

***JRiver MediaCenter  
Standard View  
Skinning Guide***

# Table of Contents

Introduction .....	3
<SKIN> .....	8
<SCALE> .....	8
<GLASS> .....	8
<Effects> .....	8
<ART> .....	9
<MAINFRAME> .....	10
<TOOLBARS> .....	12
<MENU> .....	14
<PLAYERBAR> .....	16
The PlayerBar Control Buttons .....	16
The PlayerBar Display Panel.....	16
The PlayerBar Search Box .....	18
The PlayerBar Volume Controls .....	19
The PlayerBar Position slider/progress bar .....	19
The PlayerBar Shuffle Buttons.....	19
The PlayerBar Continuous Play Buttons .....	20
The PlayerBar DSP Buttons.....	20
The PlayerBar NextDisplay and PreviousDisplay Buttons.....	20
<TREE> .....	21
<ACTIONWINDOW> .....	22
<TAGWINDOW> .....	24
<SPLITTER> .....	25
<VIEWHEADER> .....	27
<SPLITVIEWTAB> .....	28
<CARRADIO>.....	29
<LIST> .....	30
<SCROLLBAR> .....	32

## Introduction

Skinning is the customization of the MC Standard View User Interface. It is based on graphics and simple coding instructions to determine how the program looks and acts. Anyone can modify an existing MC Skin or create a new one to share with the MC community or for personal use.

The easiest way to create your own Standard View skin for JRiver MediaCenter is to modify an existing one. You can skin absolutely everything or select only certain parts to change.

P.S. If changing an existing skin or using parts of one to create a new skin Please either ask for that/those skin creators' permission, if they are still active, or acknowledge that your skin is based on other people's work in the main.xml for your creation.

Skinning involves making or altering image files to create the graphic buttons, frames, and icons etc. that appear on the software. As well as working with images, some modification of an xml file is required. Xml is a coding language and looks similar to html code. It controls things like image placement, colors of elements, stretching of elements, and so forth. The xml file may look daunting at first, but it is quite straight-forward once you understand how it is laid out. The various graphic elements of the player are placed in default positions, and this only changes if modifications are made to the xml file.

Therefore, in order to skin the MC Standard View UI, you have to modify:

- A set of images (frame, buttons, sliders etc.)
- The main.xml document for your particular skin

### **Some Important Facts.**

Creating or modifying a JRiver MediaCenter Standard View skin is not a 5-minute task nor, as you will discover later in this guide, is it as simple/easy to achieve as anyone would prefer.

So, you are going to require: - Time, Patience and the Tools that will enable you to achieve your aim/desire.

JRiver MediaCenter runs on the Windows, Apple Mac and Linux platforms and each one has its own idiosyncrasies.

The Windows and Apple Mac platforms are not Case Sensitive but the Linux platform IS therefore every statement/instruction and image name in a Skins main.xml file must use the same capitalization in their names. This is extremely important when it comes to the instructions/commands in the main.xml file; if you use mis-matched capitalization you will end up with an incomplete/broken skin on the Linux platform.

The Windows and Apple Mac platforms both have a Toolbar, on the Apple Mac it's called the Dock, but Linux doesn't.

On the Windows and Linux platforms the Windows Control Buttons, the ones used to minimize, maximize/restore or close a window, are always placed in the top right-hand corner of the window but on the Apple Mac platform they are placed in the top left-hand corner of the window in a different order.

The Apple Mac has a couple of idiosyncrasies the other two don't have, it has its own Tool/Menu Bar and the Menu Headers that appear on the Top Toolbar under Windows and Linux get moved up to the Apple Mac Tool/Menu Bar. For any Pop-up Window, like the DSP Studio Window, regardless of what skin is being used the pop-Up window header bar gets replaced by one that uses Black Text on a White Background accompanied by the Apple MacOS Close, Minimize and Maximize Buttons; but for most of the MC pop-up windows the Minimize button is "Greyed Out" to indicate it is not available for use.

## **Software tools required for making/modifying images**

You will need a graphics program (Photoshop, Paint Shop Pro, GIMP or even the simple paint program that came with your PC).

## **Allowed Image Formats**

The MC skinning engine can handle images in png, svg, bmp and gif formats.

**BUT** only png and svg images can contain areas that have no texture/color in them. Any part of a bmp or gif image that MC is to consider transparent has to have a texture/color with a hex-code of FF00FF applied to it.

## **Images used as Buttons**

Buttons can have up to 5 States: - Normal, Over (as in mouse-over), Pressed, Disabled and Clickable/Target Area (the parts of the image, usually only the visible parts, where the user can use a mouse click, and is normally only used with circular buttons).

Thankfully the MC skinning engine allows you to create a single multi-state image that contains the images for every button state your skin needs/uses. Those state-images sometimes referred to as sub-images, tiles or sprites, must all be the same size.

If you have been using MC for a while you may well have noticed that clicking on some buttons causes an immediate action/reaction. For example, when MC is sitting idle in Playing Now with some un-played tracks in the details list the Play Button is displayed in the player bar but when you click on it MC immediately replaces the Play Button with the Pause Button, the same happens in reverse when you click on the pause button.

For a lot of the MC Buttons, unless they are ones that use the 5<sup>th</sup> state, only 3 button-state sub-images are really needed, although I have encountered some skins where there are only 2-states being used for the Windows Control Buttons.

**CAUTION** – Some MC buttons use images only have 2 sub-images in them that are not states for a single button instead they are what could be considered either/or buttons; as in sub-image 2 replaces sub-image 1 when the button is clicked.

A good example of this type of button is the one at the right-hand end of the ViewHeader in Playing Now used to switch into Cover-View, what is being displayed is the 1<sup>st</sup> sub-image. But when MC switches to Cover-View what is displayed at the right-hand end of the header bar is the 2<sup>nd</sup> sub-image. Another example is the buttons, usually small triangles, displayed on the Tree that are used to Collapse/Expand Tree Sections. When collapsed one of the sub-images, from the Image holding those buttons, is displayed and when expanded the other sub-image is displayed.

