Agenda

- Overview of the ArcGIS Viewer for Silverlight
- Extensibility endpoints
  - Tools, Behaviors, Layouts, Controls
- Add-in walkthroughs
  - Query Related Records, Bookmarks
- Developer resources
- Road ahead

- Offering ID=267 – http://esriurl.com/survey
ArcGIS Viewer for Silverlight overview

- Configurable *ready-to-deploy* Web client
  - Built with ArcGIS API for Silverlight
- **Easily and quickly** create and implement GIS Web mapping applications

- Set of core tools and functionality
- **No programming or editing configuration files**
  - Ideal for novice Web application creators
Application Builder contains Viewer App

App Builder
layer management

Web App
table of contents

Web App preview
Add-Ins – Extending the Viewer

• The Viewer does the repetitive stuff
  - Map configuration
  - Overall app architecture
  - Generic tools
  - Generic app appearance (title, colors, logo)

• Add-ins enable specific workflows and branding
  - “Add-in” → code module containing functionality
    - Compiled as Silverlight packages (.xap files)
  - Seamlessly plug-in to Viewer and Builder
    - Encourages reusability
Add-Ins – Extending the Viewer

• Software for developing add-ins
  - •Microsoft Visual Studio 2012 or Microsoft Visual Studio 2010.
  - •ArcGIS API for Microsoft Silverlight 3.1.
  - •Microsoft Expression Blend Preview for Silverlight 5.
  - •Microsoft Silverlight 5 Tools for Visual Studio 2010 SP1 (if using Visual Studio 2010)
  - •Microsoft Silverlight Toolkit
  - **ArcGIS Extensibility SDK for Silverlight 3.1**
    - Visual Studio 2010 and 2012 template provided
Add-Ins – Extensibility Points

Several ways to customize:

1. **Tools** → functionality initiated by tool on toolbar
2. **Behaviors** → non-UI logic
3. **Layouts** → custom application “look and feel”
4. **Controls** → UI integrated into application
Add-Ins – Implementation

- Develop using the Extensibility API
  - Included in the Extensibility SDK
- ESRI.ArcGIS.Client.Extensibility assembly
  - Lightweight API for Viewer
  - Provides access to map, selected layer, and pop-up
  - Methods to easily show UI in dialogs
  - Hooks to store and load add-in configuration
- Any Silverlight library can be referenced
- Use the VS Template to start
Tools – Overview

- Simplest, most common extensibility endpoint
- Logic initiated by user click
- Appear as buttons on toolbars and menus
- Users can add them from Application Builder
Tools – Implementation

- Implement ICommand
  - Execute – tool logic
  - CanExecute – determine whether tool is enabled or disabled
  - CanExecuteChanged – raise if enabling/disabling is needed

- Add Attributes:
  - DisplayName*, Description, Category, DefaultIcon
  - Export* (MEF attribute)

```csharp
namespace MyTool
{
    [DisplayName("My First Tool")]
    [Description("This is a simple example of a tool")]
    [Category("Category for my tools")]
    [DefaultIcon("MyTool;component/tool16.png")]
    [Export(typeof(ICommand))]
    public class MyCommand : ICommand
    {
        public bool CanExecute(object parameter)
        {
            return true;
        }

        public event EventHandler CanExecuteChanged;

        public void Execute(object parameter)
        {
        }
    }
}
Tool Contexts

- **Main Toolbar**
  - Always visible
  - Commonly used tools
- **Attribute Table Toolbar**
  - Tools for interacting with records and selection
- **Layer Toolbar & Context Menu**
  - Tools for interacting with individual layers
- **Pop-up Toolbar**
  - Tools for interacting with individual features
- **Same implementation pattern for all contexts**
  - ICommand, ISupportsConfiguration
Tool Contexts – Common API Hooks

- **All Tools**
  - `MapApplication.Current.ShowWindow`

- **Main Toolbar**
  - `MapApplication.Current.Map`

- **Attribute Table Toolbar**
  - `Graphic.Attributes`

- **Layer Toolbar & Context Menu**

- **Pop-up Toolbar**
  - `OnClickPopupInfo` passed to `Execute` & `CanExecute`
  - `OnClickPopupInfo.PopupItem.Graphic`
Tool Configurability

- Allows configuring a tool interactively
  - Gives designer flexibility in how a tool works
  - Reusability
- Examples
  - Geoprocessing
  - Basemap Gallery
  - Bookmarks (code in Interactive SDK/AGOL)
  - Print (code on AGOL)
  - Search (code on AGOL)
Implementing Tool Configurability

- **ISupportsConfiguration**
  - **Configure** – fired when user wants to configure tool
  - **SaveConfiguration** – serialize tool config
  - **LoadConfiguration** – deserialize tool config and initialize

```csharp
#region ISupportsConfiguration members

public void Configure()
{
    // Configuration logic
}

public void LoadConfiguration(string configData)
{
    // Load the saved configuration
}

public string SaveConfiguration()
{
    // Serialize the configuration to a string
}

