

## DotNetNuke – Utilizing ASP.NET and Open Source

Siegfried W. Fischbacher\*

In late 2001 and early 2002 many programmers eagerly awaited the arrival of the new Visual Studio 2002. VS 2002 was a major jump from Visual Studio 6.0 as it moved away from tradition Windows API manipulation toward the implementation of the newer .NET-based APIs. Microsoft warned developers the COM-based APIs were now being phased out. It was a disappointing time for many developers, all those years of learning the Windows APIs (User, GDI and Kernel) were now basically a waste of time and we had a long road ahead. The many tweaks and hacks we had worked out were now defunct but in return Microsoft was offering a radical shift from workstation computing to distributed computing, the benefits were being hailed as ground-breaking. Part of this strategy included a major revamp of their Visual Studio. The newer .NET version was to be more Internet-centric and was to look toward the development of remote server based solutions. Some other major changes included changes to languages such as VB.NET, which were now to follow object orientated methods more closely, more extensive use of XML, added languages such as C#, mobile device support and, of course, the introduction Web Services (SOAP). VS 2002 was probably the most exciting advancement since the jump from a 16 to 32-bit environment in Visual Studio 4.0., but developers had a lot to learn. One of the side benefits of this whole process however, was the development of a new open source portal system named DotNetNuke. In this article I would like to showcase this new portal and well as some of its benefits of its underlying framework (i.e. .NET)

The new .NET framework is actually a OS-neutral, device independent runtime system called the *Common Language Runtime* (CLR) which is essentially a mass of class libraries that can be summed up as system classes, data and XML classes, windows and drawing classes and web classes [1]. It compiles into an Intermediate Language (IL) much the same as JAVA. While there are many aspects of the .NET framework that still allow for traditional workstation development, of interest to many was the new incarnation of ASP, aptly named ASP.NET. In an effort to ease the pain of the transition to the .NET Framework, Microsoft helped to develop a new website, named ASP.NET [2]. At the new site developers were presented with a myriad of tips, hints and methods that allowed the developer to start utilizing the new .NET framework, especially in relation to web development. One of the greater areas of interest was the starter kits that were available. One of these kits that struck a lot of interest with the programming community was the Commerce Starter Kit (originally named IBuySpy). DotNetNuke is an extremely enhanced open source version of the I Buy Spy starter kit. DotNetNuke was originally initiated at Sourceforge [3] in December 2002 by Shaun Walker of Perpetual Motion Interactive Systems Inc.

[4], it has now been downloaded by over 100,000 users and many people have contributed to its development. DotNetNuke is officially described as;

*“an open source Web Portal Framework / Content Management System application written in ASP.NET/VB.NET for the Windows OS platform.”*

While it is essentially a portal, it's also much more. For those of us who have had to develop multi-user applications, a portal system, such as DotNetNuke, is a dream come true. Many projects required the initial development of the user interface and its related infrastructure, long before actual functionality can be instigated. Logging ability, role based-security, menu systems, these were all major issues that required enormous amounts of time and effort. DotNetNuke offers the interface and its related infrastructure already in place; by also using the Internet it also offers an incredibly sensible way to deliver the system to its users. With these issues gone the developer can really start to become creative straight away as systems such as DotNetNuke have incredible polymorphic properties.

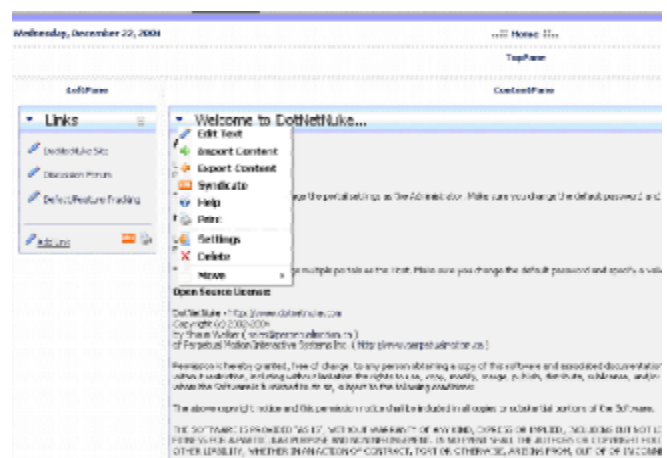


Fig. 1. Once logged on the user (if allocated editing rights) has the ability the add/edit particular modules, import/export content, get help and syndicate its contents.

DotNetNuke is easily installed on IIS 5.0 or 6.0. Originally it also required Microsoft SQL Server but it also ran on Microsoft's free SQL Server Desktop Edition. Originally many found hosting cost using SQL Server excessive so members started to code for Microsoft Access. DotNetNuke is now at version 3.0 and had been designed to run with many databases. Once installed it can be used straight away as a CMS. It offers many features in modulated form such as a user edible content page, a contacts manager, events calendar, simple document management, RSS news feed ability etc., Many developers have also developed a wide range of extra modules for use such as multi-user scheduling, CMS, e-learning systems etc. Apart from the users aspect, DotNetNuke also offers many benefits to host/developer such as a tight security system based on ASP.NET's Security Architecture, role based management, a payment system, skins, the ability to create

\* Department of Information Technology (International Program), KMITNB

multiple sites from the on a single ISP account, and many other features.

