

ASP.NET Hosting Options

Introduction

Web applications are typically designed, created, and tested in a development environment that is accessible only to the programmers working on the site. Once the application is ready to be released, it is moved to a production environment where the site can be accessed by anyone on the Internet. This deployment process introduces a number of challenges:

- A production environment must exist and be properly setup before an ASP.NET application can be deployed; moreover, the production environment must be kept up to date with the latest security patches.
- The correct set of markup files, code files, and support files must be copied from the development environment to the production environment. For data-driven applications, this might require copying the database schema and/or data, as well.
- There may be configuration differences between the two environments. The database connection string or e-mail server used in the development environment will likely be different than the production environment. What's more, the behavior of the application may depend on the environment. For example, when an error occurs in development the error's details can be displayed on screen, but when an error occurs in production, a user-friendly error page should be displayed instead, and the error details e-mailed to the developers.

To obviate the first challenge - setting up and maintaining a production environment - many individuals and businesses outsource their production environments to *web hosting providers*. A web hosting provider is a company that manages the production environment on your behalf. There are countless web host providers, each with varying prices and service levels; see the "Finding a Web Host Provider" section for tips on locating such a service provider.

This is the first in a series of tutorials that look at the steps involved in deploying an ASP.NET web application to a production environment managed by a web host provider. Over the course of these tutorials we will examine:

- What files need to be deployed to the web host provider.
- Tools for streamlining the deployment process.
- How to deploy a database.
- Tips for deploying a database that uses [the SQL-based Membership and Roles provider](#), along with ways to mimic the Website Administration Tool in a production environment.
- Strategies for smoothly updating the database in production with changes made during development.

- Techniques for logging errors that occur on production, and ways to notify developers when an error occurs.

These tutorials are geared to be concise and to provide step-by-step instructions with plenty of screen shots to walk you through the process visually. This inaugural tutorial provides an overview of the ASP.NET deployment process and advice on finding a web hosting provider. Let's get started!

An Overview of the ASP.NET Deployment Process

In a nutshell, deploying an ASP.NET application involves the following three steps:

1. Configure the web application, web server, and database in the production environment.
2. Synchronize the ASP.NET pages, code files, the assemblies in the `Bin` folder, and HTML-related support files like CSS and JavaScript files.
3. Synchronize the database schema and/or data.

The configuration information for a web application is typically located in the `Web.config` file, and includes database connection strings, error handling criteria, URL rewriting rules, and e-mail server information. Oftentimes this information is different for an application in development versus the same application in production. For instance, when developing an application it's best to use a development database so that you're not testing against the production database. As a result, the database connection strings typically differ between development and production applications. Due to these differences, part of deployment involves making changes to the web application's configuration information.

In addition to web application configuration changes, Step 1 also may entail configuration for the web server and database. For example, if an ASP.NET page creates or deletes files from a directory on the web server then the web server needs to be configured to permit these file system modifications. Similarly, there may be permission or authentication settings that need to be made to the database.

Step 2 involves synchronizing the set of essential ASP.NET pages and support files between the development and production environments. The particular set of ASP.NET-related files that need to be synchronized between the two environments depends on the type of project you created in Visual Studio, and is the discussion in the next tutorial, *Determining What Files Need to Be Deployed*. The third and fourth tutorials - *Deploying Your Site Using FTP* and *Deploying Your Site Using Visual Studio* - examine different tools and techniques for syncing these files.

When building data-driven applications there are typically two databases being used: one for development and one on production. During development, the development database's

schema may be modified to include new tables, columns, stored procedures, and triggers, or may be modified to remove or rename existing database objects. Between the time that these changes are made and the time the application is deployed to production, the development and production databases are out of sync. This asynchrony needs to be fixed during the deployment process. These challenges will be examined in future tutorials.

