

Data Binding Essentials

Kevin Dockx

<http://blog.kevindockx.com/>

@KevinDockx



pluralsight 
hardcore dev and IT training

UI



Sync



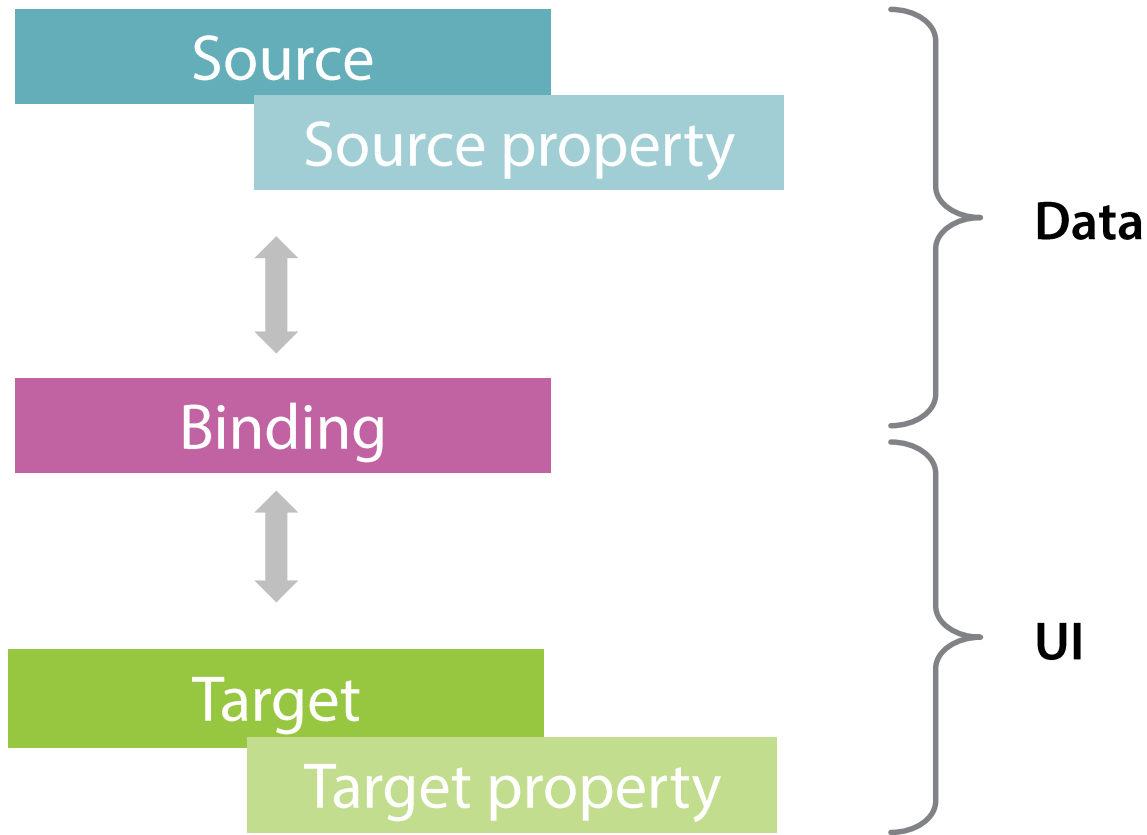
Data Object



In essence, data binding is a way to keep your model and UI in sync

- **It results in less repetitive code, less error-prone, better decoupling & reuse, easier testing through design patterns**

