

RetroFE
User Manual
V1.0

Table of Contents

1	Introduction.....	2
2	Basic Installation Instructions.....	3
3	Detailed Setup Guide.....	4
3.1	Linux vs Windows.....	4
3.2	Linux library installation.....	4
3.3	Compiling RetroFE on Linux.....	4
3.4	Windows library installation.....	5
3.5	Compiling RetroFE on Windows.....	5
3.6	Post installation.....	6
3.7	Configuration.....	6
3.8	Adding collections.....	6
4	Folder Structure.....	8
5	settings.conf.....	9
6	controls.conf.....	11
7	Collection.....	13
7.1	settings.conf.....	13
8	launcher.....	15
9	layout.....	17
9.1	General layout information.....	17
9.2	General layout structure.....	18
9.3	<layout> attributes.....	19
9.4	<sound> attributes.....	19
9.5	General attributes.....	20
9.6	<menu> attributes.....	23
9.7	<video> attributes.....	24
9.8	<image> attributes.....	25
9.9	<text> attributes.....	25
9.10	<reloadableVideo> attributes.....	26
9.11	<reloadableImage> attributes.....	27
9.12	<reloadableText> attributes.....	27
9.13	<reloadableScrollingText> attributes.....	28
9.14	mode attribute.....	29

1 Introduction

Welcome to the RetroFE User Manual. Within this manual you will learn how to configure RetroFE, use this front-end, and create/change themes. Since RetroFE is F/OSS, released under the GPL, you are free to download and change the source code within the boundaries of the GPL license. I hope you will enjoy using RetroFE to play your favorite games.

2 Basic Installation Instructions

Setting up RetroFE is relatively easy. Just go through the following steps:

- 1) Download the latest release from <http://retrofe.nl/download/>.
- 2) Extract the compressed image.
- 3) Edit the settings.conf file to configure RetroFE's general settings.
- 4) Edit the controls.conf file to configure the RetroFE's controls.
- 5) Run RetroFE to verify the front-end loads and exits correctly.
- 6) Add/remove/Edit the Collections with roms, art, etc.
- 7) Enjoy RetroFE.

If you encounter any issues, please check the log.txt file before asking for help.

A Linux executable is included in the release, but depending on your distribution you may need to compile your own executable from the source code, which can also be found in the download section.

Some people report that for Windows the installation of a codec pack for video playback may be needed. A suggestion would be to download [Shark's STANDARD video codec pack](#).

3 Detailed Setup Guide

3.1 Linux vs Windows

Please keep in mind that Linux uses / to separate its directories, while Windows uses \. Chapters that handle both Windows and Linux installations will generally use the Linux notation.

3.2 Linux library installation

In order to install and/or run RetroFE on Linux, you first need to make sure that you have the required libraries installed:

```
sudo apt-get install g++ cmake dos2unix zlib1g-dev libSDL2-2.0
libSDL2-mixer-2.0 libSDL2-image-2.0 libSDL2-ttf-2.0 libSDL2-dev
libSDL2-mixer-dev libSDL2-image-dev libSDL2-ttf-dev
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgstreamer-
plugins-good1.0-dev zlib1g-dev libglib2.0-0 libglib2.0-dev sqlite3
```

3.3 Compiling RetroFE on Linux

Some systems require you to compile your own RetroFE installation. You can find the source code in the download section of the RetroFE website. Simply unpack, enter the directory, and generate your gcc make files:

```
cmake RetroFE/Source -BRetroFE/Build
```

You can now compile RetroFE, and create a full environment:

```
cmake --build RetroFE/Build
python Scripts/Package.py --os=linux --build=full
```

Simply copy your live RetroFE system to any folder of your choosing:

```
cp -r Artifacts/linux/RetroFE /your/ideal/retrofe/path
```

and you're set.

3.4 Windows library installation

In order to compile RetroFE on Windows, you first need to make sure that you have the required libraries installed:

```
Install Python 2.7
Install sphinx with python
Install visual studio 2012
Install Microsoft Windows SDK for Windows 7 and .net Framework 4
http://www.microsoft.com/en-us/download/details.aspx?id=8279
Install cmake
Install tortoisehg
Install 7zip
Install gstreamer and gstreamer-devel to c:/gstreamer(x86, not 64
bit!) from http://gstreamer.freedesktop.org/data/pkg/windows/1.4.0
```

3.5 Compiling RetroFE on Windows

You may wish to compile your own RetroFE executable. You can find the source code in the download section of the RetroFE website. Simply unpack, enter the directory, and generate your visual studio solution files:

```
RetroFE\Source\SetupEnvironment.bat
cmake RetroFE\Source -BRetroFE/Build
-DGSTREAMER_ROOT=C:/gstreamer/1.0/x86
```

You can now compile RetroFE, and create a full environment:

```
cmake --build RetroFE\Build --config Release
python Scripts\Package.py --os=windows --build=full --
gstreamer_path=<path to your gstreamer x86 directory>
```

Simply copy your live RetroFE system to any folder of your choosing:

```
cp -r Artifacts\windows\RetroFE \your\ideal\retrofe\path
```

and you're set.

