Bindings and Security

.NET Framework (current version)

The system-provided bindings included with Windows Communication Foundation (WCF) offer a quick way to program WCF applications. With one exception, all the bindings have a default security scheme enabled. This topic helps you select the right binding for your security needs.

For an overview of WCF security, see Security Overview. For more information about programming WCF using bindings, see Programming WCF Security.

If you have already selected a binding, you can find out more about the run-time behaviors that are associated with security in Security Behaviors in WCF.

Some security functions are not programmable using the system-provided bindings. For more control using a custom binding, see Security Capabilities with Custom Bindings.

Security Functions of Bindings

WCF includes a number of system-provided bindings that meet most needs. If a particular binding does not suffice, you can also create a custom binding. For a list of system-provided bindings, see System-Provided Bindings. For more information about custom bindings, see Custom Bindings.

Every binding in WCF has two forms: as an API and as an XML element used in a configuration file. For example, the **WSHttpBinding** (API) has a counterpart in the <wsHttpBinding>.

The following section lists both forms for each binding and summarizes the security features.

BasicHttp

In code, use the BasicHttpBinding class; in configuration, use the <basicHttpBinding>.

This binding is designed for use with a range of existing technologies, including the following:

- ASP.NET Web services (ASMX), version 1.
- Web Service Enhancements (WSE) applications.
- Basic Profile as defined in the Web Services Interoperability (WS-I) specification (http://go.microsoft.com/fwlink/?LinkId=38955).
- Basic security profile as defined in WS-I.

By default, this binding is not secure. It is designed to interoperate with ASMX services. When security is enabled, the binding is designed for seamless interoperation with Internet Information Services (IIS) security mechanisms, such as basic authentication, digest, and integrated Windows security. For more information, see Transport Security Overview. This binding supports the following:

- HTTPS transport security.

- HTTP basic authentication.
- WS-Security.

For more information, see BasicHttpSecurity, BasicHttpMessageSecurity, BasicHttpMessageCredentialType, and BasicHttpSecurityMode.

WSHttpBinding

In code, use the WSHttpBinding class; in configuration, use the <wsHttpBinding>.

By default, this binding implements the WS-Security specification and provides interoperability with services that implement the WS-* specifications. It supports the following:

- HTTPS transport security.
- WS-Security.
- HTTPS transport protection with SOAP message credential security for authenticating the caller.

For more information, see WSHttpSecurity, MessageSecurityOverHttp, MessageCredentialType, SecurityMode, HttpTransportSecurity, HttpClientCredentialType, and HttpProxyCredentialType.

WSDualHttpBinding

In code, use the WSDualHttpBinding class; in configuration, use the <wsDualHttpBinding>.

This binding is designed to enable duplex service applications. This binding implements the WS-Security specification for message-based transfer security. Transport security is not available. By default, it provides the following features:

- Implements WS-Reliable Messaging for reliability.
- Implements WS-Security for transfer security and authentication.
- Uses HTTP for message delivery.
- Uses text/XML message encoding.

Using WS-Security (message-layer security), the binding allows you to configure the following parameters:

- The security algorithm suite to determine the cryptographic algorithm.
- Binding options for the following:
    - Providing service credentials available out-of-band at the client.
    - Providing service credentials negotiated from the service as part of channel setup.

For more information, see WSDualHttpSecurity and WSDualHttpSecurityMode.

NetTcpBinding

In code, use the NetTcpBinding class; in configuration, use the <netTcpBinding>.

This binding is optimized for cross-machine communication. By default, it has the following characteristics:

- Implements transport-layer security.
- Leverages Windows security for transfer security and authentication.
- Uses TCP for transport.
- Implements binary message encoding.
- Implements WS-Reliable Messaging.

Options include the following:

- Message-layer security (using WS-Security).
- Transport security with message credential—confidentiality and integrity provided by Transport Layer Security (TLS) over TCP, and credentials for authorization provided by WS-Security.

For more information, see NetTcpSecurity, TcpTransportSecurity, TcpClientCredentialType, MessageSecurityOverTcp, and MessageCredentialType.

NetNamedPipeBinding

In code, use the NetNamedPipeBinding class; in configuration, use the <netNamedPipeBinding>.

This binding is optimized for cross-process communication (usually on the same machine). By default, this binding has the following characteristics:

- Uses transport security for message transfer and authentication.
- Uses named pipes for message delivery.
- Implements binary message encoding.
- Encryption and message signing.

Options include the following:

- Authentication using Windows security.

For more information, see NetNamedPipeSecurity, NetNamedPipeSecurityMode, and NamedPipeTransportSecurity.

