Data Binding

obecto
passionate about software & people

Що е *Data Binding*?

- автоматично синхронизиране на данни между *source*-property и *destination*-property
- всяка промяна в *source*-property-то ще се отрази автоматично върху *destination*-property-то
- *source*-property-то и *destination*-property-то могат да са във различни обекти
- обвързването може да е двупосочно

[Bindable]

```
[Bindable]
public var isCompleted:Boolean = false;

[Bindable(event="divisibleByThree")]
public var index:Number = 0;
```

- registers a property as bindable (means the registered property can be a *source*-property)
- every change to the value of the property will be automatically copied to the *destination*-property
- the Flex compiler will generate corresponding *getters* and *setters* at compile time
- Flex will automatically turn your class into an *EventDispatcher*

[Bindable]

```
[Bindable(event=”itemsChanged”)]
public var itemsCount:Number = 0;

private var _items:Array;
[Bindable]
public function set items(value:Array):void
{
    if (value && value.length != _items.length)
    {
        itemsCount = value.length;
        dispatchEvent(new Event(“itemsChanged”));
    }
    _items = value;
}
public function get items() : Array
{
    return _items;
}
```

- the synchronization of the *itemsCount*-property will be only triggered when the registered *event* is dispatched
- you can also register a *getter-setter*-pair as *[Bindable]*

obecto
passionate about software & people

[Bindable]

```
[Bindable]
public class EmployeeInfo
{
    public var firstName : String;
    public var lastName : String;
    public var position : Position;
}
```

a whole class can be registered as *[Bindable]*, so all of its properties will be automatically *[Bindable]*

Curly-Brace Syntax (Binding)

```
<s:TextInput id=”firstNameInput”/>
<s:TextInput id=”lastNameInput”/>

<data:CompanyModel id=”model”/>

<data:EmployeeInfo firstName=”{firstNameInput.text}”
    position=”{model.EmployeeOfTheMonth.position}”>
    <lastName>{lastNameInput.text}</lastName/>
</data:EmployeeInfo>
```

- when you register a property as source of data binding, Flex monitors not only the property, but also the *chain of properties* leading up to it

```
<fx:String>{“Concatenating”+bindableVar}</fx:String>
```

- you can have whole expressions inside the binding

```
<fx:Number>{Math.max(width, height)}</fx:Number>
```

- you can call functions with bindable arguments

obecto
passionate about software & people

<fx:Binding>

```
<s:TextInput id=”firstNameInput”/>
<s:TextInput id=”lastNameInput”/>

<data:CompanyModel id=”model”/>

<data:EmployeeInfo id=”info”/>

<fx:Binding source=”firstNameInput.text”
    destination=”info.firstName”/>
<fx:Binding source=”lastNameInput.text”
    destination=”info.lastName”/>
<fx:Binding
    source=”model.employeeOfTheMonth.Position”
    destination=”info.position”/>
```

Remember the *Binding* special tag?

obecto
passionate about software & people

BindingUtils

```
BindingUtils.bindProperty(this, “destination”,
    hostComponent, “source”);
```

```
BindingUtils.bindProperty(this, “destination”,
    hostComponent, [“textInput”, “text”]);
```

```
public function update(value : String) : void
{
    textArea.text = value.toUpper();
}

BindingUtils.bindSetter(update, textInput, “text”);
```

- you can use the *BindingUtils* to programatically create bindings
- these methods return a *ChangeWatcher*, which is an utility class that you can use with bindable properties
- to destroy a binding you must call the *unwatch()*-method of the corresponding *ChangeWatcher*

Обобщение

- що е data binding?
- метатага [Bindable]
- използване на къдрави скоби в MXML
- използване на специалния таг <fx:Binding>
- използване на BindingUtils

obecto
passionate about software & people