UK MSDN Flash eBook: Best Technical Articles #1

January 2008 to January 2009

Eric Nelson, Technical Editor, Microsoft UK

4/4/2009

The UK MSDN Flash developer newsletter contains great short technical articles written by UK developers both inside Microsoft and in the broader developer community. This eBook pulls these great articles together in one place. This first edition contains 13 articles covering dynamic languages, testing, game development, cloud computing and more.

TABLE OF CONTENTS

msdn
Microsoft®Developer Network

msdn
Microsoft®Developer Network

INTRODUCTION

FROM THE EDITOR

Hello all,

Every two weeks we send an email out to tens of thousands of UK based developers. This email is called the MSDN Flash. The Flash contains many useful sections including a 400 to 500 word technical article on a developer related topic either written by a member of the Microsoft UK technical team or a member of the broader UK developer community. We have had some great articles over the years which to some extent end up "hidden away" in the archives of the MSDN Flash. This is a shame as the authors have worked hard to condense complex topics into short articles which are informative and take only a few minutes to read.

I decided it was time to surface the best of the articles on a regular basis and provide them in an easy to download and read eBook. This initial edition covers the period January 2008 to January 2009 but we aim to create a new eBook every six months from here on in. The next edition should therefore be out around August 2009.

In the meantime I would love to have your feedback on what you liked, what you didn't and how it could be improved in the future.

Happy reading.

Eric Nelson
Developer Evangelist, Microsoft UK
Email: ericn.nelson@microsoft.com
Blog: http://geekswithblogs.net/iupdateable/Default.aspx
Twitter: http://twitter.com/ericnel

WOULD YOU LIKE TO BECOME AN AUTHOR?

Developers value the sharing of best practices, knowledge and experiences. The MSDN Flash is a great way for you to do just that with thousands of other UK based developers.

The topics best suited to the MSDN Flash are topics **you** think a **lot** of UK based developers using Microsoft technology would like to know something **more about**. This makes for a pretty broad potential set of topics! Maybe you have something to share around SharePoint development or Test Driven Development or perhaps you have war stories around configuring WCF for interop or using Object Relational Mappers to increase productivity. All are great candidates for becoming articles in the Flash.

Please email us with your proposed article(s) and if possible a "sample of your work" such as a link to your blog.

Email us at: UKMSDN@microsoft.com.

SUBSCRIBE TO THE UK MSDN FLASH

The MSDN Flash contains a lot more than the technical articles. It is designed to help you keep track of all the latest news on developer tools and technologies which are relevant to you. This fortnightly newsletter pulls together the most recent technical news in a concise and easy to read format so you can get all the latest information within a few minutes.

The UK MSDN Flash newsletter includes:

- **UK events and training conferences:** find local events, community meetings, training courses and more

- **Feature articles:** these articles cover the latest developer topics, real technical solutions and Microsoft insights
- **Fresh news, announcements and updates:** get the latest Microsoft updates, product announcements and product beta news
- **Recommendations and insights:** key highlights, resources, tips and offers available for developers
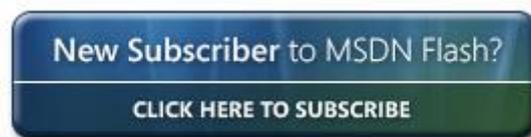
New to MSDN? Want to know more?

The Microsoft Developer Network (MSDN) is a set of online and offline services designed to help developers write applications for the Windows platform. To find out more about UK resources and programmes available from Microsoft, visit the UK Microsoft MSDN website. To stay connected with

the UK developer community visit UK Channel 9 to find screencasts, podcasts and videos from Microsoft UK along with links to UK developer blogs, events and communities.

VISUAL STUDIO 2008

SYNCHRONIZATION MADE EASY BY THE SYNC FRAMEWORK

20 February 2008: Synchronization made Easy by the Sync Framework

In November 2007 Microsoft announced the Sync Framework project. The framework comes to the rescue of an overlooked area: the complex world of data synchronization. This is achieved by merging existing technologies in a common interface, which can be easily extended using new sync providers. The current implementation supports providers for file synchronization, ADO.NET and FeedSync. It is interesting to see how these providers interact with the different type of participants:

Full Participants: These participants initialise and host the sync framework, triggering the synchronization and configuring the providers. The full participant is able to communicate with any other type of participant building synchronization networks. A good example can be a desktop application or a service managing a mesh of participants.

Partial Participants: These participants are passive members that can store information as well as alter it but do not know anything about the synchronization framework. They end up being only repositories that may be available when a full participant requests the synchronization; this means that they cannot participate without a full partner. A good example is a digital camera or an mp3 player.

Simple Participants: These only feed information into the synchronization but are not able to store information. A full participant will request data from simple participants and will synchronize with the other members. An example can be a RSS feed.

The architecture of the framework is divided in two basic components: the first part is the framework's implementation located in the *Microsoft.Synchronization* namespace and the second part is sustained by the providers, implemented in separate assemblies. This model allows the developers to extend the providers to fulfill custom requirements, either using the providers as base code or creating completely new ones. For more on the Microsoft.Synchronization assemblies, please read the online documentation.

