msdn® training

Module 1: Overview of the Microsoft .NET Platform

Contents

This course is based on the prerelease version (Beta 2) of Microsoft® Visual Studio® .NET Enterprise Edition. Content in the final release of the course may be different from the content included in this prerelease version. All labs in the course are to be completed with the Beta 2 version of Visual Studio .NET Enterprise Edition.

Microsoft

Instructor Notes

Presentation:
60 Minutes

Lab:
00 Minutes

This module provides students with an overview of the Microsoft® .NET Platform. It defines some of the terminology specific to the .NET Platform and describes its key features and benefits.

The module starts with an overview of the .NET Platform, and then introduces the .NET Framework and services. It describes the design goals of the .NET Framework and provides other information about the major components. Finally, students will learn about the programming languages that can be used with the .NET Framework and about the major enhancements to Microsoft Visual Basic® .NET version 7.0.

This module contains no labs.

After completing this module, students will be able to:

- List the main elements of the .NET Platform.
- Describe the .NET Framework and its components.
- List the major enhancements to Visual Basic .NET.

Materials and Preparation

This section provides the materials and preparation tasks that you need to teach this module.

Required Materials

To teach this module, you need the following materials:

- Microsoft PowerPoint® file 2373A_01.ppt
- Module 1, “Overview of the Microsoft .NET Platform”

Preparation Tasks

To prepare for this module, you should:

- Read all of the materials for this module.
- Read the instructor notes and the margin notes for the module.

Module Strategy

This first module must set the tone for the rest of the course. Visual Basic developers may initially be hesitant to upgrade their skills to Visual Basic .NET because of the substantial changes to the product. While it may take students a bit longer to adapt to this version of Visual Basic, the advantages are significant, and these advantages must be made clear to students from the very beginning. Use the following strategy to present this module:

- What Is the Microsoft .NET Platform?

  Discuss the core services and technologies provided by the .NET Platform. Briefly talk about the .NET Framework. It is covered in more detail in the next lesson.

- What Is the .NET Framework?

  Ensure that students understand the function that the Common Language Specification plays in language development for the .NET Platform. Stress the fact that the existing model provides no consistency in the availability of application programming interfaces (APIs). The choice of programming model also becomes the choice of language. So if you are a Visual Basic developer, you are limited by the set of functionality that it provides for the platform. Or if you are a C++ developer, your existing skills do not transfer to Active Server Pages (ASP) development.

  However, because this course focuses on Visual Basic development, point out that most students will already be familiar with many .NET concepts, such as ASP development and Web Services, Microsoft Visual Basic Scripting Edition, and Simple Object Access Protocol (SOAP)/ Visual Basic 6.0.

  Mention that Visual Basic has been upgraded significantly to support the .NET Platform. Explain that C# is a new language specifically created for the .NET Framework and that C++ has been enhanced to support the .NET Platform. Point out that several third-party languages are also .NET enabled, and that more will follow over time.

- What Are the .NET Framework Components?

  Briefly describe each of the .NET Framework components. Point out to the students each area where Visual Basic developers can benefit from the .NET Framework components, such as garbage collection, type checking, and exception handling.

  ADO .NET is also an important new tool for Visual Basic .NET developers, and they should not find this tool difficult if they are familiar with previous versions of ActiveX® Data Objects (ADO).

  Web Forms and Web Services are of course some of the major new features of the .NET Platform. Students should be aware that they no longer need to learn a scripting language in order to create Web applications; they can now use Visual Basic .NET, using their existing Visual Basic skills.

  Microsoft Windows-based applications are not to be forgotten in Visual Basic .NET. Students need to realize that there are many new and exciting aspects of the new Windows® Forms development.

- What Are the Visual Basic .NET Enhancements?

  Point out that the course focuses on Microsoft Visual Studio® .NET from the perspective of a Visual Basic developer. Describe the main areas of enhancement, but make it clear that they are covered in detail in the remainder of the course.

Overview

Topic Objective
To provide an overview of the module topics and objectives.

Lead-in
In this module, you will be introduced to the .NET Platform. You will then learn about the .NET Framework and the language support that the .NET Framework provides for the platform.

- **What Is the Microsoft .NET Platform?**
- **What Is the .NET Framework?**
- **What Are the .NET Framework Components?**
- **What Are the Visual Basic .NET Enhancements?**

Delivery Tip
This module provides an overview of the .NET systems for Visual Basic developers.

Make sure to point out the advantages of .NET over previous versions of Windows development architectures and Visual Basic whenever possible.

Microsoft® Visual Basic® .NET version 7.0 is a major enhancement to the Visual Basic product line. As a Visual Basic developer, you will find it useful to understand the rationale and features that provide the foundation for the Microsoft .NET Platform before you look at Visual Basic .NET in detail.

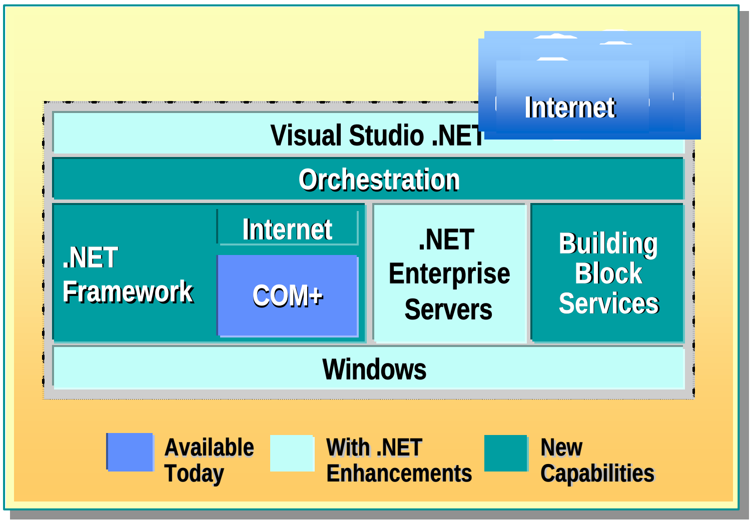
After completing this module, you will be able to:

- List the main elements of the .NET Platform.
- Describe the .NET Framework and its components.
- List the major enhancements to Visual Basic .NET.

