

# XAML for the Winforms Developer

**Philip Japikse (@skimedic)**

**Principal Architect, InterKnowlogy**

Level: Intermediate



TECH EVENTS WITH PERSPECTIVE

**IT AND DEVELOPER TRAINING THAT'S OUT OF THIS WORLD**

Visual Studio 

SharePoint 

SQL Server 

ModernApps 

**TECHMENTOR**

# PHIL.ABOUT()

- Principal Architect, InterKnowlogy
  - <http://interknowlogy.com>
- Microsoft MVP, MCSD, MCDBA, CSM, CSP
- Lead Director, Cincinnati .NET User's Group
- Co-host Hallway Conversations Podcast
  - [www.hallwayconversations.com](http://www.hallwayconversations.com)
- Founder, Agile Conferences, Inc.
  - [www.dayofagile.org](http://www.dayofagile.org)

# WHAT IS WPF?

- Windows Presentation Foundation
- Next Generation Rich User Application Platform
  - Declarative UI (XAML)
- Based on DirectX (not GDI/GDI+)
  - Vector Based
  - Supports true Hardware Acceleration
  - Resolution Independence
- Leaps ahead with binding
  - Data and other Elements

## WHAT ELSE IS IN WPF?

- Flexible flow layout model
- Styles and Templates
- OOTB document handling features
- True animation – not timer based
- A/V support (through WMP)
- Commands
- Routed Events

# WHAT'S NEW IN .NET 4?

- DataGrid, DatePicker, Calendar
- Visual State Manager
- Layout Rounding
- Multi-Touch support
- Custom Dictionaries
- Behaviors\*
- Animation Easing
- Binding String Format

# WHAT'S NEW IN .NET 4.5

- Ribbon Control
- Improved Performance with large data
- Binding to Static Properties
- Accessing Collections on non-UI Threads
- Asynchronous Validation
- Delay Updating Data Source
- Retrieving Data Binding Info from Expression
- And more...
  - <http://tinyurl.com/wpf4-5>

# WHAT'S MISSING

- WPF – What's missing
  - *Visual* Inheritance
  - MDI – complete lack of support

# AGENDA

- Elements, Controls, and More
- Layout/Layout Containers
- Commands
- Custom Dictionaries
- Behaviors
- Binding



ELEMENTS, CONTROLS, AND MORE

# ELEMENTS & CONTROLS

- Everything is an Element
  - Controls are elements that can:
    - Receive Focus
    - Accept User Input
- Content Controls
  - Hold a single element
    - Window (Special)
    - Labels, Buttons, CheckBox, RadioButton, ToolTips
    - ScrollViewer, Border, Expander
- Text Controls
  - TextBox , PasswordBox – Strings Only
  - RichTextBox – Sophisticated Content – FlowDocument

# LIST CONTROLS

- List Controls (ItemsControl)
  - Selectors
    - ListBox, ComboBox, TabControl
  - Non-Selectors
    - Menus, ListView, GridView, TreeView, ToolBar
- Range Based Controls
  - ScrollBar, ProgressBar, Slider

# EVENTS

- Routed Events replace usual .NET events
  - Direct
    - Originate in one element and don't pass on
  - Bubbling
    - Travel up the element tree
  - Tunneling
    - Travel down the element tree
- Can add event handlers for child elements
  - `<Grid Button.Click="ClickEvent" ...`

# MENUS

- Use “\_” instead of “&” for Hot-Key
- Beware for XAML order vs Grid Order
- Similar to WinForms except
  - MenuItem is a container control
  - Use <Separator> instead of “-”

# LAYOUTS AND LAYOUT CONTAINERS

# LAYOUT

- “Web”-like layout model with Flexible Flow
  - Gone are explicit sizing, absolute positioning, etc
  - Containers try to allocate element’s requested space
- Layout Process
  - Measure
    - Loop through all elements asking for preferred size
  - Arrange
    - Container places elements in appropriate position

# LAYOUT CONTAINERS

- Standard
  - Grid
    - Most commonly used (default Window Panel)
  - StackPanel, WrapPanel, DockPanel
    - Used for UI Tweaks
  - Canvas
    - Absolute (think WinForms) positioning
    - InkCanvas - Supports Digital Ink
- Specialized
  - UniformGrid, TabPanel, ToolBarPanel, ToolBarOverflowPanel



