

ArcGIS Server 9.3 for Flex API

初级教程

资料来源： ESRI 中国社区

整 理： hhduyang

时 间： 2010-1-29

1. 环境搭建.....	3
2. 创建地图.....	6
3. 使用地图浏览控件.....	8
4. Draw 控件使用.....	10
5. 查询.....	12
6. Identify 功能	16
7. Geometry Service 的使用	19
8. GP 服务的使用.....	22
9. WebServices 的使用	25

1. 环境搭建

ArcGIS Server API for Flex 是 2009 年四月新推出来的进行 RIA 开发的 Flex 库, 是 ArcGIS Server 9.3 的一部分, 使用 ArcGIS API for Flex 可以基于 ArcGIS Server 建立漂亮的富互联网应用程序 rich internet applications (RIAs), 优点是运行速度快, 用户体验效果会比目前的 WEBGIS 好。

使用 ArcGIS Server API for Flex 可以达到下面的效果

- ①显示你的地图数据并可以和数据交互
- ②在服务器上执行空间处理模型并显示结果
- ③基于 ArcGIS Online 上的底图显示你自己的数据
- ④根据属性或者位置查找你的数据并显示结果
- ⑤查找地址并显示结果
- ⑥用创新的方式可视化结果
- ⑦创建 mashups

具 体 详 情 可 以 参 考

<http://resources.esri.com/ArcGISServer/APIs/Flex/index.cfm?fa=samples> 上面有一些例子。

ArcGIS Server API for Flex 下载地址为:

<http://resources.esri.com/ArcGISServer/APIs/Flex/index.cfm?fa=home>

里面有个 download。

开发 ArcGIS Server API for Flex 的程序需要 Flex 环境的支持。

Flex SDK3 的下载地址为

<http://opensource.adobe.com/wiki/display/FlexSDK/download?build=3.0.1.2012&pkgtype=1>

有了 Flex SDK3 和 ArcGIS Server API for Flex 我们就可以开发 RIA 的 Flex 程序了。

为了开发方便, 我们最好使用 Flex builder, 可以从 adobe 的官方网站上下载试用版,

下载地址为 http://download.macromedia.com/pub/Flex/Flex_builder/FB3_win.exe

安装好 Flex builder 之后会安装 Flex SDK3, 所以就不用自己手工安装了。

第一个 ArcGIS Server API for Flex 程序。

①打开 Flex builder, 创建一个 Flex 工程 (名称 Demo)。

②右键单击工程名, 选择属性, 在属性对话框中选择 Flex build path, 选择 libaray path 选项卡, 单击 add swc 把下载的 ArcGIS Server API for Flex 添加进去, 环境就配置好了 (下

图)。



2. 创建地图

在 Flex 中创建一个 esri 的 map，你只需要使用<esri:Map>标签就可以轻松完成。

在<esri:Map>标签中可以添加属性和响应事件的消息。

如下：

```
<esri:Map width="100%" height="50%" id="EsriMap" creationComplete="EsriMapCreationComplete(event);"
extentChange="ESRIMapExtentChange(event);" mouseMove="OnDrawMouseMove(event)" />
```

width 和 height 定义的 map 的大小，id="EsriMap" 唯一标识了这个 map。

针对 map 的消息也有很多，最常用的就是 creationComplete，resize 和鼠标消息了。

现在我们 map 有了，那么我们如何让她显示数据呢，这就需要给 <esri:Map> 标签添加一个 layer 子标签。

在 ags Flex API 中有以下几种类型的 layer。

ArcGISDynamicMapServiceLayer : Allows you to work with a dynamic map service resource exposed by the ArcGIS Server REST API.

ArcGISImageServiceLayer: Allows you to work with an image service resource exposed by the ArcGIS Server REST API.

ArcGISMapServiceLayer : The base class for ArcGIS Server map services.

ArcGISTiledMapServiceLayer : Allows you to work with a cached map service resource exposed by the ArcGIS Server REST API.

ArcIMSMapServiceLayer : Allows you to work with an ArcIMS image service.

GPResultImageLayer : Allows you to view a geoprocessing task result identified by jobId and parameterName.

GraphicsLayer: A layer that contains one or more Graphic features.

其中 GraphicsLayer 是支持客户端添加 Graphic features 的图层，需要在客户端表现的，或者交互操作中产生的要素都要加到这个 layer 上。

下面，我们就添加一个 ArcGISTiledMapServiceLayer 和 GraphicsLayer 到 map 上。

```
<esri:Map width="100%" height="50%" id="EsriMap" creationComplete="EsriMapCreationComplete(event);"
extentChange="ESRIMapExtentChange(event);" mouseMove="OnDrawMouseMove(event)" >
<esri:ArcGISTiledMapServiceLayer url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_Str
```

```
etMap_World_2D/MapServer" />
```

```
<esri:GraphicsLayer id="myGraphicsLayer" spatialReference="{sr}" />
```

```
</esri:Map>
```

其中<esri:ArcGISTiledMapServiceLayer>标签中的 url 属性为提供该服务的地址。<esri:GraphicsLayer>标签其中的 spatialReference 属性定义了该图层的空间参考系。

创建一个空间参考系，只需要使用<esri:SpatialReference>标签就可以了，其中 wkid 是 esri 已经定义好的空间参考系的 ID，具体 ID 对应的空间参考可以在 <http://resources.esri.com/help/9.3/ArcGISServer/APIs/REST/index.html?gcs.html> 中查找。如下为创建一个 kid="4326"的空间参考系。

```
<esri:SpatialReference id="sr" wkid="4326" />
```

