Introduction to Windows Technologies for Building Enterprise Applications

...for Java developers

Introduction

You can use the Java Platform, Standard Edition (Java SE) toolset to build basic desktop applications and services. However, constructing enterprise solutions frequently requires you to incorporate additional technologies such as Enterprise JavaBeans, Web services, databases, workflow management, security, and many other items. These features are not provided as part of the Java SE package. Instead, you can build your own components or, more commonly, integrate software and services that third parties have developed.

The Microsoft® .NET Framework facilitates an arguably more integrated approach than that available to Java developers. It provides tools and services out of the box that enable you to build and host enterprise components, implement complex business workflows, store and retrieve data by using a database, and manage security. If you require more extensive features than the features that the .NET Framework provides directly, you can incorporate server products such as Microsoft BizTalk® Server to implement interbusiness workflow, Microsoft Host Integration Server to interchange data with corporate servers and mainframes, and Windows Server® AppFabric to provide extended hosting services and scalable caching for Web services.

This paper is not intended to provide a detailed description of how to develop enterprise applications by using the .NET Framework. Instead, it is intended to help you familiarize yourself with the array of technologies that are available for enterprise development on the Windows® platform. The assumption is that you will primarily be using the .NET Framework to develop your applications, although many of the services that are described have application programming interfaces (APIs) that are available in other languages, or facilitate interoperability between platforms. This paper describes a high-level mapping between the enterprise technologies that Java solutions commonly use and the enterprise technologies that are available in Windows and the .NET Framework.

Contents

Web Development

ASP.NET is the umbrella term for the collection of .NET Framework technologies that you can employ to develop enterprise-class Web applications.

You create ASP.NET Web Forms pages by combining HTML-like markup to define the appearance of the page, and code (that you can write in any of the languages that the .NET Framework supports) to define the logic. This model is very similar to the Java Server Pages (JSP) scheme where you implement the markup that defines the layout of a Web page in one file and code the logic behind the page in separate Java Bean classes. ASP.NET also includes a comprehensive set of server-side controls to simplify the implementation of your Web applications.

The ASP.NET Web Forms page model is so flexible that ASP.NET does not require an equivalent to Java Platform, Enterprise Edition (Java EE) servlets; you simply create an ASP.NET Web Form with no markup. For even more flexibility, you can create a custom **HttpHandler** class.

Figure 1 shows a comparison of the key components in a Java Web application and an ASP.NET Web application.