What Is the Microsoft .NET Platform?

Topic Objective
To discuss the core services and features provided by the .NET Platform.

Lead-in
The .NET Platform consists of several core services and features, as shown on this slide. One goal of this new platform is to simplify Web development. This topic will provide a brief look at the architecture of the .NET Platform.



Internet

Visual Studio .NET

Orchestration

.NET Framework

Internet

COM+

.NET Enterprise Servers

Building Block Services

Windows

Available Today

With .NET Enhancements

New Capabilities

This topic describes the components of the .NET Platform, including the .NET Framework, the .NET Building Block Services, .NET Enterprise Servers, and Microsoft Visual Studio® .NET. The .NET Platform is important to you because the goal of the .NET Platform is to simplify Web development by providing all of the tools and technologies that you need to build distributed Web applications.

The .NET Platform provides:

- A language-independent, consistent programming model across all tiers of an application.
- Seamless interoperability between technologies.
- Easy migration from existing technologies.
- Full support for the Internet's platform-neutral, standards-based technologies, including Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP).

The .NET Platform

The .NET Platform is a set of technologies designed to transform the Internet into a full-scale distributed computing platform. It provides new ways to build applications from collections of Web Services. The .NET Platform fully supports the existing Internet infrastructure, including HTTP, XML, and SOAP.

Core Technologies in the .NET Platform

The core technologies that make up the .NET Platform include:

- .NET Framework

  The .NET Framework is based on a new common language runtime. This runtime provides a common set of services for projects built in Visual Studio .NET, regardless of the language. These services provide key building blocks for applications of any type, across all application tiers.

  Microsoft Visual Basic, Microsoft Visual C++®, and other Microsoft programming languages have been enhanced to take advantage of these services. Third-party languages that are written for the .NET Platform also have access to the same services.

Delivery Tip
Because of the common language runtime, all .NET-compatible languages will use the same run-time files. This greatly aids Visual Basic developers because they will no longer need to distribute run-time libraries that are specific to Visual Basic. These files will be installed automatically in future versions of Windows.

Note Because of the common language runtime, all .NET languages will use the same run-time files. This means that there is no need to distribute Visual Basic –specific run-time libraries because .NET run-time files will be installed automatically in future versions of Microsoft Windows®.

- .NET Building Block Services

  The .NET Building Block Services are distributed programmable services that are available both online and offline. A service can be invoked on a stand-alone computer not connected to the Internet, provided by a local server running inside a company, or accessed by means of the Internet. The .NET Building Block Services can be used from any platform that supports SOAP. Services include identity, notification and messaging, personalization, schematized storage, calendar, directory, search, and software delivery.

- Visual Studio .NET

  Visual Studio .NET provides a high-level development environment for building applications on the .NET Framework. It provides key enabling technologies to simplify the creation, deployment, and ongoing evolution of secure, scalable, highly available Web applications and Web Services. It also enables a new generation of Windows-based applications with many new features available through the .NET Framework.

- .NET Enterprise Servers

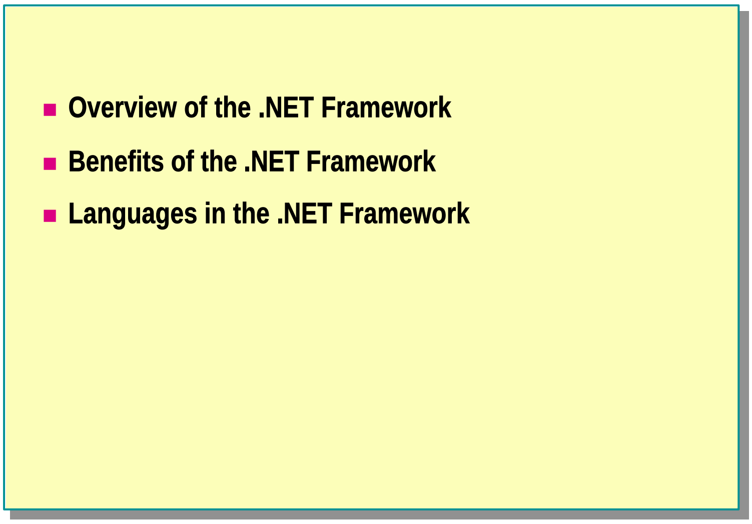
  The .NET Enterprise Servers provide scalability, reliability, management, and integration within and across organizations, and many other features as described in the following table.

Server	Description
Microsoft SQL Server™ 2000	Includes rich XML functionality, support for Worldwide Web Consortium (W3C) standards, the ability to manipulate XML data by using Transact SQL (T-SQL), flexible and powerful Web-based analysis, and secure access to your data over the Web by using HTTP.
Microsoft BizTalk™ Server 2000	Provides enterprise application integration (EAI), business-to-business integration, and the advanced BizTalk Orchestration technology to build dynamic business processes that span applications, platforms, and organizations over the Internet.
Microsoft Host Integration Server 2000	Provides the best way to embrace Internet, intranet, and client/server technologies while preserving investments in existing systems . Microsoft Host Integration Server 2000 is the replacement for SNA Server.
Microsoft Exchange 2000 Enterprise Server	Builds on the powerful Exchange messaging and collaboration technology by introducing several important new features and further increasing the reliability, scalability, and performance of its core architecture. Other features enhance the integration of Exchange 2000 with Microsoft Windows 2000, Microsoft Office 2000, and the Internet.
Microsoft Application Center 2000	Provides a deployment and management tool for high-availability Web applications.
Microsoft Internet Security and Acceleration Server 2000	Provides secure, fast, and manageable Internet connectivity. Internet Security and Acceleration Server integrates an extensible, multilayer enterprise firewall and a scalable, high-performance Web cache. It builds on Windows 2000 security and directory for policy-based security, acceleration, and management of internetworking.
Microsoft Commerce Server 2000	Provides an application framework, sophisticated feedback mechanisms, and analytical capabilities.

