**What's New in the .NET Framework 4.5**

**.NET Framework 4.5**

[Other Versions](javascript:;)

http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [.NET Framework 4](http://msdn.microsoft.com/en-us/library/ms171868(d=printer,v=vs.100).aspx)
* [.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/ms171868(d=printer,v=vs.85).aspx)
* [.NET Framework 3.5](http://msdn.microsoft.com/en-us/library/ms171868(d=printer,v=vs.90).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ms171868(d=printer,v=vs.80).aspx)

This article contains a summary of key new features and improvements in the .NET Framework 4.5 and 4.5.1:

[What's New in the .NET Framework 4.5.1](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#v451)   
[.NET Framework 4.5 – Core New Features and Improvements](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#core)   
[Tools](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#tools)   
[Parallel Computing](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#parallel)   
[Web](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#web)   
[Networking](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#networking)   
[Windows Presentation Foundation (WPF)](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#client)   
[Windows Communication Foundation (WCF)](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#windows_communication_foundation)   
[Windows Workflow Foundation (WF)](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#windows_workflow_foundation)   
[.NET for Windows Store Apps](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#tailored)   
[Portable Class Libraries](http://msdn.microsoft.com/en-us/library/ms171868(d=printer).aspx#portable)

This article does not provide comprehensive information about each new feature and is subject to change. For general information about the .NET Framework, see [Getting Started with the .NET Framework](http://msdn.microsoft.com/en-us/library/hh425099.aspx). For supported platforms, see [System Requirements](http://msdn.microsoft.com/en-us/library/8z6watww.aspx). For download links and installation instructions, see [Installing the .NET Framework 4.5, 4.5.1](http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx).

[What's New in the .NET Framework 4.5.1](javascript:void(0))

For a list of new APIs, see [New Types and Members in the .NET Framework 4.5.1](http://msdn.microsoft.com/en-us/library/dn527271.aspx).

Core new features and enhancements include:

* Automatic binding redirection for assemblies. Starting with Visual Studio 2013, when you compile an app that targets the .NET Framework 4.5.1, binding redirects may be added to the app configuration file if your app or its components reference multiple versions of the same assembly. You can also enable this feature for projects that target older versions of the .NET Framework. For more information, see [How to: Enable and Disable Automatic Binding Redirection](http://msdn.microsoft.com/en-us/library/2fc472t2.aspx).
* Ability to collect diagnostics information to help developers improve the performance of server and cloud applications. For more information, see the [WriteEventWithRelatedActivityId](http://msdn.microsoft.com/en-us/library/system.diagnostics.tracing.eventsource.writeeventwithrelatedactivityid.aspx) and [WriteEventWithRelatedActivityIdCore](http://msdn.microsoft.com/en-us/library/system.diagnostics.tracing.eventsource.writeeventwithrelatedactivityidcore.aspx) methods in the [EventSource](http://msdn.microsoft.com/en-us/library/system.diagnostics.tracing.eventsource.aspx) class.
* Ability to explicitly compact the large object heap (LOH) during garbage collection. For more information, see the [GCSettings.LargeObjectHeapCompactionMode](http://msdn.microsoft.com/en-us/library/system.runtime.gcsettings.largeobjectheapcompactionmode.aspx) property.
* Additional performance improvements such as ASP.NET app suspension, multi-core JIT improvements, and faster app startup after a .NET Framework update. For details, see the [.NET Framework 4.5.1 announcement](http://blogs.msdn.com/b/dotnet/archive/2013/06/26/announcing-the-net-framework-4-5-1-preview.aspx) and the [ASP.NET app suspend](http://blogs.msdn.com/b/dotnet/archive/2013/10/09/asp-net-app-suspend-responsive-shared-net-web-hosting.aspx) blog post.

