

Building ActionScript Components

Aim: To make building custom ActionScript components the *first* option when developing in Flexlandia.

Naming

Don't describe its appearance – eg. RedButton, RoundedBorderedWindow.

Do describe its function or UI element type – eg. ActionBarButton, InspectorWindow, NewElementFormPanel

Extending

Extend UIComponent or as high up the inheritance tree as possible. Loc8 is massive so we want to make sure each component is taking up as little memory as possible.

Avoid extending HBox or VBox! – custom layout is *easy!!*

Property Names

Keep in mind re-use for the data model:

- use generic terms like dataProvider – not assetData, label – not workOrderLabel

Multiple Components

Don't try and implement *everything* in the one component. Break it down further. Generalise!

The benefit is it makes it easier to extend and reuse in multiple situations.

It also allows it to become a style target for simple CSS styling.

MX Source Code

TextMate with the mx sdk source is your best friend.

ActionScript Component Cheat Sheet

Component Children

Child Components

```
private var _icon:SuperImage;
```

note: an improvement here would to have the icon type to be an interface ie. ILoadedImage

Create Children

Create child objects of the component.

```
override protected function createChildren():void
{
    super.createChildren();

    if(!_icon)
    {
        this._icon = new SuperImage();
        this.addChild(this._icon);
    }
}
```

Updated Display List – *where you lay it all out*

Draws the object and/or sizes and positions its children.

```
override protected function updateDisplayList(unscaledWidth:Number,
unscaledHeight:Number):void
{
    super.updateDisplayList(unscaledWidth, unscaledHeight);

    this._icon.x = 0;
    this._icon.y = 2;
    this._icon.width = 16;
    this._icon.height = 16;
}
```

Measure – *helps the component know minimum sizes and measurements*

Calculates the default size, and optionally the default minimum size, of the component.

```
override protected function measure():void
{
    super.measure();

    measuredWidth += 16;
}
```

Component Properties

Property Setup

```
private var _iconSource:String;
private var _iconSourceChanged:Boolean = false;

[Bindable('iconChanged')]
public function set iconSource(source:String):void
{
    if(this._iconSource == source) return;

    this._iconSource = source;
    this._iconSourceChanged = true;

    this.invalidateProperties();
}

public function get iconSource():String
{
    return this._iconSource;
}
```

Commit Properties

Processes the properties set on the component.

```
override protected function commitProperties():void
{
    super.commitProperties();

    if(this._iconSourceChanged)
    {
        this._icon.source = this._iconSource;
        this._iconSourceChanged = false;

        this.invalidateDisplayList();
    }
}
```

Component Events

Meta Data – *defining lets FlexBuilder MXML auto-complete – happy compiler*

```
[Event(name='iconClick', type='mx.events.MouseEvent')]
```

Event Notes

- Minimise internal event handlers – where possible override event handlers from parent component.
eg. override protected function mouseClickedHandler(event:MouseEvent):void