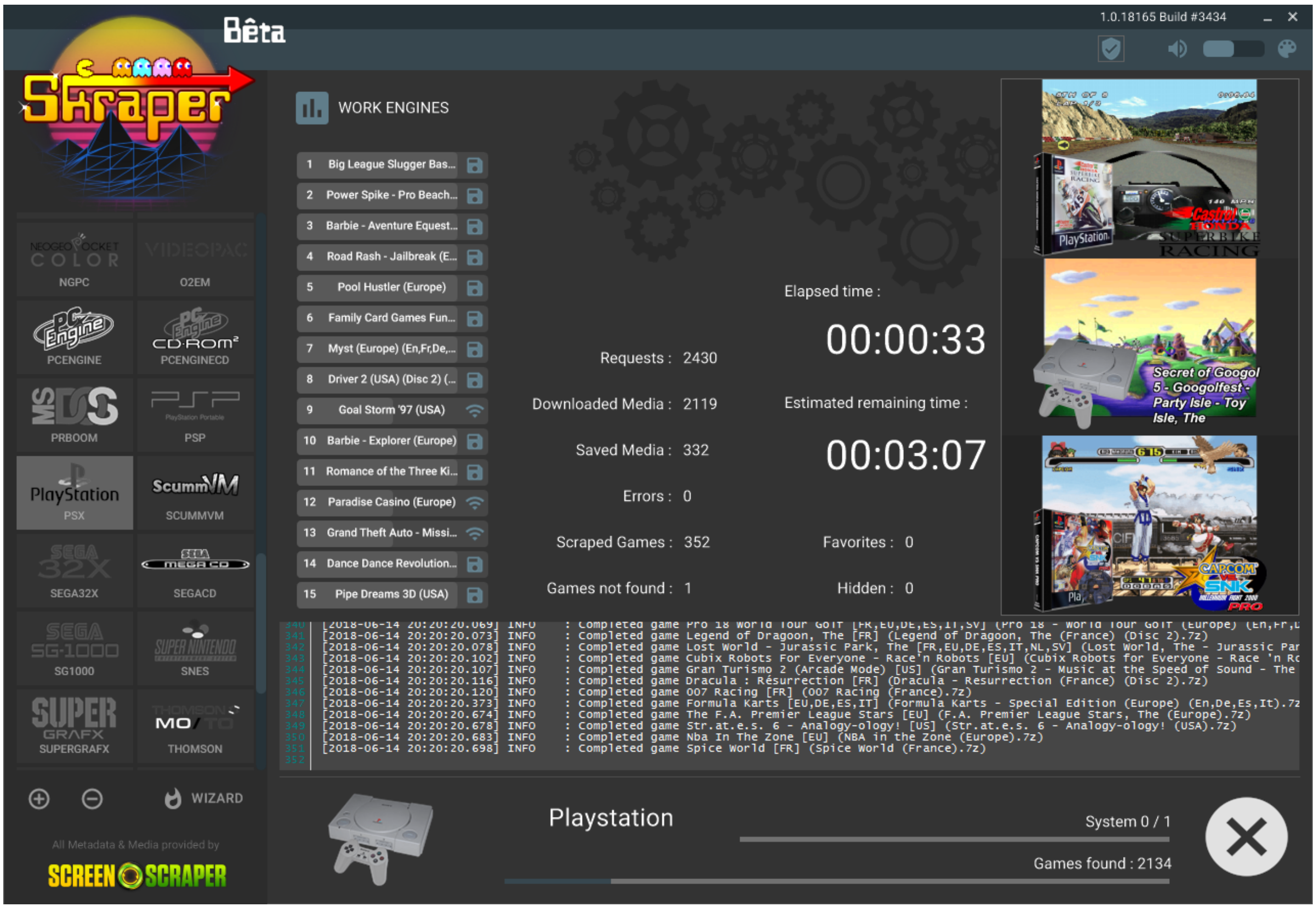
Skraper « User-mix » documentation

Target: SkraperUI Beta 7



Bêta
1.0.18165 Build #3434
NGPC
O2EM
PCENGINE
PCENGINECD
PRBOOM
PSP
PSX
SCUMMVM
SEGA32X
SEGACD
SG1000
SNES
SUPERGRAFX
THOMSON
WIZARD
All Metadata & Media provided by
WORK ENGINES
1 Big League Slugger Bas...
2 Power Spike - Pro Beach...
3 Barbie - Aventure Equest...
4 Road Rash - Jailbreak (E...
5 Pool Hustler (Europe)
6 Family Card Games Fun...
7 Myst (Europe) (En,Fr,De,...
8 Driver 2 (USA) (Disc 2) (...
9 Goal Storm '97 (USA)
10 Barbie - Explorer (Europe)
11 Romance of the Three Ki...
12 Paradise Casino (Europe)
13 Grand Theft Auto - Missi...
14 Dance Dance Revolution...
15 Pipe Dreams 3D (USA)

