

Swell

Short reference

Copyright © 2003, Grégoire Lejeune

A long wave on water that moves continuously without breaking

About this project

What is Swell ?

Swell is a simple XML to SWF convert tool. It has been developed with Ming and Perl.

This reference is based on swell version 0.0.1a.

And Ming ?

Ming is a C library for generating SWF ("Flash") format movies, plus a set of wrappers for using the library from c++ and popular scripting languages like PHP, Python, and Ruby.

You can find more information about Ming at <http://ming.sourceforge.net/>.

Ming is copyright (C) 2000-2002 Opaque Industries.

What about Macromedia ?

Macromedia(r) does not sponsor, affiliate, or endorse this product and/or services.

Licence

Swell is copyright (C) 2003 Gregoire Lejeune <gregoire.lejeune@free.fr>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

<swell>

swell : top-level element.

Contents

info | set | import | include | fill | text | movie | shape | sprite | textfield | morph | sound

Attributs

- version : swell version [0.0.1a]

Example

```
<swell version="0.0.1">
  <!-- Swell script -->
</swell>
```

<info>

info : adds some information about the swell script.

Contents

file | author | date | composer

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>
  <!-- Swell script -->
</swell>
```

<file>

file : gives the swell script file name.

Attributs

- *name* : file name

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <!-- Swell script -->
</swell>
```

<author>

author : informations about the swell's script author.

Attributs

- *mail* : author mail
- *firstname* : author first name
- *lastname* : author last name

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <!-- Swell script -->
</swell>
```

<date>

date : gives the creation and last modification date of the script.

Attributs

- *creation* : creation date
- *modification* : last modification date

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
  </info>
</swell>
```

```
<composer name="fumseck" version="0.0.1a"/>
</info>

<!-- Swell script -->

</swell>
```

<composer>

composer : informations about the swell's script composer tool.

Attributs

- *name* : tool name used
- *version* : tool version

Example

```
<swell version="0.0.1">

  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <!-- Swell script -->

</swell>
```

<set>

set : sets (or modify) some configuration informations.

Attributs

- *unit* : set the unit [tp | pt | cm | in]

Example

```
<swell version="0.0.1">

  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <set unit="pt"/>

  <!-- Swell script -->

</swell>
```

<import>

import : import a swell library.

Attributes

- *lib* : library name

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <import lib="geometric"/>

  <!-- Swell script -->
</swell>
```

<include>

include : includes an external swell file.

Attributes

- *file* : path and file name

Example

```
<swell version="0.0.1">
  <info>
    <file name="testText.swell"/>
    <author firstname="gregoire" lastname="lejeune" mail="greg@swell.net"/>
    <date creation="2003-02-22" modification="2003-02-26"/>
    <composer name="fumseck" version="0.0.1a"/>
  </info>

  <!-- Swell script -->

  <include file="/path/to/my.swell"/>

  <!-- Swell script -->
</swell>
```

<fill>

fill : creation of a new fill.

Contents

insert

Attributs

- *name* : name of the new fill
- *type* : type of the fill [[solid](#) | [linear_gradient](#) | [radial_gradient](#) | [tiled_bitmap](#) | [clipped_bitmap](#)]
- *scale* : fill scale
- *rotate* : fill rotation

Example

```
<swell version="0.0.1">
  <!-- Swell script -->
  <fill name="fill14" type="linear_gradient" scale="0.05" rotate="90">
    <insert color="yellow" ratio="0" alpha="255"/>
    <insert color="red" ratio="1" alpha="0"/>
  </fill>
</swell>
```

<insert>

insert : adds a fill composant.

Attributs

- *color* : color name
- *bitmap* : image file (jpeg)
- *ratio* : ratio
- *alpha* : alpha

Example

```
<swell version="0.0.1">
  <!-- Swell script -->
  <fill name="imagefill" type="clipped_bitmap">
    <insert bitmap="bw.jpeg"/>
  </fill>
  <fill name="colorfill" type="solid">
    <insert color="red"/>
  </fill>
  <!-- Swell script -->
</swell>
```

<text>

text : creates a new text element.

Contents

moveto | #PCDATA

Attributs

- *name* : name of the text shape
- *font* : text font
- *height* : text font size
- *color* : color of the text
- *spacing* : spacing between caracteres
- *alpha* : alpha of the text color

Example

