ClassicPro::Flex

Skinning Manual

Table of Contents

1) Overview..2
2) Getting Started..2
2.1 skin.xml...2
2.2 flex.xml...3
2.3 Skinning Essentials..3
2.3.1 Quick Skin Refresh...3
3) Bitmap Reference...3
3.1 The Player Window...3
3.1.1 background.png *...3
3.1.2 background-alpha.png *..4
3.1.3 background-overlay1.png...4
3.1.4 background-overlay2.png...4
3.1.5 display.png..4
3.1.6 display-overlay.png..4
3.1.7 albumart-region.png..4
3.1.8 display-mask.png ^..5
3.1.9 ghost.png..5

1 Overview

Skinning for Winamp Modern can be very time consuming due to the need of XML and script knowledge. Especially people who are new to coding have big problems getting their great design concepts done as a working Winamp Modern Skin.

ClassicPro::Flex is trying to reduce coding for a freeform skin to an minimum so everyone can create a Winamp Modern skin straight out of a PSD file.

2 Getting Started

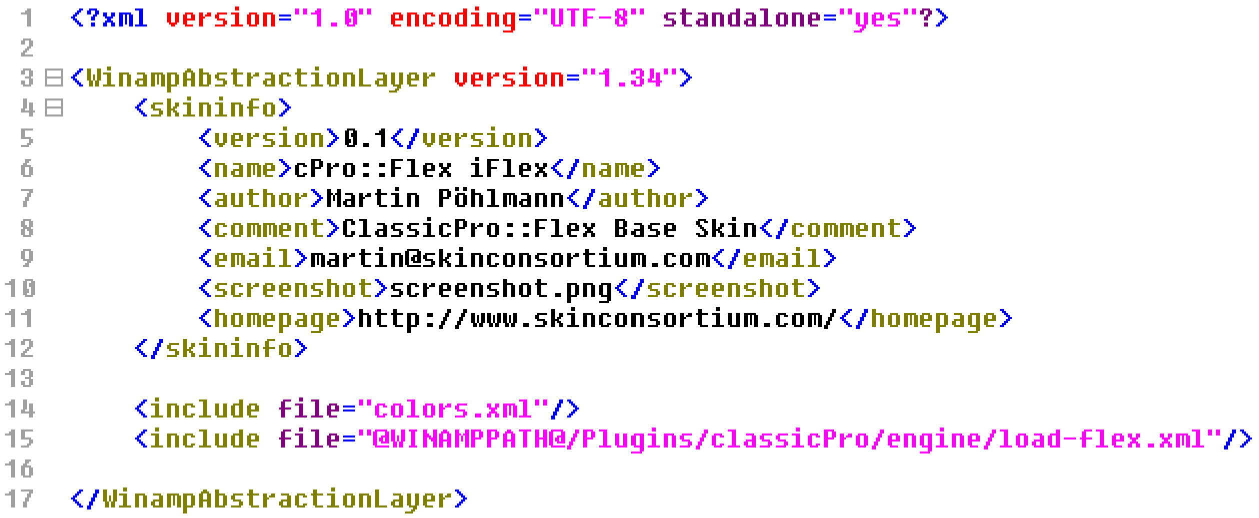
In order to create a new ClassicPro::Flex skin you will first need a design concept. I recommend using Adobe Photoshop for creating a skin design but if you do not want to buy this software GIMP is a very good free alternative.

It is also helpful to create your skin as XION skin first. XION's skin format bases on loading PSD files with tagged layers. So all you need is to give your flattened layer a name like 'play' if you want a button for starting playback. Goto http://xion.r2.com.au for more information about XION.

Back to topic - we want to create a ClassicPro::Flex skin. So goto your Winamp's skin directory (usually located in “C:/Program Files/Winamp/Skins/”) and open the file “cProFlex - iFlex.wal” with a zip program (I recommend 7zip which can be downloaded from http://7zip.org). After this file is opened mark all files located within the archive using CTRL+A and click on extract. Extract the files to a new subfolder within the Winamp skin directory starting with cProFlex, like cProFlex - MySkin. If everything went fine you can open “C:/Program Files/Winamp/Skins/cProFlex - MySkin/“ in Windows Explorer now.