It is not always possible to create detailed images at a size MC normally expects/uses. Thankfully, provided you create those images at one of the Sizes that appear on the View > Size Menu that is above 100% MC provides a Scale= instruction you can use to get MC to Scale those images down to the size MC will use them at.

## **The importance of the main.xml document**

The main.xml document is laid out to identify different areas of the application. It is divided into sections, such as MAINFRAME, PLAYERBAR and ACTION WINDOW to show where it interacts with a certain part of the player. Where it indicates a particular image by name it will also indicate the variables that pertain to that image. Variables such as number of states, how that image stretches, any transparent color associated with that image. This is also where many text and background color changes are made. It is good to become familiar with this document as you will find yourself referring to it quite often. One of the reasons why you should allocate plenty of time when creating a new skin is simply because, just like skinning a cat, there are countless ways to create an MC standard view skin and there isn't a single skin available that uses all of them so you are probably going to have to download quite a few of the available skins to find ones that contain instructions similar to the ones you will need especially when it comes to handling images.

## **Software tools required for creating/modifying a main.xml**

You will also need a good text editor preferably one that can differentiate between the various programming/scripting languages such as Notepad++ (Windows Only) Notepad (included free with Windows) can be used as long as you are extremely careful as it will not be able, unlike better editors, to detect any syntax or some other potential errors.

Your aim, especially when creating skins, should be to come up with a skin other people like and would willingly use. So never forget that some MC users may be running MC under a different, older, version of the operating system or a different platform.

## **Some of the things you should avoid: -**

Lack of contrast between Background and Text Colors. Using dark text on a dark background or light text on a light background is one of the MC users' favorite targets for criticism.

Don't use the fact that Microsoft or Apple do things in a particular way as an example of using those methods in your skin. Both Microsoft and Apple are prone to changing their ideas about how to present things and don't always use their own preferred way in every app or OS window/view. There's a reason why MC users may be on a different OS to you and one of them is because they don't want, or detest, the way Microsoft and Apple try to make the OS and every Ap look the same.

Your aim should be to gain praise and not criticism because some MC user's will remember if you created a skin that annoyed them and never even look at any other skin you may create.

## **Knowing where stuff goes.**

So, where are these images and xml file? When you download JRiver Media Center the application is contained in various folders that reside on your PC hard drive. All Standard View skins are found in the Skins folder: -

.....\J River\Media Center nn\Skins\Standard View

Where nn is the version of MC installed on your computer

In the Skins folder you will see that each skin has its own folder. For example, there is a folder entitled Aruba, and this contains all the elements that form the Aruba skin (graphics and xml file).

**BUT** do not try to modify the skins main.xml file or any of its images while its folder is still in the relevant MC folder.

Instead, you should copy the folders for any skins you want to change or use as a base for your new creation to somewhere like your designated Documents Folder where it will be safe from being unintentionally overwritten. This is extremely important if it is one of the MC "Stock Skins", the ones that get installed by an MC Update, you are modifying.

Thankfully all of the Stock Skins and the Skins available from the MC Skins Download Page that were created by people who are no longer active in MC now use the same

main.xml layout and it is the various sections of that uniform main.xml the next part of this guide will lead you through

## <SKIN>

This is probably the most important section in the main.xml because it's where you tell MC the Name of your skin, if it's a new skin, or its new name, if you are modifying an existing skin.

Using a single instruction in this format

```
<SKIN Name="Skin Name" Author=" " EMail="" About=" "/>
```

The only mandatory parts are the Name="Skin Name" instruction and, unless you really wish to remain anonymous the Author="" one

**NOTE** – Because MC and the OS, unfortunately, allows 2 skins with the same Skin Name, but Different Folder Names, to exist in the Skins/Standard View Skins Folder the Name you assign to Your Skin must be Unique for a New Skin Or a Unique Variation of the Name of the skin you are modifying.

If you do not give a "Stock Skin" you are modifying a New Unique Name and do not change the Name of the Folder it is in the next time MC gets updated the unmodified "Stock Skin" will overwrite your Modified version of that skin.

A similar situation will arise if you modified a "non-Stock Skin" and for any reason reinstall the original version of that skin.

But if you modify a Skin without changing the Skin Name but do change its Folder Name you end up in the awkward situation where your platforms equivalent of Explorer lists 2 Folders with Different Names but MC Lists 2 Skins with the Same Name

## <SCALE>

The MC Skinning Engine defaults to using images at the size it is set up to use. However, it is not always possible to create detailed images at the size MC requires/uses

When the Scale function/option is enabled the Skin Creator gains the ability to instruct MC, on an image-by-image basis, what scaling factor to use when handling individual images.

## <GLASS>

Switching this function/option on gives certain images a Glassy Look by allowing anything sitting behind Transparent Parts of the Main UI to be visible

Both the Scale and Glass functions/options are turned on using an instruction in this format

```
<Data Enabled="1"/> OR <Data Enabled="0"/> to turn them off
```

## <Effects>

Enables the Effects Brightness function/option that allows the skin to transition between Day (Lighter) and Night (Darker) Modes using an instruction in the format

```
<Data Enabled="1" Style="1"/>
```

The transition from Darker (Night) to Lighter (Day) happens, gradually, in the 30 minutes before Sunrise (Local Time)

Whilst the transition from Day to Night happens, gradually, in the 30 minutes before Sunset (Local Time)

## <ART>

Skin Creators are allowed to use Modified Versions of some of the Icons/Images in the relevant JRiver/MediaCenter nn/Data/Default Art Folder that appear in various parts of the User Interface.

This section is used to Instruct the skinning Engine that it is to use the Modified Icons/images in the relevant skins folder and not the default ones using one instruction in this format for every Modified Icon/Image it is to use.

```
<Entry Name="MC Name" Bitmap="Image name"/>
```

Where:

MC Name is the name of the Image/icon in the Data/Default Art Folder

Image name is the name of the Image/Icon in the Skins Folder to be used as its replacement.

