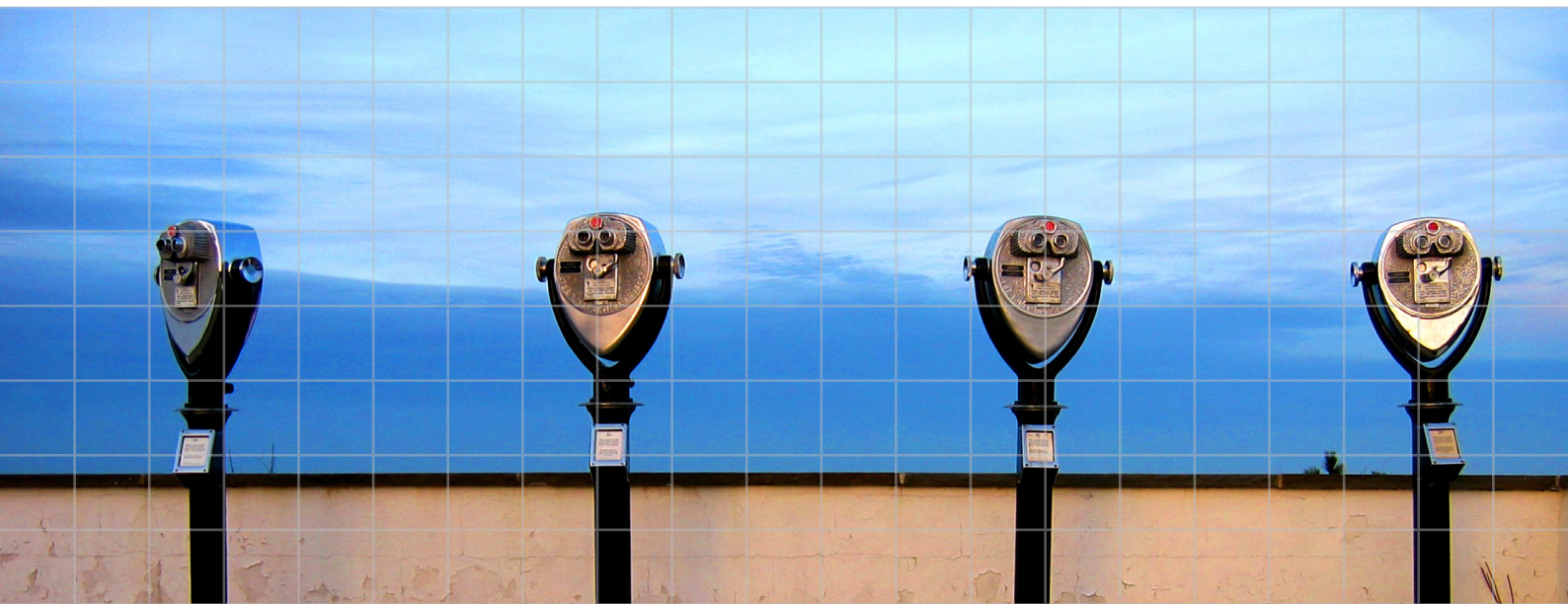


A Beginner's Guide to extending the ArcGIS Viewer for Flex



Introduction

ArcGIS Viewer for Flex is a free web-based client for viewing data on ArcGIS Server and interacting with other resources such as geoprocessing services and geocoding services. Bespoke functionality is easily added to the core Viewer application by writing custom modules of code called “widgets” and installing them in the Viewer. This guide provides non-developers with a high-level overview of how to build custom widgets using the Flex development framework, and also links to more advanced documentation for further reading.

Using the ArcGIS Viewer for Flex

The core Flex Viewer application is built on the ArcGIS API for Flex and displays an interactive basemap inside a Web browser using Adobe Flash Player. Additional widgets (for example the Navigation widget on the left for zooming and panning) are overlaid on the main map or inserted into the widget tray in the top-right of the screen, as shown in figure 1.

By editing an XML-formatted configuration file, widgets can be added, configured, and removed from the Viewer. Customising Flex Viewer in this way is straightforward and does not require any programming knowledge.

Part of an XML configuration file (config-all.xml) is shown in figure 2.

The configuration file shows a list of different widgets (“Bookmarks”, “Draw and Measure”, etc) that are placed inside a horizontal “widget container”, as seen in the top-right of the Flex Viewer screenshot above.

To remove a widget from the Viewer, we can delete all references to it in the configuration file, save our changes and then reload the Viewer in our Web browser.

