Flex in a Week, Flex 4
Video 1.10: Understanding data binding

In the last exercise, you created a user interface for your application and added some styling and skinning to your components.

In the next video, you will learn how to request data from an XML file and display it in the UI components.

Before you do that, however, you need to have a better understanding of data binding, which is the subject of this video and associated exercise.

Intro_app.camrec

A data binding sets up a property to watch another property for changes in value.

In this Employee Registration example, I will show you how to dynamically build an email address from the first and last name text field values in this form.

I am entering Brad for the first name and VanBrocklin for the last name.

The email address field watches the first and last name fields for changes in values and uses those values to build the email address.

Define_id_property.camrec

```
<mx:FormItem label="First Name:">
   <s:TextInput id="firstName"/>
</mx:FormItem>
```

This is the starter file for the application.

I am locating the FormItem container for the first name TextInput control.

In order to watch a property of a component, you must first give the component an identifier that you can use to reference it uniquely in the application.

Remember that, based on the methodology used in this series, we place one property per line except when the properties are related.

Another coding practice is to place the id property on the same line as the MXML tag, since it is the unique identifier for the instance, and will be easier to see at the top.

I am defining an id property for the TextInput control and setting it with a value of firstName.

Notice that I'm using a lowercase f and an uppercase N for the value.

This follows the coding practice to use camel case for variable names.

Email_binding_to_firstname.camrec

```
<mx:FormItem label="Email:">
    <s:TextInput text="{firstName.text}"/>
</mx:FormItem>
```

I want to start by filling the Email form element's TextInput control with the value of the firstName TextInput control.

The key to this functionality is to know that the text property of a TextxInput control holds the value that is displayed in the text field.

I am adding a text property to the email TextInput control and then typing curly braces to denote a data binding.

Then I dictate that the firstName control's text property, which is the property that contains the value typed into the firstName TextInput control, is the value I want to watch.

I am saving the file and running the application.

When I type a first name into the text field, I see that the value is immediately reflected in the Email field.

Watching_firstName_x_property.camrec

```
<mx:FormItem label="Email:">
    <s:TextInput text="{firstName.x}"/>
</mx:FormItem>
```

As you can see here with the Label control that I am using for the Employee Registration header text, components can have many properties.

This header control has text, color, fontSize, and many more properties.

I can watch any of those properties with a data binding.

In the Email form field's TextInput control, I am changing the property to watch to the firstName control's width property.

When I save and run the application, you can see that the output in the Email form field is 196, which is the value of the width property set in the firstName control.

While this isn't a very practical illustration, it does illustrate to you that you can watch any property of a component.

Mix_Bindings_with_Static_Text.camrec

```
<mx:FormItem label="Email:">
   <s:TextInput text="{firstName.text}.{lastName.text}@xyzcompany.com"/>
</mx:FormItem>
```

I've changed the watched property in the Email text control back to the firstName.text property.

I've also given the Last Name TextInput control an id property of lastName.

As the user types the new employee's first and last names into this form, I want to build the email address on the fly.

The standard email address for this fictional company is the first name and last name separated by a period followed by the @xyzcompany.com domain.

Next, I will show you how to mix these static text elements with the data bindings.

After the firstName.text binding, I am typing a period and then adding a binding for the lastName.text watched property.

I am following that with an @ sign and the xyzcompany.com fictional domain.

When I save the file and run the application you can see that the period and the domain appear by default in the Email form field.

As I type the first name and last name values, the email address is generated.

Note that this is a one-way binding.

If I type a first name, say, that has a space in it, like Mary space Lou, the space is also added to the email address.

However, if I remove the space from the email address, this does not change the first name's text property value.

Flex does support two-way binding and you will learn about that in a later video of this series.

Next_step.camrec

Next step

Exercise: Generating an email address using data binding

For your next step, work through the exercise titled “Generating an email address using data binding”.