那么，我们如何定义我们想显示的范围呢？

很简单，在<esri:Map>下面添加<esri:extent>子标签，我们就可以来控制当前的显示范围了。

```
<esri:extent>
```

```
<esri:Extent id="esriMapExtent" xmin="116" ymin="39.5" xmax="116.5" ymax="40.5" />
```

```
</esri:extent>
```

其中 x 是经度，y 是纬度。

这样一个北京地区的 map 就可以显示到我们面前了。

完整代码如下

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<mx:Application
```

```
xmlns:mx="http://www.adobe.com/2006/mxml"
```

```
xmlns:esri="http://www.esri.com/2008/ags"
```

```
pageTitle="Using ArcGIS API for Flex to connect to a cached ArcGIS Online service"
```

```
styleName="plain">
```

```
<esri:SpatialReference id="sr" wkid="4326" />
```

```
<esri:Map crosshairVisible="true">
```

```
<esri:extent>
```

```
<esri:Extent id="esriMapExtent" xmin="116" ymin="39.5" xmax="116.5" ymax="40.5" />
```

```
</esri:extent>
```

```
<esri:ArcGISTiledMapServiceLayer
```

```
url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_StreetMap_World_2D/MapServer" />
```

```
<esri:GraphicsLayer id="myGraphicsLayer" spatialReference="{sr}" />
```

```
</esri:Map>
```

```
</mx:Application>
```

3. 使用地图浏览控件

地图浏览包括放大，缩小，漫游，复位，上级窗口，下级窗口等
在 ArcGIS API for Flex 中，esri 已经封装好了一个地图浏览，我们在程序中可以直接使用，只需要几十行代码，就可以完成复杂的任务。

首先在上一讲的基础上创建一个 Navigation 控件。使用<esri:Navigation>标签就可以了。

```
<esri:Navigation id="navToolbar" map="{myMap}"/>
```

将<esri:Navigation>标签的 map 属性设置为你要控制的 map，也就是说把控件绑定到 map 上。然后我们创建一个工具条，用来做对 map 浏览的控制

```
<mx:ControlBar horiz paddingBottom="0" paddingTop="0">
  <mx:Button label="放大" click="navToolbar.activate(Navigation.ZOOM_IN)" />
  <mx:Button label="缩小" click="navToolbar.activate(Navigation.ZOOM_OUT)" />
  <mx:Button label="漫游" click="navToolbar.activate(Navigation.PAN)" />
  <mx:Button label="上级窗口"
    click="navToolbar.zoomToPrevExtent()" enabled="{!navToolbar.isFirstExtent}"/>
  <mx:Button label="下级窗口" click="navToolbar.zoomToNextExtent()" enabled="{!navToolbar.isLastExtent}"/>
  <mx:Button label="复位" click="navToolbar.zoomToFullExtent()"/>
```

button 的 click 消息响应函数直接调用 <esri:Navigation>控件的函数就可以了。放大，缩小，漫游调用的都是激活工具的函数。剩余 3 个 button 的响应函数分别

调用<esri:Navigation>控件 zoomToPrevExtent，zoomToNextExtent，zoomToFullExtent 函数就 ok 了。代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  xmlns:esri="http://www.esri.com/2008/ags"
  pageTitle="Using ArcGIS API for Flex to connect to a cached ArcGIS
Online service"
  styleName="plain">
  <esri:Navigation id="navToolbar" map="{myMap}"/>
  <esri:SpatialReference id="sr" wkid="4326"/>
  <mx:ControlBar horizontalGap="0" paddingBottom="0" paddingTop="0">
    <mx:Button label="放大" click="navToolbar.activate(Navigation.ZOOM_IN)" />
    <mx:Button label="缩小" click="navToolbar.activate(Navigation.ZOOM_OUT)" />
    <mx:Button label="漫游" click="navToolbar.activate(Navigation.PAN)" />
    <mx:Button label="上级窗口" click="navToolbar.zoomToPrevExtent()"/>
```



```
enabled="{!navToolbar.isFirstExtent}"/>
    <mx:Button label="下级窗口" click="navToolbar.zoomToNextExtent()" enabled="{!
navToolbar.isLastExtent}"/>
    <mx:Button label="复位" click="navToolbar.zoomToFullExtent()"/>
</mx:ControlBar>
<esri:Map crosshairVisible="true" id = "myMap">
    <esri:extent>
        <esri:Extent id = "esriMapExtent" xmin="116" ymin="39.5" xmax="116.5" ymax="40.5"/>
    </esri:extent>
    <esri:ArcGISTiledMapServiceLayer
url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_StreetMap
_World_2D/MapServer" />
    <esri:GraphicsLayer id="myGraphicsLayer" spatialReference="{sr}"/>
</esri:Map>
</mx:Application>
```

4. Draw 控件使用

ArcGIS API for Flex 在客户端提供了强大的绘制功能，可以通过简单的调用来完成复杂的绘制工作，达到很好的交互效果，提供完美的用户体验，感觉不出来是在 web 上操作。

下面我们就一起来实现绘制功能。

首先，我们需要定义一个 draw 控件，使用<esri:draw>标签来完成创建工作，如下面的代码

```
<esri:daw id="drawToolbar" map="{myMap}" graphicsLayer="{myGraphicsLayer}" />
```

