

This course introduces Windows Presentation Foundation or WPF (“Avalon”), the new .NET technology from Microsoft for building rich Windows applications. It was originally part of .NET 3.0, previously called “WinFX” by Microsoft. WPF includes an XML-based markup language for defining program elements, Extensible Application Markup Language (XAML). WPF applications can be created using only code or a combination of code and XAML pages. This course covers the essentials of WPF, providing an orientation to this technology and a firm foundation for creating applications. The course is current to .NET 4.0 and Visual Studio 2010. WPF is a complex technology that can have a steep learning curve. This course approaches the subject in a practical manner, introducing the student to the fundamentals of creating Windows applications using the features of WPF. It includes coverage of both traditional concepts such as controls and new concepts such as XAML, flexible layout, logical resources, dependency properties, routed events, and the loosely-coupled command architecture of WPF. Data binding is discussed in detail, including visual data binding using Visual Studio 2010 and accessing databases using the Entity Data Model.

Audience: NET programmers who want to be able to create rich Windows applications.

Prerequisites: A working knowledge of C# and the .NET Framework.

Number of Days: 4 days

1. Introduction to WPF

- History of Microsoft GUI
- Why WPF?
- When Should I Use WPF?
- WPF and .NET Framework 3.0
- .NET Framework 4.0
- Visual Studio 2010
- WPF Core Types and Infrastructures
- XAML
- Controls
- Data Binding
- Appearance
- Layout and Panels
- Graphics
- Media
- Documents and Printing
- Plan of Course
- Application and Window
- Creating a Button
- Providing an Event Handler
- Specifying Initial Input Focus
- Complete First Program
- Device-Independent Pixels
- Class Hierarchy

Content Property

Simple Brushes

Panels

Children of Panels

TwoControls – Code

Automatic Sizing

2. XAML

What is XAML?

Default Namespace

XAML Language Namespace

.NET Class and Namespace

Elements and Attributes

XAML in Visual Studio 2010

Adding an Event Handler

Layout in WPF

Controlling Size

Margin and Padding

Thickness Structure

Children of Panels

TwoControls – XAML

Automatic Sizing

TwoControls - Code

Orientation

Access keys

- Access keys in XAML
- Content Property
- Checked and Unchecked Events
- Property Element Syntax
- Type Converters
- 3. WPF Controls**
 - Buttons in WPF
 - Using the Button Class
 - Toggle Buttons
 - IsThreeState
 - CheckBox
 - CheckBox Code
 - ToolTip
 - RadioButton
 - GroupBox
 - Images
 - TextBox
 - Initializing the TextBox
 - Clipboard Support
 - Items Controls
 - Selector Controls
 - Using a ListBox
 - Multiple-Selection ListBox
 - Selected Items
 - Using the ComboBox
 - Storing Objects in List Controls
 - Collection Items in XAML
- 4. Layout**
 - Layout in WPF
 - Controlling Size: Review
 - Margin and Padding: Review
 - Thickness Structure: Review
 - SizeDemo Program
 - Top Panel
 - Content Property
 - XAML vs. Code
 - Type Converter
 - Alignment
 - Alignment inside a Stack Panel
 - Vertical Alignment
 - Horizontal Alignment
 - Vertical Alignment in a Window
 - Content Alignment
 - FlowDirection
 - Transforms
 - Panels
- Shapes
- Size and Position
- Attached Properties
- StackPanel
- Children of StackPanel
- WrapPanel
- DockPanel
- Grid
- Using the Collections Editor
- Star Sizing
- Grid.ColumnSpan
- Scrolling
- Scaling
- ScrollViewer and Viewbox
Compared
- 5. Dialogs**
 - Dialog Boxes in WPF
 - MessageBox
 - MessageBox Show Method
 - Common Dialog Boxes
 - Custom Dialogs
 - Modal Dialogs
 - New Product Dialog
 - XAML for New Product Dialog
 - Code for the New Product Dialog
 - Bringing up the Dialog
 - Dialog Box Owner
 - Displaying the Dialog
 - Communication with Parent
 - XAML for Modeless Dialog
 - Handler for the Apply Button
 - Handler for the Close Button
 - Instances of a Modeless Dialog
 - Checking for an Instance
- 6. Menus and Commands**
 - Menus in WPF
 - Menu Controls
 - A Simple Menu
 - The Menu Using XAML
 - Handling the Click Event
 - The Menu Using Procedural
Code
 - Icons in Menus
 - Context Menu
 - XAML for Context Menu
 - Separator

- Keyboard Shortcuts
- WPF Command Architecture
- Command Bindings
- Custom Commands
- MenuCalculator Command Bindings
- Input Bindings
- Menu Items
- Running MenuCalculator
- Checking Menu Items
- Common Event Handlers
- Menu Checking Logic
- Calculation Logic
- Automatic Checking
- 7. Toolbars and Status Bars**
 - Toolbars in WPF
 - XAML for Toolbars
 - Commands and Events
 - Images on Buttons
 - Tool Tips
 - Other Elements on Toolbars
 - Status Bars
- 8. Dependency Properties and Routed Events**
 - Dependency Properties
 - Change Notification
 - Property Value Inheritance
 - Support for Multiple Providers
 - Logical Trees
 - Visual Trees
 - Routed Events
 - Event Handlers
 - Routing Strategies
 - Ready-made Routed Events in WPF
- 9. Resources**
 - Resources in .NET
 - Resources in WPF
 - Binary Resources
 - Loose Files as Resources
 - Logical Resources
 - Logical Resources in Code
 - Static Resources
 - Dynamic Resources
- 10. Data Binding**
 - What is Data Binding?
 - Binding in Procedural Code
 - Binding in XAML
 - Binding to Plain .NET Properties
 - Binding to a Collection
 - Controlling the Selected Item
 - Data Context
 - Data Templates
 - Specifying a Data Template
 - Value Converters
 - Using a Value Converter in XAML
 - Collection Views
 - Sorting
 - Grouping
 - Filtering
 - Collection Views in XAML
 - Data Providers
 - ObjectDataProvider
 - XmlDataProvider
 - Data Access with Visual Studio 2010
 - SmallPub Database
 - ADO.NET Entity Framework
 - Navigation Code
 - DataGrid Control
 - Editing the Book Table
 - Class Library
 - Database Updates
 - Refreshing the DataGrid
- 11. Styles, Templates, Skins and Themes**
 - WPF and Interfaces
 - Styles
 - Style Definition
 - Applying Styles
 - Style Inheritance
 - Style Overriding
 - Sharing Styles
 - Typed Styles
 - Triggers
 - Multiple Conditions
 - Validation

- Templates
 - Improving the Template
 - Templated Parent's Properties
 - Respecting Visual States
 - Using Templates with Styles
- Skins
 - Changing Skins
- Themes

12. **WPF and Windows Forms**

Interoperation

- Interoperating with Windows Forms
 - Add a Form to a WPF Application
 - Add a WPF Window to a Windows Forms Application
- Mixing WPF and Windows Forms in the Same Window
 - Hosting a Windows Forms Control
 - Using Code
 - WindowsFormsHost via Code
 - Windows Forms MonthCalendar
 - WindowsFormsHost via XAML