#endregion
```
Demo – Visual Studio template

- Tool: ICommand, ISupportsConfiguration
- Running code from Visual Studio
- Configuration options in App Builder
- Attaching to process (debugging)
- Saved configuration in Deployed Application
Enhanced Configuration

- **ISupportsWizardConfiguration**
  - Show tool’s config UI within the add/edit wizard
  - Seamless configuration experience

- **Same members as ISupportsConfiguration, plus:**
  - Pages – pages shown for configuring the tool
  - DesiredSize – how much space you want for your UI
  - CurrentPage – the page currently being shown
  - PageChanging – called by host when page is about to change
  - OnCompleted – called by host on wizard completion
  - OnCancelled – called by host on wizard cancellation
Demo – Bookmarks tool

- Example of MVVM Pattern
- MapApplication.Current.Map
- ISupportsWizardConfiguration
Behaviors - Overview

• Functionality without user interaction
• Examples
  • Introductory/terms of use dialog
  • Initialize extent from query string
  • Show coordinates
  • Constrain extent
Behaviors - Implementation

- Inherit from `Behavior<Map>`
  - Override `OnAttached` and `OnDetached`

- Optionally implement `ISupportsConfiguration`
  - Configure, `LoadConfiguration`, `SaveConfiguration`

- Add attributes for `DisplayName`, `Category`
  - Don’t forget MEF Export attribute!

```csharp
[Export(typeof(Behavior<Map>))]
[DisplayName("My Behavior")]
public class MyBehavior : Behavior<Map>, ISupportsConfiguration {
```
Demo - Map Scale Settings behavior

- Behavior
- ISupportsConfiguration (Configurable behavior)
- Frequently requested!
Layouts - Overview

- Define application look and feel
- Loose xaml files – all markup, no code
- Position of viewer elements
  - Side panel
  - Navigation control
  - Scalebar
  - Title, logo, links
  - More
- Styles used by Viewer
  - Tool buttons
  - Pop-ups
  - Navigation control
  - More
Layouts - Implementation

- Start with copy of an existing layout
- Open-ended
  - Anything that can be declared in XAML
- Some well-known elements expected by Viewer
  - If it has an x:Name, don’t remove it

```xml
<ContentControl x:Name="ScaleBarContainer"
    Margin="40,0,0,2"
    HorizontalAlignment="Left"
    HorizontalContentAlignment="Stretch"
    VerticalAlignment="Bottom"
    Foreground="WhiteSmoke">
    <ContentControl.Effect>
        <DropShadowEffect BlurRadius="10" ShadowDepth="2" Opacity="1" Direction="300" />
    </ContentControl.Effect>
</ContentControl>

<ContentControl x:Name="AttributionDisplayContainer"
    HorizontalAlignment="Right" VerticalAlignment="Bottom"
    Margin="0,0,20,10"/>
```
Layouts - Deployment

- Copy to Builder\Templates\Default\Config\Layouts
- Create preview image
  - Name the same as xaml file
  - Ideal size is 580 x 359
- Tip - clear browser cache
Controls - Overview

- Pieces of UI that are “built-in”
- Included in layout
- Can implement a tool to toggle on/off
- Out-of-the-box examples
  - Map
  - Navigation control
  - FeatureDataGrid (attribute table)
- Interactive SDK – Chart control in Layout
Demo - Chart control in Layout

```xml
<Borders x:Name="ChartContainer" Style="{StaticResource GlassyBorder_Bottom}" Margin="5,0,0,5" VerticalAlignment="Bottom" d:LayoutOverrides="HorizontalAlignment, VerticalAlignment" HorizontalAlignment="Left" Width="291">
  <Grid>
    <Border Style="{StaticResource GlassyBorder_Top}" Margin="0" />
    <StackPanel Background="{StaticResource BackgroundBrush}" Margin="5">
      <charting:Chart Title="Sample Pie Chart" BorderThickness="0" MaxHeight="250" VerticalContentAlignment="Center" Height="253" VerticalAlignment="Bottom" Margin="1,0,0,0" HorizontalAlignment="Left" Width="288" d:LayoutOverrides="VerticalAlignment">
        <charting:PieSeries FontSize="10">
          <charting:PieSeries.ItemsSource>
            <controlsToolkit:ObjectCollection>
              <sys:Double>45</sys:Double>
              <sys:Double>26</sys:Double>
              <sys:Double>18</sys:Double>
              <sys:Double>11</sys:Double>
            </controlsToolkit:ObjectCollection>
          </charting:PieSeries.ItemsSource>
        </charting:PieSeries>
      </charting:Chart>
    </StackPanel>
  </Grid>
</Borders>
```
Summary

• The Viewer is configurable, but lots for devs, too
  • Devs are the bridge from generic to specific

• Many ways to make the Viewer do what you need

• Add-Ins are inherently reusable

• Easy, flexible, rich development environment
Existing tools and samples

- Anything authored by “SLDevTeam” or in the Interactive SDK (http://www.arcgis.com/home/search.html?q=owner%3ASLDevTeam%20silverlight%20viewer&t=content)
- Configurable Search tool
- Configurable Print tool (ArcGIS Server Print service)
- Query Related Records tool
- Measure tool
- Add WMS Layer – (most requested on forums)
- Zoom in past basemap (“Map Scale Settings behavior” on AGOL)
- Go to Coordinates tool
- Constrain Extent behavior
Developer Resources

- ArcGIS Viewer for Silverlight Interactive SDK
- ArcGIS API for Silverlight Interactive SDK
- Help and conceptual doc
- FORUMS!!!
Wrap-Up

• **Session resources**
  - Code on code gallery/AGOL – links.esri.com/silverlightviewer
  - PDF of slides will be posted – proceedings.esri.com
  - Session is recorded!

• **Sessions of interest**
  - Software Development and Design Using MVVM

• Please fill out your session evals!
  - Offering ID=267
    - http://esriurl.com/survey

• Contact me – kdalton@esri.com