<esri:draw >标签的 map 属性要绑定要你需要绘制的 map 上。graphicsLayer 属性要绑定到一个 graphicsLayer 上，所有绘制的结果都会添加到 graphicsLayer 上，graphicsLayer 的创建可以参考第二讲的内容。控件创建好了，我们下面就可以调用这个控件的函数来完成绘制任务。和上一讲一样，我们创建一个 controlbar 来响应这些绘制的操作。

```
<mx:ControlBar horiz paddingBottom="0" paddingTop="0">
    <mx:Button label="点" click="drawToolbar.activate(Draw.MAPPOINT)" />
    <mx:Button label="线" click="drawToolbar.activate(Draw.POLYLINE)" />
    <mx:Button label="流线" click="drawToolbar.activate(Draw.FREEHAND_POLYLINE)" />
    <mx:Button label="矩形" click="drawToolbar.activate(Draw.EXTENT)" />
    <mx:Button label="多边形" click="drawToolbar.activate(Draw.POLYGON)" />
    <mx:Button label="流多边形" click="drawToolbar.activate(Draw.FREEHAND_POLYGON)" />
"/>
<mx:Button label="结束绘制" click="drawToolbar.deactivate()" />
</mx:ControlBar>
```

在 button 的 click 响应函数里，我们只需要调用 draw 控件的 activate 函数来激活当前绘制工具就可以了。当你不想绘制的时候，调用 deactivate 函数就 ok 了，释放掉当前选择的绘制工具。代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
    xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:esri="http://www.esri.com/2008/ags"
    pageTitle="Using ArcGIS API for Flex to connect to a cached ArcGIS
Online service"
    styleName="plain">
    <esri:Draw id="drawToolbar" map="{myMap}" graphicsLayer="{myGraphicsLayer}" />
    <esri:SpatialReference id="sr" wkid="4326"/>
    <mx:ControlBar horizontalGap="0" paddingBottom="0" paddingTop="0">
```

```

<mx:Button label="点" click="drawToolbar.activate(Draw.MAPPOINT)" />
<mx:Button label="线" click="drawToolbar.activate(Draw.POLYLINE)" />
<mx:Button label="流线" click="drawToolbar.activate(Draw.FREEHAND_POLYLINE)" />
<mx:Button label="矩形" click="drawToolbar.activate(Draw.EXTENT)" />
    <mx:Button label="多边形" click="drawToolbar.activate(Draw.POLYGON)" />
    <mx:Button label="流多边形" click="drawToolbar.activate(Draw.FREEHAND_POLYGON)"/>
    <mx:Button label="结束绘制" click="drawToolbar.deactivate()"/>
</mx:ControlBar>
<esri:Map crosshairVisible="true" id = "myMap">
    <esri:extent>
<esri:Extent id = "esriMapExtent" xmin="116" ymin="39.5" xmax="116.5" ymax="40.5"/>
</esri:extent>
        <esri:ArcGISTiledMapServiceLayer
url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_StreetMap
_World_2D/MapServer" />
        <esri:GraphicsLayer id="myGraphicsLayer" spatialReference="{sr}"/>
</esri:Map>
</mx:Application>

```

5. 查询

在 gis 中，针对要素的查询是一个最基本的操作，也是最常用的操作之一。下面我们介绍如何使用 arcgis api for flex 来查询我们需要的东西。要在 arcgis api for flex 中进行查询操作，首先需要定义一个查询任务面板。使用<esriqueryTask>标签就可以了。

```
<esriqueryTask id="queryTask"
  url="http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/Demographics/ESRI_Census_USA/MapServer/5">
  <esriquery id="query"
    text="{qText.text}"
    returnGeometry="true"
    spatialRelati>
    <esriqueryFields>
      <mx:String>MED_AGE</mx:String>
      <mx:String>POP2007</mx:String>
    </esriqueryFields>
  </esriquery>
</esriqueryTask>
```

id 唯一标识这个查询任务，url 告诉查询面板去哪查。

<esriquery>定义一个查询，text 是你需要查询的东西，<esriqueryFields>子标签告诉 Query 查询的结果返回哪些字段的内容。

QueryTask 定义好之后，我们还需要在界面上来调用这个 QueryTask。因此我们定

义一个文本输入框和一个查询按钮

```
<mx:Panel title="Query a layer (search for a state)" layout="horizontal" backgroundColor="0xB2BFC6"
  borderStyle="solid">
  <mx:TextInput width="100%" id="qText" enter="doQuery()" text="California"/>
  <mx:Button label="Do Query" click="doQuery()"/>
</mx:Panel>
```

文本输入框 用来输入你想要查询的内容，button 用来执行查询的动作。

那么这个 doQuery()怎么实现呢？我们在 mxml 的标签中已经无法实现，这就需要引

入 activeScript 脚本。我们需要在 mxml 中使用 activeScript 脚本来编写代码，实现我们想要的功能。

关于 activeScript 的语法大家可以参考 activeScript 的相关书籍。

要在 mxml 文档中插入 `activescript`，需要使用 `<mx:Script>` 标签

```
<mx:Script>
    <![CDATA[
        ]]>
</mx:Script>
```

`activescript` 是一种类 `java` 语言，它本身有一个 `AVM`，把 `activescript` 编译成 `java` 的代码，然后再通过 `JVM` 转换成字节码执行。

我们下面就开始实现 `doQuery()`;

首先，我们要用 `import` 指令引入我们需要的命名空间，和 `java` 基本一样