3.6 Post installation

After copying the RetroFE system to the directory of your choice, you're set to give your installation a first test by running the RetroFE executable in that directory. RetroFE comes with a pre-installed Sega Genesis system with one game so you can check if the installation went according to plan.

3.7 Configuration

The first configuration step is editing the RetroFE system configuration file `settings.conf`. In here you configure the screen settings, global theme, base paths, etc.

The second configuration step is editing the RetroFE controls file `controls.conf`. In here you configure the keys used to control the RetroFE front-end. Note that the default select key is space, and not enter as some people expected.

3.8 Adding collections

RetroFE starts with two (almost empty) collections, but more can be added easily. As an example, let's set up the Nintendo Entertainment System collection. First, enter the RetroFE/collections directory, and create an empty collection using the following command: `../retrofe -createcollection "Nintendo Entertainment System"`

Next we add the roms and artwork:

1. Download a NES romset from your favourite source, and place the roms in the RetroFE/collections/Nintendo Entertainment System/roms directory.
2. Download a device image, logo, and video for the system, and place the `device.png`, `logo.png`, and `video.mp4` files in the RetroFE/collections/Nintendo Entertainment System/system_artwork directory.
3. Download games artwork (artwork_front, logo's, screenshots, titleshots, videos, etc.) from your favourite art source, and place them in the RetroFE/collections/Nintendo Entertainment System/medium_artwork directory.

Now we configure the new system by editing RetroFE/collections/Nintendo Entertainment System/settings.conf. If you stick to the default directory structure, this file can be extremely simple:

```
list.extensions = nes
launcher = NES
```

The first line defines the ROM file extension as .nes; this should match the file extensions in your RetroFE/collections/Nintendo Entertainment System/roms directory. The second line defines the name of the launcher used for this collection. Before the collection can be used, this launcher needs to be [configured](#):

Edit the RetroFE/launchers/NS.conf (matching the launcher name defined in the settings.conf) file. I'm currently using MAME 0.179 for this purpose, so the launcher can be simple:

```
executable = mame
arguments = nes -cart "%ITEM_FILEPATH%"
```

As an example: if RetroFE starts the game Willow (USA).nes, this launcher will execute the command:

```
mame nes -cart "RetroFE/collections/Nintendo Entertainment
System/roms/Willow (USA).nes".
```

The last step is to add the newly created collection to the main menu by editing RetroFE/collections/Main/menu.xml, and add a line:

```
<item collection="Nintendo Entertainment System"/>
```

in the <menu> section. If you use a menu.txt file instead of a menu.xml file, simply add:

```
Nintendo Entertainment System
```

to your menu.txt file. When this is done, your newly added collection is ready for testing by running the RetroFE executable.

Happy playing! :)

4 Folder Structure

RetroFE uses the following folder structure (keep in mind that Windows uses \ in stead of /):

File/Directory	Description
./	Base folder
./log.txt	Log output
./controls.conf	Controller configuration (up, down, select, back, etc.)
./settings.conf	Global front-end settings (display options, layout, base paths, etc.)
./meta/	Folder that contains files imported into meta.db.
./meta.db	Game information database (year, manufacturer, genre, etc.). Delete this file if you have changed any of the files in the /meta/ directory.
./core/	Windows specific library files needed for RetroFE to run. This folder also includes the Windows retrofe.exe.
./RetroFE.lnk	Windows link to ./core/retrofe.exe
./Collections/	Directory that contains the (games) collections, menus, artwork, and roms.
./launchers/	Directory that contains the configuration files for the launchers (emulators).
./layouts/	Directory that contains the layouts/themes for the front-end display.

