.NET Framework 4

**What is New in the .NET Framework 4**

Updated: March 2012

This topic contains information about key features and improvements in the .NET Framework version 4. This topic does not provide comprehensive information about all new features and is subject to change.

The .NET Framework 4 introduces an improved security model. For more information, see [Security Changes in the .NET Framework 4](http://msdn.microsoft.com/en-us/library/dd233103.aspx). For lists of new namespaces, new types, and new members added to existing types, see [New Types and Members in the .NET Framework 4](http://msdn.microsoft.com/en-us/library/ff641764.aspx). Other new features and improvements in the .NET Framework 4 are described in the following sections:

* [Application Compatibility and Deployment](http://msdn.microsoft.com/en-us/library/ms171868.aspx#application_compatibility_and_deployment)
* [Core New Features and Improvements](http://msdn.microsoft.com/en-us/library/ms171868.aspx#core_new_features_and_improvements)
* [Managed Extensibility Framework](http://msdn.microsoft.com/en-us/library/ms171868.aspx#managed_extensibility_framework)
* [Parallel Computing](http://msdn.microsoft.com/en-us/library/ms171868.aspx#parallel_computing)
* [Networking](http://msdn.microsoft.com/en-us/library/ms171868.aspx#networking)
* [Web](http://msdn.microsoft.com/en-us/library/ms171868.aspx#web)
* [Client](http://msdn.microsoft.com/en-us/library/ms171868.aspx#client)
* [Data](http://msdn.microsoft.com/en-us/library/ms171868.aspx#data)
* [Windows Communication Foundation](http://msdn.microsoft.com/en-us/library/ms171868.aspx#windows_communication_foundation)
* [Windows Workflow Foundation](http://msdn.microsoft.com/en-us/library/ms171868.aspx#windows_workflow_foundation)

The following optional updates to .NET Framework 4 are also available:

* [Visual Studio 2010 Service Pack 1 (SP1)](http://go.microsoft.com/fwlink/?LinkID=210826), which includes an [update](http://go.microsoft.com/fwlink/?LinkId=210824) to the .NET Framework 4.
* [Update 4.0.3 for Microsoft .NET Framework 4](http://msdn.microsoft.com/en-us/library/ms171868.aspx#updates) and earlier updates, which include several enhancements for the .NET Framework.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifApplication Compatibility and Deployment

The .NET Framework 4 is highly compatible with applications that are built with earlier .NET Framework versions, except for some changes that were made to improve security, standards compliance, correctness, reliability, and performance.

The .NET Framework 4 does not automatically use its version of the common language runtime to run applications that are built with earlier versions of the .NET Framework. To run older applications with .NET Framework 4, you must compile your application with the target .NET Framework version specified in the properties for your project in Visual Studio, or you can specify the supported runtime with the [<supportedRuntime> Element](http://msdn.microsoft.com/en-us/library/w4atty68.aspx) in an application configuration file.

If your application or component does not work after .NET Framework 4 is installed, please submit a bug on the [Microsoft Connect](http://go.microsoft.com/fwlink/?LinkId=154815) Web site. You can test compatibility as described in the [.NET Framework 4 Application Compatibility](http://go.microsoft.com/fwlink/?LinkId=154814) topic and learn about new features by using the [Visual Studio 2010 and .NET Framework 4 Walkthroughs](http://go.microsoft.com/fwlink/?LinkId=154809).

For guidance about migrating to the .NET Framework 4, see [Migration Guide to the .NET Framework 4](http://msdn.microsoft.com/en-us/library/ff657133.aspx) and [Version Compatibility in the .NET Framework](http://msdn.microsoft.com/en-us/library/ff602939.aspx).

The following sections describe deployment improvements.

**Client Profile**

The .NET Framework 4 Client Profile supports more platforms than in previous versions and provides a fast deployment experience for your applications. Several new project templates now target the Client Profile by default. For more information, see [.NET Framework Client Profile](http://msdn.microsoft.com/en-us/library/cc656912.aspx).

