Elevation Profile Widget for ArcGIS Viewer for Flex

# Introduction

Here are instructions for using the accompanying widget source and object code files for creating elevation profile charts. This widget has been tested and found to work with Viewer 2.1 through 2.5.

It also works with Viewer 3.1 – 3.6, including the Application Builder. See the end of this document for an addendum on using the widget with Viewer 3.

These instructions assume you have successfully set up a running instance of Viewer 2.x; you can find assistance here:

<http://help.arcgis.com/en/webapps/flexviewer/help/gettingstarted.htm>

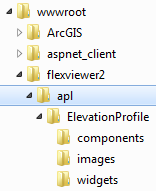
# Installation into Viewer instance

You may install and use the widget without loading it into the Flash Builder development environment or compiling it. This section describes how to add the supplied, compiled modules into an existing Viewer instance. These instructions assume you have a running instance of the Viewer 2 application, beta 2 or above. You may obtain it here:

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

**Installing into Viewer 2.1 – 2.4:**

Copy the supplied directory “apl” into your Viewer instance directory. Note that there are two versions included: use the one underneath “**2.1 – 2.3.1 > runtime**” if you’re using Viewer 2.1 through 2.3.1; otherwise, use the one under “**2.4 > runtime**” if you’re using FlexViewer 2.4. Do not use the one underneath “code” unless you’re working with the project in the Flash Builder development environment (see the section [Installation – Source Code](#_Hlk312229397) below).



Now you’ll need to make sure there’s an entry for the widget in the main config file. The easiest way is to use the one supplied underneath the proper version’s “runtime” directory: copy it over your existing “config.xml” file.

Note that the only way the supplied config.xml differs from the one distributed with the Viewer is the addition of the following entry in the “widgetcontainer” section:

<widget label="Elevation Profiles" y="350" x="20"

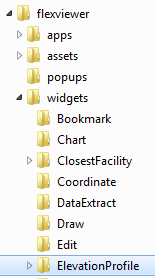
icon="apl/ElevationProfile/images/i\_widget.png"

url="apl/ElevationProfile/widgets/ElevationProfileWidget.swf"

config="apl/ElevationProfile/widgets/ElevationProfileWidget.xml"/>

**Installing into Viewer 2.5:**

With the latest version, the Elevation Profile widget has been relocated to live alongside all other Viewer widgets, in the main “widgets” directory. Copy the supplied “2.5 > runtime > widgets” directory into your FlexViewer location. Again, do not use the one underneath “code” unless you’re planning to change and recompile the code using Flash Builder (see [Installation – Source Code](#_Hlk312229397) below).



Make sure there’s an entry for the widget in the FlexViewer main config file. The easiest way is to use the one supplied underneath the “2.5 > runtime” directory: copy it over your existing “config.xml” file.

The only way the supplied config.xml differs from the one distributed with FlexViewer is the addition of the following entry in the “widgetcontainer” section:

<widget label="Elevation Profiles" y="350" bottom="20"

icon="widgets/ElevationProfile/images/i\_widget.png"

url="widgets/ElevationProfile/ElevationProfileWidget.swf"

config="widgets/ElevationProfile/ElevationProfileWidget.xml"

preload="open" />

**Testing the widget**

Now that the main config.xml points to the new widget, open the viewer in a browser. You may need to refresh the browser cache and reload to see the Elevation Profile widget appear. Click one of the widget’s line drawing tools and draw a line on the map. After a few seconds, you should see a corresponding profile chart appear in a pop-up window.

**NOTE**: If you draw a line that crosses itself (self-intersects), the line will be topologically simplified on the server; the resulting geometric paths sent back may be out of order and the chart could be invalid.

Move the mouse over the chart’s profile line. Where there are available data points, a tooltip should pop up to provide elevation and distance-along-profile information. Also, you should see a cross graphic along the line you drew in the map, indicating the geographic location of the data point the mouse is highlighting.

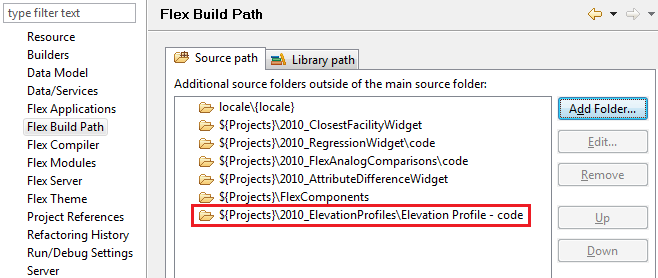
# Installation – Source Code

Source code for this widget is available under the supplied directory “Elevation Profile - code”. This code is not a standalone project, but is meant for inclusion into an existing FlexViewer 2 project. First, make sure the Viewer project is open in Flash Builder.

First, you’ll need to add the source directory to the project build path. Open the project properties in Flash Builder; then choose the “Flex Build Path” item. Under the “source path” tab, click “Add Folder…” and browse to the location of the supplied code directory.

|  |  |
| --- | --- |
| v2.1 – 2.3.1, 2.4: | v2.5: |
|  |  |

Once you click OK, it should show up in your project’s build path list:



Next, you must add two items to the project’s modules list using the Project Properties > Flex Modules dialog page. You’ll need to add them one at a time, using the “Add…” button and dialog.