Improvements when debugging your .NET Framework apps in Visual Studio 2013 include:

* Return values in the Visual Studio debugger. When you debug a managed app in Visual Studio 2013, the Autos window displays return types and values for methods. This information is available for desktop, Windows Store, and Windows Phone apps. For more information, see [Examine return values of method calls](http://msdn.microsoft.com/library/e3245b37-8e2e-4200-ba84-133726e95f1f(v=vs.120).aspx) in the MSDN Library.
* Edit and Continue for 64-bit apps. Visual Studio 2013 supports the Edit and Continue feature for 64-bit managed apps for desktop, Windows Store, and Windows Phone. The existing limitations remain in effect for both 32-bit and 64-bit apps (see the last section of the [Supported Code Changes (C#)](http://msdn.microsoft.com/library/ms164927(v=vs.120).aspx) article).
* Async-aware debugging. To make it easier to debug asynchronous apps in Visual Studio 2013, the call stack hides the infrastructure code provided by compilers to support asynchronous programming, and also chains in logical parent frames so you can follow logical program execution more clearly. A Tasks window replaces the Parallel Tasks window and displays tasks that relate to a particular breakpoint, and also displays any other tasks that are currently active or scheduled in the app. You can read about this feature in the "Async-aware debugging" section of the [.NET Framework 4.5.1 announcement](http://blogs.msdn.com/b/dotnet/archive/2013/06/26/announcing-the-net-framework-4-5-1-preview.aspx).
* Better exception support for Windows Runtime components. In Windows 8.1, exceptions that arise from Windows Store apps preserve information about the error that caused the exception, even across language boundaries. You can read about this feature in the "Windows Store app development" section of the [.NET Framework 4.5.1 announcement](http://blogs.msdn.com/b/dotnet/archive/2013/06/26/announcing-the-net-framework-4-5-1-preview.aspx).

Starting with Visual Studio 2013, you can use the [Managed Profile Guided Optimization Tool (Mpgo.exe)](http://msdn.microsoft.com/en-us/library/hh873180.aspx) to optimize Windows Store apps as well as desktop apps.

For new features in ASP.NET 4.5.1, see [ASP.NET 4.5.1 and Visual Studio 2013](http://go.microsoft.com/fwlink/?LinkID=309094) on the ASP.NET site.

Note that the .NET Framework team has started continuous delivery of features such as [immutable collections](http://msdn.microsoft.com/en-us/library/dn385366.aspx) with NuGet. Features are released out of band (OOB) to expand platform support and to introduce new functionality. For more information, see [The .NET Framework and Out-of-Band Releases](http://msdn.microsoft.com/en-us/library/dn151288.aspx).

[.NET Framework 4.5 – Core New Features and Improvements](javascript:void(0))

* Ability to reduce system restarts by detecting and closing .NET Framework 4 applications during deployment. See [Reducing System Restarts During .NET Framework 4.5 Installations](http://msdn.microsoft.com/en-us/library/hh527997.aspx).
* Support for arrays that are larger than 2 gigabytes (GB) on 64-bit platforms. This feature can be enabled in the application configuration file. See the [<gcAllowVeryLargeObjects> element](http://msdn.microsoft.com/en-us/library/hh285054.aspx), which also lists other restrictions on object size and array size.
* Better performance through background garbage collection for servers. When you use server garbage collection in the .NET Framework 4.5, background garbage collection is automatically enabled. See the Background Server Garbage Collection section of the [Fundamentals of Garbage Collection](http://msdn.microsoft.com/en-us/library/ee787088.aspx) topic.
* Background just-in-time (JIT) compilation, which is optionally available on multi-core processors to improve application performance. See [ProfileOptimization](http://msdn.microsoft.com/en-us/library/system.runtime.profileoptimization.aspx).
* Ability to limit how long the regular expression engine will attempt to resolve a regular expression before it times out. See the [Regex.MatchTimeout](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.matchtimeout.aspx) property.
* Ability to define the default culture for an application domain. See the [CultureInfo](http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo.aspx) class.
* Console support for Unicode (UTF-16) encoding. See the [Console](http://msdn.microsoft.com/en-us/library/system.console.aspx) class.
* Support for versioning of cultural string ordering and comparison data. See the [SortVersion](http://msdn.microsoft.com/en-us/library/system.globalization.sortversion.aspx) class.
* Better performance when retrieving resources. See [Packaging and Deploying Resources in Desktop Apps](http://msdn.microsoft.com/en-us/library/sb6a8618.aspx).
* Zip compression improvements to reduce the size of a compressed file. See the [System.IO.Compression](http://msdn.microsoft.com/en-us/library/system.io.compression.aspx) namespace.
* Ability to customize a reflection context to override default reflection behavior through the [CustomReflectionContext](http://msdn.microsoft.com/en-us/library/system.reflection.context.customreflectioncontext.aspx) class.
* Support for the 2008 version of the Internationalized Domain Names in Applications (IDNA) standard when the [System.Globalization.IdnMapping](http://msdn.microsoft.com/en-us/library/system.globalization.idnmapping.aspx) class is used on Windows 8.
* Delegation of string comparison to the operating system, which implements Unicode 6.0, when the .NET Framework is used on Windows 8. When running on other platforms, the .NET Framework includes its own string comparison data, which implements Unicode 5.x. See the [String](http://msdn.microsoft.com/en-us/library/system.string.aspx) class and the Remarks section of the [SortVersion](http://msdn.microsoft.com/en-us/library/system.globalization.sortversion.aspx) class.
* Ability to compute the hash codes for strings on a per application domain basis. See [<UseRandomizedStringHashAlgorithm> Element](http://msdn.microsoft.com/en-us/library/jj152924.aspx).
* Type reflection support split between [Type](http://msdn.microsoft.com/en-us/library/system.type.aspx) and [TypeInfo](http://msdn.microsoft.com/en-us/library/system.reflection.typeinfo.aspx) classes. See [Reflection in the .NET Framework for Windows Store Apps](http://msdn.microsoft.com/en-us/library/hh535795.aspx).

