

View/ViewModel Communication in WPF



Brian Noyes

@briannoyes | www.solliance.net

Overview



Commands

Attached Properties / Behaviors

Property Change Notifications

Commands

Based on classic Command design pattern

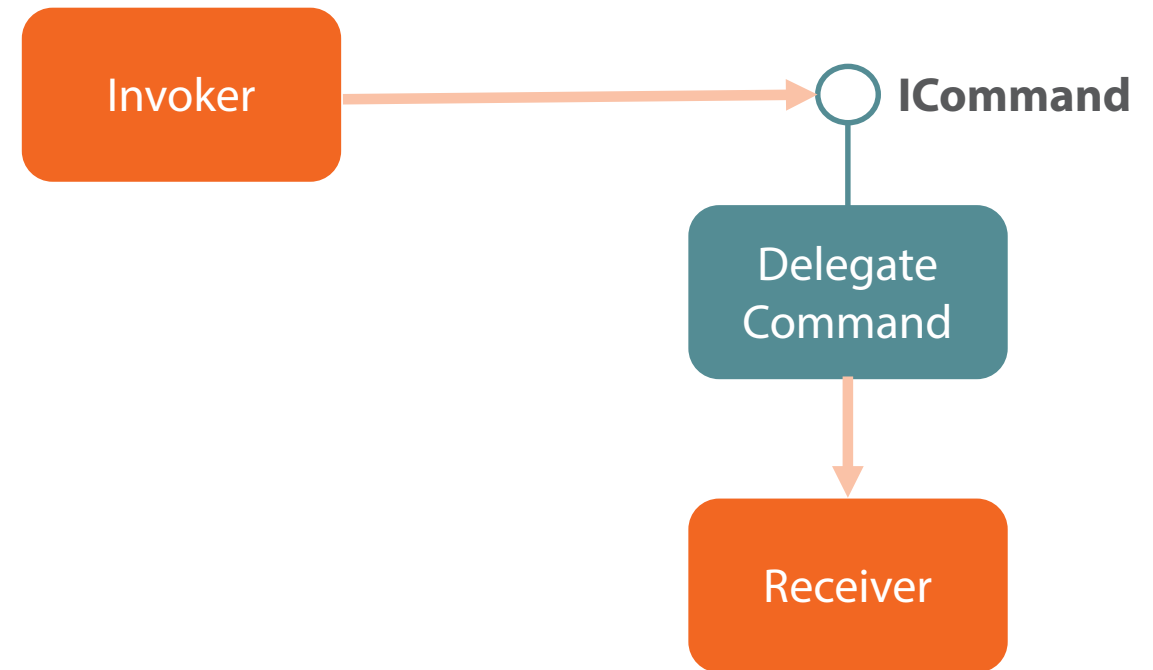
Invoker – View control

Receiver - ViewModel

Use delegating command implementation

Supports decoupled command handling invocation

Supports enabling/disabling associated control



Attached Properties / Behaviors

- Attached Properties form the basis for Behaviors
- Behaviors can be defined directly with Attached Properties or with Blend SDK
- Blend SDK is preferred way
 - Uses Attached Properties itself to attach Blend SDK-based Behaviors to elements
- Behaviors can form a communication bridge between View and ViewModel
 - Events / property changes in View trigger commands or method calls into ViewModel
 - Behavior in View can listen for events / property changes from ViewModel and modify UI accordingly

Property Change Notifications

Need to trigger bindings to refresh as properties change

Two options:

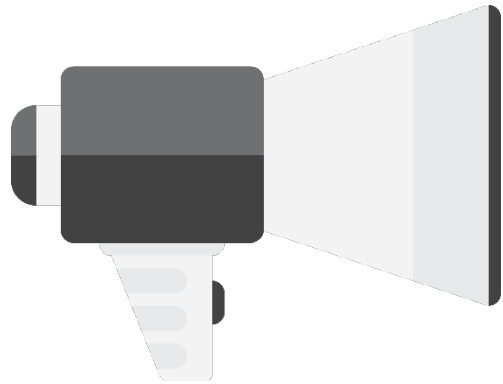
DependencyProperties

INotifyPropertyChanged (INPC)

INotifyPropertyChanged more appropriate for ViewModels and Models



Summary



Commands are primary form of View -> ViewModel communication

Behaviors make it easy to bridge the gap between View events and changes and ViewModel logic

PropertyChange notifications keep your Views in sync