Although not necessary/essential in order to keep any such images/icons together and to make them more easily recognizable some skin creators prefer to add a prefix, i.e. to the name of the replacement image.

**N.B.** Although there are several Images/Icons in the Default Art Folder that contain sub-images; if your skin is using a modified version of SmallIcons.png your Skin Folder Must also contain a copy of the Images.xml file from the Data/Default Resources Folder.

Images.xml acts as the Index for the sub-images in SmallIcons.png and the copy in your skin folder must contain an entry for every modified sub-image in your version of SmallIcons.png.

## <MAINFRAME>

Controls the look/appearance of the Top, Left, Right and Bottom Borders of the Main Window UI; is also used to place the Platform Dependent Windows Control Buttons (Close, Minimize and Maximize/Restore) in the Top Left or Right Corner of the TopBorder

### Assets

MC Name		Type	Purpose
Data	Optional	See note 1	Controls how small the user can make the Main UI
TopBorder	Mandatory	Image	Supplies the image used for the large Main UI top border
LeftBorder	Mandatory	Image	Supplies the image to use for the Main UI left border
RightBorder	Mandatory	Image	Supplies the image to use for the Main UI right border
BottomBorder	Mandatory	Image	Supplies the image to use for the Main UI bottom border
MinimizeButton	Mandatory	Multi-State Image	Used to minimize the Main UI
MaximizeButton	Mandatory	Multi-State Image	Used to put the Main UI into Fullscreen Mode
RestoreButton	Mandatory	Multi-State Image	Used to return the Main UI to Windowed Mode
CloseButton	Mandatory	Multi-State Image	See note 2
Splitter	Mandatory	<Colors ... />	See note 3
SlimPlayer	Optional	Image	Supplies the image to use for the Slim Player (see note 4)
FullscreenPlayer	Optional	Image	Supplies the image to use for the Fullscreen Player (see note 4)

### Notes

1. Uses an instruction in the  
`<Data MinimumWidth="nnn" MinimumHeight="nnn"/>`  
 format where nnn is the relevant size in pixels.
2. In Standard View the CloseButton is used to Close MC  
 In Mini View the CloseButton's action is dependent on the Tools > Options > General > Behaviour > Mini View close mode setting; where it can be set to either Close the program or Return to Standard View.  
 In Cover and Display View the CloseButton will be on the Dropdown PlayerBar and its purpose is to return the user to the Previous View
3. This instruction is used to specify the Hexadecimal values for the Colors to be applied to the Splitter and it's Arrows in this format  
`<Entry Name="Splitter" Face3DColor="nnnnnn" Shadow3DColor="nnnnnn" Hilight3DColor="nnnnnn"/>`  
 Where nnnnnn is the hexadecimal value of the color to be used

### Colors used on the Splitter

MC Color Name	Alternative Name	Type	Purpose
Face3D	Face3DColor	Optional	Colors the splitter bars if an image file is NOT specified in the dedicated SPLITTER section
Shadow3D	Shadow3DColor	Optional	Colors the Arrows on the splitter bars
Hilight3D	Hilight3DColor	Optional	Colors the edges of the Arrows on the splitter bars to create a 3D effect

**N.B.** This is the only place in the main.xml where you can customise the look of the Arrows, by setting a Shadow3D and Hilight3D Color, that appear on the Splitter

4. The SlimPlayer and FullscreenPlayer are 2 versions of the DrowDown PlayerBar used in Cover and Display View and the one used in Mini View.  
The SlimPlayer is a smaller version of the Standard View PlayerBar that provides similar, MC view dependent functionality to the Standard View PlayerBar  
The FullscreenPlayer is also a smaller version of the Standard View PlayerBar with a layout that reduces its functionality

**N.B.** If the instructions for the SlimPlayer and FullscreenPlayer are not present MC will default to using the SlimPlayer layout.

**NOTE** If both the FullScreenPlayer and SlimPlayer are specified in the MainFrame Section then regardless of the order they appear in MC will only use/display the FullScreenPlayer!!!

### **Instructions used for the Platform Dependent Windows Control Buttons.**

**Note** – Because of the way the MC Skinning Engine is structured you have to provide 3 almost identical instructions, 1 for each platform, for each of the Windows Control Buttons

These take the form of:-

```
<Entry Name="MC Button Name" Platform="Platform name" Bitmap="Image name" NumberImages="n" Alignment="a" OffsetX="x" OffsetY="y"/>
```

Where:-

MC Button Name is the appropriate MC Button name from the Assets Panel

Platform name is either Windows, Linux or Mac

Image name is the name of the Multi-State Image to be used

The custom is to use the same Images, Alignment, OffsetX and OffsetY values for use on the Windows and Linux Platforms, but you have to create/supply a set of Round Button Images for use on the Mac Platform. CloseButton = red, MinimizeButton = yellow, MaximizeButton and RestoreButton = green.

n - NumberImages is the actual number of states/sub-images in that Image.

a - Alignment = 0 (Left Aligned) for the Mac Platform or 2 (Right Aligned) for the Windows and Linux Platforms.

x - OffsetX is the horizontal distance, in pixels, from the relevant alignment edge, must be a positive number for left aligned buttons (Mac) and a negative number for right aligned buttons (Windows and Linux).

y - OffsetY is the vertical distance, in pixels, from the top edge.

## <TOOLBARS>

Controls the look and location of the Dropdown Menu Headers/Names (Text) and Shortcut Icons (Images) and Names (Text) on the MainFrame Top and Bottom Borders.

