

Skraper « User-mix », documentation

Cible : SkraperUI Beta 7



NEOGEO, COCKET
COLOR

NGPC

VIDEO PAC

O2EM

PC Engine

PCENGINE

CD-ROM

PCENGINECD

MSX

PRBOOM

PSP

PSP

PlayStation

PSX

ScummVM

SCUMMVM

SEGA 32X

SEGA32X

SEGA CD

SEGA CD

SEGA SG-1000

SG1000

SNES

SNES

SUPER GRAFX

SUPERGRAFX

THOMSON

THOMSON

Bêta

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

Requests : 2430

Downloaded Media : 2119

Saved Media : 332

Errors : 0

Scraped Games : 352

Games not found : 1

Estimated remaining time :

00:03:07

Favorites : 0

Hidden : 0

2018-06-14 20:20:20.069 INFO : Completed game Pro 18 world tour Goit [FR,EU,DE,ES,IT,SV] (Pro 18 - world tour Goit (Europe) (En,Fr,It))

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 Park

2018-06-14 20:20:20.102 INFO : Completed game Cubix Robots For Everyone - Race'n Robots [EU] (Cubix Robots for Everyone - Race 'n Robots

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 : Resurrection [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

1

Table des matières

Introduction	3
Balises « Globales »	4
La balise « Information »	5
La balise « Viewport »	5
Les balises « Drawings » et « Item »	6
Les balises « Display » et « Rotation »	9
Mode d'utilisation « Standard »	9
Mode d'utilisation « Projection »	11
La balise « Fallback » (Plan de secours)	12
Filtrage d'un « Item » par Systèmes et/ou Orientation	13
Variable « %SYSTEM% »	14
Filtrage avancé « Children »	15
Future implémentation	17
ANNEXE 1	18
Tous les systèmes	18
ANNEXE 2	19
Systèmes « Parents / Enfants »	19

Introduction

Ce document décrit le contenu des fichiers « User mix » XML destinés à être utilisés dans le logiciel SkraperUI. Ces fichiers permettent de créer ses propres mix afin de les adapter à ses besoins.

L'Extensible Markup Language (XML) est un métalangage informatique de balisage générique qui dérive du SGML. Sa syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire. Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes.

Les fichiers User mix sont une liste d'objets que SkraperUI peut lire et interpréter pour créer des compositions d'images basées sur les ressources graphiques fournies par la base de données <http://www.screenscraper.fr>

- Une description générale du format XML est disponible ici : https://fr.wikipedia.org/wiki/Extensible_Markup_Language
- Une documentation plus technique est disponible ici : <http://pages.videotron.com/fyergeau/w3c/xml10/REC-xml-19980210.fr.html> ou <https://www.w3.org/TR/xml/>

Balises « Globales »

Les fichiers XML User Mix sont constitués de 3 types de balises principales afin de composer une image :

- **Information** : Contient la description et les informations textuelles du Mix.
- **Viewport** : Contient la description de l'image de sortie (largeur, hauteur, couleur de fond, format, ...).
- **Drawings** : Contient la liste (ou le tableau) d'un ou plusieurs « Item » (éléments graphiques).

Voici un exemple de la structure générale :

```
<?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 ... informations ... />
  <Viewport ... description de l'image de sortie ... />
  <Drawings>
    <Item>
      ... élément graphique #1 ...
    </Item>
    <Item>
      ... élément graphique #2 ...
    </Item>
    <Item ..... />
      ... etc.

    <Item ..... />
    <Item>
      ... dernier élément graphique ...
    </Item>
  </Drawings>
</ImageComposition>
```

Notes :

- L'utilisation de commentaires est possible en utilisant la balise suivante : `<!-- ...Texte de mon commentaire... -->` (L'utilisation de « -- » est interdite dans le texte du commentaire).
- « <https://www.skraper.net/ImageComposition.xsd> » qui est présent dans ImageComposition (Voir l'exemple ci-dessus) sert d'indicateur de schéma et permet de vérifier la validité du XML.
- L'utilisation d'un validateur XML pour valider la syntaxe et mettre en forme de façon harmonieuse est toujours une bonne idée. Des validateurs et formateurs « en ligne » de format XML sont accessibles gratuitement ici : <https://www.freeformatter.com/xml-validator-xsd.html> ou <https://jsonformatter.org/xml-formatter>

