



Flash to Unity's Style Sheets

Friday, October 4th, 2013



Abstract:

This document details the base functionality of Flash to Unity's style sheets and how do they relate to the properties defined in the various elements they can modify. It delves into the several attributes the styles can give, and how to apply them to Flash to Unity objects.



Table of Contents

Introduction.....	4
Styling Components.....	4
Stylizing in Flash to Unity	5
Components that can be Stylized	5
General.....	7
UI Components	7
Texts.....	10



Introduction

The style sheets are components used to separate logic from presentation. At their most basic state, they are definitions for the style of the components they modify. For the effects of this document and Flash to Unity, “style” is defined as the series of modifiable properties a component may have. Style sheets are perfect when several components may need to have identical or very similar property configurations, and the user may find it tedious to modify each and every component on its own.

Styling Components

The style sheet manager is in charge of managing the way each component applies its respective style. It provides base functionality for applying styles, as well as setting transforms and rotations. The style sheet manager works with components that can be styled, which need to derive from the “IF2uStilyzedObject” interface.

Stylized objects, as previously stated, derive from a base interface, which gives them the ability to read specially formatted data from an XML file and apply whatever commands result from the combination of tags in said file. Base stylized objects have three main components:

- A “transform” property: This is the transform used to modify the objects.
- A “StyleKey” property: This is the key used to look for the correct style tag in the XML file.
- A “SetAttributesFromStyle” method: This method allows objects that can be stylized to look for their respective tags and apply whatever changes necessary to match the style contained within. This very method must be implemented on each and every object that can be styled.



Stylizing in Flash to Unity

Flash to Unity reads styles from the “style_sheet.xml” file by default, which is located at “Resources > xmls > settings > f2u.” The file contains several default values, and the user may define more tags at their leisure. The base structure of the XML is as follows:

```
<?xml version="1.0" ?>
<StyleSheet>
    <!--Components-->
</StyleSheet>
```

The next section will detail how to add or modify component tags appropriately.

Components that can be Stylized

Flash to Unity supports stylization of several of its components by default. The following subsections will detail each and every component that can be stylized, along with every attribute that can be modified from the style sheet. Please do note that each subsection will contain an XML snippet regarding the respective component. Please also note that all attributes are optional; a missing attribute will just leave the component with the value unchanged. The following types of attributes are acceptable within Flash to Unity:

- Boolean: may contain either a “true” or a “false”.
- Keycode: May contain the string representation of one of the keycodes used on Unity.
- String: May contain an arbitrary string.
- Int: may contain an integer value.
- Float: May contain a floating point value.
- Float [a:b]: May contain a float within the scope of a and b, inclusive. If the first element of a line is of this type, it’s assumed all floats on that line have the same restrictive scope.



- Priority: May contain either “Low,” “Normal,” “Highest,” or “StopAll.”
- Orientation: May contain either “Vertical” or “Horizontal.”
- InputField: May contain any of these values:
 - “InputField”
 - “MultilineInputField”
 - “PasswordField”
- Keyboard: May contain any of these values:
 - “Default”
 - “ASCIICapable”
 - “NumbersandPunctuation”
 - “URL”
 - “Number”
 - “Phone”
 - “NamePhone”
 - “EmailAddress”
- Anchor: May contain any of these values:
 - “UpperRight”
 - “MiddleRight”
 - “LowerRight”
 - “UpperCenter”
 - “MiddleCenter”
 - “LowerCenter”
 - “UpperLeft”
 - “MiddleLeft”
 - “LowerLeft”
- Overflow: May contain any of these values:
 - “None”
 - “FadeOut”
 - “NewLine”



- "Clamp"

General

Transform properties

The following properties are settable on all components:

```
<Component>
  <Name xPos      = "[float]",
        yPos      = "[float]",
        zPos      = "[float]",
        scale     = "[float, float]",
        rotation  = "[float, float float]",
        color     = "[float[0:0.5], float, float, float]" />
</Component>
```

UI Components

Animated Buttons

```
<F2UUIAnimatedButton>
  <Name throwEventOnPress = "[boolean]",
        shortcut          = "[keycode]"
        waitForAnimation  = "[boolean]" />
</F2UUIAnimatedButton>
```

Buttons

```
<F2UUIButton>
  <Name throwEventOnPress      = "[boolean]",
        shortcut               = "[keycode]",
        waitForSound           = "[boolean]",
        soundGroupNames       = "[string]",
        soundPriority           = "[priority]",
        onClickSound           = "[string]",
        pressedColor           = "[float[0:1], float, float, float]",
        disableColor           = "[float[0:1], float, float, float]",
        rolloverColor          = "[float[0:1], float, float, float]",
        pressedTextColor       = "[float[0:1], float, float, float]" />
```



```
        disableTextColor          = "[float[0:1], float, float, float]",
        rolloverScale             = [float, float, float],
        enableTextTransformations = "[boolean]" />
</F2UUIButton>
```

Hyperlink Buttons

```
<F2UUIHyperlinkButton>
    <Name url                    = "[string]",
        shortcut                 = "[keycode]",
        waitForSound             = "[boolean]",
        soundGroupName           = "[string]",
        soundPriority             = "[priority]",
        onClickSound             = "[string]",
        rolloverTextColor        = "[float[0:1], float, float, float]",
        pressedTextColor         = "[float[0:1], float, float, float]",
        disableTextColor         = "[float[0:1], float, float, float]",
        visitedTextColor         = "[float[0:1], float, float, float]",
        rolloverScale            = "[float, float, float]" />
</F2UUIHyperlinkButton>
```

Scroll Panels

```
<F2UUIScrollPanel>
    <Name itemsDistance          = "[float]",
        orientation              = "[orientation]",
        isDraggable              = "[boolean]",
        isLoopable               = "[boolean]",
        startDragMovement        = "[float]",
        swipeDeclaration          = "[float]",
        returningDeceleration     = "[float]" />
</F2UUIScrollPanel>
```

Sliders

```
<F2UUISlider>
    <Name orientation            = "[orientation]",
        scrollPercentage          = "[float[0:1]]" />
</F2UUISlider>
```



Progress Bars

```
<F2UUIProgressBar>
    <Name progress = "[float[0:1]]" />
</F2UUIProgressBar>
```

Check Boxes

```
<F2UUICheckBox>
    <Name throwEventOnPress = "[boolean]" />
</F2UUICheckBox>
```

Text Inputs

```
<F2UUITextInput>
    <Name throwEventOnPress      = "[boolean]",
        text                    = "[string]",
        maxLength               = "[int]",
        type                    = "[inputField]",
        selectedColor           = "[float[0:1], float, float, float]",
        keyboardType            = "[keyboard]",
        keyboardHideInput       = "[boolean]",
        keyboardAutoCorrection  = "[boolean]",
        keyboardSecure          = "[boolean]",
        keyboardAlert           = "[boolean]",
        keyboardPlaceholderText = "[string]" />
</F2UUITextInput>
```



Texts

<F2UText>

```
<Name position           = "[float, float]",
  scale                   = "[float, float]",
  rotation                 = "[float, float, float]",
  color                   = "[float[0:0.5], float, float, float]",
  letterSpacing           = "[float]",
  lineSpacing             = "[float]",
  useColorGradient        = "[boolean]",
  bottomGradientColor    = "[float[0:0.5], float, float, float]",
  anchor                  = "[anchor]",
  kerning                 = "[boolean]",
  size                    = "[float, float]",
  overflowType            = "[overflow]",
  fadeOutOverflowOffset  = "[float]" />
```

<F2UText>