FLEX - VIDEOPLAYER CONTROL

Introduction

The Video control supports playback of FLV and F4v files. This control contains a full-featured UI for controlling video playback.

Class declaration

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

```
public class VideoPlayer
   extends SkinnableComponent
```

Public properties

S.N.	Property & Description
1	**autoDisplayFirstFrame : Boolean** If autoPlay = false, then autoDisplayFirstFrame controls whether the video is loaded when the source is set.
2	**autoPlay : Boolean** Specifies whether the video starts playing immediately when the source property is set.
3	**autoRewind : Boolean** Specifies whether the FLV file should rewind to the first frame when play stops, either by calling the stop method or by reaching the end of the stream.
4	**bytesLoaded : Number** [read-only] The number of bytes of data that have been downloaded into the application.
5	**bytesTotal : Number** [read-only] The total size in bytes of the data being downloaded into the application.
6	**currentTime : Number** [read-only] Current time of the playhead, measured in seconds, since the video starting playing.
7	**duration : Number** [read-only] Duration of the video's playback, in seconds.

8	**loop : Boolean** Indicates whether the media should play again after playback has completed.
8	**mediaPlayerState : String** [read-only] The current state of the video.
10	**muted : Boolean** Set to true to mute the video, false to unmute the video.
11	**pauseWhenHidden : Boolean** Controls whether the video continues to play when it is "hidden".
12	**playing : Boolean** [read-only] Contains true if the video is playing or is attempting to play.
13	**scaleMode : String** The scaleMode property describes different ways of sizing the video content.
14	**source : Object** The video source.
15	**videoObject : Video** [read-only] The underlying flash player flash.media.Video object.
16	**volume : Number** The volume level, specified as a value between 0 and 1.

Public methods

S.N.	Method & Description
1	**VideoPlayer** Constructor.
2	**pause:void** Pauses playback without moving the playhead.

3	**play:void** Causes the video to play.
4	**seek*time*:*Number*:void** Seeks to given time in the video.
5	**stop:void** Stops video playback.

Protected methods

S.N.	Method & Description
1	**formatTimeValue*value*:*Number*:String** Formats a time value, specified in seconds, into a String that gets used for currentTime and the duration.

Events

S.N.	Method & Description
1	**bytesLoadedChange** Dispatched when the data is received as a download operation progresses.
2	**complete** Dispatched when the playhead reaches the duration for playable media.
3	**currentTimeChange** Dispatched when the currentTime property of the MediaPlayer has changed.
4	**durationChange** Dispatched when the duration property of the media has changed.
5	**mediaPlayerStateChange** Dispatched when the MediaPlayer's state has changed.

Methods inherited

This class inherits methods from the following classes:

- 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 VideoPlayer Control Example

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

Step	Description
1	Create a project with a name *HelloWorld* under a package *com.tutorialspoint.client* as explained in the *Flex - Create Application* chapter.
2	Add a folder *video* to *src* folder. And add sample video to it.
3	Modify *HelloWorld.mxml* as explained below. Keep rest of the files unchanged.
4	Compile and run the application to make sure business logic is working as per the requirements.

Following is the way to embed an video in flex application.

```
<s:VideoPlayer source="video/just for laugh magic trick.flv"
   width="350" height="250" loop="true"/>
```

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"/>
   <s:BorderContainer width="630" height="480"
      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="500" height="300" >
               <s:layout>
                  <s:HorizontalLayout  gap="10" verticalAlign="middle"
```

```
                    horizontalAlign="center"/>
                </s:layout>
                <s:VideoPlayer
                    source="video/just for laugh magic trick.flv"
                    width="350" height="250"
                    loop="true"/>
            </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]