

Localization using Resource Bundles

Import the Project

1. If not already created, create a directory named `adobeFlexTraining` on your **C** drive.
2. In Flex Builder, delete the **ReservationSystem** project if created previously. If you don't delete the contents the files will be overwritten with the new project files.
3. In Flex Builder, select **File > Import > Flex Project**.
4. In the dialog window, select **Archive File** and browse to where `Ex26_Starter.zip` is located in your local file system.
Note: This starter file has the code you created through the end of Day 4 exercises. This exercise will only delve into a portion of that code. For more information, review Day 4 exercises.
5. Uncheck **Use default location**.
6. Enter `C:\adobeFlexTraining\ReservationSystem`.
7. Click **Finish**.

Localization Settings

When trying to localize a flex application, we need to make sure the framework resources for that locale are created. By default, only `en_US` is created in the sdk.

(To understand how to create locales for other languages, please look at this link:

http://livedocs.adobe.com/flex/3/html/help.html?content=110n_3.html)

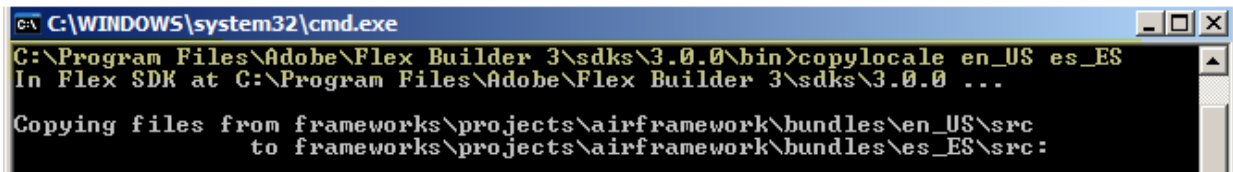
8. Open a command prompt window and navigate to the directory where Flex Builder is installed.

Note: If you are using Eclipse with Flex Builder plugin, your default path is:
`C:\Program Files\Adobe\Flex Builder 3\Plug-in\sdk\3.1.0\bin`

If you are using the standalone version of flex Builder, your default path is:
`C:\Program Files\Adobe\Flex Builder 3\sdk\3.0.0\bin`

The directory 3.1.0 or 3.0.0 refers to the version of flex builder installed on your machine. Please look at your path within Windows Explorer to make sure you have the correct directory name.

9. Type the following command on the command prompt window
copylocale en_US es_ES.

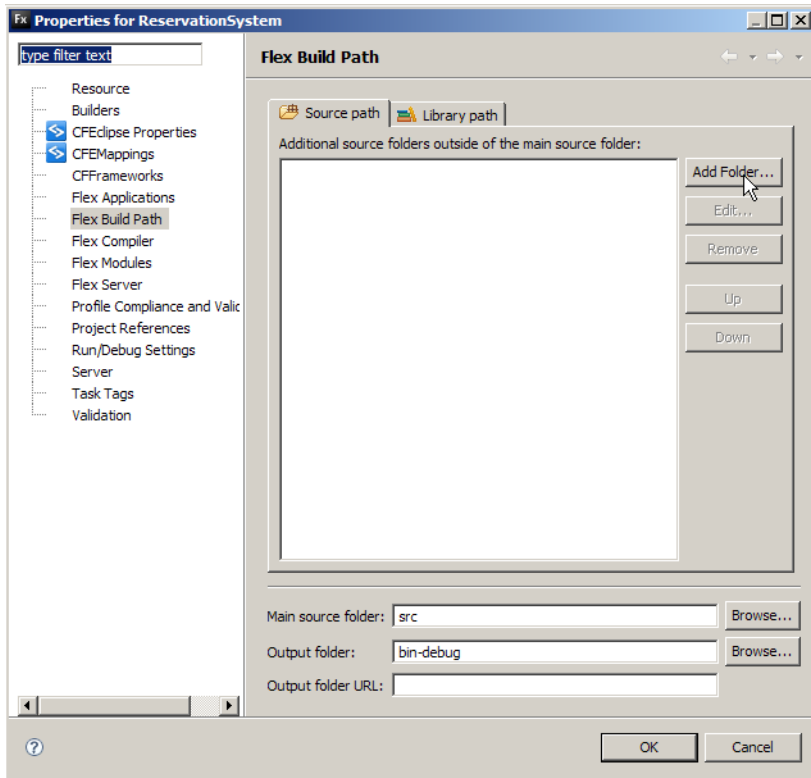


```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\Adobe\Flex Builder 3\sdk\3.0.0\bin>copylocale en_US es_ES
In Flex SDK at C:\Program Files\Adobe\Flex Builder 3\sdk\3.0.0 ...
Copying files from frameworks\projects\airframework\bundles\en_US\src
to frameworks\projects\airframework\bundles\es_ES\src:
```

10. Running this command will create the resources, and you will see a new directory es_ES under the path C:\Program Files\Adobe\Flex Builder 3 Plug-in\sdk\3.1.0\frameworks\locale.