```
<mx:Script>
    <![CDATA[
        import com.esri.ags.Graphic;
        import com.esri.ags.tasks.FeatureSet;
        import com.esri.ags.tasks.Query;
        import mx.controls.Alert;
        import mx.rpc.AsyncResponder;
    ]]>
</mx:Script>
```

然后我们定义 `doQuery()` 函数： 注意 `activescript` 代码 要放到 `<mx:Script>` 标签中

```
private function doQuery() : void
{
    queryTask.execute( query, new AsyncResponder( onResult, onFault ));
}
```

在 `doQuery()` 函数中直接调用了 `queryTask` 的 `execute` 方法，这是一个异步调用。

成功响应 `onResult` 函数，失败则响应 `onFault` 函数。

查询已经写好了，那么我们怎么得到查询的结果呢？得到结果肖恩么表现呢？

这就需要我们在 `onResult` 函数中做一些事情了。

首先，定义 `onResult` 函数

```
function onResult( featureSet : FeatureSet, token : Object = null ) : void
{
    var displayFieldName : String = featureSet.displayFieldName;
    for each ( var myGraphic : Graphic in featureSet.features )
    {
        // ToolTip
        myGraphic.toolTip = "The 2007 population of "
```

```
        + myGraphic.attributes[displayFieldName] + " was "
        + myNumberFormatter.format(myGraphic.attributes.POP2007)
        + "\nMedian Age: " + myGraphic.attributes.MED_AGE + ".";
    // show on map
    myGraphicsLayer.add( myGraphic );
}
}
```

查询结果返回一个 FeatureSet，我们现在遍历这个 FeatureSet，然后把每个 feature 绘制到 GraphicLayer 上。

如果查询失败了怎么办呢，我们是不是要弹个东西出来告诉用户呢？

这就需要我们在 onFault 函数中做一些工作

```
function onFault( info : Object, token : Object = null ) : void
{
    Alert.show( info.toString() );
}
```

我们弹个对话框出来告诉用户，查找失败啦！

完整代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
    xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:esri="http://www.esri.com/2008/ags"
    pageTitle="Query Task"
>
    <mx:Script>
        <![CDATA[
            import com.esri.ags.Graphic;
            import com.esri.ags.tasks.FeatureSet;
            import com.esri.ags.tasks.Query;
            import mx.controls.Alert;
            import mx.rpc.AsyncResponder;

            private function doQuery() : void
            {
                queryTask.execute( query, new AsyncResponder( onResult, onFault ) );
                function onResult( featureSet : FeatureSet, token : Object = null ) : void
                {
                    var displayFieldName : String = featureSet.displayFieldName;
                    for each ( var myGraphic : Graphic in featureSet.features )
                    {
                        // ToolTip
                        myGraphic.toolTip = "The 2007 population of "
                            + myGraphic.attributes[displayFieldName] + " was "
                            + myNumberFormatter.format(myGraphic.attributes.POP2007)
                            + "\nMedian Age: " + myGraphic.attributes.MED_AGE + ".";
                    }
                }
            }
        ]]>
    </mx:Script>
</mx:Application>
```

```
        // show on map
        myGraphicsLayer.add( myGraphic );
    }
}
function onFault( info : Object, token : Object = null ) : void
{
    Alert.show( info.toString() );
}
}
]]>
</mx:Script>
<mx:NumberFormatter id="myNumberFormatter" useThousandsSeparator="true"/>
<!-- Layer with US States -->
<esri:QueryTask id="queryTask"
    url="http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/Demogra
    phics/ESRI_Census_USA/MapServer/5">
    <esri:Query id="query"
        text="{qText.text}"
        returnGeometry="true"
        spatialRelationship="esriSpatialRelEnvelopeIntersects">
        <esri:outFields>
            <mx:String>MED_AGE</mx:String>
            <mx:String>POP2007</mx:String>
        </esri:outFields>
    </esri:Query>
</esri:QueryTask>
<mx:Panel title="Query a layer (search for a state)" layout="horizontal"
background-color="0xB2BFC6" borderStyle="solid">
    <mx:TextInput width="100%" id="qText" enter="doQuery()" text="California"/>
    <mx:Button label="Do Query" click="doQuery()"/>
</mx:Panel>
<esri:Map>
    <esri:extent>
        <esri:Extent xmin="-170" ymin="15" xmax="-65" ymax="75"/>
    </esri:extent>
    <esri:ArcGISTiledMapServiceLayer
    url="http://server.arcgisonline.com/ArcGIS/rest/services/NPS_Physical_W
    orld_2D/MapServer" />
    <esri:GraphicsLayer id="myGraphicsLayer"/>
</esri:Map>
</mx:Application>
```

6. Identify 功能

identify 是 GIS 中比较常用的工具之一，在 ArcGIS API for Flex 中 esri 为我们提供了一个 Identify Task 来轻松完成 identify 的功能。

首先，还是使用<esri:IdentifyTask>标签来创建一个 Identify Task。

```
<!-- Identify Task -->
<esri:IdentifyTask id="identifyTask"
    identifyComplete="identifyCompleteHandler(event)"
    url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer"/>
```

当 identifyTask 执行完毕的时候响应 identifyComplete 消息，我们就可以把 identify 的结果做一些处理，比如添加到 Graphic layer 上。