To see an example of the Flex Viewer in action, go to:

<http://help.arcgis.com/en/webapps/flexviewer/live/index.html?config=config-all.xml>

By specifying **config=config-all.xml** in the URL, the Viewer will be loaded using the **config-all.xml** file. The complete **config-all.xml** configuration file can be seen at:

<http://help.arcgis.com/en/webapps/flexviewer/live/config-all.xml>

You can also try loading the Flex Viewer with a different configuration (e.g. **config-thematic.xml**):

<http://help.arcgis.com/en/webapps/flexviewer/live/index.html?config=apps/config-thematic.xml>

The **config-thematic.xml** file can be accessed at:

<http://help.arcgis.com/en/webapps/flexviewer/live/apps/config-thematic.xml>

This demonstrates how easy it is to customise the appearance and functionality of Flex Viewer by editing a single configuration file.

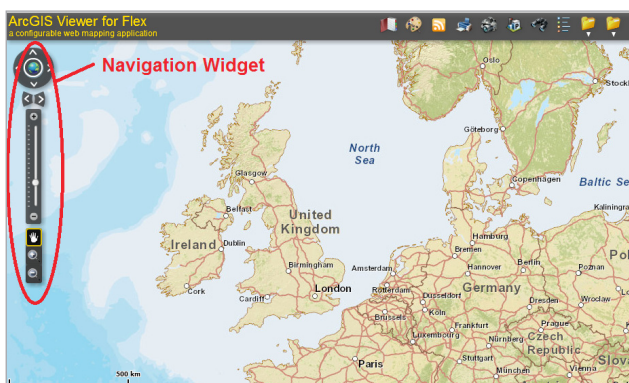


Figure 1

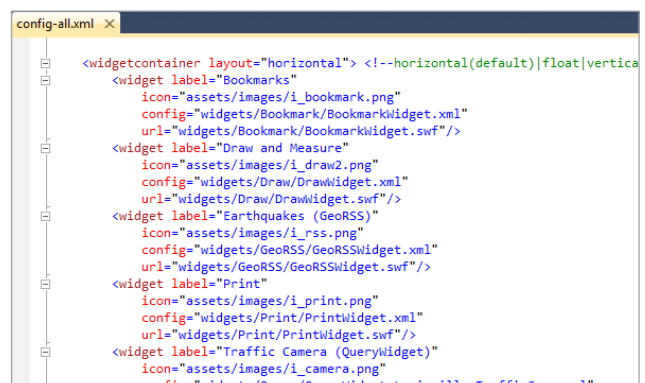


Figure 2

Downloading the ArcGIS Viewer for Flex

The latest version of Flex Viewer (with source code) can be downloaded from

<http://help.arcgis.com/en/webapps/flexviewer/index.html>

It is important to download the source code, not the compiled and ready-to-deploy package, if you are intending to develop your own widgets. There is no need to download the Flex API package separately as this is included in the Flex Viewer download.

Inside the **src** folder of the Flex Viewer download, there are sample configuration files that you can use as templates to create your own (figure 3).

Each widget has its own folder inside the **widgets** directory (figure 4).

A widget's source code is typically written in a combination of two different languages: **MXML** and **ActionScript**.

These are similar to HTML and JavaScript respectively: MXML describes the static properties of a widget, such as its appearance (the layout of buttons, menus, etc), whereas ActionScript is used to describe the program's behaviour, such as the actions that happen when a button is clicked.

As an example, in the **Coordinate** widget's directory there are three files (figure 5).

The **CoordinateWidget.xml** document is used to configure this individual widget (the XML configuration

files seen earlier were for configuring the core Viewer).

The **DegToDMS.as** file contains ActionScript code, and the **CoordinateWidget.mxml** file contains MXML markup.

In many of the sample widgets, there is only a single **MXML** file and no separate **AS** file because ActionScript code can be embedded inside an MXML file, and it is often convenient to do so. However, the MXML and ActionScript are sometimes placed in different files so that the ActionScript code can be maintained separately. For larger widgets, there may also be multiple MXML and AS files in the widget directory.