[Managed Extensibility Framework (MEF)](javascript:void(0))

In the .NET Framework 4.5, the Managed Extensibility Framework (MEF) provides the following new features:

* Support for generic types.
* Convention-based programming model that enables you to create parts based on naming conventions rather than attributes.
* Multiple scopes.
* A subset of MEF that you can use when you create Windows Store apps. This subset is available as a [downloadable package](http://go.microsoft.com/fwlink/?LinkId=256238) from the NuGet Gallery. To install the package, open your project in Visual Studio, choose Manage NuGet Packages from the Project menu, and search online for the Microsoft.Composition package.

For more information, see [Managed Extensibility Framework (MEF)](http://msdn.microsoft.com/en-us/library/dd460648.aspx).

[Asynchronous File Operations](javascript:void(0))

In the .NET Framework 4.5, new asynchronous features were added to the C# and Visual Basic languages. These features add a task-based model for performing asynchronous operations. To use this new model, use the asynchronous methods in the I/O classes. See [Asynchronous File I/O](http://msdn.microsoft.com/en-us/library/kztecsys.aspx).

[Tools](javascript:void(0))

In the .NET Framework 4.5, Resource File Generator (Resgen.exe) enables you to create a .resw file for use in Windows Store apps from a .resources file embedded in a .NET Framework assembly. For more information, see [Resgen.exe (Resource File Generator)](http://msdn.microsoft.com/en-us/library/ccec7sz1.aspx).

Managed Profile Guided Optimization (Mpgo.exe) enables you to improve application startup time, memory utilization (working set size), and throughput by optimizing native image assemblies. The command-line tool generates profile data for native image application assemblies. See [Mpgo.exe (Managed Profile Guided Optimization Tool)](http://msdn.microsoft.com/en-us/library/hh873180.aspx). Starting with Visual Studio 2013, you can use Mpgo.exe to optimize Windows Store apps as well as desktop apps.

[Parallel Computing](javascript:void(0))

The .NET Framework 4.5 provides several new features and improvements for parallel computing. These include improved performance, increased control, improved support for asynchronous programming, a new dataflow library, and improved support for parallel debugging and performance analysis. See the entry [What’s New for Parallelism in .NET 4.5](http://go.microsoft.com/fwlink/?LinkId=235061) in the Parallel Programming with .NET blog.

[Web](javascript:void(0))

ASP.NET 4.5 and 4.5.1 add model binding for Web Forms, WebSocket support, asynchronous handlers, performance enhancements, and many other features. For more information, see the following resources:

* [ASP.NET 4.5 and Visual Studio 2012](http://msdn.microsoft.com/en-us/library/hh420390.aspx) in the MSDN Library.
* [ASP.NET 4.5.1 and Visual Studio 2013](http://go.microsoft.com/fwlink/?LinkID=309094) on the ASP.NET site.

