

2011 Esri China  
Developer Summit

# 2011Esri中国开发者大会

## ArcGIS Microsoft Silverlight/WPF API——高级篇

Esri中国（北京）有限公司 马克玲



分享地理价值

# 讨论话题...

- 用户体验和RIA应用程序设计
- Silverlight API的可扩展性
- 如何提升Graphics性能
- WPF/Silverlight/WP高效代码重用

# 用户体验和RIA应用程序设计

```
function init() {  
    var map = new esri.Map("map");  
    var tl = ledmapServiceLayer = new  
    esri.layers.ArcGISLayer("ces/esri_imagery",  
    "ces/esri_imagery", true);  
    map.addLayer(tl);  
}  
  
function getDriveTimePolys(results) {  
    var features = results[0].features;  
    for (var f=0; f<features.length; f++) {  
        var feature = features[f];  
        if (feature.type === "Polygon") {  
            var polys = feature.geometry.paths;  
            polys.forEach(function(polys) {  
                polys.forEach(function(poly) {  
                    poly.forEach(function(coords) {  
                        coords.forEach(function(coord) {  
                            coord.x = coord.x * 1000000;  
                            coord.y = coord.y * 1000000;  
                        });  
                    });  
                });  
            });  
        }  
    }  
}
```

# 高效地使用地图

- 何时展现你的Web地图应用...
  - “茫然失措...”
  - “我现在该做什么？”
  - “哪些图标符号是我假设要使用的？”
- 设计应用
  - 保持简洁
  - 简洁是关键



# Demo：用户体验和RIA应用程序设计

以地图为中心的应用

以表单为中心的应用

```
function init() {  
    var map = new esri.Map("map");  
    var tl = ledMapService.layer = new  
    esri.Layer({url: "CGI?Service=MapServer&Request=Layers",  
    urlParams: {esri: "ces/Esri_Imagery",  
    map: ".addLayer(" + ledMapService.url + "};  
}  
  
function getDriveTimePolys(results) {  
    var features = results(0).features;  
    for (var f=0; f<features.length; f++) {  
        var feature = features[f];  
        var poly = feature.geometry;  
        var polysymbol = new esri.Symbol("point", new esri.Path(0, 0, 0, 0, 50));  
        polysymbol.setColor(feature.attributes["color"]);  
        polysymbol.setSize(100);  
        polysymbol.setStrokeColor("black");  
        polysymbol.setZIndex(1);  
        new dojo.Color(0, 0, 0, 0.5);  
        polysymbol.setColor(new dojo.Color(0, 0, 0, 0.5));  
        polysymbol.setStrokeColor("black");  
        polysymbol.setZIndex(1);  
        feature.setSymbol(polysymbol);  
        else {if (feature.attributes["color"] == "red") {  
            polysymbol.setColor("red");  
            polysymbol.setSize(100);  
            polysymbol.setStrokeColor("black");  
            polysymbol.setZIndex(1);  
            feature.setSymbol(polysymbol);  
        }  
    }  
}
```

# Demo : Silverlight API的可扩展性

自定义符号

自定义控件模板

自定义数据源

```
function onMapClick {  
    var map = new esri.Map("map");  
    var tl = ledMapService.layer = new  
    esri.layers.ArcGISLayer("http://services.esri.com/arcgis/rest/services/Esri_Imagery  
    /MapServer/Map", esri.ArcGISMapServiceStyle);  
    map.addLayer(tl);  
}  
  
function getDriveTimePolys(results)  
var features = results(0).features;  
for (var f=0; f<features.length; f++)  
    var feature = features[f];  
    var polys = feature.attributes["Polys"];  
    var polysSymbol = new esri.Symbol(  
        "polys", polys, esri.SymbolFillStyle.SOLID,  
        new dojo.Color([0, 0, 0, 0.5]));  
    feature.setSymbol(polysSymbol);  
    feature.setSymbolClass("polys");  
}
```





# 如何在客户端渲染数以万计的graphics?

- 用户真的想同时在客户端看到数百万的要素吗？
- 大量要素下载到客户端将需要多长时间？
- 尝试并限制同时显示的要素数
  - 在缩小时，使用MaximumResolution 属性来隐藏图层
  - 将要素整合为簇，以“集合”的形式显示要素。
  - 让用户选择感兴趣区域，并仅为那个区域显示数据（例如：  
FeatureLayer.Mode = Selection )



# 优化符号

- 简单的符号模板比复杂的渲染更快
- 尽可能的共享资源 ( brushes 等等 )
- 通常影响渲染性能的因素：
  - Binding
  - Visual States
  - Disable Hit Test(IsHitTestVisible= “False” )
- PictureMarkerSymbol 性能，参考：  
<http://esriurl.com/PictureMarkerPerformance>

# 低效的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>
```

# 高效的Marker symbols

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





# WPF/Silverlight/WP 高效代码重用