Source, Target and Binding Explained

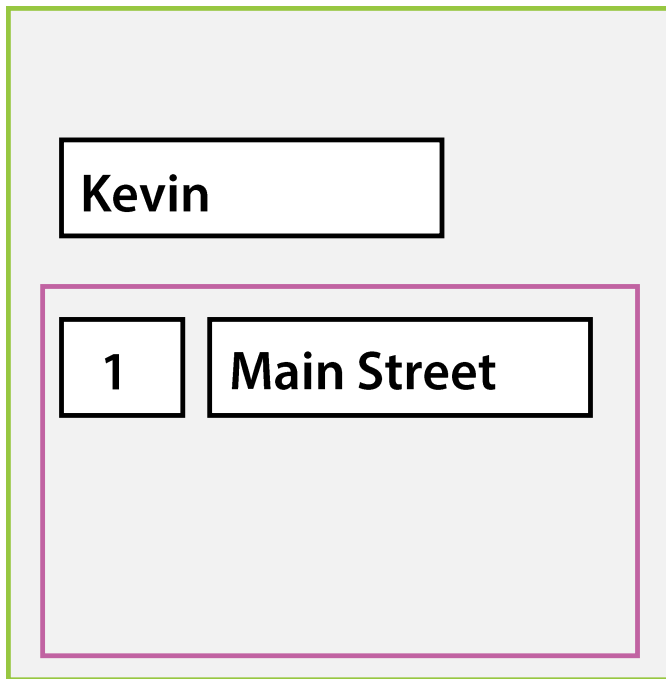


**Target property must
be a dependency
property**

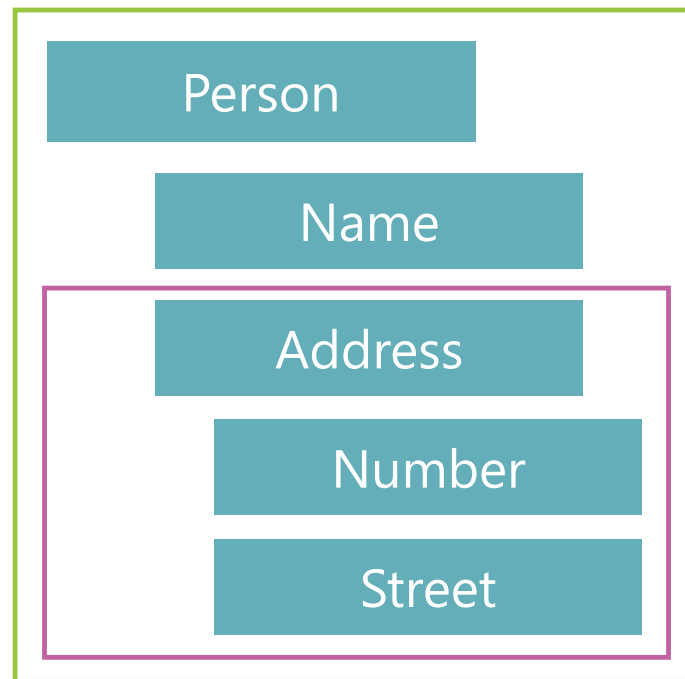
On DataContext and Scope

The DataContext property is defined on FrameworkElement. It's the data source used for data binding

UI



Data Object



Binding Modes

The binding mode allows us to specify how we want to sync between source and target

Different
binding
modes

- **OneWay:** changes to the source property update the target property, but not the other way around
- **TwoWay:** changes to source or target properties to automatically update the other
- **OneWayToSource:** only update the source property when the target property changes
- **OneTime:** a OneWay binding that only propagates once: to initialize the target property

Notifying the UI

The UI isn't auto-notified of a change on a backing class

... for good reasons

- Performance
- Dependent notifications

To allow reporting change

- Implement **INotifyPropertyChanged** on the model
- Raise the **PropertyChanged** event to notify the UI

Controlling Sync Timing: UpdateSourceTrigger

For binding modes that update the source, we can specify the timing of these updates with UpdateSourceTrigger

TextBox-based controls

- **Default:** when the focus is lost
- **PropertyChanged:** on each keystroke
- **Explicit:** when UpdateSource() is called from code

Working With Converters

A value converter allows us to provide custom logic to a binding to convert from source to target type (and back), so we can bind between properties that have incompatible types

To use a converter

- Create a class implementing **IValueConverter**
- Implement **Convert**, and optionally **ConvertBack**
- Assign an instance of the class to the **Converter** property of the Binding markup extension

Summary

- **Source, Target & the Binding markup extension**
- **DataContext and scope**
- **Binding Modes**
 - How to sync
 - Notifying the UI
- **Control synchronization timing**
- **Converters**
 - Binding different property types