

Write Themes For EmulationStation

Batocera offers many options for EmulationStation theming. A lot of features have been added to Batocera 5.24, and a lot new more are added in Batocera 29th release (BATOCERA 29 below).

The page here gives a lot of information, but does not necessarily keep up with the very latest possibilities offered by the Batocera version of EmulationStation. **To make sure you are up to date with the latest and greatest**, you should [check the documentation page on GitHub](#).

If you'd like to work with Batocera's Carbon theme, feel free to download it from <https://github.com/fabricecaruso/es-theme-carbon> (click the green "Code" button and download as a ZIP).

Themes

EmulationStation allows each system to have its own "theme." A theme is a collection **views** that define some **elements**, each with their own **properties**.

The first place ES will check for a theme is in the system's <path> folder, for a theme.xml file: *
[SYSTEM_PATH]/theme.xml

If that file doesn't exist, ES will try to find the theme in the current **theme set**. Theme sets are just a collection of individual system themes arranged in the "themes" folder under some name. A theme set can provide a default theme that will be used if there is no matching system theme. Here are examples :

```
...
  themes/
    my_theme_set/
      snes/
        theme.xml
        my_cool_background.jpg
      nes/
        theme.xml
        my_other_super_cool_background.jpg
      common_resources/
        scroll_sound.wav

        theme.xml (Default theme)
...
  another_theme_set/
    resources/
      snes-logo.svg
      snes-back.jpg
      nes-logo.svg
      nes-back.jpg
      theme.xml (Default theme)
```

The theme set system makes it easy for users to try different themes and allows distributions to include multiple theme options. Users can change the currently active theme set in the “UI Settings” menu. The option is only visible if at least one theme set exists.

There are places where ES can load theme sets from:

Windows :

- [HOME]/.emulationstation/themes/[CURRENT_THEME_SET]/[SYSTEM_THEME]/theme.xml
- [HOME]/.emulationstation/themes/[CURRENT_THEME_SET]/theme.xml

Linux :

- /userdata/themes/[CURRENT_THEME_SET]/[SYSTEM_THEME]/theme.xml
- /userdata/themes/[CURRENT_THEME_SET]/theme.xml

[SYSTEM_THEME] is the <theme> tag for the system, as defined in es_systems.cfg. If the <theme> tag is not set, ES will use the system’s <name>.

If both files happen to exist, ES will pick the first one (the one located in the home directory).

Again, the [CURRENT_THEME_SET] value is set in the “UI Settings” menu. If it has not been set yet or the previously selected theme set is missing, the first available theme set will be used as the default.

Simple Example

Here is a very simple theme that changes the description text’s color:

```
<theme>
  <formatVersion>7</formatVersion>
  <view name="detailed">
    <text name="description">
      <color>00FF00</color>
    </text>
    <image name="my_image" extra="true">
      <pos>0.5 0.5</pos>
      <origin>0.5 0.5</origin>
      <size>0.8 0.8</size>
      <path>./my_art/my_awesome_image.jpg</path>
    </image>
  </view>
</theme>
```

How it works

Everything must be inside a `<theme>` tag.

The `<formatVersion>` tag *must be specified*. This is the version of the theming system the theme was designed for.

The current version is 7 for batocera-emulationstation.

Version 4 themes are RetroPie & Recalbox themes - see their documentations for version 4 themes.

A *view* can be thought of as a particular “display” within EmulationStation. Views are defined like this:

```
<view name="ViewNameHere">
    ... define elements here ...
</view>
```

An *element* is a particular visual element, such as an image or a piece of text. You can either modify an element that already exists for a particular view (as is done in the “description” example), like this:

```
<elementTypeHere name="ExistingElementNameHere">
    ... define properties here ...
</elementTypeHere>
```

Or, you can create your own elements by adding `extra="true"` (as is done in the “my_image” example) like this:

```
<elementTypeHere name="YourUniqueElementNameHere" extra="true">
    ... define properties here ...
</elementTypeHere>
```

“Extra” elements will be drawn in the order they are defined (so define backgrounds first!). When they get drawn relative to the pre-existing elements depends on the view. Make sure “extra” element names do not clash with existing element names! An easy way to protect against this is to just start all your extra element names with some prefix like “e_”.

