

# Styling and Skinning Flex Applications

Andrew Trice

Cynergy Systems  
December 06, 2006



## Topics for the day...

- Understanding & Creating Custom Styles
- Understanding Scale 9
- Skinning With Images
- Understanding the Drawing API
- Basic Programmatic Skins
- Advanced Programmatic Skins



- Styling is customization of your application UI based on stylesheets.
- Skinning is customization of your application UI based on graphic or programmatic assets.
- Skinning relies on definitions within your stylesheet.
- Styling is extremely powerful by itself. Incorporation of graphic skinning with styling can dramatically alter the appearance of your application.



- The Flex Style Explorer is extremely helpful for working with Styles! Don't limit yourself to this; it does not cover everything.  
<http://examples.adobe.com/flex2/consulting/styleexplorer/Flex2StyleExplorer.html>
- Styles are defined on Flex framework components. You can customize these styles (and add your own).
- **Application { newStyle: 123 }** will define the style property “newStyle” within the Application stylesheet. You can access this at runtime using the method **getStyle**.
- You can take advantage of this to have your application or components respond to styles at runtime.



- On the initialize event of a component, you can use the stylesheet to populate values in your application.

```
public function onInitialize() : void
{
    //this sets the styles of child components of my component based on
    //a single style that is defined to the control
    panel.styleName = getStyle("panelStyleName");
    dragBar.styleName = getStyle("dragBarStyleName");
    button1.styleName = getStyle("button1StyleName");
    button2.styleName = getStyle("button2StyleName");
    headerTitle.styleName = getStyle("titleStyleName");
    dateChooser.styleName = getStyle("dateChooserStyleName");

    //set properties based on css styles
    dragBar.height = dragBar.getStyle("height");
    button1.label = button1.getStyle("label");
    button2.label = button2.getStyle("label");
}
```



- Skins are graphics displayed by components.
- Skins are defined either inline in your code using style properties or by using CSS stylesheets.
  - Examples:
    - upSkin, overSkin, downSkin on a mx:Button component
    - borderSkin on a mx:Canvas
- Skins can be created using images/graphics (JPG, GIF, PNG, SWF) or can be created programmatically using the drawing API.

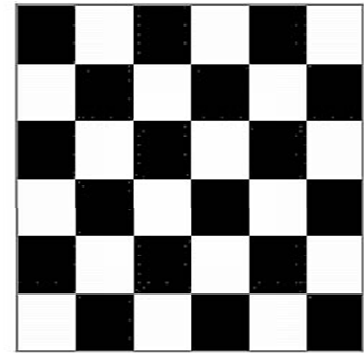
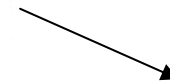


- Graphic skins can be created using static images (JPG, GIF, PNG) or Flash-based vector graphics.
- Graphic skins are easier to implement (without any graphics coding) and are created using traditional graphics tools... You won't have to give up Fireworks, Flash, Photoshop, or whichever tool you are currently used to using.
- Graphic skins generally have a larger file size than programmatic skins.
- Create a skin by embedding within style definition
  - `upSkin: Embed(source="assets/demo.png");`



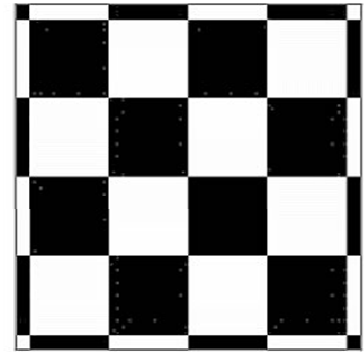
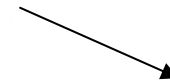
- Graphic skins scale according to the defined width and height of the component. They can scale in 2 ways...

- Uniform scaling: The entire image is scaled horizontally/vertically in proportion with the content of the original image.



- Scale9 Grid

- Content in the center region is scaled normally.
- Content in the corners is not scaled.
- Content in the top and bottom regions is scaled only horizontally. Content in the left and right regions is scaled only vertically.
- All fills (including bitmaps, video, and gradients) are stretched to fit their shapes.
- [http://livedocs.macromedia.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=Liv eDocs\\_Parts&file=00000976.html#186094](http://livedocs.macromedia.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=Liv eDocs_Parts&file=00000976.html#186094)



- Programmatic skins are created using ActionScript and the drawing API. (The drawing API is one of the most powerful features of the Flash/Flex platform, which sets it far ahead of traditional HTML-based content.)
- Programmatic skins are more difficult to implement and require coding, but are very flexible and can be changed at runtime.
- Create a skin by creating a class that subclasses either `mx.skins.ProgrammaticSkin` or `mx.skins.RectangularBorder`
- Apply a skin within css stylesheet for your component:
  - `upSkin: ClassReference( "ButtonSkin" );`
- Look at the existing Halo skins as a reference! They are a great resource.
  - `C:\Program Files\Adobe\Flex Builder 2\Flex SDK 2\frameworks\source\mx\skins\halo`



- Now, lets get to some code examples...