```
<swell version="0.0.1">
  <!-- Swell script -->
  <text font="Calligraph421-BT" height="2" color="MediumAquamarine" name="txt1">
    <moveto x="0" y="0"/>
    Powered by SWELL
    <moveto x="0" y="20"/>
    (c)2003 by Greg
  </text>
  <!-- Swell script -->
</swell>
```

<shape>

shape : creates a new shape element.

Contents

curveto | curve | circle | linestyle | line | lineto | move | moveto | leftfillmoveto | rightfillmoveto | leftfillrotateto | rightfillrotateto | leftfillscaletto | rightfillscaletto | leftfillskewto | rightfillskewto

Attributs

- *name* : shape name
- *leftfill* : fill name to use on left
- *rightfill* : fill name to use on right

Example

```
<swell version="0.0.1">
  <fill name="fill" type="solid">
    <insert color="red" />
  </fill>
  <shape name="shp1" leftfill="fill">
    <linestyle width="40" color="blue"
```

```

        alpha="150"/>
        <curveto x="50" y="100" cx="0" cy="100"/>
        <curve x="50" y="-100" cx="50" cy="0"/>
    </shape>

    <shape name="shp2" leftfill="fill">
        <linestyle width="40" color="blue"
            alpha="150"/>
        <circle radius="50"/>
    </shape>

    <shape name="shp3" leftfill="fill">
        <linestyle width="40" color="blue"
            alpha="150"/>
        <move x="60" y="0"/>
        <line x="60" y="60"/>
        <line x="-100" y="0"/>
        <line x="40" y="-60"/>
    </shape>

    <movie width="100" height="300"
        output="testShape.swf">
        <frame>
            <add x="2" y="2" id="shp1" />
            <add x="80" y="155" id="shp2" />
            <add x="2" y="210" id="shp3" />
        </frame>
    </movie>

</swell>

```

<linestyle>

linestyle : line style definition in a shape.

Attributes

- *width* : line width
- *color* : line color
- *alpha* : color alpha

Example

```

<swell version="0.0.1">
    <fill name="fill" type="solid">
        <insert color="red" />
    </fill>

    <shape name="shp1" leftfill="fill">
        <linestyle width="40" color="blue" alpha="150"/>
        <curveto x="50" y="100" cx="0" cy="100"/>
        <curve x="50" y="-100" cx="50" cy="0"/>
    </shape>

    <shape name="shp2" leftfill="fill">
        <linestyle width="40" color="blue" alpha="150"/>
        <circle radius="50"/>
    </shape>

```

```

<shape name="shp3" leftfill="fill">
  <linestyle width="40" color="blue" alpha="150"/>
  <move x="60" y="0"/>
  <line x="60" y="60"/>
  <line x="-100" y="0"/>
  <line x="40" y="-60"/>
</shape>

<movie width="100" height="300" output="testShape.swf">
  <frame>
    <add x="2" y="2" id="shp1" />
    <add x="80" y="155" id="shp2" />
    <add x="2" y="210" id="shp3" />
  </frame>
</movie>
</swell>

```

<button>

button : creates a new button element.

Contents

shapes | action

Attributs

- *name* : button name

Example

```

<swell version="0.0.1">
  <set unit="pt"/>
  <import lib="geometric"/>

  <fill name="fill1" type="solid">
    <insert color="red"/>
  </fill>
  <fill name="fill2" type="solid">
    <insert color="green"/>
  </fill>
  <fill name="fill3" type="solid">
    <insert color="yellow"/>
  </fill>

  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="up"
fill="fill1"/>
  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="down"
fill="fill2"/>
  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="over"
fill="fill3"/>

  <button name="but1">
    <shapes hit="up" up="up" down="down" over="over"/>
  </button>

  <movie width="30" height="20" output="testButton.swf" rate="16"
background="CornflowerBlue">
    <frame>

```

```

    <add y="5" x="5" id="but1"/>
  </frame>
</movie>

</swell>

```

<shapes>

shapes : defines the shapes for each button status.

Attributs

- hit : shape for status hit
- up : shape for status up
- down : shape for status down
- over : shape for status over

Example

