

Windows® Presentation Foundation (WPF) and Silverlight Introduction: Hands-On - 4 Days

Course 975 Overview

- You Will Learn How To**
- Create compelling, modern user experiences with Windows Presentation Foundation (WPF)
 - Employ Visual Studio to create WPF windows and pages with XAML, C# and VB
 - Program stand-alone, browser-based (XBAP), navigation-style and Silverlight applications using XAML
 - Produce dynamic content with WPF controls, styles, templates, triggers and animation
 - Leverage the MVVM design pattern and the binding infrastructure for data presentation
 - Build rich Internet applications for Windows and Mac computers using Silverlight
- Course Benefits** Windows Presentation Foundation (WPF) is the next generation platform for creating user interfaces (UIs) with the Microsoft .NET Framework. Using WPF, developers and designers can easily create programs that are dynamic, data-driven and visually appealing. In this course, you program applications that integrate the major components of WPF. You also learn how to target your WPF applications for multiple platforms with Silverlight.
- Who Should Attend** .NET and other developers who want to leverage WPF to build modern, visual applications for Windows, Internet Explorer or Silverlight. .NET programming experience at the level of Course 502, "Programming with .NET Introduction," is assumed.
- Hands-On Training** You gain hands-on experience building WPF applications with XAML, VB and C#. Exercises include:
- Creating WPF stand-alone, browser-based and Silverlight projects using Visual Studio
 - Programming XAML applications and user interfaces with WPF containers and controls
 - Enhancing controls with styles, templates and triggers
 - Integrating Silverlight into Web applications for a rich user experience
 - Synchronizing user interfaces to bindable data sources

Windows® Presentation Foundation (WPF) and Silverlight Introduction: Hands-On - 4 Days

Course 975 Outline

Building XAML Windows and Pages

Introducing WPF

- .NET architectural overview
- Advantages of WPF

Creating WPF applications in Visual Studio

- Setting application and window properties
- Implementing behaviours in code-behind files
- Publishing WPF applications with ClickOnce

Coding in XAML

- Adding controls to containers
- Setting properties using attribute and element syntax
- Referencing external namespaces using XAML
- Handling bubbling and tunneling routed events

Architecting WPF Programs

Building browser-based applications (XBAP)

- Loading XAML pages into the browser
- Moving between pages with the navigation service
- Hyperlinking to other pages and Web sites

Constructing navigation applications

- Loading pages into a navigation window
- Passing data between pages
- Navigating page history
- Creating irregularly shaped and transparent windows

Designing WPF User Interfaces

Leveraging WPF controls

- Button
- TextBox
- TextBlock
- Label
- Checkbox
- Slider
- Image
- Menus
- Status bars
- RichTextBox
- Positioning controls in the Grid and Canvas containers

Managing sophisticated layouts

- Defining grid rows and columns
- Assigning controls to grid cells
- Spanning multiple cells
- Resizing rows and columns with the GridSplitter
- Positioning elements with StackPanel and WrapPanel
- Docking controls to windows using the DockPanel

Styling WPF Controls

Altering the appearance of UI elements

- Margin
- Padding
- Fonts
- Background
- Foreground
- Opacity
- Transparency
- Specifying styles using elements

Maintaining consistent styles with resources

- Naming and extending control styles
- Packaging styles in resource dictionaries
- Customizing controls with templates
- Drawing shapes and gradients
- Converting scalable vector graphics to XAML

Activating WPF Applications

Animating UIs for dynamic presentations

- Moving, sizing and rotating elements with transformations
- Transforming elements with animations
- Adding behaviours to styles using event and property triggers

Optimizing for code reuse

- Streamlining event handling with WPF commands
- Utilizing the common and custom dialogs
- Creating user controls
- Specifying dependency properties

Automating Data Access with Data

Binding

Binding control properties with XAML

- ItemsControl
- ComboBox

- ListBox
- Formatting content with data templates
- Enhancing forms with scrollable and expandable regions

Accessing data using data providers

- Displaying XML data with the XMLDataProvider and XPath
- Accessing collections using the ObjectDataProvider
- Programming custom type-converters

Model-View-ViewModel (MVVM) design pattern

- Binding to the ViewModel class
- Validating user input
- Implementing the INotifyPropertyChanged interface
- Refreshing records automatically using ObservableCollection

Crafting Cross-Platform Silverlight UIs

Creating Silverlight Web applications

- Coding Silverlight with C#, VB and JavaScript
- Delivering video online with the Media Element control
- Employing Silverlight controls
- Exploiting WCF services

Running Silverlight applications outside the browser

- Installing Silverlight applications on Windows and OS X
- Downloading updates
- Detecting the network

Integrating Silverlight with ASP.NET

- Accessing the browser
- Controlling HTML page content
- Invoking JavaScript functions from Silverlight and vice versa