5 settings.conf

The global settings.conf file defines the overall RetroFE configuration. The following parameters can be defined:

Variable	Allowed values	Description
fullscreen	yes, true, no, false	Set the front-end to full-screen mode.
horizontal	stretch, #pixels	Screen pixel width (i.e. 1920). Stretch uses the desktop width.
vertical	stretch, #pixels	Screen pixel height (i.e. 1080). Stretch uses the desktop height.
hideMouse	yes, true, no, false	Hide the mouse cursor on the screen when RetroFE is active.
showParenthesis	yes, true, no, false	Hide the parenthesis (and text between them) in the title of an item.
showSquareBrackets	yes, true, no, false	Hide the square brackets (and text between them) in the title of an item.
firstCollection	<collection name>	Specify the name of the collection RetroFE should load on start-up (e.g. Main).
videoEnable	yes, true, no, false	Display videos. Set to “no” for old/slow systems.
videoLoop	#loops	Number of times the video playback will loop. Use 0 for a continuous loop.
exitOnFirstPageBack	yes, true, no, false	Exit RetroFE when the back button is pressed on the first collection.
attractModeTime	Time (in seconds)	Enter 0 to disable attract mode, otherwise enter the number of seconds to wait before the menu scrolls to another random item.
rememberMenu	yes, true, no, false	Remember the last highlighted menu item, and return to that item when re-entering a menu.
autoFavorites	yes, true, no, false	Automatically switch to the Favorites playlist if it is available.
minimize_on_focus_loss	yes, true, no, false	Overwrite system SDL default for <code>SDL_VIDEO_MINIMIZE_ON_FOCUS_LOSS</code> to force RetroFE to (not) minimize when the focus on the full-screen window is lost.

Variable	Allowed values	Description
baseMediaPath	Base path to your art files	Override if you choose to have your media stored outside of the RetroFE folder structure. Can be used by your collection's settings.conf.
baseItemPath	Base path to your rom files	Override if you choose to have your roms stored outside of the RetroFE folder structure. Can be used by your collection's settings.conf.

6 controls.conf

The controls.conf file contains the configuration of your controller. The following control codes can be configured:

Control	Description
up	Scrolls menu backward (for vertical menus)
down	Scrolls menu forward (for vertical menus)
left	Scrolls menu backward (for horizontal menus)
right	Scrolls menu forward (for horizontal menus)
pageUp	Scrolls menu backward by a page.
pageDown	Scrolls menu forward by a page.
letterUp	Scrolls menu to the previous letter in the alphabet.
letterDown	Scrolls menu to the next letter in the alphabet.
addPlaylist	Adds a game to the favorites playlist.
removePlaylist	Removes a game from the favorites playlist.
nextPlaylist	Switches between the full game list, and the favorites playlist.
random	Selects a random game.
select	Selects the active menu item.
back	Leaves the current menu.
quit	Exits RetroFE.

RetroFE supports the following gamepad/joystick codes:

Keycode	Description
joyXbuttonY	Gamepad button (X = joypad number, Y = button number)
joyXHatYLeftUp	Gamepad hat left-up direction (X = joypad number, Y = button number)
joyXHatYLeft	Gamepad hat left direction (X = joypad number, Y = button number)
joyXHatYLeftDown	Gamepad hat left-down direction (X = joypad number, Y = button number)
joyXHatYRightUp	Gamepad hat right-up direction (X = joypad number, Y = button number)
joyXHatYRight	Gamepad hat right direction (X = joypad number, Y = button number)
joyXHatYRightDown	Gamepad hat right-down direction (X = joypad number, Y = button number)
JoyXAxis0+	First POV stick, first axis positive direction (X = joypad number)
JoyXAxis0-	First POV stick, first axis negative direction (X = joypad number)
JoyXAxis1+	First POV stick, second axis positive direction (X = joypad number)
JoyXAxis1-	First POV stick, second axis negative direction (X = joypad number)
JoyXAxis2+	Second POV stick, second axis positive direction (X = joypad number)
JoyXAxis2-	Second POV stick, first axis negative direction (X = joypad number)
JoyXAxis3+	Second POV stick, second axis positive direction (X = joypad number)
JoyXAxis3-	Second POV stick, second axis negative direction (X = joypad number)

RetroFE supports the following mouse codes:

Keycode	Description
mouseButtonLeft	The left mouse button.
mouseButtonMiddle	The middle mouse button.
mouseButtonRight	The right mouse button.
mouseButtonX1	The X1 mouse button.
mouseButtonX2	The X2 mouse button.

RetroFE supports the keyboard codes from the [SDL keyboard codes list](#).

7 Collection

A RetroFE collection is a set of items sharing a common identity and launcher (e.g. a console type like SNES). It generally consists of a settings.conf file, roms, and art.

7.1 settings.conf

The collection's settings.conf file supports the following settings. The default value is used when the variable is not set in the settings.conf file. Keep in mind that Windows uses \ in stead of /.

