

# FROM WINDOWS FORMS TO WPF WITH MVVM

Reed Copsey, Jr.

C Tech Development Corporation

Blog - <http://reedcopsey.com>

Twitter - <http://twitter.com/ReedCopsey>

# Graphical User Interfaces

- ⦿ Exists for Usability
- ⦿ User Experience is Entire Purpose
- ⦿ Works on data
  - Display
  - Editing

# Developing a GUI: The Model

- Domain Specific Data and Logic
- Core of any program
- Should be unaware of User Interface

DEMO:

THE MODEL

# Windows Forms

- ◎ Based on wrapping Windows API
  - Each control has separate Window Handle
  - Event Driven API
- ◎ Original Client GUI Technology for .NET
  - Basically unchanged since .NET 2.0
  - Minor changes since .NET 1.1
- ◎ Familiar API

DEMO:

APPLICATION IN  
WINDOWS FORMS

# Windows Forms: Problems

- ⦿ Customization is tricky
  - Custom look and feel requires custom control or manual drawing
- ⦿ Custom drawn via GDI+
  - Resolution dependence
- ⦿ Performance/Resource Usage
- ⦿ No new Development after 2005

# Windows Presentation Foundation (WPF)

- ⦿ Released in .NET 3.0, Improved in .NET 3.5
  - Actively developed - .NET 4 features many improvements
- ⦿ Easy customization
- ⦿ Better Developer and Design Story
- ⦿ Extensive Framework
  - 2D and 3D graphics, media support, etc.

# Windows Presentation Foundation (WPF)

- ⦿ Can be used like Windows Forms
  - Event-Driven API still works
- ⦿ Required Learning
  - Different Designer
  - Changes in Layout System

DEMO:

APPLICATION IN WINDOWS  
PRESENTATION FOUNDATION

# Don't do this!

- ⦿ Event Driven API has issues
  - Leads to tight coupling of logic to user interface
  - Spaghetti code
- ⦿ Better alternatives, even in WinForms
  - MVC
  - MVP
  - User Interface Process Application Block
    - 134 Page Guidance on avoiding spaghetti

# DataBinding in WPF

- ◎ Effective DataBinding Requires:
  - Simple to bind user interface element to data source
  - Clean way to specify data source for a collection of objects
  - Consistency in binding specification
  - Flexibility in what triggers updates
  - Validation
  - Conversion in binding itself

# DataContext

- ⦿ Can be any object of any type
  - INotifyPropertyChanged
  - INotifyCollectionChanged
- ⦿ Propagates down Visual Tree
  - Allows all controls in Window to share DataContext automatically

DEMO:

**DATABINDING**

# Templating

- ◎ Control Behavior separated from Visual Representation
  - Control “look” can be changed without changing code
  - Allows for improved designability
- ◎ DataTemplate
  - Provide custom “look” for any type
  - Basic means of customization

DEMO:

TEMPLATING

# Commands

- ⦿ ICommand
  - Execute: Executes the command
  - CanExecute: Returns boolean
- ⦿ Can implement using delegates
- ⦿ Invoked via DataBinding from multiple Sources
  - Buttons
  - Gestures
  - MenuItems

