

Executive Summary

Microsoft's .NET technology offers persuasive richness for complex web applications. Moving to .NET technology opens whole new world of utilities which were difficult to prefigure before, this document will bring focus on essential aspects of Migration from ASP to ASP.NET.

Microsoft's .NET Framework is rapidly gaining momentum as a standard for business systems that place a premium on interoperability, both within the enterprise and with outside clients. While In this paper, we discuss the benefits that companies are seeking to gain in migrating to ASP.NET, and the various approaches they can adopt to ride on the ASP.Net freeway. This paper will serve as an aid to reader in understanding some key consideration of proposed Migration

The main Intent of this white paper is to remove the hindrances of reader towards migrating onto ASP.NET platform from classic ASP. This paper provides all the valuable information that is utilitarian in setting a clear mind frame of user that migration is bringing, what new to him? It also focuses on the understanding aspects of ASP.net to clear reader's expectation from proposed migration. . It helps reader to understand the procedural steps undertaken in the migration. Reader will be able to evaluate the complexity of a proposed migration. This paper brings the strategical aspect into the view of reader in order to help him deciding the correct migration path and finally he would be able to answer one final question-"why and how I should go for an ASP to ASP.NET migration".

Introduction

ASP.NET is part of Microsoft's .NET technology family, addressing the development of dynamic Web applications. It is a significant enhancement on the previous generation technology, ASP. ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web Services. ASP.NET pages execute on the server and generate markup such as Hypertext Markup Language (HTML), Wireless Markup Language (WML), or Extensible Markup Language (XML) that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface.

Although both Active Server Pages (ASP) and ASP.NET have the same goal of delivering efficient, flexible architectures for Web applications, the essential difference between these technologies is that ASP.NET provides a truly object-oriented, event-driven developer experience. Therefore, by using ASP.NET instead of ASP, you write fewer lines of code to create the same result. It also means that with ASP.NET, it is much easier for you to write more organized, cleanly structured code. Clearer, simpler code is easier to debug, test, deploy, and maintain; companies and organizations that have migrated Web sites from ASP to ASP.NET

with Hanu Software as their trusted change partner report dramatic improvements in code efficiency and code volume.

This white paper examines migration to the new technology both for new applications and for existing ASP applications. It considers the benefits as well as the technical, business and management difficulties. This document answers the following important questions:

This paper is divided into sections that address:

- Pros and Cons of ASP.NET
- Difficulties that will be encountered when moving to ASP.NET
- Strategies and Paths for moving an existing ASP application to ASP.NET

And finally this paper will give an insight that what are the critical factors which ensures successful Migration from ASP to ASP.NET

Contents

1	Novelty with ASP.NET	4
1.1	Better language support	4
1.2	Programmable controls	4
1.3	Event-driven programming.....	4
1.4	XML-based components	4
1.5	User authentication, with accounts and roles	4
1.6	Higher scalability.....	5
1.7	Increased performance - Compiled code	5
1.8	Easier configuration and deployment.....	5
2	Comparison in ASP and ASP.Net sample files.....	5
3	Difficulties in Moving to ASP.NET	7
3.1	Developers may not be conversant with .NET.....	7
3.2	Migrating existing applications to ASP.NET	7
4	Migration strategies-ASP to ASP.NET	9
4.1	Migration Overview.....	9
4.2	Key Considerations.....	10
4.3	Porting the ASP Application to ASP.NET.....	11
4.4	Strategies for Migration.....	11
4.5	Deciding on a Migration Path	12
4.6	ASP.NET Development tools	14
5	Conclusion.....	14

1 Novelty with ASP.NET

This section examines the primary benefits of moving to ASP.NET, focusing on the benefits emphasized by Microsoft. It is an critical evaluation of these advantages by Hanu Software. Following eight entities shows precise gain of ASP.NET over classic ASP.

1.1 Better language support

- ASP .NET uses the new ADO .NET.
- ASP .NET supports full Visual Basic, not VBScript.
- ASP .NET supports C# (C sharp) and C++.
- ASP .NET supports JScript as before.

