This page is specific to

Microsoft Visual Studio 2010/.NET Framework 4

**What Is Windows Communication Foundation**

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data. A few sample scenarios include:

* A secure service to process business transactions.
* A service that supplies current data to others, such as a traffic report or other monitoring service.
* A chat service that allows two people to communicate or exchange data in real time.
* A dashboard application that polls one or more services for data and presents it in a logical presentation.
* Exposing a workflow implemented using Windows Workflow Foundation as a WCF service.
* A Silverlight application to poll a service for the latest data feeds.

While creating such applications was possible prior to the existence of WCF, WCF makes the development of endpoints easier than ever. In summary, WCF is designed to offer a manageable approach to creating Web services and Web service clients.

**Features of WCF**

WCF includes the following set of features. For more information, see [WCF Feature Details](http://msdn.microsoft.com/en-us/library/ms733103.aspx).

* **Service Orientation**  
    
  One consequence of using WS standards is that WCF enables you to create *service oriented* applications. Service-oriented architecture (SOA) is the reliance on Web services to send and receive data. The services have the general advantage of being loosely-coupled instead of hard-coded from one application to another. A loosely-coupled relationship implies that any client created on any platform can connect to any service as long as the essential contracts are met.
* **Interoperability**  
    
  WCF implements modern industry standards for Web service interoperability. For more information about the supported standards, see [Interoperability and Integration](http://msdn.microsoft.com/en-us/library/ms730017.aspx).
* **Multiple Message Patterns**  
    
  Messages are exchanged in one of several patterns. The most common pattern is the request/reply pattern, where one endpoint requests data from a second endpoint. The second endpoint replies. There are other patterns such as a one-way message in which a single endpoint sends a message without any expectation of a reply. A more complex pattern is the duplex exchange pattern where two endpoints establish a connection and send data back and forth, similar to an instant messaging program. For more information about how to implement different message exchange patterns using WCF see [Contracts](http://msdn.microsoft.com/en-us/library/aa702732.aspx).
* **Service Metadata**  
    
  WCF supports publishing service metadata using formats specified in industry standards such as WSDL, XML Schema and WS-Policy. This metadata can be used to automatically generate and configure clients for accessing WCF services. Metadata can be published over HTTP and HTTPS or using the Web Service Metadata Exchange standard. For more information, see [Metadata](http://msdn.microsoft.com/en-us/library/ms731823.aspx).
* **Data Contracts**  
    
  Because WCF is built using the .NET Framework, it also includes code-friendly methods of supplying the contracts you want to enforce. One of the universal types of contracts is the data contract. In essence, as you code your service using Visual C# or Visual Basic, the easiest way to handle data is by creating classes that represent a data entity with properties that belong to the data entity. WCF includes a comprehensive system for working with data in this easy manner. Once you have created the classes that represent data, your service automatically generates the metadata that allows clients to comply with the data types you have designed. For more information, see [Using Data Contracts](http://msdn.microsoft.com/en-us/library/ms733127.aspx)
* **Security**  
    
  Messages can be encrypted to protect privacy and you can require users to authenticate themselves before being allowed to receive messages. Security can be implemented using well-known standards such as SSL or WS-SecureConversation. For more information, see [Windows Communication Foundation Security](http://msdn.microsoft.com/en-us/library/ms732362.aspx).
* **Multiple Transports and Encodings**  
    
  Messages can be sent on any of several built-in transport protocols and encodings. The most common protocol and encoding is to send text encoded SOAP messages using is the HyperText Transfer Protocol (HTTP) for use on the World Wide Web. Alternatively, WCF allows you to send messages over TCP, named pipes, or MSMQ. These messages can be encoded as text or using an optimized binary format. Binary data can be sent efficiently using the MTOM standard. If none of the provided transports or encodings suit your needs you can create your own custom transport or encoding. For more information about transports and encodings supported by WCF see [Transports in Windows Communication Foundation](http://msdn.microsoft.com/en-us/library/ms729350.aspx).
* **Reliable and Queued Messages**  
    
  WCF supports reliable message exchange using reliable sessions implemented over WS-Reliable Messaging and using MSMQ. For more information about reliable and queued messaging support in WCF see [Queues and Reliable Sessions](http://msdn.microsoft.com/en-us/library/ms732355.aspx).
* **Durable Messages**  
    
  A durable message is one that is never lost due to a disruption in the communication. The messages in a durable message pattern are always saved to a database. If a disruption occurs, the database allows you to resume the message exchange when the connection is restored. You can also create a durable message using the Windows Workflow Foundation (WF). For more information, see [Workflow Services](http://msdn.microsoft.com/en-us/library/dd456788.aspx).
* **Transactions**  
    
  WCF also supports transactions using one of three transaction models: WS-AtomicTtransactions, the APIs in the [System.Transactions](http://msdn.microsoft.com/en-us/library/system.transactions.aspx) namespace, and Microsoft Distributed Transaction Coordinator. For more information about transaction support in WCF see [Transactions](http://msdn.microsoft.com/en-us/library/ms730266.aspx).
* **AJAX and REST Support**  
    
  REST is an example of an evolving Web 2.0 technology. WCF can be configured to process "plain" XML data that is not wrapped in a SOAP envelope. WCF can also be extended to support specific XML formats, such as ATOM (a popular RSS standard), and even non-XML formats, such as JavaScript Object Notation (JSON).
* **Extensibility**  
    
  The WCF architecture has a number of extensibility points. If extra capability is required, there are a number of entry points that allow you to customize the behavior of a service. For more information about available extensibility points see [Extending WCF](http://msdn.microsoft.com/en-us/library/ms733848.aspx).

**WCF Integration with Other Microsoft Technologies**

WCF is a flexible platform. Because of this extreme flexibility, WCF is also used in several other Microsoft products. By understanding the basics of WCF, you have an immediate advantage if you also use any of these products.

The first technology to pair with WCF was the Windows Workflow Foundation (WF). Workflows simplify application development by encapsulating steps in the workflow as “activities.” In the first version of Windows Workflow Foundation, a developer had to create a host for the workflow. The next version of Windows Workflow Foundation was integrated with WCF. That allowed any workflow to be easily hosted in a WCF service; you can do this by automatically choosing the WF/WCF a project type in Visual Studio 2010 and Visual Studio 2010.

Microsoft BizTalk Server R2 also utilizes WCF as a communication technology. BizTalk is designed to receive and transform data from one standardized format to another. Messages must be delivered to its central message box where the message can be transformed using either a strict mapping or by using one of the BizTalk features such as its workflow engine. BizTalk can now use the WCF Line of Business (LOB) adapter to deliver messages to the message box.

Microsoft Silverlight is a platform for creating interoperable, rich Web applications that allow developers to create media-intensive Web sites (such as streaming video). Beginning with version 2, Silverlight has incorporated WCF as a communication technology to connect Silverlight applications to WCF endpoints.

Microsoft .NET Services is a cloud computing initiative that uses WCF for building Internet-enabled applications. Use .NET Services to create WCF services that operate across trust boundaries.

The hosting features of Windows Server AppFabric application server is specifically built for deploying and managing applications that use WCF for communication. The hosting features includes rich tooling and configuration options specifically designed for WCF-enabled applications.