



Migrating to ASP.NET

White Paper

Abstract

This paper describes the benefits of migrating to ASP.NET

Tometa creates custom software for you

Tometa Software designs and develops robust software solutions for virtually all industries including in-house (vertical market) and retail software, some of which is on the shelves at your local software store. We focus our unique combination of creative, technical, and problem-solving skills on meeting our client's objectives. Because of our clarity of purpose, commitment to process, and broad professional skill sets, we are able to provide our clients with world-class solutions that are functionally superior and fully aligned with our client's strategic focus.

Balancing development speed, quality and cost is what we are all about. Tometa combines agile development practices with fixed pricing, so you know what the cost, end product, and delivery time table look like—up front. If we underestimate the effort, we complete the overrun on our dime. Simple as that. That's why large enterprise firms like Alcoa and NASDAQ choose Tometa.

Tometa's agile development expertise and low-overhead US location keep our prices comparable to offshore vendors – without offshore challenges. Using a fixed pricing model, we provide upfront visibility into a project's ultimate costs, end product and delivery schedule. Our clients like knowing that we have “skin in the game” – a fixed price that aligns our goals with yours, incenting us to get the job done right and fast.

Lastly, as a Microsoft Certified Gold Partner, Tometa Software, can customize its products or create custom web, client/server, and traditional applications. With programming experience in C#, C++, Visual Basic, CGI, HTML, RPG, Delphi, Java and many others; Tometa Software is uniquely positioned to meet your needs as a development firm.

[Check us out today](#)

In the world of web development and web applications, technology is making advancements by leaps and bounds. From PHP (Preprocessing HTML Pages) to ASP (Active Server Pages), there are some very powerful tools available in creating web applications today. The newest tool for web development is Microsoft's ASP.NET, based upon Microsoft's .NET framework. This article will discuss the benefits in migrating a web application to the new ASP.NET architecture.

ASP.NET offers significant improvements over ASP in such areas as developer productivity, improved performance and scalability, reliability, security, deployment and XML (Extensible Markup Language) web service infrastructure. And while most programmers might choose to rewrite existing applications using the new ASP.NET format, a web application can contain both ASP and ASP.NET code. So it is not necessary to completely rewrite an entire web application all at once.

Developer Productivity

ASP.NET makes it dramatically easier to build robust, real world web applications in much less time than an ASP web application. Server controls provided with ASP.NET enable the programmer to use HTML-like style within the code. This allows the programmer to build outstanding user-friendly pages using far less code content as in ASP. Within an ASP environment, a programmer would have to work with script after script after script to facilitate allowing users to upload a file through an ASP page. With ASP.NET, the task of putting together an upload page consists of placing one control on the page! It's that simple...(well, a programmer would need to change a few settings as needed, but all in all – very quick and easy). Just as simple to accomplish is displaying data, getting user input, validating that input and then processing the information for further use.

Unlike ASP which only supports interpreted VBScript and Jscript (Visual Basic Script and JavaScript), ASP.NET supports more than 25 .NET languages like VB.NET, C#, and JScript.NET.

Using Visual Studio .NET, from Microsoft, a developer can add the productivity of Visual Basic style development with a “drag and drop” method of visually creating a web page. Visual Studio .NET also provides built in debugging, testing and deployment methods at the click of the mouse. While a developer can harness all of these new tools in developing a web application through ASP.NET, a programmer can also create robust ASP.NET web pages and applications through the use of the simplest text editors like Notepad.

Using Visual Studio .NET and the .NET Framework, a developer has access to over 4500 classes span across a rich spectrum of controls and aspects like performance monitoring and logging, file upload, image generation, XML processing, data access, transactions, regular expressions, message queuing, SMTP mail and so much more. Any of these features implemented in ASP would take line after line of code and even 3rd party components to complete the job and make the web application work as it was intended.

And probably one of the most advantageous developer aspects would be the fact that all ASP.NET pages are completely compatible with all browsers! Including Internet Explorer, Opera, AOL and even Netscape.

Improved Performance and Scalability

Caching is the process of temporarily storing frequently accessed data to enhance the performance and scalability of a web application. For example, assume that several news stories, stored in a database on a server, needed to be displayed on the homepage of a web site. Without the ability to cache the data, the database would have to be queried each and every time a visitor hits the page. Forcing the web application to make calls to the server and the database to retrieve that information with each new visitor to the web site. This causes the database to work harder than it really needs to given the fact that all visitors review the same data.

While ASP does contain a basic type of caching exposed through a special kind of state referred to as Application State. When data is added into ASP application state, it is available to all users visiting the web application or web site. Although application state is very useful, in a variety of circumstances, it provides no type of automatic expiration for the content being stored.

With ASP.NET, although it still supports application state, it enrolls a completely new caching mechanism with automatic content expiration. Now, content expiration is available to developers. This allows data caches to automatically expire and be refreshed without forcing the developer to write any code. Data caches can even be linked to files so that a given cache is invalidated and then refreshed when the linked file changes in some manner.

Besides being able to cache data such as news headlines, ASP.NET cache objects can also be used to cache entire web pages or even portions of web pages, which can greatly enhance a web site's scalability and allow database servers to focus on more intensive tasks, thus improving on the performance of a web application as well. Adding these ASP.NET cache objects is as simple as including a line of code at the top of a page, or, to cache a portion of a page simply add the cache object to the control or area that is to be cached.

In terms of compiled execution, ASP.NET is much faster than its predecessor ASP. ASP.NET will automatically detect any changes, dynamically compile the files if needed, and store the compiled results to reuse for subsequent requests. Dynamic compilation ensures that your application is always up to date, and compiled execution makes it faster.

