

Introduction

The TitleWindow class extends Panel class to include a close button and move area.

Class declaration

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

```
public class TitleWindow  
    extends Panel
```

Public methods

S.N.	Method & Description
------	----------------------

- | | |
|---|------------------------------------|
| 1 | TitleWindow
Constructor. |
|---|------------------------------------|

Events

S.N.	Event & Description
------	---------------------

- | | |
|---|--|
| 1 | close
Dispatched when the user selects the close button. |
| 2 | windowMove
Dispatched after the user dragged the window successfully. |
| 3 | windowMoveEnd
Dispatched when the user releases the mouse button after dragging. |
| 4 | windowMoveStart
Dispatched when the user presses and hold the mouse button in the move area and begins to drag the window. |
| 5 | windowMoving
Dispatched when the user drags the window. |

Methods inherited

This class inherits methods from the following classes:

- spark.components.Panel
- spark.components.SkinnableContainer
- 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 TitleWindow Example

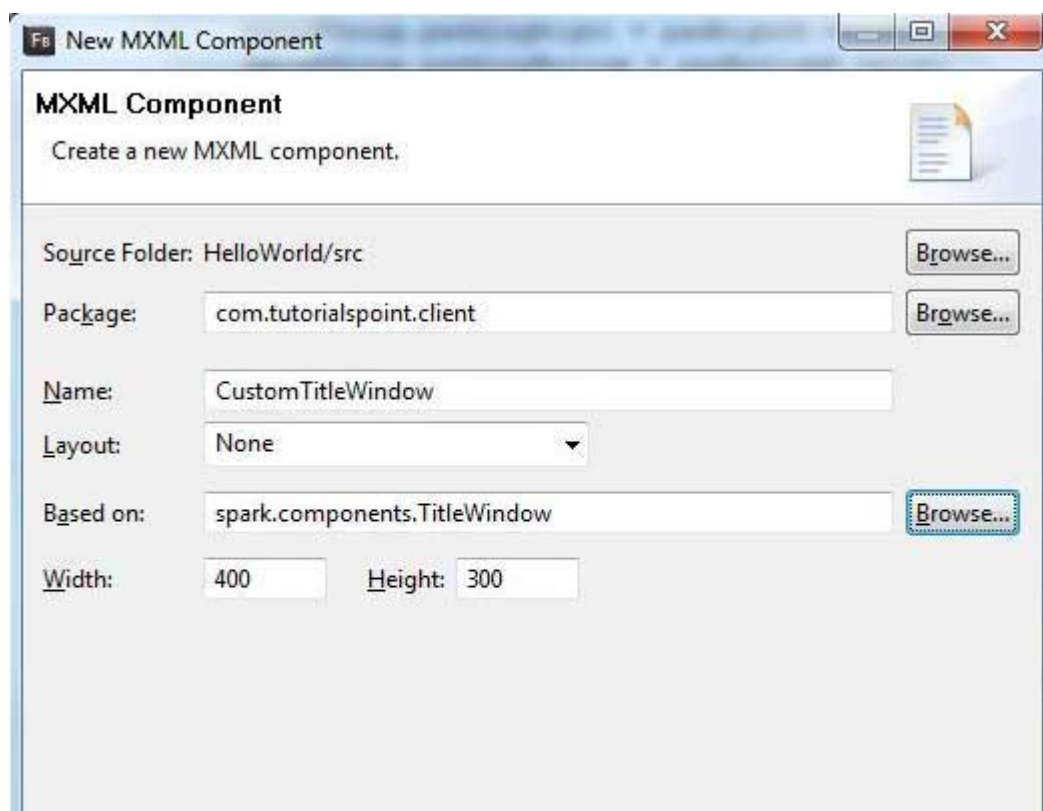
Let us follow the following steps to check usage of TitleWindow in a Flex application by creating a test application:

Step 1: Create a Project

Create a project with a name *HelloWorld* under a package *com.tutorialspoint.client* as explained in the *Flex - Create Application* chapter.

Step 2: Create a custom Title Window component

Launch Create MXML Component wizard using the option **File > New > MXML Component**





Enter Package as **com.tutorialspoint.client**, name as **CustomTitleWindow** and choose based on as existing flex TitleWindow control **spark.component.TitleWindow**. Modified it as mentioned below.

Following is the content of the modified mxml file

src/com.tutorialspoint.client/CustomTitleWindow.mxml.

```
<?xml version="1.0" encoding="utf-8"?>
<s:TitleWindow 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="400" height="300"
  title="Title Window Container"
  close="PopUpManager.removePopUp(this);">
  <s:layout>
    <s:VerticalLayout verticalAlign="middle" horizontalAlign="center" />
  </s:layout>
  <fx:Script>
    <![CDATA[
      import mx.managers.PopUpManager;
    ]]>
  </fx:Script>
  <s:Label text="Content area of title window"/>
</s:TitleWindow>
```

Step 3: Modify HelloWorld.mxml

Modify *HelloWorld.mxml* as explained below. Keep rest of the files unchanged.

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.core.IFlexDisplayObject;
      import mx.managers.PopUpManager;
      private function showWindow():void{
        var popUp:IFlexDisplayObject =
          PopUpManager.createPopUp(this, CustomTitleWindow, true);
        PopUpManager.centerPopUp(popUp);
      }
    ]]>
  </fx:Script>
  <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:VerticalLayout gap="10" verticalAlign="middle"
            horizontalAlign="center"/>
        </s:layout>
      </s:VGroup>
    </s:BorderContainer>
```

```
<s:Button
    label="Click to show the TitleWindow container"
    click="showWindow();" />
</s:Panel>
</s:VGroup>
</s:BorderContainer>
</s:Application>
```

Step 4: Compile and Run application

Compile and run the application to make sure business logic is working as per the requirements.

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](#)]