Elapsed time : 00:00:33

Estimated remaining time : 00:03:07

Requests : 2430

Downloaded Media : 2119

Saved Media : 332

Errors : 0

Scraped Games : 352

Games not found : 1

Favorites : 0

Hidden : 0

```
[2018-06-14 20:20:20.069] INFO    : Completed game Pro 18 World Tour Golf [FR,EU,DE,ES,IT,SV] (Pro 18 - World Tour Golf (Europe) (En,Fr,D
[2018-06-14 20:20:20.073] INFO    : Completed game Legend of Dragoon, The [FR] (Legend of Dragoon, The (France) (Disc 2).7z)
[2018-06-14 20:20:20.078] INFO    : Completed game Lost World - Jurassic Park, The [FR,EU,DE,ES,IT,NL,SV] (Lost World, The - Jurassic Par
[2018-06-14 20:20:20.102] INFO    : Completed game Cubix Robots For Everyone - Race'n Robots [EU] (Cubix Robots for Everyone - Race 'n Ro
[2018-06-14 20:20:20.107] INFO    : Completed game Gran Turismo 2 (Arcade Mode) [US] (Gran Turismo 2 - Music at the Speed of Sound - The
[2018-06-14 20:20:20.116] INFO    : Completed game Dracula : Résurrection [FR] (Dracula - Resurrection (France) (Disc 2).7z)
[2018-06-14 20:20:20.120] INFO    : Completed game 007 Racing [FR] (007 Racing (France).7z)
[2018-06-14 20:20:20.373] INFO    : Completed game Formula Karts [EU,DE,ES,IT] (Formula Karts - Special Edition (Europe) (En,De,Es,It).7z
[2018-06-14 20:20:20.674] INFO    : Completed game The F.A. Premier League Stars [EU] (F.A. Premier League Stars, The (Europe).7z)
[2018-06-14 20:20:20.678] INFO    : Completed game Str.at.e.s. 6 - Analogy-ology! [US] (Str.at.e.s. 6 - Analogy-ology! (USA).7z)
[2018-06-14 20:20:20.683] INFO    : Completed game Nba In The Zone [EU] (NBA in the Zone (Europe).7z)
[2018-06-14 20:20:20.698] INFO    : Completed game Spice World [FR] (Spice World (France).7z)
```

Playstation

System 0 / 1

Games found : 2134

Summary

Introduction

This document aim to describe the content of « User mix » XML files. Those files use the XML language to describe complex image compositions..

User mix files contain a list of object descriptors that Skraper can read and interpret to create rich image compositions, using texts, local and remote resources from http://www.screenscraper.fr.

- A Generic XML description is available here: https://fr.wikipedia.org/wiki/Extensible_Markup_Language

- A more technical documentation can be found here: http://pages.videotron.com/fyergeau/w3c/xml10/REC-xml-19980210.fr.html or https://www.w3.org/TR/xml/

Global Objects

To build an image composition, those 3 objects are required:

- **Information**: Contains names, description, author name and various textual information.
- **Viewport**: Describe the output image properties (width, height, background, format ...).
- **Drawings**: Provide a list (or array) of one or more graphic item.

Here is an example of the general structure:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ImageComposition xsi:noNamespaceSchemaLocation="https://www.skraper.net/ImageComposition.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Information ... information ... />
  <Viewport ... output image ... />
  <Drawings>
    <Item>
            ... graphic item #1 ...
    </Item>
    <Item>
            ... graphic item #2 ...
    </Item>
    <Item ..... />
            ... etc.

    <Item ..... />
    <Item>
            ... last graphic element ...
    </Item>
  </Drawings>
</ImageComposition>
```

Notes:

- XML comments are valid, using: **<!--** *...comment text on one or more lines...* **-->**
- « https://www.skraper.net/ImageComposition.xsd » can be used to validate your XML file. In some advanced text editors, the XSD Schema allows auto-completion.
- It is always a good idea to validate and format your XML files. Online validators and formatter are freely available here: https://www.freeformatter.com/xml-validator-xsd.html and https://jsonformatter.org/xml-formatter

Information Object

The information object should contain 4 text attributes:

- ShortName
- LongName
- Description
- Author

```
<Information ShortName="Mix4" LongName="4 Images Mix" Description="3DBox, Support, Screenshot, Wheel"
Author="Archangel54"/>
```

Currently, only the LongName attribute is used in Skraper. Anyway, it is recommended to provide all the 4 attributes. No need for more description, attribute are easy to understand.

Any omitted attribute will have the default value “Unknown”.

Viewport Object

The viewport object describe the output image properties, including the image format (only jpg/png supported).

```
<Viewport Color="#80FFFFFF" Width="1024" Height="768" ImageFormat="Png" CompressionPercent="0"/>
```

- **Width & Height** are the size of the output image, in pixels (1024 x 768 in the above example).
- **Color** is the background color. The whole image is cleared using this color, before any graphic element is drawn:
  - Color format is #AARRGGBB, where AA, RR, GG & BB are respectively Alpha, Red, Green, Blue, in hexadecimal notation, from 00 up to FF.
  - The Alpha component define the opacity of the color. 00 means the color if full transparent. FF means full opaque. Usually, #00000000 or #00FFFFFF are common full transparent background colors, but any value may be used.
  - For example: *Color="#80FF0000"* produces a 50% transparent Red color that will blend with the background the output image is drawn onto (most likely the front-end background)
- **ImageFormat** define the output file format. PNG and JPG are supported. It is recommended to use the lossless PNG format.
- **CompressionPercent** defines the JPG compression rate in percent. It may vary from 100% (best quality) to 1% (worst quality). It is not recommended to go under 80%.

Missing attributes are replaced with the following default values:

- Color="#00FFFFFF"
- Width="800"
- Height="600"
- OutputFormat="Png"
- CompressionPercent="100"

Note: The output image is called hereafter, the « Viewport ».

Drawings and Item objects

Here is the core drawing system, and things are becoming more complex.

The « Drawings » object is a container for a list of « Item ». Every single « Item » describe what resource and where and how it has to be drawn in the Viewport.
Items are drawn in the reading direction. That is, next item may partly or totally cover previous items.

A simple item contains:

```
<Drawings>
  <Item Type="Screenshot"></Item>
</Drawings>
```

- **Type** represent the resource to draw. It must be a valid string among the following::

  - "NoRessource"
  - "Screenshot"
  - "ScreenshotTitle"
  - "FanArt"
  - "Box2DFront"
  - "Box2DSide"
  - "Box2DBack"
  - "Box3D"
  - "Support"
  - "Wheel"
  - "WheelCarbon"
  - "WheelSteel"
  - "Classification"
  - "RecalBoxMixV1"
  - "RecalBoxMixV2"
  - "PictoPlayers"
  - "PictoNote"
  - "SupportTexture"
  - "BoxTexture"
  - "SteamGrid"
  - "ScreenMarquee"
  - "ScreenMarqueeSmall"
  - "Marquee"
  - "Include"
  - "LocalFile"
  - "Region1"
  - "Region2"
  - "SystemBezel43"
  - "SystemBezel169"
  - "SystemWheel"
  - "SystemCarbonWheel"
  - "SystemSteelWheel"
  - "SystemMonochromeLogo"
  - "SystemIcon"
  - "SystemIconMini"
  - "SystemPhoto"
  - "SystemPicture"
  - "SystemControllerPhoto"
  - "SystemScreenMarquee"
  - "SystemWallPaper"
  - "Text"

Most of them are remote resources downloaded from the ScreenScraper database, but the coloured ones are different:

- The "NoRessource" has a special meaning (see later).
- "Blue" resources are local resources.
- "Text" are… texts, either local or remote.
- "System…" are platform-wide remote resources.

For remote resources from the ScreenScraper database, there is not much to say. Types are self-describing.

Let's take a deeper look into some special resources:

- ***Region1*** and ***Region2*** are round region/country flags (128x128 pixels) corresponding to the current scraped game.
    - If the game has 2 or more regions, *Region1* and *Region2* draws the corresponding flags.
    - If the game has only one region, only *Region1* draws a flag.
    - If the game has no region, both *Region1* et *Region2* are empty.

  Keep in mind that *Region1* and *Region2* depends of what is actually stored in the ScreenScraper database, and not from regions in filenames (unless you checked the corresponding option in Skraper).

- ***LocalFile*** draws local images. Image files must be located in the XML folder. When using this resource type, the attribute ***LocalFilePath*** must contain the actual file path, relative to the XML file.

Example:

```
<Drawings>
  <Item Type="LocalFile" LocalFilePath="NoScreenshot.png"></Item>
</Drawings>
```

- ***Include*** allows to draw another mix in the current one. As for ***LocalFile*** you must specify the mix file using ***LocalFilePath.*** This is particularly useful when you have common parts in multiple mixes, or when you want to project (see later) complex images. The resulting image from the sub-mix can be rotated, projected and processed like any other resource.

Example:

```
<Drawings>
  <Item Type ="Include" LocalFilePath="ScreenShot.include.xml"></Item>
</Drawings>
```

- ***NoRessource*** draws nothing. This type is only useful to use the item as a conditional container for child items. See later how to use drawing conditions.

- ***Text*** draws static or dynamic texts from the current scraped game (game name, release date, players …). You must specify text and text formatting using the following attributes:

  - ***Text***="My Text", ***Text***="%variable%" or ***Text***="My Text1 %variable% Text": You can draw any text, containing variables or not. The font size will be automatically calculated so that the text takes the maximum space in the given drawing rectangle. New lines (carriage return / line feed) may be achieved by using the special variable '%%'. You may also use the following variable to draw current scraped game properties:
    - %name%
    - %description%
    - %publisher%
    - %developer%
    - %players%
    - %genre%
    - %region%
    - %releasedate%

  - ***TextColor*** defines the text color and works like the background color attribute in the viewport. To make the text more readable in all situations, it's outlined using the inverse color. For example, a white text will be outlined with a thin black line, so that it is readable on any dark or light background. Default color is black.

  - ***FontFamily*** select the font or font family. If the specified font is not available, the system automatically select the more appropriate one. Default font is Arial

  - ***FontStyle*** apply one or more font style to the text. Style must be a combination of the 4 following string, space separated: "Bold Italic Strikeout Underline". Default value is empty (no style)

  - ***TextAlign*** modify the text alignment, vertically and horizontally. This attribute uses the same value set than the « Anchor » attribute (see next page). Default value is "TopLeft".

Example:

```
<Drawings>
  <Item Type="Text" Text="%name%" TextColor="#FFFFFFFF" FontFamilly="Arial" FontStyle="Bold Italic" TextAlign="VCenterHCenter"></Item>
</Drawings>
```

Display and Rotation objects

These two objects may appear under every « Item ». They describe where and how the resource has to be drawn in the Viewport. Let's take a closer look to the "Display" object. Its main attribute is "Mode" which define how to draw the image:

- "blit": Standard mode. The image is drawn in a given rectangle.
- "distort": The image is distorted and projected in a 4-point polygon. The projection is automatically calculated to match the given polygon. This mode is useful to project an image as a texture on another object (the screen of an arcade cabinet for example!).

Standard mode "blit"

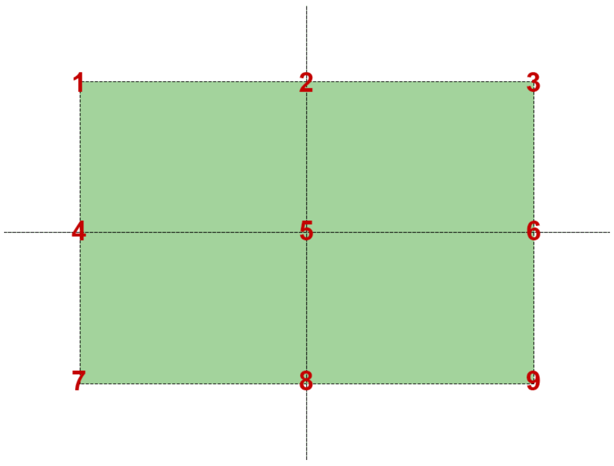
Here is a concrete sample:

```
<Drawings>
  <Item Type="Screenshot">
    <Display Mode="Blit" X="472" Y="277" Width="469" Height="430" Transparency="0.0"
Anchor="VCenterHCenter" KeepRatio="false" Antialiasing="None" />
    <Rotation XOffset="50%" YOffset="50%" Angle="0.0" />
  </Item>
</Drawings>
```

- **Anchor** represent the reference point from which the image is drawn, relative to the X,Y coordinates. By default, the reference point of an image is its top-left corner. That means the image is drawn to the bottom-right, starting from the top-left corner. You can whose between 9 predefined anchor points:

Here is the list of valid values:

- "TopLeft" *(1)*
- "TopHCenter" *(2)*
- "TopRight" *(3)*
- "VCenterLeft" *(4)*
- "VCenterHCenter" *(5)*
- "VCenterRight" *(6)*
- "BottomLeft" *(7)*
- "BottomHCenter" *(8)*
- "BottomRight" *(9)*



For example, to draw an image stuck in the bottom-right corner, use:

- X="100%"
- Y="100%"
- Anchor="BottomRight"

To center any image in the viewport, use:

- X="50%"
- Y="50%"
- Anchor="VCenterHCenter".

Note: It is not possible to specify the anchor point in pixel or percent, because this can be achieved by appropriate combinations of coordinates & anchors.

- **X**, **Y**, **Width** & **Height** position (X & Y) and resize (Width & Height) the image in the viewport. Values may be expressed in 3 different ways:
    - Direct numeric values ("25") are coordinates or sizes in pixels.
    - Percent values ("50%") are coordinates and size in percentage of the viewport size. Examples :
        - "X": "50%" means the image is drawn from half the width of the viewport.
        - "Width": "33%" means the image is resized to the tier of the viewport width.
    - Exclamation values ("100!") works like percent values, except they are relative to the opposite size. Examples :
        - X="100!" or Width="100!" means 100% of the Viewport height.
        - Y="50!" or Height="50!" means half the Viewport width.

      Exclamation values are useful when using rotations. For example, "Width": "100!" will set the image width equals to the viewport height. Rotate the image by 90° and the resulting image will perfectly fit the viewport height.

  Default values are “0” for X &Y, and “100%” for Width & Height.

- **KeepRatio** controls the way the image is resized, keeping its aspect ratio or not:
    - *"false"* : The image is strictly resized according to Width & Height parameters. The image aspect may be modified and the resulting image may appear stretched or crushed either horizontally or vertically..
    - *"true"* : The image is resized to fit the given rectangle while keeping its aspect ratio. That is, either the width or the height may be modified.

- **Antialiasing** set the anti-aliasing level applied while drawing the image. 3 different rendering are available:
    - *"None»:* No antialiasing at all. Pixels are beautiful pixels! Ideal for screenshots.
    - *"Medium»:* Middle antialiasing (bilinear). Give a smoothed aspect between pixels while keeping the pixels visible.
    - *"High»:* High quality antialiasing (bicubic). Ideal for non-rectangular shapes, like 3D boxes, logos, …

  Default value is “High”.

- **Transparency** is a floating value representing the global transparency of the image, between 0.0 (full opaque) to 1.0 (full transparent). Keep in mind it is not an “Alpha” value, and it is independent of the Apha channel of the image. Default value is 0.0.

- **Effect** apply a color filter to the whole image. Possible values:
    - *"None"*: No filtering, default value.
    - *"BlackAndWhite"* : Each pixel is converted into its grey luminosity level.
    - *"Sepia"*: Well-known Sepia filter, giving a vintage photo aspect to the image.
    - *"GreenGameboy"*: Give a yellow-green aspect to the image, close to the screen aspect of the very first Nintendo Gameboy console.

- **XOffset**, **XOffset** & **Angle** rotate the image:
    - **Angle** is a floating point value representing the rotation angle in degree, from 0.0° to 360.0°. Default value is “0.0”.
    - **XOffset** and **XOffset** specify the rotation point relative to the size of the image. Value can be direct pixel values or percent values (relative to the image size, NOT the viewport size). Default values are “0”/”0”. Example: **XOffset=”50% YOffset=”50%” Angle=”90”** rotates the image by 90° around its center.

Summary of default values:

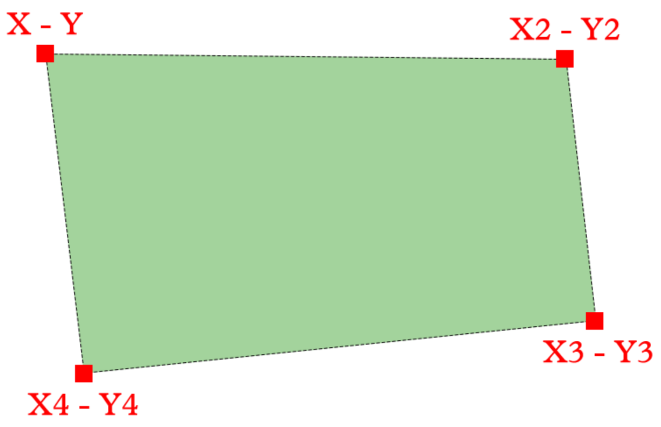
- Mode="Blit"
- Type="NoResource"
- LocalFilePath= *empty*
- Anchor="TopLeft"
- X="0"
- Y="0"
- Width="100%"
- Height="100%"
- KeepRatio="true"
- Antialiasing="High"
- Transparency="0.0"
- Effect="None"
- XOffset="50%"
- YOffset="50%"
- Angle="0.0"

Projection mode

Example:

```
<Drawings>
  <Item Type="Screenshot">
    <Display Mode="Distord" X="79" Y="122" X2="219" Y2="161" X3="249" Y3="342" X4="106" Y4="362"
Antialiasing="None" Transparency="0.0" />
  </Item>
</Drawings>
```

- **X**, **Y**, **X2, Y2, X3, Y3, X4, Y4** define the 4 corners of the image as a free 4-points clock-wise polygon. An anti-clockwise polygon will flip the image around the X-Y/X3-Y3 axis. Shifting the coordinates (starting from X2/Y2 for example) rotates the image also. The image is projected in the given polygon according to segment length (largest segments appear closest to human eyes, shortest segments appear farthest). The polygon must be concave. That means every coordinate must NOT enter the triangle composed by the 3 other coordinates, and segments must NOT cross.



- **Antialiasing** set the anti-aliasing level applied while drawing the image. It works like in mode=”blit”, except that different algorithm are used:
    - *"None"*: No antialiasing.
    - *"Medium"*: Medium antialiasing (FSAAx2).
    - *"High"*: High antialiasing (FSAAx4).

- **Transparency** (see “blit” mode)

- **Effect** (see “blit” mode)

Notes:

- Image distortion is a CPU-intensive algorithm and is affected by the FSAA multiplier. When using the “High” value, an intermediate image is created, 16 times bigger than the original!
- Rotation may be applied to a projected image, but it is recommended to pre-rotate the 4 points polygon instead.

Fallback object

Every “Item” may contains a “Fallback” object. This object is itself an “Item” and may contains also a “Fallback” object, allowing fallback chain. A fallback object is “executed” when the parent resource is not available:

```
<Item Type="Screenshot">
  <Display X="50%" Y="0%" Width="90%" Height="90%" Anchor="TopHCenter" Antialiasing="None" />
  <Rotation XOffset="50%" YOffset="50%" Angle="20.0" />
  <Fallback Type="ScreenshotTitle">
    <Display X="50%" Y="0%" Width="90%" Height="90%" Anchor="VCenterHCenter" Antialiasing="None" />
    <Rotation XOffset="50%" YOffset="50%" Angle="20.0" />
    <Fallback Type="LocalFile" LocalFilePath="NoScreenshot.png">
      <Display X="50%" Y="0%" Width="90%" Height="90%" Anchor="VCenterHCenter" Antialiasing="None"/>
      <Rotation XOffset="50%" YOffset="50%" Angle="20.0" />
    </Fallback>
  </Fallback>
</Item>
```

The “item” object try to get the current scraped game’s screenshot from the ScreenScraper database. If not available, its fallback object try to get the title screenshot. If not available, the next fallback draw a local file, always available.

If a fallback does not define its sub-objects “Display” or “Rotation”, they are inherited from the parent “Item”. That is, the example above can be simplified:

```
<Item Type="Screenshot">
  <Display X="50%" Y="0%" Width="90%" Height="90%" Anchor="VCenterHCenter" Antialiasing="TopHCenter"/>
  <Rotation XOffset="50%" YOffset="50%" Angle="20.0" />
  <Fallback Type="ScreenshotTitle">
    <Fallback Type="LocalFile" LocalFilePath="NoScreenshot.png"/>
  </Fallback>
</Item>
```

Both fallback objects inherit the “Display” and “Rotation” sub-objects from the initial “Item”.

Another example:

```
<Item Type="Screenshot">
  <Display X="50%" Y="0%" Width="90%" Height="90%" Anchor="TopRight" Antialiasing="TopHCenter" />
  <Rotation XOffset="0%" YOffset="0%" Angle="20.0" />
  <Fallback Type="ScreenshotTitle">
    <Fallback Type="LocalFile" LocalFilePath="NoScreenshot.png">
      <Rotation Angle="45.0" />
    </Fallback>
  </Fallback>
</Item>
```

Again, both fallback objects inherit the “Display” sub-object. The first fallback object also inherit the “Rotation” sub-object. However the deeper falback object redefine the whole “Rotation”. That means XOffset and YOffset are not inherited and take their default values (both “50%”). The rotation angle is defined to 45°.

System/Orientation filtering

“Item” objects can be filtered by system or by orientation.

By orientation means the “Item” is drawn only if the image is oriented in portrait or landscape.

The System filtering is a bit more complex. Basically it allows to draw items only if the current scraped system (or platform) is, or is not, in a given list. That is, the output image may look different, depending of the scraped system. System identifiers must be ScreenScraper’s numeric identifiers: MAME = 75, Megadrive = 1, Amstrad CPC = 65 …

Unfortunately, you cannot get easy access to the complete system list. You can use the list in the Annex #1, or use the API sample to get all systems.

Example :