Each of the providers can be configured individually using the *SyncOptions* and the *SyncScopeFilter* classes, which provide developers the ability to shape the behaviour, e.g. what files should be included on the synchronization. The providers have the freedom to publish methods to enforce those options and construct the metadata information, which enumerates the changes since the last synchronization. Once we have this metadata setup, the sync framework takes control after we call the SyncAgent.Synchronize() method. During execution the full participant may handle progress events in order to query the status of the metadata synchronization, providing a richer user experience. This process is the same no matter what kind of content is synchronized and it is upon the providers to hide the internals of the implementation.

As I hope you can see, using the sync framework is an easy and powerful exercise that helps you implement heterogeneous synchronization in your applications. For more information, please visit the sync developer centre where you can find an active community already utilising this technology!

Salvador Alvarez Patuel

Application Development Consultant (ADC)
Read Salvador's blog

The Microsoft Sync Framework V2.0 CTP is available for download.

msdn

WMI PROVIDER EXTENSIONS IN .NET FRAMEWORK V3.5

6 February 2008: WMI Provider Extensions in .NET Framework v3.5

Windows Management Instrumentation (WMI) is the management infrastructure for Windows-based systems. Windows Vista ships with 100+ WMI providers to enable the management of various operating system components. Applications instrumented with WMI can be managed using tools like Microsoft Operations Manager (MOM), HP OpenView, Windows Script Host (WSH) and PowerShell. The WMI infrastructure provides a repository in which providers can install their management information. Consumers can query & analyze this management information and then using the providers can manage applications by reading/writing configuration settings and calling methods on managed entities. For more on WMI, please start on the "About" page.

In the .NET Framework v2.0 (in the System.Management.dll) the System.Management namespace provides the functionality for developing WMI consumer applications while the System.Management.Instrumentation namespace provides the classes to write WMI providers. The provider API in .NET Framework v2.0 had some limitations e.g. a managed WMI provider couldn't have writeable properties, and couldn't expose methods. To overcome these limitations and improve the overall provider development experience, .NET Framework v3.5 introduced extensions to the WMI Provider API. The new API/extensions can be found in the System.Management.Instrumentation namespace across two assemblies: System.Core.dll and System.Management.Insrumentation.dll. These new extensions enable a familiar attribute based programming model for writing custom WMI providers and exposes all of the features of the native COM based API to managed code. The new API allows you to create both coupled and decoupled WMI providers. Coupled providers are implemented as a dll which must be signed and installed in the GAC. The WMI service itself hosts coupled providers. Decoupled providers are hosted by your own application which must be up and running to enable consumers to access your management information.

So to turn a C# or VB class into a management class you apply the ManagementEntity attribute to the class itself. Methods in your class are exposed to WMI using the ManagementTask attribute while ManagementConfiguration and ManagementProbe attributes are used to expose read/write and read-only properties respectively. The hosting model of your provider (coupled or decoupled) is specified using the WmiConfiguration attribute. As always, these attributes are fully described in the documentation.

Once you have described your management classes and hosting model using the aforementioned attributes, you then create a custom installer class by deriving from the new DefaultManagementInstaller class. This installer class will be invoked when you run InstallUtil.exe against your assembly and will extract the management information from your management classes, creates a MOF (Managed Object Format) script, compiles it and persist it in WMI repository. This installation process is same regardless of the hosting model (coupled or decoupled) of your provider. After the installation, your WMI provider will be available to any WMI consumer application.

WMI is a great technology and by creating a WMI provider your applications can be managed by any management software and custom scripts in a standardized way. For examples of managed WMI providers visit the MSDN "How To" samples.

Zulfiqar Ahmed
Application Development Consultant (ADC)

Read Zulfiqar's blog

NON-ATTRIBUTED DATA CONTRACTS IN WCF

29 October 2008: Non-Attributed Data Contracts in WCF
DataContractSerializer in WCF follows a strict opt-in model so you need to explicitly specify which types and type members are included in the message. That is why DataContractSerializer expects you to explicitly mark your classes as serialisable and will raise an exception if it encounters any type used in the signature of a service operation that is not marked as serialisable. This rule even applies to primitive types such as integer and boolean but WCF has built-in data contracts for those types so those types will just work. When it comes to your own types however, you need to decorate the type with DataContractAttribute and decorate all of the fields and properties you want to publish with DataMemberAttribute, regardless of their access modifier (private/internal/public). There are other ways of marking your type as serialisable (for example by decorating the type with SerializableAttribute) but we are focusing on the data contracts here.

If you don't decorate the type with DataContractAttribute, DataContractSerializer won't be able to serialise it. This proved to a problem for some customers who have large codebases and want to publish their existing types using WCF. So the WCF team listened to customer feedback and changed the serialisation behaviour to allow "non-attributed" data contracts such as Customer to be used in WCF services with.NET Framework 3.5 SP1:

```
public class Customer
{
    public Guid Id;
    public string Name { get; set; }
    [IgnoreDataMember]
    public byte[] Notes { get; set; }
}
```

When the DataContractSerializer encounters a non-attributed data contract, it first checks to see whether the type is valid for serialisation. For example, the type has to be public and it must have a default constructor. It then says: "I assume you meant to decorate this type with DataContractAttribute. You also meant to decorate its **read/write public fields** and **properties** with DataMemberAttribute". This behaviour is very similar to that of XmlSerializer, which is used in ASMX Web Services.

When you omit the DataContractAttribute from the type declaration, you are effectively changing the behaviour of the serialiser from an opt-in model to an opt-out model. If you want to exclude a member from the message, you can decorate that member with IgnoreDataMemberAttribute, which is only effective when applied to members of non-attributed data contracts. Likewise, DataMemberAttribute is only effective when applied to members of attributed data contracts.