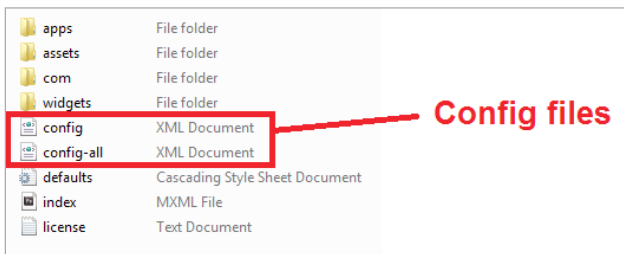


Figure 3

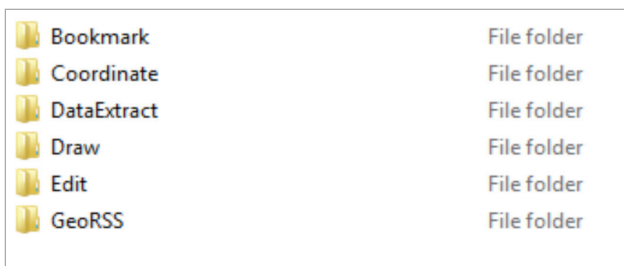


Figure 4

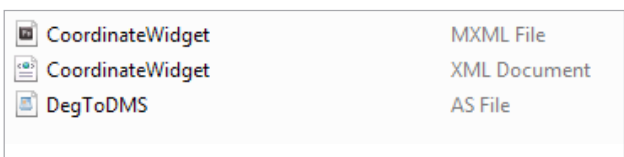


Figure 5

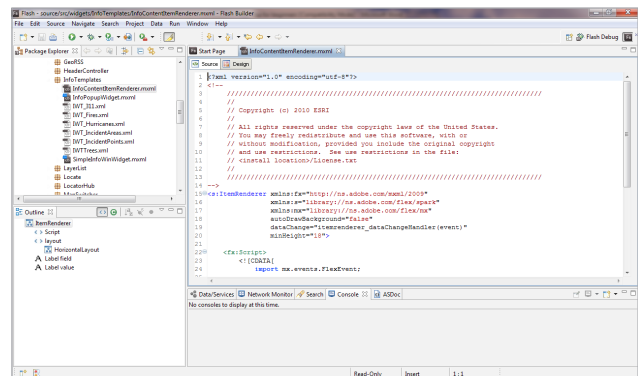


Figure 6

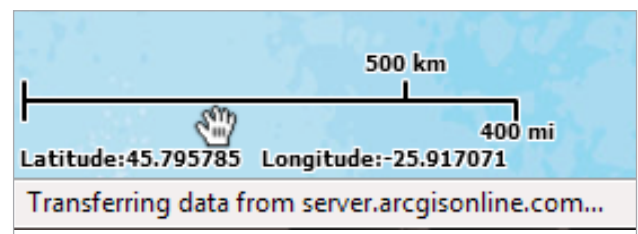


Figure 7

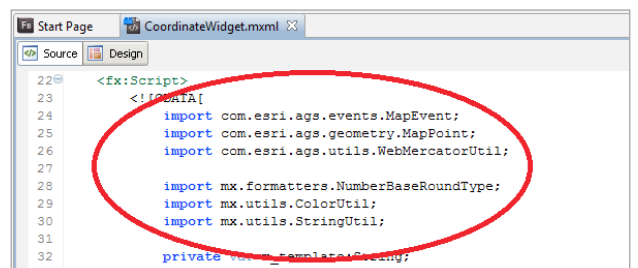


Figure 8

Start Developing

Esri, Inc. strongly recommend using the commercial **Flash Builder 4** software to develop Flex Viewer widgets. A free 60-day trial of the software (figure 6) is available from:

http://www.adobe.com/go/try_flashbuilder

Flash Builder 4, shown above, comes with the Flex development framework, a set of software tools to author Flash content. The ArcGIS API for Flex and the ArcGIS Viewer for Flex (which is built on the API for Flex) both utilise Flex.

Flash Builder provides a convenient graphical user interface to edit a widget's MXML and ActionScript source code and automatically build it into a **.swf** (pronounced "swiff") file. A **.swf** file contains compiled Flash content that can be run in a client's web browser using Adobe Flash Player, and it is loaded by Flex Viewer when a user tries to access your widget.

The development process is essentially:

1. Write or modify MXML/ActionScript code
2. Build the code into a **.swf** file
3. Load the **.swf** file in a browser and test that it works
4. Repeat

To illustrate the structure of an **.mxm** file, we will look at a simple widget that comes bundled with the core Flex Viewer: the **Coordinate** Widget. This widget tracks the user's mouse cursor (the "hand" icon) as it moves across the basemap, and displays the hand's current coordinates (latitude/longitude) in the bottom-left corner of the map (figure 7).

