



# Overview of .NET Framework Technologies for Building Web Applications and Services

*...for Java developers*

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

## Introduction

You can use the application programming interfaces (APIs) that are defined in the Java Platform, Enterprise Edition (JEE) specification to build dynamic, interactive Web applications and Web services by using Java. However, many Java developers also use frameworks such as Struts or Spring, which build on elements in the JEE specification to create large, maintainable Web applications.

The Microsoft® .NET Framework offers a corresponding model that's called ASP.NET. This model provides a comprehensive set of features for building Web sites by using either a page-centric development model, or a Model-View-Controller (MVC) pattern-based development model. The .NET Framework also includes Windows® Communication Foundation (WCF), which enables you to build both interoperable Web services based on common Internet standards, and service-oriented applications that use other protocols and message formats.

This paper is not intended to provide a detailed description of how to develop Web applications and services by using the .NET Framework. Instead, it is intended to help you familiarize yourself with the technologies that the .NET Framework provides for Web development. This paper describes a high-level mapping between the Web technologies that Java developers commonly use and the Web technologies that are available in the .NET Framework.

## Contents

Introduction .....	ii
Contents .....	ii
ASP.NET Web Forms .....	3
ASP.NET MVC .....	4
Web Services.....	5
SOAP-Based Web Services.....	6
RESTful Web Services.....	6
Silverlight and WCF RIA Services.....	7
Summary .....	8

## ASP.NET Web Forms

ASP.NET is the Web application framework that Microsoft has implemented as part of the .NET platform. It is designed to run in the context of a Web server such as Internet Information Services (IIS).

[ASP.NET Web Forms](#) provides the traditional way of developing Web applications in ASP.NET. An ASP.NET page plays the same role as a JavaServer Pages (JSP) page or a servlet in the JEE model. An ASP.NET page consists of two key elements: a file that contains HTML-like markup, much like a JSP page, and a code file that contains the user interface logic, much like a Java Bean that is referenced from a JSP page.

You can build up an ASP.NET page by adding server-side controls that the ASP.NET runtime in the Web server renders as HTML. The notation for adding a control to the page is similar to the notation that is used in JSP pages for using custom tags that are defined in a tag library. ASP.NET includes a comprehensive set of prebuilt server-side controls that perform a range of common functions. The following code example shows how to add the ASP.NET Login control to a Web form. This control contains built-in logic that renders HTML that prompts the user for a name and a password, and includes a **Log In** button. This ASP.NET control also integrates directly with ASP.NET authentication mechanisms, minimizing the amount of code that you have to write. There's no direct equivalent in the standard Java tag libraries that offers the same functionality with the same minimal coding requirements.

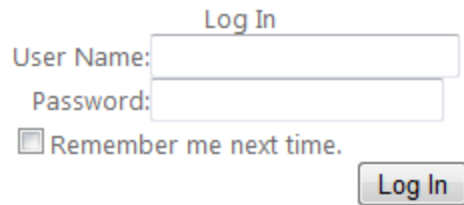
```
<asp:Login ID="MyLoginControl" runat="server">
</asp:Login>
```

Figure 1 shows how this control appears in the browser.



Did you know?

You can get started quickly with ASP.NET development by downloading [WebMatrix™](#). This software includes a lightweight development tool, IIS Express, Microsoft SQL Server® Compact, and a set of ASP.NET extensions that enable you to build stand-alone ASP.NET pages by using the new Razor syntax. WebMatrix also includes a set of easy-to-use database and HTML helpers for performing common Web tasks. You can install WebMatrix side by side with Microsoft Visual Studio® 2010.

The image shows a standard ASP.NET login control. It features a title "Log In" at the top center. Below the title are two text input fields: "User Name:" followed by a text box, and "Password:" followed by a text box. Underneath the password field is a checkbox labeled "Remember me next time.". At the bottom center of the control is a "Log In" button.

**Figure 1. The ASP.NET Log In control**

In addition to server-side controls that define standard user interface elements such as text boxes, lists, and buttons, ASP.NET includes more advanced collections of controls to display data, perform validation, handle security, and implement [AJAX](#) functionality. Just like Java tag libraries for JSP, you can also define your own custom ASP.NET controls.