**In-Process Side-by-Side Execution**

This feature enables an application to load and start multiple versions of the .NET Framework in the same process. For example, you can run applications that load add-ins (or components) that are based on the .NET Framework 2.0 SP1 and add-ins that are based on the .NET Framework 4 in the same process. Older components continue to use the older .NET Framework version, and new components use the new .NET Framework version. For more information, see [In-Process Side-by-Side Execution](http://msdn.microsoft.com/en-us/library/ee518876.aspx).

**Portable Class Library**

When you install [Visual Studio 2010 Service Pack 1 (SP1)](http://go.microsoft.com/fwlink/?LinkID=210826) and the [Portable Library Tools](http://go.microsoft.com/fwlink/?LinkId=210823), you can create portable class libraries that run on a variety of .NET Framework platforms without recompiling. For more information, see [Portable Class Libraries](http://msdn.microsoft.com/en-us/library/gg597391.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifCore New Features and Improvements

The following sections describe new features and improvements provided by the common language runtime and the base class libraries.

**Diagnostics and Performance**

Earlier versions of the .NET Framework provided no way to determine whether a particular application domain was affecting other application domains, because the operating system APIs and tools, such as the Windows Task Manager, were precise only to the process level. Starting with the .NET Framework 4, you can get processor usage and memory usage estimates per application domain.

You can monitor CPU and memory usage of individual application domains. Application domain resource monitoring is available through the managed and native hosting APIs and event tracing for Windows (ETW). When this feature has been enabled, it collects statistics on all application domains in the process for the life of the process. See the new [AppDomain..::.MonitoringIsEnabled](http://msdn.microsoft.com/en-us/library/system.appdomain.monitoringisenabled.aspx) property.

You can now access the ETW events for diagnostic purposes to improve performance. For more information, see [CLR ETW Events](http://msdn.microsoft.com/en-us/library/dd264810.aspx) and [Controlling .NET Framework Logging](http://msdn.microsoft.com/en-us/library/dd264809.aspx). Also see [Performance Counters and In-Process Side-By-Side Applications](http://msdn.microsoft.com/en-us/library/dd537616.aspx).

The [System.Runtime.ExceptionServices..::.HandleProcessCorruptedStateExceptionsAttribute](http://msdn.microsoft.com/en-us/library/system.runtime.exceptionservices.handleprocesscorruptedstateexceptionsattribute.aspx) attribute enables managed code to handle exceptions that indicate corrupted process state.

**Globalization**

The .NET Framework 4 provides new neutral and specific cultures, updated property values, improvements in string handling, and other improvements. For more information, see [What's New in Globalization and Localization](http://msdn.microsoft.com/en-us/library/dd997383.aspx).

**Garbage Collection**

The .NET Framework 4 provides background garbage collection. This feature replaces concurrent garbage collection in previous versions and provides better performance. For more information, see [Fundamentals of Garbage Collection](http://msdn.microsoft.com/en-us/library/ee787088.aspx).

**Code Contracts**

Code contracts let you specify contractual information that is not represented by a method's or type's signature alone. The new [System.Diagnostics.Contracts](http://msdn.microsoft.com/en-us/library/system.diagnostics.contracts.aspx) namespace contains classes that provide a language-neutral way to express coding assumptions in the form of preconditions, postconditions, and object invariants. The contracts improve testing with run-time checking, enable static contract verification, and support documentation generation. For more information, see [Code Contracts](http://msdn.microsoft.com/en-us/library/dd264808.aspx).

**Design-Time-Only Interop Assemblies**

You no longer have to ship primary interop assemblies (PIAs) to deploy applications that interoperate with COM objects. In the .NET Framework 4, compilers can embed type information from interop assemblies, selecting only the types that an application (for example, an add-in) actually uses. Type safety is ensured by the common language runtime. See [Using COM Types in Managed Code](http://msdn.microsoft.com/en-us/library/3y76b69k.aspx) and [Walkthrough: Embedding Type Information from Microsoft Office Assemblies (C# and Visual Basic)](http://msdn.microsoft.com/en-us/library/ee317478.aspx).