CoordinateWidget.mxml is the main source file for this widget, and it contains ActionScript code embedded inside MXML markup. It is a typical example of an MXML/ActionScript code file, and its structure can be divided into three sequential parts:

1. Import libraries
2. ActionScript code (program logic)
3. MXML (presentation markup)

In the first part, we import program libraries. A **library** is a re-usable module of code whose functionality can be used inside other software modules (figure 8).

For example, **CoordinateWidget** imports the **com.esri.ags.util**. **WebMercatorUtil** library, which contains functionality for working with the Web Mercator map projection.

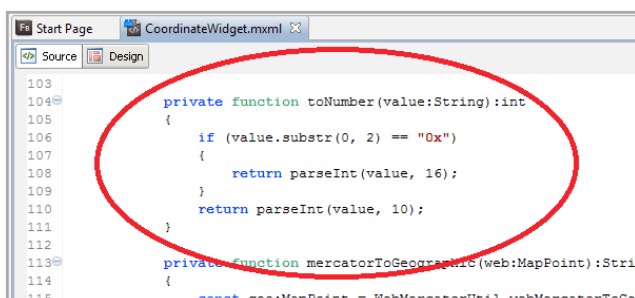
After the libraries have been imported, the program logic (the behaviour of the widget when the user moves the mouse, clicks various buttons, etc) is specified. Logic consists of **variables** that store data and chunks of code called **functions** that manipulate data. For example, an ActionScript function for converting a piece of text into a number is shown in figure 9.

Non-developers or developers new to Flex development are strongly encouraged to read Chapter 2 of Adobe's Learning ActionScript® 3.0 guide, available for download at the link below.

http://help.adobe.com/en_US/as3/learn/as3_learning.pdf (2.2 MB file)

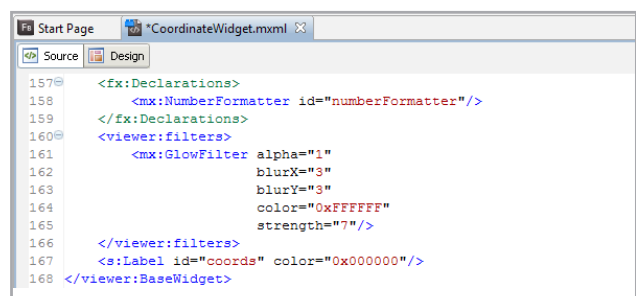
Chapter 2: Getting started with ActionScript covers basic programming concepts in an accessible manner. Since we use the **Flex** toolkit and **Flash Builder** software to develop Flex Viewer widgets, it is safe to ignore any references to other Flash authoring software such as Flash Professional.

Finally, the third section contains MXML markup that specifies the appearance of the widget (figure 10)



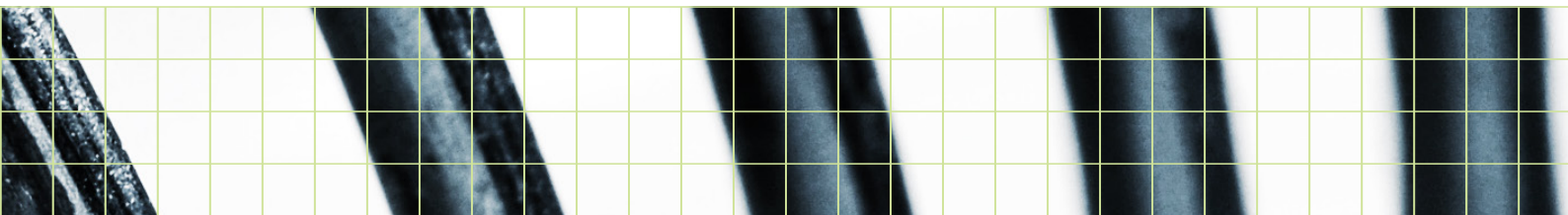
```
103
104
105     private function toNumber(value:String):int
106     {
107         if (value.substr(0, 2) == "0x")
108         {
109             return parseInt(value, 16);
110         }
111         return parseInt(value, 10);
112     }
113
114     private function mercatorToGeographic(web:MapPoint):Stri
115     {
116         const geo:MapPoint = WebMercatorUtil.webMercatorToGe
```

Figure 9



```
157 <fx:Declarations>
158     <mx:NumberFormatter id="numberFormatter"/>
159 </fx:Declarations>
160 <viewer:filters>
161     <mx:GlowFilter alpha="1"
162         blurX="3"
163         blurY="3"
164         color="0xFFFFFFFF"
165         strength="7"/>
166 </viewer:filters>
167 <s:Label id="coords" color="0x000000"/>
168 </viewer:BaseWidget>
```

Figure 10



Interacting with geoprocessing services

Coordinate was a simple widget that did not involve interacting with server-side resources such as geoprocessing services.