Some developers consider the non-attributed data contracts to be against one of the fundamental principles of WCF, which is to use explicit boundaries for services and this is a valid concern. But remember that this change is primarily intended for backward compatibility purposes so if you are writing a new application, you are advised to not use "non-attributed" data contracts and use WCF's opt-in model instead.

Mehran Nikoo
Application Development Consultant, Microsoft UK

Read Mehran's blog

INTRODUCING THE ADO.NET ENTITY FRAMEWORK

2 April 2008: Introducing the ADO.NET Entity Framework

The ADO.NET team has a new technology brewing that aims to raise the level of abstraction at which applications perform data access: Entity Framework. It provides a mechanism via which you can define a conceptual model on top of a database schema and then program against that model. The model is often referred to as the Entity Data Model (EDM).

The Entity Framework is a provider-based mechanism meaning that it has the potential to work with non-Microsoft databases[1] just like ADO.NET has done since its inception. The central idea of the Entity Framework is that the schema of your relational database may not be something that you want to surface directly to application programmers. One common case would be one where the schema is a little "less than optimal" and you'd like to tidy it up a bit for application access. Another case might be where the schema is highly normalised and hence not ideal for an application developer who is used to object oriented programming models.

In order to surface the EDM to an application developer, the Entity Framework needs three different groups of metadata:

1. The EDM itself. This is modelled using an XML grammar known as Conceptual Schema Definition Language (CSDL).
2. The definition of the underlying database schema. This is modelled using an XML grammar known as Store Schema Definition Language (SSDL).
3. A mechanism via which the framework is to translate (1) into (2) and vice versa. This is modelled using Mapping Schema Language (MSL).

In terms of programming the Entity Framework, we pass the metadata (CSDL+MSL+SSDL) to the object model and it surfaces the defined EDM in one of two ways:

1. Using the traditional ADO.NET data access API. There is a new EntityClient provider that has EntityConnections, EntityCommands and EntityDataReaders. The EntityClient executes queries written in a database neutral query language known as Entity SQL.
2. Using an Object Relational Mapping API that ships as part of the Entity Framework. This supports building queries using LINQ, so it is usually referred to as "LINQ to Entities".

As a simple example, we might have a DB2 store with a table called Customers which we map into conceptual entity sets: "Customers" and "PriorityCustomers". A developer can then write a LINQ query against those entity sets without needing to write any SQL or being exposed to the underlying store schema!

Mike Taulty

Read Mike's blog

> Visual Studio 2010 and .NET Framework 4.0 will include Entity Framework V2.0.

---

[1] There are now many released providers for the Entity Framework.

msdn
Microsoft®Developer Network

INTRODUCING THE .NET CLIENT PROFILE

03 September 2008: Introducing the .NET Client Profile

Visual Studio 2008 Service Pack 1 (SP1) has shipped along with .NET Framework 3.5 SP1. Together they deliver plenty of new features but there is one feature which has been relatively little talked about - a new variant of the Framework known as the .NET Client Profile.

The .NET Client Profile is a slimmed down version of .NET Framework 3.5 SP1 that includes libraries commonly needed for client applications such as Windows Presentation Foundation, Windows Forms and System.XML but omits Server-side technologies such as ASP.NET.

The intent is to make it easier to deploy the .NET Framework as part of your application by offering the ability to "brand" the installation whilst also minimising the installation overhead for the user who wants to run your application.

There are two variants of the installer for the .NET Client Profile:

1. The online bootstrapper. This is a just 270KB and needs a network connection to complete the installation which involves downloading an additional 28MB of files
2. The offline installer. This includes the complete .NET Client Profile and a full .NET Framework 3.5 SP1 and hence is around 250MB

Both installers operate in the same way:

1. Install the .NET Client Profile on a machine found to have no previous .NET Framework installed. At a later date Windows Update will make this a full install.
2. Install the full .NET Framework 3.5 SP1 on a machine found to have any previous .NET Framework installed.

The .NET Client Profile is therefore of most interest if your user base includes Windows XP machines which do not have the .NET Framework installed. The installation of the .NET Client Profile will take far less time (the install is actually smaller than the Adobe PDF Reader!) and will require no reboot.

Visual Studio 2008 Service Pack 1 has support for targeting the .NET Client Profile and will warn if your application attempts to use features beyond the targeted Framework. There is also support for publishing and deploying applications that use the Client Profile.

If you'd rather see this in action than read about it then there's a simple video here;

Video Showing .NET Client Profile Installation

and you can read more about the .NET Client Profile from this post I made and the links I reference.

Happy installations!

Mike Taulty
www.mtaulty.com

XNA GAME STUDIO

[9 July 2008: XNA Game Studio](#)

Platform games, shoot-em-ups, racing games, and adventure games. Peek this and poke that and the next generation of software developers were born. But why don't people write games for fun anymore? Perhaps games and their programming languages went in different directions, we all grew up and wrote database applications. Writing simple games became hard and the immediacy and fun just got drowned out in the API calls and lines of C++.

Move forward a couple of decades, and enter [XNA Game Studio](#). The founding principle behind XNA Game Studio was to enable people who like coding for fun to create and share great games. Games can be written once and with minimal changes be deployed to run on Windows or the Xbox 360. Version 2 of XNA enabled the creation of networked games over Xbox Live and with version 3 games can now target the Zune!