Properties control how a particular *element* looks - for example, its position, size, image path, etc. The type of the property determines what kinds of values you can use. You can read about the types below in the “Reference” section. Properties are defined like this:

```
<propertyNameHere>ValueHere</propertyNameHere>
```

**** BATOCERA 5.24 :**

A *customView* can be thought of as a custom “gamelist view” within EmulationStation. Custom Views must inherit one of the standard views (basic, detailed, video or grid) or inherit from another custom view. Custom view inherit every element, attribute and behaviour of its parent view, so only differences or changes are needed.

It is defined like this :

```
<customView name="ViewNameHere" inherits="BaseViewNameHere">
    ... define elements here ...
</customView>
```

Advanced Features

It is recommended that if you are writing a theme you launch EmulationStation with the `--windowed` switches. This way you can read error messages without having to check the log file. You can also reload the current gamelist view and system view with F5

The `<include>` tag

You can include theme files within theme files, similar to `#include` in C (though the internal mechanism is different, the effect is the same). Example:

`~/.emulationstation/all_themes.xml:`

```
<theme>
  <formatVersion>4</formatVersion>
  <view name="detailed">
    <text name="description">
      <fontPath>./all_themes/myfont.ttf</fontPath>
      <color>00FF00</color>
    </text>
  </view>
</theme>
```

`~/.emulationstation/snes/theme.xml:`

```
<theme>
  <formatVersion>4</formatVersion>
  <include>../all_themes.xml</include>
  <view name="detailed">
    <text name="description">
      <color>FF0000</color>
    </text>
  </view>
</theme>
```

Is equivalent to this `snes/theme.xml`:

```
<theme>
  <formatVersion>4</formatVersion>
  <view name="detailed">
    <text name="description">
      <fontPath>./all_themes/myfont.ttf</fontPath>
```

```

        <color>FF0000</color>
    </text>
</view>
</theme>

```

Notice that properties that were not specified got merged (<fontPath>) and the snes/theme.xml could overwrite the included files' values (<color>). Also notice the included file still needed the <formatVersion> tag.

Subsets : Create theme variations declaring subsets

** Batocera 5.24

Subset are usefull to create multiple variations in a theme (colorsets, iconsets, different system views...)

When a subset is defined, you can choose one of the named elements in the subset. Only one is active at a time. Only the active item is included, the other includes of the same subset are ignored.

"name" attribute must be unique among all includes.

Common subsets are "colorset", "iconset", "menu", "systemview" or "gamelistview", but you can also add your custom subset names.

```

<theme>
    <include subset="colorset" name="nice
colors">./../nice_colors.xml</include>
    <include subset="colorset" name="other colors"
>./../other_colors.xml</include>

    <include subset="systemview" name="nice
systemview">./../nice_sys.xml</include>
    <include subset="systemview" name="new carousel"
>./../carousel.xml</include>

    <include subset="mycustomsubset" name="my
choice">./../choice1.xml</include>
    <include subset="mycustomsubset" name="my other choice"
>./../choice2.xml</include>
</theme>

```

Theming multiple views simultaneously

Sometimes you want to apply the same properties to the same elements across multiple views. The name attribute actually works as a list (delimited by any characters of \t\r\n , - that is, whitespace and commas). So, for example, to easily apply the same header to the basic, grid, and system views:

```

<view name="basic, grid, system">
    <image name="logo">

```

```
<path>./snes_art/snes_header.png</path>
</image>
</view>
<view name="detailed">
  <image name="logo">
    <path>./snes_art/snes_header_detailed.png</path>
  </image>
</view>
```

