2011 Esri Developer Summit
Palm Springs, CA

Language Localization for the ArcGIS Viewer for Flex

Andy Gup
Derek Law

esri

Agenda

- What is localization?
- Product overview
- Localizing the Viewer (no code)
- Localizing the Viewer (developer)
- Summary

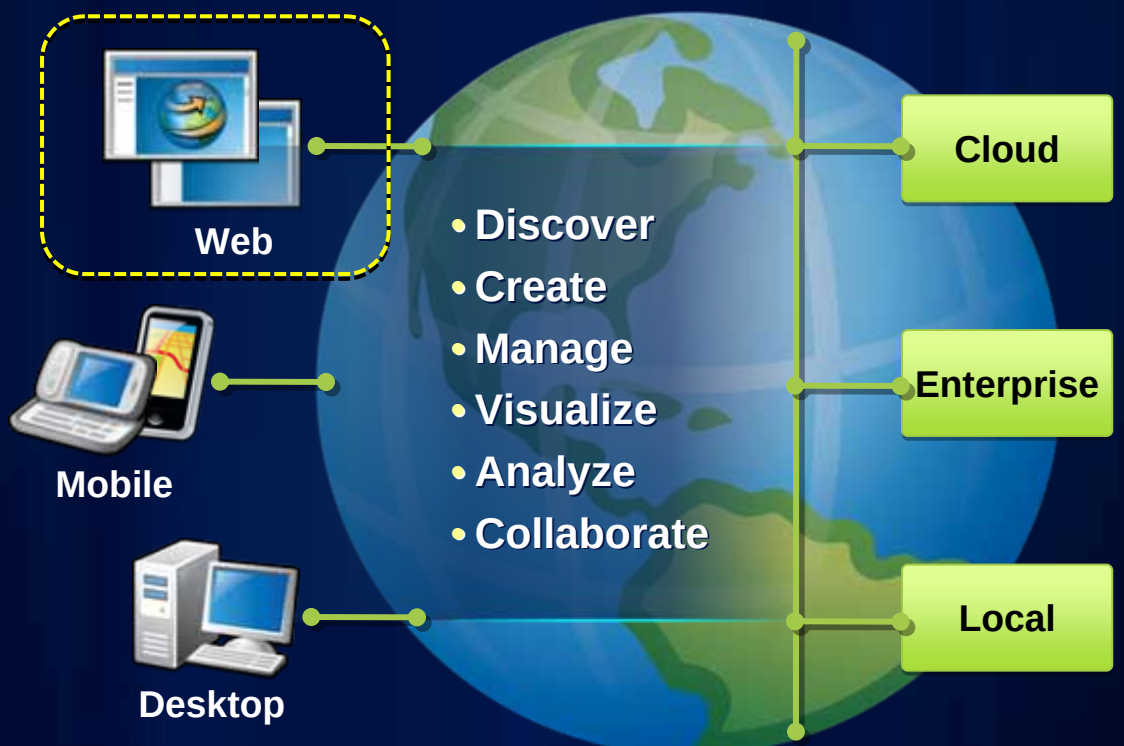
What is Localization?

- Changing the language settings of the ArcGIS Viewer for Flex
- Currently support 6 languages:
  - *Chinese, English, French, German, Japanese, & Spanish*
- 5 more coming soon:
  - *Arabic, Italian, Portuguese Brazilian, & Russian*

- Motivation:
  - International product support and adoption
  - Increase user base

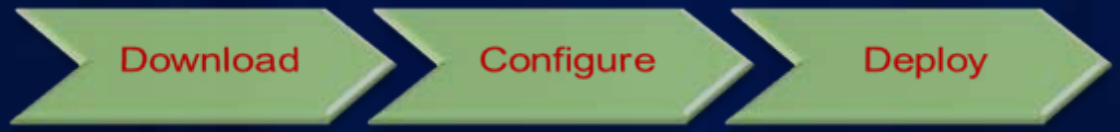
ArcGIS 10 — A Complete System

Easier
More Powerful
and Everywhere



ArcGIS Viewer for Flex

- Configurable *ready-to-deploy* Web client
  - Built with the ArcGIS API for Flex
- Easily and quickly build and implement GIS Web mapping applications



- No programming required, designed with the "non-developer" in mind
- Supported by ESRI Technical Support