XNA comprises a collection of code frameworks that simplify the game creation process. Namespaces for handling player input, storage, sound, maths, networking, and of course, graphics are provided – the latter provides 2- and 3-D handling, including sprite management and collision detection. Key to the framework is two core infrastructure elements – the *Game Class*, and the *Content Pipeline*.

The *Game Class* provides a convenient set of event handlers that constitute a typical game loop – get input, update the game state, update the screen. You insert your game code in the handler routines, and XNA does the heavy lifting.

The *Content Pipeline* is key to how XNA can provide both independence between deployment devices and handle game resources from multiple sources. When a game initialises, the game resources (such as sounds, meshes, textures etc.) are loaded into the pipeline. During game play, you can access your resources from the pipeline as needed – and because XNA uses generics, you can ensure type-safe access as well.

As well as the Framework, XNA provides a set of extensions to Visual Studio to allow remote deployment and debugging of your game to an Xbox 360, or Zune. Currently v2.0 supports Visual Studio 2005, and 3.0 supports Visual Studio 2008.

Finally, bringing the whole community together is the [XNA Creator's Club](#) where you'll find everything you need – from discussion forums to complete game source. More recently we launched the Creator's Club Online where you can share your games with the community or peer-review and download games from other XNA fans.

Getting started with XNA couldn't be simpler – especially since everything you'll need to write great games for Windows is FREE, and available for [download](#). Everything you need to know about creating your first game can be seen in the [Quick Start guide](#).

C'mon - what are you waiting for? Me? I'm off to play snake...

Andy Sithers
Academic Evangelism Manager

[Read Andy's team blog](#)

LANGUAGES

COMBINING DYNAMIC LANGUAGES AND STATIC LANGUAGES

10 December 2008: Combining Dynamic Languages and Static Languages

Traditionally, there has been tension between dynamic languages and static languages. Each language has its distinct strengths and weaknesses, which means a choice: constrain yourself to one language or face the pain of language interoperability.

The Dynamic Language Runtime (DLR) solves this problem. The Common Language Runtime (CLR) provides a unified platform for statically typed languages. The DLR extends the CLR to provide a unified platform for dynamic languages on the .NET platform. This means that there is a common, unified platform for static and dynamic languages. By having one platform, which means one object world, the pain of language interoperability is history.

In addition to dynamic languages such as IronPython and IronRuby, C# and VB will also, in the 4.0 release of the .NET platform, be able to take advantage of the DLR. In C# this capability is achieved through a new type: dynamic. Declaring a dynamic type in C# 4.0 tells the compiler that method resolution is to be performed at runtime. There's a DynamicObject abstract class (which implements the IDynamicObject interface) that allows you to create your own dynamic objects in your chosen .NET language. In Visual Basic 10, the latebinder will be updated to recognise objects that implement the IDynamicObject interface, which will allow developers to take full advantage of dynamic languages from VB.

To talk to languages and platforms the DLR uses binders. As you would expect, there's a binder that allows the DLR to call .NET. Binders are also being built to allow calling Javascript (in Silverlight), Python, Ruby and COM. This model removes the n squared problem of language interoperability and replaces it with a standard, well-defined and well-understood platform.

To find out more about the DLR, the following sessions from PDC2008 are a great place to start.

TL10 Deep Dive: Dynamic Languages in .NET – Jim Hugunin

TL16 The Future of C# - Anders Hejlsberg

To see what you can do with the DLR today, download the latest version of IronPython, which comes with the DLR. Then write some ASP .NET with IronPython or call Python from C#.

To experiment with the dynamic capabilities of C# 4.0, download the CTP of Visual Studio 10 and .NET 4.0.

Mark Bloodworth
Architect Evangelist, Microsoft

GETTING STARTED WITH IRONPYTHON

[16 April 2008: Getting Started with IronPython](#)

The Dynamic Language Runtime (DLR) is a framework for creating dynamic languages that run on the Common Language Runtime (CLR). At the Lang.NET symposium the DLR team made quite a splash with [IronPython](#). IronPython version 1.0 was released in late 2006 and then the DLR was abstracted out. IronPython 2.0 was released in December of 2008[2].

There is increasing developer interest in dynamic languages, and Python is a particularly good one. It is widely used, semantically concise and easy to learn. As a dynamic language, types are enforced at runtime rather than compile time. Developers used to statically typed languages may find this removal of the "safety net" worrying, but in practise it turns out not to be a problem, particularly if you are using best practises for your testing cycles. There is a performance cost for these runtime checks, but dynamic typing makes it easy to do things that are either cumbersome or not even possible with statically typed languages. Benefits include first class (and higher order) functions and types, late binding, easy introspection, duck-typing and metaprogramming. For more on dynamic typing, read [Strong Typing versus Strong Testing](#).

The most important thing about IronPython is how well it is integrated with the .NET framework, and how easy it is to work with managed objects. This makes mixing C# and IronPython very straightforward. Beyond writing full applications, other practical uses for IronPython include embedding it into applications to provide a scripting environment, using it as a standalone language for system administration and utilising it as a tool for exploring assemblies. This last use case is best done with the interactive interpreter, which is also a great place to start experimenting.