This is equivalent to:

```
<view name="basic">
  <image name="logo">
    <path>./snes_art/snes_header.png</path>
  </image>
</view>
<view name="detailed">
  <image name="logo">
    <path>./snes_art/snes_header_detailed.png</path>
  </image>
</view>
<view name="grid">
  <image name="logo">
    <path>./snes_art/snes_header.png</path>
  </image>
</view>
<view name="system">
  <image name="logo">
    <path>./snes_art/snes_header.png</path>
  </image>
</view>
```

Theming multiple elements simultaneously

You can theme multiple elements *of the same type* simultaneously. The name attribute actually works as a list (delimited by any characters of `\t\r\n` , - that is, whitespace and commas), just like it does for views, as long as the elements have the same type. This is useful if you want to, say, apply the same color to all the metadata labels:

```
<theme>
  <formatVersion>3</formatVersion>
  <view name="detailed">
    <!-- Weird spaces/newline on purpose! -->
    <text name="md_lbl_rating, md_lbl_releasedate, md_lbl_developer,
md_lbl_publisher, md_lbl_genre, md_lbl_players, md_lbl_lastplayed,
md_lbl_playcount">
      <color>48474D</color>
    </text>
  </view>
```

```
</theme>
```

Which is equivalent to:

```
<theme>
  <formatVersion>3</formatVersion>
  <view name="detailed">
    <text name="md_lbl_rating">
      <color>48474D</color>
    </text>
    <text name="md_lbl_releasedate">
      <color>48474D</color>
    </text>
    <text name="md_lbl_developer">
      <color>48474D</color>
    </text>
    <text name="md_lbl_publisher">
      <color>48474D</color>
    </text>
    <text name="md_lbl_genre">
      <color>48474D</color>
    </text>
    <text name="md_lbl_players">
      <color>48474D</color>
    </text>
    <text name="md_lbl_lastplayed">
      <color>48474D</color>
    </text>
    <text name="md_lbl_playcount">
      <color>48474D</color>
    </text>
    <text name="md_lbl_gametime">
      <color>48474D</color>
    </text>
  </view>
</theme>
```

Just remember, *this only works if the elements have the same type!*

Element rendering order with z-index

You can now change the order in which elements are rendered by setting `zIndex` values. Default values correspond to the default rendering order while allowing elements to easily be shifted without having to set `zIndex` values for every element. Elements will be rendered in order from smallest `z-index` to largest.

Defaults

screen

** Batocera 5.24

- text name="clock" - 10 (customize the clock)
- controllerActivity name="controllerActivity" - 10 (customize the connected joystick indicators
- Extra Elements extra="true" - 10

system

- Extra Elements extra="true" - 10
- carousel name="systemcarousel" - 40
- text name="systemInfo" - 50

basic, detailed, grid, video

- image name="background" - 0
- Extra Elements extra="true" - 10
- textlist name="gamelist" - 20
- imagegrid name="gamegrid" - 20
- Media
 - image name="md_image" - 30
 - video name="md_video" - 31
 - image name="md_marquee" - 35
- Metadata - 40
 - Labels
 - text name="md_lbl_rating"
 - text name="md_lbl_releasedate"
 - text name="md_lbl_developer"
 - text name="md_lbl_publisher"
 - text name="md_lbl_genre"
 - text name="md_lbl_players"
 - text name="md_lbl_lastplayed"
 - text name="md_lbl_playcount"
 - text name="md_lbl_gametime"
 - Values
 - rating name="md_rating"
 - datetime name="md_releasedate"
 - text name="md_developer"
 - text name="md_publisher"
 - text name="md_genre"
 - text name="md_players"
 - datetime name="md_lastplayed"
 - text name="md_playcount"
 - text name="md_gametime"
 - text name="md_description"
 - text name="md_name"

- System Logo/Text - 50
 - text name="logoText"
 - image name="logo"

Theme variables

Theme variables can be used to simplify theme construction. There are 2 types of variables available.