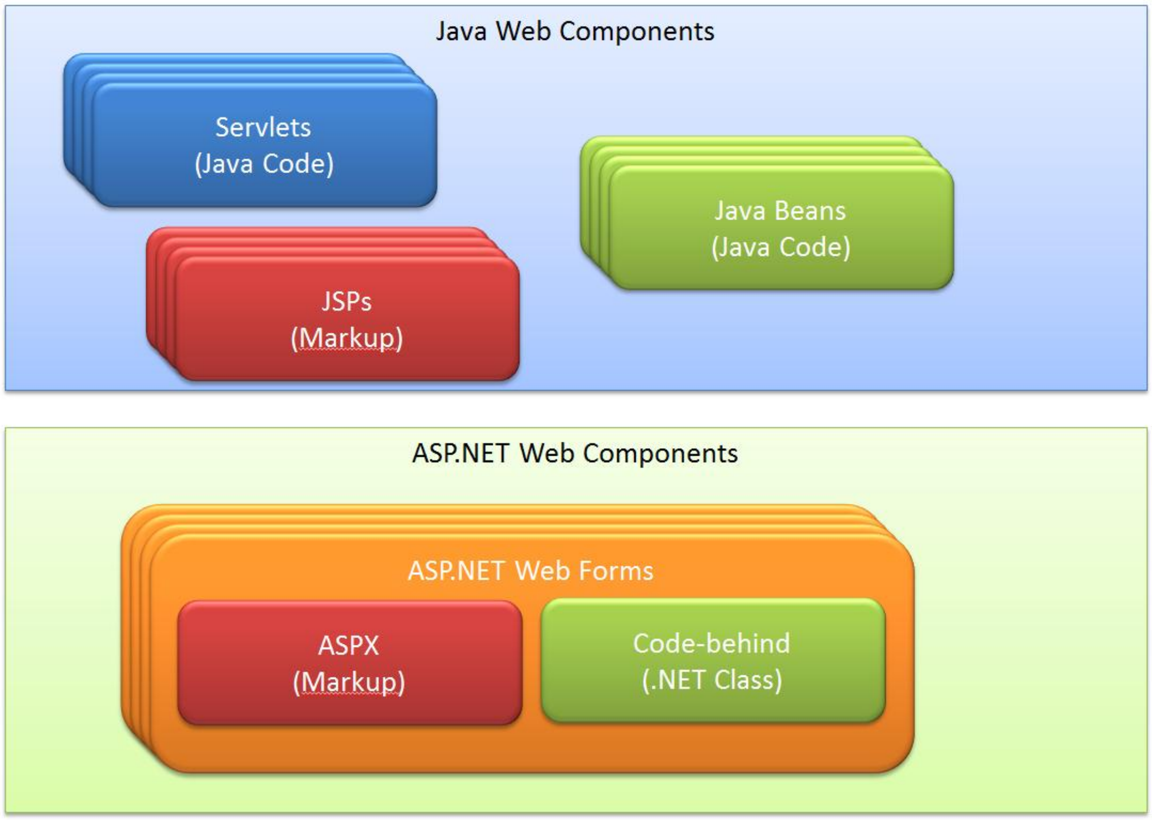

Figure 1. Web application components

ASP.NET MVC offers you an alternative approach to ASP.NET Web Forms for building Web applications. ASP.NET MVC is based on the Model-View-Controller (MVC) pattern, which is often seen as a more robust model for building, testing, and maintaining large Web-based applications. This MVC pattern is the same as that used as the basis for popular Java frameworks such as Struts, Spring MVC, and JavaServer Faces, although the implementation is different.

Both ASP.NET Web Forms and ASP.NET MVC are supported by the tight integration of the development tool (Microsoft Visual Studio®) and the deployment platform (Internet Information Services (IIS)).

Web Services

Many enterprise applications use 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 parts of the various SOAP and REST Web services standards are available through Windows Communication Foundation (WCF), a unified programming model for building service-oriented applications. WCF significantly extends ASP.NET Web services beyond SOAP over HTTP to enable you to use a much wider selection of message formats and transport protocols from a common programming model. WCF supports interoperability with applications that are developed by using the JAX-WS and JAX-RS standards in addition to specific Windows and .NET protocols and formats.

Workflow

Windows Workflow Foundation, also a part of the .NET Framework, enables you to build executable models of real-world, long-running business processes. You can design these processes by using a graphical designer that is provided with Visual Studio. In addition, you can host these executable models by using IIS or your own custom applications. Figure 2 shows an example workflow in the Visual Studio workflow designer.