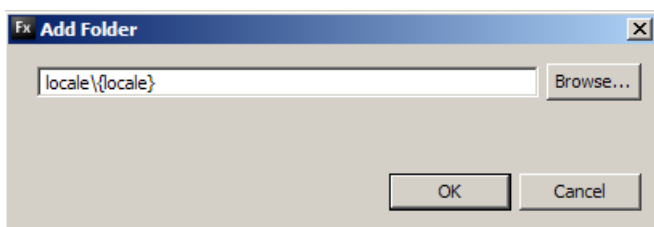


11. Add resources directory to the project's source path. In Flex Builder, select **Project > Properties.**
12. Select the **Flex Build Path** from the left hand menu.
13. In the dialog box, select the **Source path** tab.
14. Click the **Add Folder** button.



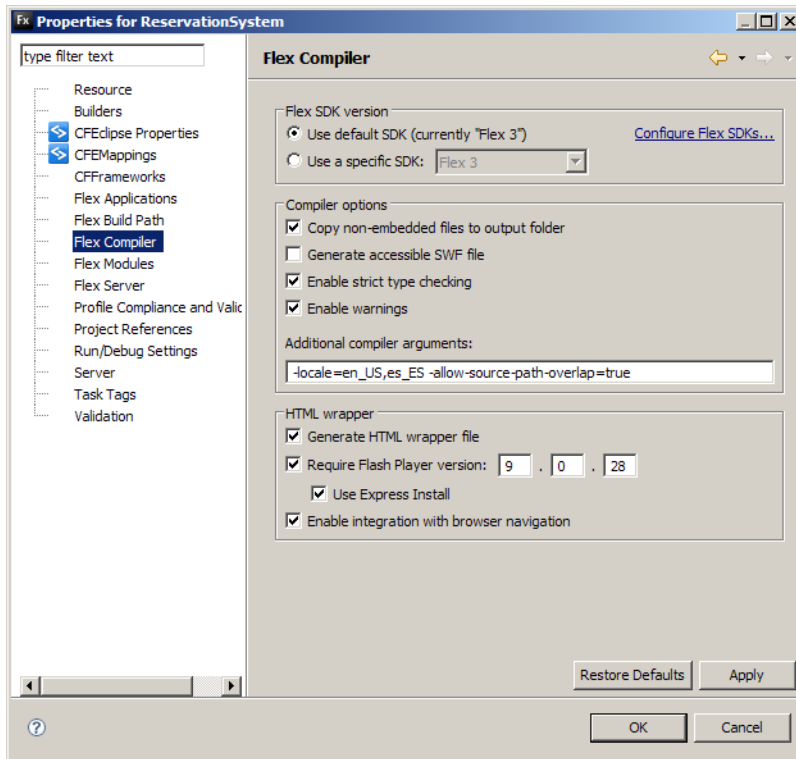
15. Navigate to the resource properties files parent directory. For this exercise, it should be in C:\Program Files\Adobe\Flex Builder 3 Plugin\sdk\3.1.0\frameworks\locale.
16. Append {locale} to the end of the directory path. This will create a generalized source path entry so that all locales are included.

Note: Look at the Source Path created after step 13. If it does not resolve to locale/{locale}, edit the path and enter only the relative path locale/{locale} in place of the complete path example: C:\Program Files\Adobe\Flex Builder 3 Plugin\sdk\3.1.0\frameworks\locale\{locale}.



17. Click **OK** to add this directory to your project's source path.
18. In the left hand menu of the dialog window, select **Flex Compiler**.