1.2 Programmable controls

- ASP .NET contains a large set of HTML controls. Almost all HTML elements on a page can be defined as ASP .NET control objects that can be controlled by scripts.
- ASP .NET also contains a new set of object oriented input controls, like programmable list boxes and validation controls.
- A new data grid control supports sorting, data paging, and everything you expect from a dataset control.

1.3 Event-driven programming

- All ASP .NET objects on a Web page can expose events that can be processed by ASP .NET code.
- Load, Click and Change events handled by code makes coding much simpler and much better organized.

1.4 XML-based components

- ASP .NET components are heavily based on XML. Like the new AD Rotator, that uses XML to store advertisement information and configuration.

1.5 User authentication, with accounts and roles

- ASP .NET supports forms-based user authentication, including cookie management and automatic redirecting of unauthorized logins.
- (You can still do your custom login page and custom user checking)

- ASP .NET allows for user accounts and roles, to give each user (with a given role) access to different server code and executables.

1.6 Higher scalability

- Much has been done with ASP .NET to provide greater scalability. Server to server communication has been greatly enhanced, making it possible to scale an application over several servers. One example of this is the ability to run XML parsers, XSL transformations and even resource hungry session objects on other servers.

1.7 Increased performance - Compiled code

- The first request for an ASP .NET page on the server will compile the ASP .NET code and keep a cached copy in memory. The result of this is greatly increased performance

1.8 Easier configuration and deployment

- Configuration of ASP .NET is done with plain text files.
- Configuration files can be uploaded or changed while the application is running. No need to restart the server. No more metabase or registry puzzle.
- No more server restart to deploy or replace compiled code. ASP .NET simply redirects all new requests to the new code.

2 Comparison in ASP and ASP.Net sample files

Sample ASP file

server-side directive	→	<code><%@ language=javascript %></code>
server-side function	→	<code><script language='JScript' runat=server> function Add(x, y) { return x+y; } </script></code>
server-side evaluation syntax	→	<code><html> <body> <h1>Test ASP Page</h1> <h2>2+2=<%=Add(2,2)%></h2> <table border=2> <% for (var i=0; i<10; i++) { %> <tr><td>Row<%=i%> Col0</td><td>Row<%=i%> Col1</td></tr> <% } %> </table></code>
interspersed server-side script	→	<code><% Response.Write("<h2>Written directly to Response</h2>"); %> </body> </html></code>

Sample ASP.NET file

```
server-side directive → <%@ Page Language='C#' %>
server-side function → <script runat=server>
int Add(int x, int y)
{
    return x+y;
}
</script>

<html> <body>
<h1>Test ASP.NET Page</h1>

server-side evaluation → <h2>2+2=<%=Add(2,2)%></h2>
server-side syntax      <table border=2>
                        <% for (int i=0; i<10; i++) { %>
                            <tr><td>Row<%=i%> Col0</td><td>Row<%=i%> Col1</td></tr>
                        <% } %>
                        </table>

interspersed server-side script → <% Response.Write("<h2>Written directly to Response</h2>"); %>

</body> </html>
```

Fundamental changes

- ASP.NET is more than just the next version of ASP
- Pages are compiled into assemblies improving performance and diagnostics
- Code-behind encourages better separation of code from HTML
- Extensible, server-side control architecture
- Server-side data binding model
- Form validation architecture
- Web services allow assemblies to expose themselves as

3 Difficulties in Moving to ASP.NET

3.1 Developers may not be conversant with .NET

➤ *.NET is test for developers*

One of ASP's great assets is its very simple scripting language. .NET uses modern, full-featured languages. ASP.NET therefore places much greater demands on web developers, comparable to those placed on conventional software developers. Developers who have not come from a traditional software development background may require more extensive training to adapt to the new concepts. Even experienced software developers may not have fully used object-orientated features previously.

There could be performance scaling down and productivity decline while developers gets accustomed to .NET

➤ *Quality of code matters!*

The greatest difficulty in all programming is good design.