◆ What Is the .NET Framework?

Topic Objective
To provide an overview of the topics covered in this lesson.

Lead-in
The .NET Framework provides all the common services you need to enable your applications to run.



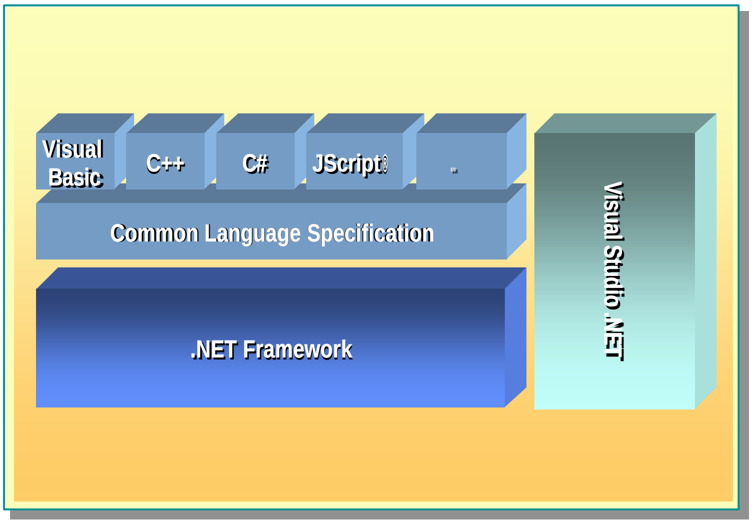
- Overview of the .NET Framework
- Benefits of the .NET Framework
- Languages in the .NET Framework

In this lesson, you will learn how the .NET Framework provides all common services required for your applications to run. You will learn how services are available in any .NET-compatible language through the common language specification (CLS). You will also learn what languages are supported in the .NET Framework.

Overview of the .NET Framework

Topic Objective
To explain how various components of the .NET Framework fit together.

Lead-in
The .NET Framework provides all common services required to run your Visual Basic .NET applications. These services are available in any .NET-compatible language through the Common Language Specification.



The .NET Framework provides all the common services required to run your Visual Basic .NET applications.

Building Components in the .NET Framework

Before COM, applications were completely separate entities with little or no integration. Using COM, you can integrate components within and across applications by exposing common interfaces. However, as a developer, you must still write the code to wrap, manage, and clean up after components and objects.

In the .NET Framework, components are built on a common foundation. You no longer need to write the code to allow objects to interact directly with each other. In the .NET environment, you no longer need to write component wrappers because components do not use wrappers. The .NET Framework can interpret the constructs that developers are accustomed to using in object-oriented languages. The .NET Framework fully supports class, inheritance, methods, properties, events, polymorphism, constructors, and other object-oriented constructs.

The Common Language Specification

The Common Language Specification (CLS) defines the common standards to which languages and developers must adhere if they want their components and applications to be widely useable by other .NET-compatible languages. The CLS allows Visual Basic .NET developers to create applications as part of a multiple-language team with the knowledge that there will be no problems integrating the different languages. The CLS even allows Visual Basic .NET developers to inherit from classes defined in different languages.

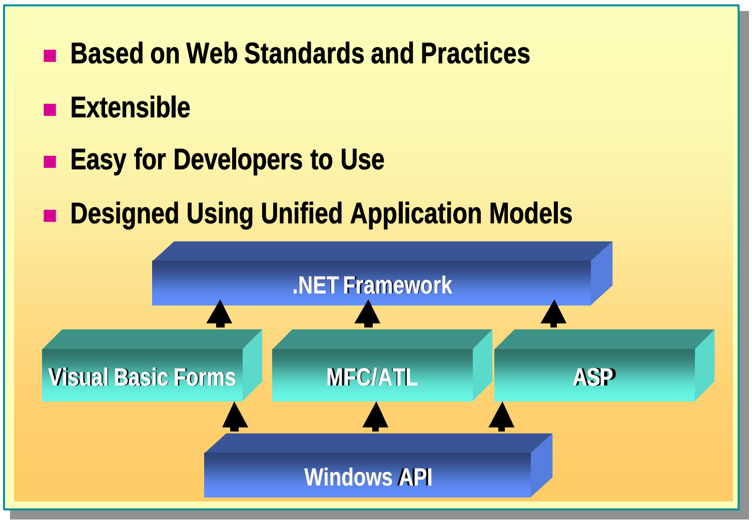
Visual Studio .NET

In the .NET Framework, Visual Studio .NET provides the tools you can use for rapid application development.

Benefits of the .NET Framework

Topic Objective
To explain how the framework design goals make it easy to build applications.

Lead-in
Before you learn about the various .NET Framework components, it is helpful to understand the design goals of the .NET Framework. This will help you to see how the .NET Framework simplifies software development.



- Based on Web Standards and Practices
- Extensible
- Easy for Developers to Use
- Designed Using Unified Application Models

.NET Framework

Visual Basic Forms

MFC/ATL

ASP

Windows API

In this topic, you will learn about some of the benefits of the .NET Framework.

