

FLEX - DROPDOWNLIST CONTROL

http://www.tutorialspoint.com/flex/flex_dropdownlist_control.htm

Copyright © tutorialspoint.com

Introduction

The DropDownList control contains a drop-down list from which the user can select a single value similar to that of the SELECT form element in HTML.

Class declaration

Following is the declaration for **spark.components.DropDownList** class:

```
public class DropDownList
    extends DropDownListBase
```

Public properties

S.N.	Property & Description
1	prompt : String The prompt for the DropDownList control.
2	typicalItem : Object [override] Layouts use the preferred size of the typicalItem when fixed row or column sizes are required, but a specific rowHeight or columnWidth value is not set.

Public methods

S.N.	Method & Description
1	DropDownList Constructor.

Methods inherited

This class inherits methods from the following classes:

- spark.components.supportClasses.DropDownListBase
- spark.components.List
- spark.components.supportClasses.ListBase
- spark.components.SkinnableDataContainer
- spark.components.supportClasses.SkinnableContainerBase
- spark.components.supportClasses.SkinnableComponent
- mx.core.UIComponent

- mx.core.FlexSprite
- flash.display.Sprite
- flash.display.DisplayObjectContainer
- flash.display.InteractiveObject
- flash.display.DisplayObject
- flash.events.EventDispatcher
- Object

Flex DropDownList Control Example

Let us follow the following steps to check usage of DropDownList control in a Flex application by creating a test application:

Step	Description
1	Create a project with a name <i>HelloWorld</i> under a package <i>com.tutorialspoint.client</i> as explained in the <i>Flex - Create Application</i> chapter.
2	Modify <i>HelloWorld.mxml</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to make sure business logic is working as per the requirements.

Following is the content of the modified mxml file **src/com.tutorialspoint/HelloWorld.mxml**.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx"
  width="100%" height="100%" minWidth="500" minHeight="500"
  >

  <fx:Style source="/com/tutorialspoint/client/Style.css"/>

  <fx:Script>
    <![CDATA[
      import mx.collections.ArrayCollection;
      import spark.events.IndexChangeEvent;

      [Bindable]
      public var data:ArrayCollection = new ArrayCollection(
      [
        {value:"France", code:"FR"},
        {value:"Japan", code:"JP"},
        {value:"India", code:"IN"},
        {value:"Russia", code:"RS"},
        {value:"United States", code:"US"}
      ]
      );

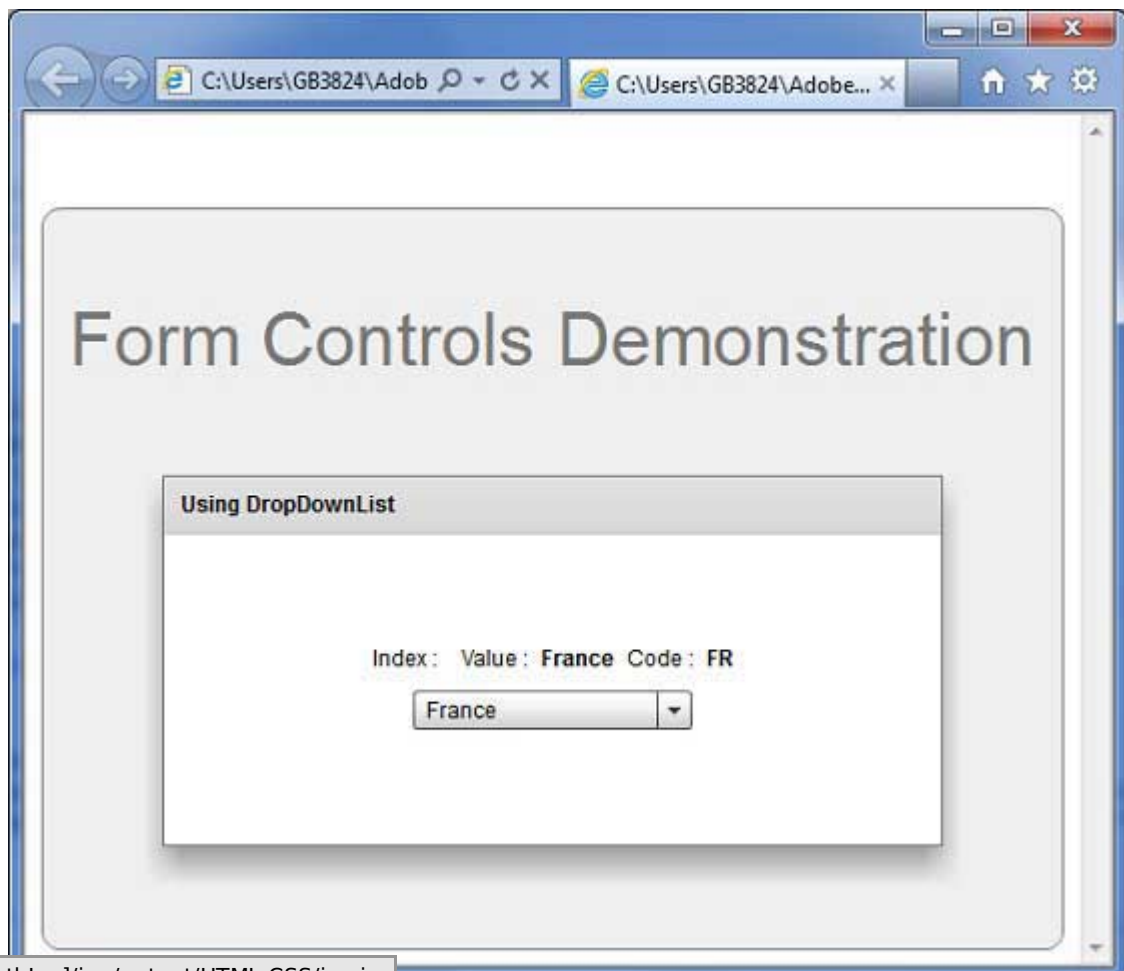
      protected function dropDownList_changeHandler
      (event:IndexChangeEvent):void
      {
        ddlIndex.text=dropDownList.selectedIndex.toString();
        ddlSelectedItem.text=dropDownList.selectedItem.value;
        ddlCode.text=dropDownList.selectedItem.code;
      }
    ]]>
```

```

</fx:Script>
<s:BorderContainer width="550" height="400"
styleName="container">
  <s:VGroup width="100%" height="100%" gap="50" horizontalAlign="center"
verticalAlign="middle">
    <s:Label
fontSize="40" color="0x777777" styleName="heading"/>
    <s:Panel
width="420" height="200">
      <s:layout>
        <s:VerticalLayout gap="10" verticalAlign="middle"
horizontalAlign="center"/>
      </s:layout>
      <s:HGroup>
        <s:Label text="Index :"/>
        <s:Label />
        <s:Label text="Value :"/>
        <s:Label
text="{dropDownList.selectedItem.value}" fontWeight="bold"/>
        <s:Label text="Code :"/>
        <s:Label
text="{dropDownList.selectedItem.code}" fontWeight="bold"/>
      </s:HGroup>
      <s:DropDownList
width="150" change="dropDownList_changeHandler(event)"
selectedIndex="0" labelField="value"/>
    </s:Panel>
  </s:VGroup>
</s:BorderContainer>
</s:Application>

```

Once you are ready with all the changes done, let us compile and run the application in normal mode as we did in [Flex - Create Application](#) chapter. If everything is fine with your application, this will produce following result: [[Try it online](#)]



Loading [MathJax]/jax/output/HTML-CSS/jax.js