# ATTACHED PROPERTIES

- Properties defined in one element used in another
  - Grid.Row/Grid.Column
- Translated into method calls
  - Grid.SetRow(control,x)

# DEMO

Layouts

# COMMANDS

# COMMANDS

- WPF Command Model adds:
  - Delegates to the Appropriate Commands
  - Keeps state of control tied into state of Command
    - Can\_Execute
- Consists of:
  - Commands – represents an application task
  - Command Bindings – links command to code
  - Command Sources – trigger the command
  - Command Target – element on which the command is being performed

# DEMO

Commands

# CUSTOM DICTIONARIES

# CUSTOM DICTIONARIES

- Enable by setting `SpellCheck.IsEnabled="True"`
- Text File (<name>.lex) for custom words
- Add as a resource, assign to the `CustomDictionaries` collection, copy to output directory
- Four languages supported
  - English, Spanish, German, French

# DEMO

Custom Dictionaries



BEHAVIORS

# BEHAVIORS

- Introduced in Expression Blend v3
  - Encapsulate functionality into reusable components
    - Drag & Drop
    - Pan and Zoom
    - Input Validation
    - Watermark Text
    - InvokeCommand
  - Additional Behaviors are available from the Expression Gallery
    - <http://msdn.microsoft.com/en-us/expression/jj873995>
-

# DEMO

Behaviors

BINDING

# ELEMENT BINDING

- Allows for powerful UI constructs all defined in XAML
- Can bind elements value directly to another element
  - ▶ No more coding “onChanged” events!

# DATA BINDING

- Supports
  - IDataErrorInfo
  - INotifyPropertyChanged
  - INotifyCollectionChanged
- Can bind to \*any\* Dependency Property
- Choice of
  - Datasets
  - Data Objects

# DEMO

Data Binding

OBSERVABLES



# OBSERVABLES

- Models
  - Leverage INotifyPropertyChanged
    - Beware of magic strings
- Collections
  - Leverage ObservableCollection
    - Implements
      - INotifyCollectionChanged
      - INotifyPropertyChanged

# WHERE TO IMPLEMENT?

- Anemic Model
  - Implemented in ViewModel
  - “Cleaner” model
  - Lots of code duplication
- Anemic ViewModel
  - Implemented in Model
  - Less code duplication
  - Mixing of concerns?

VALIDATION

---

# VALIDATION METHODS

- Raise an error on your data object
  - Normally errors are ignored on bindings
  - Set `NotifyOnValidationError`
- Use `INotifyDataErrorInfo` or `IDataErrorInfo`
  - `INotifyDataErrorInfo` is new in WPF 4.5
- Define validation at the binding level
- Note: Validation only occurs with `TwoWay` or `OneWayToSource` binding modes

## VALIDATESONEXCEPTION

- Raises exceptions from bound property
- Sets `Validation.HasError = True`
- Creates `ValidationError` object
- If `NotifyOnValidationError = true`
  - WPF raises `Validation.Error`

## IDataErrorInfo

- Add a string indexer for the Item property
- Error property is not used by WPF
- Relies on INotifyPropertyChanged
- Requires ValidatesOnDataErrors in binding

# INOTIFYDATAERRORINFO

- New in WPF 4.5
- Contains
  - ErrorsChanged event
  - HasError Boolean
  - GetErrors() method to return the errors
- Relies on INotifyPropertyChanged

# CUSTOM VALIDATION RULES

- Derive from `ValidationRule`
- Add additional properties for custom configuration
- Override `Validate()`
  - Return new `ValidationResult(true || false, [Error Message])`
- Add to `Binding.ValidationRules` collection



# DEMO

Validation

QUESTIONS?



## CONTACT ME

- [skimedic@gmail.com](mailto:skimedic@gmail.com)
- [www.skimedic.com/blog](http://www.skimedic.com/blog)
- [www.twitter.com/skimedic](http://www.twitter.com/skimedic)
- [www.hallwayconversations.com](http://www.hallwayconversations.com)
  
- [www.about.me/skimedic](http://www.about.me/skimedic)

Thank  
You!