After downloading IronPython, you start the interpreter by running "ipy.exe", which you can also use at the command line to execute Python scripts. You will be presented with a version string and a 'prompt', at which you can enter code and 'interactively' see the results. This means that you can inspect objects to see what methods are available and what happens when you call them. You start by adding references to assemblies (at runtime of course). Here's a simple example using *System.Management*:

```
IronPython 1.1 (1.1) on .NET 2.0.50727.1433
Copyright (c) Microsoft Corporation. All rights reserved.
>>> import clr
>>> clr.AddReference("System.Management")
>>> from System.Management import *
>>> mo = ManagementObject("Win32_PerfFormattedData_PerfOS_Processor.Name='_total'")
>>> for p in mo.Properties:
... print p.Name, '=', p.Value
...
```

For more recipes illustrating how to work with IronPython, visit the [IronPython Cookbook](#). However, the best resource for IronPython is my book "[IronPython in Action](#)".

Michael Foord, Senior Software Engineer at [Resolver Systems](#).

Read [Michael's Blog](#).

You can [hear Michael talk](#) about IronPython on .NET Rocks recorded in March 2009

---

[2] This article was written before the release of IronPython 2.0 and has been updated to reflect this.

msdn

TOOLS AND DEVELOPMENT PRACTICES

PEX: AUTOMATED TESTING FOR .NET

01 October 2008: Pex: Automated testing for .NET

Have you ever had your application produce an error due to an edge case scenario or a combination of inputs you were not expecting? Maybe the code went down a route that was not fully tested by either your manual or automated tests. This happens to all software at some point during its lifetime and it can be costly to fix. This raises the question, what can we do to minimize the chance of this happening?

A new project from Microsoft Research might have the answer. A prototype called Pex (Program EXploration) automatically generates test case inputs to cover all the different possible combinations, with the aim of 100% code coverage from the minimum possible set of automated tests. I am excited about the potential of this framework; I think it will improve the quality of code from a minimum cost and effort.

Generally, I am against programs that attempt to automatically generate test inputs and code, however Pex is very different. It monitors the different execution paths the application could take, how different inputs affect the execution path and determines how to execute those statements via different inputs.

By using the provided Visual Studio Addin, together with your test, you can start 'Pex Exploration'. Pex will launch and by starting with the simplest possible input for the test, it begins to learn what happens. Under the covers, it uses a mixture of static (the code) and dynamic (the code running) analysis together with a constraint solver (Z3), to learn and understand your code and the execution. As a result, Pex can make educated choices about the inputs required to effectively test the application. The code it generates is a simple method call to a test together with the inputs that found the defect.

The tests Pex requires are Parameterised Unit Tests, similar to MbUnit's RowTest. These unit tests look very similar to normal unit tests and should follow the same principals such as isolated, focused and readable, however instead of the inputs being hard coded they are a parameter to the test. Pex uses these tests to control the execution via different inputs, and to hook into the code executed by the test in order to monitor and learn.

As a result, Pex will generate a set of unit tests covering all possible edge cases and combinations based on your parameterised unit test. With this set, there a good chance of 100% statement coverage and more importantly 100% branch coverage to minimise the chance of an unexpected result and enable you to have more confidence in your code.

Pex is available to download from research.microsoft.com/Pex under a non-commercial academic license. For more information on how to use Pex, you can read my blog post or watch the Channel 9 video. While Pex is only a research project, could this provide a base for the future?

Ben Hall

Read Ben's Blog

A Pex session was delivered at the Professional Developers Conference in October of 2008

msdn

WEB

THE INTEGRATED PIPELINE IN IIS7

[23 January 2008: The Integrated Pipeline in IIS7](#)
Back in late 2005 I developed an HttpModule for ASP.NET that parsed the output of web pages and checked their validity as XHTML documents. The module appended a little report to each page with details of any validation errors. You can still [get the source from this blog post](#). Perhaps the biggest limitation of this tool was that it could only validate the output from pages processed by the ASP.NET pipeline, usually those ending in the *.aspx* extension. There was no way of using the module to validate static pages or even pages generated by other server frameworks such as ASP or PHP... until now!

The new Integrated Pipeline is one of the most exciting features of Internet Information Services (IIS) version 7, which ships with Windows Server 2008. It allows your HttpModules to participate in all requests, including those for static content, PHP pages and even images. This enhancement is compatible with the existing ASP.NET HttpModule system. To my delight, this means that the XHTML Validator Module worked without any modifications and can now be applied against any type of page. Better still, you can use the HttpModules that ship as part of the ASP.NET framework in this way too.

Since ASP.NET version 2.0, its membership features have provided an easy way to build secure websites with roles-based access and a plethora of ways to manage user accounts. In fact, you can now do this with virtually no coding required by using the new login controls, SQL Membership Provider and [Web Site Administration Tool](#)! But what if you wanted to secure access to resources that don't normally pass through the ASP.NET pipeline (such as an image or a ZIP file)?

With IIS7 using the existing Forms Authentication HttpModule – coupled with the membership features mentioned above – provides a consistent way to secure your web site's assets. Whilst it was possible to achieve this for certain static types in previous versions of IIS, it was much trickier to deploy and didn't work for most dynamic types. Now we have a hassle-free approach that works consistently across all file types.