* System Variables * Theme Defined Variables

System Variables

System variables are system specific and are derived from the values in es_systems.cfg. *

system.name * system.fullName * system.theme * lang ** Batocera 5.24

Theme Defined Variables

Variables can also be defined in the theme.

```
<variables>
  <themeColor>8b0000</themeColor>
</variables>
```

Usage in themes

Variables can be used to specify the value of a theme property:

```
<color>${themeColor}</color>
```

or to specify only a portion of the value of a theme property:

```
<color>${themeColor}c0</color>
<path>./art/logo/${system.theme}.svg</path>
<path>./art/logo/${system.theme}-${lang}.svg</path> <!-- use it to select a
localized image -->
```

Filter using attributes

** Batocera 5.24

System attributes allow filtering elements and adapt display under conditions :

These attributes apply to every XML element in the theme.

- tinyScreen - type : BOOLEAN - Allow elements to be active only with small screens like GPI Case (if true), or disable element with normal screens (if false)

- `ifHelpPrompts`- type : BOOLEAN - Allow elements to be active only if help is visible/invisible in ES menus.
- `lang` - type : STRING - Allow elements to be used only if the lang is the current language in EmulationStation. lang is 2 lower characters. (fr, br, en, ru, pt....)
- `ifSubset` - type : STRING - BATOCERA 29 : Allow filtering elements in a xml file when subsets are active.
- `xml <image ifSubset="systemview:horizontal|transparent|legacy, iconview=standard">` - name of the subset followed with : and the name value in the subset. To test multiple values, split with a comma |. To test multiple subsets, split with a comma ,

Usage in themes

Variables can be used to specify the value of a theme property:

```
<!-- Include GPICase settings -->
<include tinyScreen="true">./gpicas.xml</include>

<!-- Change size of grid if help is hidden -->
<view name="grid" ifHelpPrompts="false">
  <imagegrid name="gamegrid">
    <size>0.950 0.84</size>
  </imagegrid>
</view>

<!-- Change text of extra item in french -->
<text name="info" extra="true">
  <text>Manufacturer</text>
  <text lang="fr">Constructeur</text>
<text>
```

or to specify only a portion of the value of a theme property:

Reference

Views, their elements, and themable properties:

basic

- `helpsystem name="help"` - ALL
 - The help system style for this view.
- `image name="background"` - ALL
 - This is a background image that exists for convenience. It goes from (0, 0) to (1, 1).
- `text name="logoText"` - ALL
 - Displays the name of the system. Only present if no "logo" image is specified. Displayed

at the top of the screen, centered by default.

- `image name="logo"` - ALL
 - A header image. If a non-empty path is specified, `text name="logoText"` will be hidden and this image will be, by default, displayed roughly in its place.
 - `textlist name="gamelist"` - ALL
 - The gamelist. `primaryColor` is for games, `secondaryColor` is for folders. Centered by default.
-

detailed

- `helpsystem name="help"` - ALL
 - The help system style for this view.
- `image name="background"` - ALL
 - This is a background image that exists for convenience. It goes from (0, 0) to (1, 1).
- `text name="logoText"` - ALL
 - Displays the name of the system. Only present if no "logo" image is specified. Displayed at the top of the screen, centered by default.
- `image name="logo"` - ALL
 - A header image. If a non-empty path is specified, `text name="logoText"` will be hidden and this image will be, by default, displayed roughly in its place.
- `textlist name="gamelist"` - ALL
- The gamelist. `primaryColor` is for games, `secondaryColor` is for folders. Left aligned by default.
- Metadata
 - Labels
 - `text name="md_lbl_rating"` - ALL
 - `text name="md_lbl_releasedate"` - ALL
 - `text name="md_lbl_developer"` - ALL
 - `text name="md_lbl_publisher"` - ALL
 - `text name="md_lbl_genre"` - ALL
 - `text name="md_lbl_players"` - ALL
 - `text name="md_lbl_lastplayed"` - ALL
 - `text name="md_lbl_playcount"` - ALL
 - `text name="md_lbl_gametime"` - ALL
- Values * All values will follow to the right of their labels if a position isn't specified.
 - `image name="md_image"` - POSITION | SIZE | Z_INDEX - Path is the "image" metadata for the currently selected game.
 - `video name="md_video"` - POSITION | SIZE | Z_INDEX
 - BATOCERA 5.24 : Path is the "video" metadata for the currently selected game.
- `image name="md_marquee"` - POSITION | SIZE | Z_INDEX
 - Path is the "marquee" metadata for the currently selected game.
- `rating name="md_rating"` - ALL
 - The "rating" metadata.
- `datetime name="md_releasedate"` - ALL
 - The "releasedate" metadata.
- `text name="md_developer"` - ALL
 - The "developer" metadata.