Variable	Default value	Description
list.path	/collections/<collection name>/roms	Location of your ROMS.
list.includeMissingItems	false	Include items defined in the include.txt, even if the ROM does not exist.
list.extensions		Comma separated values of ROM extensions to include in the list (i.e. zip, nes).
list.menuSort	false	Defines whether the items in a collection will be sorted alphabetically.
launcher		Specifies which launcher to use when a game is selected from the menu: It uses launchers/<launcher name>.conf.
metadata.type	<collection name>	Collection name to reference in the meta.db database to request information like manufacturer, year, genre, etc.
manufacturer		Name of the collection's manufacturer to be displayed by the theme, e.g. Nintendo
year		Year the collection's system came out, e.g. 1985 for the NES, to be displayed by the theme.
genre		Type fo collection, e.g. Arcade, Console, to be displayed by the theme.
media.screenshot	collections/<collection name>/medium_artwork/screenshot	Screen shot to be displayed by the theme.

Variable	Default value	Description
media.screentitle	collections/<collection name>/medium_artwork/screentitle	Title shot to be displayed by the theme.
media.artwork_back	collections/<collection name>/medium_artwork/artwork_back	Back of the box to be displayed by the theme.
media.artwork_front	collections/<collection name>/medium_artwork/artwork_front	Front of the box to be displayed by the theme.
media.logo	collections/<collection name>/medium_artwork/logo	Wheel logo to be displayed by the theme.
media.medium_back	collections/<collection name>/medium_artwork/medium_back	Back of the cartridge/disk to be displayed by the theme.
media.medium_front	collections/<collection name>/medium_artwork/medium_front	Front of the cartridge/disk to be displayed by the theme.
media.video	collections/<collection name>/medium_artwork/video	Video to be displayed by the theme.
media.system_artwork_back	collections/<collection name>/system_artwork	Directory containing the system artwork of a collection.

All of these settings can take advantage of the following variables:

Variable	Description
%BASE_MEDIA_PATH%	The root directory where the artwork resides, defined by the global settings.conf.
%BASE_ITEM_PATH%	The root directory where the roms reside, defined by the global settings.conf.
%ITEM_COLLECTION_NAME%	The name of the current collection.

8 launcher

A launcher config file describes how to launch a program (i.e. emulator, application, or game) when a launchable menu item is selected.

See below for a list of supported configuration properties.

Property	Description
executable	Path of where the executable exists.
arguments	Arguments to pass when executing the launcher (i.e. ROM name).

Windows example:

```
executable = D:\Emulators\Nestopia\nestopia.exe
arguments  = "%ITEM_FILEPATH%"
```

%ITEM_FILEPATH% is a reserved variable name. See the variables table below for other variables that may be used. Also note the quotes around "%ITEM_FILEPATH%" to help not confuse the executable from confusing an item with spaces as multiple arguments.

Assuming that "Super Mario Bros" was the selected item, the RetroFE will attempt to execute:

```
"D:\Emulators\Nestopia\nestopia.exe" "D:\ROMs\Nintendo\Super Mario Bros.nes".
```

You can also use relative paths (relative to the root folder of RetroFE):

```
executable = ..\Emulators\Nestopia\nestopia.exe
arguments  = "%ITEM_FILEPATH%"
```

The following reserved variable names are available:

Variable	Description	Example
%ITEM_FILEPATH%	Full item path	D:\ROMs\Nintendo\Super Mario Bros.nes
%ITEM_NAME%	The item name	Super Mario Bros
%ITEM_FILENAME%	Filename without path	Super Mario Bros.nes
%ITEM_DIRECTORY%	Folder where file exists	D:/ROMs/Nintendo
%ITEM_COLLECTION_NAME%	Name of collection for item	Nintendo Entertainment System
%RETROFE_PATH%	Folder location of Frontend	D:\Frontends\RetroFE
%RETROFE_EXEC_PATH%	Location of RetroFE	D:\Frontends\RetroFE\RetroFE.exe

A more elaborate Windows example:

```
# Have fceux load a save state automatically for the ROM
executable = D:\Emulators\fceux\fceux.exe
arguments = "%ITEM_FILEPATH%" -loadstate "%ITEM_DIRECTORY%\
%ITEM_NAME%.fcs"
```

9 layout

9.1 General layout information

Layouts define the look and feel for your frontend. Each layout is a folder containing at least the following:

File/Directory	Description
./layouts/<layout name>/	Folder containing the theme files.
./layouts/<layout name>/splash.xml	Splash screen layout configuration file.
./layouts/<layout name>/layout.xml	Main screen layout configuration file.