The general workflow for creating a geoprocessing widget is as follows:

1. Create a geoprocessing model or tool (e.g. using ModelBuilder or a Python script)
2. Publish the model and associated data as a geoprocessing service on ArcGIS Server
3. Build a Flex widget to collect input parameters from a user and send it to the geoprocessing service using a REST endpoint (a URL)
4. Collect the output of the geoprocessing service and draw it on a map

As an example, the **Data Extract** widget comes bundled with Flex Viewer and allows the user to select data from the current map and download it as a file geodatabase (in a .zip file). It interacts with the Clip and Ship geoprocessing service, which is executed asynchronously.

A simpler example is the **ViewShed** widget from the ArcGIS Viewer for Flex Code Gallery:

<http://help.arcgis.com/en/webapps/flexviewer/gallery.html>

ViewShed is a synchronous geoprocessing service that calculates the visible extent from a particular point on a map. The source code demonstrates how a user's input (i.e. a click on the map) is collected in the widget and submitted to the geoprocessing service on Server, and how the viewshed polygon is drawn when it is received.

Using ArcGIS API for Flex code samples

The ArcGIS Viewer for Flex is built on the ArcGIS API for Flex and adds a framework for easily configuring Flex applications without having to rebuild them every time a change is made. Because both the Viewer and the API use the Flex framework to communicate with ArcGIS Server, much of the ActionScript code in the ArcGIS API for Flex code samples can be re-used in Flex Viewer widgets.

The ArcGIS API for Flex code samples are available at:

<http://help.arcgis.com/en/webapi/flex/samples/index.html>

Note that the MXML markup in the Flex API code samples is not entirely compatible with Flex Viewer, although the ActionScript code can be re-used without change.

Further Reading and Resources

This guide introduced general Flex Viewer concepts using existing widgets as illustrative examples. Hopefully you will now be comfortable reading some of the more developer-focused documentation available on the web.

For a step-by-step guide to using and developing widgets for the ArcGIS Viewer for Flex, you can start with ArcGIS.com:

<http://help.arcgis.com/en/webapps/flexviewer/help/index.html>

Adobe Developer Documentation

If you are a first-time programmer, Adobe's Learning ActionScript® 3.0 is highly recommended. It is not specific to Esri technology, and Chapter 2 is especially recommended for beginners:

http://help.adobe.com/en_US/as3/learn/as3_learning.pdf (2.2 MB file)

A more advanced developer's reference is the ActionScript® 3.0 Developer's Guide. This is an extremely long document (1,000+ pages) and is not meant to be read from start to finish in one sitting. However, it is sensibly divided into many different chapters that cover virtually every aspect of development using the Flex framework:

http://help.adobe.com/en_US/as3/dev/as3_devguide.pdf (17.2 MB file)

Flex code documentation

The official ActionScript 3.0 language reference is a guide to various classes and methods you will need:

http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/class-summary.html

ArcGIS.com has equivalent documentation for its Flex classes:

<http://help.arcgis.com/en/webapi/flex/apiref/index.html>

Esri UK: visionary thinking

Esri UK is the leading provider of geographic information system (GIS) technology, helping businesses become more profitable and public service more efficient through a better understanding and analysis of location-based information. Esri UK offers an extensive range of GIS technology and professional services and is the only company in the UK providing a complete and entirely integrated GIS solution.

Esri UK has been providing GIS solutions for over 20 years to a wide range of markets including Central & Local Government, Commercial, Defence & National Security, Education, Public Safety, Utilities & Telecommunications, as well as catering for system integrators and application developers through the Esri Developer Network.

Customers include Birmingham City Council, Defra, the Environment Agency, Metropolitan Police Service, Ministry of Defence, Ordnance Survey, RSA Group, Scottish Power and The Crown Estate.

Esri UK is part of the Esri global business network with more than 300,000 customers in 150 countries worldwide. Esri technology runs on more than one million desktops and thousands of web and enterprise servers, providing the backbone for the world's mapping and spatial analysis.

Esri UK helps organisations to think and plan geographically in order to make better decisions, keep communities safe and create a more sustainable world.

It's what we call **Visionary Thinking**.

For more information, visit www.esriuk.com