[Networking](javascript:void(0))

The .NET Framework 4.5 provides a new programming interface for HTTP applications. For more information, see the new [System.Net.Http](http://msdn.microsoft.com/en-us/library/system.net.http.aspx) and [System.Net.Http.Headers](http://msdn.microsoft.com/en-us/library/system.net.http.headers.aspx) namespaces.

Support is also included for a new programming interface for accepting and interacting with a WebSocket connection by using the existing [HttpListener](http://msdn.microsoft.com/en-us/library/system.net.httplistener.aspx) and related classes. For more information, see the new [System.Net.WebSockets](http://msdn.microsoft.com/en-us/library/system.net.websockets.aspx) namespace and the [HttpListener](http://msdn.microsoft.com/en-us/library/system.net.httplistener.aspx) class.

In addition, the .NET Framework 4.5 includes the following networking improvements:

* RFC-compliant URI support. For more information, see [Uri](http://msdn.microsoft.com/en-us/library/system.uri.aspx) and related classes.
* Support for Internationalized Domain Name (IDN) parsing. For more information, see [Uri](http://msdn.microsoft.com/en-us/library/system.uri.aspx) and related classes.
* Support for Email Address Internationalization (EAI). For more information, see the [System.Net.Mail](http://msdn.microsoft.com/en-us/library/system.net.mail.aspx) namespace.
* Improved IPv6 support. For more information, see the [System.Net.NetworkInformation](http://msdn.microsoft.com/en-us/library/system.net.networkinformation.aspx) namespace.
* Dual-mode socket support. For more information, see the [Socket](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.aspx) and [TcpListener](http://msdn.microsoft.com/en-us/library/system.net.sockets.tcplistener.aspx) classes.

[Windows Presentation Foundation (WPF)](javascript:void(0))

In the .NET Framework 4.5, Windows Presentation Foundation (WPF) contains changes and improvements in the following areas:

* The new [Ribbon](http://msdn.microsoft.com/en-us/library/system.windows.controls.ribbon.ribbon.aspx) control, which enables you to implement a ribbon user interface that hosts a Quick Access Toolbar, Application Menu, and tabs.
* The new [INotifyDataErrorInfo](http://msdn.microsoft.com/en-us/library/system.componentmodel.inotifydataerrorinfo.aspx) interface, which supports synchronous and asynchronous data validation.
* New features for the [VirtualizingPanel](http://msdn.microsoft.com/en-us/library/system.windows.controls.virtualizingpanel.aspx) and [Dispatcher](http://msdn.microsoft.com/en-us/library/system.windows.threading.dispatcher.aspx) classes.
* Improved performance when displaying large sets of grouped data, and by accessing collections on non-UI threads.
* Data binding to static properties, data binding to custom types that implement the [ICustomTypeProvider](http://msdn.microsoft.com/en-us/library/system.reflection.icustomtypeprovider.aspx) interface, and retrieval of data binding information from a binding expression.
* Repositioning of data as the values change (live shaping).
* Ability to check whether the data context for an item container is disconnected.
* Ability to set the amount of time that should elapse between property changes and data source updates.
* Improved support for implementing weak event patterns. Also, events can now accept markup extensions.

For more information, see [What's New in WPF Version 4.5](http://msdn.microsoft.com/en-us/library/bb613588.aspx).

[Windows Communication Foundation (WCF)](javascript:void(0))

In the .NET Framework 4.5, the following features have been added to make it simpler to write and maintain Windows Communication Foundation (WCF) applications:

* Simplification of generated configuration files.
* Support for contract-first development.
* Ability to configure ASP.NET compatibility mode more easily.
* Changes in default transport property values to reduce the likelihood that you will have to set them.
* Updates to the [XmlDictionaryReaderQuotas](http://msdn.microsoft.com/en-us/library/system.xml.xmldictionaryreaderquotas.aspx) class to reduce the likelihood that you will have to manually configure quotas for XML dictionary readers.
* Validation of WCF configuration files by Visual Studio as part of the build process, so you can detect configuration errors before you run your application.
* New asynchronous streaming support.
* New HTTPS protocol mapping to make it easier to expose an endpoint over HTTPS with Internet Information Services (IIS).
* Ability to generate metadata in a single WSDL document by appending ?singleWSDL to the service URL.
* Websockets support to enable true bidirectional communication over ports 80 and 443 with performance characteristics similar to the TCP transport.
* Support for configuring services in code.
* XML Editor tooltips.
* [ChannelFactory](http://msdn.microsoft.com/en-us/library/system.servicemodel.channelfactory.aspx) caching support.
* Binary encoder compression support.
* Support for a UDP transport that enables developers to write services that use "fire and forget" messaging. A client sends a message to a service and expects no response from the service.
* Ability to support multiple authentication modes on a single WCF endpoint when using the HTTP transport and transport security.
* Support for WCF services that use internationalized domain names (IDNs).