Goals of the .NET Framework

The .NET Framework was designed to meet the following goals:

- Based on Web standards and practices

  The .NET Framework fully supports the existing Internet technologies, including Hypertext Markup Language (HTML), XML, SOAP, Extensible Stylesheet Language for Transformations (XSLT), XPath, and other Web standards. The .NET Framework favors loosely connected, stateless Web services.

- Extensible

  The hierarchy of the .NET Framework is not hidden from the developer. You can access and extend .NET classes (unless they are sealed) by using inheritance. You can also implement cross-language inheritance.

Delivery Tip
Mention that .NET Framework classes that are sealed cannot be extended.

- Easy for developers to use

  In the .NET Framework, code is organized into hierarchical namespaces and classes. The Framework provides a common type system, referred to as the unified type system, that is used by any .NET-compatible language. In the unified type system, everything is an object. There are no variant types, there is only one string type, and all string data is Unicode.

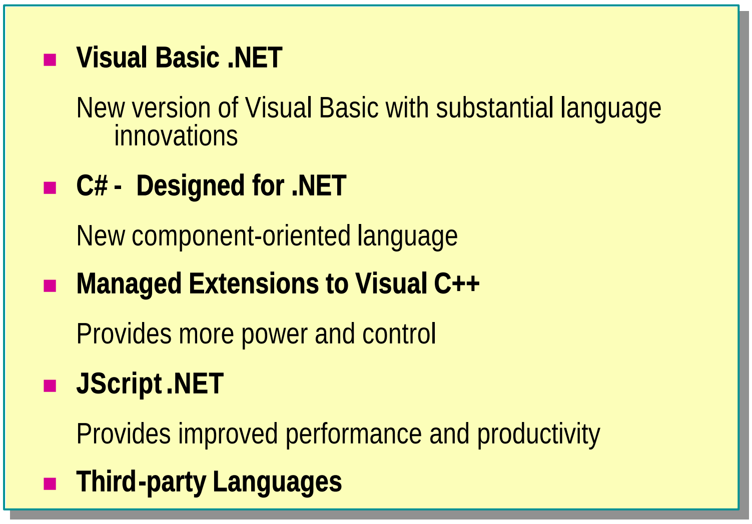
- Designed using unified application models

  A .NET class' functionality is available from any .NET-compatible language or programming model.

Languages in the .NET Framework

Topic Objective
To discuss the languages that are supported in the .NET Framework.

Lead-in
In this topic, you will learn about the languages that Microsoft is providing for the .NET Framework with Visual Studio .NET. You'll also learn about some of the languages for .NET that are being created by third-party developers.



- **Visual Basic .NET**

  New version of Visual Basic with substantial language innovations

- **C# - Designed for .NET**

  New component-oriented language

- **Managed Extensions to Visual C++**

  Provides more power and control

- **JScript .NET**

  Provides improved performance and productivity

- **Third-party Languages**

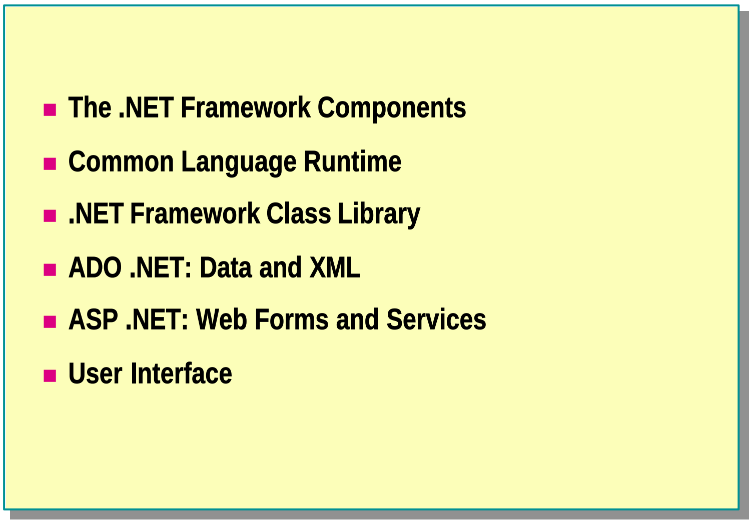
The .NET Platform is language neutral—all common services needed to enable programs to run are provided by the .NET Framework. The .NET Framework provides support for numerous programming languages, including Microsoft and third-party languages.

Language	Description
Microsoft Visual Basic .NET	Visual Basic .NET provides substantial language innovations over previous versions of Visual Basic. Visual Basic .NET supports inheritance, constructors, polymorphism, overloading, structured exceptions, stricter type checking, free threading, and many other features. With this release, Visual Basic Scripting Edition provides full Visual Basic functionality.
Microsoft Visual C#™	C# was designed for the .NET Platform and is the first modern component-oriented language in the C and C++ family. It can be embedded in ASP .NET pages. Some of the key features of this language include classes, interfaces, delegates, boxing and unboxing, namespaces, properties, indexers, events, operator overloading, versioning, attributes, unsafe code, and XML documentation generation. No header or Interface Definition Language (IDL) files are needed.
Managed Extensions C++	The Visual C++ upgrade is a managed, minimal extension to the C++ language. This extension provides access to the .NET Framework that includes garbage collection, single-implementation inheritance, and multiple-interface inheritance. This upgrade also eliminates the need to write plumbing code for components. It offers low-level access where useful.
Microsoft JScript® .NET	JScript .NET has been rewritten to be fully .NET compatible. It includes support for classes, inheritance, types, and compilation. It provides improved performance and productivity features. JScript .NET is also integrated with Visual Studio .NET. You can use of any .NET Framework class in JScript .NET.
Third-party languages	Various third-party languages are supporting the .NET Platform. These languages include APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, and SmallTalk.

◆ What Are the .NET Framework Components?

Topic Objective
To provide an overview of the topics covered in this lesson.