- `text name="md_publisher" - ALL`
 - The "publisher" metadata.
- `text name="md_genre" - ALL`
 - The "genre" metadata.
- `text name="md_players" - ALL`
 - The "players" metadata (number of players the game supports).
- `datetime name="md_lastplayed" - ALL`
 - The "lastplayed" metadata. Displayed as a string representing the time relative to "now" (e.g. "3 hours ago").
- `text name="md_playcount" - ALL`
 - The "playcount" metadata (number of times the game has been played).
- `text name="md_gametime" - ALL`
 - The "gametime" metadata (how long the game has been run for).
- `text name="md_description" - POSITION | SIZE | FONT_PATH | FONT_SIZE | COLOR | Z_INDEX`
 - Text is the "desc" metadata. If no pos/size is specified, will move and resize to fit under the lowest label and reach to the bottom of the screen.
- `text name="md_name" - ALL`
 - The "name" metadata (the game name). Unlike the others metadata fields, the name is positioned offscreen by default

video

This kind of view is used for retrocompatibility with RetroPie themes, but is deprecated.

- `helpsystem name="help" - ALL`
 - The help system style for this view.
- `image name="background" - ALL`
 - This is a background image that exists for convenience. It goes from (0, 0) to (1, 1).
- `text name="logoText" - ALL`
 - Displays the name of the system. Only present if no "logo" image is specified. Displayed at the top of the screen, centered by default.
- `image name="logo" - ALL`
 - A header image. If a non-empty path is specified, `text name="logoText"` will be hidden and this image will be, by default, displayed roughly in its place.
- `textlist name="gamelist" - ALL`
 - The gamelist. `primaryColor` is for games, `secondaryColor` is for folders. Left aligned by default.
- Metadata
 - Labels
 - `text name="md_lbl_rating" - ALL`
 - `text name="md_lbl_releasedate" - ALL`
 - `text name="md_lbl_developer" - ALL`
 - `text name="md_lbl_publisher" - ALL`
 - `text name="md_lbl_genre" - ALL`
 - `text name="md_lbl_players" - ALL`
 - `text name="md_lbl_lastplayed" - ALL`
 - `text name="md_lbl_playcount" - ALL`
 - `text name="md_lbl_gametime" - ALL`

- Values

- All values will follow to the right of their labels if a position isn't specified.
- `image name="md_image" - POSITION | SIZE | Z_INDEX`
 - Path is the "image" metadata for the currently selected game.
- `image name="md_marquee" - POSITION | SIZE | Z_INDEX`
 - Path is the "marquee" metadata for the currently selected game.
- `video name="md_video" - POSITION | SIZE | Z_INDEX`
 - Path is the "video" metadata for the currently selected game.
- `rating name="md_rating" - ALL`
 - The "rating" metadata.
- `datetime name="md_releasedate" - ALL`
 - The "releasedate" metadata.
- `text name="md_developer" - ALL`
 - The "developer" metadata.
- `text name="md_publisher" - ALL`
 - The "publisher" metadata.
- `text name="md_genre" - ALL`
 - The "genre" metadata.
- `text name="md_players" - ALL`
 - The "players" metadata (number of players the game supports).
- `datetime name="md_lastplayed" - ALL`
 - The "lastplayed" metadata. Displayed as a string representing the time relative to "now" (e.g. "3 hours ago").
- `text name="md_playcount" - ALL`
 - The "playcount" metadata (number of times the game has been played).
- `text name="md_gametime" - ALL`
 - The "gametime" metadata (how long the game has been run for).
- `text name="md_description" - POSITION | SIZE | FONT_PATH | FONT_SIZE | COLOR | Z_INDEX`
 - Text is the "desc" metadata. If no pos/size is specified, will move and resize to fit under the lowest label and reach to the bottom of the screen.
- `text name="md_name" - ALL`
 - The "name" metadata (the game name). Unlike the others metadata fields, the name is positioned offscreen by default

