

# ESRI Developer Summit

March 22–25, 2010  
Palm Springs, CA

## Patterns and Best Practices for Building Applications with the ArcGIS API for Microsoft Silverlight/WPF

*Art Haddad*

*Rex Hansen*

*Morten Nielsen*

*@ajhaddad*

*@rex\_hansen*

*@sharpgis*



# Agenda

- **Introductions**
- **Overview**
- **Deep Dive**
  - Behaviors
  - API Extensibility
  - Templates
  - Graphics Performance Tips/Tricks
  - Custom Symbols
  - Using WCF



Microsoft  
**Silverlight**



Microsoft  
**Windows  
Presentation  
Foundation**

**ArcGIS API for Silverlight/WPF V2**  
**OVERVIEW**

# What is the ArcGIS API for Microsoft Silverlight/WPF?

- **Powerful Web and Desktop Mapping API**
- **Designed for the Silverlight and WPF Platforms**
- **Utilize ArcGIS Server, Bing Maps services**
- **Silverlight provides cross browser support**
- **WPF supports native 64 bit (x64) platforms**
- **Integrates with Microsoft development environments**



**Behaviors**

**DEEP DIVE**



# Behaviors

- **What are they?**
  - Behaviors are self-contained, re-usable snippets of interactivity that can be assigned interactively to art board content
- **What problems do they solve?**
  - Makes development simpler
  - Functionality is cleanly encapsulated and reusable
- **What's the value to me?**
  - Code Re-use and XAMLify

## Behavior in XAML

```
<i:Interaction.Behaviors>  
    <esriBehaviors:ShowCoordinatesBehavior/>  
</i:Interaction.Behaviors>
```

# Behaviors: Actions

- Provide functionality to do something
- NOTE: Aren't particularly useful on their own
  - No way to activate that functionality.
- Invoking actions requires a *trigger*.



# Behaviors: Triggers

- **Objects that contain one or more actions and invoke those actions in response to some stimulus.**
- **One very common trigger is an EventTrigger.**
- **Other examples**
  - **might include a trigger that fires on a timer, or a trigger that fires when an unhandled exception is thrown.**