In ASP, implementing the principles of good design is extremely difficult, so this goal is not normally a realistic one. As a result, ASP applications tend to be complex and difficult to maintain, with very low code re-use.

With .NET there is the opportunity to produce better-designed applications, with resulting benefits in maintainability and quality. However these design skills are new demands on ASP programmers, and have a long and steep learning curve.

While it is possible to write ASP.NET applications in the ASP style, this will negate many of the advantages of .NET technology. Managing developers through their learning curve is the key to a successful move to .NET technology.

ASP.NET implementation requires more formal development processes than used for ASP development.

3.2 Migrating existing applications to ASP.NET

➤ *Compatibility issues*

Migrating your application to ASP.NET may not be easy; however, it should not be that difficult either. ASP.NET is very much compatible with ASP. This is impressive given the fact that ASP.NET is a complete overhaul of ASP. The designers of ASP.NET had an initial goal of being 100 percent backwards

compatible with ASP but subsequently had to back off this goal in favor of improving the platform for the long haul. Not to worry—the changes made were for the better and should not require a lot of work on your part to implement. The actual changes made can be categorized into the following sections:

- Core API changes
- Structural changes
- Visual Basic language changes
- COM-related changes
- Application configuration changes
- State management issues
- Security-related changes
- Data access

➤ *Coexistence (Is it possible?)*

It is very important to understand how ASP and ASP.NET can coexist. Both ASP and ASP.NET applications can run side by side on a server without adversely affecting each other. This is primarily due to the fact that separate file extensions (.asp versus .aspx) and separate configuration models (metabase/registry versus XML-based configuration files) are used between the two technologies. The two systems have totally separate processing engines.

It is entirely possible to have part of one application running ASP and another part of the same application running ASP.NET. This is very beneficial if you need to move a large, rapidly changing site to ASP.NET one piece at a time. Some would argue that you may be better off porting and deploying the entire site all at once. This may be the case for certain classes of Web applications, but Hanu thinks that there are a lot of sites out there where this may not be feasible due to the sheer size, complexity, and rapid evolution of the site's content and presentation. After all, if you are sitting on a profitable Web site, chances are the people paying the bills will not allow you to stop implementing their new features so that you can move things over to this hot new technology. Additionally, if you are going to put forth the effort to move to ASP.NET as a long-term investment, you will want to use this chance to make as many architectural and design improvements as you can. For these types of situations, coexistence using a phased-in approach is an absolute must

4 Migration strategies-ASP to ASP.NET

Migrating an application from conventional ASP to ASP.NET is a work of Art, lot of professional may not be agree with this but Hanu Software accomplish migration by considering each and every pin aspect as an Artesian would do.

Hanu's professionals has High quality of conception or execution, as found in works of beauty; aesthetic value.

At Hanu we refer ASP to ASP.NET migration as the process of making application work exceptionally well. Which, from concept to creation, hold a fidelity to the creative impulse.

To put to on comfortable zone, we are presenting some more aspect, which would be helpful to understand migration issues more easily.

4.1 Migration Overview

Installing ASP.NET will not break your existing ASP applications. It uses a separate file name extension (.aspx instead of .asp), separate configuration settings, and an entirely separate common language runtime (Asp.dll has not been modified). ASP pages and applications can continue to use the existing ASP engine, with no interference from ASP.NET. That said, the benefits of migrating your existing applications to ASP.NET are enormous. ASP.NET easily provides many times the features of traditional ASP, and moving your ASP applications to the new platform provides a huge opportunity for improvement. Among the new features you can take advantage of are:

- Improved performance and scalability

- Web farm support and XCopy deployment

- Output caching and custom security

- Web Forms page controls

- XML Web services infrastructure

ASP.NET is designed to help preserve your investment in traditional ASP and COM technologies. It balances support for existing ASP syntax and semantics with the need for a forward-looking platform that can last well into the next age of Internet application development. While ASP.NET preserves the majority of ASP's feature set, 100% compatibility between the two was not possible if the platform was to move forward, so there are a few changes to the old way of doing things.

