**What is New in Windows Communication Foundation**

This topic discusses features new to Windows Communication Foundation (WCF).

**Configuration Based Activation**

Normally when hosting a Windows Communication Foundation (WCF) service under Internet Information Services (IIS) or Windows Process Activation Service (WAS), you must provide a .svc file. The .svc file contains the name of the service and an optional custom service host factory. This additional file adds manageability overhead. The configuration-based activation feature removes the requirement to have a .svc file and therefore the associated overhead. For more information, see [Configuration-Based Activation in IIS and WAS](http://msdn.microsoft.com/en-us/library/ee358764.aspx), [Configuration-Based Activation](http://msdn.microsoft.com/en-us/library/dd807499.aspx).

**System.Web.Routing Integration**

When hosting a Windows Communication Foundation (WCF) service in IIS, you place a .svc file in the virtual directory. This .svc file specifies the service host factory to use as well as the class that implements the service. When making requests to the service you specify the .svc file in the URI, for example: http://contoso.com/EmployeeServce.svc. For programmers writing REST services, this type of URI is not optimal. URIs for REST services specify a specific resource and normally do not have any extensions. The System.Web.Routing integration feature allows you to host a WCF service that responds to URIs without an extension. For more information, see [System.Web.Routing Integration](http://msdn.microsoft.com/en-us/library/ee358760.aspx), [SystemWebRouting Integration Sample](http://msdn.microsoft.com/en-us/library/ee523210.aspx).

**Multiple IIS Site Bindings Support**

When hosting a Windows Communication Foundation (WCF) service under Internet Information Services (IIS) 7.0, you may want to provide multiple base addresses that use the same protocol on the same site. This allows the same service to respond to a number of different URIs. This is useful when you want to host a service that listens on http://www.contoso.com and http://contoso.com. It is also useful to create a service that has a base address for internal users and a separate base address for external users. For more information, see [Supporting Multiple IIS Site Bindings](http://msdn.microsoft.com/en-us/library/ee358763.aspx),

**Routing Service**

The Routing Service is a generic SOAP intermediary that acts as a message router. The core functionality of the Routing Service is the ability to route messages based on the message content, which allows a message to be forwarded to a client endpoint based on a value within the message itself, in either the header or the message body. For more information, see [Routing](http://msdn.microsoft.com/en-us/library/ee517421.aspx), [Routing Services](http://msdn.microsoft.com/en-us/library/dd699773.aspx) .

**Support for WS-Discovery**

The Service Discovery feature enables client applications to dynamically discover service addresses at runtime in an interoperable way using WS-Discovery. The WS-Discovery specification outlines the message exchange patterns (MEPs) required for performing light-weight discovery of services, both by multicast (ad hoc) and unicast (utilizing a network resource). For more information, see [WCF Discovery](http://msdn.microsoft.com/en-us/library/dd456782.aspx), [Discovery (Samples)](http://msdn.microsoft.com/en-us/library/dd483326.aspx).

**Support for HTTP Decompression**

WCF clients now have built-in support for HTTP decompression. A client can decompress received messages that were compressed with the gzip/deflate format.

The following table contains links that instruct how to enable an IIS server for compression.

|  |  |
| --- | --- |
| IIS 6.0 | http://go.microsoft.com/fwlink/?LinkID=189308&clcid=0x409 |
| IIS 7.0 | http://go.microsoft.com/fwlink/?LinkId=189309 |

By default, a WCF client sends an **Accept-Encoding** header to the server to notify that it can decompress data. This default behavior can be turned off by creating a custom binding based on the [HttpTransportBindingElement](http://msdn.microsoft.com/en-us/library/aa751804.aspx) and setting the [DecompressionEnabled](http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.httptransportbindingelement.decompressionenabled.aspx) property to **false**.

**Standard Endpoints**

Standard endpoints are pre-defined endpoints that have one or more of their properties (address, binding, contract) fixed. For example, all metadata exchange endpoints specify IMetadataExchange as their contract, so the developer does not have to specify the contract. The standard MEX endpoint therefore has a fixed IMetadataExchange contract. For more information, see [Standard Endpoints](http://msdn.microsoft.com/en-us/library/ee358762.aspx), .

**Workflow Services**

With the introduction of a set of messaging activities, it is easier than ever to implement workflows that send and receive data. These messaging activities allow you to model complex message exchange patterns that go outside of the traditional send/receive or RPC-style method invocation. For more information, see [Workflow Services](http://msdn.microsoft.com/en-us/library/dd456788.aspx), [Services](http://msdn.microsoft.com/en-us/library/dd699765.aspx), [Services](http://msdn.microsoft.com/en-us/library/ee624145.aspx).

**Target Framework Attribute**

The target framework attribute is used to specify the version of the .NET Framework an application hosted in IIS or WAS is targeting. It allows you to build applications that target .NET Framework 2.0, 3.5, or 4 using Visual Studio. It is an attribute set within a <**compilation**> tag in an application's Web.config file as shown in the following example.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl16_code');" \o "Copy Code)

<compilation debug="false"

targetFramework="4.0">

<assemblies>

<add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

</assemblies>

</compilation>

When an application hosted in IIS or WAS targets a version of the .NET Framework that is not installed, an exception is thrown that indicates the problem. If the target framework moniker is not specified in the application's Web.config file, the value is inferred from the application pool version configured in IIS.