### Assets

MC Name		Type	Purpose
TopBar	Mandatory	See note 1	Controls the look/positioning of the Menu Headers and/or Optional Shortcut Icons and Names on the Mainframe TopBorder
BottomBar	Mandatory	See note 2	Controls the look/positioning of the Optional Menu Headers and/or Shortcut Icons and Names on the Mainframe BottomBorder
StatusBar	Optional	See note 3	Provides the ability to use an image and/or a different text color to be used on the StatusBar part of the Mainframe BottomBorder.
Colors	Mandatory	<Colors ... />	See next panel

### Colors used on the Top and Bottom Toolbars

MC Color Name	Alternative Name	Type	Purpose
Text	TextColor	Mandatory	Colors Menu Headers and Icon Names
HilightText	HilightTextColor	Optional	Is applied to any Menu Header or Icon Name when the mouse hovers over it.
SelectedText	SelectedTextColor	Optional	Is applied to any Menu Header or Icon Name when Selected (Clicked).
Separator	SeparatorColor	Optional	Is applied to the thin vertical lines user can add to toolbars to visually separate/group items

### Notes

- As with the placing of the Windows Control Buttons in the MainFrame section you have to supply both positioning and color instructions for each platform in this format

```
<EntryGroup Name="TopBar">
<Entry Platform="Windows" OffsetX="x" OffsetY="y">
<Colors Text="nnnnnn" HilightText="nnnnnn" SelectedText="nnnnnn"
Separator="nnnnnn"/>
</Entry>
<Entry Platform="Linux" OffsetX="x" OffsetY="y">
<Colors Text="nnnnnn" HilightText="nnnnnn" SelectedText="nnnnnn"
Separator="nnnnnn"/>
</Entry>
<Entry Platform="Mac" OffsetX="x" OffsetY="y">
<Colors Text="nnnnnn" HilightText="nnnnnn" SelectedText="nnnnnn"
Separator="nnnnnn"/>
</Entry>
</EntryGroup>
```

Where:-

x - OffsetX is the horizontal distance, in pixels, from the relevant alignment edge, must be a positive number for left aligned buttons and a negative number for right aligned buttons.

y - OffsetY is the vertical distance, in pixels, from the top edge.

nnnnnn is the hexadecimal value of the color to be used

For Windows & Linux Platforms the Menus/Icons are usually placed slightly in from the left edge of the toolbar but for the MAC Platform they are placed to the right of the Left Aligned Windows Control Buttons by increasing OffsetX

Since MC uses the same TopBar Image on all 3 platforms the colors used on the TopBar should also be the same and as with the Windows Control Buttons in the Mainframe section it is customary to use the same OffsetX and OffsetY settings for the Windows and Linux platforms.

**N.B.** – The Menu Headers and Shortcut Icons are usually Left Aligned, the MC Default, on both the Top and Bottom ToolBar for some very obvious reasons.

2. The BottomBar sits on the MainFrame BottomBorder and as with the TopBar you have to supply both a positioning and colors instruction for the BottomBar in this format

```
<Entry Name="BottomBar OffsetX="x" OffsetY="y">  
<Colors Text="nnnnnn" HilightText="nnnnnn" SelectedText="nnnnnn"  
Separator="nnnnnn"/>  
</Entry>
```

Where:-

x - OffsetX is the horizontal distance, in pixels, from the relevant alignment edge, must be a positive number for left aligned buttons and a negative number for right aligned buttons.

y - OffsetY is the vertical distance, in pixels, from the top edge.

nnnnnn is the hexadecimal value of the color to be used

**N.B.** – Except when the MainFrame BottomBorder is a different color to its TopBorder its customary to use the same Color Settings for the Top and Bottom Toolbars.

3. The StatusBar is technically the part of the BottomBar between the right-hand edge of any Menus and/or Shortcut Icons/Text the user has added to the BottomBar and the right-hand end of the BottomBar; and any text appearing in it is displayed centred within it BottomBar in this format

```
<Entry Name="StatusBar" Text="nnnnnn"/>
```

Or this format if an image is to be used

```
<Entry Name="StatusBar" Bitmap="Image Name" Margins="20,0,20,0"  
DrawMode="REGION_B_HSTRETCH" Text="nnnnnn" />
```

Where:

nnnnnn is the hexadecimal value of the color to be used

Image Name is the name of the image to be used

the rest of the statement are any additional instructions used to control the drawing of that image.

## <MENU>

Controls the look/appearance of all the MC Menus and unlike any other section/part of the MC UI the Menu Background can be either an Image or a Color plus an Optional Border Color

### Assets

MC Name		Type	Purpose
Back	Optional	Image	Image to provide the Menu Background See note 1
Colors	Mandatory	<Colors ... />	See next panel
Item	Optional	Image	A transparent/glassy image that sits over Menu Entries when a mouse hovers over them See note 2

1. The back image is supplied using this format style

```
<Entry Name="Back" Bitmap="Image name" Margins="4,2,2,2"
DrawMode="REGION_E_HSTRETCH,REGION_E_VTILE"/>
```

Where Image name is the name of the Image to be used and the rest of the statement are any additional instructions used to control the drawing of that image.

2. The optional Item image is supplied using this format style

```
<Entry Name="Item" Bitmap="Image name" NumberImages="2" Columns="4,?-Flex,4"
Rows="4,?-Flex,4,?-Flex,4"/>
```

Where Image name is the name of the Image to be used and the rest of the statement are any additional instructions used to control the drawing of that image.