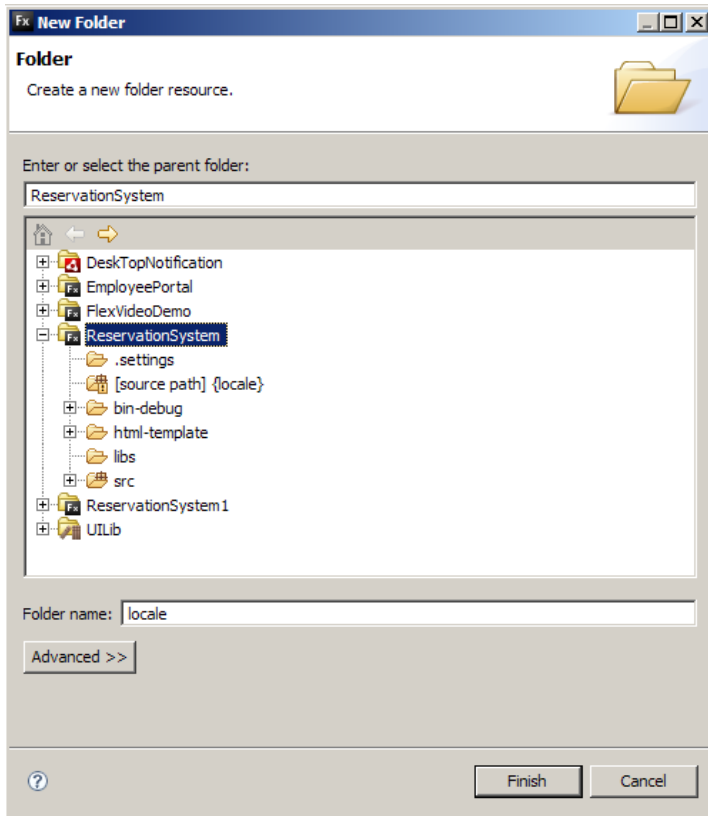
19. In the **Additional compiler arguments**, add the following statement: -
locale=en_US,es_ES -allow-source-path-overlap=true.



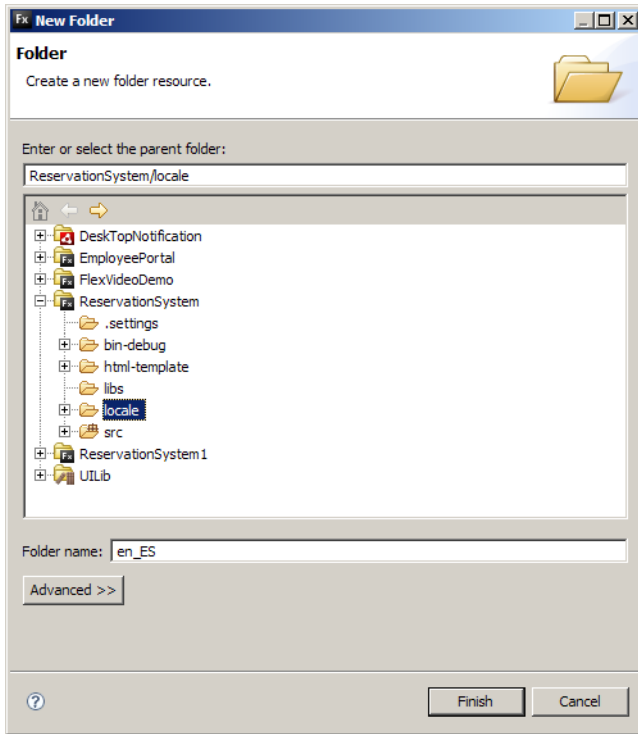
20. Click **OK** to save the settings.

Create the properties files

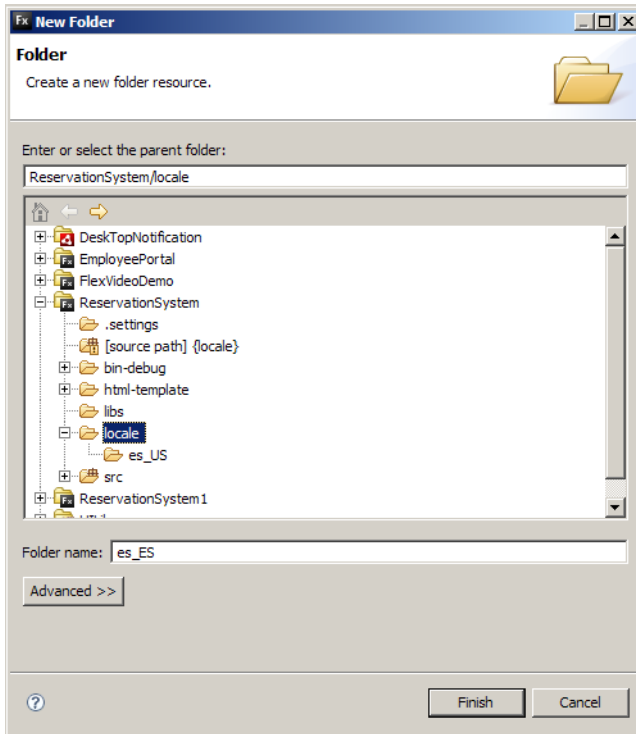
21. Select **File > New > Folder**.
22. In the **New Folder** dialog, ensure the **ReservationSystem** project is selected and enter `locale` for the **Folder name**.