grid

- `helpsystem name="help" - ALL`
 - The help system style for this view.
- `image name="background" - ALL`
 - This is a background image that exists for convenience. It goes from (0, 0) to (1, 1).
- `text name="logoText" - ALL`
 - Displays the name of the system. Only present if no "logo" image is specified. Displayed at the top of the screen, centered by default.
- `image name="logo" - ALL`
 - A header image. If a non-empty path is specified, `text name="logoText"` will be hidden and this image will be, by default, displayed roughly in its place.
- `imagegrid name="gamegrid" - ALL`
 - The gamegrid. The number of tile displayed is controlled by its size, margin and the

default tile max size.

- `gridtile name="default"` - ALL
 - Note that many of the default gridtile parameters change the selected gridtile parameters if they are not explicitly set by the theme. For example, changing the background image of the default gridtile also change the background image of the selected gridtile. Refer to the gridtile documentation for more informations.
- `gridtile name="selected"` - ALL
 - See default gridtile description right above.
- `text name="gridtile"` - ALL
 - BATOCERA 5.24 : Describes the text label in the grid. See default text description right above
- `text name="gridtile_selected"` - ALL
 - BATOCERA 5.24 : Describes the text label in the grid when the gridtile is selected.
- `image name="gridtile.marquee"` - ALL
 - BATOCERA 5.24 : Describes the image to use in the grid as a marquee / wheel image.
- Metadata
 - Labels
 - `text name="md_lbl_rating"` - ALL
 - `text name="md_lbl_releasedate"` - ALL
 - `text name="md_lbl_developer"` - ALL
 - `text name="md_lbl_publisher"` - ALL
 - `text name="md_lbl_genre"` - ALL
 - `text name="md_lbl_players"` - ALL
 - `text name="md_lbl_lastplayed"` - ALL
 - `text name="md_lbl_playcount"` - ALL
 - `text name="md_lbl_gametime"` - ALL
 - Values
 - All values will follow to the right of their labels if a position isn't specified.
 - `rating name="md_rating"` - ALL
 - The "rating" metadata.
 - `datetime name="md_releasedate"` - ALL
 - The "releasedate" metadata.
 - `text name="md_developer"` - ALL
 - The "developer" metadata.
 - `text name="md_publisher"` - ALL
 - The "publisher" metadata.
 - `text name="md_genre"` - ALL
 - The "genre" metadata.
 - `text name="md_players"` - ALL
 - The "players" metadata (number of players the game supports).
 - `datetime name="md_lastplayed"` - ALL
 - The "lastplayed" metadata. Displayed as a string representing the time relative to "now" (e.g. "3 hours ago").
 - `text name="md_playcount"` - ALL
 - The "playcount" metadata (number of times the game has been played).
 - `text name="md_gametime"` - ALL
 - The "gametime" metadata (how long the game has been run for).
 - `text name="md_description"` - POSITION | SIZE | FONT_PATH | FONT_SIZE | COLOR | Z_INDEX
 - Text is the "desc" metadata. If no pos/size is specified, will move and resize

to fit under the lowest label and reach to the bottom of the screen.