### Colors used on the Menus

MC Color Name	Alternative Name	Type	Purpose
Back	BackColor	Optional	Background color for menu if an image is not used
Border	BorderColor	Optional	Color applied as a border around the background if an image is not used
Text	TextColor	Mandatory	Text color for every Active/Available option/entry in the menu
SelectedBack	SelectedBackColor	Optional	Background color for the selected option/entry in the menu it is also used as the background of an available option/entry when the mouse hovers over it
SelectedText	SelectedTextColor	Optional	Text color for the selected option/entry in the menu
Disabled	DisabledColor	Optional	Color for any option/entry that is Not Available (Disabled)
Separator	SeparatorColor	Optional	Color of the horizontal separators used to visually separate/group items in the menus

And nnnnnn is the hexadecimal value of the color to be used

**NOTE:** SelectedText and/or SelectedBack are NOT applied/used for disabled items/entries in the menu.

## <PLAYERBAR>

Controls the look/appearance of the PlayerBars in Standard, Cover, Display and Mini Views

### The PlayerBar Control Buttons

#### Assets

MC Name		Type	Purpose
PreviousButton	Mandatory	Multi-State Image	Supplies the image for the previous button
PlayButton	Mandatory	Multi-State Image	Supplies the image for the play button
PauseButton	Mandatory	Multi-State Image	Supplies the image for the pause button
StopButton	Optional	Multi-State Image	Supplies the image for the stop button
NextButton	Mandatory	Multi-State Image	Supplies the image for the next button

**NOTE** The Play and Pause Buttons are displayed in the same location on the Player Bar. If MC is not playing anything the Play Button will be displayed but if it is playing something the Pause Button will be displayed.

### The PlayerBar Display Panel

This is the large rectangular panel used to display information about the track MC is currently playing; whose content, alignment and presence/absence of optional eye-candy is totally under the user's control.

#### Assets

MC Name		Type	Purpose
Display	Mandatory	See note 1	
Colors	Optional	<Colors ... />	See next panel
DisplayFont	Optional	See note 2	Controls the Size and Weight of the Title and Status lines Text in the player bar display panel

1. The PlayerBar Instruction is supplied using this format  
<Entry Name="Display" Bitmap="Image name" Margins="n,n,n,n"  
DrawMode="REGION\_E\_HSTRETCH,REGION\_E\_VSTRETCH"/>

If none of the Optional Assets are added

OR this one if they are

```
<Entry Name="Display" Bitmap="Image name" Margins="n,n,n,n"  
DrawMode="REGION_E_HSTRETCH,REGION_E_VSTRETCH">
```

Where:

Image name is the name of the image to be used

The Margins values are used to control the size of the DisplayPanel and the rest of the statement are any additional instructions used to control the drawing of the image.

**NOTE** - A Bitmap is only needed to provide a border around the DisplayPanel or to make it visually different to the rest of the part of the MainFrame TopBar it is sitting on.

2. The DisplayFont Instruction is supplied using this format

```
<DisplayFont TitleSize="sss" TitleWeight="www" StatusSize="sss"  
StatusWeight="www"/>
```

Where:

TitleSize and StatusSize Appears to be a number that when divided by 100 is used to change the Font Size.

TitleWeight and StatusWeight appear to be series of numbers that run from 100 to 900 and increment by 100 used to indicate a characters Thickness/Weight, where 100 equates to Thinnest, 400 equates to Normal, 700 equates to Bold and 900 equates to Boldest

**N.B.** Determining what the Font Size and Weight Values actually do is something I will investigate further

### Colors used in the PlayerBar Display Panel

MC Color Name	Alternative Name	Type	Purpose
TitleText	TextColor	Optional	Used to color the text in the upper line of the display panel
StatusText	HilightTextColor	Optional	Used to color the text in the lower line of the display panel
WaveformBarA	SelectedTextColor	Optional	Used to color the un played part of the track
WaveformBarB	SeparatorColor	Optional	Used to color the played part of the track

**NOTE** The Waveform Bar is displayed instead of the Position slider/progress bar for tracks/videos that have Waveform Metadata when the PlayerBar Display Panel "Show Waveform (if available)" option is turned on

The skin creator is also allowed to use a <Style ... /> instruction that is used to indicate if the Progress Bar, Shuffle, Continuous Play and DSP Buttons are to appear inside or outside (the MC default) of the Display Panel.

However, I have encountered Skins where the Progress Bar, Shuffle, Continuous Play and DSP Buttons are inside the Display Panel without using a <Style ... /> instruction.

So, I'm going to have to do some extensive research to determine what factors are used to control when the Progress Bar, Shuffle, Continuous Play and DSP Buttons are displayed inside the Display Panel.

## The PlayerBar Search Box

### Assets

MC Name		Type	Purpose
Search	Mandatory	Image	Supplies the image for the player bar search box
Colors	Optional	<Colors ... />	See next panel
SearchMenuButton	Mandatory	Image	Supplies the image for the button that appears at the front of the search bar that is used to give access to the Search Menu
SearchClearButton	Optional	Image	Supplies the image for the button that appears at the end of the search bar that is used to clear any search box content

### Colors used in the PlayerBar Search Box

MC Color Name	Alternative Name	Type	Purpose
Text	Text Color	Optional	Used to color any text entered into the Search Box
SelectedBack	SelectedBackColor	Optional	Supplies the background color for a selected entry in the search box
SelectedText	SelectedTextColor	Optional	Supplies the text color for the selected entry in the search box

**N.B.** Any colors that are applied to the text in the PlayerBar Search Box are also applied to any text in the Search Box at the bottom of the Advanced tag Window

**NOTE** – The PlayerBar Search Box is the only PlayerBar Element that the Skin Creator can control the position/location of by using a `OffsetY="n"` instruction to modify/adjust its Vertical Location; where n is the distance in pixels below the Top of the MainFrame.

## The PlayerBar Volume Controls

### Assets

MC Name		Type	Purpose
MuteOffButton	Optional	Multi-State Image	Supplies the image to be displayed when MC is Not Muted See note 1
MuteButton	Optional	Multi-State Image	Supplies the image to be displayed when MC is Muted See note 1
VolumeSlider	Optional	Multi-State Image	Supplies the image to be used for the volume slider track and/or background
VolumeSliderFront	Mandatory	Multi-State Image	Supplies the image to be used for the volume slider foreground
VolumeSliderThumb	Optional	Multi-State Image	Supplies the image to be used for the volume slider thumb/handle

1. Because the MuteOffButton and MuteButton are displayed in the same location on the PlayerBar only 2-states (normal and over) are ever seen.  
Clicking on whichever one is displayed provides access the Volume Controls Menu.