**Dynamic Language Runtime**

The *dynamic language runtime* (DLR) is a new runtime environment that adds a set of services for dynamic languages to the CLR. The DLR makes it easier to develop dynamic languages to run on the .NET Framework and to add dynamic features to statically typed languages. To support the DLR, the new [System.Dynamic](http://msdn.microsoft.com/en-us/library/system.dynamic.aspx) namespace is added to the .NET Framework.

The expression trees are extended with new types that represent control flow, for example, [System.Linq.Expressions..::.LoopExpression](http://msdn.microsoft.com/en-us/library/system.linq.expressions.loopexpression.aspx) and [System.Linq.Expressions..::.TryExpression](http://msdn.microsoft.com/en-us/library/system.linq.expressions.tryexpression.aspx). These new types are used by the dynamic language runtime (DLR) and not used by LINQ.

In addition, several new classes that support the .NET Framework infrastructure are added to the [System.Runtime.CompilerServices](http://msdn.microsoft.com/en-us/library/system.runtime.compilerservices.aspx) namespace. For more information, see [Dynamic Language Runtime Overview](http://msdn.microsoft.com/en-us/library/dd233052.aspx).

**Covariance and Contravariance**

Several generic interfaces and delegates now support covariance and contravariance. For more information, see [Covariance and Contravariance in Generics](http://msdn.microsoft.com/en-us/library/dd799517.aspx).

**BigInteger and Complex Numbers**

The new [System.Numerics..::.BigInteger](http://msdn.microsoft.com/en-us/library/system.numerics.biginteger.aspx) structure is an arbitrary-precision integer data type that supports all the standard integer operations, including bit manipulation. It can be used from any .NET Framework language. In addition, some of the new .NET Framework languages (such as F# and IronPython) have built-in support for this structure.

The new [System.Numerics..::.Complex](http://msdn.microsoft.com/en-us/library/system.numerics.complex.aspx) structure represents a complex number that supports arithmetic and trigonometric operations with complex numbers.

**Tuples**

The .NET Framework 4 provides the [System..::.Tuple](http://msdn.microsoft.com/en-us/library/system.tuple.aspx) class for creating tuple objects that contain structured data. It also provides generic tuple classes to support tuples that have from one to eight components (that is, singletons through octuples). To support tuple objects that have nine or more components, there is a generic tuple class with seven type parameters and an eighth parameter of any tuple type.

**File System Enumeration Improvements**

New file enumeration methods improve the performance of applications that access large file directories or that iterate through the lines in large files. For more information, see [How to: Enumerate Directories and Files](http://msdn.microsoft.com/en-us/library/dd997370.aspx).

**Memory-Mapped Files**

The .NET Framework now supports [memory-mapped files](http://msdn.microsoft.com/en-us/library/dd997372.aspx). You can use memory-mapped files to edit very large files and to create shared memory for interprocess communication.

**64-Bit Operating Systems and Processes**

You can identify 64-bit operating systems and processes with the [Environment..::.Is64BitOperatingSystem](http://msdn.microsoft.com/en-us/library/system.environment.is64bitoperatingsystem.aspx) and [Environment..::.Is64BitProcess](http://msdn.microsoft.com/en-us/library/system.environment.is64bitprocess.aspx) properties.

You can specify a 32-bit or 64-bit view of the registry with the [Microsoft.Win32..::.RegistryView](http://msdn.microsoft.com/en-us/library/microsoft.win32.registryview.aspx) enumeration when you open base keys.

**Other New Features**

The following list describes additional new capabilities, improvements, and conveniences. Several of these are based on customer suggestions.