在执行 identify 之前，首先要对 identify 的参数设置一下，我们需要一个 IdentifyParameters 对象。下面的代码是 AS3 脚本的代码，用来创建 IdentifyParameters 和 identify 执行。

```
var identifyParams : IdentifyParameters = new IdentifyParameters();
    identifyParams.returnGeometry = true;
    identifyParams.tolerance = 3;
    identifyParams.width = 600;
    identifyParams.height = 550;
    identifyParams.geometry = geometry;
    identifyParams.layerOption = IdentifyParameters.LAYER_OPTION_ALL;
    identifyParams.mapExtent = map.extent;
    identifyTask.execute( identifyParams );
```

其中 tolerance 是容差半径 width:Width of the map currently being viewed in pixels.height : Height of the map currently being viewed in pixels。geometry 是用来做 identify 的几何，常用的有点选，矩形选择，多边形选择等参数设置好了之后，直接调用 identifyTask.execute(identifyParams);就 ok 了。

那么我们用来做 identify 的几何怎么来呢，在什么时候去做 Identify 呢？首先回答第一个问题，做 identify 的几何我们可以利用第四讲中 draw 控件使用鼠标交互来获得，这也是 RIA 的特点之一。

那么在什么时候做 identify 呢？

就在做 identify 的几何画完之后做，嘿嘿，等于没说嘛，当然要在画完了就做。

现在我们就来完成上面的工作

定义一个 draw 控件

```
<esri:draw id="drawToolbar" map="{map}"  
graphicsLayer="{myGraphicsLayer}" drawEnd="drawEndHandler(event)">
```

记得添加上 drawEnd 消息的响应函数 drawEndHandler(event), 这个事件会在 draw 之后响应。用 as3 脚本实现 drawEndHandler 和 identifyCompleteHandler 函数

```
private function drawEndHandler(event:drawEvent):void  
{  
    var geometry : Geometry = event.geometry;  
    var identifyParams : IdentifyParameters = new IdentifyParameters();  
    identifyParams.returnGeometry = true;  
    identifyParams.tolerance = 3;  
    identifyParams.width = 600;  
    identifyParams.height = 550;  
    identifyParams.geometry = geometry;  
    identifyParams.layerOption = IdentifyParameters.LAYER_OPTION_ALL;  
    identifyParams.mapExtent = map.extent;  
    identifyTask.execute( identifyParams );  
    private function identifyCompleteHandler(event:IdentifyEvent):void  
    {  
        for each (var result:IdentifyResult in event.identifyResults)  
        {  
            myGraphicsLayer.add(result.feature);  
        }  
    }  
}
```

完整代码如下:

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:Application  
    xmlns:mx="http://www.adobe.com/2006/mxml"  
    xmlns:esri="http://www.esri.com/2008/ags"  
    layout="absolute"  
    pageTitle="Identify Features on the Map"  
>  
    <mx:Script>  
        <![CDATA[  
            import com.esri.ags.Graphic;  
            import com.esri.ags.events.DrawEvent;  
            import com.esri.ags.events.IdentifyEvent;  
            import com.esri.ags.geometry.Geometry;  
            import com.esri.ags.symbol.Symbol;  
            import com.esri.ags.tasks.IdentifyParameters;  
            import com.esri.ags.tasks.IdentifyResult;
```

```
import com.esri.ags.toolbars.Draw;
private function drawEndHandler(event:DrawEvent):void
{
    var geometry : Geometry = event.geometry;
    var identifyParams : IdentifyParameters = new IdentifyParameters();
    identifyParams.returnGeometry = true;
    identifyParams.tolerance = 3;
    identifyParams.width = 600;
    identifyParams.height = 550;
    identifyParams.geometry = geometry;
    identifyParams.layerOption = IdentifyParameters.LAYER_OPTION_ALL;
    identifyParams.mapExtent = map.extent;
    identifyTask.execute( identifyParams );
}
private function identifyCompleteHandler(event:IdentifyEvent):void
{
    for each (var result:IdentifyResult in event.identifyResults)
    {
        myGraphicsLayer.add(result.feature);
    }
}
]]>
</mx:Script>
<!-- Draw ToolBar -->
<esri:Draw id="drawToolBar" map="{map}"
    graphicsLayer="{myGraphicsLayer}" drawEnd="drawEndHandler(event)">
</esri:Draw>
<!-- Identify Task -->
<esri:IdentifyTask id="identifyTask"
    identifyComplete="identifyCompleteHandler(event)"
    url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Special
    ty/ESRI_StatesCitiesRivers_USA/MapServer"/>
<mx:Panel width="100%" height="100%">
    <mx:Button label="Identify" click="drawToolBar.activate(Draw.MAPPOINT)"/>
    <esri:Map id="map" width="100%" height="100%">
        <esri:ArcGISDynamicMapServiceLayer
            url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Special
            ty/ESRI_StatesCitiesRivers_USA/MapServer" />
        <esri:GraphicsLayer id="myGraphicsLayer"/>
    </esri:Map>
</mx:Panel>
</mx:Application>
```

7. Geometry Service 的使用

Geometry service 顾名思义,就是提供针对几何层级的服务,比如说 Project, Simplify , Buffer, Areas And Lengths , Lengths 等
 详细信息可以到下面的地址去看一下

<http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Geometry/GeometryServer>

这个网址发布了上面 5 个 Geometry service。

下面我们就来讲解如何在你的 Flex 地图中使用 Geometry service,我们拿最常用 buffer 来做例子。

首先 定义一个 GeometryService

