Flex in a Week, Flex 4.5
Video 5.02: Utilizing the Text Layout Framework

In the last video, you learned about the five Spark text controls.

In this video, you will continue to learn about properties available in the controls.

You will also learn more about the Text Layout Framework, including the flow elements that can be used to format the text.

Here is the Employee Portal application.

In the last video, you learned how to add the Employee Portal text headers, implement some Spark primitive text controls and add a scroll bar to the Monthly Events panel display.

In this video, I will continue to improve the display for the Cafeteria Special and Monthly Events custom components.

I will format the calorieContent text using the Text Layout Editor and create vertical text.

I will also separate the content in the Monthly Events panel into headers and paragraphs.

The Flash Text Engine is a new set of classes in Flash Player 10 that supports low-level text functionality such as rendering individual lines of text.

The Text Layout Framework, or TLF, is a set of classes built on top of the Flash Text Engine.

These classes are provided by the Flex framework in a SWC file that is automatically referenced in your applications.

The TLF provides high-level text formatting functionality, including multiple lines and paragraphs as well as scrolling, selection and editing.

It also includes some HTML-like flow elements for formatting your text controls.

Here is a list of the TLF flow elements.

In this video, I will show you how to use the paragraph and span flow elements.

In the main application file, I am CTRL + clicking on the MonthlyEvents custom component to locate the RichEditableText control that displays the monthly events content in the scrollable text of the last panel container.

Between the RichEditableText tag and the content, I am adding an opening and closing paragraph flow tag.

To the opening paragraph tag, I am adding the fontWeight property and setting its value to bold.

Between the tags, I am typing the text COMPANY PICNIC.

Now I am surrounding each line of text with a paragraph tag set, making sure there is no space between the p tag and the first word of the text.

I'm copying the paragraph tag that contains the COMPANY PICNIC text and pasting it before the second paragraph.

I'm changing the text to FAMILY DAY AY THE BALL PARK.

I am saving the file and running the application.

You can see that the text is no longer editable. Notice that there is a lack of vertical spacing between the paragraph title text and the content text.

Back in the MonthlyEvents.mxml file, to the RichEditableText opening tag, I'm assigning the paragraphSpaceBefore property value to 10.

This property will add the defined amount of pixels – 10 – before the paragraph tag.

I'm saving the file and running the application again.

You can see the added spacing between the paragraph title and content.

Notice the padding around the scroll bar.

In the MonthlyEvents component file, I am copying the padding properties from the VerticalLayout class and pasting them into the opening RichEditableText control tag.

I am removing all of the padding properties, except paddingBottom, from the VerticalLayout tag.

When I save the file and run the application, you can see that the scroll bar is now flush with the right of the Monthly Events panel and the content still has nice padding around it.

From the main application file, I’m opening the Cafeteria.mxml component file by CTRL + clicking on it.

I am copying the text between the span tags in the RichText control.

In my browser, I am navigating to this URL (http://labs.adobe.com/technologies/textlayout/demos/).

I am pasting the text in the Text Layout Editor window.

After I select the text, I am using the Character menu to change the Font to Arial and setting the Type to Italic.

Next, I am clicking on this Markup button at the bottom.

Notice that the tags in the Markup window are very similar to flow tags you've already seen, but you will need to change them to use the Spark namespace instead of this flow namespace.

From the code preview window, I am selecting and copying all the content between the TextFlow tags and then returning to the Cafeteria.mxml file and deleting everything between the RichText tags.

Now I am pasting my new code.

When I save the file, I get an error because of the incorrect namespace.

I am selecting Edit > Find/Replace window to find all of the instances of flow, and I am replacing them with an s for the Spark namespace.

I hitting Replace All, and then clicking Close.

Notice that all of the instances of the flow namespace have been replaced with s.

I am saving the file and running the application.

You can see that the calorieContent text is has been formatted.

Lastly, I want to make this Calorie Information header appear as vertical text on the left.

In the Cafeteria component file, I am surrounding the Label control with the Calorie Information text and the RichText control with an HGroup container.

I'm highlighting them and using the tab to indent them.

To the HGroup container, I am assigning the width property value to 100%.

This will take up the entire available width of the panel area.

To the Label control, I am adding a rotation property with a value of 270 degrees.

The HGroup container will force this Label control to sit vertically to the left of the content in the RichText control.

When I save the file and run the application, you can see that the text is vertical but the information in the panel is not centered and some of it escapes the panel content area

This is because of the padding properties that were previously assigned to the VerticalLayout class within the component file.

I am changing the paddingLeft and paddingRight properties to have a value of 5 and assigning the horizontalAlign property with a value of center.

When I save the file and run the application again, you can see that all the elements are centered with in the panel.

For your next step, work through the exercise titled "Using text controls".