La balise « Information »

Elle contient uniquement 3 attributs « textes » :

- ShortName
- LongName
- Description
- Author

Voici un exemple :

```
<Information ShortName="Mix4" LongName="4 Images Mix" Description="3DBox, Support, Screenshot, Wheel" Author="Archange154"/>
```

Seul l'attribut « LongName » est utilisée par SkrapperUI (pour le moment) mais il est fortement recommandé de remplir correctement les 3 champs. Pour l'utilisation des attributs, leurs noms parlent d'eux-mêmes, inutile de s'étaler davantage.

La balise « Viewport »

Elle contient les attributs décrivant le format avec lequel l'image de sortie sera construite et sauvegardée.

Voici un exemple :

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

- **Width** et **Height** représentent la taille de l'image de sortie en pixels (Dans l'exemple : 1024 x 768 pixels).
- **Color** définit la couleur de fond de l'image de sortie (avant que les éléments graphiques n'y soient ajoutés)
 - Le format utilisé est « A, R, G, B » (Alpha, Rouge, Vert, Bleu) où chaque valeur peut varier de 00 à FF (Valeurs hexadécimales).
 - Alpha définit la transparence du fond. Si la composante Alpha est égale à 00, la couleur est totalement transparente, si elle est égale à FF, la couleur est totalement opaque.
 - Par exemple : `Color="#80FF0000"` produira un rouge transparent à 50%, qui se fondra avec l'arrière-plan sur lequel l'image sera exploitée (très probablement, l'arrière-plan de votre front-end).
- **ImageFormat** définit le format d'enregistrement de l'image de sortie. Il supporte les formats « Png » et « Jpg », mais actuellement, la valeur est ignorée. Le format de sortie est toujours Png.
- **CompressionPercent** est le taux de compression en % du format Jpg. Il peut varier de 0 à 100 (La valeur est ignorée pour le format Png).

Si des attributs sont manquants, les attributs suivants seront utilisés par défaut :

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

Note : L'image de sortie sera appelée dans la suite du document « Viewport ».

Les balises « Drawings » et « Item »

Nous arrivons maintenant au « cœur » du Mix mais, à partir d'ici, les choses deviennent un peu plus complexes.

La balise « Drawings » décrit une liste (ou un tableau) de balises enfants appelées « Item ». Chaque « Item » décrit une ressource « ScreenScraper » qui sera téléchargée et utilisée dans le Viewport.

Les éléments sont tracés dans le Viewport dans le sens de lecture. Cela signifie que l'élément X+1 peut recouvrir totalement ou partiellement l'élément X.

L'Item le plus simple contient :

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

- **Type** représente la ressource graphique à télécharger et à ajouter au mix. Il doit être une chaîne de caractère valide parmi les suivantes :

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

La plupart d'entre elles correspondent directement à une ressource graphique de la base de donnée <http://www.screenscraper.fr>.

Mais, intéressons-nous aux 6 ressources « spéciales » ne se trouvant pas dans la base ScreenScraper.fr :

- **Region1** et **Region2** sont des drapeaux ronds (de taille d'origine : 128 x 128 pixels) correspondants aux régions du jeu en cours de scrape.
 - Si le jeu a deux régions, *Region1* et *Region2* contiendront une image.
 - Si le jeu n'a qu'une région, seule *Region1* contiendra une image.
 - Si le jeu n'a pas de régions, *Region1* et *Region2* seront vides.

Gardez en tête que *Region1* et *Region2* dépendent uniquement des régions définies dans la base de donnée ScreenScraper.fr pour le jeu en cours de scrape et non aux « tags » de régions que vous pourriez voir dans le nom du fichier (même si théoriquement, ils devraient être identiques).