```
<esri:GeometryService id="myGeometryService"
    url="http://sampleServer2.ArcGISOnline.com/ArcGIS/rest/services/Geometry/GeometryServer"/>
```

id 唯一标识了这个 GeometryService, url 指定了 GeometryService 的地址
 和 identify 工具类似, 要进行 buffer 操作我们先创建一个需要做 buffer 的几何,
 然后定义一个 BufferParameters , 然后执行 buffer 操作, 最后将 buffer 的结果绘制到 GraphicsLayer 上。

下面我们介绍一下 BufferParameters 参数的意义 distances 为 buffer 半径,
 features 为需要做 buffer 的要素集合, unit 为单位 bufferSpatialReference 为
 buffer 操作时的空间参照系。

设置完参数后, 要添加 buffer 完成时间的 EventListener, 当 buffer 完成后我们就可以处理这个 buffer 的结果了, 如下面 AS3 脚本

```
var bufferParameters : BufferParameters = new BufferParameters();
    bufferParameters.features = [point];
    bufferParameters.distances = [3000];
    bufferParameters.unit = BufferParameters.UNIT_METER;
    bufferParameters.bufferSpatialReference = new
    SpatialReference( 02113); myGeometryService.addEventListener
    (GeometryServiceEvent.BUFFER_COMPLETE, bufferCompleteHandler);
    myGeometryService.buffer( bufferParameters );
```

关键的一点, bufferParameters.features 参数设置的 features 一定要有空间参考系, 不然 buffer 不成功。其余的绘制操作, 将结果绘制到 GraphicsLayer 上的操作和上一讲基本相同, 就不再讲了。代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  xmlns:esri="http://www.esri.com/2008/ags"
  layout="absolute"
>
  <mx:Script>
    <![CDATA[
      import com.esri.ags.Graphic;
      import com.esri.ags.events.DrawEvent;
      import com.esri.ags.events.IdentifyEvent;
      import com.esri.ags.geometry.Geometry;
      import com.esri.ags.symbol.Symbol;
      import com.esri.ags.toolbars.Draw;
      import com.esri.ags.SpatialReference;
      import com.esri.ags.events.GeometryServiceEvent;
      import com.esri.ags.tasks.BufferParameters;
      private function drawEndHandler(event:DrawEvent):void
      {
        var geometry : Geometry = event.geometry;
        var myMapPoint:Graphic = new Graphic();
        myMapPoint.geometry = geometry;
        geometry.spatialReference = new SpatialReference(4016);
        var bufferParameters : BufferParameters = new BufferParameters();
        bufferParameters.features = [myMapPoint];
        bufferParameters.distances = [300000];
        bufferParameters.unit = BufferParameters.UNIT_METER;
        bufferParameters.bufferSpatialReference = new SpatialReference(102113);
        myGeometryService.addEventListener(GeometryServiceEvent.BUFFER_COMPLETE,
        bufferCompleteHandler);
        myGeometryService.buffer( bufferParameters );
      }

      function bufferCompleteHandler( event : GeometryServiceEvent ) : void
      {
        myGeometryService.removeEventListener
        (GeometryServiceEvent.BUFFER_COMPLETE, bufferCompleteHandler);
        for each ( var graphic : Graphic in event.graphics )
        {
          graphic.symbol = sfs;
          myGraphicsLayer.add( graphic );
        }
      }
    ]>
  </mx:Script>
  <esri:SimpleFillSymbol id="sfs" color="0xFF0000">
```

```
<esri:SimpleLineSymbol color="0x000000"/>
</esri:SimpleFillSymbol>
<!-- Draw ToolBar -->
<esri:Draw id="drawToolBar" map="{map}"
graphicsLayer="{myGraphicsLayer}" drawEnd="drawEndHandler(event)">
</esri:Draw>

<esri:GeometryService id="myGeometryService"
url="http://sampleServer2.ArcGISOnline.com/ArcGIS/rest/services/Geometr
y/GeometryServer"/>
<mx:Panel width="100%" height="100%">
<mx:Button label="buffer" click="drawToolBar.activate(Draw.MAPPOINT)"/>
<esri:Map id="map" width="100%" height="100%">
<esri:ArcGISDynamicMapServiceLayer
url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Special
ty/ESRI_StatesCitiesRivers_USA/MapServer" />
<esri:GraphicsLayer id="myGraphicsLayer"/>
</esri:Map>
</mx:Panel>
</mx:Application>
```

8. GP 服务的使用

在 ArcGIS API for Flex 中 esri 还为我们提供了 GP 服务, 在 ARCGISONLINE 上的 gp 服务有 CreateDriveTimePolygons 和 Viewshed, 下面我们以前 CreateDriveTimePolygons 服务为例来看看在 ArcGIS API for Flex 中如何使用 GP 服务。首先使用 `<esri:Geoprocessor>` 标签创建一个 gp 服务, url 指向提供 gp 服务的地址。

```
<esri:Geoprocessor id="gp"
url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Network
/ESRI_DriveTime_US/GPServer/CreateDriveTimePolygons" />
```

剩下的步骤和上一讲 Geometry service 的使用基本相同, 即设置参数, 调用 gp 服务, 得到结果, 表现结果。我们直接看一下下面的代码和例子就 ok 了, 这个例子比较简单。

