

SSX Skin Format

Version 1.0

Introduction

The SSX skin format is the file format that forms the basis for a skinning engine built into a media player application known as the Saturn Project. Please read the following Notes section as it carefully explains how the skin format works and provides information on the overall format of the file

Notes

1. The file **must** have the extension “.ssx” otherwise it will not be read.
2. The name of the SSX file must be in lower-case, and must be named either **main.ssx**, **wshade.ssx**, **playlist.ssx**, or **video.ssx**, otherwise the application will assume the skin file is missing and will abort the loading of the skin.
3. If the first line of the skin file header does not read as “SSX” then the application will assume the skin file is invalid and will abort loading the skin.
4. Every entry in the skin file is mandatory, meaning that the file header and footer must exist along with fully filled in element data. If the skinner decides that they wish to enter a null value then they must not leave the line blank, but instead put a zero on the line to indicate the value is either null or set to default.
5. The skinner must take responsibility for the creation of the skin file because, by design, the skinning engine does not correct entries in the skin file. This is to give the skinner maximum flexibility and control over the skinning process while avoiding hampering their efforts with the error checking process.
6. Any line that starts with a '#' (hash symbol) will be treated as a comment line and ignored by the skin parser. Comments must be on a single line by themselves.

7. Co-ordinates should be specified as the Top-Left and Bottom-Right co-ordinates of the picture in the picture file.
8. All co-ordinates that are specified must be comma delimited, and all sets of co-ordinates must be separated by a space,

e.g. 123,45 678,90

Both sets of co-ords has a comma between the X and Y co-ords, and both sets of co-ords must have a space between them for separation. If the co-ords are not specified like this then the application will read them wrong and the skin will turn out invalid. There is also the possibility that the application will encounter a serious or critical error and may crash.

9. Any problems or errors in the skin file or the skinning engine are logged to a log file called "skin_error.log".

Graphical Skin Files

All graphical files in an SSX skin directory must be in bitmap format. These files must have specific file names and contain specific content. These files are listed below:

- **main.bmp** – This file must contain the background bitmap for the main media player area.
- **wshade.bmp** – This file must contain the background bitmap for the windowshade mode.
- **video.bmp** – This file must contain the background bitmap for the video window.
- **playlist.bmp** – This file must contain the background bitmap for the playlist window.
- **text.bmp** – This file must contain the text used in displaying the file name of the current media playing. An accompanying file called **text.txt** (explained in the next section) should also be provided to provide the locations of each character within the picture file, allowing the skinner to create a skin with a customised font size.

Textual Skin Files

All text files in an SSX skin directory must be in text format (Win32 text format). While the content of these text files is flexible, the content must be file specific. These files are listed below:

- **main.ssx** – This file is the skin file that contains the skin data for the main application interface. The file syntax for the SSX format is described in the

following sections. If this skin file doesn't exist then the skin will not be loaded.

- **wshade.ssx** – This file is the alternative skin file for when the application is sent into windowshade mode, a very minimalist state where only a few basic buttons and controls are visible. If this skin file doesn't exist then windowshade mode will not be activated.
- **playlist.ssx** – The playlist skin file specifies the design data for the video window. If this skin file doesn't exist then the playlist window will not be loaded.
- **video.ssx** – The video skin file specifies the design data for the video window. If this skin file doesn't exist then the video window will not be loaded.
- **text.txt** – This file provides the locations of every character used in the character set used to display media file titles and such in the application. An example of the file is shown below:

```
# start of the file
6
a 1,1 10,11
b 11,1 21,11
c 22,1 32,11
A 33,1 43,11
B 44,1 55,11
C 56,1 66,11
# end of the file
```

The first line of the file should contain the number of characters in the character set, this number is also the same number of lines in the file. The file should be constructed so that each line has one letter with the accompanying top-left & bottom-right co-ordinates of the characters pixel area, each entry on the line should be separated by a space. Any line that starts with a # (hash) will be treated as a comment and ignored. A space character (" ") should be represented by the upper-case letters "SPC".

Element List

The Element List is a list of all the valid elements that can be used to construct a skin using the SSX format. Element names must follow a specific naming convention (shown in the Key below); the letter must be a capital letter and must be followed by two digits. Numbers less than ten must start with a zero e.g. 01, 02, 03.