```

<swell version="0.0.1">
  <set unit="pt"/>
  <import lib="geometric"/>

  <fill name="fill1" type="solid">
    <insert color="red"/>
  </fill>
  <fill name="fill2" type="solid">
    <insert color="green"/>
  </fill>
  <fill name="fill3" type="solid">
    <insert color="yellow"/>
  </fill>

  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="up"
fill="fill1"/>
  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="down"
fill="fill2"/>
  <rectangle width="2" height="5" linewidth="4" linecolor="blue" name="over"
fill="fill3"/>

  <button name="but1">
    <shapes hit="up" up="up" down="down" over="over"/>
  </button>

  <movie width="30" height="20" output="testButton.swf" rate="16"
background="CornflowerBlue">
    <frame>
      <add y="5" x="5" id="but1"/>
    </frame>
  </movie>

</swell>

```

<textfield>

textfield : creates a new text field element.

Contents

bounds | margins | #PCDATA

Attributs

- *name* : textfield name
- *align* : text alignment [left | right | center | justify]
- *noedit* : textfield is editable [0 | 1]
- *password* : textfield is type password [0 | 1]
- *drawbox* : add a box [0 | 1]
- *multiline* : textfield with with multiple lines [0 | 1]
- *wordwrap* : word wrap [0 | 1]
- *noselect* : text is not selectable [0 | 1]
- *color* : text color
- *alpha* : text color alpha
- *font* : text font
- *height* : text font size
- *indentation* : text indentation
- *spacing* : text spacing
- *autosize* : autosize the textfield [0 | 1]
- *html* : text is html [0 | 1]

Example

```
<swell version="0.0.1">
  <set unit="tp"/>

  <textfield name="txt1"
    align="left"
    noedit="0"
    password="0"
    drawbox="1"
    multiline="1"
    wordwrap="1"
    noselect="0"
    color="red"
    alpha="100"
    font="Bimini"
    height="10"
    indentation="1"
    spacing="1">
    <bounds width="25" height="15"/>
    <margins left="2" right="2"/>
    Welcome
  </textfield>

  <movie width="300" height="260"
    output="testTextField.swf" rate="5"
    background="DimGray">
    <frame name="frameadd">
      <add x="10" y="10" id="txt1"/>
    </frame>
  </movie>
```

```
</swell>
```

<bounds>

bounds : sets the width and height of the bounding box of the textfield.

Attributes

- *width* : bounding box width
- *height* : bounding box height

<margins>

margins : sets the margins of the text block.

Attributes

- *left* : left margin
- *right* : right margin

<morph>

morph : creates a new morph element.

Contents

shape1 | shape2

Attributes

- *name* : morph name

<shape1> and <shape2>

shape1 and *shape2* : first (1) and last (2) shapes for a *morph*.

Cf. <shape>.

<sound>

sound : creates a new sound element.

Attributes

- *name* : sound name
- *file* : sound file name (mp3)

Example

```
<swell version="0.0.1">
  <set unit="tp" />

  <sound name="sound" file="/path/to/my.mp3"/>

  <!-- frames = rate * mp3time -->
  <movie width="210" height="170" rate="24" frames="312" output="testSound.swf">
    <frame>
```

```
        <add id="sound" />
    </frame>
</movie>
</swell>
```

<color>

color : creates a color.

Attributes

- *name* : color name
- *r* : red
- *g* : green
- *b* : blue

Example

```
<swell version="0.0.1">
  <set unit="tp" />

  <color name="rand" r="82" g="17" b="144"/>

  <text font="Helmet" height="40" color="rand" name="txtSwell">
    SWELL
  </text>

  <movie width="320" height="240" output="test.swf">
    <frame>
      <add x="130" y="130" id="txtSwell"/>
    </frame>
  </movie>
</swell>
```

<sprite>

sprite : creates a new sprite (movie clip) element.

Contents

frame

Attributes

- *name* : sprite name
- *frames* : number total of frames

Example

```
<swell version="0.0.1">
  <set unit="pt"/>

  <!-- define shapes "star1" and "star2" -->

  <sprite name="sp1">
    <frame>
```

```
    <add id="star1"/>
  </frame>
  <frame>
    <add id="star2"/>
  </frame>
</sprite>

