Practical WPF

Learn by Working Professionals

WPF Course Division

Day 1

- WPF prerequisite
- What is WPF
- WPF XAML System
- WPF trees
- WPF Properties

Day 2
- Common WPF Controls
- WPF Command System
- WPF Event System
- WPF Layout System

Day 3

- Data Binding
- MVVM
- Dispatcher
- Resources

Day 4
- Exercise
- Tools

WPF prerequisite

Client Based Development

Wpf is a client based technology so if a student/professional is not aware of the client side technologies than it is very hard for her to visualize and think in terms of client code and merely getting training in WPF or similar technologies will not help her to fully utilize her thinking / learned concepts.

So our institute identifies this GAP and prepare our training course keep this in mind and dedicate certain hours of whole training towards trying to fill this GAP. Here are the topics which will be covered in this and this can be totally skipped if person attending training is aware of it.

- What is client ?
- Role of client in the application eco system
- Client based technologies
- Difference between Rich Client / Thin Client / Thick Client
- Difference between Web / Mobile / Windows based clients
- Where does WPF lies ?

What is WPF?

Here are the details which will be covered under this after attending this student/professional will be able to understand WPF motive and applicability.

- Definition and Origin Of WPF
- How WPF is different?
- When actually , you should use WPF
- WPF Object hierarchy
- Root object of WPF

WPF XAML System

WPF introduces new declarative way of defining UI elements so it very important to understand how xaml has changed the way we design our user interfaces. In this class , student / professional will be able understand following concepts.

- Declarative System For UI Elements
- XAML Declaration and
- How XAML turned into actual visual UI element?
- List of things which can be done in XAML
- Demonstration of Real life design pattern used
- Difference between xaml and baml.

WPF trees

WPF introduces declarative way of defining User interfaces . This declarative way is actually a hierarchical view which consist of trees. These trees are very important to understand certain WPF concepts like handling of routed events. After attending this class student/professional can understand below concepts

- Why Tree based structure
- Difference between various trees (Logical and Visual)
- Why 2 trees (Logical and Visual)
- Why tree traversal is needed in WPF?
- When to use Logical and Visual tree?
- Walking and Printing Various trees

WPF Properties

WPF introduces new way of defining properties which are different than usual property declaration in .NET. This new concept of WPF properties has deep root in the practical business applications. After attending this class student/professional can understand below concepts

- What are dependency properties?
- Why do we actually need dependency properties?
- Property Inheritance With Example
- In Built Property Change Notification with examples
- Defining and creating own dependency properties.
- How dependency properties are actually different than normal properties with practical implementation with examples.
- What are attached properties?
- Why we need attach properties ?
- Difference between attached and dependency properties
- Defining and creating own attached properties.
- Design pattern used in attached properties
- Converting normal .NET property into Dependency Property
- Practical Examples and Exercise of Both Attached and dependency properties.

WPF Event System

WPF introduces new way of handling user input which changes the way various user input events are handled After attending this class student/professional can understand below concepts

- What are WPF Routed Events?
- Why WPF needed routed events?
- Routing Strategy
- Example showing practical routed events in action
- How routed events are handled ?
- Creating custom routing event

WPF Command System

WPF introduces loosely coupled way of handling user input which helps in maintainable code to be written .After attending this class student/professional can understand below concepts

- What are WPF Routed Command?
- Why WPF needed routed command?
- Difference between routed command and routed events
- How command helps in the maintainable code?
- Creating custom routing command
- New Concept of 'Delegate Command'

WPF Layout System

WPF changes the way how designers /programmer used to layout the UI elements .After attending this class student/professional can understand below concepts

- What is layout?
- Why layout is so important in WPF or any other UI technology?
- Various WPF Commonly used Layout Examples (Grid, Dock, Stack, Canvas, Wrap)
- WPF Control Alignment
- WPF Control Margin

WPF Data Binding

WPF changes the way how control is bind to actual data. WPF Binding feature helps in the separating out UI components with actual model data so that programmer can emphasis on actual business problem rather than writing control data binding code. After attending this class student/professional can understand below concepts

- Introduction to the Data Context
- What is Data binding?
- Data binding benefits
- Data Binding Modes With Examples
- Databinding in XAML With Example
- Value Converters With Example
- Data Validation In Data Binding
- Data binding Errors and Exception
- How to resolve data binding problems.

Introduction To The MVVM (Model-View–ViewModel)Pattern

While developing WPF, Microsoft realize that they have introduced new design pattern which is actually take advantage of WPF data binding feature and this design pattern have power to write testable and maintainable code. After attending this class student/professional can understand below concepts

- Introduction to the MVVM
- How MVVM is different?
- Separate introduction to the Model , View , ViewModel
- Separating out responsibility.
- Converting already written code to the MVVM.
- Introduction to the Behaviour and Triggers from Expression Blend
- Usage of Behaviour and Triggers to strictly attach to the MVVM.
- Introduction to the Observable Collection and Practical example of Using it through MVVM.

WPF Style and Template

Wpf has capability to control how to display user elements through style and Templates .
After attending this class student/professional can understand below concepts

- Introduction to the Style
- Why we needed Style ?
- Examples of Style
- Selecting Style Based On Model Properties
- Introduction to The Template
- Applying Control Template
- Template Binding Through Control Template
- Introduction to The Data template
- Selecting data template based upon the Model Properties

WPF Dispatcher

Wpf dispatcher is a concept to make application responsive in case of long running client process and it has deep root in multithreaded application written in WPF. After attending this class student/professional can understand below concepts

- Introduction to Dispatcher
- Why WPF is in need for Dispatcher.
- Design Pattern implemented in Dispatcher
- Role of dispatcher in WPF
- Using dispatcher in WPF.

WPF Resources

Wpf has new way of managing application wide resources . Resource is a reusable unit which can be referenced in xamls . After attending this class student/professional can understand below concepts

- Introduction to Resources
- Why we need resources.
- Static Vs Dynamic Resources
- Examples Of Static Resource and Dynamic Resources.

Practical WPF Application

We have learnt almost all useful features of WPF now its high time apply these concepts with   real   world   application.   After   attending   this   class   student/professional   can understand below concepts

- Introduction to Exercise
- Dividing Exercise into individual unit of Work (UI , Server, Database)
- Designing of Exercise with learnt WPF concepts
- Implementation Of MVVM.