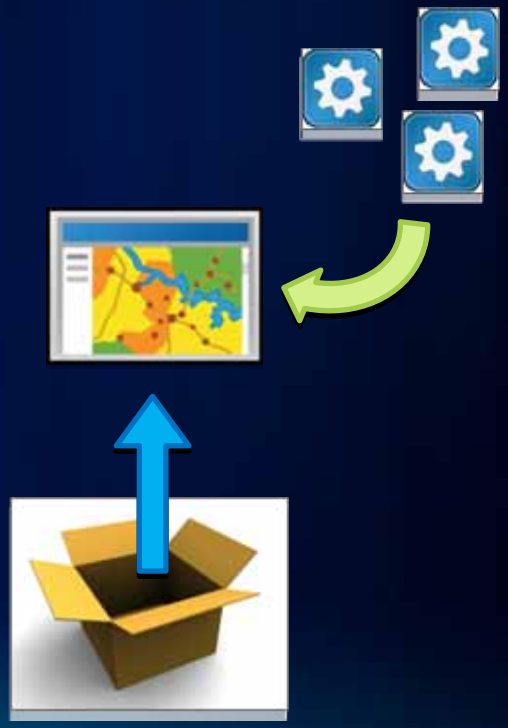
Working with the ArcGIS Viewer for Flex

"Ready-to-deploy" Web client

- Application hosted on-premise
- Browser requires Adobe Flash Player plug-in

- Configuration files enable changes to
  - Appearance
  - Functionality (via widgets)
  - Data content

- **Widgets** → modular functionality
  - Consist of
    - XML configuration file
    - Compiled Flash (.swf) file



ArcGIS Viewer for Flex – Resource Center

http://links.esri.com/flexviewer

- Download the Viewer application
- Help documentation, widget and configuration samples, and support forum

ArcGIS Viewer for Flex – Users

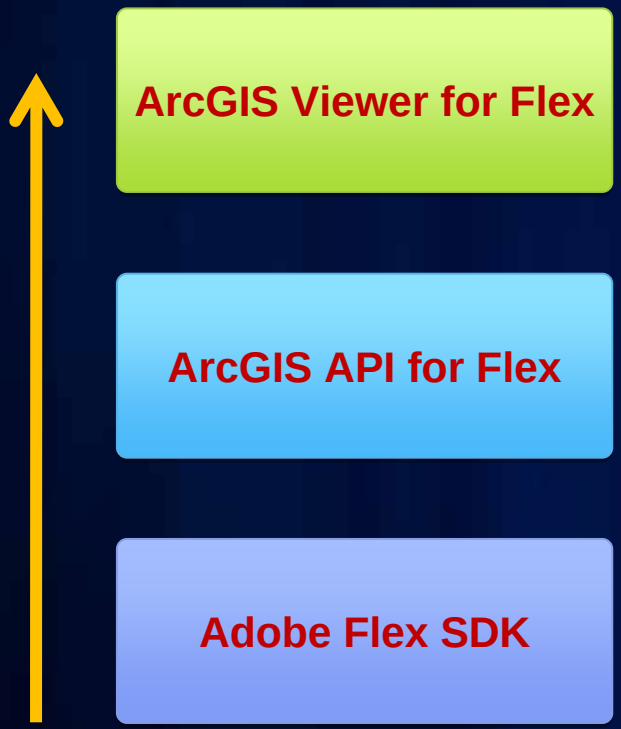
2 main groups deploy the Viewer

- **Configurators (no code)**
  - Configure and deploy the Viewer without programming
  - Leverage existing widgets
  - E.g., GIS analysts, Website designers, Managers

- **Developers (code)**
  - Extend Viewer functionality
    - Change behavior of existing widgets
    - Create new widgets
    - Extend Viewer application
  - E.g., Business partners, Website developers, Flex API developers

