Windows Workflow Foundation Using C#

Duration: 4 Days *(Face-to-Face & Remote-Live)*, or 28 Hours *(On-Demand)*

Price: $2095 *(Face-to-Face & Remote-Live)*, or $1495 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom or remote-live attendance.

Students Will Learn

- Understanding the concept of workflow and the architecture of Windows Workflow Foundation (WF)

- Implementing workflow applications using WF

- Using standard activities from the Built-In Activity Library

- Implementing custom activities for workflow applications

- Providing hosting for workflow applications

- Implementing persistence in workflow applications

- Implementing and consume workflow services

- Debugging workflow applications

- Handling errors and perform compensation in workflow applications

Course Description

Microsoft's Windows Workflow Foundation (WF) provides a framework for creating scalable applications consisting of independent program units called activities. This course covers WF 4.5 and uses Visual Studio 2015, which provides a visual designer for creating workflows. The course begins with an overview of the concepts of workflow and the programming model provided by WF. The essentials of WF programming are then covered followed by a systematic treatment of the major features of WF, including the Built-In Activity Library. Primitive and control flow activities are discussed, and the alternatives to hosting workflows are covered. Collection and parallel activities are covered. The creation of custom activities, both using the designer and code, is explained. Flowchart and state machine activities are covered. Persistence and bookmarks are discussed. Workflow Services, using Windows Communication Foundation (WCF) is covered. The course concludes with coverage of debugging, error handling, transactions and compensation. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Course Prerequisites

C# programming experience.

Course Overview

Windows Workflow Foundation Overview

- How Windows Workflow Foundation Supports Workflow Applications
- Concepts of Workflows and Activities
- Runtime Services Provided in WF
- Differences Between WF 3 and WF 4/4.5

Getting Started with WF

- Overall Structure of Workflow Programs
- Implementing Workflows Using C# Code
- Using the Sequence Activity to Implement Workflows Consisting of Multiple Activities
- Creating Workflow Projects Using Visual Studio 2015
- Implementing Workflows Declaratively in XAML by Using the Visual Studio Workflow Designer
- Using the `WriteLine` and `Assign` Activities in Workflow Programs
- Use of Variables, Arguments and C# Expressions in Workflow Programs
- Control Flow Activities
- Illustrate Using the `While` Activity

Primitive and Control Flow Activities

- Using the `InvokeMethod` Activity to Call Both Your Own Methods and Methods of the .NET Framework Class Library
- Performing Input in Your Workflow Programs
- Using the Delay Activity in Workflow Programs
- Using Control Flow Activities:
    - `Parallel`
    - `If`
    - `While`
    - `DoWhile`
    - `Switch`
- Implementing More Complex Workflows that Have Nested Control Structures

Custom Activities

- Overview of Using Custom Activities in Workflow Foundation
- Approaches for Authoring Custom Activities
- Explaining the Use of Arguments in Custom Activities
- Outlining the `Activity` Class Hierarchy
- Implementing Custom Activities Using the `CodeActivity` Class
- Implementing Custom Activities in the Workflow Designer Using the `Activity` Class

Workflow Hosting

- Methods for Hosting Workflows in Workflow Foundation
- Using the `WorkflowInvoker` Class to Invoke a Workflow as if it were a Method Call
- Using the `WorkflowApplication` Class to Host Workflows in More Complex Scenarios
- Assigning Code to Delegate Members

Collection and Parallel Activities

- Using the Collection Activities Provided by the Built-In Activity Library
    - `AddToCollection`
    - `RemoveFromCollection`
    - `ExistsInCollection`
    - `ClearCollection`
- Using the `ForEach<T>` Activity
- Using `ParallelForEach<T>` Activity
- Implementing Asynchronous Behavior

of the `WorkflowApplication` Class to Respond to Life Cycle Events of the Workflow
- Methods of Manually Stopping a Workflow
    - Cancel
    - Terminate
    - Abort
- Hosting Workflows in a Windows Application
- Hosting Workflows in a Web Application

in Workflows Using `AsyncActivity`

Flowchart and State Machine

- Modeling Styles in Windows Workflow 4.5
- Implementing Workflows Using Flowchart Activities
- Using the Flowchart Activity Designer in Visual Studio
- Implementing Workflows Using State Machine Activities
- Using the State Machine Activity Designers in Visual Studio
- Using of Triggers and Entry/Exit Actions in State Machines
- Handling Threading Issues in Workflows with State Machines

Persistence

- Explain the Need of a Persistence Mechanism in Long Running Workflows
- Describing the Role of Bookmarks in Persistence
- How to Setup a Persistence Store in SQL Server
- Implementing a Workflow that Persists its State
- Implementing a Workflow that Persists its State, Unloads, and then Loads its State from the Persistence Store
- How to Serialize Workflow Instance Identification
- Implementing a Long Running Workflow

Workflow Services

- How WCF Unites and Extends Existing Distribution Technologies
- Concepts of Address, Binding, Contract and Endpoint
- How WCF Works with WF
- Messaging Activities Provided by WF
- Implementing a WCF Service by Exposing a Workflow as a Service
- Calling a WCF Service from a Workflow
- Implementing Workflow Service Contracts with Multiple Operations by Using the `Parallel` Activity

Debugging and Error Handling

- Using Visual Studio to Debug Workflow Applications
- Handling Exceptions in Workflows
- Use of Transactions and Compensation in Workflows
- Implementing Compensation in Workflow Applications