The [global settings.conf](#) file contains the default layout used by RetroFE. It is however possible to give a collection a completely different layout by adding the layout.xml and layout artwork to the ./layouts/<layout name>/collections/<collection name>/layout/ directory. A splash.xml is not required/used for a collection's layout. This way you can give each collection a completely different look and feel.

9.2 General layout structure

A layout generally consists of the following structure:

```
<layout>

  <!-- Sound parameters -->
  <sound/>

  <!-- Menu structure -->
  <menu/>

  <!-- Video -->
  <video/>

  <!-- Image -->
  <image/>

  <!-- Text -->
  <text/>

  <!-- Video connected to the selected item -->
  <reloadableVideo/>

  <!-- Image connected to the selected item -->
  <reloadableImage/>

  <!-- Text connected to the selected item -->
  <reloadableText/>

  <!-- Scrolling text connected to the selected item -->
  <reloadableScrollingText/>

</layout>
```

9.3 <layout> attributes

The <layout> tag can use the following attributes:

Attribute	Description
width	The virtual width to use for this layout. This will be scaled automatically by the frontend if the screen resolution is different.
height	The virtual height to use for this layout. This will be scaled automatically by the frontend if the screen resolution is different.
loadFontSize	The size (quality) of the font to load. Lower font sizes is more blurred, Higher font sizes are a little more pixelated. 24 is an ideal value.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
minShowTime	Minimum amount of time (in seconds) to show the current layout (only applicable in splash.xml).

Example:

```
<layout width="1920" height="1080" font="Roboto-Bold.ttf"
loadFontSize="64" fontColor="dedede">
```

9.4 <sound> attributes

The <sound> tag can use the following attributes:

Attribute	Description
type	triggers on the following events: "load" (page load), "unload" (page exit), "highlight" (scroll), "select" (entering game/sub-menu).
src	Location of the sound file (relative to the layout folder).

Example:

```
<sound type="load" src="load.wav"/>
```

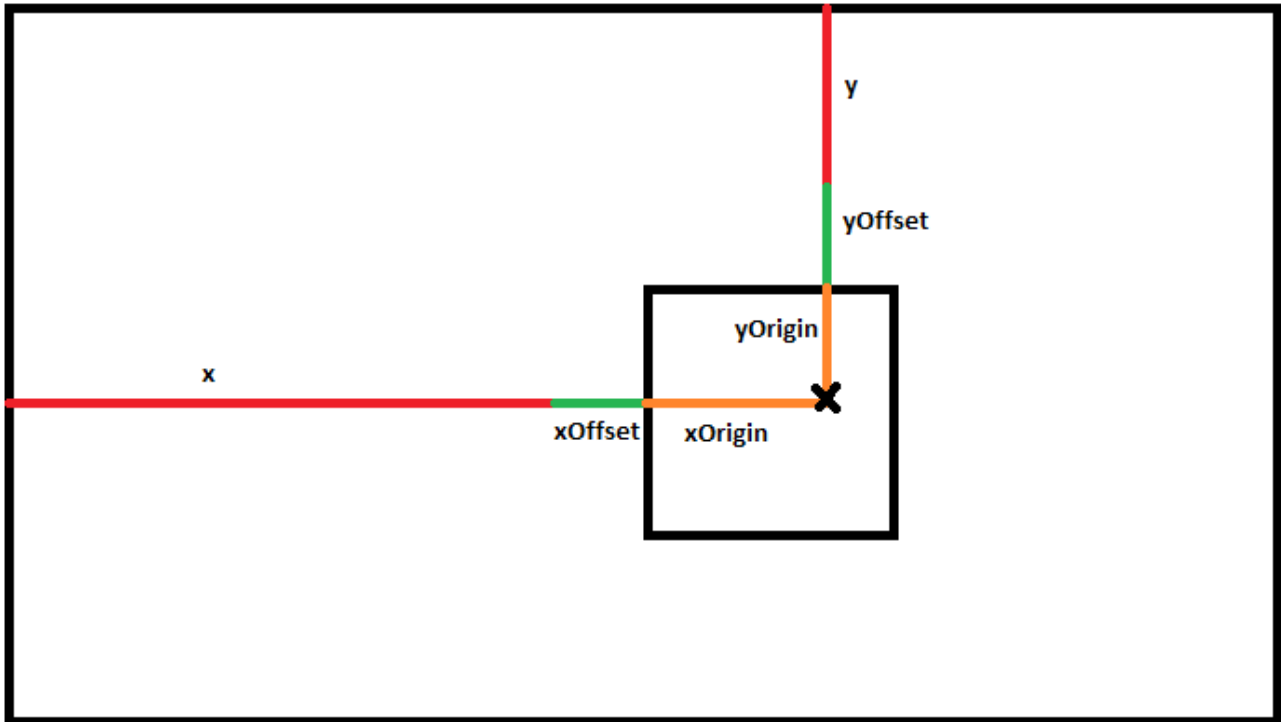
9.5 General attributes

The other tags all support a general set of attributes:

Attribute	Description
x	X coordinate of where to place the component.
y	Y coordinate of where to place the component.
xOffset	Relative X offset of how many pixels to shift the object from x (x + xOffset).
yOffset	Relative Y offset of how many pixels to shift the object from y (y + yOffset).
xOrigin	X offset on image to use as the pin point for placement. Supports the special settings "left", "center", and "right".
yOrigin	Y offset on image to use as the pin point for placement. Supports the special settings "top", "center", and "bottom".
alpha	0 = 0% visible, 0.5=50% visible, 1=100% visible.
angle	Angle to rotate image, in degrees.
width	Width of the component. Image will be scaled appropriately if not specified. Supports the special settings "stretch".
height	Height of the component. Image will be scaled appropriately if not specified. Supports the special settings "stretch".
minWidth	Minimum width to allow the image to be (if scaling is needed).
minHeight	Minimum height to allow the image to be (if scaling is needed).
maxWidth	Maximum width to allow the image to be (if scaling is needed).
maxHeight	Maximum height to allow the image to be (if scaling is needed).
backgroundColor	Fill the component with a background color.
backgroundAlpha	Background color transparency: 0 = 0% visible, 0.5=50% visible, 1=100% visible.
reflection	Location of the reflection compared to the image: top, bottom, left, right.
reflectionDistance	Distance between the reflection and the original image in pixels.
reflectionScale	Scale in vertical (reflection top/bottom) or horizontal (reflection left/right) direction of the reflection compared to the original image.
reflectionAlpha	Transparency of the reflection compared to the original image: 0 = 0% visible, 0.5=50% visible, 1=100% visible.
containerX	X coordinate of the cropping container.
containerY	Y coordinate of the cropping container.
containerWidth	Width of the cropping container.
ContainerHeight	Height of the cropping container.
layer	Layer on which the item should be displayed: 0 = back, 19 = front.

RetroFE supports 20 layers of images (0-19) to allow items to overlap in a controlled fashion.

Usage of the x/y/xOffset/yOffset/xOrigin/yOrigin parameters:



Usage of the xOffset/yOffset parameters is generally reserved for when you use values like "top|bottom|left|right|center" for x and/or y, e.g. x="center" xOffset="10" will place the object 10 pixels right of the center.

All these parameters can also be used in animations to create an even more beautiful layout experience. These animations use the following syntax:

```
<action type>
  <set duration="<time in seconds>"
    <animate type="<animation parameter>" from="<from setting>"
to="<to setting>" algorithm="<algorithm_setting>"/>
  </set>
</on<action_type>
```

It is possible to put multiple animates in the same set; these animations will happen at the same time. It is possible to put multiple sets in the same action type: these animations will happen in sequence. It is possible to omit the from attribute; this will animate from the current value. It is possible to use type="nop" (no operation), without any other attributes (<animate type="nop"/>. This can e.g. be used to delay an animation by preceding it with a set containing just a "nop" animation.

RetroFE supports the following action types:

Action type	Description
onEnter	Action happens on entering RetroFE.
onExit	Action happens on exiting RetroFE.
onIdle	Action happens while no keys are being pressed.
onMenuScroll	Action happens when the menu starts scrolling.
onHighlightEnter	Action happens when the item is being selected.
onHighlightExit	Action happens when item is no longer selected.
onMenuEnter	Action happens on entering a menu.
onMenuExit	Action happens on exiting a menu.

RetroFE supports the following algorithms:

easeInquadratic	easeOutquadratic	easeInoutquadratic	easeIncubic	easeOutcubic
easeInoutcubic	easeInquartic	easeOutquartic	easeInoutquartic	easeInquintic
easeOutquintic	easeInoutquintic	easeInsine	easeOutsine	easeInoutsine
easeInexponential	easeOutexponential	easeInoutexponential	easeIncircular	easeOutcircular
easeInoutcircular	linear			

Example:

```
<!-- Flaming background, slowly pulsating using transparency. -->
<image src="bg.png" x="0" y="0" height="stretch" width="stretch"
layer="0">
  <onIdle>
    <set duration="2">
      <animate type="alpha" from="1" to="0.2"
algorithm="easeinquadric"/>
    </set>
    <set duration="1">
      <animate type="alpha" from="0.2" to="1"
algorithm="easeinquadric"/>
    </set>
  </onIdle>
</image>
```