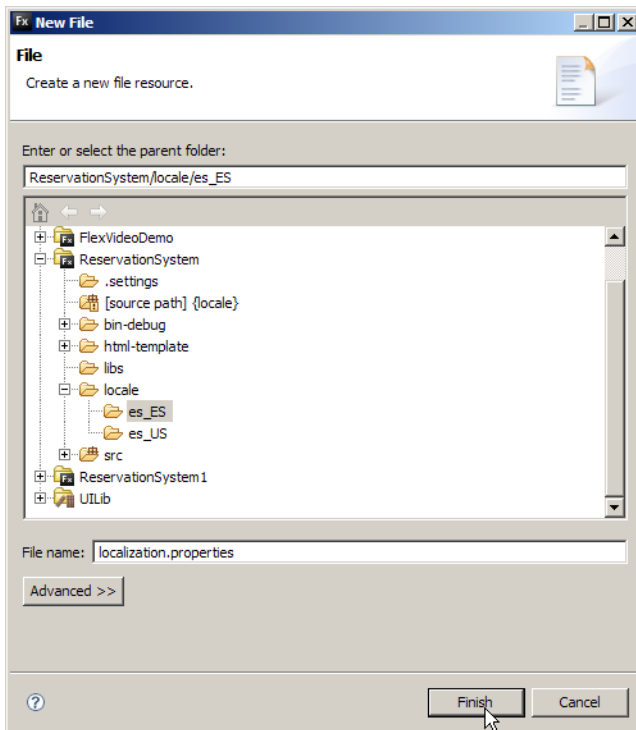
23. Click **Finish**.
24. Select **File > New > Folder**.
25. In the **New Folder** dialog, drill down into the **ReservationSystem** project and select the `locale` folder.
26. Enter `en_US` for the **Folder name**.



27. Click **Finish**.
28. Select **File > New > Folder**.
29. In the **New Folder** dialog, drill down into the **ReservationSystem** project and select the `locale` folder.
30. Enter `es_ES` for the **Folder name**.



31. In the **Flex Navigator** view, open the **ReservationSystem > locale** folder.
32. Right-click on the **en_US** folder and select **New > File**.
33. Enter `localization.properties` for the **File name**.
34. Click **Finish**.



35. Copy the following properties into the file.

```
# locale/en_US/localization.properties
# Localized Strings for RequestTicket.mxml
submit_button=Submit
personname=Full Name
address=Address
city=City
state=State
zip=Postal Code
room= Room
room_request= Room Request
date_needed= Date Needed
amenities=Amenities
phone=DayTime Phone
About You=About You
```

36. Save the file.

37. In the **Flex Navigator** view, right-click on the **local > es_ES** folder.

38. Select **New > File**.

39. Enter `localization.properties` for the **File name**.

40. Click **Finish**.

41. Copy the following properties into the file.

```
# locale/es_ES/localization.properties
# Localized Strings for RequestTicket.mxml
personname=Nombre
submit_button= Envio
address=Dirección
city=Ciudad
state=Estado-Provincia-Zona
zip=Código postal
room=Habitación
room_request= Peticion a la Habitacion
date_needed=Datos requeridos
amenities=Facilidades
phone=Numero de Telefono
About You=Sobre Ud
```

42. Save the file.

43. Open the `RequestTicket.mxml` file located in the **ReservationSystem > src > components** folder.

44. Below the ending `</mx:Script>` tag, create `<mx:Metadata>` tag set.
45. Define the `ResourceBundle` as localization. The localization `ResourceBundle` within the `<mx:Metadata>` tags refers to a properties file in your compiler's source path that stores the localized data.

Your code should appear as follows.

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

46. Within the `<mx:Script>` tags create a `Bindable` private variable named `locales` with a datatype set to the `Array` class.
47. Assign the values `["en_US", "es_ES"]` to the `locales` variable.

Your code should appear as follows:

```
[Bindable]
private var locales:Array = ["en_US", "es_ES"];
```

48. Create a private function named `comboBoxChangeHandler` that takes no parameters and returns `void`.
49. Within the `comboBoxChangeHandler` function, the locale selected in the combo box is assigned to the current locale at runtime. The `locales` array stores all the locales available to the user.

