This page is specific to

Microsoft Visual Studio 2010/.NET Framework 4

**Windows Communication Foundation Bindings Overview**

Bindings are objects that are used to specify the communication details that are required to connect to the endpoint of a Windows Communication Foundation (WCF) service. Each endpoint in a WCF service requires a binding to be well-specified. This topic outlines the types of communication details that the bindings define, the elements of a binding, what bindings are included in WCF, and how a binding can be specified for an endpoint.

**What a Binding Defines**

The information in a binding can be very basic, or very complex. The most basic binding specifies only the transport protocol (such as HTTP) that must be used to connect to the endpoint. More generally, the information a binding contains about how to connect to an endpoint falls into one of the following categories.

Protocols

Determines the security mechanism being used: either reliable messaging capability or transaction context flow settings.

Encoding

Determines the message encoding (for example, text or binary).

Transport

Determines the underlying transport protocol to use (for example, TCP or HTTP).

**The Elements of a Binding**

A binding basically consists of an ordered stack of binding elements, each of which specifies part of the communication information required to connect to a service endpoint. The two lowest layers in the stack are both required. At the base of the stack is the transport binding element and just above this is the element that contains the message encoding specifications. The optional binding elements that specify the other communication protocols are layered above these two required elements. For more information about these binding elements and their correct ordering, see [Custom Bindings](http://msdn.microsoft.com/en-us/library/aa347793.aspx).

**System-Provided Bindings**

The information in a binding can be complex, and some settings may not be compatible with others. For this reason, WCF includes a set of system-provided bindings. These bindings are designed to cover most application requirements. The following classes represent some examples of system-provided bindings:

* [BasicHttpBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.basichttpbinding.aspx): An HTTP protocol binding suitable for connecting to Web services that conforms to the WS-I Basic Profile specification (for example, ASP.NET Web services-based services).
* [WSHttpBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.wshttpbinding.aspx): An interoperable binding suitable for connecting to endpoints that conform to the WS-\* protocols.
* [NetNamedPipeBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.netnamedpipebinding.aspx): Uses the .NET Framework to connect to other WCF endpoints on the same machine.
* [NetMsmqBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.netmsmqbinding.aspx): Uses the .NET Framework to create queued message connections with other WCF endpoints.

For a complete list, with descriptions, of all the WCF-provided bindings, see [System-Provided Bindings](http://msdn.microsoft.com/en-us/library/ms730879.aspx).

**Using Your Own Bindings**

If none of the system-provided bindings included has the right combination of features that a service application requires, you can create your own binding. There are two ways to do this. You can either create a new binding from pre-existing binding elements using a [CustomBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.custombinding.aspx) object or you can create a completely user-defined binding by deriving from the [Binding](http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.binding.aspx) binding. For more information about creating your own binding using these two approaches, see [Custom Bindings](http://msdn.microsoft.com/en-us/library/aa347793.aspx) and [Creating User-Defined Bindings](http://msdn.microsoft.com/en-us/library/ms733893.aspx).

**Using Bindings**

Using bindings entails two basic steps:

1. Select or define a binding. The easiest method is to choose one of the system-provided bindings included with WCF and use it with its default settings. You can also choose a system-provided binding and reset its property values to suit your requirements. Alternatively, you can create a custom binding or a user-defined binding to have higher degrees of control and customization.
2. Create an endpoint that uses the binding selected or defined.

**Code and Configuration**

You can define bindings in two ways: through code or through configuration. These two approaches do not depend on whether you are using a system-provided binding or a custom binding. In general, using code gives you complete control over the definition of a binding at design time. Using configuration, on the other hand, allows a system administrator or the user of a WCF service or client to change the parameters of a binding without having to recompile the service application. This flexibility is often desirable because there is no way to predict specific machine requirements on which a WCF application is to be deployed. Keeping the binding (and the addressing) information out of the code allows them to change without requiring recompilation or redeployment of the application. Note that bindings defined in code are created after bindings specified in configuration, allowing the code-defined bindings to overwrite any configuration-defined bindings.