## The PlayerBar Position slider/progress bar

### Assets

MC Name		Type	Purpose
PositionSlider	Mandatory	Multi-State Image	Supplies the image to be used for the position slider track and/or background
PositionSliderFront	Mandatory	Multi-State Image	Supplies the image to be used for the position slider foreground
PositionSliderThumb	Optional	Multi-State Image	Supplies the image to be used for the position slider thumb/handle

## The PlayerBar Shuffle Buttons

### Assets

MC Name		Type	Purpose
ShuffleOffButton	Mandatory	Multi-State Image	Supplies the image used to indicate that Shuffle Mode is turned Off
ShuffleButton	Mandatory	Multi-State Image	Supplies the image used to indicate that Shuffle Mode is turned On
ShuffleAutoButton	Optional	Multi-State Image	Supplies the image used to indicate that Auto Shuffle Mode is turned On

## The PlayerBar Continuous Play Buttons

### Assets

MC Name		Type	Purpose
ContinuousOffButton	Mandatory	Multi-State Image	Supplies the image used to indicate that Continuous Play Mode is turned Off
ContinuousButton	Mandatory	Multi-State Image	Supplies the image used to indicate that Continuous Play Mode is turned On
ContinuousSongButton	Optional	Multi-State Image	Supplies the image used to indicate that Continuous Play Mode is turned On for the Current Track
ContinuousStopButton	Optional	Multi-State Image	Supplies the image used to indicate that MC is in "Stop after every track" Mode

## The PlayerBar DSP Buttons

### Assets

MC Name		Type	Purpose
DSPButton	Mandatory	Multi-State Image	Supplies the image used for the DSP Button
DSPDirectButton	Optional	Multi-State Image	Supplies the image used for the DSP Direct Button That is displayed when MC is using a Direct Audio Path

**NOTE** The DSPDirectButton is displayed instead of the DSPButton if one is provided and only if MC is using a Direct Audio Path.

## The PlayerBar NextDisplay and PreviousDisplay Buttons.

These 2 Buttons are only displayed on the Display View PlayerBar under the large Close Display Button that is displayed instead of the Playerbar Search Box,

They are used to allow the user to Cycle Backwards or Forwards through the Available TrackInfo Plugins and Visualizations.

### Assets

MC Name		Type	Purpose
PreviousDisplayButton	Optional	Multi-State Image	Supplies the image used for the PreviousDisplayButton
NextDisplayButton	Optional	Multi-State Image	Supplies the image used for the NextDisplayButton

The instruction used take this form

```
<Entry Name=" MC Name " Bitmap="Image name" NumberImages="3" Scale="3.5"/>
```

Where:-

MC Button Name is the appropriate MC Button name from the Assets Panel

Image name is the name of the Multi-State Image to be used

and the rest of the statement are any additional instructions used to control the drawing of that image.

## <TREE>

The Tree is the tall narrow panel to the Left of the Vertical Splitter Bar in Standard View

### Assets

MC Name		Type	Purpose
Background	Optional	Image	Image to provide the Tree Background See note 1
Colors	Mandatory	<Colors ... />	See next panel
Selection	Optional	Image	A transparent/glassy image that sits over Tree Entries when a mouse hovers over them See note 2
ExpandButton	Optional	Image with 2 sub-images/sprites/tiles	The 1 <sup>st</sup> sub-image acts as the Expand Button whilst the 2 <sup>nd</sup> one acts as the Contract Button See note 3

1. The Background image is supplied using this format

```
<Entry Name="Background" Bitmap="Image name" Margins="4,2,2,2"  
DrawMode="REGION_E_HSTRETCH,REGION_E_VTILE"/>
```

Where Image name is the name of the Image to be used and the rest of the statement are any additional instructions used to control the drawing of that image.

### Colors used on the Tree

MC Color Name	Alternative Name	Purpose
Back	BackColor	Background color for the tree if an image is not used
Text	TextColor	Text color for every option/entry in the tree
OverText	OverTextColor	Color applied to the text when the mouse hovers over it
SelectedBack	SelectedBackColor	Background color for the selected option/entry in the tree it is also used as the background of an Active/Available option/entry when the mouse hovers over it
SelectedText	SelectedTextColor	Text color for the selected option/entry in the tree

2. The optional Selection image is supplied using this format style

```
<Entry Name="Selection" Bitmap="Image name" NumberImages="2" Columns="4,?-Flex,4"  
Rows="4,?-Flex,4,?-Flex,4"/>
```

Where Image name is the name of the Image to be used and the rest of the statement are any additional instructions used to control the drawing of that image.

3. The optional ExpandButton image is supplied using this format style  
<Entry Name="ExpandButton" Bitmap=" Image name" NumberImages="2"/>  
Where Image name is the name of the Image to be used.

## <ACTIONWINDOW>

This section controls the Action Window and Parts of the Windows that are opened when clicking on one of the entries in the Action Window.

### Assets

MC Name		Type	Purpose
Template	Mandatory		
Background	Mandatory		
Colors	Optional	<Colors ... />	See Colors panel
Tools	Optional	Multi-State Image	Supplies the image to use for the Tools/Menu button that is only displayed at the left-hand end of the Header Bar of the AW Instance used to host the Modern and Legacy Tag Windows
Close	Optional	Multi-State Image	Supplies the image to use for the close button that is displayed at the right-hand end of the Header Bar of every AW that is opened as a result of clicking on an entry in the Primary AW
Minimize	Optional	Multi-State Image	Supplies the image to use for the AW minimize (collapse) button displayed in every AW
Restore	Optional	Multi-State Image	Supplies the image to use for the AW restore (expand) button displayed in every AW