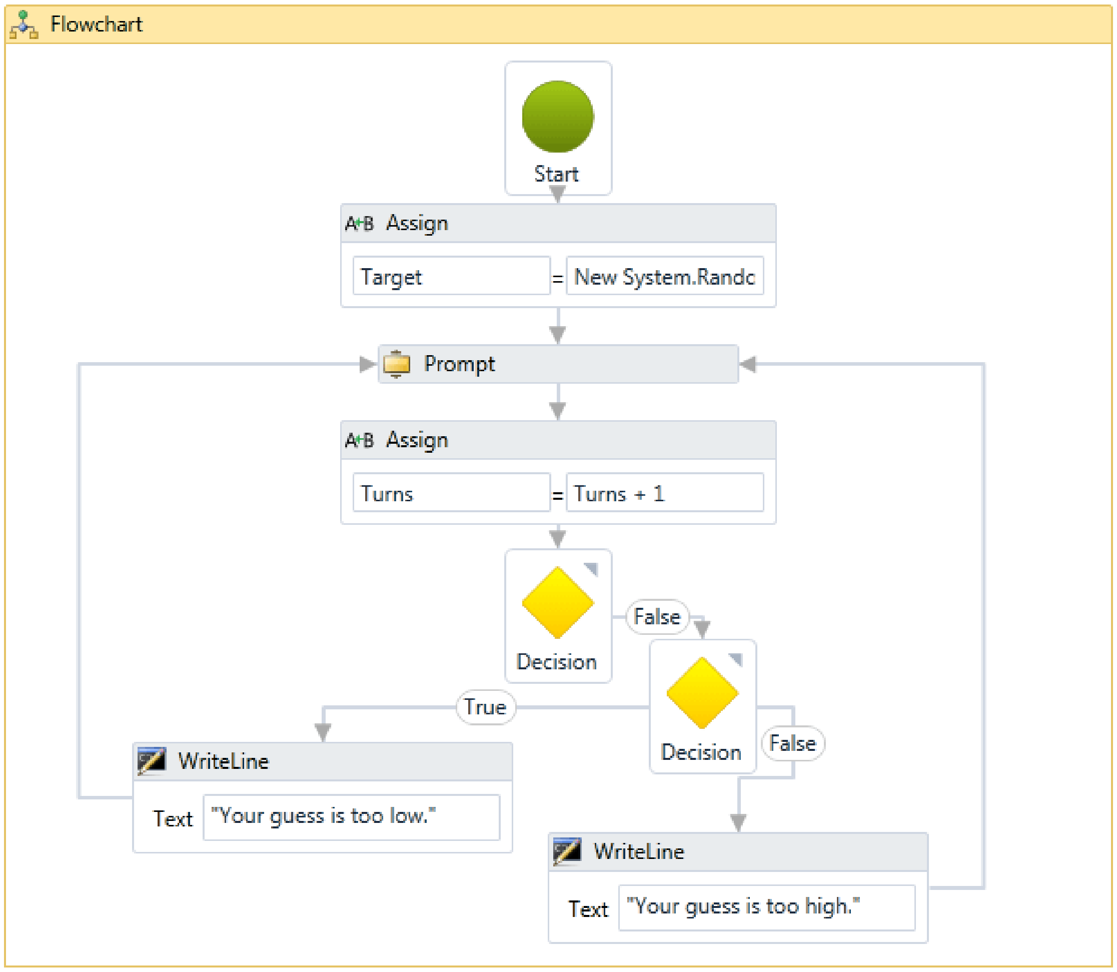

Figure 2. Example workflow in the workflow designer

Although the Java EE specifications do not explicitly include workflow, many Java EE application servers include support for running business processes that are defined by using Business Process Execution Language (BPEL) or Business Process Model Notation (BPMN), such as the JBoss jBPM suite.

Data Access

Most enterprise applications require some kind of persistent data store, very often in the form of a relational database. In the .NET Framework, the ADO.NET classes enable programmatic access to a wide variety of relational databases in the same way that Java enables you to access relational databases through the Java Database Connectivity (JDBC) API. The low-level data access model in ADO.NET is similar in concept to the Java model. For example, in ADO.NET, connection and command objects fulfill many of the same roles that connection and statement objects fulfill in JDBC. ADO.NET also includes a higher-level abstraction called the DataSet, which facilitates working with disconnected data.

ADO.NET includes an object relational mapping (ORM) framework called the Entity Framework that enables you to work at a high level of abstraction from the underlying data store. This is similar to the approach that Java frameworks such as Java Data Objects (JDO) or Hibernate take.

In addition to manipulating data programmatically by using the classes in the ADO.NET libraries, you can also use Language-Integrated Query (LINQ), which enables you to embed constructs that resemble Structured Query Language (SQL) in your application code that can be checked at compile time. LINQ relies on lambda expressions, which is a feature that is planned for a future release of the Java language.

Did you know?

The Entity Framework uses a set of XML models to define the object relational mapping.

Also, Visual Studio includes a designer tool for working with these models.

Message Queues

Windows includes a robust message queue service called Message Queuing (also known as MSMQ) that enables you to use asynchronous, durable messaging in your application. For a Java EE application, your application server, or third-party solutions such as Apache ActiveMQ, provide this functionality and access it through the Java Message Service (JMS) API. The .NET Framework provides basic message queuing functionality through the classes in the **System.Messaging** namespace, but WCF also includes a set of Message Queuing bindings that you can use to expose or consume Message Queuing queues. Java EE also includes message-driven beans that are configured to permanently listen for messages on a message queue. You can achieve a similar result in Windows by creating a Windows service that listens for messages on a Message Queuing message queue. JMS defines two messaging models: a point-to-point model that uses queues, and a publish-subscribe model that uses topics. Although Message Queuing only offers a point-to-point model, you can easily implement the publish-subscribe model by using WCF.

Did you know?

You can handle federation with external security providers by using the Windows Identity Framework classes to implement a claims-based identity solution.

Authentication and Authorization

Active Directory® is the core security platform in Windows that provides the authentication and authorization services that a Lightweight Directory Access Protocol (LDAP) server often provides to a Java application. You can easily configure an ASP.NET application to integrate with Active Directory, or with other security service providers.

ASP.NET also enables you to build your own custom user management and authorization solutions through the membership and role providers included in ASP.NET.

Platforms

In addition to the APIs and services that are available as part of the .NET Framework, it is useful to understand how Microsoft server applications might host your enterprise applications. In the Java EE world, you can select a Java EE application server that meets your requirements. Then, you can look for other products that offer additional services that your application might need (for example, a business process workflow engine) and integrate them. In Windows world, most enterprise application services that your application might need (for example, the Web server or message queuing) are provided as a part of the server operating system, and are designed to work together to minimize the integration effort that is required.

Where the server platform does not include specific functionality, Microsoft offers additional products such as BizTalk Server, Microsoft SharePoint® Server, Microsoft Exchange Server, and a host of others that can fill the gaps (for a complete list, see the Microsoft Server and Tools Web page). That's not to say that enterprise applications for Windows necessarily use exclusively Microsoft technologies; plenty of third-party components and server applications are available that offer specialized services such as enterprise resource planning, security, or document management in the enterprise domain.

Summary

As you can now see, there are many similarities between developing enterprise applications by using the .NET Framework and Java. This paper has summarized some of the key enterprise technologies that you can use in your .NET Framework applications.