Lead-in
In this lesson, you will learn about the different components that make up the .NET Framework.



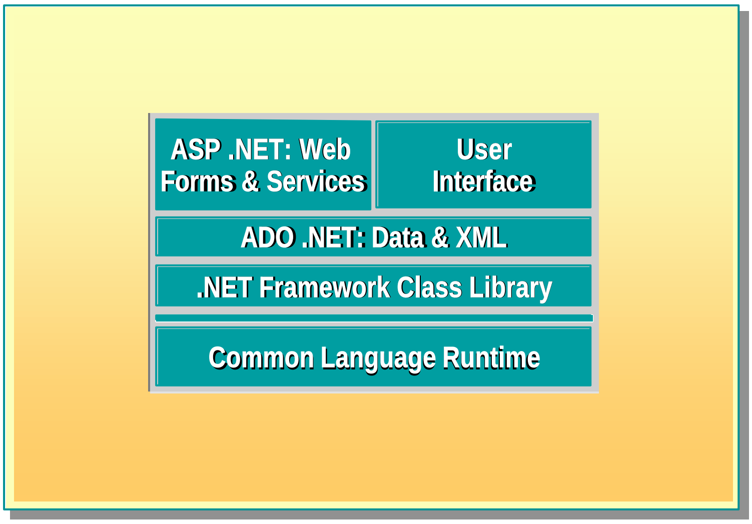
- The .NET Framework Components
- Common Language Runtime
- .NET Framework Class Library
- ADO .NET: Data and XML
- ASP .NET: Web Forms and Services
- User Interface

In this lesson, you will learn about the Microsoft .NET Framework components. The .NET Framework is a set of technologies that are an integral part of the .NET Platform. The .NET Framework provides the basic building blocks to develop Web applications and Web Services by using ASP .NET.

The .NET Framework Components

Topic Objective
To provide an overview of the .NET Framework components.

Lead-in
Visual Basic .NET uses these components that are provided by the .NET Framework.



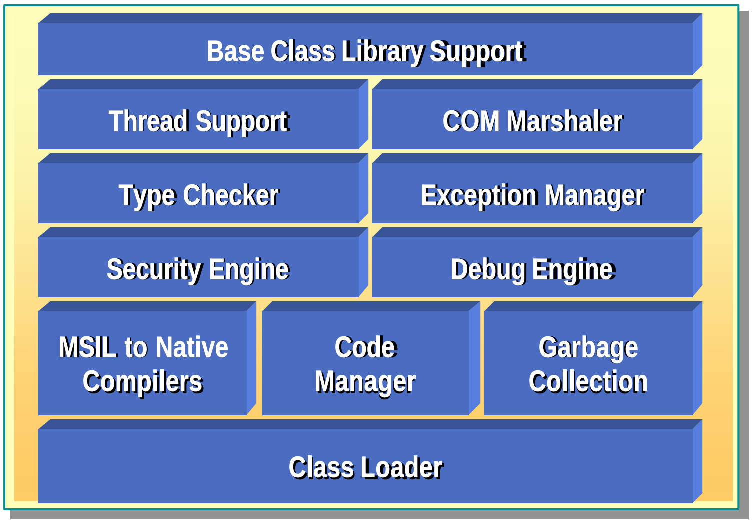
The components of the .NET Framework are as follows:

- Common language runtime
- .NET Framework Class Library
- ADO .NET: data and XML
- ASP .NET: Web Forms and Services
- User interface

Common Language Runtime

Topic Objective
To highlight some of the key components of the common language runtime.

Lead-in
This topic will give you an overview of the various components of the common language runtime. Each component will be briefly described. As a Visual Basic developer, you will never physically see these pieces. But discussing them gives you a better understanding of the richness of the runtime.



Base Class Library Support

Thread Support

COM Marshaler

Type Checker

Exception Manager

Security Engine

Debug Engine

MSIL to Native Compilers

Code Manager

Garbage Collection

Class Loader

Delivery Tip
Many of these issues are directly beneficial to Visual Basic .NET developers and will be highlighted throughout the course.

The Common Language Runtime simplifies application development, provides a robust and secure execution environment, supports multiple languages, simplifies application deployment and management, and provides a managed environment.

What Is a Managed Environment?

A managed environment is one in which the environment provides common services automatically. Examples of the types of services a managed environment provides are garbage collection and security.

Common Language Runtime Components

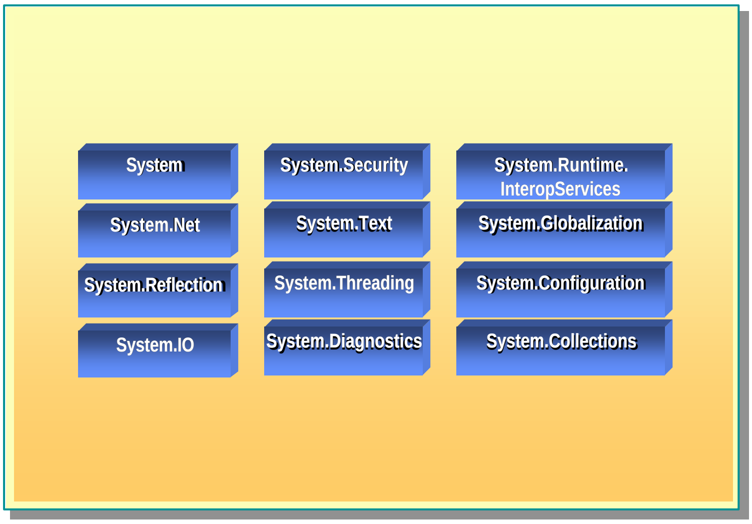
The Common Language Runtime features are described in the following table.