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    pageTitle="Service Area"
    xmlns:esri="http://www.esri.com/2008/ags"
    layout="absolute"
>
    <mx:Script>
        <![CDATA[
            // Synopsis:
            // The map has a click event that calls computeServiceArea
            (event) when you click the map
            //      <esri:Map click="computeServiceArea(event)">
            // The computeServiceArea function sends a request to a GP
            task to create the geometries for the different drive times
            // The return drivetime features are used as the
            graphicProvider for a graphics layer.
            //      graphicsLayer.graphicProvider = fs.features;
            // The graphics layer is using a symbolFunction which will
            symbolize the drivetimes in different colors
            //      <esri:GraphicsLayer id="graphicsLayer"
            symbolFunction="fillFunc"/>
            // The fillFunc() is using the "ToBreak" attributes, which
            the GP task returned, to set different symbols on different drive
            times.

            import com.esri.ags.Graphic;
            import com.esri.ags.symbol.Symbol;
            import com.esri.ags.tasks.ExecuteResult;
```

```
import com.esri.ags.tasks.FeatureSet;
import com.esri.ags.tasks.ParameterValue;
import com.esri.ags.geometry.MapPoint;
import com.esri.ags.geometry.Geometry;
import mx.controls.Alert;
import mx.rpc.AsyncResponder;
import mx.utils.ObjectUtil;
private var driveTimes:String = "1 2 3";
private function computeServiceArea( event : MouseEvent ) :
void
{
    graphicsLayer.clear();
    var mapPoint : MapPoint = myMap.toMapFromStage(event.stageX, event.stageY);
    var graphic : Graphic = new Graphic(mapPoint, sms_circleAlphaSizeOutline);
    graphicsLayer.add(graphic);
    var featureSet:FeatureSet = new FeatureSet([graphic]);
    var params:Object = {
        "Input_Location" : featureSet,
        "Drive_Times" : driveTimes
    };
    gp.execute(params, new AsyncResponder( onResult, onFault ));
    function onResult(
        gpResult : ExecuteResult,
        token : Object = null
    ) : void
    {
        var pv : ParameterValue = gpResult.parameterValues[0];
        var fs : FeatureSet = pv.value as FeatureSet;
        graphicsLayer.graphicProvider = fs.features;
    }
    function onFault( info : Object, token : Object = null ) : void
    {
        Alert.show( info.toString() );
    }
}
private function fillFunc( g : Graphic ) : Symbol
{
    var toBreak : Number = g.attributes.ToBreak;
    if ( toBreak == 1 )
    {
        return rFill;
    }
    if ( toBreak == 2 )
    {

```

```

        return gFill;
    }
    return bFill;
}
]]>
</mx:Script>
<esri:SimpleMarkerSymbol id="sms_circleAlphaSizeOutline" alpha="0.5" size="15" style="circle"/>
<esri:SimpleFillSymbol id="rFill" alpha="0.5" color="0xFF0000"/>
<esri:SimpleFillSymbol id="gFill" alpha="0.5" color="0x00FF00"/>
<esri:SimpleFillSymbol id="bFill" alpha="0.5" color="0x0000FF"/>
<esri:Map id="myMap" click="computeServiceArea(event)" openHandCursorVisible="false">
    <esri:extent>
        <esri:Extent xmin="-95.41" ymin="38.86" xmax="-95.1" ymax="39.06"/>
    </esri:extent>
    <esri:ArcGISTiledMapServiceLayer
        url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_StreetMap
        _World_2D/MapServer"/>
    <esri:GraphicsLayer id="graphicsLayer" symbolFunction="fillFunc"/>
</esri:Map>
<esri:Geoprocessor id="gp"
    url="http://sampleServer1.ArcGISOnline.com/ArcGIS/rest/services/Network
    /ESRI_DriveTime_US/GPService/CreateDriveTimePolygons"/>
</mx:Application>

```


9. WebServices 的使用

Flex 本身对 webservices 有着良好的支持，我们可以调用互联网上的各种 webservices 来结合 esri 的 map 做出自己想要的东西我们就拿现在比较流行的天气预报来做例子，我们要实现的目标就是通过 webservices 查询到武汉的天气情况，然后使用 infowindow 显示到 esri 的 map 上，在讲解过程中，我们会重点解释 infowindow 的用法。现在提供天气预报的 webservices 很多，我们选用 <http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?>来进行天气的查询。

那么，在 Flex 中如何使用 webservices 呢？

只要使用 <mx:WebService>标签创建一个 WebService 就可以了，id 唯一标识这个 webservice，wsdl 指向提供 webservice 的地址。

```
<mx:WebService id="weatherWS"
wsdl="http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL"
showBusyCursor="true"/>
```

我们直接在 application 创建完成的时候调用这个 webservice，得到结果后直接显示到 map 上。

```
private function Init():void
{
    weatherWS.addEventListener(ResultEvent.RESULT, WSGetWeatherResult);
    weatherWS.getWeatherbyCityName("武汉");
}
private function WSGetWeatherResult(event:ResultEvent):void
{
    weatherWS.removeEventListener(ResultEvent.RESULT, WSGetWeatherResult);
    var arrC:ArrayCollection =event.result as
ArrayCollection;
    if(arrC.length > 0)
    {
        var str:String = arrC.getItemAt(0).toString();
        // var infWd :InfoWindow = new InfoWindow(EsriMap);
        var str2:String = arrC.getItemAt(1).toString();
        myMap.infoWindow.title= str+"."+str2;
        var vbox :VBox = new VBox();
        var vbox2 :VBox = new VBox();
        var hbox :HBox = new HBox();
        var canvas:Canvas = new Canvas();
        var path:String = "assets\\weather\\";
```