For more information, see [What's New in Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkId=228173).

[Windows Workflow Foundation (WF)](javascript:void(0))

In the .NET Framework 4.5, several new features were added to Windows Workflow Foundation (WF), including:

* State machine workflows, which were first introduced as part of the .NET Framework 4.0.1 ([.NET Framework 4 Platform Update 1](http://go.microsoft.com/fwlink/?LinkID=215092)). This update included several new classes and activities that enabled developers to create state machine workflows. These classes and activities were updated for the .NET Framework 4.5 to include:
  + The ability to set breakpoints on states.
  + The ability to copy and paste transitions in the workflow designer.
  + Designer support for shared trigger transition creation.
  + Activities for creating state machine workflows, including: [StateMachine](http://msdn.microsoft.com/en-us/library/system.activities.statements.statemachine.aspx), [State](http://msdn.microsoft.com/en-us/library/system.activities.statements.state.aspx), and [Transition](http://msdn.microsoft.com/en-us/library/system.activities.statements.transition.aspx).
* Enhanced Workflow Designer features such as the following:
  + Enhanced workflow search capabilities in Visual Studio, including Quick Find and Find in Files.
  + Ability to automatically create a Sequence activity when a second child activity is added to a container activity, and to include both activities in the Sequence activity.
  + Panning support, which enables the visible portion of a workflow to be changed without using the scroll bars.
  + A new Document Outline view that shows the components of a workflow in a tree-style outline view and lets you select a component in the Document Outline view.
  + Ability to add annotations to activities.
  + Ability to define and consume activity delegates by using the workflow designer.
  + Auto-connect and auto-insert for activities and transitions in state machine and flowchart workflows.
* Storage of the view state information for a workflow in a single element in the XAML file, so you can easily locate and edit the view state information.
* A NoPersistScope container activity to prevent child activities from persisting.
* Support for C# expressions:
  + Workflow projects that use Visual Basic will use Visual Basic expressions, and C# workflow projects will use C# expressions.
  + C# workflow projects that were created in Visual Studio 2010 and that have Visual Basic expressions are compatible with C# workflow projects that use C# expressions.
* Versioning enhancements:
  + The new [WorkflowIdentity](http://msdn.microsoft.com/en-us/library/system.activities.workflowidentity.aspx) class, which provides a mapping between a persisted workflow instance and its workflow definition.
  + Side-by-side execution of multiple workflow versions in the same host, including [WorkflowServiceHost](http://msdn.microsoft.com/en-us/library/system.servicemodel.activities.workflowservicehost.aspx).
  + In Dynamic Update, the ability to modify the definition of a persisted workflow instance.
* Contract-first workflow service development, which provides support for automatically generating activities to match an existing service contract.

For more information, see [What's New in Windows Workflow Foundation](http://go.microsoft.com/fwlink/?LinkId=228176).

[.NET for Windows Store apps](javascript:void(0))

Windows Store apps are designed for specific form factors and leverage the power of the Windows operating system. A subset of the .NET Framework 4.5 or 4.5.1 is available for building Windows Store apps for Windows by using C# or Visual Basic. This subset is called .NET for Windows Store apps and is discussed in an [overview](http://go.microsoft.com/fwlink/?LinkId=228491) in the Windows Dev Center.

[Portable Class Libraries](javascript:void(0))

The Portable Class Library project in Visual Studio 2012 (and later versions) enables you to write and build managed assemblies that work on multiple .NET Framework platforms. Using a Portable Class Library project, you choose the platforms (such as Windows Phone and .NET for Windows Store apps) to target. The available types and members in your project are automatically restricted to the common types and members across these platforms. For more information, see [Cross-Platform Development with the .NET Framework](http://msdn.microsoft.com/en-us/library/gg597391.aspx).