Flex in a Week, Flex 4
Video 1.09: Introducing styling and skinning

Implement the same as before except now use new Company Vehicle Request form

In this video, you will have a quick introduction to various techniques for changing the look-and-feel of your application.

First you will learn how to change the application appearance using Cascading Style Sheets, or CSS.

You will also watch me apply a Spark skin to the application and use the Theme Browser to change the component theme.

Convert_to_CSS.camrec

This is the main application file that I modified in the last video.

The first task I will show you in this video is how to create and use a CSS style in Flash Builder's Design mode.

I am switching to Design mode and then selecting the blue Label control that reads Company Vehicle Request Form.

In the Properties view you can see all of the Text properties that I assigned earlier.

Now I am clicking the Convert to CSS button to have Flash Builder turn all of the properties currently assigned to this text field into a CSS style.

New_Style_Rule.camrec

In the New Style Rule dialog, I am clicking the New button next to the Define style in drop-down list.

I am making sure that the src folder is selected, and then typing Styles for the CSS filename and pressing the Finish button.

Now I am selecting All components with style name for the Selector type and then typing addHeader for the Name field, which will be the name of the CSS selector.

addHeader_applied.camrec

```
<s:Label x="80" y="34"
    text="Add new employee"
    styleName="addHeader"/>
```

When I click OK you can see that the Styles.css file opens up and that it has the addHeader custom selector created with all of the properties defined for the text field.

Back in the main application file, in Design mode, you can see that the Label control now has as Style property set to addHeader.

When I switch over to Source mode, you can see that the control now has the styleName property set to addHeader.

There is also a Style tag that has been added to my MXML application file which points to the Styles.css file which is now in the source default package.

SparkSkin.camrec

If you want to change the look and feel of a component, you can either style it, like you have done to the Label control, or you can skin it.

In this video I will not go into depth about skinning; this topic will be discussed in great detail in Day 5 of this training.

However, I will discuss a few key elements necessary for you to implement a skin.

When you create a skin, you base it on a Flex framework class named SparkSkin.

Next you define a HostComponent that states what you are skinning.

In this case, I am creating a skin for the Application container.

Most components have at least two states: a normal and a disabled state.

Other components, for instance Button controls, have additional states like up, down and over.

This Rect tag represents the background for the Application tag and defines a rectangle shape.

Note that the horizontalCenter property is a constraint that directs the Flash Player to render this background image in the center of the application; in other words, zero pixels from the horizontal center of the application.

The width property sets the width of the rectangle to 400 pixels, as you might expect.

The height property ensures that the background is always as tall as the application window.

The fill and stroke properties set the rectangle to a light-gray color with a darker gray border.

Lastly, the Group container declares that the content children of the application, represented by the skin part named contentGroup, will be displayed vertically starting 20 pixels from the top of the browser and consistently centered in the browser horizontally.

Skin_Class_Files.camrec

The Spark skin that I will apply to this application is in the skins directory of my project and is named AppSkin.mxml.

When I open it, you can see that it looks very similar to the skin file that I just discussed.

Back in the main application file, I am locating the opening Application tag and adding a skinClass property to it.

This is the property I will use to tie the skin to the application.

Note that Flash Builder lists the available skins, including the AppSkin custom skins that I created.

When I select the AppSkin option from the code assist dropdown, Flash Builder puts the full skin path, skins.AppSkin, into the skinClass property.

It also generates a Script tag block to import the skin into the application via ActionScript.

You will learn more about the Script block later in this video series, however, this code is not necessary for the skin to apply properly, so I'm going to delete it.

Delete_minWidth_and_minHeight.camrec

When I save the file and run the application, you can see the same application as before, but now with a gray background.

You can also see that the application content isn't displayed in the center of the browser.

This is because the application defines minWidth and a minHeight properties which are larger than the 800 x 600 pixel browser dimensions that I have defined while I am recording this presentation.

I am deleting the two properties then saving the file and running the application again.

The skin is now centered in the browser.

The form itself is a little off-center, but that is because I have defined the form display with absolute pixel positioning.

I am not going to worry about fixing that right now.

Yellow_Background.camrec

```
<s:Rect horizontalCenter="0"
    width="750" height="100%"
    radiusX="10">

<s:fill>
   <mx:SolidColor color="#FFFBCF" />
</s:fill>
```

Back in the AppSkin.mxml file, I am changing the fill color to a light yellow by typing #FFFBCF.

I am also adding a radiusX property with a value of 10 pixels to the Rect tag, to round the rectangle's corners.

When I save the file and run the application, you can see that the application background color is now yellow with rounded corners.

It still does have a gray border.

Theme_browser.camrec
The last feature of Design mode that I will show you in this video is the Theme Browser.

I am switching back to Design mode and then selecting the Appearance tab.

When I click on the Spark link, Flash Builder opens a dialog with many theme options.

I am expanding the last set of options and selecting Sage.

I am clicking OK and then running the application.

You can see that the application's controls are now styled using the Sage theme.

Next_step.camrec

Next step

Exercise: Creating a user interface

For your next step, work through the exercise titled "Creating a user interface".