Expression Blend

**DEMO – USING BEHAVIORS**

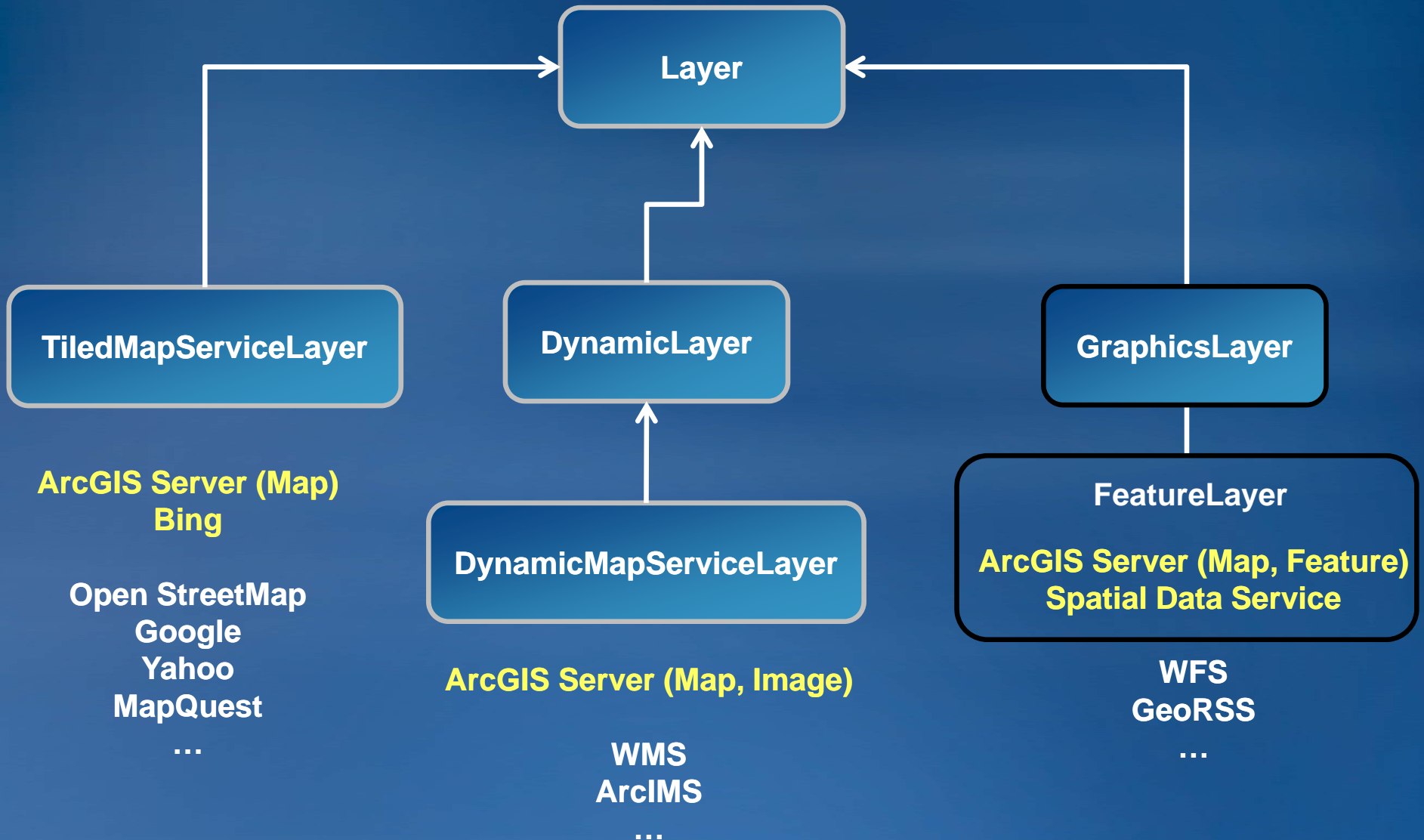
# Behaviors – Being practical

- **Not a Magic bullet**
- **Built as part of the Expression Blend Interactivity SDK**
  - Designed as part of Expression Blend
- **Some things are simpler by using a button with simple code behind**
  - Not a replacement for custom controls

**Extensibility (Rex Hansen)**

**DEEP DIVE**

# Creating custom layers



\* Included

**Demo**

# **CUSTOM TILED MAP SERVICE LAYER**



**Templates (Morten Nielsen)**

**DEEP DIVE**

# Two out-of-the-box templates (maybe more to come...)

## Standard Template



## Showcase Template



# Customizing the templates

1. Add your own data to the map.
2. Modify shared resources for quick update of titles, color scheme etc.
3. Add more items/buttons to the menu.
4. Reuse application's actions and user-controls like draggable windows to create new toolbars, views etc.
5. Add more ESRI and 3<sup>rd</sup> party controls.
6. Write custom application code.
7. Keep on extending...