2.1 *skin.xml*

skin.xml is the most important file in this directory. It represents the entry point for the Winamp Modern skinning engine. It includes all author information and loads the ClassicPro::Flex engine. This is how skin.xml will most likely look like:



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<WinampAbstractionLayer version="1.34">
    <skininfo>
        <version>0.1</version>
        <name>cPro::Flex iFlex</name>
        <author>Martin Pöhlmann</author>
        <comment>ClassicPro::Flex Base Skin</comment>
        <email>martin@skinconsortium.com</email>
        <screenshot>screenshot.png</screenshot>
        <homepage>http://www.skinconsortium.com/</homepage>
    </skininfo>

    <include file="colors.xml"/>
    <include file="@WINAMPPATH@/Plugins/classicPro/engine/load-flex.xml"/>

</WinampAbstractionLayer>
```

I will only explain the important things for you, all other stuff should be kept as is. The <skininfo/> Tag should hold your author information. Feel free to alter anything but screenshot.

2.2 *flex.xml*

This is the second most important file and will be parsed by the ClassicPro::Flex engine. [TODO]

2.3 *Skinning Essentials*

Some nice to know stuff before we start creating the skin.

2.3.1 *Quick Skin Refresh*

You can easily refresh the current skin by hitting F5.

3 Bitmap Reference

Now that we have created a working directory for our skin design. So let's make the design go live.

All files suffixed with a * are required files. Any other file can be deleted in order to remove its object from the layout

All files suffixed with a ~ are required if the previously explained file is found in the skin directory.

All files suffixed with a ^ are placed in another position in the image stack (see 3.1).

3.1 *The Player Window*

The Player Window is the window holding all the playback controls and songinformation stuff. Basically there are two layouts: The 'normal' layout and the 'shade' layout.

The images for the normal layout are stored in YOUR_SKIN_DIR/player/normal/ and for the shade layout in YOUR_SKIN_DIR/player/shade/

Both layouts have the same core functionality so both directories hold the same files. Thus I only need to explain how you build the normal layout since the shade layout follows exactly the same rules except in bitmapID/gammaID every occurrence of normal is replaced by shade.

> ! *The order of how the following images are explained does also represent the order of how they are stacked in the layout if nothing else is mentioned. So background.png will be stacked below background-overlay1.png and so on.*

3.1.1 *background.png **

```
BitmapID:   cpro.flex.player.normal.bg
GammaID:    player.normal.background
```

This file represents the background of your layout if no desktop alphablending [TODO screenshot] (smooth edges, shadows) is used and thus it defines the height and width of your layout. No other bitmaps are displayed in the area exceeding the bounding defined via this image!

Every time you save this file with a new dimension the layout will also get a new height/width if the skin is reloaded.

! *Keep in mind that buttons and other objects may 'loose' their correct placement if you resize this image!*

3.1.2 background-alpha.png *

```
BitmapID:   cpro.flex.player.normal.bg.alpha
GammaID:    player.normal.background
```

This file represents the background of your layout if desktop alphablending is enabled so it can include smoothed edges and a drop shadow.

!! *File This file MUST have the same height/width and should have the same overall design as background.png*

3.1.3 background-overlay1.png

```
BitmapID:   cpro.flex.player.normal.bg.overlay1
GammaID:    player.normal.background.overlay1
```

This file represents an overlay drawn above the background image and may be used to create sections with different colorthemes.

! *Keep in mind that all of the following files may not need to have the same height/width as background.png but I recommend to use the same size since it is easier just hiding the non-necessary layers in Photoshop and save the ones needed for a specific image (like a metal overlay).*

3.1.4 background-overlay2.png

```
BitmapID:   cpro.flex.player.normal.bg.overlay2
GammaID:    player.normal.background.overlay2
```

This file represents an overlay drawn above the background image and background-overlay1 and may be used to create sections with different colorthemes.

! *For creating more themeable overlays you need to define them in the group player.normal.underlay. Please refer to [TODO]*

3.1.5 display.png

```
BitmapID:   cpro.flex.player.normal.display
GammaID:    player.normal.display
```

Another overlay commonly used as the background of the display. Of course you can use this for anything else if you do not have a display in your design concept.

3.1.6 display-overlay.png

```
BitmapID:   cpro.flex.player.normal.display.overlay
GammaID:    player.normal.display.overlay
```

This bitmap may be used to create structures above the display background. It has a different GammaID as the Background and thus it can be colored in a complete different way.

3.1.7 albumart-region.png

```
BitmapID:   cpro.flex.player.normal.display.overlay
```

```
GammaID:    none
```

This image defines the bounding rect for an albumart object. Remove this file if you do not need such an object.

3.1.8 display-mask.png ^

```
BitmapID:   cpro.flex.player.normal.display.mask
GammaID:    player.normal.display
```

Used to create half hidden areas in the display (see Songticker in iFlex skin). Has the same gammagroup as the display and is not clickable (ghosted).

!! *This overlay is placed on top of the image stack below ghost.png so it can mask all elements like Songticker of Albumart.*

3.1.9 ghost.png

```
BitmapID:   cpro.flex.player.normal.ghost
GammaID:    player.normal.ghost
```

Commonly used to create a glass overlay above the display area. This image is not clickable.