```
function onMapClick {  
    var map = new esri.Map("map");  
    var tl = ledmapServiceLayer;  
    var layers = AGISITiledMapServiceLayer;  
    var map.addLayer(tl);  
    map.addLayer(layers);  
}  
  
function getDriveTimePolys(results) {  
    var features = results(0).features;  
    for (var f=0, l=features.length; f<l; f++) {  
        var feature = features[f];  
        var poly = feature.geometry.paths[0];  
        var polys = [];  
        for (var i=0, l=poly.length; i<l; i++) {  
            polys.push(poly[i]);  
        }  
        return polys;  
    }  
}
```

# 代码重用的优势

- 写一次！
  - 重用 ( WPF ) ,重用 ( Silverlight ) ,重用 ( Phone ) ,重用 ( Xbox ? ) , 重用 ( ? ? ? )
  - 换句话说：三倍你的生产力
- 总是同步DataContracts
  - 为服务器端和客户端编译数据契约——简单重用



# 如何实现？

- 写一次，但是：
  - Test, Test, Test...
  - 换句话：三倍你的测试投入
- 不同的屏幕 —— 不同的体验
- Silverlight 是WPF的子集
  - 但是 Silverlight 发展的速度快于WPF!
- Windows Phone 版本总是落后于Silverlight

# Demo : Silverlight/WPF/WP代码复用

```
function onInit() {  
    var map = new esri.Map("map");  
    var tl = ledmapServiceLayer;  
    var sr = layers.ArcGISRestServiceLayer;  
    var cgisr = layers.ArcGISRestServiceLayer;  
    map.addLayer(tl);  
    map.addLayer(sr);  
    map.addLayer(cgisr);  
}  
  
function getDriveTimePolys(results) {  
    var features = results(0).features;  
    for (var f=0; f<features.length; f++) {  
        var feature = features[f];  
        var poly = new esri.Polyline({  
            paths: feature.geometry.paths,  
            color: "#FF0000",  
            width: 5,  
            style: "solid",  
            dash: []  
        });  
        new dojo.Color(0, 0, 0, 0.5);  
        poly.symbol = new esri.Symbol({  
            color: "#FF0000",  
            style: "solid",  
            dash: []  
        });  
        poly.symbol.setAlpha(0.5);  
        feature.setSymbol(poly);  
    }  
}
```

# 实现细节

- 为平台指定代码使用编译条件：
  - `#if WINDOWS_PHONE`
  - `//Windows phone 7`
  - `#elseif SILVERLIGHT`
  - `// Silverlight`
  - `#else`
  - `// .NET`
  - `# endif`



# 实现细节——样例

- 在主题样式声明上不同：

```
Silverlight/Phone:  
Public HoverControl(){  
    # if SILVERLIGHT  
        this.DefaultStyleKey = typeof(HoverControl);  
    #endif  
}
```

```
WPF:  
Static HoverControl(){  
    # if !SILVERLIGHT  
        DefaultStyleKeyProperty.OverrideMetadata(  
            typeof(HoverControl),  
            new FrameworkPropertyMetadata(typeof(HoverControl)));  
    #endif  
}
```

# 实现细节：

- Xaml条件：
  - 在XAML中不能包含编译条件！
  - 这意味着可能必须要复写XAML

```
<Grid x:Name="LayoutRoot" Background="White">  
    <controls:HoverControl>  
#if SILVERLIGHT  
        <TextBlock Text="Hello from Silverlight" />  
#else  
        <TextBlock Text="Hello from .NET" />  
#endif  
    </controls:HoverControl>  
</Grid>
```

# Xaml reader

- *#if SILVERLIGHT*
- element = XamlReader.Load(xaml);
- *#else*
- using(MemoryStream xamlStream=  
•     new MemoryStream(UTF8Encoding.Default.GetBytes(xaml)))
- element = XamlReader.Load(xamStream);
- *#endif*



# Create bitmap

- `BitmapImage bmi = new BitmapImage();`
- `!SILVERLIGHT`
- `bim.BeginInit();`
- `#endif`
- `Image img = new Image();`
- `bmi.UriSource = new Uri(strUrl, UriKind.Absolute);`
- `SILVERLIGHT`
- `img.ImageFailed += img_ImageFailed;`
- `#else`
- `bmi.DownloadFailed += bmi_DownloadFailed;`
- `bmi.EndInit();`
- `#endif`
- `img.Source = bmi;`

## Stop — start animations

- *# if SILVERLIGHT*
- myStoryboard.Begin();
- myStoryboard.Stop();
- *#else*
- myStoryboard.Begin(element,true);
- myStoryboard.Stop(element);
- *#endif*

# 事件触发顺序的差异？

- `<UserControl Loaded="UserControl_Loaded">`
- `<my: Control Loaded = "MyControl_Loaded" />`
- `</UserControl>`
  
- Constructor
- Loaded
- OnApplyTemplate

## Silverlight

- 1、 UserControl Constructor
- 2、 MyControl Constructor
- 3、 MyControl Loaded
- 4、 UserControl Loaded
- 5、 MyControl.OnApplyTemplate

## WPF

- 1、 UserControl Constructor
- 2、 MyControl Constructor
- 3、 MyControl.OnApplyTemplate
- 4、 UserControl Loaded
- 5、 MyControl Loaded

# 二进制的兼容性

- 装配集共享
  - 为Silverlight 4编译的
  - 可以在WPF中重用
- 然而
  - 仅下列装配集是支持的：
    - Mscorlib.dll
    - System.dll
    - System.Core.dll
    - System.ComponentModel.Composition.dll
    - Microsoft.VisualBasic.dll
  - 没有UI 逻辑
- Windows Phone:
  - 编译为Silverlight 3, 可原样重用



## 底线：

- 你可以重用绝大多数的代码，也应该重用！
- ...但是：注意不同屏幕的开发差异：
  - Web：链接，导航，沙箱
  - Desktop：菜单，多窗口，文件访问
  - Phone：触控为中心，更小的屏幕



分 享 地 理 价 值