9.6 <menu> attributes

The <menu> tag is used to define a menu structure, and uses the following structure:

```
<menu>
  <itemDefaults/>
  <item/>
</menu>
```

The <menu> tag supports the following attributes:

Attribute	Description
type	The type of menu to display. Set to custom to specify all the points on the screen. Set to vertical to have a vertical scrolling list prebuilt for the layout.
imageType	If specified, uses an image to be displayed for each menu item (if it exists). Text will be used if the image could not be found.
orientation	Set to "horizontal" to use the Left/Right controls to scroll. set to "vertical" to use Up/Down.
scrollTime	The amount of time (in seconds) it takes for an item to scroll to the next point on the menu (i.e 0.750 = 750 milliseconds).
scrollAcceleration	The acceleration rate to scroll by when holding down the up/down/left/right scroll key.

The <itemDefaults> tag supports the following attributes:

Attribute	Description
spacing	Used when the menu type is set to vertical . Defines the spacing in pixels for all edges of a menu item.
index	Used when the menu type is set to vertical . Specify a options for a particular menu item when in list mode (first=first visible item, last=last visible item, start=first - 1, end - last + 1)
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.

The <item> tag supports the following attributes:

Attribute	Description
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.

9.7 <video> attributes

The <video> tag supports the following attributes:

Attribute	Description
src	The location of the image to display (relative to the location of the active layout folder).
numLoops	The number of loops the video will play before stopping (default: 1, use 0 for infinite looping).

Example:

```
<video src="bg.png" numLoops="0" x="0" y="0" height="stretch"
width="stretch" layer="0"/>
```

9.8 <image> attributes

The <image> tag supports the following attribute:

Attribute	Description
src	The location of the image to display (relative to the location of the active layout folder).

Example:

```
<image src="bg.png" x="0" y="0" height="stretch" width="stretch"
layer="0"/>
```

9.9 <text> attributes

The <text> tag supports the following attributes:

Attribute	Description
value	The text message to display.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.

Example:

```
<text value="Year:" x="300" y="850" xOrigin="right" yOrigin="top"
fontSize="48" layer="7"/>
```

9.10 <reloadableVideo> attributes

The <reloadableVideo> tag can be used to display a video of the selected item, and supports the following attributes:

Attribute	Description
imageType	The type of image to display if a video cannot be found: "numberButtons", "numberPlayers", "year", "title", "manufacturer", "genre".
mode	See mode attribute for more details.
textFallback	Set to true to have text displayed of a the item title if an image cannot be loaded.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.
backgroundColor	Fill the component with a background color.
backgroundAlpha	Background color transparency: 0 = 0% visible, 0.5=50% visible, 1=100% visible.

Example:

```
<reloadableVideo imageType="screenshot" x="400" y="300"
xOrigin="center" yOrigin="center" height="480" maxWidth="640"
layer="3"/>
```

9.11 <reloadableImage> attributes

The <reloadableImage> tag can be used to display an image of the selected item, and supports the following attributes:

Attribute	Description
type	The type of image to display: "numberButtons", "numberPlayers", "year", "title", "manufacturer", "genre".
mode	See mode attribute for more details.
textFallback	Set to true to have text displayed of a the item title if an image cannot be loaded.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.

Example:

```
<reloadableImage type="logo" mode="system" x="1350" y="50"
xOrigin="center" yOrigin="top" height="250" maxWidth="1100"
layer="7"/>
```

9.12 <reloadableText> attributes

The <reloadableText> tag can be used to display textual information about the selected item, and supports the following attributes:

Attribute	Description
type	The type of text to display: "numberButtons", "numberPlayers", "year", "title", "manufacturer", "genre", "playlist", "collectionName", "collectionSize", "collectionIndex".
mode	See mode attribute for more details.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.

Example:

```
<reloadableText type="year" x="310" y="850" xOrigin="left"
yOrigin="top" fontSize="48" maxWidth="300" layer="7"/>
```

9.13 <reloadableScrollingText> attributes

Attribute	Description
type	The type of text file to load. Files are generally placed as <collection name>/medium_artwork/<type>/<game name>.txt or <collection name>/system_artwork/<type>.txt.
mode	See mode attribute for more details.
font	Location of the font (relative to the layout folder).
fontColor	Default RGB color of the font (in hex, i.e. "6699AA").
fontSize	Default font size of to display if not specified by a component.
loadFontSize	The size (quality) of the font to load. For the best results, use the same value as fontsize.
direction	The direction in which the text should scroll: horizontal, vertical.
scrollingSpeed	The speed at which the text should scroll in pixels per second.
startPosition	The x/y offset of the start position of the text.
startTime	The delay before the text starts scrolling.
endTime	The delay between the finish of the scroll, and restarting the display of the text.
alignment	The text alignment for vertical scrolls: left, right, centered, justified.