- **LocalFile** permet d'utiliser une image « locale » à la place d'une ressource à télécharger. C'est-à-dire, un fichier image présent sur votre disque dur qui doit se trouver dans le même répertoire que votre fichier « Usermix.xml ». Quand vous utilisez **LocalFile**, vous devez ajouter un attribut **LocalFilePath**, qui contient le chemin de l'image locale, chemin relatif au fichier XML.

Exemple :

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

- **Include** permet d'utiliser un autre Mix à la place d'une ressource à télécharger. C'est-à-dire, un fichier mix xml présent sur votre disque dur qui doit se trouver dans le même répertoire que votre fichier « Usermix.xml ». Quand vous utilisez **Include**, vous devez ajouter un attribut **LocalFilePath**, qui contient le chemin du mix, chemin relatif au fichier XML. Include va donc charger et tracer le mix dans son propre Viewport (pas celui du mix courant). L'image résultante deviendra celle de cet item, comme si c'était n'importe quelle autre ressource image.

Exemple :

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

- **NoResource** permet de n'utiliser aucun type.

- **Text** permet de tracer le texte de son choix (ou provenant d'une variable prédéfinie). L'utilisation de **Text** implique également celle d'attributs permettant de formater ce texte :
 - **Text**="Mon Texte", **Text**="%variables%" ou **Text**="Mon Texte %variables% Texte" : On peut tracer un texte fixe prédéfini (Mon Texte) et/ou un texte qui dépend du jeu en cours de scrape. Il est possible de forcer un changement de ligne en ajoutant dans « Mon Texte » le caractère spécial « %% ».
%variables% peut prendre les valeurs suivantes :

- %name%	- %players%
- %description%	- %genre%
- %publisher%	- %region%
- %developer%	- %releasedate%
 - **TextColor** définit la couleur de fond du texte et fonctionne comme l'attribut « Color » du Viewport. Une couleur inverse est ajoutée en liseré. Par exemple, si la couleur choisie est blanche, le texte sera tracé blanc avec un léger filet noir autour, pour le détacher sur fond clair.
 - **FontFamily** permet de choisir avec quelle police sera tracé le texte. Si la police n'existe pas ou n'est pas disponible, Windows choisira la police la plus proche. La taille de la police est calculée automatiquement afin de s'ajuster au mieux à la taille du rectangle défini pour le tracé du texte.
 - **FontStyle** permet de mettre en forme le texte grâce à la combinaison des 4 valeurs suivantes, séparées par des espaces : "Bold Italic Strikeout Underline"
 - **TextAlign** permet de modifier l'alignement du texte. Les possibilités sont identiques à celles de l'attribut « Anchor » de la balise « Display » (Voir page suivante).

Exemple :

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


Les balises « Display » et « Rotation »

Ces deux balises sont dépendantes de « Item ». Elles permettent de mettre en forme et de placer dans le Viewport un Item grâce à leurs attributs. Il y a 2 manières de mettre en forme un média :

- Le mode « Standard » permet de placer des images de manière classique, elles seront toujours rectangulaires.
- Le mode « Projection » permet de placer des images en les déformant de manière à leur donner un effet « 3D ».

Mode d'utilisation « Standard »