Component	Description
Class loader	Loads the implementation of a loadable type into memory and prepares it for execution.
Microsoft intermediate language (MSIL) to native compiler	Converts MSIL to native code (just-in-time).
Code manager	Manages code execution.
Garbage collection	Provides automa tic lifetime management of all of your objects. This is a multiprocessor, scalable garbage collector.
Security engine	Provides evidence-based security based on the origin of the code as well as the user.
Debug engine	Allows you to debug your application and trace the execution of code.
Type checker	Will not allow unsafe casts or uninitialized variables. IL can be verified to guarantee type safety.
Exception manager	Provides structured exception handling, which is integrated with Windows Structured Exception Handling (SEH). Error reporting has been improved.
Thread support	Provides classes and interfaces that enable multithreaded programming.
COM marshaler	Provides marshaling to and from COM.
.NET Framework Class Library support	Integrates code with the runtime that supports the .NET Framework Class Library. The .NET Framework Class Library is covered in the following lesson.

.NET Framework Class Library

Topic Objective
To provide an overview of the .NET Framework Class Library and the most common namespace: **System**.

Lead-in
In this topic, you will learn how the .NET Framework Class Library exposes features of the runtime and provides other high-level services.



System
System.Security
System.Runtime.InteropServices
System.Net
System.Text
System.Globalization
System.Reflection
System.Threading
System.Configuration
System.IO
System.Diagnostics
System.Collections

Delivery Tip
The .NET Framework Class Library provides many powerful new features for Visual Basic .NET developers.

For example, the **Collections** namespace adds many new possibilities, such as sorting, queues, stacks, and auto-sizing arrays.

The **Threading** system class also creates new possibilities to Visual Basic .NET developers for creating true multithreaded applications.

A simple overview of namespaces at this point may be useful.

The .NET Framework Class Library exposes features of the runtime and provides other high-level services that every programmer needs through an object hierarchy. This object hierarchy is known as a namespace.

System Namespaces

The **System** namespace contains fundamental classes and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions. Other classes provide services supporting data type conversion, method parameter manipulation, mathematics, remote and local program invocation, application environment management, and supervision of managed and unmanaged applications.

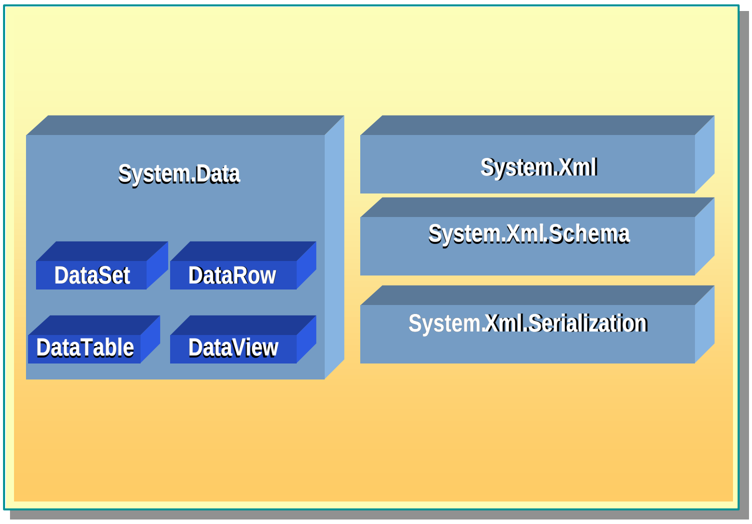
The **System.Collections** namespace provides sorted lists, hash tables, and other ways to group data. The **System.IO** namespace provides file I/O, streams, and so on. The **System.NET** namespace provides Transmission Control Protocol/Internet Protocol (TCP/IP) and sockets support.

For more information about namespaces, search for " namespaces" in the .NET Framework SDK documentation. Namespaces will also be covered in Module 2, "Development Environment Features," in Course 2373A, *Programming with Microsoft Visual Basic .NET (Prerelease).*

ADO .NET: Data and XML

Topic Objective
To explain the data and XML support in the runtime.

Lead-in
The .NET Framework provides a new set of ADO .NET classes to handle data.



Delivery Tip
ADO .NET has been created especially for Internet usage, as it uses XML as its core technology. This means that developers no longer need to pass **Recordset** objects around the system, potentially limiting which clients they can service. It also improves the disconnected programming model seen in previous versions of ADO.

Point out that Visual Basic developers' knowledge of ADO will greatly assist them in moving to ADO .NET.

Check students' understanding of basic XML concepts.

ADO .NET is the next generation of ActiveX® Data Object (ADO) technology. ADO .NET provides improved support for the disconnected programming model. It also provides rich XML support.

System.Data Namespace

The **System.Data** namespace consists of classes that constitute the ADO .NET object model. At a high level, the ADO .NET object model is divided into two layers: the connected layer and the disconnected layer.

The **System.Data** namespace includes the **DataSet** class, which represents multiple tables and their relations. These data sets are completely self-contained data structures that can be populated from a variety of data sources. One data source could be XML, another could be OLE DB, and a third data source could be the direct adapter for SQL Server.