It's easy to configure too, especially with the revamped IIS Manager that comes with version 7. First, your Application Pool must be setup to use the Integrated Pipeline: just set the Managed Pipeline dropdown to 'Integrated' in the Basic Settings dialog. Second, you need to modify the registration of the HttpModule itself by un-checking the "Invoke only for requests to ASP.NET applications or managed handlers" checkbox. You'll find this by navigating to your web site and clicking the Modules icon in the IIS group. This last change actually modifies the appropriate web.config file by moving the module from the "system.web/modules" section to the new "system.webServer/modules" section.

Visit MSDN online for more on [creating and registering a custom HTTP module](#). For more on IIS7, please visit the [IIS homepage](#). As I always say, "easy peasy".

Josh Twist
[Application Development Consultant (ADC)](#)

[Read Josh's Blog](#)

VISUAL STUDIO 2010

INTRODUCING PARALLEL EXTENSIONS TO THE .NET FRAMEWORK

6 August 2008: Introducing Parallel Extensions to the .NET Framework

Dual, quad, and eight-core processors are becoming the norm. *Is your application capable of utilising all available processors?* In order to achieve this level of utilisation on an *n*-processor machine, an application ideally needs at least *n* threads concurrently performing operations.

Although writing multi-threaded applications has become simpler over the years, many still find it challenging. Also many of the constructs such as the .NET ThreadPool have deficiencies that make them less suitable as the manycore shift begins to accelerate.

Over the past few years, Microsoft has been hard at work developing a new set of parallel extensions for .NET, aptly named Parallel Extensions to the .NET Framework. As one of its core components, it includes an implementation of LINQ-to-Objects that automatically executes queries in parallel, scaling to utilise most or all of the available processors without developers explicitly managing the distribution of work across all processors. This technology is called Parallel LINQ or PLINQ and the query syntax is almost identical to that of LINQ-to-Objects.

```
var q = from c in customers.AsParallel()
        join r in regions on c.RegionID equals r.RegionID
        where c.City == "London" && r.Name == "Kingston"
        select c;
```

When a repetitive computation can be parallelised, the Parallel Extensions to the .NET Framework offers a few interesting data-oriented operations such as **For** and **ForEach** that execute a loop in which iterations may run in parallel on parallel hardware. For example the following classic *foreach* loop can easily be converted to a parallel *foreach*:

Sequential execution:
```
foreach (var c in formulae) Calculate(f);
```
Possible parallel execution:
```
Parallel.ForEach(formulae, f => Calculate(f));
```

The actual parallelisation of actions is managed by the Task Parallel Library (TPL) that provides an abstraction layer on top of raw threads. TPL uses the notion of **Task** that is the smallest unit of work and could potentially be executed by any thread owned by TPL. Therefore correctly identifying tasks that could be executed in parallel is crucial. It is then the job of the scheduler component of TPL to decide whether those tasks should execute sequentially or in parallel. This decision is usually made based on available system resources and possible user preferences.

```
// Create a task to call Calculate(object o) and schedule it for execution
Task t = Task.Create(Calculate);
// Other activities...
// Wait for this task to complete
t.Wait();
```

One of the other additions is a set of lightweight and scalable thread-safe data structures and synchronization primitives such as a concurrent dictionary and a spin lock.

As you can see, there is a lot of promise in Parallel Extensions. If you want to find out more, I'd encourage you to download the June 2008 Community Technology Preview build[3].

Pedram Rezaei
Application Development Consultant (ADC), Read Pedram's blog

---

[3] The Parallel Extensions will ship as part of the .NET Framework 4.0 and therefore can also be explored in CTP or Beta releases of Visual Studio 2010.

msdn
Microsoft®Developer Network

CLOUD COMPUTING

AZURE SERVICES PLATFORM

26 November 2008: Azure Services Platform

The release of a major new operating system is a rare thing in IT. At the PDC in LA last month we witnessed one such rare event, the release of the first genuine operating system for cloud computing: Windows Azure. Why do I say it's an OS and why is it the first for cloud computing? Well if you consider what Azure does, scheduling services, provisioning resources, managing resource contention, handling service interruptions, monitoring and quota management, then this is very much what you would expect an OS to provide. Azure also provides some uniquely cloud based capabilities such as being able to scale-up **and** scale-down your compute & storage needs based on your needs, very high resilience and geo-scale availability across the globe. All of these capabilities are made possible through the magic of the fabric controller which runs on the tens of thousands of servers inside Microsoft data centres around the world.

A fundamental tenet of Azure development is that it is an open, standards based & interoperable platform that will support both Microsoft and non-Microsoft languages and environments. Azure supports multiple internet protocols including HTTP, REST, SOAP and XML. As a .NET developer you can write a .NET application and host it on Windows Azure with essentially a click of the mouse.

As impressive as Windows Azure is, it is just one element of the Microsoft Azure Services Platform. The Azure Services Platform currently consists of four groupings of cloud technologies, each providing a specific set of services to application developers. Alongside Windows Azure, we have .NET Services, Live Services and SQL Services.

The .NET services components (access, service bus and workflow) help you to create secure, federated applications that span organisational boundaries. Live Services allows synchronising of data across desktops and devices, finding and downloading of applications and enables you to add social networking capability to your applications. SQL Services adds hierarchical storage for on-premise and cloud based applications. In the future, the Azure Services Platform will be extended to include services built around SharePoint and MS Dynamics CRM.