* To support culture-sensitive formatting, the [System..::.TimeSpan](http://msdn.microsoft.com/en-us/library/system.timespan.aspx) structure includes new overloads of the **ToString**, **Parse**, and **TryParse** methods, as well as new **ParseExact** and **TryParseExact** methods.
* The new [String..::.IsNullOrWhiteSpace](http://msdn.microsoft.com/en-us/library/system.string.isnullorwhitespace.aspx) method indicates whether a string is null, empty, or consists only of white-space characters. New overloads have been added to the **String.Concat** and **String.Join** methods that concatenate members of [System.Collections.Generic..::.IEnumerable<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/9eekhta0.aspx) collections.
* The [String..::.Concat](http://msdn.microsoft.com/en-us/library/system.string.concat.aspx) method lets you concatenate each element in an enumerable collection without first converting the elements to strings.
* Two new convenience methods are available: [StringBuilder..::.Clear](http://msdn.microsoft.com/en-us/library/system.text.stringbuilder.clear.aspx) and [Stopwatch..::.Restart](http://msdn.microsoft.com/en-us/library/system.diagnostics.stopwatch.restart.aspx).
* The new [Enum..::.HasFlag](http://msdn.microsoft.com/en-us/library/system.enum.hasflag.aspx) method determines whether one or more bit fields or flags are set in an enumeration value. The [Enum..::.TryParse](http://msdn.microsoft.com/en-us/library/system.enum.tryparse.aspx) method returns a Boolean value that indicates whether a string or integer value could be successfully parsed.
* The [System..::.Environment..::.SpecialFolder](http://msdn.microsoft.com/en-us/library/system.environment.specialfolder.aspx) enumeration contains several new folders.
* You can now easily copy one stream into another with the [CopyTo](http://msdn.microsoft.com/en-us/library/system.io.stream.copyto.aspx) method in classes that inherit from the [System.IO..::.Stream](http://msdn.microsoft.com/en-us/library/system.io.stream.aspx) class.
* New [Path..::.Combine](http://msdn.microsoft.com/en-us/library/system.io.path.combine.aspx) method overloads enable you to combine file paths.
* The new [System..::.IObservable<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd990377.aspx) and [System..::.IObserver<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd783449.aspx) interfaces provide a generalized mechanism for push-based notifications.
* The [System..::.IntPtr](http://msdn.microsoft.com/en-us/library/system.intptr.aspx) and [System..::.UIntPtr](http://msdn.microsoft.com/en-us/library/system.uintptr.aspx) classes now include support for the addition and subtraction operators.
* You can now enable lazy initialization for any custom type by wrapping the type inside a [System..::.Lazy<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd642331.aspx) class.
* The new [System.Collections.Generic..::.SortedSet<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd412070.aspx) class provides a self-balancing tree that maintains data in sorted order after insertions, deletions, and searches. This class implements the new [System.Collections.Generic..::.ISet<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd412081.aspx) interface.
* The compression algorithms for the [System.IO.Compression..::.DeflateStream](http://msdn.microsoft.com/en-us/library/system.io.compression.deflatestream.aspx) and [System.IO.Compression..::.GZipStream](http://msdn.microsoft.com/en-us/library/system.io.compression.gzipstream.aspx) classes have improved so that data that is already compressed is no longer inflated. Also, the 4-gigabyte size restriction for compressing streams has been removed.
* The new [Monitor..::.Enter(Object, Boolean%)](http://msdn.microsoft.com/en-us/library/dd289498.aspx) method overload takes a Boolean reference and atomically sets it to **true** only if the monitor is successfully entered.
* You can use the [Thread..::.Yield](http://msdn.microsoft.com/en-us/library/system.threading.thread.yield.aspx) method to have the calling thread yield execution to another thread that is ready to run on the current processor.
* The [System..::.Guid](http://msdn.microsoft.com/en-us/library/system.guid.aspx) structure now contains the [TryParse](http://msdn.microsoft.com/en-us/library/system.guid.tryparse.aspx) and [TryParseExact](http://msdn.microsoft.com/en-us/library/system.guid.tryparseexact.aspx) methods.
* The new [Microsoft.Win32..::.RegistryOptions](http://msdn.microsoft.com/en-us/library/microsoft.win32.registryoptions.aspx) enumeration lets you specify a volatile registry key that does not persist after the computer restarts.
* Registry keys are no longer restricted to a maximum length of 255 characters.

