

# Flex 3 Localization Support

- **resources can be separated into resource modules, rather than into the application itself**
- **resources for a locale can be compiled into an application module - multiple locales can be compiled into single module**
- **resources can be accessed through a *ResourceManager***
- **binding expressions will automatically update when the locale changes**
- **components get notified when the locale changes**
- **images and sounds can be used as resources, not just plain strings**
- **you can programmatically create resources at runtime**

# Static Localization

```
<mx:Label text="@Resource(bundle='myResources', key='GREETING')"/>
```

**you can use the *@Resource()* compiler directive to specify the localized values of properties**

**the simplest way to create a bundle is to provide a *.properties* file defining the keys and values in the bundle - the MXML compiler will compile it into a subclass of *mx.resources.ResourceBundle***

**place the properties files in a the following folder *locale/{language-code}\_{COUNTRY-CODE}/* and compile with this arguments**

```
-locale=en_US -source=./locale/{locale}
```

# Creating new locales

```
bin/copylocale en_US it_IT
```

**Flex framework comes with only two framework locales - *en\_US* and *ja\_JP*, so what if we need *it\_IT*?**

# Run-time Localization

```
<mx:Label text="{resourceManager.getString('myResources', 'GREETING')}" />
```

**instead of the *@Resource()* compiler directive use data binding to the *resourceManager***

```
resourceManager.localeChain = ["en_US", "en_IN"];
```

**set the *resourceManager.localeChain* property in order to change the locale - with this property Flex support incomplete locales**

**in order to include more locales into one application SWF use the following compiler option:**

```
-locale=en_US,en_IN
```

```
<mx:Metadata>  
    [ResourceBundle("myResources")]  
</mx:Metadata>
```

**when you're not using the *@Resource()* compiler directive you need to help up the compiler with figuring out what resource bundles your application or component needs**

# What can you put in a bundle?

```
COUNTRY=United States  
PRICE=19.99;  
AGE=21  
SENIOR=true  
LOGO=Embed("logo.png")  
SORTER=ClassReference("sort.StringSorter")
```

**String, Number, Integer,  
Boolean, Image and  
Class**

# Compiling Resource Modules

```
mxm1c -locale=en_UK  
      -source-path=locale/{locale}  
      -include-resource-bundles=voting_form,thank_you_form  
      -output=bin/locale/{locale}/Resource.swf
```

# Loading Resource Modules

```
private var requestedLocale : String;
private function localeComboBox_changeHandler(event:Event):void
{
    requestedLocale = String(localeComboBox.selectedItem);
    var resourceModuleUrl : String = StringUtil.substitute(
        "locale/{0}/Resources.swf",
        localeComboBox.selectedItem);
    var dispatcher : IEventDispatcher =
        resourceManager.loadResourceModule(resourceModuleUrl, false);
    dispatcher.addEventListener(
        ResourceEvent.COMPLETE,
        handleResourceComplete);
}

private function handleResourceComplete(e : ResourceEvent) : void
{
    resourceManager.localeChain = [requestedLocale];
}
```

# Summary

- **Flex 3 localization features**
- **static localization**
- **creating new locales**
- **run-time localization**
- **compilation or resource modules**
- **run-time loading or resource modules**