Flex in a Week, Flex 4.5
Video 5.01: Introducing the text controls

In this final Day of training, you will learn how to improve the look-and-feel of your applications using Spark text controls, styling and skinning.

In this video, you learn about the first topic: the Spark text controls.

You have worked with this Employee Portal application before to create states and transitions.

On this day of training, we strip this application down to focus on the visual display.

By the end of the day of training, you will have converted it to look like this, with vertical and formatted text, vertical Panel container headers that display different colors, and a Button control that animates.

Flex 4.5 contains both the new Spark text controls and the old MX text controls.

As with all other components that you have learned about in this series, use the Spark equivalents whenever possible.

This video will focus on the Spark components.

Starting with Flex 4, both Spark and MX components can take advantage of the Flash Text Engine and the Text Layout Framework, but the MX controls take advantage of them to a lesser degree.

You will learn more about the Text Layout Framework in the next video.

I have search the Help files for this article: About the Spark text controls in the Using Adobe Flex 4.5 content.

This chart lists the Spark text controls and gives you useful information about whether the control is editable and supports multiline text.

As you just saw in the Help files, there are 5 Spark text controls, split into two categories.

The input text controls are TextArea and TextInput.

The three text primitives are Label, RichText and RichEditableText.

As you might expect, the input text controls accept user input.

You have used the TextInput control in the exercises for this video series already: it allows users to enter and edit a single line, and a single style, of text.

The TextArea control, on the other hand, allows users to enter and edit multiple lines of text.

Additionally, this control supports vertical and horizontal scrollbars as well as other rich formatting options like multiple paragraphs, inline graphics, multiple columns, and more.

However, the TextArea control does not include any user interface controls for applying rich formatting.

All of the formatting must be changed programmatically.

The second category of Spark text controls is the text primitives.

These controls offer new text functionality available in Flash Player 10 and AIR 1.5.

The implementation is achieved using the new Flash Text Engine and the Text Layout Framework.

You will learn more about these elements in the next video.

The text primitives are typically used in component skins.

I am scrolling down the page in the Using Flex 4.5 documentation that I showed you earlier and clicking on the link to the Spark text control feature overview.

You can see that this table lists the features available in each of the text primitive controls.

You can see that the Label control, in the first column, is the lightest-weight control, with fewer features than the RichText or RichEditableText controls.

This table is extremely useful for you to quickly identify which control would best fit your needs.

As I just mentioned, the Label control is the lightest of the three text primitives.

A perfect use for it is as the label field in a Button control since this requires a limited amount of text.

There is an MX equivalent control with the same name, but it is not exactly the same.

The Spark Label control can display multiple lines of text, while the MX control can only display a single line of text.

Both can only use one format, or style, at a time, however.

The Spark Label control can also render bi-directional text and support rotation and alpha property settings, even if you are using device fonts, or non-embedded fonts.

This control is also completely non-interactive and doesn't support hyperlinks, scrolling, selection or editing.

Back in the Employee Portal application, I will show you how to implement and format some Label controls.

I am switching to Design mode.

The application dimensions look small in Design mode because of my limited presentation window and because the Application container does not define any width and height values, which means that it will resize to the available area.

I will expand the view in a moment, after I place the components.

From the Components view, I am dragging an HGroup container and placing it above the Employee of the Month Panel container.

In the Properties view, I am setting the X value to 24 and the Y value to 21 and removing the Width and Height properties.

In the Layout segment, I am setting the Gap property to a value of 15.

Next, from the Components view, I am dragging a Image control and placing it inside to the HGroup container.

In the Properties view, I am clicking on the folder icon next to the Source field and opening the logo.png file from the assets folder.

I am switching to Source mode and changing the Image control to a BitmapImage control.

I am using the BitmapImage control instead of the Image control because I am not going to skin the logo.

I'm switching back to Design mode and dragging a Label control inside the HGroup container to the right of the logo.

With the Label control selected, I am using the Properties view to assign the text property a value of Employee, a fontWeight value of bold and a fontSize value of 38.

I am dragging another Label control and dropping it next to the Employee Label control within the HGroup container.

With the new Label control selected, I am using the Properties view to change the text value to Portal and the size to 38.

You can see that the E in Employee is not aligned with the bottom of the logo.

The logo bottom is actually aligned with the bottom of the descenders on the p and y.

I’m selecting the Employee Label control and switching back to Source mode.

Using code hinting, I am adding the alignmentBaseline property value to ideographicBottom.

I am doing the same for the Portal Label control.

When I save the file and switch back to Design mode, you can see that the logo is now aligned with the bottom of the letter E rather than the descenders of the p and y.

The RichText control is the middle-weight Spark text primitive and is equivalent to the MX Text control.

It can display rich text with multiple character and paragraph formats but it's completely non-interactive and does not support scrolling, selecting or editing.

I’m opening the Cafeteria component file by CTRL + clicking on the component name in the main application file.

I am locating the calorieContent Label control and changing the Label control tags to RichText control tags because I will eventually want to format this content and I don’t need scrollbars.

Within the RichText control tags, I am surrounding the text with a span flow tags, saving the file and running the application.

You can see that the Calorie Information is displayed, but it is not formatted.

I will format the text in the next video.

The RichEditableText control is the heaviest weight of the three text primitives.

Its MX equivalent is the TextArea control, but the RichEditableText control is considered chromeless.

It does render a scrollable rectangle of selectable and editable text and it can draw a background and use all the text formatting styles, but it does not support drawing a border or scrollbars.

You combine it with other components to add the scrollbars.

To create a scrollable RichEditableText control you use the Scroller class.

You surround the text control component instance with a Scroller instance.

I am opening the MonthlyEvents custom component from the main application by CTRL + clicking on it.

This component displays the company's events in the last panel on the right.

I am changing the Label control around the text to a RichEditableText control.

To the opening tag, I’m changing the width property to 100%.

I am saving the application and running it.

You can see that much of the text is not displayed and there is no scroll bar.

When I click the text, you can see that the text is editable.

Back in the MonthlyEvents.mxml file, I am surrounding the RichEditableText control with the Scroller control.

To the Scroller tag, I’m moving the width and height properties from the RichEditableText control.

I’m assigning the editable property to the RichEditableText control with a value of false.

When I save the file and run the application, the Monthly Events Panel container now has a scroll bar on the right and you can see all of the content, which is no longer editable.

For your next step, watch the video titled “Utilizing the Text Layout Framework”.