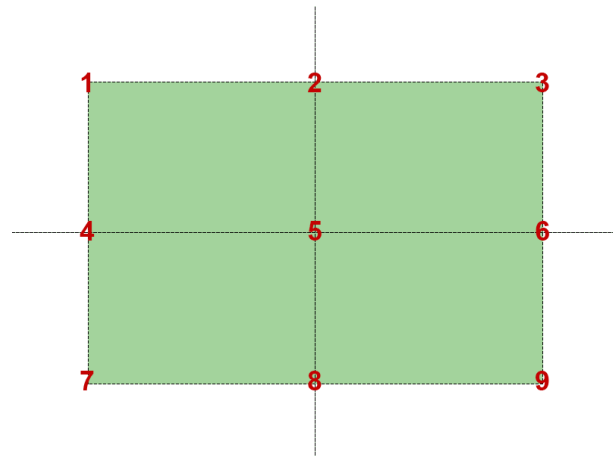
Voici un exemple complet :

```
<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** représente le point de référence (ou point d'encrage) de l'Item afin de le placer dans le Viewport. Par convention, le point de référence d'une image est son extrémité « haut – gauche », ce qui signifie que l'image sera tracée en partant vers la droite et vers le bas à partir de ce point. Cependant, vous pouvez choisir un autre point de référence parmi les 9 prédéfinis. L'élément sera tracé relativement à partir de ce nouveau point.

Les valeurs d'encrage valides sont :

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



Par exemple, pour tracer une image qui part du coin en bas à droite et qui va vers le centre, on utilise :

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

Autre exemple, pour centrer une image dans le Viewport (quelle que soit sa taille) :

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

Note : Pour le moment, il n'est pas possible de spécifier un point d'encrage en pixel, en pourcentage, ou en dehors de l'image.

- **X, Y, Width et Height** permettent de positionner (X et Y) et de redimensionner (Width et Height) l'image dans et par rapport au Viewport. Les valeurs peuvent être exprimées de 3 manières différentes :
 - Les valeurs numériques directes (Ex : "25") sont des coordonnées ou des tailles en pixels.
 - Les valeurs pourcentages (Ex : "50%") sont des coordonnées ou des tailles en pourcentage de la taille du "Viewport". Exemples :
 - "X": "50%" signifie que l'image sera tracée à partir de 50% de la largeur du Viewport (à partir du 400ème pixel avec un Viewport de taille 800 x 600 pixels)
 - "Width": "33%" signifie que la largeur de l'élément graphique sera redimensionnée à 33% de la largeur du Viewport.
 - Les valeurs exclamations (Ex : "100!") sont des coordonnées ou des tailles en pourcentage de la taille opposée du Viewport. Exemples :
 - X="100!" ou Width="100!" signifie 100% de la hauteur du Viewport.
 - Y="100!" ou Height="100!" signifie 100% de la largeur du Viewport.

Les valeurs exclamations sont utiles pour contrôler la taille des images lorsqu'on utilise les rotations. Par exemple, "Width": "100!" retaille la largeur de l'image à 100% de la hauteur du Viewport. Tournez l'image de 90°, et elle rentrera parfaitement dans le Viewport.
- **KeepRatio** est une valeur indiquant si l'aspect de l'image doit être conservé ou non. Deux valeurs sont possibles :
 - *"false"* : l'image sera retaillée strictement selon les valeurs "Width"/"Height", et pourra donc être déformée.
 - *"true"* : l'image sera retaillée pour entrer en mieux dans le rectangle "Width"/"Height", tout en conservant l'aspect originel de l'image (pas de déformation).
- **Antialiasing** est le type d'anti aliasing à appliquer lors du redimensionnement et du tracé de l'image. Trois valeurs sont possible avec chacune un rendu différent :
 - *"None"* : Aucun anti-aliasing. Si l'image est agrandie, elle gardera son aspect "8bit" que nous aimons tant. C'est idéal pour les screenshots.
 - *"Medium"* : Anti-aliasing moyen (AA bilinéaire). Donne un aspect plus fin tout en gardant les pixels visibles.
 - *"High"* : Anti-aliasing fort utilisant un algorithme bi cubique de haute qualité (AA bicubique). Idéal pour les formes non rectangulaires comme les logos, les boîtiers 3D, ...
- **Transparency** est une valeur à virgule flottante, indiquant la transparence globale de l'image à appliquer lors du tracé. Elle peut aller de 0.0 (pas de transparence) à 1.0 (complètement transparent). Gardez à l'esprit que ce n'est pas une valeur « Alpha ».
- **Effect** permet d'ajouter un filtre à l'image. Les valeurs possibles sont :
 - *"None"* : Aucun filtre.
 - *"BlackAndWhite"* : Filtre « Noir & Blanc ».
 - *"Sepia"* : Filtre « Sepia ».
 - *"GreenGameboy"* : Filtre « Game Boy ». Cela donne une couleur jaune/verte comme l'affichage de l'écran de la première Nintendo GameBoy.
- **XOffset, YOffset et Angle** permettent d'apporter une rotation à l'image.
 - *Angle* est une valeur flottante représentant l'angle de rotation, autour d'un point défini (voir point suivant), pouvant aller de 0.0° à 360.0°.
 - *XOffset* et *YOffset* spécifient le point de référence pour la rotation. Les valeurs sont exprimables en pixels ou en pourcentage (comme pour X, Y, Width et Height). Cependant, les pourcentages (ou pixels) s'appliquent à la taille de l'image à tourner, pas à la taille du Viewport.

Si des attributs sont manquant, les attributs suivants seront utilisés par défaut :

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

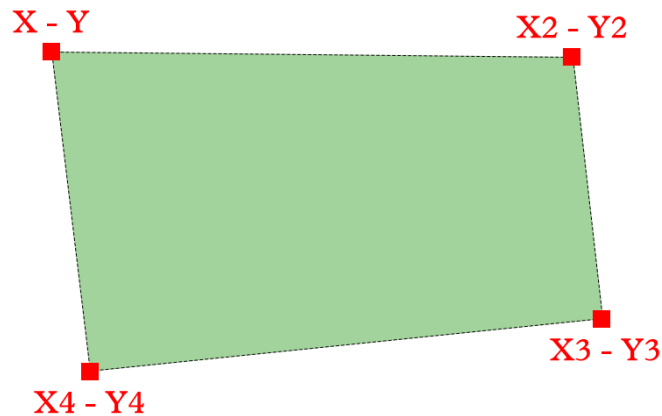
Note : Les balises « Display » et « Rotation » ne sont pas obligatoires pour un « Item ». Si l'une, l'autre ou les deux sont absentes, leurs effets sont ignorés.

[Mode d'utilisation « Projection »](#)

Voici un exemple complet :