Key:

Ixx – Item element

Dxx – Display element

Cxx – Control element

ITEM Elements

- I01 - Play/Pause
- I02 - Stop
- I03 - Previous Track (reverse)
- I04 - Next Track (forward)
- I05 - Eject (open file[s])
- I06 - Repeat
- I07 - Shuffle
- I08 - Volume Mute
- I09 - Playlist window
- I10 - Playlist menu button
- I11 - Video window
- I12 - Video window menu button
- I13 - Options window
- I14 - File Information
- I15 - Power (Close application)
- I16 - Minimise (hide application)
- I17 - Windowshade mode (minimise)
- I18 - Windowshade mode (maximise)

DISP Elements

- D01 - Scrolling text area
- D02 - Playback indicator
- D03 - Stereo/Mono channel indicator
- D04 - Kbps indicator
- D05 - KHz indicator
- D06 - Current time of playback
- D07 - Total time of playback
- D08 - Shuffle indicator
- D09 - Repeat indicator

CTRL Elements

- C01 - Volume scrollbar
- C02 - Media Position scrollbar
- C03 - Volume balance scrollbar

File Header

<<descriptor>>

<<version>>

<<skin name>>

<<authors name>>

<<creation date>>

<<copyright>>

<<comments>>

<<skin data>>

- 1 <<descriptor>>
 File format descriptor (must be "SSX")
- 2 <<version>>
 SSX version format (default to "1")
- 3 <<skin name>>
 Skin Name (a zero will default to "unknown")
- 4 <<authors name>>
 Authors Name (a zero will default to "unknown")
- 5 <<creation date>>
 Creation Date (a zero will default to "unknown")
- 6 <<copyright>>
 Copyright information (a zero will default to "unknown copyright")
- 7 <<comments>>
 Comments about skin (optional, zero leaves the field blank)
- 8 <<skin data>>
 Important information about the skin e.g. height, width, bitmap cutting
 (separated by a space)

File Footer

<<EOF>>

- 1 EOF
 Indicates the very end of the skin file

Item Element

Item elements are essentially picture buttons on the skin. While items are not actually buttons, they're functionality gives the appearance that they are. They are simply bitmaps being manipulated by user interaction.

```
ITEM
<<item name>>
<<area co-ordinates>>
<<graphic file name>>
<<picture co-ords – normal state>>
<<picture co-ords – mouse over>>
<<picture co-ords – mouse click>>
<<click region co-ords>>
<<bitmap cutting>>
<<tooltip>>
<<shortcut key>>
END
```

- 1 ITEM
 Start of item data.
- 2 <<item name>>
 The element name of the item, see Element List for more information.
- 3 <<area co-ordinates>>
 X/Y co-ords for the top-left & bottom right corners of the element.
- 4 <<graphic file name>>
 File name of the target picture file containing the item graphic.
- 5 <<picture co-ords – normal state>>
 Top-left & Bottom-right co-ords of the item in its normal state.
- 6 <<picture co-ords – mouse over>>
 Top-left & Bottom-right co-ords of the item in its mouse-over state. A zero will default the co-ords to the **normal state** co-ords.
- 7 <<picture co-ords – mouse click>>
 Top-left & Bottom-right co-ords of the item in its mouse-click state. A zero will default the co-ords to the **normal state** co-ords.
- 8 <<click region co-ords>>
 Co-ordinates of the region within the user can click in to activate the item. Max of 12 points, zero defaults to X/Y co-ords of item area
- 9 <<bitmap cutting>>
 Whether bitmap requires a select region to be 'cut out' from it, must be either "true" or "false".

- 10 <<tooltip>>
Tooltip text to show when mouse hovers over button.
- 11 <<shortcut key>>
ASCII value of the shortcut key binding used to select the item.
- 12 END
end of item data.