Compared with other cloud environments, Azure provides comprehensive compute and storage services but also adds Azure Platform building block services such as .NET, Live & SQL Services which considerably enriches the developer toolkit. Another key difference is that with Azure you can use your normal Visual Studio tools and environment to create cloud based applications in a variety of languages.

If you haven't had a chance to explore Azure yet, then I'd urge you to take a look at the SDK and the Azure Platform Services. I think you'll be impressed.

Gurprit Singh
Director, Emerging Technologies, Microsoft UK

WINDOWS AZURE

07 January 2009: Windows Azure

A few weeks back my colleague (and boss) Gurprit Singh wrote about the Azure Services Platform and touched briefly on Windows Azure, the new Operating System for the Cloud from Microsoft. In this article I describe the services Windows Azure provides and how you can get started. Windows Azure provides four key capabilities.

Firstly it provides **computation**. In this first community technology preview (CTP) Windows Azure applications are written using .Net Framework 3.5.1 and two application models (or roles) are supported. The first of these is the Web Role. This is an application which can receive requests over HTTP or HTTPS written using ASP.Net for web applications or ASMX\WCF for Web Services. The second role is the Worker Role. Application code in a Worker Role cannot receive input directly from external sources, instead application code in a Worker Role will normally receive input from a Web Role via a queue.

Next it provides **storage services**. Rather than the familiar file and folder based data abstractions that we find in other versions of Windows, in Windows Azure we find three new ones addressing large scale storage requirements. These are blob storage, table storage and queues. Blob storage is used for storing opaque pieces of data and associated metadata supporting blobs up to 50GB. Table storage enables data to be organised into collections of keyed entities containing properties. Queues are used for persisting messages and are also used to pass requests between the Web and Worker roles. The storage service automatically scales data across multiple servers and provides redundant copies for availability. Besides being available to applications running inside Windows Azure data in the storage system can also be accessed externally via HTTP using RESTful Web Services. LINQ can also be used to query Windows Azure Table storage via the ADO.Net Data Services client.

Windows Azure applications are subject to Code Access Security just like any other .NET code. Rather than running with Full Trust, Azure applications are restricted by Windows Azure Code Access Policy[4]. Both the Web and Worker roles are stateless which means applications need to either save their state within the storage service or return it to the client. These constraints contribute to Windows Azures ability to deliver its third major feature, **Service Management**. For example, a developer declares how many instances of a particular role are needed and Windows Azure then automatically takes care of deploying the application, making sure that the required number of instances are running, monitoring and managing failures and applying updates as needed.

Finally development for Windows Azure can be done without having to be connected to the cloud. The Windows Azure SDK provides a local developer version of Windows Azure to enable testing and debugging of applications alongside the normal samples and documentation. There are also a set of tool extensions and templates for Visual Studio 2008 and Visual Web Developer 2008. Together these make developing for Windows Azure extremely **familiar to .Net Developers**.

Getting started with your first Windows Azure application is easy. First download the SDK and tools, then watch the PDC sessions and use the walkthroughs to kick start your efforts. Whilst you don't need to register for an account to build your applications you can also register for access to the Windows Azure service here.

Simon Davies
Architect Evangelist
Microsoft UK
Read Simon's blog

---

[4] In March 2009 at MIX in Vegas we announced full trust would be available for Windows Azure applications.

msdn
Microsoft®Developer Network

MEET THE AUTHORS

ANDREW SITHERS

Andy works with UK university faculties, schools and departments to build long-term relationships that will ensure the best and most appropriate use of Microsoft's developer platform, and that students can use these tools to build themselves a great technology career.

Andy can claim one of the more interesting routes into the software industry. He's held a number of academic positions, extolling the usefulness of software to university students and staff alike, as well as building educational software. Since leaving his ivory tower, Andy has worked variously as a Technical Writer, Consultant and Developer. His software efforts are scattered far and wide - from Bristol to Brisbane, taking in Belfast on the way.

Before joining Microsoft, Andy was a Team Leader & Software Architect for a data solutions company, and introduced .net as a way to produce great applications for UK and US retail supply chains.

Academic Evangelism Lead, Microsoft UK
Blog: http://blogs.msdn.com/sithers
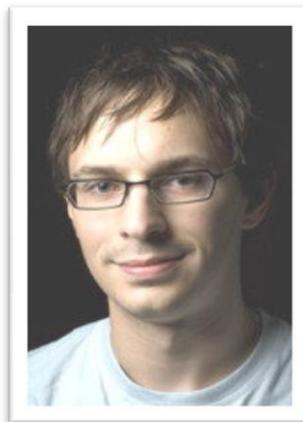
BEN HALL

Ben Hall is a UK C# developer\tester with a strong passion for software development and loves writing code. Ben works at Red Gate Software as a Test Engineer and enjoys exploring different ways of testing software, including both manual and automated testing, focusing on the best ways to test different types of applications. Ben is a C# MVP. Over the past few months I had been researching Pex and using the tool on some sample applications, which resulted in a series of blog posts and a user group presentation. However, to educate more people into why Pex is important I decided to write a MSDN Flash article to introduce the tool to a wider audience.

Test Engineer
Blog: http://blog.benhall.me.uk
Email: Ben@BenHall.me.uk

GURPRITPAL SINGH

Gurpritpal (Gurprit) was appointed Director of Emerging Technologies in June 2006. In this role he is responsible for developing and orchestrating the execution of the Microsoft UK subsidiary's strategy for emerging technologies. His main focus areas are the next generation internet, related developer and platform technologies, new business models and working with partners and customers on next generation internet solutions.