<movie width="30" height="20">
  <frame>
    <add x="5" y="10" id="sp1"/>
  </frame>
</movie>
</swell>
```

<movie>

movie : the SWF movie.

Contents

frame

Attributes

- *width* : movie width
- *height* : movie height
- *output* : output file name - if output is not specified, output is STDOUT
- *rate* : movie rate (frame/s)
- *background* : background color
- *frames* : number total of frames

Example

```
<swell version="0.0.1">

  <set unit="pt"/>

  <text font="Calligraph421-BT" height="2" color="MediumAquamarine" name="txt1">
    <moveto x="0" y="0"/>
    Powered by SWELL
    <moveto x="0" y="20"/>
    (c)2003 by Greg
  </text>

  <movie width="30" height="20" output="testText.swf" rate="5"
background="DimGray">
    <frame>
      <add x="5" y="10" id="txt1"/>
    </frame>
    <frame>
      <remove id="txt1"/>
    </frame>
  </movie>
</swell>
```

<frame>

frame : creates a new frame in the movie or in a movie clip.

Contents

add | remove | action | move | moveto | rotate | rotateto | scale | scaletto | coloradd | colormult | depth | skew | skewto

Attributs

- name : name of the frame
- nonext : no next frame [0 | 1]

Example

```
<swell version="0.0.1">
  <set unit="pt"/>

  <text font="Calligraph421-BT" height="2" color="MediumAquaMarine" name="txt1">
    <moveto x="0" y="0"/>
    Powered by SWELL
    <moveto x="0" y="20"/>
    (c)2003 by Greg
  </text>

  <movie width="30" height="20" output="testText.swf" rate="5"
background="DimGray">
    <frame>
      <add x="5" y="10" id="txt1"/>
    </frame>
    <frame>
      <remove id="txt1"/>
    </frame>
  </movie>
</swell>
```

<action>

action : in movie context, adds ActionScript code to the current frame of the movie.

action : in button context, activate a script when a button receives an event.

Contents

#PCDATA

Attributs

- event : associate an event to a button status [mouseup | mouseover | mousedown | mouseupoutside | dragover | dragout]

Example

```
<swell version="0.0.1">
  <set unit="tp" />
  <textfield name="txt1" align="left"
drawbox="1" multiline="1"
wordwrap="1" color="red">
```

```
        font="Courier-New" height="30"
        indentation="1" spacing="1">
<bounds width="1" height="1" />
<margins left="1" right="1" />
frame 12
</textfield>

<movie width="400" height="400"
        output="testAction.swf" rate="1"
        background="DimGray">
<frame name="frameadd">
  <add x="10" y="10" id="txt1" />
</frame>
<frame name="start">
  <action>
    txt1 = "frame 1";
  </action>
</frame>
<frame>
  <action>
    txt1 = "frame 2";
  </action>
</frame>
<frame>
  <action>
    gotoAndPlay( "start" );
  </action>
</frame>
</movie>
</swell>
```

<add>

add : places an element on the movie's frame.

Attributes

- *id* : name of the element to add
- *name* : new name for the element in the movie
- *x* : x position in the movie
- *y* : y position in the movie

Example

```
<swell version="0.0.1">
  <set unit="pt"/>

  <text font="Calligraph421-BT"
        height="2"
        color="MediumAquamarine"
        name="txt1">
    <moveto x="0" y="0"/>
    Powered by SWELL
    <moveto x="0" y="20"/>
    (c)2003 by Greg
  </text>

  <movie width="30"
        height="20">
```

```
        output="testText.swf"
        rate="5"
        background="DimGray">
    <frame>
        <add x="5" y="10" id="txt1" name="texte"/>
    </frame>
    <frame>
        <remove id="texte"/>
    </frame>
</movie>
</swell>
```

<remove>

remove : removes an element.

Attributes

- *id* : name of the element

Example

```
<swell version="0.0.1">
  <set unit="pt"/>

  <text font="Calligraph421-BT"
        height="2"
        color="MediumAquamarine"
        name="txt1">
    <moveto x="0" y="0"/>
    Powered by SWELL
    <moveto x="0" y="20"/>
    (c)2003 by Greg
  </text>