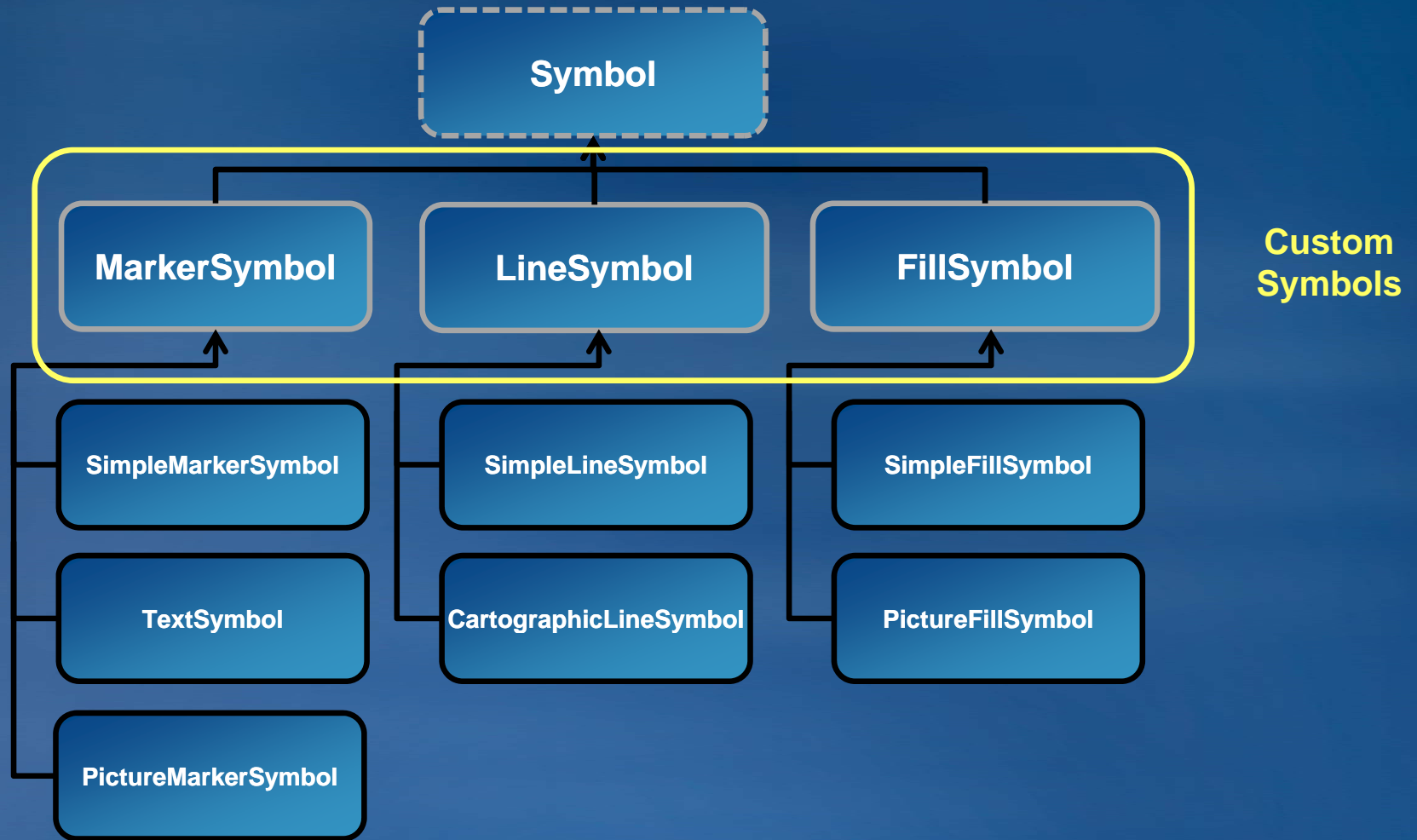
**Demo**

# **CUSTOMIZING THE TEMPLATE**

**Custom Symbols**

**DEEP DIVE**

# Symbol Hierarchy



----- Abstract

———— Class

———— Sealed



# ControlTemplate

Template defines the UI Elements inside symbol.

- **MarkerSymbol:**
  - Can be any UIElement
- **Fill and LineSymbol:**
  - Only: `<Path x:Name="Element" />` only

# MarkerSymbol.ControlTemplate

```
<esriSymbols:MarkerSymbol x:Key="MySymbol">
  <esriSymbols:MarkerSymbol.ControlTemplate>
    <ControlTemplate>
      <Ellipse Width="10"
                Height="10"
                Fill="Red" />
    </ControlTemplate>
  </esriSymbols:MarkerSymbol.ControlTemplate>
</esriSymbols:MarkerSymbol>
```

# FillSymbol.ControlTemplate

```
<esriSymbols:FillSymbol x:Key="MySymbol">  
  <esriSymbols:FillSymbol.ControlTemplate>  
    <ControlTemplate>  
      <Path x:Name="Element" Fill="Blue"  
        Stroke="Red"  
        StrokeThickness="2.0" />  
    </ControlTemplate>  
  </esriSymbols:FillSymbol.ControlTemplate>  
</esriSymbols:FillSymbol>
```

# LineStyle.ControlTemplate

```
<esriSymbols:FillSymbol x:Key="MySymbol">
  <esriSymbols:FillSymbol.ControlTemplate>
    <ControlTemplate>
      <Path x:Name="Element" Fill="Blue"
          Stroke="Red"
          StrokeThickness="2.0" />
    </ControlTemplate>
  </esriSymbols:FillSymbol.ControlTemplate>
</esriSymbols:FillSymbol>
```

## Binding to properties

```
public class MySymbol : MarkerSymbol {
    public double Size { get; set; }
}
---
<esriSymbols:MarkerSymbol x:Key="MySymbol">
  <esriSymbols:MarkerSymbol.ControlTemplate>
    <ControlTemplate>
      <Ellipse Width="{Binding Symbol.Size}"
        Height="{Binding Symbol.Size}"
        Fill="{Binding Attributes[Color]}" />
    </ControlTemplate>
  </esriSymbols:MarkerSymbol.ControlTemplate>
</esriSymbols:MarkerSymbol>
```

# Visual States

- Defines visual appearance for a control's state
- Animation defines the transition from one state to another.

