Flex in a Week, Flex 4.5
Video 5.07: Drawing with MXML graphics

In this video, I will introduce you to the Spark primitive classes that allow you to draw shapes using MXML tags.

I will also show you how to integrate shapes drawn in other tools that can export their graphics in the FXG format.

I've located the spark.primitives in AS Docs and here is a list of classes that you can use to draw in MXML.

Here is the Employee Portal application as it appeared at the end of the last video.

You can see the blue header area has four rounded corners.

Back in the code for the TopContainerSkin class that is applied to that SkinnableContainer for the header, you can see that the blue rectangle is drawn using the Rect MXML tag and a nested fill property.

Note that the SolidColor tag is capitalized, which gives you a hint that it is a class that can be instantiated for other graphics classes, not just the Rect class.

Besides drawing your own shapes using the MXML graphics tags, you can also integrate shapes drawn in other tools.

FXG is an XML-based language for sharing graphics between applications that support the format.

The Adobe Creative Suite can export FXG files and Adobe Flash Catalyst can create FXG graphics that you can import right into your Flash Builder projects.

An FXG file starts with a root Graphic tag and contains a Path node, which defines the actual graphic element code.

Flash Player can render FXG graphics and text primitives, fills, strokes, gradients and bitmaps and filters, masks, alphas and blend modes.

The implementation of the FXG format makes it easier than ever for designers and developers to work together.

For instance, a designer who works in Illustrator can export his graphics to an FXG file, which you can then use in your Flex application.

In Flash Builder, I am opening the TopSkin.fxg file in the skins directory.

This file was generated by Adobe Illustrator CS5.

You can see that it is an XML-based format with a parent Graphic node.

The code that you are interested in is this Path block, which defines the blue background color, with the one rounded right upper corner.

I am opening the TopContainerSkin.mxml file.

I am commenting out the Rect block.

Below the commented tags, I am adding a TopSkin instance.

The skins namespace is used to add the TopSkin.fxg file to the application. When I added the skins tag, the skins namespace was added to the Skin class.

I am saving the file and running the application.

You can see the new TopSkin file has been added to the application. Notice that only the top right corner of the SkinnableContainer skin is rounded.

For your next step, work through the exercise titled “Creating and applying skins”.