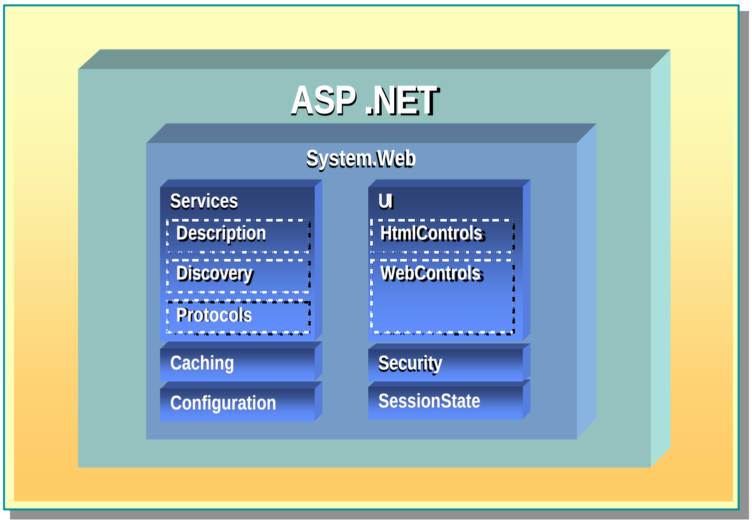
System.Xml Namespace

The **System.Xml** namespace provides support for XML. It includes an XML parser and a writer, which are both W3C compliant. Transformations are provided by the **System.Xml.Xsl** namespace and the implementation of XPath that allows data graph navigation in XML. The **System.XML.Serialization** namespace provides the entire core infrastructure for Web Services, including features such as moving back and forth between objects and an XML representation.

ASP .NET: Web Forms and Services

Topic Objective
To introduce Web Forms and Web Services.

Lead-in
The Internet is quickly evolving from today's Web sites that just deliver UI pages to browsers, to a next generation of programmable Web sites that link organizations, applications, services, and devices directly together. These programmable Web sites are more than passively accessed sites; they are reusable, intelligent Web Services.



ASP .NET is a programming framework that is built on the common language runtime and that can be used on a server to build powerful Web applications. ASP .NET Web Forms provide an easy and powerful way to build dynamic Web user interfaces (UIs). ASP .NET Web Services provide the building blocks for constructing distributed Web-based applications. Web Services are based on open Internet standards, such as HTTP and XML. You can think of a Web Service as a reusable component that is accessible across the Internet, rather than being limited to Windows clients on a local area network.

Delivery Tip
Web development becomes almost as simple as Windows-based application development with Visual Basic .NET.

Creating Web applications can be accomplished in the one IDE, whether they are Web page applications or Web Services.

The common language runtime provides built-in support for creating and exposing Web Services by using a programming abstraction that is consistent and familiar to both Active Server Pages (ASP) and Visual Basic developers. The resulting model is both scalable and extensible. This model is based on open Internet standards (HTTP, XML, SOAP, and SDL) so that it can be accessed and interpreted from any client or Internet-enabled device. Some of the more common ASP .NET classes are described in the following paragraphs.

System.Web Namespace

In the **System.Web** namespace, there are services such as caching, security, configuration, and others that are shared between Web Services and Web user interface (UI).

System.Web.Services Namespace

The **System.Web.Services** namespace handles Web Service requirements such as transportation protocols and service discovery.

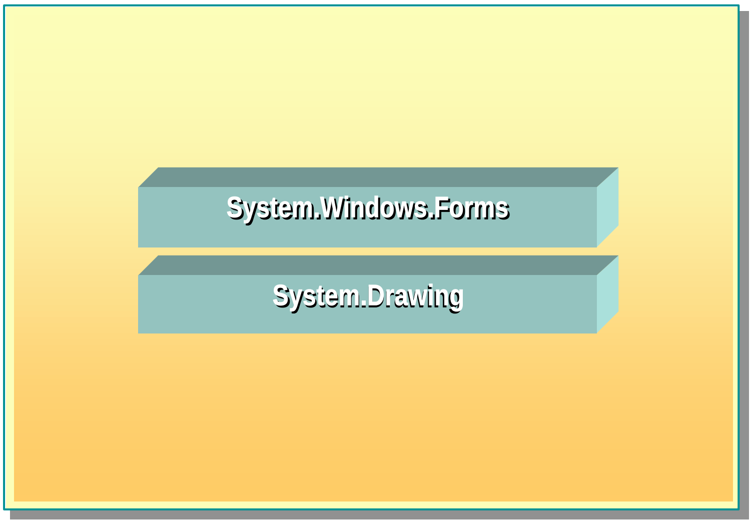
System.Web.UI Namespace

The **System.Web.UI** namespace provides two classes of controls: HTML controls and Web controls. The **HTMLControl**s provide direct mapping of HTML tags, such as <INPUT>. There are also **WebControl**s that are richer and allow you to structure controls with templates (for example, a grid control).

User Interface

Topic Objective
To explain how the .NET Framework handles the user interface for traditional Windows applications.

Lead-in
In this lesson, you will learn how .NET handles the user interface. The **System.Windows.Forms** class provides the client-side UI, and **System.Drawing** provides the next generation of GDI services.



Windows applications can provide more powerful user interfaces than ever by using the .NET Framework's **System.Windows.Forms** and **System.Drawing** namespaces. The new .NET Windows Forms will look very familiar to Visual Basic developers.

System.Windows.Forms Namespace

You can use the **System.Windows.Forms** namespace classes to build the client UI. This class lets you implement the standard Windows UI in your .NET applications. Many functions that were previously only accessible by means of application programming interface (API) calls are now available as part of the forms themselves, making development much easier and more powerful.

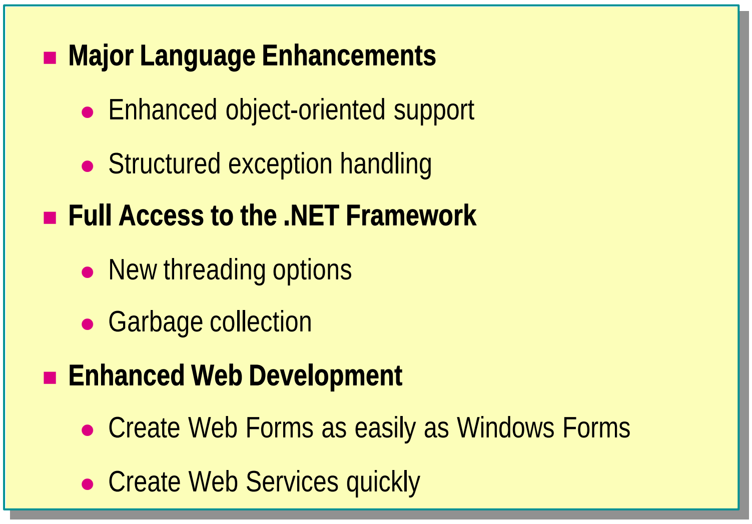
System.Drawing Namespace

The **System.Drawing** namespace provides access to GDI+ basic graphics functionality. More advanced functionality is provided in the **System.Drawing.Drawing2D**, **System.Drawing.Imaging**, and **System.Drawing.Text** namespaces.