Because of this new feature, if you try to host a WCF service written with .NET Framework 3.5 on a machine running .NET Framework 4, you may get a [ProtocolException](http://msdn.microsoft.com/en-us/library/system.servicemodel.protocolexception.aspx) with the following text:

Unhandled Exception: System.ServiceModel.ProtocolException: The content type text/html; charset=utf-8 of the response message does not match the content type of the binding (application/soap+xml; charset=utf-8). If using a custom encoder, be sure that the IsContentTypeSupported method is implemented properly. The first 1024 bytes of the response were: '<html> <head> <title>The application domain or application pool is currently running version 4.0 or later of the .NET Framework. This can occur if IIS settings have been set to 4.0 or later for this Web application, or if you are using version 4.0 or later of the ASP.NET Web Development Server. The &lt;compilation&gt; element in the Web.config file for this Web application does not contain the required'targetFrameworkMoniker' attribute for this version of the .NET Framework (for example, '&lt;compilation targetFrameworkMoniker=&quot;.NETFramework,Version=v4.0&quot;&gt;'). Update the Web.config file with this attribute, or configure the Web application to use a different version of the .NET Framework.</title>...

This error occurs because the application domain that IIS is running within is running .NET Framework 4 and the WCF service is expecting to run under .NET Framework 3.5. For more information about how to fix this problem, see [How to: Host a WCF Service Written with .NET Framework 3.5 in IIS Running Under .NET Framework 4](http://msdn.microsoft.com/en-us/library/dd759047.aspx).

**WCF REST**

**Caching**

The .NET Framework 4 enables you to leverage the declarative caching mechanism already available in ASP.NET in your WCF REST services. This allows you to cache responses from your WCF REST service operations. When a user sends an HTTP GET to your service that is configured for caching, ASP.NET sends back the cached response and the service method is not called. When the cache expires, the next time a user sends an HTTP GET, your service method is called and the response is once again cached.

The .NET Framework 4 also allows you to implement conditional HTTP GET caching. In REST scenarios a conditional HTTP GET is often used by services to implement intelligent HTTP caching as described in the [HTTP Specification](http://go.microsoft.com/fwlink/?LinkId=165800). For more information, see [Caching Support for WCF Web HTTP Services](http://msdn.microsoft.com/en-us/library/ee230443.aspx), [Caching and Automatic Help Page](http://msdn.microsoft.com/en-us/library/dd807517.aspx).

**Formats Support**

The WCF web HTTP programming model allows you to dynamically determine the best format for a service operation to return its response in. Responses can be set automatically for XML and JSON based on the accept header. Helper APIs have been added to programmatically set the format of an operation. For more information, see [WCF Web HTTP Formatting](http://msdn.microsoft.com/en-us/library/ee476510.aspx), [Automatic Format Selection](http://msdn.microsoft.com/en-us/library/ee662964.aspx), [Advanced Format Selection](http://msdn.microsoft.com/en-us/library/ee662965.aspx).

**HTTP REST Error Handling**

WCF web HTTP error handling enables you to return errors from WCF REST services that specify an HTTP status code and return error details using the same format as the operation (for example, XML or JSON). For more information, see [WCF Web HTTP Error Handling](http://msdn.microsoft.com/en-us/library/ee513227.aspx), .

**Deployment Features**

The configuration needed to run a service has been simplified and new standard endpoints have been introduced to further simply service configuration. For more information about the new simplified configuration see, [Simplified Configuration](http://msdn.microsoft.com/en-us/library/ee358768.aspx). For more information about standard endpoints see, [Standard Endpoints](http://msdn.microsoft.com/en-us/library/ee358762.aspx).

When hosting a Windows Communication Foundation (WCF) service in IIS you place a .svc file in the virtual directory. This .svc file specifies the service host factory to use as well as the class that implements the service. When making requests to the service you specify the .svc file in the URI, for example: http://contoso.com/EmployeeServce.svc. For programmers writing REST services this type of URI is not optimal. URIs for REST services specify a specific resource and normally do not have any extensions. The System.Web.Routing integration feature allows you to host a WCF REST service that responds to URIs without an extension. For more information, see [System.Web.Routing Integration](http://msdn.microsoft.com/en-us/library/ee358760.aspx).

**Cross Domain JavaScript**

JSON Padding (JSONP) is a mechanism that enables cross site scripting support in web browsers. JSONP is designed around the ability of web browsers to load scripts from a site different from the one the current loaded document was retrieved from. The mechanism works by padding the JSON payload with a user-defined callback function name, for example:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl29_code');" \o "Copy Code)

callback({ “a” = \“b\” });

In the above example the JSON payload, {“a” = \”b\”}, is wrapped in a function call, callback. The callback function must already be defined in the current web page. The content type of a JSONP response is “application/javascript”. For more information, see [JSONP](http://msdn.microsoft.com/en-us/library/ee834511.aspx).

**WCF REST Service Help Page**

.NET Framework version 4 provides an automatic help page for WCF REST services. This help page lists a description of each operation, request and response formats, and schemas. For more information, see [WCF Web HTTP Service Help Page](http://msdn.microsoft.com/en-us/library/ee230442.aspx), [Caching and Automatic Help Page](http://msdn.microsoft.com/en-us/library/dd807517.aspx).