The text will scroll across the screen until it is no longer visible. Then it will be reset. For a vertical scroll: if the startPosition is 0, and the text fits within the defined size, the text will not scroll.

Example:

```
<reloadableScrollingText type="story" alpha="0" x="145" y="355"
xOrigin="left" yOrigin="top" width="940" height="215"
font="MC360.ttf" fontSize="35" fontColor="f2f2f2"
direction="vertical" alignment="justified" scrollingSpeed="15"
startPosition="0" startTime="4" endTime="1" alpha="1" layer="7"/>
```

9.14 mode attribute

Several parameters allow for a mode attribute: system, common, layout, systemlayout, and commonlayout. This attribute can be omitted. This attribute has the following effects:

- Mode system and systemlayout use the information from the collection you're in rather than the selected item.
- Mode layout, systemlayout, and commonlayout use the art in the layouts/<layout name>/collections/<collection name>/ directory rather than collections/<collection name>/ directory.
- Mode common and commonlayout use the art in the collections/_common/medium_artwork directory rather than the collections/<collection name>/medium_artwork directory.

The item path also depends on whether the selected item is a collection or a game; it will select the first found file.

Example:

You're in the Main collection, have the SNES collection selected, and are displaying a reloadableImage of type "logo":

```
No mode used:
  collections/Main/medium_artwork/logo/SNES.png
  collections/Main/medium_artwork/logo/default.png
Mode layout:
  layouts/<layout
name>/collections/Main/medium_artwork/logo/SNES.png
  layouts/<layout
name>/collections/Main/medium_artwork/logo/default.png
Mode system (not very useful in the main menu):
  collections/Main/medium_artwork/logo/Main.png
  collections/Main/system_artwork/logo.png
  collections/Main/medium_artwork/logo/default.png
Mode systemlayout (not very useful in the main menu):
  layouts/<layout
name>/collections/Main/medium_artwork/logo/Main.png
  layouts/<layout name>/collections/Main/system_artwork/logo.png
  layouts/<layout name>/collections/Main/system_artwork/default.png
Mode common (not very useful for type logo):
  collections/_common/medium_artwork/logo/SNES.png
  collections/_common/medium_artwork/logo/default.png
Mode commonlayout (not very useful for type logo):
  layouts/<layout
name>/collections/_common/medium_artwork/logo/SNES.png
  layouts/<layout
name>/collections/_common/medium_artwork/logo/default.png
```

Example:

You're in the SNES collection, have the 1942 game selected, and are displaying a reloadableImage of type "logo":

```
No mode used:
  collections/SNES/medium_artwork/logo/1942.png
  collections/SNES/medium_artwork/logo/default.png
Mode layout:
  layouts/<layout
name>/collections/SNES/medium_artwork/logo/1942.png
  layouts/<layout
name>/collections/SNES/medium_artwork/logo/default.png
Mode system:
  collections/SNES/system_artwork/logo.png
Mode systemlayout:
  layouts/<layout name>/collections/SNES/system_artwork/logo.png
Mode common (not very useful for type logo):
  collections/_common/medium_artwork/logo/1942.png
  collections/_common/medium_artwork/logo/default.png
Mode commonlayout (not very useful for type logo):
  layouts/<layout
name>/collections/_common/medium_artwork/logo/1942.png
  layouts/<layout
name>/collections/_common/medium_artwork/logo/default.png
```

Example:

You're in the SNES collection by Nintendo, have the 1942 game by Capcom selected, and are displaying a reloadableImage of type "manufacturer":

```
No mode used:
  collections/SNES/medium_artwork/manufacturer/Capcom.png
  collections/SNES/medium_artwork/manufacturer/default.png
Mode layout:
  layouts/<layout
name>/collections/SNES/medium_artwork/manufacturer/Capcom.png
  layouts/<layout
name>/collections/SNES/medium_artwork/manufacturer/default.png
Mode system:
  collections/SNES/system_artwork/Nintendo.png
Mode systemlayout:
  layouts/<layout
name>/collections/SNES/system_artwork/Nintendo.png
Mode common:
  collections/_common/medium_artwork/manufacturer/Capcom.png
  collections/_common/medium_artwork/manufacturer/default.png
Mode commonlayout:
  layouts/<layout
name>/collections/_common/medium_artwork/manufacturer/Capcom.png
  layouts/<layout
name>/collections/_common/medium_artwork/manufacturer/default.png
```