Viewer localization - architecture

Several different levels of localization



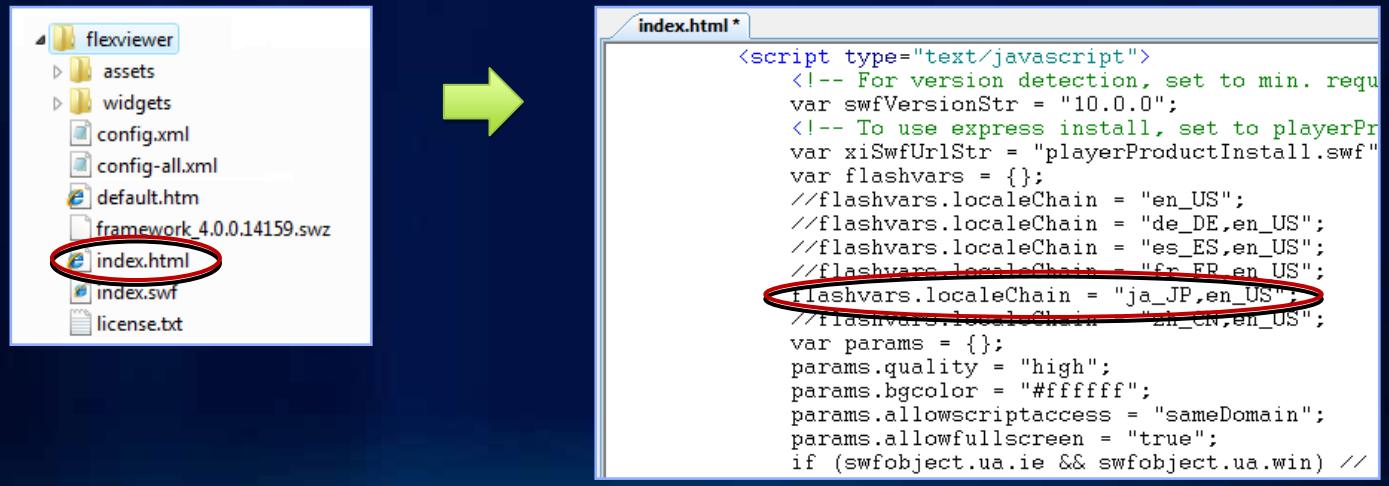
Application items:
Widget text and UI

API level items:
Draw/graphic tool labels

General items:
OK, Cancel button controls

Localizing the Viewer (no code)

- Can partially localize the Viewer, **not a complete language translation**
- Edit ***Index.html*** Viewer configuration file
  - Uncomment supported language option



Localizing the Viewer:
A Developers Perspective

Localizing the Viewer (developer)

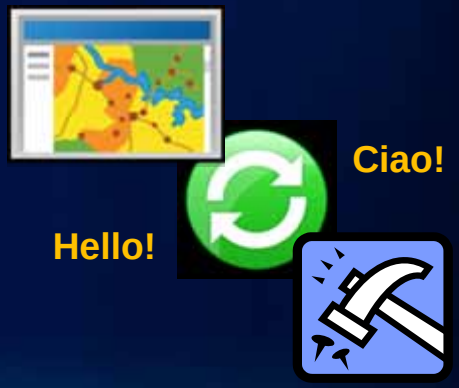
- **Requirements**
  - Adobe Flex SDK 4.0
  - Flash Player 10.1 (debug)
  - IDE (Flash Builder 4)
  - ArcGIS API for Flex (latest ver.)
  - ArcGIS 10 services

  - **Viewer source code**

    http://esriurl.com/flexviewersource

Flash Player

Flash Builder



Some things to consider:

- **How many languages to support?**
- **What type of GIS Services?**
  - **Basemaps, operational layers, etc.**
  - **Example: Locators**
- **Which User Interface (UI) items?**
  - **Which widgets?**
  - **Icons, styles, buttons, labels, fields, colors, etc.**
  - **Message text, errors, *InfoWindow* text, etc.**

Localize items that address your requirements

Considerations – not just Strings

- More than just Strings – also includes number formatting, images, currency, etc.