ASP.NET includes features to help you define a common look and feel across your Web application. You can use [master pages](#) to define a regular page layout for your Web site, and [skins](#) to apply standard colors and fonts to your server-side ASP.NET controls. Visual Studio supports all of these features, giving you a great design in addition to a great development environment.

In a JEE Web application, you can use the application context object and session object to manage state information. ASP.NET includes equivalent application and session objects in addition to a page-level state object. ASP.NET also supports distributed session state in a Web-farm scenario "out of the box."

You can declaratively configure many features of an ASP.NET Web application in a web.config file that is stored with the Web application on the Web server. For example, a web.config file can contain settings that define security policies, integrate additional components, and configure logging. This configuration file is an XML document, just like a web.xml file in a Java Web project.

## ASP.NET MVC

[ASP.NET MVC](#) offers an alternative model to the traditional ASP.NET Web Forms page-based development model. ASP.NET MVC is an implementation of the well-known Model-View-Controller pattern, although the implementation differs from Java implementations of this pattern, such as Struts, Spring MVC, and JavaServer Faces.

ASP.NET MVC uses a routing table to map incoming HTTP requests to an appropriate controller instance. ASP.NET MVC then calls the controller's **Execute** method, which determines which action method it should call. The action method receives the user input from the request and generates a response (the View) to return to the user.

## Web Services

Many enterprise solutions incorporate Web services to expose functionality to client applications by using standard Web technologies and protocols. Originally Microsoft implemented XML Web services as part of [ASP.NET](#). However, the more extensive features that have now been standardized as elements of the various SOAP and REST Web services standards are available through [Windows Communication Foundation](#), a unified programming model for building service-oriented applications. WCF significantly extends ASP.NET Web services beyond SOAP over HTTP to enable you to employ a much wider selection of message formats and transport protocols. You can host a WCF service by using IIS, or you can build your own custom host application.



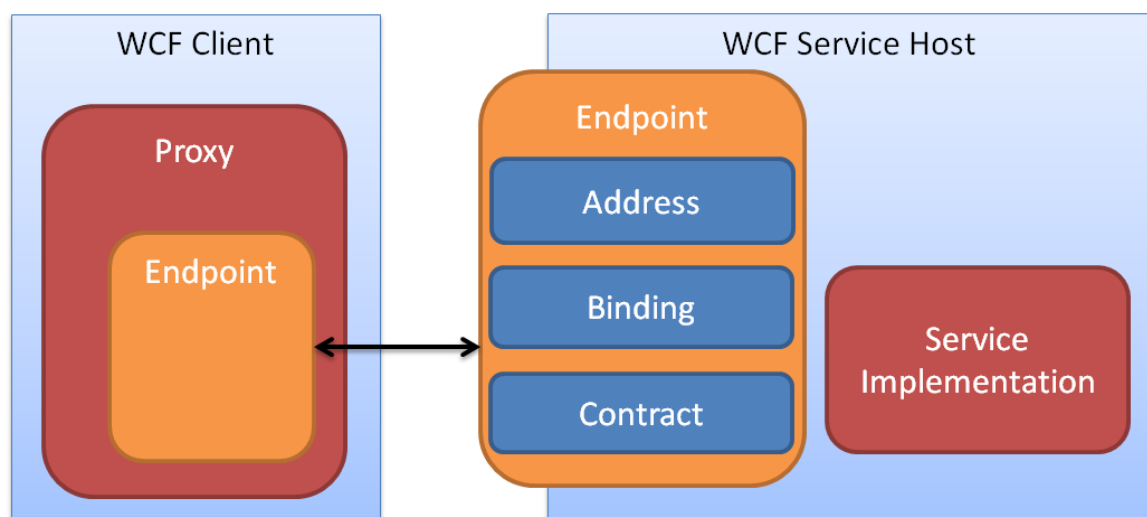
### More information

For a more in-depth introduction to WCF, take a look at the "[Introducing Windows Communication Foundation](#)" white paper.

When you create a WCF service, you specify how to communicate with the service by defining an endpoint (either in a configuration file or in code). An endpoint definition consists of the following:

1. An address where you can find the endpoint.
2. A binding that specifies how you can communicate with the endpoint. A binding definition includes details of the communication protocol to use, how you should encode the message, and any necessary security requirements.
3. A contract that identifies the operations that are available at the endpoint.