The good news is that your ASP skills will translate easily to ASP.NET. There are only a few differences, which are usually easy to fix. However, migrating ASP applications to ASP.NET does require some work. Relatively simple pages might migrate without any changes, but more complex applications probably will require some modifications.

4.2 Key Considerations

- *Reinventing of wheel should be avoided*

Fact: Moving code to ASP.NET is a complex exercise.

Fact: The more code that must be moved to .NET, the more resources will be required, the more bugs will be introduced and the greater the chance that the project will fail.

Conclusion: The amount of code to be re-written should be minimized.

- *.NET is to eradicate problems, not for creating new ones.*

Fact: Users will not understand .NET and will not be tolerant of problems.

Conclusion: Moving functioning code to .NET should be avoided, as new bugs will invariably be introduced. .NET migration should focus on new functionality and areas of the system where problems already exist. Users should be made to feel that .NET is an improvement, not a (temporary) step backwards.

- *Be careful, if are not an Expert.*

Fact: There are currently few developers expert in .NET technology.

Fact: .NET services are very extensive: developers will not be familiar with the full range of functionality.

Conclusion: A good strategy recognizes these facts and does not require developers to become instant experts. Small steps are preferable to big ones. Small steps give opportunities for intermediate learning for developers, and make project management feasible.

Failure to recognize that developers will not be experts in all of .NET's functionality may result in an application that runs under .NET but does not take advantage of its capabilities.

As developers become more capable in ASP.NET they may attempt to revisit previously 'completed' pages to make them more 'fluent'; previously 'completed' milestones may require further developer time. Taking smaller steps enables faster learning and mitigates this risk.

4.3 Porting the ASP Application to ASP.NET

When porting an ASP application to ASP.NET, you will need to decide how much time you want to spend incorporating the new features of ASP.NET into the existing ASP application. Changing an ASP page's file extension from .asp to .aspx and correcting for some syntactical changes is usually enough to get an ASP page working as an ASP.NET page. Such a port, although it can be completed quickly, does not take advantage of many of the new features of .NET, including ASP.NET Web controls, Microsoft ADO.NET, the Microsoft .NET Framework classes, etc. Although a more complete port may take more time, your finished ASP.NET pages will be readable and maintainable, and more feature-rich.

As a developer, you need to weigh the tradeoffs when deciding what approach to take when moving an ASP application to ASP.NET. If you are pressed for time, a simple port from ASP to ASP.NET may be in order; if no such deadline looms, it may be worthwhile to take the time to build a rich ASP.NET application from the ground up. You can also, of course, take an incremental approach. If you ran a large site that utilized a number of COM components to implement business rules, you may decide to port over just the UI portion of your Web application, and continue to use the classic COM components.

4.4 Strategies for Migration

Migration is not necessarily an “all or nothing” proposition. Hanu Software classifies migrations into several distinct categories, based on factors such as the desired timeframe to achieve results and the strategic value that the organization places on the move to .NET.

As the .NET Migration Matrix in Figure illustrates, depending on the levels of importance assigned to these factors, there are five distinct approaches an organization may adopt.

Organizations for which the Perceived strategic value is Relatively low and/or rapid time to results is desired generally select one of two approaches

“CHECKING OUT” – Migrating an application (or a near standalone sub-system) for evaluation purposes. Evaluation metrics include not only the typical performance measures, but also generally include measuring the migration project experience (time to complete, unexpected issues, etc.). !

“NEW MASK” – Migration of the user interface tier of a system, which often includes not just a system migration to .NET, but also general user interface enhancements. Organizations that are seeking more long-term benefits and place a higher strategic value on the migration pursue different approaches.

“HAND SHAKE” – In this approach, essentially a subset of “checking out” an organization migrates a specific functionality, intending that it interoperate as part of the existing application. An example would be migrating a portion of reusable business logic and providing a Web services interface to it.

“ON THE ROAD” – The organization decides to invest in re-architecting a system in order to make best use of the .NET framework. Some development occurs, but the primary objective is to provide a comprehensive, detailed design that will guide subsequent development projects.