Your code should appear as follows:

```
private function comboBoxChangeHandler():void
{
    this.resourceManager.localeChain =
[localeComboBox.selectedItem];
}
```

50. After the beginning `<mx:Form>` tag, create a new `<mx:ComboBox>` tag set with the following properties:
 - `id = localeComboBox`
 - `dataProvider = {locales}`
 - `change = comboBoxChangeHandler()`

Your code should appear as follows:

```
<mx:ComboBox id="localeComboBox"
  dataProvider="{locales}"
  change="comboChangeHandler()"/>
```

51. Using the **Outline** view locate the FormItem labeled Full Name:.
52. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization','personname')}.

Your code should appear as follows.

```
<mx:FormItem
label="{resourceManager.getString('localization','personname')}">
  <mx:TextInput id="fullname"/>
</mx:FormItem>
```

53. Using the **Outline** view locate the FormItem labeled Address:.
54. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization','address')}.

Your code should appear as follows:

```
<mx:FormItem
label="{resourceManager.getString('localization','address')}">
  <mx:TextInput id="address"/>
</mx:FormItem>
```

55. Using the **Outline** view locate the FormItem labeled City:.
56. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization','city')}.

Your code should appear as follows:

```
<mx:FormItem
label="{resourceManager.getString('localization','city')}">
  <mx:TextInput id="city"/>
</mx:FormItem>
```

57. Using the **Outline** view locate the FormItem labeled State:.
58. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization','state')}.

Your code should appear as follows:

```
<mx:FormItem  
label="{resourceManager.getString('localization','state')}}">  
    <mx:TextInput id="state"/>  
</mx:FormItem>
```

59. Using the **Outline** view locate the FormItem labeled Postal Code:.
60. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization','zip')}.

Your code should appear as follows:

```
<mx:FormItem  
label="{resourceManager.getString('localization','zip')}}">  
  
    <mx:TextInput id="postalcode"/>  
  
</mx:FormItem>
```

61. Using the **Outline** view locate the FormItem labeled Daytime Phone:.
62. Change the label property for the Phone to have the following string:
{resourceManager.getString('localization','phone')}.

Your code should appear as follows:

```
<mx:FormItem  
label="{resourceManager.getString('localization','phone')}}">  
  
    <mx:TextInput id="phone"/>  
  
</mx:FormItem>
```

63. Using the **Outline** view locate the Submit button nested inside the last FormItem.
64. Change the label property for the Submit button to have the following string:
{resourceManager.getString('localization','submit_button')}

Your code should appear as follows:

```

<mx:FormItem>

    <mx:Button id="button"
    label="{resourceManager.getString('localization','submit_but
ton')}" />

</mx:FormItem>

```

65. Using the **Outline** view locate the <mx:Label> with a text property of About You.
66. Change the text property to have the following string:
{resourceManager.getString('localization' , 'about_you')}.

Your code should appear as follows:

```

<mx:Label
text="{resourceManager.getString('localization','about_you')}"
fontWeight="bold"
x="20" y="20"/>

```

67. Using the **Outline** view locate the <mx:Label> with a text property of Room Request.
68. Change the text property for Room Request to have the following string:
{resourceManager.getString('localization' , 'room_request')}.

Your code should appear as follows:

```

<mx:Label
text="{resourceManager.getString('localization','room_request'
)}"
fontWeight="bold"
x="20"/>

```

69. Using the **Outline** view locate the FormItem labeled Room:.
70. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization' , 'room')}.

Your code should appear as follows:

```

<mx:FormItem
label="{resourceManager.getString('localization','room')}">
    <mx:DateField id="dateNeeded"

```

```
selectableRange="{{rangeStart : new Date()}}" />

</mx:FormItem>
```

71. Using the **Outline** view locate the FormItem labeled Date Needed:.
72. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization' , 'date_needed')}.

Your code should appear as follows:

```
<mx:FormItem
label="{resourceManager.getString('localization','date_needed')}}"
>
    <mx:DateField id="dateNeeded"
        selectableRange="{{rangeStart : new Date()}}" />

</mx:FormItem>
```

73. Using the **Outline** view locate the FormItem labeled Amenities:.
74. Change the label property for the FormItem to have the following string:
{resourceManager.getString('localization' , 'amenities')}

Your code should appear as follows:

```
<mx:FormItem
label="{resourceManager.getString('localization','amenities')}">

    <mx>List id="thelist"
        width="100%" height="122"
        fontSize="9"
        borderStyle="none"
        backgroundAlpha="0"
        leading="0" />

</mx:FormItem>
```

75. Save the file and run the application. On the Request Ticket form after login, you should see a combo box. Clicking on the combo box and selecting es_ES would result in the form being localized into Spanish.