  <movie width="30"
        height="20"
        output="testText.swf"
        rate="5"
        background="DimGray">
    <frame>
        <add x="5" y="10" id="txt1" name="texte"/>
    </frame>
    <frame>
        <remove id="texte"/>
    </frame>
  </movie>
</swell>
```

<rotate>

rotate : adds the specified number of degrees to the element's current rotation value.

Attributes

- *id* : name of the element
- *radius* : number of degrees

<rotateto>

rotateto : sets the rotation attribut of the element, expressed in degrees.

Attributs

- *id* : name of the element
- *radius* : number of degrees

<scale>

scale : sets the horizontal and vertical scale to the specified values.

Attributs

- *id* : name of the element
- *x* : x scale value
- *y* : y scale value

<scaletto>

scaletto : sets the horizontal and vertical scale by multiplying the current scale by the given values.

Attributs

- *id* : name of the element
- *x* : x scale value
- *y* : y scale value

<skew>

skew : sets the horizontal and vertival skew attributes of the specified element.

Attributs

- *id* : name of the element
- *x* : horizontal skew attribute
- *y* : vertival skew attribute

<skew>

skew : sets the horizontal and vertival skew attributes of the specified element.

Attributs

- *id* : name of the element
- *x* : horizontal skew attribute
- *y* : vertival skew attribute

<coloradd>

coloradd : adds the given values to the color components of the specified element.

Attributs

- *id* : name of the element.
- *red* : red value

- green : green value
- blue : blue value
- alpha : alpha

<colormult>

colormult : multiplies each of the color components of the element by the given values.

Attributs

- *id* : name of the element
- red : red value
- green : green value
- blue : blue value
- alpha : alpha

<depth>

depth : sets the depth of the element.

Attributs

- *id* : name of the element
- *value* : depth value

<ratio>

ratio : sets the element ratio.

Attributs

- *id* : name of the element
- *value* : ratio value

<move>

move : in shape context, moves the current point of the pen from (x0, y0) to (x0+x, y0+y).

move : in frame context, moves the given element from (x0, y0) to (x0+x, y0+y).

Attributs

- *id* : element name (**frame context**)
- *x* : x coordinate
- *y* : y coordinate

<moveto>

moveto : in scale context, moves the current point of the pen to (x, y).

moveto : in frame context, moves the given element to (x, y).

Attributs

- *id* : element name (**frame context**)
- *x* : x coordinate
- *y* : y coordinate

<line>

line : draws a straight line from the current point (x0, y0) to the point (x0+x, y0+y).

Attributs

- x : x coordiante
- y : y coordiante

<lineto>

lineto : draws a straight line from the current point to the point (x, y).

Attributs

- x : x coordiante
- y : y coordiante

<curve>

curve : draws a curved line from the current point (x0, y0) to the point (x0+x, y0+y) using (x0+cx, y0+cy) as a control point.

Attributs

- x : x coordinate
- y : y coordinate
- cx : x controle coordinate
- cy : y controle coordinate

<curveto>

curveto : draws a curved line from the current point to the point (x, y) using (cx, cy) as a control point.

Attributs

- x : x coordinate
- y : y coordinate
- cx : x controle coordinate
- cy : y controle coordinate

<leftfillmoveto> / <rightfillmoveto>

leftfillmoveto / *rightfillmoveto* : same as *moveto*, but for the left/right fill of a shape.

Attributs

- x : x position
- y : y position

<leftfillrotateto> / <rightfillrotateto>

leftfillrotateto / *rightfillrotateto* : same as *rotateto*, but for the left/right fill of a shape.

Attributs

- *radius* : number of degrees

<leftfillscaleto> / <rightfillscaleto>

leftfillscaleto / *rightfillscaleto* : same as *scaleto*, but for the left/right fill of a shape.

Attributs

- *x* : x position
- *y* : y position

<leftfillskewto> / <rightfillskewto>

leftfillskewto / *rightfillskewto* : same as *skewto*, but for the left/right fill of a shape.

Attributs

- *x* : x skew
- *y* : y skew