English (en_US)

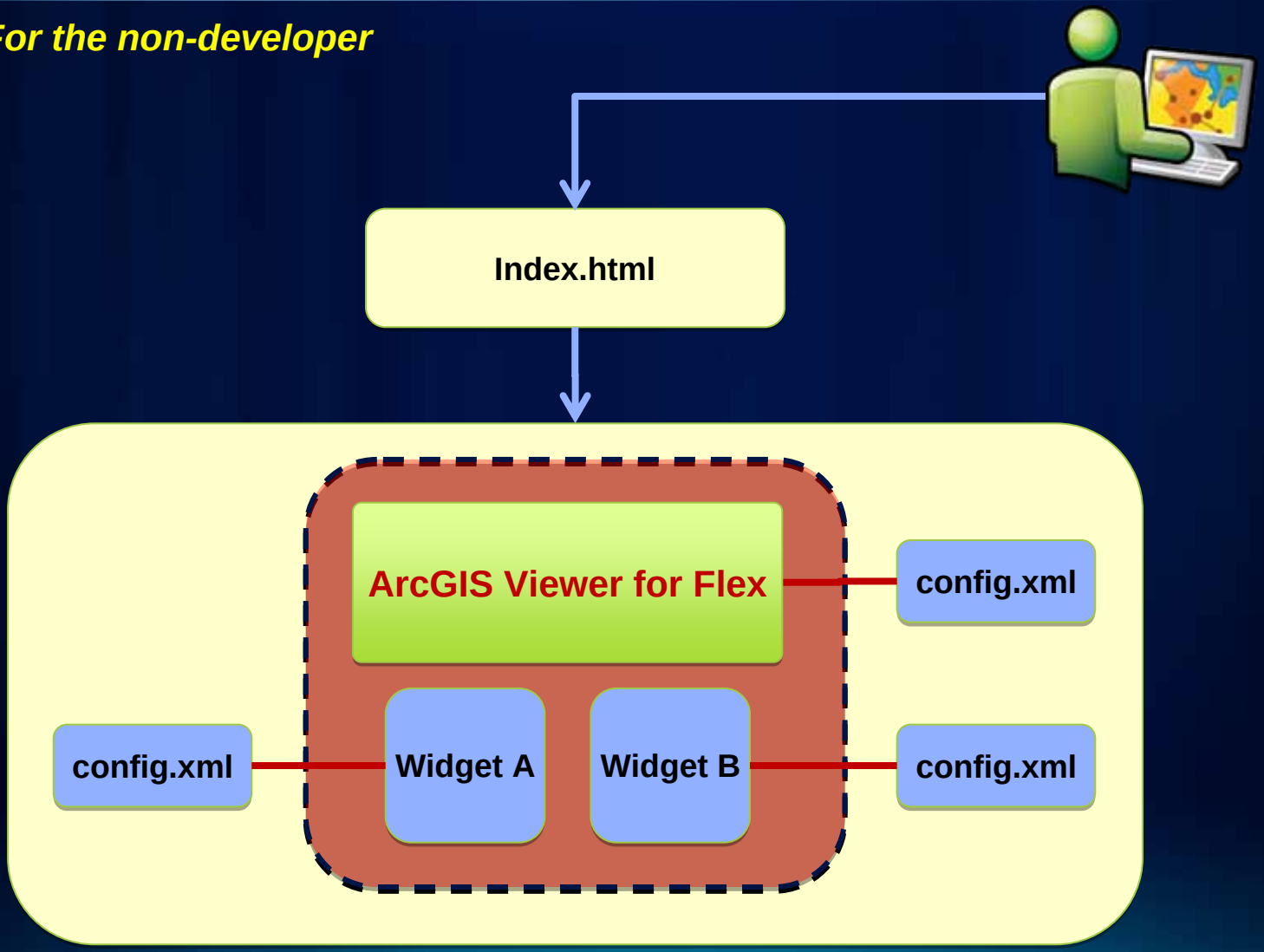
Date	02/28/11
Time	3:22 PM
Currency	$1,000.00

Spanish (es_ES)