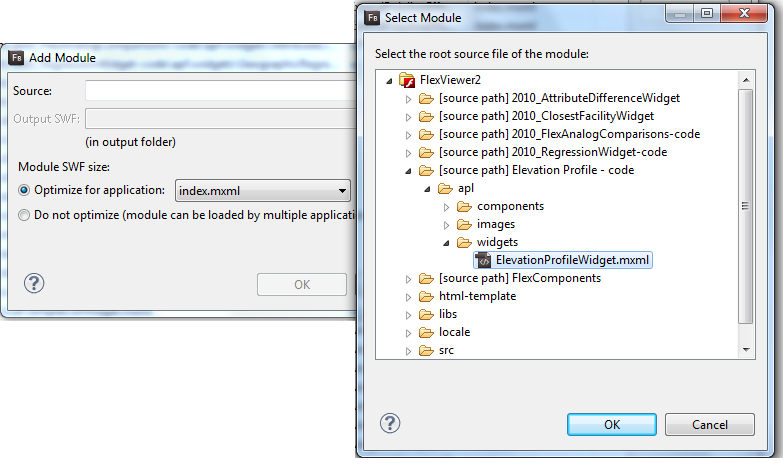
**v2.1 – 2.4:**

* Elevation Profile - code\apl\ElevationProfile\widgets\ElevationProfileWidget.mxml
* Elevation Profile - code\apl\ElevationProfile\components\ChartWindow.mxml

**v2.5:**

* Elevation Profile - code\widgets\ElevationProfile\ElevationProfileWidget.mxml
* Elevation Profile - code\widgets\ElevationProfile\components\ChartWindow.mxml

You can choose to have these modules optimized for the index.html project, or compile them unoptimized. Since these modules likely aren’t useful in any application other than FlexViewer, it’s better to have them optimized.



# Configuration File

The configuration file for the widget (as opposed to the main Viewer configuration file) is located in the same directory as the .swf or .mxml file. It’s quite basic, containing only a very few options.

* elevationSvcURL: The URL for the elevation service that provides the elevation data for this widget.
* baseChartAtZero: By default, the chart will always use zero as the base elevation for the chart. Changing this to “false” will instead use the minimum value along the user-drawn line or curve. This is useful for viewing a small geographic area that features only small elevation changes throughout the area’s extent. **This option is only available in the v2.5 version of the component.**
* profileLineWidth: The width (in pixels) of the user-drawn line for which an elevation profile chart will be generated.
* profileLineColor: The hexadecimal RBG color of the user-drawn elevation profile line.
* description: A descriptive blurb of text displayed in the widget, directing the user how to use the widget.
* chartTitle: The title displayed at the top of the elevation profile chart window.
* unitsAndLabeling: Titles and conversion factors for units of distance. This lets you define your own units for both elevation and distance-from-start. Provided are sections for meters/kilometers and for feet/miles. Only one <unitsAndLabeling> section should be uncommented. There’s also a commented template for creating your own section.

# Addendum: Usage with Viewer version 3

The distribution file now includes new sub-archives, “3.x (AppBuilder compatible)”. This will need to be extracted from its containing archive and saved to your disk as a standalone .zip file. Then the Viewer Application Builder’s “Manage Widgets” utility can be pointed to this .zip file in order to import the widget. That archive contains the source code and compiled widget as well; you may use them with Flash Builder and Viewer 3 just as with version 2.x, as described above.

# Update History

v3.6.1 (3/13/2014):

* Updated to use the new elevation profile geoprocessing service instead of the older elevation server object extension. For more information, see this page: <https://developers.arcgis.com/rest/elevation/>

v3.6 (2/7/2014):

* Compiled for use with viewer 3.6.

v3.1.05 (2/20/2013):

* Fixed a bug that incorrectly calculated and displayed alternate elevation units (e.g. feet—see widget config xml file for more info). Thanks to Matt Shetzer of American Mapping Solutions for bringing this to my attention.
* Removed code and runtime for Flex Viewer 3.0, as it still contains the bug.

v3.1 (12/27/2012):

* Recompiled for use with viewer 3.1/Application Builder 3.1

v3.0 (7/31/2012):

* Recompiled for use with viewer 3.0
* Updated included viewer config file for use with viewer 3.0
* Added meta.xml for proper integration with Application Builder

v2.5.1 (4/25/2012):

* Added “baseChartAtZero” configuration option and supporting logic, based on ArcGIS Online user requests. This was added only to the Viewer 2.5-compatible component.

v2.5 (12/21/2011):

* Compiled for API & viewer version 2.5.
* Changed directory structure so widget now lives alongside other widgets in the “widgets” directory.

v1.2 (8/15/2011):

* Created separate runtime modules for FlexViewer 2.4.

v1.1.1 (1/14/2011):

* Added config.xml to use to replace FlexViewer’s main config.xml
* Updated readme to clarify installation process

v1.1.0 (11/23/2010):

* Updated URL
* Added freehand draw tool
* Added logic to handle multiple result geometries/paths when returned from the SOE