lot of simplification in the configuration of the product and also the use of the product.

Also the fact that we are supporting just UniData and UniVerse as the back-end databases means that we can simplify the product quite a bit, also contributing to the performance.

So on the performance front one of the major improvements we have done is to support the U2 connection pooling which leverages connection pooling on the server side. And we have done some tests internally here, and we show that if you use the U2.NET connection pooling you get about 400

percent performance improvement over regular the connections.

So that's very important thing to look at if you are using and deploying this product in a multi-user application where either a Web scenario or a Windows application scenario where you have a lot of simultaneous users coming in, the U2 connection pooling is going to give you a large performance improvement.

The other area where we have done considerable improvement is consumability. We have made the product a lot easier to install and use. The licensing is a lot more streamlined now and it's simpler.

Also, there are tutorials with the product either for Windows application development or for Web development which takes you step-by-step through developing an application from scratch and Visual Studio. They will get you up to speed on development on Visual Studio very easily.

On the functionality front, we concentrated our effort on improving the Web functionality of the product, and as of now we can support the quality functionality on the Web site that we support on the Windows site.

BRUNEL: Well, I have another .NET expert with me in the

studio today, Michael Byrne. Michael is an IT specialist with IBM, and he has been actively using .NET on a regular basis for over six years.

Michael, I'd like for you to assume that I'm a user and I've just purchased U2.NET. And Dave and Gopa have both touched on different ways you might use some of these .NET tools, but what can you tell us specifically for U2.NET?

BYRNE: I think in general I think what we've talked about before is obviously U2.NET has a Visual Studio plug-in that we've gotten into the product that allows you to basically use your Visual Studio development environment but now be able to natively view our data sources just as you would any of the other data sources such as SQL, SQL server or Oracle or any of those.

So the nice thing you get from that is your same development environment that you're used to as a .NET developer. But you've got the additional thing of wizards, I guess, that the product has what you can drag and drop from files or the structures to start building forms, either both Web or Windows applications.

So with zero coding, you can go down to, you know, several levels of MultiValue attributes and tie those all together and quickly build a complex application basically not

touching any code.

The nice thing about this that Gopa mentioned before also is there's no SQL setup required here as with our more standard IBM .NET provider which needs to do some schema manipulations that would make sure that it knows more about relational data source. So you don't have that extra step from that standpoint.

One of the other nice things that comes with the product itself is it's got its own self-generating demo database that you can then just add a bunch of records and do that so you get some actual data, live data to work with and play with.

On top of the wizards, the drag and drop wizards that we've got in there, there is also more of an ADO .NET standard programming API that you could use if you want to get in and actually do some .NET coding. That's another aspect that you can go and actually get into the code and do it from there.

Just from a usability standpoint a lot of people always ask me, how many licenses do we need to run this product? And there's no hard and fast answer because it really depends on what you're doing on the back end and make sure your accounts are sized correctly and things like that.

But from a standpoint of this, you know, a Web architecture in a way that we can use the connection pooling and other things that the product has in there, you don't have a one-to-one relationship.

So if you say you've got 100 or 200 users, you don't need that many database licenses. With the architecture you're going to get away with just probably a handful of licenses to be able to manage that load because of the multi-tier architecture of the actual product itself. So that's kind of a nice benefit.

BRUNEL: Gopa, how would you say that you U2.NET differs or stands out from these other .NET development tools?

PADMANABHAN: Kenny, you know that Visual Studio has some great features when it comes to developing, you know, standard SQL based applications. They have all kinds of wizards, they have tools which allow you to design the schemas, design the application if you are developing against a standard SQL database like SQL Server or Oracle.

With U2.NET, now the MultiValue developer has the same functionality in Visual Studio. They can drag and drop to do their application development, they can tie their files to a Windows form or a Web page and develop a MultiValue

nested application from scratch.

And as Michael mentioned, most of the time you can develop a multi-level nested application without touching any line of code. So that's one of the major strengths that U2.NET brings to a Windows MultiValue developer.

The other advantage of U2.NET is its deployment flexibility due to its multi-tier architecture. You can have the U2.NET middle tier on your application machine, your Web server, the U2 data server or on a dedicated middle tier server altogether.

Because of this flexibility of deployment, you may achieve better performance and use fewer U2 licenses than you may be able to achieve with other U2 client products intended for use in .NET.

Also, you can use the flexibility of connection pooling or not use connection pooling based on your load requirements.

So those are some of the major differentiating factors for U2.NET.

BRUNEL: Now we get to the crux of this podcast, which is how one gets the most out of U2.NET. And I want to turn my attention to you first, Dave. Tell us how do we get the most out of U2.NET?

PETERS: I could think of three things, Kenny. First U2 developers really should take advantage of the tutorials that ship with the product. They were designed to get developers who aren't familiar with .NET off to a fast start.

Secondly, developers really should take advantage of the great educational opportunity that U2 University provides. There will be multiple sessions there on U2.NET.

And finally, as Gopa mentioned earlier, developers of U2.NET should take advantage of connection pooling and give their applications a performance boost and to build in scalability.

BRUNEL: Michael, what can you tell us?

BYRNE: A couple points I think I would make. Make sure you probably go grab Gopa's developerWorks article. He's created an article to basically step you through a whole master detail building it from the beginning to end applications. So that's entitled U2.NET - Develop a Native MultiValue Style Master Detail Application. So that's a great resource.

Secondly, this is a fully supported product by IBM, so make

sure you take advantage of our worldwide support offering that we've got obviously with that, if you've have any problems that come up.

A couple other points. Look for the soon to be updated .NET training course that we've got. That's going to include U2.NET along with those other technologies, so that will give you something to be able to compare that and contrast that to all the rest of our .NET technologies.

And lastly, I think we're working on a transfer of information, which is a Camtasia screen recording that will again step you through building an application from beginning to end, to help you get up to speed as quickly as possible.

BRUNEL: Okay. Gopa, how about you?

PADMANABHAN: Right, Kenny. You know that Visual Studio is a great development tool, but for a MultiValue developer, the learning curve is pretty steep because you have to, first of all, learn this new Windows development environment. On top of that, you have to adapt yourself to the flat SQL world.

Whereas with U2.NET and the tutorials, you don't have the standard SQL learning curve. Also, the tutorials will step you through developing a non trivial application on either

Windows forms or the Web world. And that's a great way to get up to speed on Windows application development using Visual Studio.

You can also, as you go through this process if you hit a roadblock or something, go to the developerWorks forum and Kenny can give you more information on exactly where you go and share your thoughts on the product or give us feedback for enhancements of problems.

BRUNEL: Okay. Well, I would like to thank each of my guests for joining me today, and we hope that you, our listeners, have found this podcast helpful. We hope you tune in for future episodes.

Transcripts of today's podcast which include all relevant links are available on our Web site by following the U2 podcast from the main page. Our main Web site is www.ibm.com/software/U2.

And from that main page you'll also find links to developerWorks where you'll find Gopa's tutorial article as well as all the other U2 articles.

e-mail us with your feedback, comments and any suggestions you may have for future episodes, and be sure to tune in for future episodes of How to Get the Most Out of IBM U2.

[END OF SEGMENT]