## Exercise 6 – Commands

* What you'll learn
  + How to bind a Button to a Command object
* Stuff to Know
  + Command property should be of type ICommand, created as DelegateCommand instance
  + To display a message dialog -- MessageBox.Show("Message")
  + CanExecute method returns a boolean
* Best Practices
  + Use commands, rather than adding code-behind in View

### The Exercise

* Start from existing project: Exercise\_6\_Commands
  + This project includes a DelegateCommand implementation
* Steps
  + ViewModel
    - Create a MyCommand property in the ViewModel, using the DelegateCommand
    - Set the command's execute method to create a MessageBox with a message
    - Set the command's CanExecute method to always execute
  + View
    - Create a Button and bind its Command property to the MyCommand property
    - Run/test => Clicking on the button, the dialog should show up
  + CanExecute
    - Add a CanMessage bool property to your ViewModel
    - Add a CheckBox control to your view, binding IsChecked to the new property
    - Modify the CanExecute lambda (passed into the DelegateCommand) so that it depends on the CanMessage property
    - Run/test => How does the Button act when the CheckBox is unchecked vs checked ?