```
var img1 :Image = new Image;
img1.load(path+arrC.getItemAt(8).toString());
hbox.addChild(img1);
var img2 :Image = new Image;
img2.load(path+arrC.getItemAt(9).toString());
hbox.addChild(img2);

var txtTem :Text = new Text();
txtTem.text = arrC.getItemAt(5).toString();
var txtWea :Text = new Text();
txtWea.text = arrC.getItemAt(6).toString();
var txtWind :Text = new Text();
txtWind.text = arrC.getItemAt(7).toString();
vbox.addChild(txtTem);
vbox.addChild(txtWea);
vbox.addChild(txtWind);
vbox2.addChild(hbox);
vbox2.addChild(vbox);
canvas.addChild(vbox2);

myMap.infoWindow.content = canvas;
var mapPnt2:MapPoint = new MapPoint
(114.1547298,30.5127677);
myMap.infoWindow.show(mapPnt2);
myMap.centerAt(mapPnt2);

}
}
```

设置 infoWindow 有两点需要注意，一是 infoWindow 不需要自己 new 一个，只需要把你要显示的内容赋值给 myMap.infoWindow，二是显示的时候只需要调用 myMap.infoWindow.show(mapPnt2);就可以了，把你想显示的位置告诉他。

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    pageTitle="Service Area"
    xmlns:esri="http://www.esri.com/2008/ags"
    layout="absolute"
    creationComplete="Init()"
>
<mx:Script>
    <![CDATA[
        import mx.containers.HBox;
        import mx.containers.VBox;
        import mx.controls.Image;
```

```
import mx.rpc.events.ResultEvent;
//导入 flash 的命名空间
import mx.collections.ArrayCollection;
    import mx.controls.Alert;
    import mx.controls.Button;
    import mx.rpc.AsyncResponder;
    import flash.geom.Point;
    import mx.rpc.events.FaultEvent;
    import mx.printing.FlexPrintJobScaleType;
    import mx.printing.FlexPrintJob;
    import mx.rpc.soap.xml.Operation;
    import mx.containers.Canvas;
        import flash.text.*;
        import mx.controls.Text;

    //导入 esri Flex API
import com.esri.ags.geometry.MapPoint;
    import com.esri.ags.Graphic;
    import com.esri.ags.events.DrawEvent;
    import com.esri.ags.events.ExtentEvent;

    import com.esri.ags.SpatialReference;
    import com.esri.ags.layers.GraphicsLayer;
    import com.esri.ags.*;
    import com.esri.ags.InfoWindow;
    import com.esri.ags.InfoTemplate;
private function Init():void
{
    weatherWS.addEventListener(ResultEvent.RESULT, WSGetWeatherResult);
    weatherWS.getWeatherbyCityName("武汉");
}
private function WSGetWeatherResult(event:ResultEvent):void
{
    weatherWS.removeEventListener(ResultEvent.RESULT, WSGetWeatherResult);
    var arrC:ArrayCollection = event.result as ArrayCollection;
    if(arrC.length > 0)
    {
        var str:String = arrC.getItemAt(0).toString();
//        var infWd :InfoWindow = new InfoWindow(EsriMap);
        var str2:String = arrC.getItemAt(1).toString();
        myMap.infoWindow.title= str+"."+str2;
        var vbox :VBox = new VBox();
        var vbox2 :VBox = new VBox();
```

```
var hbox :HBox = new HBox();
var canvas:Canvas = new Canvas();
var path:String = "assets\\weather\\";
var img1 :Image = new Image;
img1.load(path+arrC.getItemAt(8).toString());
hbox.addChild(img1);
var img2 :Image = new Image;
img2.load(path+arrC.getItemAt(9).toString());
hbox.addChild(img2);
var txtTem :Text = new Text();
txtTem.text = arrC.getItemAt(5).toString();
var txtWea :Text = new Text();
txtWea.text = arrC.getItemAt(6).toString();
var txtWind :Text = new Text();
txtWind.text = arrC.getItemAt(7).toString();
vbox.addChild(txtTem);
vbox.addChild(txtWea);
vbox.addChild(txtWind);
vbox2.addChild(hbox);
vbox2.addChild(vbox);
canvas.addChild(vbox2);

myMap.infoWindow.content = canvas;
var mapPnt2:MapPoint = new MapPoint(114.1547298,30.5127677);
myMap.infoWindow.show(mapPnt2);
myMap.centerAt(mapPnt2);
}
}
]]>
</mx:Script>
<esri:Map id="myMap" openHandCursorVisible="false">
  <esri:extent>
    <esri:Extent xmin="113" ymin="30" xmax="115" ymax="32"/>
  </esri:extent>
  <esri:ArcGISTiledMapServiceLayer
    url="http://Server.ArcGISOnline.com/ArcGIS/rest/services/ESRI_StreetMap
      _World_2D/MapServer"/>
  </esri:Map>
</mx:WebService id="weatherWS"
  wsdI="http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL"
  showBusyCursor="true"/>
</mx:Application>
```