### Settings that can be used in the Template Instruction

MC Setting	Purpose
Margins="n,n,n,n"	Sets the distance (gaps), in pixels, between - The left side of the Tree and the left side of the AW, The top of the AW and the entry in the Tree immediately above it, The right side of the Tree and the right side of the AW, The bottom of the AW and the bottom of the Tree
InternalMargins="n,n,n,n"	Sets the distance (gaps), in pixels, between the left, top, right and bottom edges/sides of the AW and its content.
HeaderOnly="n"	
HeaderHeight="n"	Sets the height, in pixels, of the AW Header
BelowHeaderMargin="n"	Sets the distance, in pixels, between the bottom of the AW Header and the rest of the AW Content
VerticalSpacing="n"	Sets the distance (gap), in pixels, between the Primary Action Window and any Secondary AW that is opened as a consequence of clicking on one of the entries in the AW

### Colors used in the Action Window

MC Color Name	Alternative Name	Purpose
Back	BackColor	AW Background if a Background Image is not used Tag Label Background and Background for Tag Content that can't be edited/changed
Text	TextColor	Text color for the AW if a Background Image is not used Legacy Tag Labels and Content
ActiveText	ActiveTextColor	Mouse hover text color in legacy tag window
ListBack	ListBackColor	AW Background if an image is not used

ListText	ListTextColor	AW Text Color MUST be present if a Background Image is used
ListOverText	ListOverTextColor	AW Mouse Hover Text color if a Background Image is not used

**Settings that can be used in the Background Instruction**

<b>MC Setting</b>	<b>Purpose</b>
Margins="n,n,n,n"	
InternalMargins="n,n,n,n"	

## <TAGWINDOW>

This section controls the Advanced Tag Window ONLY!

### Assets

MC Name		Type	Purpose
Colors	Mandatory	<Colors ... />	See next panel
Group	Optional	Image with 2 sub-images/sprites/tiles	See note 1
FieldLabel	Optional	Image	
Field	Optional	Image with 2 sub-images/sprites/tiles	

### Colors used in the TagWindow

MC Color Name	Alternative Name		Purpose
Text		Mandatory	Supplies the color for the text on all Tag Labels and Tag Content
Group		Mandatory	Supplies the text color for the Group Labels, the headers of each user-defined tag group
Selection		Mandatory	Supplies the color applied to the text when the mouse hovers over it
EditOverlay		Optional	Used to color the Halo that appears around the Tag Name & Content when the tag is being edited It is more obvious/visible when a Large Value Tag, i.e. Lyrics, is selected The default color appears as several shades of Red

## <SPLITTER>

The Vertical and Horizontal Splitters act as separators between the various sections of the Main MC Window in Standard View.

The Vertical Splitter forms the movable boundary between the Tree and the Display Panel and Details List while the Horizontal Splitter forms the movable boundary between the Display Panel and the Details List

**NOTE** - The Colors used for the Arrows that appear on the Splitter are defined in the MAINFRAME section

If you have no image file(s), or do not wish to create them, you can use the Face3D Color setting in the MAINFRAME section to color them.

### Assets

MC Name		Type	Purpose
Data	Optional	<Data DefaultSize="n"/>	Sets the Width of the Vertical Splitter and Height of the Horizontal Splitter in pixels. Set the size to 1 to create a skin with no obviously visible splitter bars. If you delete the <Data DefaultSize="n"/> line, the splitter width/height defaults to 8px regardless of actual image size. It also affects the size of the Arrows on the Splitters
VerticalBackground	Optional	Image	Used to provide a background color/texture for the Vertical Splitter. Is only required if the Face3D Color setting in the MAINFRAME is not set Or If you want to use different background color/texture on the Vertical and Horizontal Splitters
VerticalHandle	Optional	Image	Supplies additional decoration between the Arrows on the Vertical Splitter to help make it more visible
HorizontalBackground	Optional	Image	Supplies a background color for the Horizontal Splitter. Is only required if the Face3D Color setting in the MAINFRAME is not set Or If you want to use different background colors/textures on the Vertical and Horizontal Splitters
HorizontalHandle	Optional	Image	Used to provide additional decoration between the Arrows on the Horizontal Splitter to help make it more visible

**NOTE** - MC will automatically adjust the sizes of the images provided to match those of the relevant splitter.

**N.B.** Any VerticalHandle and/or HorizontalHandle images you provide have to be the same size as the relevant Background Image and should only have Texture that is different to that of the Background image in their central areas.

## <VIEWHEADER>

The ViewHeader is where you'll find the split view tabs, links etc. It also affects the header you'll get when you double click an album in album views.

**NOTE:** You can NOT use an Image to provide the background in the ViewHeader section.

The only acceptable instruction in the ViewHeader Section is a single <Colors .... /> one containing one or more of the following:

### Colors used on the ViewHeader

MC Color Name	Alternative Name	Purpose
HeaderBackground	HeaderBackgroundColor	Background color for the Active TabBar A Faded version of the color is used as the background of TabBars in Split Views if an InactiveHeaderBackground color is not used. It is also used as the Background color for the header bar displayed when Double Clicking on an album in Album View
InactiveHeaderBackground	InactiveHeaderBackgroundColor	Background colour for Inactive TabBars in Split Views
ActiveTabText	ActiveTabTextColor	Text colour of the current tab on ALL TabBars and All TabBar Links
HeaderHotText	HeaderHotTextColor HotText	Text colour on mouse hover over the Only Tab on a TabBar OR when hovering over TabBar Links.
Background		Background color for the upper panel in Playlist Forms/Windows
Text		Background color for the Text in the upper panel in Playlist Forms/Windows It is also used as the Text color for the header bar displayed when Double Clicking on an album in Album View Is used as the Text color for All text on All TabBars if a HeaderHotText color and an ActiveTabText color are not provided.

**Caution** – Using the Text setting to color anything other than the Text in the upper panel in Playlist Forms/Windows should be avoided because unlike the ActiveTabText and HeaderHotText settings it can also color the 2/3 icons normally found at the left-hand end of every TabBar!!