```
<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** permettent de positionner les 4 coins de l'image dans et par rapport au Viewport. Les valeurs peuvent être exprimées de 3 manières différentes (Voir mode « Standard »). Les 4 points sont définis comme sur le schéma suivant :



- **Antialiasing** est le type d'anti aliasing à appliquer lors du redimensionnement et du tracé de l'image. Trois valeurs sont possibles avec chacune un rendu différent :
 - *"None"* : Aucun anti-aliasing. Si l'image est agrandie, elle gardera son aspect "8bit" que nous aimons tant. C'est idéal pour les screenshots.
 - *"Medium"* : Anti-aliasing moyen (FSAAx2). Donne un aspect plus fin tout en gardant les pixels visibles.
 - *"High"* : Anti-aliasing fort (FSAAx4). Idéal pour les formes non rectangulaires comme les logos, les boîtiers 3D, ...
- **Transparency** (Voir Mode « Standard »)
- **Effect** (Voir Mode « Standard »)

Notes :

- Les valeurs d'Antialiasing « Medium » et « High » entraînent un allongement du temps de scrape à cause du plus grand nombre de pixels à calculer lors du tracé.
- La balise « Rotation » et ses attributs fonctionnent avec ce mode mais il n'est pas conseillé de l'utiliser.

La balise « Fallback » (Plan de secours)

Toute balise « Item » peut contenir une « balise enfant » de secours nommée « Fallback » (récursivement, permettant de faire du « fallback » en chaîne). Les balises « fallback » sont utilisées lorsque la ressource demandée n'est pas disponible et se comportent exactement comme les « Item ». Un exemple vaut mieux qu'un long discours :

```
<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>
```

La balise « Item » principale demande un screenshot. Pour le jeu en cours de scrape, si le screenshot n'est pas disponible, sa balise « fallback » sera utilisée, et demandera le screenshot titre. Si le screenshot titre n'est pas disponible, c'est sa balise fallback qui sera utilisée et une image locale sera tracée.

Si elles ne sont pas redéfinies, les balises « fallback » héritent des attributs « Display » et « Orientation » de leur « parent ». Donc, l'exemple ci-dessus a un comportement identique à l'exemple suivant :

```
<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>
```

Les balises « fallback » ScreenshotTitle et LocalFile héritent des mêmes attributs « Display » et « Rotation » que le Screenshot.