Date	28/02/11
Time	15:22
Currency	€1.000,00

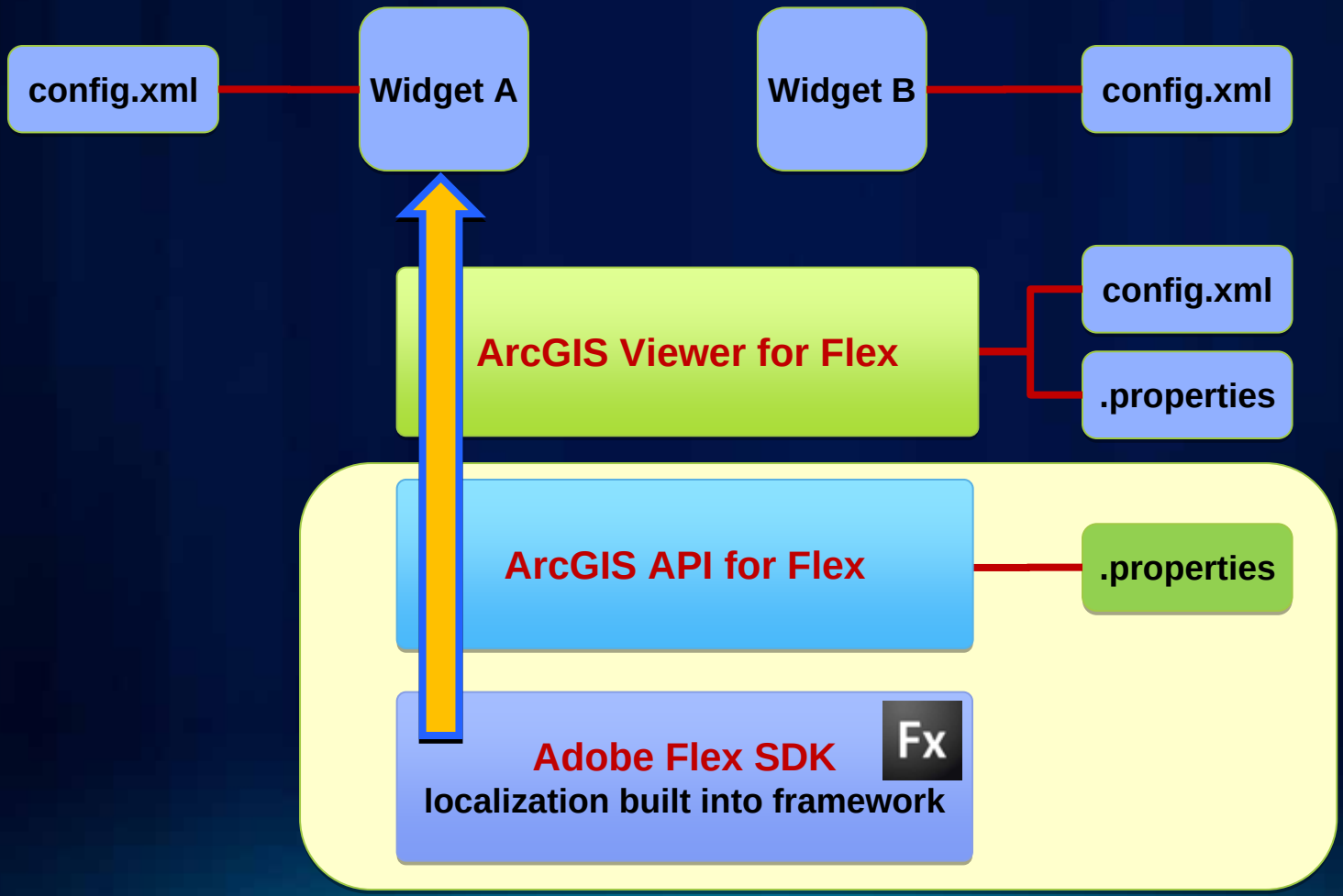
Viewer localization architecture II

For the non-developer



Viewer localization architecture III

For the developer



Steps for Localizing the Viewer

- Modify *ViewerStrings.properties* file
- Make changes to widgets
- Set Compiler Options
- Export **Release Build**
- Change *index.html* **flashvars**

Adobe Flex Localization

What is a resource property file?

- Text file that defines strings, images and other binary assets. UTF-8 and Unicode-based

Advantages of using “resource property” files

- Single location for making majority of changes
- Multiple languages --- one application!
- Easier upgrading to newer Viewer versions

Why not use *config.xml*?

- Must have separate apps for each language
- Viewer’s config.xml vs Widget’s config.xml
- How upgradeable?