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifManaged Extensibility Framework

The Managed Extensibility Framework (MEF) is a new library in the .NET Framework 4 that helps you build extensible and composable applications. MEF enables you to specify points where an application can be extended, to expose services to offer to other extensible applications and to create parts for consumption by extensible applications. It also enables easy discoverability of available parts based on metadata, without the need to load the assemblies for the parts. For more information, see [Managed Extensibility Framework Overview](http://msdn.microsoft.com/en-us/library/dd460648.aspx) and [Managed Extensibility Framework](http://go.microsoft.com/fwlink/?LinkId=144282). For a list of the MEF types, see the [System.ComponentModel.Composition](http://msdn.microsoft.com/en-us/library/system.componentmodel.composition.aspx) namespace.

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifParallel Computing

The .NET Framework 4 introduces a new programming model for writing multithreaded and asynchronous code that greatly simplifies the work of application and library developers. The new model enables developers to write efficient, fine-grained, and scalable parallel code in a natural idiom without having to work directly with threads or the thread pool. The new [System.Threading.Tasks](http://msdn.microsoft.com/en-us/library/system.threading.tasks.aspx) namespace and other related types support this new model. Parallel LINQ (PLINQ), which is a parallel implementation of LINQ to Objects, enables similar functionality through declarative syntax. For more information, see [Parallel Programming in the .NET Framework](http://msdn.microsoft.com/en-us/library/dd460693.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifNetworking

Networking improvements include the following:

* Security improvements for Windows authentication in several classes, including [System.Net..::.HttpWebRequest](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.aspx), [System.Net..::.HttpListener](http://msdn.microsoft.com/en-us/library/system.net.httplistener.aspx), [System.Net.Mail..::.SmtpClient](http://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient.aspx), [System.Net.Security..::.SslStream](http://msdn.microsoft.com/en-us/library/system.net.security.sslstream.aspx), and [System.Net.Security..::.NegotiateStream](http://msdn.microsoft.com/en-us/library/system.net.security.negotiatestream.aspx). Extended protection is available for applications on Windows 7 and Windows Server 2008 R2. For more information, see [Integrated Windows Authentication with Extended Protection](http://msdn.microsoft.com/en-us/library/dd582691.aspx).
* Support for Network Address Translation (NAT) traversal using IPv6 and Teredo. For more information, see [NAT Traversal using IPv6 and Teredo](http://msdn.microsoft.com/en-us/library/ee663252.aspx).
* New networking performance counters that provide information about [HttpWebRequest](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.aspx) objects. For more information, see [Networking Performance Counters](http://msdn.microsoft.com/en-us/library/70xadeyt.aspx).
* In the [System.Net..::.HttpWebRequest](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.aspx) class, support for using large byte range headers (64-bit ranges) with new overloads for the [AddRange](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.addrange.aspx) method. New properties on the [System.Net..::.HttpWebRequest](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.aspx) class allow an application to set many HTTP headers. You can use the [Host](http://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.host.aspx) property to set the Host header value in an HTTP request that is independent from the request URI.
* Secure Sockets Layer (SSL) support for the [System.Net.Mail..::.SmtpClient](http://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient.aspx) and related classes.
* Improved support for mail headers in the [System.Net.Mail..::.MailMessage](http://msdn.microsoft.com/en-us/library/system.net.mail.mailmessage.aspx) class.
* Support for a null cipher for use in encryption. You can specify the encryption policy by using the [System.Net..::.ServicePointManager](http://msdn.microsoft.com/en-us/library/system.net.servicepointmanager.aspx) class and the [EncryptionPolicy](http://msdn.microsoft.com/en-us/library/system.net.servicepointmanager.encryptionpolicy.aspx) property. Constructors for the [System.Net.Security..::.SslStream](http://msdn.microsoft.com/en-us/library/system.net.security.sslstream.aspx) class now take a [System.Net.Security..::.EncryptionPolicy](http://msdn.microsoft.com/en-us/library/system.net.security.encryptionpolicy.aspx) class as a parameter.
* Credentials for password-based authentication schemes such as basic, digest, NTLM, and Kerberos authentication in the [System.Net..::.NetworkCredential](http://msdn.microsoft.com/en-us/library/system.net.networkcredential.aspx) class. To improve security, passwords may now be treated as [System.Security..::.SecureString](http://msdn.microsoft.com/en-us/library/system.security.securestring.aspx) instances rather than [System..::.String](http://msdn.microsoft.com/en-us/library/system.string.aspx) instances.
* Ability to specify how a URI with percent-encoded values is converted and normalized in the [System..::.Uri](http://msdn.microsoft.com/en-us/library/system.uri.aspx) and [System.Net..::.HttpListener](http://msdn.microsoft.com/en-us/library/system.net.httplistener.aspx) classes. For more information, see the [System.Net.Configuration..::.HttpListenerElement](http://msdn.microsoft.com/en-us/library/system.net.configuration.httplistenerelement.aspx), [System.Configuration..::.SchemeSettingElement](http://msdn.microsoft.com/en-us/library/system.configuration.schemesettingelement.aspx), [System.Configuration..::.SchemeSettingElementCollection](http://msdn.microsoft.com/en-us/library/system.configuration.schemesettingelementcollection.aspx), and [System.Configuration..::.UriSection](http://msdn.microsoft.com/en-us/library/system.configuration.urisection.aspx) classes.