Director Emerging Technologies, Microsoft UK
Email: gsingh@microsoft.com

msdn

JOSH TWIST

Josh is an Application Development Consultant for Microsoft UK and works with a variety of customers, from small ISVs to large Enterprises. The Development Consultant role requires skills in all areas of the Microsoft Application Platform from architectural consulting and .NET development to debugging and performance tuning. Josh has recently been focusing on WPF and Silverlight and has acted as an advisor to Microsoft's Patterns and Practices group on both versions of their Composite Application Guidance for WPF and Silverlight.

Application Development Consultant, Microsoft UK
Blog: http://www.thejoyofcode.com/
Email: jtwist@microsoft.com

MARK BLOODWORTH

Mark Bloodworth is as an Architect Evangelist in the Development and Platform Evangelism Team at Microsoft, where he works with customers to harness technology to solve business problems.

Prior to joining Microsoft, Mark was the Chief Solutions Architect at BBC Worldwide, where he led a team responsible for architecture and systems analysis. This role encompassed technical strategy, architecture and team management. With a background in software development and design, specialising in web applications and integration most of his career has been focused on using Microsoft technologies, especially .NET, with a little Java thrown in for good measure. Since the advent of Ruby on Rail and Django, Mark has spent some time with dynamic languages and is most interested in how they can be combined effectively with the .NET framework.

Architect Evangelist, Microsoft UK
Blog: remark.wordpress.com

MICHAEL FOORD

Michael Foord has been developing with Python since 2002. He blogs and writes about Python and IronPython far more than is healthy for one individual and in 2008 was made the first Microsoft MVP for dynamic languages. Michael is the author of IronPython in Action for Manning publications and as the Resolver Systems community champion he speaks internationally on Python and IronPython. He can also be found online at http://www.voidspace.org.uk/python/weblog/. In the real world he lives in Northampton, UK with his wife Delia.

Senior Software Developer at Resolver Systems, London UK
Blog: http://www.voidspace.org.uk/python/weblog/
Email: michael@voidspace.org.uk

MEHRAN NIKOO

I work in the Application Development Consulting team at Microsoft UK. I spend most of my time working with Microsoft's enterprise customers and partners covering various areas and technologies including .NET Framework, WCF/WF, application architecture and development process. I also spend some of my time working with other teams at Microsoft, exploring and providing feedback on the existing and upcoming products from various product teams. I have been reviewing and contributing to the Application Architecture Guide 2.0, which was published by the patterns & practices team in December 2008.

Application Development Consultant (ADC), Microsoft UK
Blog: http://mehranikoo.net
Email: mnikoo@microsoft.com

MIKE TAULTY

Mike Taulty works in the Developer and Platform Group at Microsoft in the UK where he has spent the past few years helping developers understand and get the best from the Microsoft platform. Prior to this, Mike spent 3 years with Microsoft Consulting Services as a consultant on developer technologies.

Before joining Microsoft, Mike spent the previous 9 years working as a software developer for a number of enterprises, consultancies and software vendors working with a variety of operating system, client, communication and server technologies.

Mike holds a BSc Hons (1st Class) in Computer Science from the University of Leeds.

Developer Evangelist, Microsoft UK
Email: Mike.taulty@microsoft.com
Blog: http://mtaulty.com/

PEDRAM REZAEI

Pedram is a senior Software Design Engineer at Microsoft Corporation working on a variety of technologies. During the past decade, Pedram has led software development teams, architected complex solutions, and participated in developing many enterprise scale solutions. During his previous role at Microsoft UK, he worked as an Application Development Consultant helping ISVs and enterprises alike to design and develop software solutions based on Microsoft technologies and products. Pedram also often speaks at community events and authors technical articles and whitepapers.

Senior Software Design Engineer, Microsoft
Blog: http://blogs.msdn.com/pedram

SALVADOR ALVAREZ PATUEL

Salvador Alvarez Patuel is a senior application development consultant at Microsoft, dealing with a diverse of architectural and development challenges. He has been in the development arena for more than 13 years across different sectors, the most recent one was in the media industry where he was the main technical architect responsible for designing real time bidding engines. He has been also the co-author of the book "Silverlight 2 for ASP.NET developers", you may find him delivering sessions across different Microsoft events.

Senior Application Development Consultant, Microsoft UK
Blog: http://blogs.msdn.com/salvapatuel

SIMON DAVIES

Simon has been working with some of the early adopters of Windows Azure such as Active Web Solutions and Bluehoo for the past year or so helping them understand what the platform is and how to get the best out of it.

Application Architect, Microsoft UK
Blog: http://blogs.msdn.com/simondavies

ZUFIQAR AHMED

Hi, I'm an ADC (Application Development Consultant) working as part of Microsoft Premier Support. Being ADC, I get exposed to most of Microsoft development technologies however my expertise is in WCF,WF and technologies currently positioned under the "Oslo" umbrella. For past year or so I'm also looking into federation and SAML stuff in the context of Microsoft "Geneva" technology stack. "Geneva" Framework (previously known as "Zermatt") is a welcome addition to WCF to enable few key scenarios. My article on "Zermatt" was just an overview of the technology and I have covered it in much more detailed on my blog.

Senior Consultant, Microsoft UK
Blog: http://zamd.net