ViewerStrings.properties File

```
65  # -- Draw Strings --
66  #
67  addTextLabel=\u30c6\u30ad\u30b9\u30c8\u306e\u8ffd\u52a0
68  drawPointLabel=\u30dd\u30a4\u30f3\u30c8\u306e\u63cf\u753b
69  drawLineLabel=\u30e9\u30a4\u30f3\u306e\u63cf\u753b
70  drawFreehandLineLabel=
    \u30d5\u30ea\u30fc\u30cf\u30f3\u30c9\u30e9\u30a4\u30f3\u306e\u63cf
    b
71  drawPolygonLabe\u30dd\u30ea\u30b4\u30f3\u306e\u63cf\u753b
72  drawFreehandPolygonLabel=\u30d5\u30ea\u30fc\u30cf\u30f3\u30c9
    \u30dd\u30ea\u30b4\u30f3\u306e\u63cf\u753b
73  drawRectangleLabel=\u56db\u89d2\u5f62\u306e\u63cf\u753b
74  drawCircleLabel=\u5186\u306e\u63cf\u753b
75  drawEllipseLabel=\u6955\u5186\u306e\u63cf\u753b
76  drawClearLabel=\u63cf\u753b\u3092\u6d88\u53bb\u3057\u307e\u3059
77
```

ViewerStrings.properties File II

English (en_US)

Date	02/28/11
Time	3:22 PM
Currency	$1,000.00

```
THEMECOLOR=0x0000FF
DATE_FORMAT=MM/DD/YY
TIME_FORMAT=L:NN A
CURRENCY_PRECISION=2
CURRENCY_SYMBOL=$
THOUSANDS_SEPARATOR=,
DECIMAL_SEPARATOR=.
```

Spanish (es_ES)

Date	28/02/11
Time	15:22
Currency	€1.000,00

```
THEMECOLOR=0xFF0000
DATE_FORMAT=DD/MM/YY
TIME_FORMAT=HH:NN
CURRENCY_PRECISION=2
CURRENCY_SYMBOL=€
THOUSANDS_SEPARATOR=.
DECIMAL_SEPARATOR=,
```

ViewerStrings.properties

```
private function init():void
{
    graphicsLayer = new GraphicsLayer();
    graphicsLayer.name = "Draw Features";
    map.addLayer(graphicsLayer);
    if (configXML)
    {
        // drawing labels

        // points
        drawPointLabel = configXML.labels.drawpointlabel || getDefaultString("drawPointLabel");
        addTextLabel = configXML.labels.addtextlabel || getDefaultString("addTextLabel");
        markerColorLabel = configXML.labels.markercolorlabel || getDefaultString("markerColorLabel");
        markerAlphaLabel = configXML.labels.markeralphalabel || getDefaultString("markerAlphaLabel");
        markerSizeLabel = configXML.labels.markersizelabel || getDefaultString("markerSizeLabel");
        markerStyleLabel = configXML.labels.markerstylelabel || getDefaultString("markerStyleLabel");
        markerOutlineColorLabel = configXML.labels.markeroutlinecolorlabel || getDefaultString("markerOutlineColorLabel");
        markerOutlineWidthLabel = configXML.labels.markeroutlinewidthlabel || getDefaultString("markerOutlineWidthLabel");
```

```
getDefaultString("markerOutlineColorLabel");
```

```
<mx:FormItem id="markerOutlineColorWidth"
             width="100%"
             includeInLayout="{drawingPoint}"
             label="{markerOutlineColorLabel}"
             visible="{drawingPoint}">
```

Flash Builder compiler arguments

```
-locale=ja_JP -source-path=locale/{locale}
```

Localization – Additional considerations

- **Choose appropriate font size for text**
  - May need to increase for different language
  - E.g., Asian character support
- **May need to redesign UI layout**
  - Western languages read left to right
  - Some languages are opposite, right to left
  - E.g., Arabic, Hebrew
- **Cultural aspects/UI design**
  - Try to determine Web UI best practices

Tips & tricks – Multiple Language support

- Allow end user to override
- Fixed/pre-determined (easiest)
- User selected
- ActionScript
  - `Capabilities.language` property
- HTML/JavaScript via `ExternalInterface` Class
  - `Accept-Language` header
  - `userLanguage` or `systemLanguage` property

Other Adobe Resources

- **Adobe Help Doc for localization**
  - **http://esriurl.com/adobelocalize (Basic)**
  - **http://esriurl.com/advancedadobelocalize (Advanced)**

Summary

- ArcGIS Viewer for Flex – can be localized to meet business specific needs and requirements
- There are configuration and developer approaches

esri