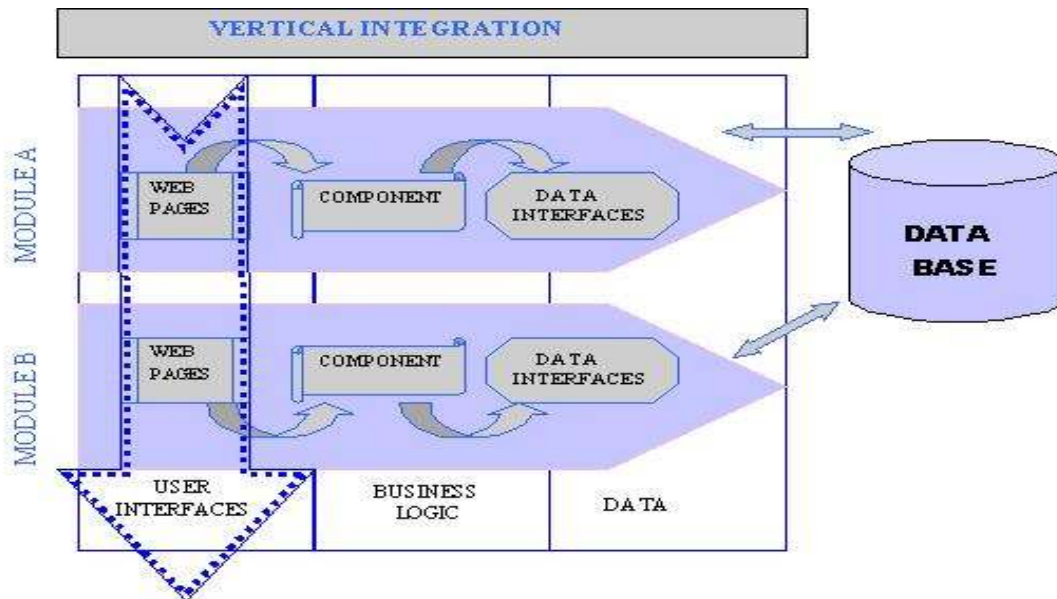
“FULL FLEDGE” – This approach encompasses full design, development, and deployment of the system in .NET, having determined the system must migrate due to strategic business needs.

4.5 Deciding on a Migration Path

As in all software development efforts, good planning is half the work. Based on clients migration goals, the category that best fits clients migration, the complexity and architecture of existing system, budget, and available resources, we can implement one of two distinct migration paths: vertical or horizontal migration.

Vertical Migration

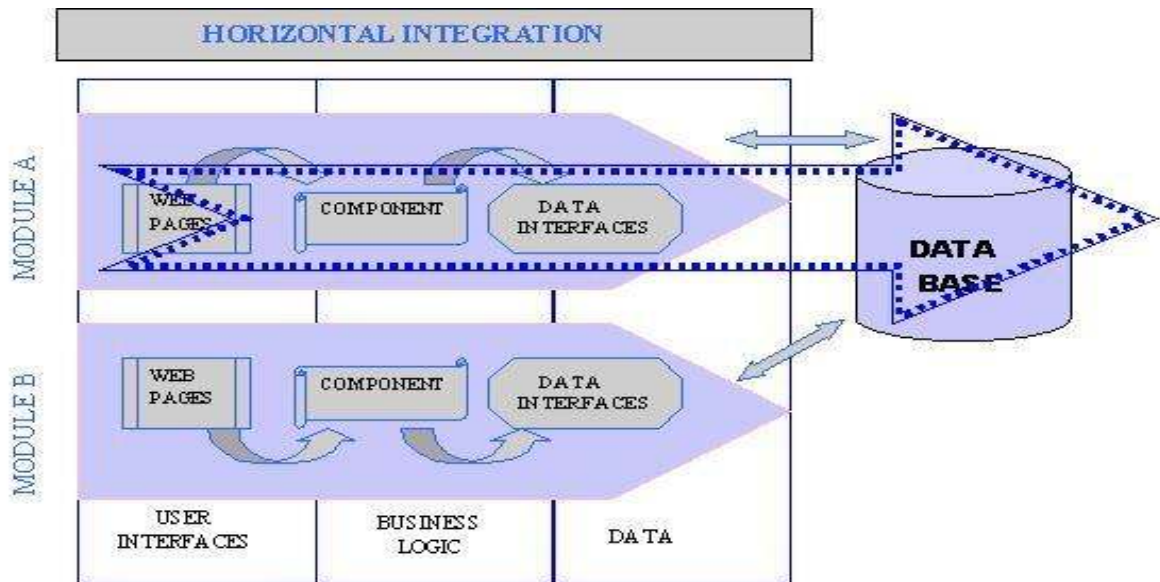
This approach calls for migrating an application one tier at a time. As shown in the Figure vertical migration approach would proceed through all the pages of the user interface tier for each module, followed by the business and data tiers. This strategy is most appropriate when the application’s modules are tightly coupled



