# Silverlight Demo Script

## In Browser Site Demos

* [www.silveos.com](http://www.silveos.com)
  + Not an “app” as such, but shows the highly interactive capabilities of SL
  + Press F11 to go full screen
  + Show Solitaire – highly interactive capabilities of SL UI’s (really hard on the web)
  + Show web browser – go to Synergex home page – host web content inside SL apps
  + Show YouTube viewer – search Synergy SPC – multi-media capabilities
* <http://demos.telerik.com/silverlight/salesdashboard>
  + This one’s a demo site for one of the control vendors
  + Great job of presenting a business dashboard using SL
  + Click different sales people to show data – animations and transitions
  + Click filter boxes at top – fast and natural response – running a LOCAL app
  + Notice the grid control – looks and feels like a Windows grid, responsive
    - Click “View All” in tree”
    - Sort by required date
    - Group by customer
    - Filter for USA
* <http://www.perpetuumsoft.com/Demo/slviewer/default.aspx>
  + Another business – related demo, reporting (always a challenge with web solutions)
    - Again, it’s a vendor demo, but nicely done.
  + Run the Orders report, scroll around
  + Click the page width button (left of the hand) – zooming – again, difficult to implement on the web
  + Click print button. Web browsers tend to download a document, or popup a browser, SL can interact with my local printer, so no browser headers and footers on my report.
* <http://gis.hudson.oh.us/workorders>
  + Real live app ... City of Hudson, OH ... extranet app re city work orders
  + Work Order Categories tab – select Active Work Orders
  + Zoom out – hover over icons to see detail.
  + While hovering over – click Zoom To
  + Center top, change BaseMap to 2009 Aerials
  + Top-right, click Toggle Full Screen – app has some control over local browser

## Out of Browser Site Demos

* <http://tasktimer.labee.nl/#/Start>

## My Silverlight Demo

* Open C:\SPC2010\_STEVE\13\_SILVERLIGHT\SilverlightDemo.sln
  + Briefly run the demo and show what it does
  + Created using the “Silverlight Application” project template in VS2010
    - Default look and feel, navigation, etc. Great way to get started.
* Two development projects
  + SilverlightDemo SL application
  + SilverlightDemo.Web Hosting web site (ASP.NET)
* Web Site
  + Single page (default.aspx) which hosts the SL app – page generated automatically
  + Various services (SL demo uses the WCF service)
  + Reference to SynergyComponent (same one used in MDU session) – back to xfServerPlus
  + Services basically expose the various Synergy methods to remote clients
    - Show GetCustomers in the WCF service – just a thin wrapper for the Synergy method.
    - Doesn’t expose Synergy data classes directly, more fields than needed, could have done this with alternate repository structures also for better performance.
    - Classes exposed to remote clients are also in project (look at Address).
  + Same services also used by other demos you’ll see later in the SPC.
* Silverlight app
  + “Service reference” to WCF service – generates LOCAL wrapper classes in the SL client, makes coding with the WCF service transparent.
  + MainPage.xaml
    - UserControl representing main “screen” of app
    - Has “links” to other user controls in the application
    - Hosts those other UserControls within the main “Frame”
  + SL APP uses the MVVM model – not required, but makes a lot of sense.
  + ViewModels folder – CustomersViewModel
    - Pretty much all the logic for the application is in here
    - Implements INotifyPropertyChanged interface – key to signaling changes to the view
    - Exposes data for a View to bind to
    - Exposes commands for a View to execute
    - Not going to go into a lot more detail, because Richard is covering this concept extensively.
  + Views folder – Customers.xaml
    - Overall layout uses a 2 row by 3 column grid
    - Left column has a DataGrid bound to customers
    - Center column has a DataGrid bound to contacts
    - Right column has a DataForm
    - Things to point out
      * Customers grid
        + <Navigation:Page.Resources> Instantiates a “CustomersViewModel” as vm
        + Main layout grid binds DataContext to vm
        + Customers grid ItemsSource bound to Path=Customers (property within DataContext)
        + Grid column bound to CustomerName (within grids ItemsSource)
        + Trigger on SelectionChanged hooked to SelectedCustomerChanged command in VM
      * Contacts grid
        + basically does the same thing, but with contacts
      * Data Form
        + Bound to SelectedContact in the VM
        + BeginningEdit command bound to StartEditingContact command in VM to ensure latest data and get current GRFA
        + EditEnded command bound to SaveContact command in VM to save the data.
      * What’s the point of all of this “declarative” programming?
        + In a WinForms app, or an ASP.NET web app there would be TONS of code behind to implement the client
        + Look at the Code behind – NOTHING!

All the code is in the VM

A new client could be easily built around the existing VM

Re-use!