

Windows Communication Foundation (WCF) is Microsoft’s new framework for building distributed systems. It unifies and builds on the diverse set of existing distribution mechanisms, which include ASP.NET Web services and .NET remoting. WCF enables developers to produce highly configurable, secure, reliable and transactional services using a single simplified programming model. And since WCF supports the WS-\* series of Web service standards, it enables simple interoperability with other platforms and technologies.

This 3 day course provides a thorough grounding in this important technology. The first chapter covers the essential concepts and shows how to implement WCF services and clients. The “ABC” of address, binding and contracts are covered in detail. Service contracts and data contracts are elaborated, and instance management is covered. The course includes a discussion of error handling and security and concludes with coverage of the WCF Routing Service. A large number of working examples and lab exercises are provided. The course uses .NET 4.5 and Visual Studio 2012.

### Course Objectives:

- Learn what WCF is, and how it unites existing distribution mechanisms.
- Gain an understanding of addressing and binding in WCF services.
- Use service contracts and data contracts.
- Implement WCF services and clients.
- Perform configuration in both code and configuration files.
- Understand faults and handle errors in WCF applications.
- Implement security in WCF applications.
- Understand the WCF Routing Service.

**Audience:** .NET Programmers who wish to use WCF to create and consume web services.

**Prerequisites:** Students should have a good working knowledge of building .NET applications with C#. Knowledge of building distributed systems and Web services will also be an advantage.

**Number of Days:** 3 days

<b>1</b>	<b>WCF Essentials</b> What is WCF? WCF Services Service Orientation WCF = ABC Address, Binding, Contract Hosting Services A Service Contract Visual Studio WCF Test Host Closing the Test Host Manually Self-Hosting ServiceHost Class	Host Life Cycle WCF Clients Channels Channel Factory Base Address Uri Class Configuration Files Simplified Host Code Proxy Initialization Metadata Exchange Behaviors A Service in a Browser
----------	---	---

	Standard Endpoints		XSD for Data Contract
	WCF Architecture		Arrays
	ServiceHost and Channel Factory		Array in XML Schema
	Service Contexts and Instances		Array in Proxy
<b>2</b>	<b>Addresses and Bindings</b>		Generic Collections
	Addresses		Generic Collections in XML Schema
	Transports		Generic Collections in Proxy
	Bindings		Enumerations in Data Contracts
	Message Exchange Patterns		Employee Client Program
	Security		Saving and Restoring
	Choosing a Binding		Serialization in WCF and .NET
	HTTP Bindings		SOAP Serialization
	TCP and Named Pipe Bindings		DataContract Serialization
	MSMQ Bindings		JSON Serialization
	Importance of BasicHttpBinding		Using XmlSerializer
	Working with Endpoints		Restoring Data
	Default Endpoints & Bindings		Versioning
	Helper Methods		New and Missing Members
	ServiceDescription Class		New Client of Old Service
	Multiple Endpoints Configuration File		Round Trip
	Simple Host Code		Required Members
<b>3</b>	<b>Service Contracts</b>		OnDeserializing Event
	Service Contracts at Class Level	<b>6</b>	<b>More about Service Contracts</b>
	Service Contracts at Interface Level		Versioning Service Contracts
	Benefits of Interface Level Definition		Version 1 Service
	A Service with Multiple Contracts		Version 2 Service
	ServiceContractAttribute		New Operations
	Attributes in WSDL		Version 3 Service
	Viewing WSDL Files		Version 1 Client/Version 3 Service
	Contract Inheritance		Version 2 Client/Version 2 Service
	Operation Overloading		Version 3 Client/Version 3 Service
	Enabling Operation Overloading		Message Exchange Patterns
	Operation Overloading Client		Request-Reply
<b>4</b>	<b>Instance Management</b>		Oneway / Duplex
	Behaviors		Callbacks
	WCF Behaviors		Invoking a Callback
	Configuring Behaviors in Code		Callback on a Client
	WCF Instancing Models		Asynchronous Proxies
	Per-Call Instancing		Threading Considerations
	Per-Session Instancing		Task-Based Asynchronous Patterns
	Sessions and Threading		Task-Based Client
	Singleton Instancing		WebSockets
	Which Model to Use?	<b>7</b>	<b>Handling Errors</b>
	Windows Forms WCF Clients		Errors in Distributed Systems
<b>5</b>	<b>Data Contracts</b>		Errors in .NET and WCF
	Data Contracts		Service Library Code

	Client Code	Registering ASP.NET
	Client Exception Handling	A Service Contract
	Fault Exceptions	A Website for the Service
	Exception Details in Faults	WCF ServiceTemplate
	Exception Details	Service Configuration
	Exception Dialog	Referencing the Class Library
	Fault Contracts	Examining the Service in the Browser
	Custom Faults	WCR Clients
	Faulted Channels	Creating WCF Clients
<b>8</b>	<b>WCF Security</b>	Service as an IIS Application
	Services and Security	Converting to an Application
	Security Aspects of Services	Configuring as an Application
	Transfer Security	Moving a WCF Solution
	Transport Security	
	Configuring Transport Security	
	Host and Client Security Configuration	
	Client's Security Configuration	
	Message Security	
	Configuring Message Security	
	Other Security Modes	
	Certificates	
	Managing Certificates	
	Exception Details	
	Client Certificate Configuration	
	Sending Credentials	
	Username Credentials	
<b>9</b>	<b>WCF Routing Service</b>	
	Service Contract and Implementation	
	Service Configuration	
	Hosting the Service	
	Client Application	
	Configuring the Router	
	Router Configuration File	
	Routing Contracts	
	Message Filters	
	EndpointName Message Filter	
	EndPointName Router Configuration	
	Incoming Endpoints and the Client	
	Error Handling	
	WCF Routing Scenarios	
<b>10</b>	<b>Appendix A – Learning Resources</b>	
<b>11.</b>	<b>Appendix B. Hosting in IIS 7.5</b>	
	Internet Information Services	
	Installing IIS 7.5	
	WCF with IIS 7.5	
	.NET Framework Version	