Autre exemple :

```
<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>
```

Les balises « fallback » ScreenshotTitle et LocalFile héritent des mêmes attributs « Display » que le Screenshot. Pour « Rotation » ScreenshotTitle hérite de la Rotation de Screenshot mais pour Fallback LocalFile », Rotation est redéfinie. *XOffset* et *YOffset* ne sont pas précisés donc ils prennent les valeurs par défaut ("50%"), Angle prend la valeur "45.0".

Filtrage d'un « Item » par Systèmes et/ou Orientation

Les balises « Item » supportent un filtrage basique sur la valeur du « Système ScreenScaper ». Cela signifie qu'un objet peut être tracé ou non en fonction du « Système ScreenScaper » utilisé pour le jeu en cours de scrape. Le filtrage est basé sur les identifiants numériques des systèmes utilisés sur <http://www.screenscraper.fr>.

Malheureusement, vous ne pouvez pas y avoir accès directement, mais vous pouvez utiliser la liste en Annexe 1 ou obtenir la liste complète à partir l'API de démonstration ici :

<https://www.screenscraper.fr/api/systemesListe.php?devid=xxx&devpassword=yyy&softname=zzz&output=json&ssid=test&sspassword=test>

Pour ajouter un filtre à un objet « Item », il faut lui ajouter un attribut « IfSystem » et/ou « IfOrientation ».

Exemple :

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

- **IfSystem** spécifie le type de filtre pour le système (inclusion ou exclusion). Cinq valeurs sont possibles :
 - *"IsInTheList :IdSystem#1,...,IdSystem#X"* : L'Item sera tracé uniquement pour les systèmes spécifiés dans la liste.
 - *"IsNotInTheList :IdSystem#1,...,IdSystem#X"* : L'Item sera toujours tracé, sauf pour les systèmes spécifiés dans la liste.
 - *"IsInTheListOrAncestors :IdSystem#1,...,IdSystem#X"* : L'Item sera tracé uniquement pour les systèmes spécifiés dans la liste ainsi que tous leurs systèmes « enfants » (cf. Annexe 2).
 - *"IsNotIntheListOrAncestors :IdSystem#1,...,IdSystem#X"* : L'Item sera toujours tracé, sauf pour les systèmes spécifiés dans la liste ainsi que tous leurs systèmes « enfants » (cf. Annexe 2).
 - *"DoNotCare"* : Pas de système particulier (Valeur par défaut).
- **IfOrientation** spécifie le type d'orientation que doit avoir l'Item pour qu'il soit inclus. Trois valeurs sont possibles :
 - *"IsPortrait"* : La hauteur de l'élément est supérieure à sa largeur (Ex : 200 x 500 pixels).
 - *"IfLandscape"* : La hauteur de l'élément est inférieure ou égale à sa largeur (Ex : 500 x 200 pixels).
 - *"IsWhatever"* : Pas d'orientation particulière (Valeur par défaut).

Notes :

- Dans une chaine de « fallback », un filtre est appliqué à l'Item dans lequel il est défini et il est hérité par ses « fallback ».
- Il est également envisageable de filtrer ces derniers de la même façon.
- Il est possible de filtrer par n° de système, par orientation ou les deux en même temps.

Exemple n°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>
```

Cet « Item » de type « Wheel » ne sera tracé que pour les systèmes n°1 et n°2 ainsi que tous leurs systèmes « enfants » :

- Megadrive, Megadrive 32X, Mega-CD, Megadrive - Sonic The Hedgehog 2 Hacks
- Master-system

Exemple n° 2 :

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

Cet « Item » de type « Box3D » ne sera tracé que si son orientation est de format « Portrait ».

Variable « %SYSTEM% »

Comme pour le filtrage que l'on vient de voir, la variable « %SYSTEM% » a pour but de filtrer un objet « Item » sur la valeur du « Système ScreenScraper ». Cependant, contrairement au filtrage, cette variable s'utilise **uniquement** avec un « Item » de type « LocalFile ». Cela permet donc de créer des fichiers locaux particuliers pour les systèmes que l'on souhaite personnaliser.

La variable « %SYSTEM% » est à insérer dans le nom du fichier local à rechercher (« LocalFilePath »).

Exemple :

