Contents

If you're new to Punch, you've come to the right place.

Belly up to the Happy Hour bar, a Silverlight 5 application that you'll build from scratch. MVVM, convention over configuration, view composition, screen navigation, … they are all introduced here in short, easy-to-follow lessons that build progressively from a blank solution to a plausible application.

This series gets you started with Punch, guiding you towards proven architecture and practices that will sustain your application over time. We show you how … and we explain why.

Getting started

Download Punch, install the tools, and get ready to start developing XAML-based business apps.

Bare bones MVVM

Build a simple app with a single view. Migrate the supporting view logic out of the code-behind and into a homebrew *ViewModel*.

An MVVM Punch

Take that same app and introduce it to Punch. Boilerplate handwritten code melts away as Punch conventions do the wiring for you. Maybe this MVVM thing isn't so complicated.

Conventions and diagnostics

Conventions seem like magic. What happens when the magic fails? We learn more about conventions and discover that Punch is logging to the output window as the conventions are applied. Knowing how to interpret these logs will help you diagnose problems.

Data in a ListBox

We create a model with a single entity, a *DrinkOrder*, representing a tasty beverage ordered from the bar. Each ordered drink appears in a *ListBox*, its values bound to controls in an *ItemTemplate* in the same conventional manner we've seen so far.

Entity Views

We discard the *ItemTemplate* in favor of *View* class. *DrinkOrder* serves as its own *ViewModel*. We add a new convention so Punch can associate a *View* with an entity in the *Model*. In fact, a *ViewModel* can have multiple *View*s.

Talk to the View

View/ViewModel data binding isn't always the best way to trigger a behavior in the *View*. When the binding logic becomes too convoluted and difficult to follow, it's better to switch to a different strategy and let the *ViewModel* talk directly to the view.

We'll revive the View code-behind and teach the ViewModel to call into the View via an interface.

Images and ValueConverters

Modern application data includes pictures. In XAML you need a *ValueConverter* to translate from picture representation in data (typically as a source string) into images on screen. We see how Cocktail conventions can embed customizable image *ValueConverter*s in the bindings.

TempHire reference app

After you've finished Happy Hour and understand the core design patterns associated with Punch-built apps, you'll be ready for the Punch reference application: TempHire.

TempHire is a reference architecture designed for medium-to-large business applications. It demonstrates a range of Punch capabilities including a Code First entity model, repository patterns, the Model-View-ViewModel (MVVM) presentation style, view composition, screen navigation, data-binding, validation, multiple editing contexts, multiple data stores, security, testing, and much more.

Watch the TempHire videos for a detailed walkthrough—provided by Senior Director of Professional Services and lead developer of Punch, Marcel Good, and V.P. of Technology, Ward Bell.