# FLEX - SKINNABLECONTAINER

 

## Introduction

The SkinnableContainer class is the base class for all the skinnable containers that have visual content.

## Class declaration

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

```
public class SkinnableContainer
   extends SkinnableContainerBase
      implements IDeferredContentOwner, IVisualElementContainer
```

## Public properties

| S.N. | Property & Description |
|---|---|
| 1 | **autoLayout : Boolean**<br>If true, measurement and layout are done when the position or size of a child is changed. |
| 2 | **creationPolicy : String**<br>Content creation policy for this component. |
| 3 | **deferredContentCreated : Boolean**<br>[read-only] Contains true if deferred content has been created. |
| 4 | **layout : LayoutBase**<br>The layout object for this container. |
| 5 | **mxmlContent : Array**<br>[write-only] The visual content children for this Group. |
| 6 | **mxmlContentFactory : IDeferredInstance**<br>[write-only] A factory object that creates the initial value for the content property. |
| 7 | **numElements : int**<br>[read-only] The number of visual elements in this container. |

## Public methods

| S.N. | Method & Description |
| --- | --- |
| 1 | **SkinnableContainer**<br>Constructor. |
| 2 | **addElement***element*:*IVisualElement*:**IVisualElement**<br>Adds a visual element to this container. |
| 3 | **addElementAt***element*:*IVisualElement, index*:*int*:**IVisualElement**<br>Adds a visual element to this container. |
| 4 | **createDeferredContent:void**<br>Create the content for this component. |
| 5 | **getElementAt***index*:*int*:**IVisualElement**<br>Returns the visual element at the specified index. |
| 6 | **getElementIndex***element*:*IVisualElement*:**int**<br>Returns the index position of a visual element. |
| 7 | **removeAllElements:void**<br>Removes all visual elements from the container. |
| 8 | **removeElement***element*:*IVisualElement*:**IVisualElement**<br>Removes the specified visual element from the child list of this container. |
| 9 | **removeElementAt***index*:*int*:**IVisualElement**<br>Removes a visual element from the specified index position in the container. |
| 10 | **setElementIndex***element*:*IVisualElement, index*:*int*:**void**<br>Changes the position of an existing visual element in the visual container. |
| 11 | **swapElements***element*1:*IVisualElement, element*2:*IVisualElement*:**void**<br>Swaps the index of the two specified visual elements. |
| 12 | |

**swapElementsAt***index1*:*int, index2*:*int*:**void**

Swaps the visual elements at the two specified index positions in the container.

## Protected methods

| S.N. | Method & Description |
|---|---|
| 1 | **createChildren:void**<br>[override] Create content children, if the creationPolicy property is not equal to none. |
| 2 | **partAdded***partName*:*String, instance*:*Object*:**void**<br>[override] Called when a skin part is added. |
| 3 | **partRemoved***partName*:*String, instance*:*Object*:**void**<br>[override] Called when an instance of a skin part is being removed. |

## Events

| S.N. | Event & Description |
|---|---|
| 1 | **contentCreationComplete**<br>Dispatched after the content for this component has been created. |
| 2 | **elementAdd**<br>Dispatched when a visual element is added to the content holder. |
| 3 | **elementRemove**<br>Dispatched when a visual element is removed from the content holder. |

## Methods inherited

This class inherits methods from the following classes:

- spark.components.supportClasses.SkinnableContainerBase
- mx.core.UIComponent
- mx.core.FlexSprite
- flash.display.Sprite
- flash.display.DisplayObjectContainer

- flash.display.InteractiveObject
- flash.display.DisplayObject
- flash.events.EventDispatcher
- Object

## Flex SkinnableContainer Example

Let us follow the following steps to check usage of SkinnableContainer 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 | Create a skin *SkinnableContainerSkin* for a host Component *SkinnableContainer* under a package *com.tutorialspoint.skin* as explained in *Flex - Style with skin* chapter. Keep rest of the files unchanged. |
| 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 content of the modified mxml file **src/com.tutorialspoint/skin/SkinnableContainerSkin.mxml**.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Skin xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:fb="http://ns.adobe.com/flashbuilder/2009" alpha.disabled="0.5">
   <fx:Metadata>
      <![CDATA[
         /**
         * @copy spark.skins.spark.ApplicationSkin#hostComponent
         */
         [HostComponent("spark.components.SkinnableContainer")]
      ]]>
   </fx:Metadata>
   <s:states>
      <s:State name="normal" />
      <s:State name="disabled" />
   </s:states>
   <s:Rect left="0" right="0" top="0"
      bottom="0" radiusX="5" radiusY="5">
      <s:stroke>
         <s:LinearGradientStroke weight="2"/>
      </s:stroke>
      <s:fill>
         <s:LinearGradient rotation="90">
            <s:entries>
               <s:GradientEntry color="0xdddddd"/>
               <s:GradientEntry color="0x020202" alpha=".5" />
            </s:entries>
         </s:LinearGradient>
      </s:fill>
   </s:Rect>
   <s:Group
      top="0" bottom="0" minWidth="0" minHeight="0">
      <s:layout>
         <s:BasicLayout/>
      </s:layout>
```

```
   </s:Group>
</s:Skin>
```

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:VerticalLayout  gap="10" verticalAlign="middle"
               horizontalAlign="center"/>
            </s:layout>
            <s:SkinnableContainer
               skinClass="com.tutorialspoint.skin.SkinnableContainerSkin"
               width="50%" height="50%" horizontalCenter="0"
               verticalCenter="0">
               <s:HGroup horizontalCenter="0" verticalCenter="0">
                  <s:BorderContainer width="50" height="50"
                     borderWeight="2" color="0x323232" />
                  <s:BorderContainer width="50" height="50"
                     borderWeight="2" color="0x323232" />
                  <s:BorderContainer width="50" height="50"
                     borderWeight="2" color="0x323232" />
               </s:HGroup>
            </s:SkinnableContainer>
         </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 ]