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifWeb

ASP.NET version 4 introduces new features in the following areas:

* Core services, including a new API that lets you extend caching, support for compression for session-state data, and a new application preload manager (autostart feature).
* Web Forms, including more integrated support for ASP.NET routing, enhanced support for Web standards, updated browser support, new features for data controls, and new features for view state management.
* Web Forms controls, including a new Chart control.
* MVC, including new helper methods for views, support for partitioned MVC applications, and asynchronous controllers.
* Dynamic Data, including support for existing Web applications, support for many-to-many relationships and inheritance, new field templates and attributes, and enhanced data filtering.
* Microsoft Ajax, including additional support for client-based Ajax applications in the Microsoft Ajax Library.
* Visual Web Developer, including improved IntelliSense for JScript, new auto-complete snippets for HTML and ASP.NET markup, and enhanced CSS compatibility.
* Deployment, including new tools for automating typical deployment tasks.
* Multi-targeting, including better filtering for features that are not available in the target version of the .NET Framework.

For more information about these features, see [What's New in ASP.NET 4 and Visual Web Developer](http://msdn.microsoft.com/en-us/library/s57a598e.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifClient

Windows Presentation Foundation (WPF) version 4 contains changes and improvements in the following areas:

* New controls, including [Calendar](http://msdn.microsoft.com/en-us/library/system.windows.controls.calendar.aspx), [DataGrid](http://msdn.microsoft.com/en-us/library/system.windows.controls.datagrid.aspx), and [DatePicker](http://msdn.microsoft.com/en-us/library/system.windows.controls.datepicker.aspx).
* [VisualStateManager](http://msdn.microsoft.com/en-us/library/system.windows.visualstatemanager.aspx) supports changing states of controls.
* Touch and Manipulation enables you to create applications that receive input from multiple touches simultaneously on Windows 7.
* Graphics and animation supports layout rounding, Pixel Shader version 3.0, cached composition, and easing functions.
* Text has improved text rendering and supports customizing the caret color and selection color in text boxes.
* Binding is supported on the [Command](http://msdn.microsoft.com/en-us/library/system.windows.input.inputbinding.command.aspx) property of an [InputBinding](http://msdn.microsoft.com/en-us/library/system.windows.input.inputbinding.aspx), dynamic objects, and the [Text](http://msdn.microsoft.com/en-us/library/system.windows.documents.run.text.aspx) property.
* XAML browser applications (XBAPs) support communication with the Web page and support full-trust deployment.
* New types in the [System.Windows.Shell](http://msdn.microsoft.com/en-us/library/system.windows.shell.aspx) namespace enable you to communicate with the Windows 7 taskbar and pass data to the Windows shell.
* The WPF and Silverlight Designer in Visual Studio 2010 has various designer improvements to help create WPF or Silverlight applications.

For more information, see [What's New in WPF Version 4](http://msdn.microsoft.com/en-us/library/bb613588.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifData

**ADO.NET**

ADO.NET provides new features for the Entity Framework, including persistence-ignorant objects, functions in LINQ queries, and customized object layer code generation. For more information, see [What's New in ADO.NET](http://msdn.microsoft.com/en-us/library/ex6y04yf.aspx).