Figure 2 illustrates the relationships between these components.



**Figure 2. High-level WCF architecture**

A wide selection of bindings are available “out of the box,” so WCF enables you to define various types of Web service and interoperate with a wide selection of platforms, including Java-based clients and services. The following sections describe three of the types of Web service that you can implement with WCF.

### SOAP-Based Web Services

In the Java world, the JAX-WS and SOAP with Attachments API for Java (SAAJ) specifications define a model for creating Web services that work by exchanging SOAP messages over HTTP. This style of Web service uses Web Services Description Language (WSDL) documents to exchange metadata about the service between the client and server. One of the advantages of this style of Web service is that the metadata makes the operations in the service discoverable, which in turn facilitates tools that can, for example, automatically generate proxies for Web service clients. To implement this style of Web service in WCF, you can define a class that exposes the required functionality and mark it with the **ServiceContract** attribute. You identify the methods that you want to expose as operations that are available to clients by annotating them with the **OperationContract** attribute. The WCF runtime uses these attributes to generate the appropriate WSDL description of your service, and to route requests that are sent to the host environment to the appropriate method in a service.

### RESTful Web Services

REST is an alternative style of Web service, based on the Representative State Transfer concept. It uses URIs to identify resources and HTTP verbs to specify operations. It



Did you know?

WCF Data Services uses a RESTful API that enables you to easily build applications that can consume data over the Web by using any technology that operates with REST.

typically uses formats such as XML or JSON as the message payload. The RESTful approach is often seen as a more lightweight approach to building Web services than using SOAP because it requires less complicated infrastructure code to make it work. The JAX-RS specification defines a Java API and annotations for implementing RESTful Web services. You implement a WCF REST Web service in a similar manner, again by defining a class that provides methods that return data (resources), and tagging these methods with the **WebGet** attribute. The **WebGet** attribute specifies the URI that a client application should query to invoke a method and

---

retrieve the data that this method returns.

## Silverlight and WCF RIA Services

[Microsoft Silverlight](#)® enables you to create rich Internet applications that run in most Web browsers by using an installed browser plug-in to provide the runtime environment. Silverlight supports vector graphics and video, and many Silverlight applications use data and resources that are hosted on a server. Its rich set of controls, including PivotViewer for displaying large amounts of data, make it a highly productive development platform. [WCF RIA Services](#) is designed to simplify building such applications by providing a pattern that enables you to write application logic to run on the middle tier, which a Silverlight application that is running in the browser can invoke. This provides a middle-tier platform that is optimized for rich Internet applications instead of working with standard WCF services or servlets in the Java world. You may be familiar with JavaFX or Apache Pivot for building rich Internet applications in the Java world (both extend the functionality that is available with Java applets,) or with Adobe Flash and Adobe ActionScript. When you use JavaFX and Adobe Flash, you use a scripting language, whereas when you use Silverlight, you use one of the compiled .NET languages. Silverlight also includes a much richer collection of controls than either JavaFX or Apache Pivot. In addition, you can use Microsoft Expression Blend® to design the user interface for your Silverlight application.



### Did you know?

In addition to supporting rich Internet applications, Silverlight is also the primary development environment for the Windows Phone 7 platform. You can also use it to develop desktop applications.

## Summary

There are many similarities between the technologies that are available for developing Web applications by using the .NET Framework and Java. You can develop any Web application by using the .NET platform that you can develop by using Java. However, the tools that are available for .NET often speed up that development process. This paper has summarized some of the key Web technologies that you can use to develop .NET Framework Web applications.

As a reference, the following table provides a high-level mapping between the Java and .NET technologies that are discussed in this paper.

Topic	Java technology	.NET technology
Web site development	JSP pages and servlets	ASP.NET Web Forms
Large-scale Web site development	Struts, Spring, or JavaServer Faces	ASP.NET MVC
Server-side Web controls	Tag libraries	ASP.NET controls
SOAP-based Web services	JAX-WS and SAAJ	WCF
RESTful Web services	JAX-RS	WCF
Rich Internet applications	JavaFX, Apache Pivot, or Java applets (or Adobe Flash)	Silverlight
Middle-tier rich Internet application components	Servlets	WCF RIA Services