and minimal changes are required to the interfaces between application tiers. The initial architectural review in this case could yield opportunities for shared code across all three tiers rather than result in separating the modules into distinct applications.

Horizontal Migration

In a Horizontal migration, the application code is migrated module by module, completing the migration for all three tiers of one module before moving on to the next. As shown in following Figure horizontal migration would proceed through Module A’s Web pages, business logic components, and database interface before proceeding to the migration of Module B.



A horizontal migration is most appropriate when application modules are loosely coupled and when interfacing between different application tiers is a tedious task. An architectural review all of the application’s modules should be conducted before any module migration begins. This review should uncover opportunities for creating reusable code, and may lead to separating the modules into independent applications..

4.6 ASP.NET Development tools

Several available software packages exist for developing ASP.NET applications:

- Visual Studio .NET
- ASP.NET Web Matrix
- Macromedia Dreamweaver MX 2004
- Macromedia Dreamweaver 8
- Macromedia HomeSite 5.5 (For ASP Tags)
- Visual Web Developer 2005 Express Edition (for ASP.NET 2.0)
- Delphi 2005

Though Hanu Software uses its own custom made tool for an ASP to ASP.NET migration but to enrich the architecture and design of a application above mentioned packages are very utilitarian.

5 Conclusion

Your .NET migration will involve series of decisions, some of which are particular to .NET, but many of which seasoned project managers and software developers will recognize them as themes that are consistent across successful development projects. Following are some measures, which can be set as checkpoints to your migration strategy.

Define what success means for your project.

What are goals of the migration?

What measures will you use to gauge success?

Determine the boundaries of the project

What functionality will be migrated and how will migrated functionality will interact with other systems

Develop a test plans

If possible create a migration team composed of developers who know the current system and developers with .NET experience

With this approach you can address business needs today and develop a team that will be even better equipped to support them in the future

About Hanu Software

Hanu Software is a global consulting and IT services company that provides end-to-end software solutions for various sectors such as Publishing, Finance, Insurance, Retail and others. Headquartered in Monmouth Jct., NJ, and with an offshore development center in Gurgaon (Delhi), India, Hanu Software offers a unique value proposition to its customers. We believe in providing highest quality software solutions to our customers at a very cost-effective rate.

Hanu Locations

Headquarter & Onshore Technology Center -

74 Heather Ct
Monmouth Jct, NJ 08852
PH: 732.668.3691
E-mail: asingh@hanusoftware.com

Marketing Office -

23 Filare Street
Irvine CA 92620
PH: 949.679.0778
E-mail: ltrombley@hanusoftware.com

Offshore Solution Delivery Center (OSDC) -

270, Sector 17 A
Gurgaon, India - 122001
PH: 011.91.124.5018494
E-mail: contact@hanusoftware.com

Acknowledgments

The following Hanu Software Employees contributed to the development of this white paper

Manoj Srivastava-Practice Manager, having a tenacious IT experience, helps clients to develop business application to leverage .NET platform

Amit Kataria-Marketing Consultant, focuses on .NET development projects, Management Scholar for renowned Indian College.