```
<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>
```

Lors du scrape :

- Si le système en cours est Megadrive, l'image tracée sera le fichier local nommé « default_Screenshot_1.png ».
- Si le système en cours est MAME, l'image tracée sera le fichier local nommé « default_Screenshot_75.png ».
- Si le système en cours est PC Engine, l'image tracée sera le fichier local nommé « default_Screenshot_31.png ».
- ...

Notes :

- Si le fichier local recherché n'est pas trouvé, l'image tracée sera le fichier local nommé « default_Screenshot_0.png » (donc, ne pas oublier de le créer lors de la conception du mix).
- Si « default_Screenshot_0.png » n'existe pas, une erreur de SkraperUI se produira.
- La variable est utilisable à n'importe quel endroit dans le nom de fichier, pas uniquement à la fin (« default_%SYSTEM%_Screenshot.png » ou « %SYSTEM%_default_Screenshot.png » sont parfaitement valides).

Filtrage avancé « Children »

Nous avons vu comment filtrer un « Item ». Mais il est possible grâce à la balise « Children » de tracer un ou plusieurs objets « enfants » en fonction de l'objet « parent ».

Une balise « Children » dispose de deux attributs permettant de la définir : « DrawIfNoParent » et « Reference » comme dans l'exemple suivant :

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

- **DrawIfNoParent** est un booléen qui peut prendre pour valeurs :
 - *"True"* : Permet de forcer le tracé des enfants même si le parent et ses fallbacks ne sont pas disponibles.
 - *"False"* : Les enfants sont tracés uniquement si le parent ou l'un de ses fallbacks est disponible.
- **Reference** permet de modifier le rectangle de référence pour les enfants. Trois valeurs sont possibles :
 - *"DoNotChange"* : La référence pour le positionnement des enfants est le dernier rectangle défini. Il va descendre dans la hiérarchie des enfants.
 - *"Parent"* : La référence pour le positionnement des enfants est le rectangle du parent courant (ce rectangle peut donc descendre dans toute la hiérarchie des enfants, tant qu'il n'est pas explicitement redéfini sur un autre Parent ou Viewport)
 - *"Viewport"* : La référence pour le positionnement des enfants est le rectangle du Viewport.

Les valeurs par défaut sont : DrawIfNoParent="False" Reference="DoNotChange".

Les quatre exemples qui suivent seront plus parlant que de longues explications :

Exemple n° 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>
```

Lors du scrape, selon le format de la « Box3D », le « Support » ne sera pas tracé aux mêmes coordonnées (La « Box3D » est tracée dans les deux cas) :

- En rouge : Si la « Box3D » est de format Portrait, le « support » sera tracé à X=20% et Y=90%.
- En vert : Si la « Box3D » est de format Paysage, le « support » sera tracé à X=5% et Y=70%.

Exemple n° 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>
```

Ici, on se sert du format de l'Item « Screenshot » comme variable pour le filtrage pour ses « Children » mais on ne le trace pas grâce à l'attribut "TestOnly". La « Box3D » ne sera tracée que si le « Screenshot » est au format paysage.

TestOnly est un booléen qui peut prendre comme valeurs : *true* ou *false* (*false* étant la valeur par défaut). Si sa valeur est *true*, il permet de définir un « Item » qui ne sera pas tracé. TestOnly est le seul attribut qui n'est pas hérité pas les fallback.

Exemple n° 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>
```

Dans ce troisième exemple, le filtrage permet de définir pour quels systèmes les objets « GraphicElement » contenus dans « Children » seront tracés. Dans ce cas, le « Screenshot » et la « Box3D » seront tracés uniquement pour les systèmes 1 et 2 (Megadrive et Master System).

