

PV178: Programming for .NET Framework

Windows Presentation Foundation

Vojtěch Forejt, forejt@fi.muni.cz
Martin Osovský, osovsky@ics.muni.cz

Faculty of Informatics and Institute of Computer Science
Masaryk University

April 16, 2009

Drawbacks of Most GUI Libraries

- UI design and application logic are not separated
- Do not take advantage of current technologies
- Aimed at one particular area (web, client GUI, 2D or 3D graphics)

WPF Overview

- Framework for working with graphics
 - user interface, 2D and 3D
- Separation of UI design and application logic
- Goes beyond WinForms, but does not supersede it
- `Windows.Forms.*` namespaces

- From .NET Framework 3.0, pre-alpha in Mono

Example

- ColorPicker3D

WPF Assemblies and Namespaces

- `WindowsBase.dll`
- `PresentationCore.dll`
- `PresentationFoundation.dll`

- `System.Windows`
- `System.Windows.Controls`
- `System.Windows.Markup`
- `System.Windows.Media`
- `System.Windows.Navigation`
- `System.Windows.Shapes`

XAML

- stands for *eXtensible Application Markup Language*
- Declarative markup language
- Allows to create UI using XML syntax
- Objects – tags; Properties – tags or attributes
 - `<Button Cursor="Hand">Click</Button>`
or
 - `<Button>`
 `<Button.Cursor>Hand</Button.Cursor>`
 Click
 `</Button>`
- Objects are instantiated automatically

Example

- SimpleWpfExample

XAML cont.

- XAML file “defines” a partial class specified by `Class` attribute of root element
- Root element determines the base class
- Application logic is placed in separate file with a partial class, written e.g. in `C#`

XAML – How Does It Work

- XAML is compiled to BAML (Binary Application Markup Language)
- BAML is stored in assembly as a resource
- When the application starts, BAML is read and appropriate objects are created

Visual Tree

- Contains elements that are rendered on the screen
- (Roughly) corresponds to the tree in XAML file

Layouts

- **WrapPanel** – elements positioned from left to right (or from top to bottom, depending on `Orientation`), wrapped at the end of line
- **DockPanel** – child elements positioned to the edges of the panel
- **Grid** – table-like positioning
- **Canvas** – absolute positioning
- **StackPanel** – Positions element in line (horizontally or vertically, depending on `Orientation`)

Example

- WpfLayoutsExample

Controls

- Well-known controls are available
 - TextBox, Button, ListBox, TabControl, ListView, CheckBox, RadioButton,...
- Several new controls
 - Expander
 - PasswordBox

Controls cont.

- Controls contain
 - **Methods**
 - **Properties**
 - **Events** – routed events, extension of ordinary CLR events
 - **Commands** – Cut, Paste,...
- Application logic in C# (or another language)
- Data binding
- Templates

Routed Events

- Extend CLR events
- Invoke handlers on multiple elements of visual tree (“routed” through a tree)
- Routing strategies
 - **Bubbling** – event handlers are invoked from the source to the root
 - **Direct** – only handler on the source is invoked
 - **Tunelling** – event handlers are invoked from the tree to the source
- Typically, `<event>` is bubbling and is preceded by `Preview<event>`
- Setting `Handled` property to true prevents invoking all following handlers

Example

- RoutedEventsExample

Commands

- Counterpart of Actions in Java
- Represent high-level tasks (Cut, Copy, Paste...) that can be executed by or on different objects.
- Extend RoutedCommand class
 - Methods CanExecute and Execute
 -
- **Command source** – control executing the command
- **Command target** – control on which the command is executed
- **CommandBinding** class – associates a command with event handlers
 - events CanExecute and Execute

Example

- WpfCommandExample

Data Binding

- Connects UI with application logic
- Synchronization between data and its visual representation
- **Binding target** – (almost) any property of a control
- **Binding source** – property of a control, ADO.NET, XML, collection

Example

- WpfDataBindingExample

Example

- ColorPicker3D