What Are the Visual Basic .NET Enhancements?

Topic Objective
To introduce some of the major enhancements in Visual Basic .NET.

Lead-in
Visual Basic .NET provides major language enhancements, full access to the .NET Framework, and enhanced Web development.



- **Major Language Enhancements**
  - Enhanced object-oriented support
  - Structured exception handling
- **Full Access to the .NET Framework**
  - New threading options
  - Garbage collection
- **Enhanced Web Development**
  - Create Web Forms as easily as Windows Forms
  - Create Web Services quickly

Visual Basic .NET provides major language enhancements, full access to the .NET Framework, and enhanced Web development.

Delivery Tip
Discuss each point on the slide and the advantages each one gives developers.

Major Language Enhancements

In Visual Basic .NET, Visual Basic has undergone some significant changes to allow Visual Basic developers to create powerful, robust, and scalable enterprise applications.

- Enhanced object-oriented support

  Allows Visual Basic .NET developers to use class inheritance, constructors, destructors, overloading, interfaces, and polymorphism. This gives Visual Basic .NET as much object-oriented power as any other .NET language, such as C# or Visual C++ with managed extensions.

- Structured exception handling

  Simplifies exception handling, and allows you to use powerful features such as nested exceptions. This greatly improves the previous Visual Basic error handler.

Full Access to the .NET Framework

Visual Basic .NET developers have full access to the .NET Framework, including the entire .NET Framework Class Library.

- New threading options

  Allow you to create applications that use multithreaded capabilities. For the first time, Visual Basic developers will be able to create applications that are capable of rivaling Visual C++–based applications in this area.

- Garbage collection

  Ensures that applications created in Visual Basic .NET do not hold on to unreferenced memory. This background process addresses memory-related issues such as memory leaks and circular references that existed in previous versions of Visual Basic and other languages.

Enhanced Web Development

Visual Basic .NET enables developers to quickly create powerful Web applications.

- Create Web Forms easily

  You can do this as easily as Windows Forms by using the familiar technique of writing code behind the event of a control. No longer do developers need to use one integrated development environment (IDE) for Web page development (such as Microsoft Visual InterDev®) and another for component development. All aspects of a Web application are now simply created in a single environment in a way that will be easy for Visual Basic developers to understand.

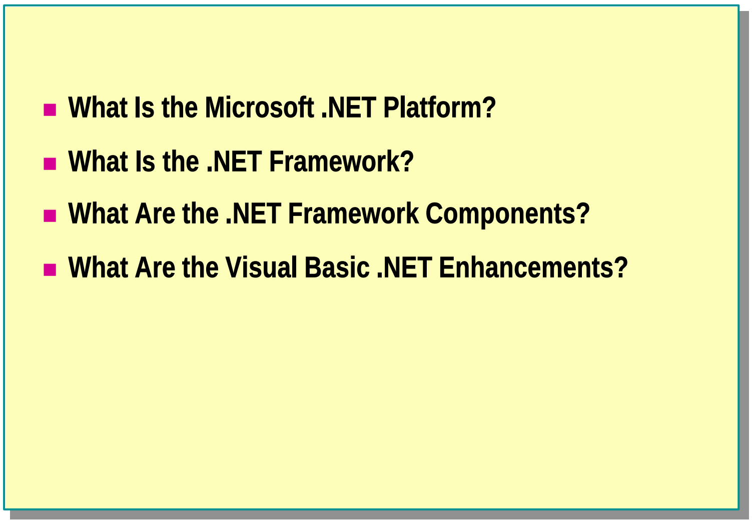
- Create Web Services quickly

  You can do this quickly in a fashion similar to creating components in previous versions of Visual Basic. Web Services is a powerful technology that allows you to access your components (or other people's components) across the Internet by means of HTTP. Any Visual Basic developer who has created components will easily make the transition to Web Services.

Review

Topic Objective
To reinforce module objectives by reviewing key points.

Lead-in
The review questions cover some of the key concepts taught in the module.



- What Is the Microsoft .NET Platform?
- What Is the .NET Framework?
- What Are the .NET Framework Components?
- What Are the Visual Basic .NET Enhancements?

1. What is the .NET Platform?

   The .NET Platform is a set of technologies designe d to transform the Internet into a full-scale distributed computing platform. It provides new ways to build applications from collections of Web Services. The .NET Platform fully supports the existing Internet infrastructure (HTTP, XML, SOAP).

2. What are the core technologies in the .NET Platform?

   .NET Framework, .NET Enterprise Servers, .NET Building Block services, Visual Studio .NET, and Windows.

3. List the components of the .NET Framework.

   Common language runtime, .NET Framework Class Library, data and XML, Web Services and Forms, and Windows Forms.

4. What is the purpose of common language runtime?

   It provides an environment in which you can execute code.

5. What is the purpose of common language specification?

   It defines a set of features that all .NET-compatible languages should support.

6. What is a Web Service?

   A Web Service is a programmable Web component that can be shared between applications on the Internet or an intranet.

7. What is a managed environment?

   A managed environment is one in which the environment provides services, such as garbage collection, security, and other similar features.