```
<Item Type="Screenshot" IfSystem="IsNotInTheList:9,52" IfOrientation="IsLandscape" />
```

You can add an orientation filter, a system filter, or both on a single “Item”, using “IfSystem” and/or “IfOrientation” attributes.

- **IfSystem** specify the filter type (inclusion or exclusion) and the list of systems. 5 possibles values :
  - *"IsInTheList:IdSystem#1,…,IdSystem#X"*: The “Item” is drawn only if the scraped system is in the given list.
  - *"IsNotInTheList:IdSystem#1,…,IdSystem#X"*: The “Item” is NOT drawn if the scraped system is in the given list.
  - *"IsInTheListOrAncestors:IdSystem#1,…,IdSystem#X"*: The “Item” is drawn only if the scraped system is in the given list or in any child-system. (See Annex #2).
  - *"IsNotIntheListOrAncestors:IdSystem#1,…,IdSystem#X"*: The “Item” is NOT drawn if the scraped system is in the given list or in any child-system. (See Annex #2).
  - *"DoNotCare"*: No filtering (default value).

- **IfOrientation** specify the orientation the image must match to be drawn. 3 possible values:
  - *"IsPortrait"*: The image height is higher than the width (Ex: 200x500 pixels).
  - *"IfLandscape"*: The image width is higher than or equal to the height (Ex: 500x200 or 500x500 pixels).
  - *"IsWhatever"*: No filtering (default value).

Notes:

- Defined filters are obviously applied to the fallback chains.

Example #1:

```
<Item Type="Wheel" IfSystem="IsInTheListOrAncestors:1,2">
  <Display X="100%" Y="100%" Width="50%" Height="33%" KeepRatio="true" Anchor="BottomRight"
Antialiasing="High" Transparency="0.0" />
  <Rotation XOffset="50%" YOffset="50%" Angle="0.0" />
</Item>
```

The “wheel” is drawn only if the current system is:

- 1 - Megadrive (and it’s children: Megadrive 32X, Mega-CD, Megadrive - Sonic The Hedgehog 2 Hacks)
- 2 - Master-system (this system has no child)

Example #2:

```
<Item Type="Wheel" IfOrientation="IsLandscape">
  <Display X="200" Y="25" Width="50%" Height="50%" Anchor="BottomRight" Antialiasing="High" />
</Item>
```

This “3D Box” is drawn only if the image is of type “Portrait” (vertical image). SNES landscape-oriented 3D boxes won’t be drawn.

System variable « %SYSTEM% »

When drawing local files, you may want to draw different images regarding the current scraped system. The “%SYSTEM%” is designed for such situation. Use “%SYSTEM%” in the filename, and it will be replaced by the numeric identifier of the current scraped system.

Example:

```
<Item Type="LocalFile" LocalFilePath="default_Screenshot_%SYSTEM%.png">
  <Display X="100%" Y="100%" Width="50%" Height="33%" KeepRatio="true" Anchor="BottomRight" />
  <Rotation XOffset="50%" YOffset="50%" Angle="45.0" />
</Item>
```

While scraping:

- If the current scraped system is “Megadrive”, Skraper will look for the file « default_Screenshot_1.png ».
- If the current scraped system is “MAME”, Skraper will look for the file « default_Screenshot_75.png ».
- If the current scraped system is “PC-Engine”, Skraper will look for the file « default_Screenshot_31.png ».
- …

Notes:

- If the local file corresponding to the current scraped system is not found, Skraper will look for the file « default_Screenshot_0.png » (Do not forgot this file!).
- If « default_Screenshot_0.png » does not exist, Skraper will output an error log.
- The “%SYSTEM%” variable may be used anywhere in the filename, not only at the end.

Advanced filtering and object tree

Every “Item” may have children. In basic cases, there is no interests in building complex trees. However, when applying some filtering, it can be very efficient to build complex sub-objects..

The “Children” object is like que “Drawing” object, and may contain a list of “Item”, except that they are drawn right after the parent “Item”. As for “Fallbacks”, “Display” & “Rotation” sub-objects are inherited when they are not defined. The “children” container has 2 attributes to control different things:

```
<Children DrawIfNoParent="true" Reference="Parent" />
```

- **DrawIfNoParent** is a Boolean value:
  - *"True"*: Force the drawing of children even if the parent and its fallbacks are not available.
  - *"False"*: Children are drawn only if the parent or one of its fallbacks is drawn.
- **Reference** allow to change the reference rectangle (the Viewport) for the children.:
  - *"DoNotChange"*: Do not change the reference. The last reference will keep going down in the tree.
  - *"Parent"*: The rectangle of the current “Item” becomes the new reference for all the children. This rectangle may go down into the tree until it is explicitly redefined.
  - *"Viewport"* : The reference for all children is reset to the Viewport rectangle.

Default values are: DrawIfNoParent="False" Reference="DoNotChange".

Here are 4 examples of advanced filtering with complex trees:

Example #1:

```
<Item Type="Box3D" IfOrientation="IsPortrait">
  <Display X="0%" Y="100%" Width="50%" Height="50%" Anchor="BottomLeft" Antialiasing="High" />
  <Children>
    <Item Type="Support">
      <Display X="20%" Y="90%" Width="100%" Height="100%" Anchor="BottomLeft" />
    </Item>
  </Children>
</Item>
<Item Type="Box3D" IfOrientation="IsLandscape">
  <Display X="0%" Y="100%" Width="50%" Height="50%" Anchor="BottomLeft" Antialiasing="High" />
  <Children>
    <Item Type="Support">
      <Display X="5%" Y="70%" Width="100%" Height="100%" Anchor="BottomLeft" />
    </Item>
  </Children>
</Item>
```

Depending of the “3D Box” orientation, the “Support” is drawn to different coordinates:

- In Red: If the box is a landscape image, the support is drawn to X=20% and Y=90%.
- In Green: If the box is a portrait image, the support is drawn to X=5% and Y=70%.

Example #2:

```
<Item Type="Screenshot" IfOrientation="IsLandscape" TestOnly="true">
  <Children>
    <Item Type="Box3D">
      <Display X="5%" Y="70%" Width="20%" Height="20%" Anchor="BottomLeft" />
      <Rotation XOffset="25%" YOffset="25%" Angle="45.0" />
    </Item>
  </Children>
</Item>
```

In this example, the screenshot orientation is filtered to draw the “3D Box” only in landscape mode. However the screenshot itself is NOT drawn, vecause of the **TestOnly** attribute.

TestOnly is a Boolean attribute. When set to “true”, the item is never drawn. It is useful only for filtering purpose ad it is not inherited by any fallback or children.

Example #3:

```
<Item IfSystem="IsInTheList:1,2" TestOnly="true">
  <Children>
    <Item Type="Screenshot">
      <Display X="0%" Y="0%" Width="50%" Height="50%" Anchor="TopLeft" />
      <Fallback Type="ScreenshotTitle" />
    </Item>
    <Item Type="Box3D">
      <Display X="5%" Y="70%" Width="20%" Height="20%" Anchor="BottomLeft" />
      <Rotation XOffset="25%" YOffset="25%" Angle="45.0"
    </Item>
  </Children>
</Item>
```

In this third example, children are drawn only if the current scraped system is “Megadrive” or “Master System”. Note that the “Item” Type is not defined (“NoResource” by default) because in this particular case, it has no interest.

Example #4:

```
<Item IfSystem="IsNotInTheList:9,52" TestOnly="true">
  <Children>
    <Item Type="Screenshot" IfOrientation="IsLandscape" TestOnly="true">
      <Fallback Type="ScreenshotTitle" TestOnly="true"/>
      <Children DrawIfNoParent="true">
        <Item Type="LocalFile" LocalFilePath="Background_H_%SYSTEM%.png">
          <Display X="100%" Y="0%" Width="100%" Height="100%" Anchor="TopRight" />
        </Item>
        <Item Type="Screenshot">
          <Display X="474" Y="297" Width="714" Height="567" Anchor="TopRight" Antialiasing="None"/>
          <Fallback Type="ScreenshotTitle" IfOrientation="IsLandscape">
            <Fallback Type="LocalFile" LocalFilePath="NoScreenshot.png" /></Fallback>
        </Item>
        <Item Type="LocalFile" LocalFilePath="Foreground_H_%SYSTEM%.png">
          <Display X="100%" Y="0%" Width="100%" Height="100%" Anchor="TopRight" />
        </Item>
      </Children>
    </Item>
  </Children>
</Item>
```

This last example is more complex with a 2-state filtering by system, then by orientation (both filtering should be merged in the same item, but this sample is extracted from a real case, much more complex). Then the screenshot or title screenshot is drawn between 2 layers of local files.

Note:

Child “Items” and “Fallbacks” do not inherit the attributes of the “Item” itself. However, their sub-objects “Display” and “Rotate” are inherited unless they are explicitly redefined..

Future extensions

In the future, new color filtering and a possible “barrel effect” (to simulate the old CRT “bump”) should be added

ANNEX #1

Available Systems

System Name	ID
3DO	29
Aamber Pegasus	83
Acclaim	166
Action Max	81
Adam	89
Adventure Vision	78
Alpha Denshi Co.	182
Amcoe	178
American Laser Games	170
Amiga	64
Amiga (AGA)	111
Amiga CD	134
Amiga CD32	130
Amiga CD32 (hack)	139
Amiga CDTV	129
Android	63
Another Arcade Emulator	35
Apple II	86
Arcadia 2001	94
Archimedes	84
Astrocade	44
Atari 2600	26
Atari 2600 Supercharger	39
Atari 5200	40
Atari 7800	41
Atari 800	38
Atari Classics	160
Atari ST	42
Atari XE	43
Atlus	185
Atom	36
Atomiswave	53
Banpresto	186
BBC Micro	37
BK	93
Camputers Lynx	88
Capcom Classics	151
Capcom Play System	6
Capcom Play System 2	7
Capcom Play System 3	8
Cave	47
CD-i	133
Century Electronics	179
Channel F	80
Cinematronics	192
Coleco	183
Colecovision	48
Comad	177
Commodore 64	66
CoreGrafX	50
CPC	65
Daphne	49
Data East Classics	162
Dragon 32/64	91
Dreamcast	23
Dynax	173
EG2000 Colour Genie	92
Eighting / Raizing	152
Electron	85
Exidy	165
EXL 100	96
Family Computer Disk System	106
Flipper	197
FM-7	97
Future Pinball	199
Gaelco	194
Game & Watch	52
Game Boy	9
Game Boy Advance	12
Game Boy Color	10
Game Gear	21
Game Master	103
Game Pocket Computer	95
Game.com	121
Gamecube	13
GBA e-Reader	119
GP32	101
GX4000	87
IGS	176
Incredible Technologies	193
Intellivision	115
Irem Classics	148
Jaguar	27
Jaguar CD	171
Jaleco	159
Jupiter Ace	126
Kaneko	174
Konami Classics	158
Linux	145
Loopy	98
Lynx	28
Mac OS	146
Mame / FBA / Libretro	75
Master System	2
Mega Duck	90
Mega-CD	20
Megadrive - Sonic The Hedgehog 2 Hacks	203
Megadrive / Genesis	1
Megadrive 32X	19
Mega-Play	196
Mega-Tech	195
Midway Classics	150
Mikrosha	124
Mitchell	189
MO5	140
Model 2	54
Model 3	55
MSX	113
MSX R Turbo	118
MSX2	116
MSX2+	117
Namco Classics	155
Namco System 22	156
Naomi	56
Neo-Geo	142
Neo-Geo CD	70
Neo-Geo MVS	68
Neo-Geo Pocket	25
Neo-Geo Pocket Color	82
NES	3
N-Gage	30
Nichibutsu	180
Nintendo 3DS	17
Nintendo 64	14
Nintendo 64DD	122
Nintendo Classics	161
Nintendo DS	15
Nintendo Power	110
NMK	163
Oric 1 / Atmos	131
PC Dos	135
PC Engine	31
PC Engine CD-Rom	114
PC Engine SuperGrafx	105
PC Win3.xx	136
PC Win9X	137
PC Windows	138
PC-FX	72
Pecom 64	125
Pinball FX2	143
Pinball FX3	201
PlayChoice	184
Playstation	57
Playstation 2	58
Playstation 3	59
Playstation minis	172
Plus/4	99
PS Vita	62
Psikyo	167
PSP	61
PV-1000	74
Sammy Classics	164
Satellaview	107
Saturn	22
ScummVM	123
Sega Classics	147
Sega ST-V	69
Seibu Kaihatsu	190
SemiCom	187
Seta	149
SG-1000	109
Sharp X68000	79
Snes - Super Mario World Hacks	202
SNK Classics	154
Sufami Turbo	108
Super A'can	100
Super Cassette Vision	67
Super Game Boy	127
Super Game Boy 2	128
Super Nintendo / Super Famicom	4
Taito Classics	157
Technos	169
Tecmo	153
the Pinball Arcade	200
TI-99/4A	205
TO7	141
Toaplan	191
TRS-80 Color Computer	144
Type X	112
Universal	188
V.Smile	120
Vectrex	102
Vic-20	73
Video System Co.	175
Videopac G7000	104
Virtual Boy	11
Visco	181
Visual Pinball	198
Wii	16
Wii U	18
WonderSwan	45
WonderSwan Color	46
Xbox	32
Xbox 360	33
ZX Spectrum	76
ZX81	77

ANNEX #2

Parent/Children Systems

Megadrive / Genesis (1)	
	Megadrive 32X (19)
	Mega-CD (20)
	Megadrive - Sonic The Hedgehog 2 Hacks (203)
NES (3)	
	Family Computer Disk System (106)
Super Nintendo / Super Famicom (4)	
	Satellaview (107)
	Sufami Turbo (108)
	Nintendo Power (110)
	Snes - Super Mario World Hacks (202)
Game Boy (9)	
	Super Game Boy (127)
	Super Game Boy 2 (128)
Game Boy Advance (12)	
	GBA e-Reader (119)
Nintendo 64 (14)	
	Nintendo 64DD (122)
Atari 2600 (26)	
	Atari 2600 Supercharger (39)
Jaguar (27)	
	Jaguar CD (171)
PC Engine (31)	
	CoreGrafX (50)
	PC Engine SuperGrafx (105)
	PC Engine CD-Rom (114)
PSP (61)	
	Playstation minis (172)
Amiga (64)	
	Amiga (AGA) (111)
	Amiga CDTV (129)
	Amiga CD32 (130)
	Amiga CD (134)
	Amiga CD32 (hack) (139)
Mame / FBA / Libretro (75)	
	Capcom Play System (6)
	Capcom Play System 2 (7)
	Capcom Play System 3 (8)
	Another Arcade Emulator (35)
	Cave (47)
	Daphne (49)
	Atomiswave (53)
	Model 2 (54)
	Model 3 (55)
	Naomi (56)
	Neo-Geo MVS (68)
	Sega ST-V (69)
	Type X (112)
	Neo-Geo (142)
	Sega Classics (147)
	Irem Classics (148)
	Seta (149)
	Midway Classics (150)
	Capcom Classics (151)
	Eighting / Raizing (152)
	Tecmo (153)
	SNK Classics (154)
	Namco Classics (155)
	Namco System 22 (156)
	Taito Classics (157)
	Konami Classics (158)
	Jaleco (159)
	Atari Classics (160)
	Nintendo Classics (161)
	Data East Classics (162)
	NMK (163)
	Sammy Classics (164)
	Exidy (165)
	Acclaim (166)
	Psikyo (167)
	Technos (169)
	American Laser Games (170)
	Dynax (173)
	Kaneko (174)
	Video System Co. (175)
	IGS (176)
	Comad (177)
	Amcoe (178)
	Century Electronics (179)
	Nichibutsu (180)
	Visco (181)
	Alpha Denshi Co. (182)
	Coleco (183)
	PlayChoice (184)
	Atlus (185)
	Banpresto (186)
	SemiCom (187)
	Universal (188)
	Mitchell (189)
	Seibu Kaihatsu (190)
	Toaplan (191)
	Cinematronics (192)
	Incredible Technologies (193)
	Gaelco (194)
	Mega-Tech (195)
	Mega-Play (196)
Neo-Geo Pocket Color (82)	
	Neo-Geo Pocket (25)
MSX (113)	
	MSX2 (116)
	MSX2+ (117)
	MSX R Turbo (118)
PC Dos (135)	
	PC Win3.xx (136)
	PC Win9X (137)
	PC Windows (138)
TO7 (141)	
	MO5 (140)