Finding a Web Host Provider

ASP.NET applications can be deployed to any web server that has the .NET Framework and Internet Information Services (IIS) installed. You could host a site from your personal computer, assuming you had a broadband connection to the Internet and the know how to configure your router to allow incoming web requests. You could also host a site from a computer in an intranet, as many companies do. The focus of these tutorials, however, is hosting your website with a web host provider.

Note: [IIS](#) is Microsoft's enterprise-grade web server. It ships with the non-Home editions of Windows, such as Windows Server 2008 and certain editions of Windows Vista. You do not need to install IIS to serve ASP.NET applications in a development environment, as Visual Studio includes the ASP.NET Development Web Server. However, the ASP.NET Development Web Server only accepts local connections and therefore cannot be used in a production environment.

Before you can deploy your site to a web host provider you must first decide what company to do business with. There are countless web hosting companies in the marketplace; a search for "web hosting company" returns more than five million results. How do you find the one that's right for you? Your favorite search engine is a good starting place, as are websites like [TopHosts](#) and [HostCritique](#), which compare and contrast various hosting services. I also advise asking your colleagues and coworkers for any recommendations; you can also ask for recommendations at the [Hosting Open Forum](#) here at the [ASP.NET Forums](#).

Web hosting companies typically offer shared hosting plans and dedicated hosting plans. With shared hosting a single web server hosts dozens if not hundreds of different websites. With dedicated hosting you lease a computer from the company that serves your site and your site alone. A shared hosting plan might include support for ASP.NET pages, the ability to work with Microsoft Access databases, 5 GB of disk space, and 100 GB of monthly bandwidth traffic for \$9.95 per month. Another shared hosting plan might include support for ASP.NET pages, access to the Microsoft SQL Server 2008 database server, 10 GB of disk space and 250 GB of monthly bandwidth traffic for \$19.95 per month. Dedicated hosting plans are usually much more expensive, costing several hundred dollars per month, but offer better performance and more control than shared hosting options. What plan you choose depends on your budget, how much traffic your website receives, and the features you anticipate you'll need.

Two important considerations when choosing a web host provider are customer service and quality of service. If you have a question or a configuration problem, how long does it take from submitting your problem to the web host's helpdesk until you get a response? How

reliable are the company's services? Do they frequently have database outages? How often does their e-mail server go offline? You can always ask a company to provide details about their uptime and inquire about their customer service policy, but a more surefire way is to solicit the feedback of current and past customers, which you can do through online forums, newsgroups, and e-mail listservs.

Note: Some web hosting companies focus their business on a particular technology stack, such as .NET or [LAMP](#) (**L**inux, **A**pache, **M**ySQL, and **P**HP), so make sure that the company you select hosts ASP.NET applications. Also check to ensure that they support the version of ASP.NET you are using to build your application. And if you are building a data-driven application, make sure that the web host offers the same database server and version that you are using.

Summary

ASP.NET web applications are typically designed, created, and tested in a local development environment. Once a version is ready for release, it is moved to a production environment. While it is possible to host ASP.NET websites on your personal computer or on servers within your company, many businesses and individuals choose to outsource their hosting to a web host provider.

This tutorial series examines the steps for deploying an ASP.NET application to a web host provider, exploring common challenges. This tutorial offered a high-level overview of the ASP.NET deployment process and gave tips for finding a suitable web host provider. The next tutorial looks at what ASP.NET-related files need to be copied to the production environment when deploying your website.

Happy Programming!

About the Author

[Scott Mitchell](#), author of multiple ASP/ASP.NET books and founder of 4GuysFromRolla.com, has been working with Microsoft Web technologies since 1998. Scott works as an independent consultant, trainer, and writer. His latest book is [Sams Teach Yourself ASP.NET 3.5 in 24 Hours](#). Scott can be reached at mitchell@4guysfromrolla.com or via his blog at <http://ScottOnWriting.NET>.

Special Thanks To...

This tutorial series was reviewed by many helpful reviewers. Lead reviewer for this tutorial was Teresa Murphy. Interested in reviewing my upcoming MSDN articles? If so, drop me a line at mitchell@4GuysFromRolla.com.