## <SPLITVIEWTAB>

The Tabs and Optional Close Buttons that appear on the ViewHeader

### Assets

MC Name		Type	Purpose
Data	Mandatory	<Data ... />	See next panel
Button	Mandatory	Multi-State Image	Supplies the image for the Tabs See note 1
BorderRow	Optional	Multi-State Image	Supplies the image for the TabBar bottom Border See note 2
Close	Optional	Multi-State Image	Supplies the image for the Tab Close Button

### Settings that a in the Data instruction:-

MC Setting	Purpose
InternalMarginLeft="n"	Sets the distance, in pixels, from the left edge of the tab to the start of the text on it
InternalMarginRight="n"	Sets the distance, in pixels, from the end of the text and the right edge of the tab
OverlapLeft="n"	Sets the distance, in pixels, by which the 1 <sup>st</sup> tab can overlap the last icon on the TabBar
OverlapRight="n"	Sets the distance, in pixels, by which each tab can overlap the previous one
ImageToTextSpacing="n"	Sets the distance, in pixels, between the end of the text and any image that may appear on the tab
FirstItemTextOffset="n"	Gets added to the InternalMarginLeft for the 1 <sup>st</sup> Tab to increase the distance between the edge of the tab and the start of the text
MakeFinalItemSameWidth="n"	

## <CARRADIO>

This section controls the look/feel of the 3-State Car Radio Buttons that appear between the bottom of the Display Panel/Window and the Horizontal Splitter when the Car Radio Feature is Enabled.

Although MC allows 6, 8, 10, 12, 14, 16 or 18 buttons to be shown you only have to provide the image and colors information for 3 of them in this format.

```
<Entry Name="ButtonLeft" Bitmap="Image name" NumberImages="3" Margins="3,3,2,0"
OverText="nnnnnn" PressedText="nnnnnn"/>
<Entry Name="ButtonMiddle" Bitmap="Image name" NumberImages="3" Margins="3,3,2,0"
OverText="nnnnnn" PressedText="nnnnnn"/>
<Entry Name="ButtonRight" Bitmap="image name" NumberImages="3" Margins="3,3,2,0"
OverText="nnnnnn" PressedText="nnnnnn"/>
```

Where:

ButtonLeft is the 1st button, ButtonRight is the last button and ButtonMiddle is all the other buttons.

OverText (or OverTextColor) is used to provide the hex value of the color to be used for the text on the relevant button when the mouse hovers over it.

PressedText (or PressedTextColor) is used to provide the hex value of the color to be used for the text on the relevant button when that button is pressed/clicked.

**NOTE** - Rather oddly PressedText only gets shown momentarily when the user presses on the button even though the button retains its Pressed State

## <LIST>

This section handles the Details List that is placed Below the Horizontal Splitter Bar in Playing Now and some aspects of the Display Panel in Standard and Cover Views.

**NOTE:** You can NOT use an Image to provide the background in the List Section.

### Assets

MC Name		Type	Purpose
Colors	Mandatory	<Colors ... />	See next panel
HeaderBar	Optional	Image	Supplies the image and optional text color to be used for the List Header Bar
Selection	Optional	Image	A transparent/glassy image that sits over a list entry when the mouse hovers over it See note 3
Lasso	Optional		
LinkButton	Optional		
Rating	Mandatory		
RatingClear	Optional		

Colour settings for **ALL** views with a detail list: Playing Now, all Tab Views (built-in or custom); DSP drop-downs, Search Box drop-downs, various lists within Options, etc.

### Colors used in the List

MC Color Name	Alternative Name		Purpose
Back	BackColor	Mandatory	Background color for every entry in the list Or Background color for every even-numbered line in the list if a color is provided for GridLineBack
Text	TextColor	Mandatory	Text color for every entry in the list
Current	CurrentColor HotText HotTextColor	Optional	Color applied to the text of the currently playing item when a non-playing track is selected. Useful when scrolling long lists to see which track is currently playing when another track is selected.
SelectedBack	SelectedBackColor	Optional	Background color for the selected option/entry in the list A very faded version is used as the background of an active/available option/entry when the mouse hovers over it
SelectedText	SelectedTextColor	Optional	Text color for the selected entry in the list
GridLine		Optional	Color of thin Vertical lines separating columns in detail lists (Show Gridlines must be enabled in Options > Tree and View > List > Show Gridlines)

GridLineBack		Optional	Background color for every odd-numbered line in the list
OverText		Optional	Mouse hover text color

## <SCROLLBAR>

This section is used to specify the images to be used with the various parts of the Horizontal and Vertical Scrollbars

### Assets

MC Name		Type	Purpose
HorizontalLeftArrow	Mandatory	Multi-State Image	Supplies the images for the Arrow at the left-hand end of the Horizontal Scrollbar.
HorizontalBackground	Mandatory	Multi-State Image	Supplies the images for the Horizontal Scrollbar Track.
HorizontalHandle	Mandatory	Multi-State Image	Supplies the images for the Horizontal Scrollbar Handle/Thumb.
HorizontalRightArrow	Mandatory	Multi-State Image	Supplies the images for the Arrow at the right-hand end of the Horizontal Scrollbar.
VerticalUpArrow	Mandatory	Multi-State Image	Supplies the images for the Arrow at the top of the Vertical Scrollbar.
VerticalBackground	Mandatory	Multi-State Image	Supplies the images for the Vertical Scrollbar Track.
VerticalHandle	Mandatory	Multi-State Image	Supplies the images for the Vertical Scrollbar Handle/Thumb.
VerticalDownArrow	Mandatory	Multi-State Image	Supplies the images for the Arrow at the bottom of the Vertical Scrollbar.
Stump	Mandatory	Image	The Scrollbar corner/endpoint: without this image a different background-color may appear in its place

**NOTE:** Although the Majority of MC Skins use 4-state images for the Scrollbar Arrows, Background and Handles The 4th state (Disabled) never gets shown, for any of them, anywhere in MC.