Data Sources

Brian Noyes
CTO, Solliance (www.solliance.net)
brian.noyes@solliance.net, @briannoyes

pluralsight
hardcore developer training

Outline

Entity Data Sources

Change Notifications

EditableObjects & CollectionViews

DataSet Data Sources

XML Data Sources

Binding to Entities

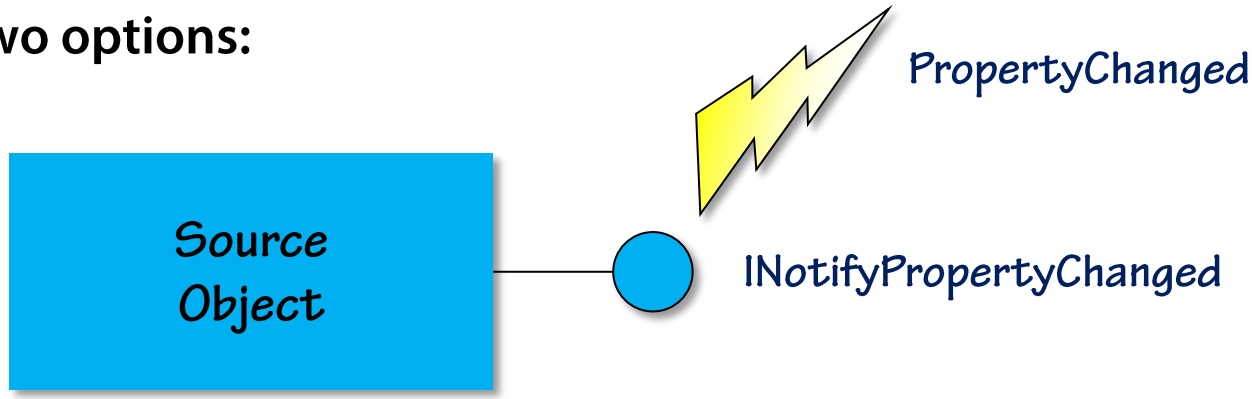
- **Data binding supports Plain Old CLR Objects (POCOs)**
  - Simple class with primitive or complex type properties
  - Cannot bind to member variables
  - Cannot bind to methods
- **Data binding to collections**
  - Object must implement IEnumerable or a derived interface (i.e. ICollection or IList or generic variants)
- **Set DataContext to object exposing properties that are entities or entity collections**
- **Binding just needs to identify Path=property name on the DataContext that it gets its value from**
  - Can be a complex Path (i.e. Customer.Address.Street)

Property / Collection Change Notifications

- **Necessary if source object properties or collections can be modified by any code other than the UI binding**
  - Async data loading
  - Command/event handling code from another control
  - Background threads
- **If no notification, UI will continue to show value that was in source property at initial binding**
- **Need notifications to keep data object and UI in sync**

Supporting Property Change Notifications

- Two options:



INotifyPropertyChanged

- **Implement interface on class**
- **Expose public event**
  - public event PropertyChangedEventHandler PropertyChanged
- **Raise PropertyChanged event in property setters**
  - If property actually changed value

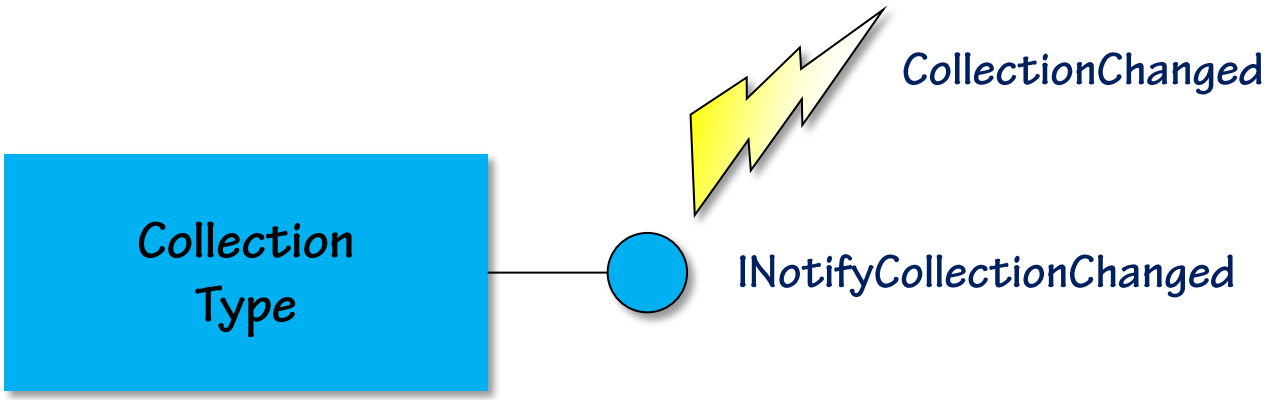
DependencyProperties

- **Can be implemented on any class derived from DependencyObject**
- **Can also be implemented as static Attached Properties**
  - Not directly relevant to instances of data source objects and change notifications
- **Raise change notification internally that Bindings monitor**
- **Typically only used for properties exposed from UI elements, controls and views**
- **Targets of Bindings must be DependencyProperties**

DependencyProperties

- **Declared as a public static readonly member variable of type DependencyProperty**
- **Conventions for naming**
- **Declared with "wrapper property" – normal .NET property that calls GetValue/SetValue on base class for programmatic access**
  - Not used by bindings at all – Bindings call GetValue/SetValue directly on DependencyObject base class
  - Required as part of DependencyProperty declaration
- **Has optional metadata associated with it to define:**
  - Default value
  - Default data binding directionality
  - Other WPF framework aspects

ObservableCollection<T>



- **Collection Changed only fires for add/remove/replace/clear**
  - Not for individual object changes
- **ObservableCollection<T> predefined class for this**
- **In general – any modifiable collection you are going to bind to should be ObservableCollection<T>**

IEditableObject Support

- **May want objects that can back out changes to multiple properties**
- **May need to execute custom logic to do that even when presented in DataGrid**
- **IEditableObject lets you support this**
- **API:**
  - BeginEdit – Save “original values” somewhere
  - EndEdit – Discard “original values”
  - CancelEdit – Overwrite current values with original values

Managing the "Current" Object

- **When you bind to a collection, WPF wraps the collection in an ICollectionView implementation**
- **ICollectionView manages the concept of which object in a data bound collection is the "Current" object**
  - Used when data source is collection, but Binding expects a single object property – i.e. TextBox.Text
- **Can obtain the Current object reference, control the index of which item is current through ICollectionView**
- **ICollectionView exposes other capabilities as well**
  - Sorting, Filtering, Paging

Binding to DataSets

- **DataSets are fading in popularity**
  - Untyped DataSets required fragile untyped code
  - Typed DataSets were a poor-man's entity
  - Can't benefit from inheritance, polymorphism
  - Encapsulation is clunky
- **Still useful at times for dynamic queries and result sets**
- **May have legacy code you need to support**
- **WPF data binding works fine with DataSets**
  - Both untyped and typed data sets
  - But not later XAML stacks (Silverlight, Phone, WinRT)

Binding to XML

- **Can data bind to values in XML elements or attributes**
- **Generally should avoid from architecture perspective**
  - UI should not be coupled to persistence mechanisms
- **Use XmlDataProvider to point to document**
- **Use XPath to refine DataContexts to portions of the document**

Summary

Entity and Collections bind easily

Need to support change notifications on objects and collections

Can add functionality w/ IEditableObject & ICollectionView

DataSets and XML can be bound to directly