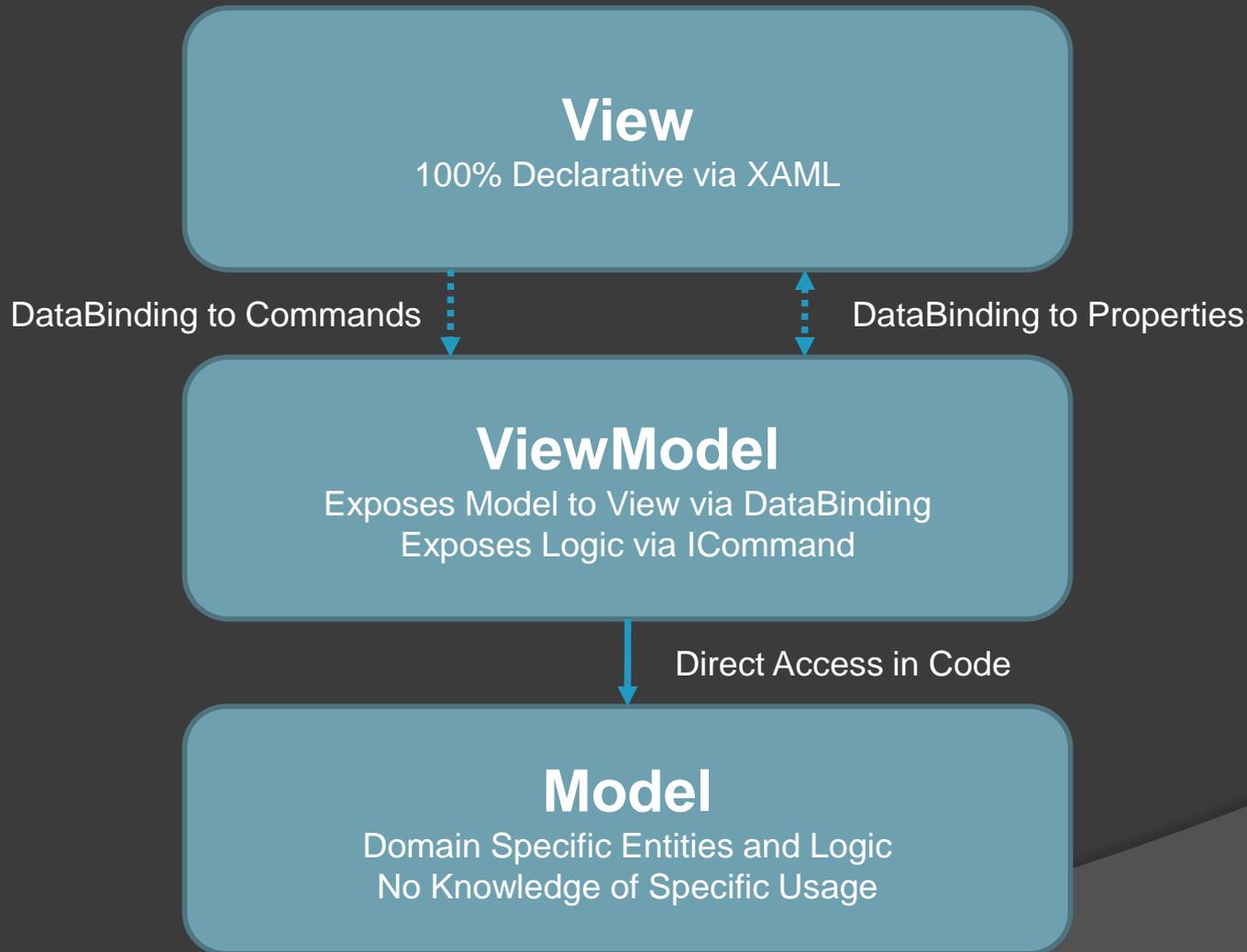
DEMO:

COMMANDS

# Model-View-ViewModel

- ⦿ Architectural Pattern
- ⦿ Model
  - Domain Specific Data and Logic
  - Isolated
- ⦿ ViewModel
  - Works with Model, exposing it for DataBinding
  - Manages Application Specific Work
- ⦿ View
  - Ideally 100% XAML
  - “Hooks” into ViewModel via DataBinding

# Model-View-ViewModel



# MVVM Advantages

- ① Flexible View – Redesign without code changes
- ① Testability
  - Easy testing in ViewModel
- ① Clean Separation of Concerns

DEMO:

APPLICATION USING  
MODEL-VIEW-VIEWMODEL

# Online Resources

- <http://WindowsClient.net>
  - Microsoft Windows Client Resource Center
- Prism: Composite Application Guidance for WPF and Silverlight
- MVVM Frameworks
- Blog and Contact: <http://reedcopsey.com>