Control Element

Control elements are basically sliders (e.g. Volume slider, media position slider), where a graphic (the control-bar) can move freely from points A to B on top of a base graphic (the control graphic).

```

CTRL
<<control name>>
<<area co-ords>>
<<co-ord position bounds>>
<<control picture file>>
<<num of control animation frames>>
<<co-ords of first control picture>>
<<control bar picture file>>
<<area co-ords of control-bar>>
<<pic co-ords – normal state>>
<<pic co-ords – alternative state>>
<<click region co-ords>>
<<bitmap cutting>>
<<tooltip>>
<<shortcut keys>>
END

```

- 1 CTRL
Start of control data.
- 2 <<control name>>
The element name of the item, see Element List for more information.
- 3 <<area co-ords>>
Top-left & bottom-right co-ords of the whole control area.
- 4 <<co-ord position bounds>>
Movement bounds of the control bar, X/Y co-ords of the start / end locations.
- 5 <<graphic file name – control>>
File name of the target picture file containing the item graphic.
- 6 <<num of control animation frames>>
Number of frames in the control bar animation, each frame must be

tilled vertically (i.e. all frames go from top to bottom of graphic file).

- 7 <<co-ords of first control picture>>
Co-ords of the first element picture in file.
- 8 <<control bar picture file>>
File name of the bar picture file.
- 9 <<area co-ords of control-bar>>
Top-left & bottom-right co-ords of the control-bar area.
- 10 <<pic co-ords – normal state>>
Top-left & Bottom-right co-ords of the item in its normal state.
- 11 <<pic co-ords – alternative state>>
Top-left & Bottom-right co-ords of the item in its alternative state. A zero will default the co-ords to the **normal state** co-ords.
- 12 <<click region co-ords>>
Co-ordinates of the region within the user can click in to activate the item. Max of 12 points, zero defaults to X/Y co-ords of item area.
- 13 <<bitmap cutting>>
Whether bitmap requires a select region to be 'cut out' from it, must be either "true" or "false".
- 14 <<tooltip>>
Tooltip text to show when mouse hovers over button.
- 15 <<shortcut keys>>
ASCII value of the shortcut key binding used to select the item.
- 16 END
End of item data.

Display Element

Display elements are areas that are used by the application to display information to the user; they are designated areas for displaying data.

```
DISP
<<display name>>
<<region co-ords>>
<<display picture file>>
<<coords of display pic in file>>
END
```

- 1 DISP
Start of display data.

- 2 <<display name>>
 The element name of the item, see Element List for more information
- 3 <<region co-ordinates>>
 X/Y co-ords for the top-left & bottom right corners of the element
- 4 <<display picture file>>
 File name of the bar picture file
- 5 <<coords of display pic in file>>
 Top-left & bottom-right co-ords of display area in file
- 6 END
 End of display data.

List Element

Display elements are areas that are used by the application to display information to the user; they are designated areas for displaying data.

LIST

<<region co-ords>>

<<foreground colour>>

<<background colour>>

END

- 1 LIST
 Start of list data.
- 2 <<region co-ords>>
 X/Y co-ords for the top-left & bottom right corners of the element.
- 3 <<foreground colour>>
 HTML colour code of the foreground colour, the colour code must **not** start with a # (hash).
- 4 <<background colour>>
 HTML colour code of the background colour, the colour code must **not** start with a # (hash).
- 5 END
 End of list data.

Example File

The following example was created merely to provide a demonstration of the structure of an SSX file, and should only be used as reference material.

```
# Below is the skin header containing important skin details
SSX
1
Satanica
Jolly Green Giant
13 Jan 04
Copyright JGJ Enterprises 2004
This is my first skin, so don't criticise it too harshly please
150 400 false
# The ITEM example below shows how to create a skin item
ITEM
I01
123,34 127,54
buttons.bmp
0,0 5,5
0,6 11,5
0,12 17,5
0
false
This is the play button
p
END
# The CTRL example below shows how to create a skin control
CTRL
C01
123,34 567,89
54,55 105,55
control.bmp
0,0 25,10
10
control_bar.bmp
0,0 5,5
6,0 11,5
6,0 11,5
0
false
This is the volume slider
32 34
END
# The DISP example below shows how to create a skin display
DISP
D01
100,45 150,55
text_display.bmp
0,0 150,10
END
LIST
```

10,10 60,60

AE4F04

FF676A

END

EOF

#This is the end of the file