Symbols have two Visual State Groups:

- SelectionStates: Selected, Unselected
- CommonStates: Normal, MouseOver



# Visual States

```
<Grid>
  <VisualStateManager>
    <VisualStateGroup Name="CommonStates">
      <VisualState Name="MouseOver">
        <Storyboard>
          <!--Animation goes here-->
        </Storyboard>
      </VisualState>
      <VisualState Name="Normal" />
    </VisualStateGroup>
    <VisualStateGroup Name="SelectionStates" />
  </VisualStateManager>
</Grid>
```

**Symbol Templates**

**DEMO**

**Graphics Performance Tips/Tricks**  
**(Morten Nielsen)**

**DEEP DIVE**

**How do I render a 1,000,000+ graphics on the client?**

**Would the user really want to see a million features at the same time?**

**How long would it take to download that many features to the client?**

**Try and limit the number of features displayed.**

- Use MaximumResolution property to hide features when zomed out.**
- Cluster features together to show fewer elements.**
- Let the user select an area of interest and only show data for that area.**

# Optimizing the symbols

- **Simpler symbol templates render faster than complex ones!**
- **Share resources as often as possible (brushes etc)**

**Things that often hurt rendering performance:**

- **Binding**
- **Visual States**
- **Disable Hit Test (`IsHitTestVisible="False"`)**

## Inefficient Marker symbol:

```
<esriSymbols:MarkerSymbol x:Key="MySymbol">
  <esriSymbols:MarkerSymbol.ControlTemplate>
    <ControlTemplate>
      <Grid>
        <Ellipse Width="{Binding Symbol.Size}"
          Height="{Binding Symbol.Size}"
          Fill="{Binding Symbol.Brush}"
        />
      </Grid>
    </ControlTemplate>
  </esriSymbols:MarkerSymbol.ControlTemplate>
</esriSymbols:MarkerSymbol>
```



## Optimized Marker symbol:

```
<esriSymbols:MarkerSymbol x:Key="MySymbol">  
  <esriSymbols:MarkerSymbol.ControlTemplate>  
    <ControlTemplate>  
  
      <Ellipse Width="10"  
                Height="10"  
                Fill="Red"  
            />  
  
    </ControlTemplate>  
  </esriSymbols:MarkerSymbol.ControlTemplate>  
</esriSymbols:MarkerSymbol>
```

## Inefficient Picture Marker symbol:

```
<esriSymbols:MarkerSymbol x:Key="MySymbol">  
  <esriSymbols:MarkerSymbol.ControlTemplate>  
    <ControlTemplate>  
      <Image Width="20" Height="20"  
        Source="http://myserver/myimage.png" />  
    </ControlTemplate>  
  </esriSymbols:MarkerSymbol.ControlTemplate>  
</esriSymbols:MarkerSymbol>
```

# Optimized Picture Marker symbol

```
<ImageBrush x:Key="MyImageBrush"
  ImageSource="http://...../myimage.png" />

<esriSymbols:MarkerSymbol x:Key="MySymbol">
  <esriSymbols:MarkerSymbol.ControlTemplate>
    <ControlTemplate>
      <Rectangle Width="20" Height="20"
        Fill="{StaticResource MyImageBrush}" />
    </ControlTemplate>
  </esriSymbols:MarkerSymbol.ControlTemplate>
</esriSymbols:MarkerSymbol>
```

**Using WCF (Rex Hansen)**

**DEEP DIVE**

# What is WCF?



- **Windows Communication Foundation**
  - .NET communication architecture
  - **Fundamentals:**
    - Host and client agree on a contract
    - Host provides an endpoint with binding, address, contract
    - Client uses proxy tied to the endpoint, thus contract
  - **Deserialization handled for you**
    - Types must be serializable
- **You get to focus on functionality, not communication!**

Demo

**SENDING GEOMETRY TO THE SERVER WITH WCF**

# Questions ?

**Please fill out the session surveys**

## **Links and resources:**

- Resource Center: <http://esriurl.com/sl>
- SDK Samples: <http://esriurl.com/slsdk>
- API Reference: <http://esriurl.com/slref>
- Silverlight/WPF Blog: <http://esriurl.com/slblog>

## **Twitter:**

- @ajhaddad @rex\_hansen @sharpgis