Exemple n° 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>
```

Dans ce dernier exemple, le filtrage « children » est un peu plus complexe. Dans un premier temps, les enfants seront tracés si le système n'est ni le 9 ni le 52. Puis, dans un deuxième temps, on filtre une nouvelle fois pour que les enfants soient tracés uniquement si le screenshot (ou le screenshot title) est au format Paysage. Dans le cas où les 2 screenshots sont inaccessibles, les enfants seront tracés (grâce à l'attribut DrawIfNoParent="true") et ce sera le fichier local NoScreenshot.png qui remplacera le screenshot (ou le title).

Note :

Les « Items » présents dans la balise « Children » n'héritent pas des attributs de l'Item « Parent ». Il faut redéfinir intégralement les attributs de « Item » des « Children ». Par contre, les Children héritent des attributs de « Display » et de « Rotation » du parent à moins qu'ils ne soient redéfinis dans chaque Item.

Future implémentation

Ajout d'un « barrel effect », pour arrondir une image et lui donner un aspect d'écran CRT.

ANNEXE 1

Tous les systèmes

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

ANNEXE 2

Systèmes « Parents / Enfants »

Megadrive / Genesis (1)	Eighting / Raizing (152)
Megadrive 32X (19)	Tecmo (153)
Mega-CD (20)	SNK Classics (154)
Megadrive - Sonic The Hedgehog 2 Hacks (203)	Namco Classics (155)
NES (3)	Namco System 22 (156)
Family Computer Disk System (106)	Taito Classics (157)
Super Nintendo / Super Famicom (4)	Konami Classics (158)
Satellaview (107)	Jaleco (159)
Sufami Turbo (108)	Atari Classics (160)
Nintendo Power (110)	Nintendo Classics (161)
Snes - Super Mario World Hacks (202)	Data East Classics (162)
Game Boy (9)	NMK (163)
Super Game Boy (127)	Sammy Classics (164)
Super Game Boy 2 (128)	Exidy (165)
Game Boy Advance (12)	Acclaim (166)
GBA e-Reader (119)	Psikyo (167)
Nintendo 64 (14)	Technos (169)
Nintendo 64DD (122)	American Laser Games (170)
Atari 2600 (26)	Dynax (173)
Atari 2600 Supercharger (39)	Kaneko (174)
Jaguar (27)	Video System Co. (175)
Jaguar CD (171)	IGS (176)
PC Engine (31)	Comad (177)
CoreGrafx (50)	Amcoe (178)
PC Engine SuperGrafx (105)	Century Electronics (179)
PC Engine CD-Rom (114)	Nichibutsu (180)
PSP (61)	Visco (181)
Playstation minis (172)	Alpha Denshi Co. (182)
Amiga (64)	Coleco (183)
Amiga (AGA) (111)	PlayChoice (184)
Amiga CDTV (129)	Atlus (185)
Amiga CD32 (130)	Banpresto (186)
Amiga CD (134)	SemiCom (187)
Amiga CD32 (hack) (139)	Universal (188)
Mame / FBA / Libretro (75)	Mitchell (189)
Capcom Play System (6)	Seibu Kaihatsu (190)
Capcom Play System 2 (7)	Toaplan (191)
Capcom Play System 3 (8)	Cinematronics (192)
Another Arcade Emulator (35)	Incredible Technologies (193)
Cave (47)	Gaelco (194)
Daphne (49)	Mega-Tech (195)
Atomiswave (53)	Mega-Play (196)
Model 2 (54)	Neo-Geo Pocket Color (82)
Model 3 (55)	Neo-Geo Pocket (25)
Naomi (56)	MSX (113)
Neo-Geo MVS (68)	MSX2 (116)
Sega ST-V (69)	MSX2+ (117)
Type X (112)	MSX R Turbo (118)
Neo-Geo (142)	PC Dos (135)
Sega Classics (147)	PC Win3.xx (136)
Irem Classics (148)	PC Win9X (137)
Seta (149)	PC Windows (138)
Midway Classics (150)	T07 (141)
Capcom Classics (151)	MO5 (140)