

Course Outline

6461- Windows Communication Foundation



Duration: 3 days (18 hours)

Target Audience:

This course is intended for application developers who know how to build and consume Web services in Microsoft .NET Framework 2.0 and how to use the common features of the base class library. The application developers do not have to understand concepts such as advanced WS-*, Web Services Enhancements (WSE), service life cycle management, and diagnostics.

Prerequisites:

Before attending this course, students should have intermediate experience of developing applications by using previous versions of Microsoft Visual Studio development system at level 200.

Topics Covered:

- Module 1: Getting Started with Windows Communication Foundation
 - Designing an Application to Be Part of a Service Oriented Architecture
 - Overview of WCF Architecture
 - Using a Language-Level Interface As a Service Contract
 - Implementing a Simple WCF Service in Visual Studio 2008
 - Consuming a simple WCF service in Visual Studio 2008
 - Lab: Creating a Simple Service
 - Creating a Simple WCF Service
 - Calling the Simple WCF Service

After completing this module, students will be able to:

- Explain how to design an application as part of a Service Oriented Architecture (SOA).
- Describe the main parts of the WCF architecture.
- Create a simple service contract for a WCF service.
- Implement a simple WCF service in Visual Studio 2008.
- Consume a simple WCF service in Visual Studio 2008.

- Module 2: Configuring and Hosting WCF Services
 - Programmatically Configuring a Managed Application to Host a WCF Service
 - Programmatically Configuring a Managed Application to Call a WCF Service
 - Defining Client and Service Settings by Using File-Based Configuration
 - Selecting a Hosting Option for a WCF Service
 - Deploying a WCF Service
 - Lab: Configure and Host a WCF Service
 - Creating a Programmatically Configured Managed Application to Host a Service
 - Calling a Service Hosted in a Managed Application by Using Programmatic Configuration
 - Defining Service Settings by Using External Configuration
 - Employing Different Hosting Options for a Service

After completing this module, students will be able to:

- Create a programmatically-configured managed application that hosts a WCF service.
- Call a WCF service hosted in a managed application by using programmatic configuration.
- Define WCF service settings by using external configuration.
- Select the best hosting option for a WCF service.
- Deploy a WCF service onto a remote host.

➤ Module 3: Endpoints and Behaviors

- Exposing WCF Services Over Different Endpoints
- Adding Behaviors to Services and Endpoints
- Interoperating with Non-WCF Web services
 - Lab: Changing Service Endpoints and Behaviors
 - Exposing Services by Using Different Bindings
 - Adding Metadata Exchange to a Service
 - Creating WCF Clients and Services That Interoperate with Non-WCF Web Services

After completing this module, students will be able to:

- Expose WCF services by using different bindings.
- Add behaviors to services and endpoints.
- Create WCF clients and services that interoperate with different types of Web services.

➤ Module 4: Debugging and Diagnostics

- Logging Messages
- Activity Tracing
 - Lab: Message Logging and Activity
 - Generating Logging Information for a Service
 - Enabling End-to-End Tracing for a Service

After completing this module, students will be able to:

- Log WCF messages.
- Trace WCF service activity.

➤ Module 5: Designing and Defining Contracts

- Designing a Coherent and Cohesive WCF Service Interface
- Defining a Service Contract
- Defining Operations on a Service
- Defining a Data Contract
 - Lab: Contracts for Services and Data
 - Defining and Implementing a One-Way Operation Contract
 - Passing Complex Data with a Data Contract
 - Defining and Implementing a Callback Contract

After completing this module, students will be able to:

- Design a coherent and cohesive service contract.
- Define a service contract.
- Define operations on a service.
- Define a data contract.

➤ Module 6: Handling Errors

- Relating .NET Exceptions to Service-Level Faults
- Using Faults in a Service
- Handling Faults and Exceptions on Clients
 - Lab: Error Handling
 - Handling Unexpected Errors in a WCF Service

- Add Fault Handling to a WCF Service and the Service Contract

After completing this module, students will be able to:

- Explain how .NET exceptions relate to service-level faults.
- Define fault information in a service contract.
- Handle service exceptions on clients.

➤ Module 7: Improving WCF Service Quality

- Managing WCF Service Instances
- Managing Concurrency Issues
- Improving WCF Service Quality
 - Lab: Improving WCF Service Quality
 - Managing WCF Service Instances
 - Managing Concurrency Issues
 - Throttling Access to a WCF Service
 - Passing Bulk Data Between a WCF Client and Service

After completing this module, students will be able to:

- Manage WCF service instances.
- Manage concurrency issues.
- Improve WCF service performance.

➤ Module 8: Implementing WCF Security

- Overview of Security in WCF
- Applying Overall Security Requirements to a Binding
- Specifying Required Client and Service Credentials
- Working With Security Information
 - Lab: Protecting a Service
 - Applying Security for Internal Network Communication
 - Applying Security for Internet Communication

After completing this module, students will be able to:

- Explain the process for implementing security in WCF.
- Apply overall security requirements to a binding.
- Specify required client and service credentials.
- Work with security information.

➤ Module 9: Implementing Transactions

- Overview of Transactions in a Service-Oriented Application
- Creating Transactional Service Operations
- Enabling the Flow of Transactions from Client to Service
 - Lab: Implementing Transactions for a Service
 - Controlling the Flow of a Transaction from Client to Service
 - Forcing a Transaction to Start When a Service Operation Is Called

After completing this module, students will be able to:

- Explain how transactions work in a service-oriented application.
- Create transactional service operations.
- Control transaction flow from client to service.