MsmqIntegrationBinding

In code, use the MsmqIntegrationBinding class; in configuration, use the <msmqIntegrationBinding>.

This binding is optimized for creating WCF clients and services that interoperate with non-WCF Microsoft Message Queuing (MSMQ) endpoints.

By default, this binding uses transport security and provides the following security characteristics:

- Security can be disabled (None).
- MSMQ transport security (Transport).

For more information, see NetMsmqSecurity and NetMsmqSecurityMode.

NetMsmqBinding

In code, use the NetMsmqBinding class; in configuration, use the <netMsmqBinding>.

This binding is intended for use when creating WCF services that require MSMQ queued message support.

By default, this binding uses transport security and provides the following security characteristics:

- Security can be disabled (None).
- MSMQ transport security (Transport).
- SOAP-based message security (Message).
- Simultaneous Transport and Message security (Both).
- Client Credential Types supported: None, Windows, UserName, Certificate, IssuedToken.

The Certificate credential is supported only when the security mode is set to either Both or Message.

For more information, see MessageSecurityOverMsmq and MsmqTransportSecurity.

WSFederationHttpBinding

In code, use the WSFederationHttpBinding class; in configuration, use the <wsFederationHttpBinding>.

By default, this binding uses WS-Security (message-layer security).

For more information, see Federation, WSFederationHttpSecurity, and WSFederationHttpSecurityMode.

Custom Bindings

If none of the system-provided bindings meets you requirements, you can create a custom binding with a custom security binding element. For more information, see Security Capabilities with Custom Bindings.

Binding Choices

The following table summarizes the features offered in the security mode setting, that is, it lists the features available when the security mode is set to **Transport**, **Message**, or **TransportWithMessageCredential**. Use this table to help you find the security features your application requires.

Setting	**Features**
Transport	Server authentication Client authentication Point-to-point security Interoperability Hardware acceleration High throughput Secure firewall High-latency applications Re-encryption across multiple hops
Message	Server authentication Client authentication End-to-end security Interoperability Rich claims Federation Multifactor authentication Custom tokens Notary/timestamp service High-latency applications Persistence of message signatures

TransportWithMessageCredential	Server authentication Client authentication Point-to-point security Interoperability Hardware acceleration High throughput Rich client claims Federation Multifactor authentication Custom tokens Secure firewall High-latency applications Re-encryption across multiple hops

The following table lists the bindings that support the various mode settings. Select a binding from the table to use to create your service endpoint.

Binding	**Transport mode support**	**Message mode support**	**TransportWithMessageCredential support**
BasicHttpBinding	Yes	Yes	Yes
WSHttpBinding	Yes	Yes	Yes
WSDualHttpBinding	No	Yes	No
NetTcpBinding	Yes	Yes	Yes
NetNamedPipeBinding	Yes	No	No
NetMsmqBinding	Yes	Yes	No
MsmqIntegrationBinding	Yes	No	No
wsFederationHttpBinding	No	Yes	Yes

Transport Credentials in Bindings

The following table lists the client credential types available when using either **BasicHttpBinding** or **WSHttpBinding** in transport security mode.

Type	Description
None	Specifies that the client does not need to present any credential. This translates to an anonymous client.
Basic	Basic authentication. For more information, see RFC 2617 – HTTP Authentication: Basic and Digest Authentication, available at http://go.microsoft.com/fwlink/?LinkId=84023.
Digest	Digest authentication. For more information, see RFC 2617 – HTTP Authentication: Basic and Digest Authentication, available at http://go.microsoft.com/fwlink/?LinkId=84023.
NTLM	NT LAN Manager (NTLM) authentication.
Windows	Windows authentication.
Certificate	Authentication performed using a certificate.
IssuedToken	Allows the service to require that the client be authenticated using a token issued by a security token service or by CardSpace. For more information, see Federation and Issued Tokens.

Message Client Credentials in Bindings

The following table lists the client credential types available when using a binding in Message security mode.

Type	Description
None	Allows the service to interact with anonymous clients.
Windows	Allows SOAP message exchanges to be made under the authenticated context of a Windows credential.
UserName	Allows the service to require that the client be authenticated using a user name credential. Note that when the security mode is set to **TransportWithMessageCredential**, WCF does not support sending a password digest or deriving keys using password and using such keys for Message mode security. As such, WCF enforces that the transport is secured when using user name credentials.
Certificate	Allows the service to require that the client be authenticated using a certificate.
IssuedToken	Allows the service to use a security token service to supply a custom token.

See Also

- Security Overview
- Securing Services and Clients
- Selecting a Credential Type
- Security Capabilities with Custom Bindings
- Security Behaviors in WCF
- Security Model for Windows Server App Fabric

;