Flex in a Week, Flex 4.5
Video 4.06: Understanding navigator containers

Up to this point in the training, you have used only layout containers, which control the sizing and positioning of children.

In this video, you will learn how to use the navigator containers, including the ViewStack container, which works with other UI controls to navigate content.

All navigator containers are designed to include a menu of sorts.

For instance, this Accordion container has pleats which allow you to switch between content areas.

The TabNavigator container allows you to navigate content through a tabbed interface.

The TabBar control looks similar to the TabNavigator container, but is really a control, not a container, that works with the ViewStack container and sits above the content, instead of containing it.

You can see that the TabNavigator container surrounds the content with a border, while the TabBar control does not.

You will also learn how to use this Spark ButtonBar control to navigate content as well.

It looks very similar to the TabBar, but is a set of buttons rather than tabs.

First, let me review components in general.

Remember that there are Spark and MX components.

Spark components were introduced in Flex 4 while MX components have been used in Flex 3.

Eventually, there will be equivalent Spark components for every MX component, but for now, you should try to use the Spark components whenever you can but freely use the MX components if necessary.

To further orient you to the topic of this video, remember that there are two types of components.

Controls are UI elements like TextInput, Label or Button components.

Containers are components that are used to lay out controls or other containers.

Both Spark and MX components consist of both controls and containers.

There are two types of containers: layout and navigator containers.

Layout containers, are used to control the sizing and position of their children, which can be controls or other containers.

Navigator containers, on the other hand, control the navigation among child containers.

Their children can only be containers, not controls.

The MX navigator containers are subclasses of the Container class and they do not currently have Spark equivalents.

In this video, you will learn how to use the Accordion and ViewStack containers plus the TabNavigator container, which is a subclass of the ViewStack container

Navigator and layout containers can work together.

All of your content is first laid out in layout containers.

However, you cannot use the Spark containers that you have already learned about like Group and Panel, directly within the MX navigator containers.

The MX navigator containers work with the MX layout containers or the NavigatorContent Spark container.

It is a best practice to use the NavigatorContent Spark container instead of the MX containers whenever possible.

You can place the Spark layout containers and controls inside of the NavigatorContent container.

The navigator containers are then used to control which of the layout containers should be displayed.

You can nest one or more layout containers in a navigator container, but only one layout container can be displayed at a time.

All navigator containers have built-in mechanisms to allow users to switch between child containers.

Remember that containers only affect the layout of their immediate children.

This means that navigator containers do not directly affect the layout of the components within the child layout containers.

This is the main starter application file for this video.

There are three custom components inside of it that represent three departments in this fictional company.

When I run the application, the components just display employees.

I am Control + clicking on the HumanResources component instance to show you that this class file extends the NavigatorContent Spark container.

You can see that this is also true about the other two custom components.

Back in the main application file, you can see that the three component instances are implemented with label properties.

These label values will be used for display in the navigator containers that I will implement next.

The Accordion container defines vertically stacked pleats that animate as they open and close.

Each pleat displays the content of a child layout container and only one pleat can be open at a time.

You implement the Accordion container in MXML with the MX Accordion tag set.

In Flash Builder, I am surrounding the three components with the Accordion tag set from the MX namespace and then indenting the components with the Tab key.

When I save the file and run the application, you can see that the employees are not properly contained within the Accordion dimensions.

However each pleat does display a title based on the label value of the custom component.

I am returning to Flash Builder and adding the resizeToContent property to the Accordion instance with a value of true.

When I save the file and run the application, you can see that the entire layout container is displayed in each pleat.

I am changing the Accordion tag set to instantiate the TabNavigator component.

When I save the file and run the application, you can see the same content, now displayed within a tabbed interface rather than with accordion pleats.

Note that you could increase the width of the container to ensure that all the tab headers appear fully.

The ViewStack navigator container does not actually have a user interface element that allows users to switch between its child containers.

Commonly, it implements a UI control, like the MX LinkBar or TabBar controls or Spark TabBar or ButtonBar controls to navigate the child containers.

Since the ViewStack container uses the UI controls for navigation display, you must bind the ViewStack container’s id property to the dataProvider property of the UI control.

I am replacing the TabNavigator container with a ViewStack tag set.

When I save the file and run the application, you can see that only the first child container's content is displayed.

There is no way to navigate to the other two layout containers.

Back in Flash Builder, I am defining an id property for the ViewStack container and setting its value to employeeDepartments.

Above the ViewStack container, I am creating a TabBar control instance and adding a dataProvider property.

The label properties from the ViewStack container’s layout child containers will be passed to the dataProvider property of the TabBar container through this binding to the employeeDepartments id of the ViewStack.

When I save the file and run the application, you can see that the TabBar container's UI display allows me to navigate between the content children of the ViewStack container.

I am replacing the TabBar control instance with a Spark ButtonBar control.

Again, when I save the file and run the application, you can see that this is just another option for the display.

The functionality is the same.

For your next step, work through the exercise titled “Navigating using navigator containers”.