**Dynamic Data**

For ASP.NET 4, Dynamic Data has been enhanced to give you even more power for quickly building data-driven Web sites. This includes the following:

* Automatic validation that is based on constraints that are defined in the data model.
* The ability to easily change the markup that is generated for fields in the [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) and [DetailsView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.detailsview.aspx) controls by using field templates that are part of a Dynamic Data project.

For more information, see [What's New in ASP.NET 4 and Visual Web Developer](http://msdn.microsoft.com/en-us/library/s57a598e.aspx).

**WCF Data Services**

ADO.NET Data Service has been renamed to WCF Data Services, and has the following new features

* Data binding.
* Counting entities in an entity set.
* Server-driven paging.
* Query projections.
* Custom data service providers.
* Streaming of binary resources.

For more information, see [What's New in WCF Data Services](http://msdn.microsoft.com/en-us/library/ee373845.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifWindows Communication Foundation

Windows Communication Foundation (WCF) provides the following improvements:

* Configuration-based activation: Removes the requirement for having an .svc file.
* System.Web.Routing integration: Gives you more control over your service's URL by allowing the use of extensionless URLs.
* Multiple IIS site bindings support: Allows you to have multiple base addresses with the same protocol on the same Web site.
* Routing Service: Allows you to route messages based on content.
* Support for WS-Discovery: Allows you to create and search for discoverable services.
* Standard endpoints: Predefined endpoints that allow you to specify only certain properties.
* Workflow services: Integrates WCF and WF by providing activities to send and receive messages, the ability to correlate messages based on content, and a workflow service host.
* WCF REST features:
  + Web HTTP caching: Allows caching of Web HTTP service responses.
  + Web HTTP formats support: Allows you to dynamically determine the best format for a service operation to respond in.
  + Web HTTP services help page: Provides an automatic help page for Web HTTP services, similar to the WCF service help page.
  + Web HTTP error handling: Allows Web HTTP Services to return error information in the same format as the operation.
  + Web HTTP cross-domain JavaScript support: Allows use of JSON Padding (JSONP).
* Simplified configuration: Reduces the amount of configuration a service requires

For more information, see [What's New in Windows Communication Foundation](http://msdn.microsoft.com/en-us/library/dd456789.aspx).

[Back to top](http://msdn.microsoft.com/en-us/library/ms171868.aspx#introduction)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifWindows Workflow Foundation

Windows Workflow Foundation (WF) provides improvements in the following areas:

* Improved workflow activity model: The [Activity](http://msdn.microsoft.com/en-us/library/system.activities.activity.aspx) class provides the base abstraction of workflow behavior.
* Rich composite activity options: Workflows benefit from new flow-control activities that model traditional flow-control structures, such as [Flowchart](http://msdn.microsoft.com/en-us/library/system.activities.statements.flowchart.aspx), [TryCatch](http://msdn.microsoft.com/en-us/library/system.activities.statements.trycatch.aspx), and [Switch<(Of <(T>)>)](http://msdn.microsoft.com/en-us/library/dd647669.aspx).
* Expanded built-in activity library: New features of the activity library include new flow-control activities, activities for manipulating member data, and activities for controlling transactions.
* Explicit activity data model: New options for storing or moving data include variable and directional arguments.
* Enhanced hosting, persistence, and tracking options: Hosting enhancements include more options for running workflows, explicit persistence using the [Persist](http://msdn.microsoft.com/en-us/library/system.activities.statements.persist.aspx)activity, persisting without unloading, preventing persistence by using no-persist zones, using ambient transactions from the host, recording tracking information to the event log, and resuming pending workflows by using a [Bookmark](http://msdn.microsoft.com/en-us/library/system.activities.bookmark.aspx) object.
* Easier ability to extend the WF Designer: The new WF Designer is built on Windows Presentation Foundation (WPF) and provides an easier model to use when rehosting the WF Designer outside of Visual Studio.