ASP.NET provides an excellent way to make the web “stateful” through the use of ASP sessions. Sessions allow customer data such as shopping carts to be remembered between calls to the web server with minimal effort. However, ASP sessions can cause scalability bottlenecks on high-volume sites because the session data is stored “in-process” on the web server. Plus, ASP sessions are only good on the server where they are created, which poses a problem when web farms (groups of servers acting as one to serve a large web application) need to be utilized to maximize a web sties’ availability.

Although “in-process” sessions are still available in ASP.NET, two other options now exist including state servers and SQL Server. Rather than storing session directly on the web server, ASP.NET sessions can now be stored on a dedicated state server that can be shared by multiple web servers, or a web farm. Sessions can also be stored in a SQL Server database. This provides better scalability and availability of sessions.

Reliability

One of the biggest problems with web applications is something called a memory leak. When a process of an application does not clear out of memory and remains there, that memory block is not usable until it get freed up by way of canceling the process that is using that memory. ASP.NET can automatically detect and recover from errors like memory leaks to ensure a web application is always available to the users.

Let’s say that a web application has a small memory leak, and that after a week the leak has tied up a significant percentage of your server’s virtual memory. ASP.NET can detect this loss of virtual memory and automatically start another copy of the process that is in trouble. ASP.NET will then direct all new requests to this new process allowing the old process to finish all of its pending requests and finally, ASP.NET will allow the troubled process to be gracefully disposed of and the leaked memory is subsequently released for further use. All of this can be done by ASP.NET automatically, with out intervention from an administrator and without losing service to the application.

Security

Security is always an issue in today’s world that demands the utmost attention and focus. ASP.NET works in conjunction with IIS (Internet Information Services) to provide a complete security model for a web application. IIS security settings

are some of the few application settings that will actually carry over to and be applied to your ASP.NET application in a similar manner to that of ASP. There are of course, many additional enhancements to the security model brought in by ASP.NET.

For example, authentication in ASP.NET is supported in the following ways:

Type	Description
Windows	ASP.NET uses Windows authentication.
Forms	Cookie-based, custom login forms.
Passport	External Microsoft provided Passport Service.
None	No authentication is performed.

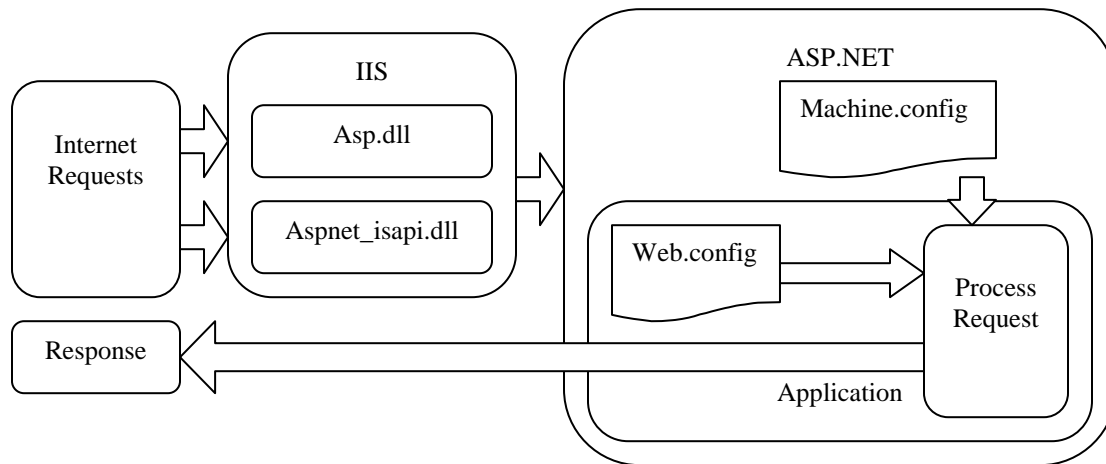
All of these authentications are the same options that are supported by ASP with the exception of the new Passport authentication option.

Deployment

ASP.NET has dramatically simplified the process of deploying a web application to the point of making it as simple as deploying an HTML page. With ASP.NET, a web application or web site is simply copied to the server, there is no need to run regsvr32 to register any of the components and configuration settings are stored in an XML (extensible markup language) file within the application.

ASP.NET allows for the dynamic updating of a running application as well, without the need of restarting a web server after the newly compiled application has been copied to the server. With ASP, the developer would have to restart the web server each time an update and deployment was made. ASP.NET allows the developer to simply copy the updated component over to the server, replacing the existing component, and ASP.NET will detect the change and start using the new component.

With ASP.NET, there is no need to migrate your existing applications as ASP.NET will run side-by-side with ASP on a Windows 2000 or Windows XP platform using IIS servers. ASP applications continue to be processed by the ASP engine while new ASP.NET applications are processed with the ASP.NET engine.



(the above diagram shows how an ASP application can reside along side an ASP.NET application in an IIS environment)

XML Web Services

XML or extensible markup language is a relatively new concept that allows the broad generalization of data to be shared by many different web applications. XML web services allow applications to communicate and share data over the internet, regardless of operating system or programming language. ASP.NET makes exposing and calling XML web services simple.

Conclusion

As you can see, ASP.NET offers a great deal of enhancements from greater developer productivity to improved performance. ASP.NET's scalability and reliability far surpass previous forms of web development. And with everything based upon Microsoft's .NET framework, ASP.NET ensures seamless compatibility with a variety of configurations.

Although there is certainly a learning curve associated with migrating from ASP to ASP.NET, the effort is well worth the time when considering the numerous productivity features offered by the .NET framework and ASP.NET.

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2004 Tometa Software, Inc. All rights reserved.