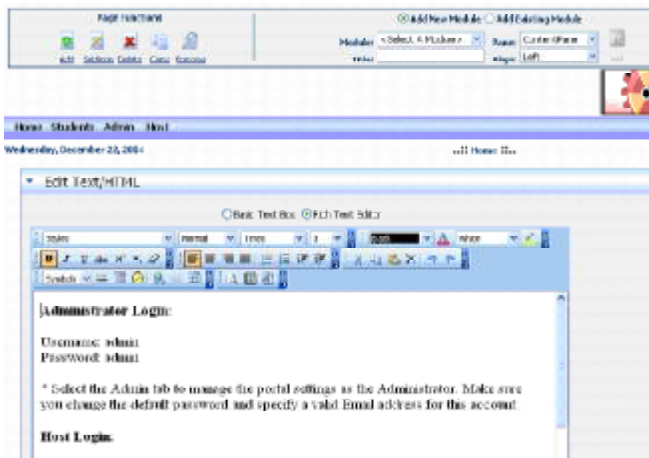


Fig. 2. This shows the sophisticated types of editing environment available to users. The .NET Framework offers the developer incredibly powerful programming functionality in this case a rich text editor.

Modules can be edited from any machine that has access to the portal. Most modules are extremely easy to use and to edit. Editing and viewing ability is dependent on the users allocated role. Modules can be placed in most areas of the portal with the interface being broken up into distinct panes. The menu system and pages are also editable (figure 1 and figure 2). Like other portals, users of DotNetNuke don't necessarily require any HTML experience etc.

Extra modules are quite easily created and added. Modules are usually written in VB.NET but other languages from Visual Studio can be used. To create modules you must follow and comply with DotNetNukes (Fig. 3) architecture. This architecture is well documented and readily available. Programmers are required to both inherit and implement many base classes for correct adaptation to the system. Recently new business logic and data access layers were implemented in accordance with Microsoft's new .NET Application Blocks. Many enhancements do make development a little more difficult but these are outweighed by the benefits, as well there is a system of templates available to help developers make the task much simpler. What can be achieved by the developer is quite exceptional, with the developer having access to the very sophisticated object hierarchy offered by the .NET Framework.

DotNetNuke is an excellent example of what open source projects can achieve, and in this case, using the .NET Framework. From a programmers perspective it offers the ability to concentrate on an applications functionality rather than its infrastructure. Its architecture is continually being implemented by its large community which is eager to ensure that it stays at the forefront of Portal development. With the pending release of ASP.NET 2.0, work is already on the way for its architecture to comply with the new requirements.

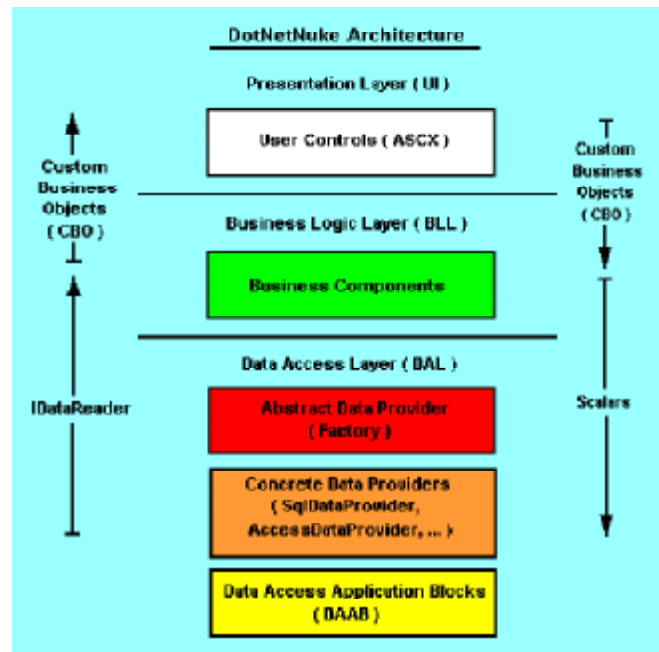


Fig. 3. The DotNetNuke Architecture, now incorporates recommendations from Microsoft's .NET Application Blocks

## REFERENCES

- [1] Michael Admundsen, Pual Litwin, ASP.NET for developers (USA: Sams Publishing, 2001) 17.
- [2] ASP.NET, Microsoft, 2004, <http://www.asp.net>.
- [3] Sourceforge, VA Software 2004, <http://www.sourceforge.net>.
- [4] Shaun Walker, .Net Nuke, Perpetual Motion Interactive Systems Inc, 2004, <http://www.dotnetnuke.com>.