- `text name="md_name" - ALL`
 - The "name" metadata (the game name). Unlike the others metadata fields, the name is positioned offscreen by default

system

- `helpsystem name="help" - ALL`
 - The help system style for this view.
- `carousel name="systemcarousel" -ALL`
 - The system logo carousel
- `image name="logo" - PATH | COLOR`
 - A logo image, to be displayed in the system logo carousel.
- `text name="logoText" - FONT_PATH | COLOR | FORCE_UPPERCASE | LINE_SPACING | TEXT`
 - A logo text, to be displayed system name in the system logo carousel when no logo is available.
- `text name="systemInfo" - ALL`
 - Displays details of the system currently selected in the carousel.
- You can use extra elements (elements with `extra="true"`) to add your own backgrounds, etc. They will be displayed behind the carousel, and scroll relative to the carousel.

menu

- `helpsystem name="help" - ALL`
 - The help system style for this view. If not defined, menus will have the same `helpsystem` as defined in system view.
- `menuBackground name="menubg" - COLOR | PATH | FADEPATH`
 - The background behind menus. you can set an image and/or change color (alpha supported)
- `menuSwitch name="menuswitch" - PATHON | PATHOFF`
 - Images for the on/off switch in menus
- `menuSlider name="menuslider" - PATH`
 - Image for the slider knob in menus
- `menuButton name="menubutton" - PATH | FILLEDPATH`
 - Images for menu buttons
- `menuText name="menutext" - FONTPATH | FONTSIZE | COLOR`
 - text for all menu entries
- `menuText name="menutitle" - FONTPATH | FONTSIZE | COLOR`
 - text for menu titles
- `menuText name="menufooter" - FONTPATH | FONTSIZE | COLOR`
 - text for menu footers or subtitles
- `menuTextSmall name="menutextsmall" - FONTPATH | FONTSIZE | COLOR`
 - text for menu entries in smallerfont

menu is used to theme `helpsystem` and ES menus.

screen

**** BATOCERA 5.24**

screen is a fixed view and has no themable properties.

It is used to work like an OSD, to display elements like clock and controllers activity.

It can also contain text, image and video extras.

Types of properties:

- NORMALIZED_PAIR - two decimals, in the range [0..1], delimited by a space. For example, 0.25 0.5. Most commonly used for position (x and y coordinates) and size (width and height).
- NORMALIZED_RECT - four decimals, in the range [0..1], delimited by a space. For example, 0.25 0.5 0.10 0.30. Most commonly used for padding to store top, left, bottom and right coordinates. **** BATOCERA 5.24**
- PATH - a path. If the first character is a ~, it will be expanded into the environment variable for the home path (\$HOME for Linux or %HOMEPATH% for Windows). If the first character is a ., it will be expanded to the theme file's directory, allowing you to specify resources relative to the theme file, like so: ../../general_art/myfont.ttf.
- BOOLEAN - true/1 or false/0.
- COLOR - a hexadecimal RGB or RGBA color (6 or 8 digits). If 6 digits, will assume the alpha channel is FF (not transparent).
- FLOAT - a decimal.
- STRING - a string of text.

Types of elements and their properties:

Common to almost all elements is a pos and size property of the NORMALIZED_PAIR type. They are normalized in terms of their "parent" object's size; 99% of the time, this is just the size of the screen. In this case, <pos>0 0</pos> would correspond to the top left corner, and <pos>1 1</pos> the bottom right corner (a positive Y value points further down). pos almost always refers to the top left corner of your element. You *can* use numbers outside of the [0..1] range if you want to place an element partially or completely off-screen.

The order you define properties in does not matter. Remember, you do *not* need to specify every property! *Note that a view may choose to only make only certain properties on a particular element themable!*

image

Can be created as an extra.

- pos - type: NORMALIZED_PAIR.
- size - type: NORMALIZED_PAIR.

- If only one axis is specified (and the other is zero), the other will be automatically calculated in accordance with the image's aspect ratio.
- **maxSize** - type: NORMALIZED_PAIR.
 - The image will be resized as large as possible so that it fits within this size and maintains its aspect ratio. Use this instead of size when you don't know what kind of image you're using so it doesn't get grossly oversized on one axis (e.g. with a game's image metadata).
- **origin** - type: NORMALIZED_PAIR.
 - Where on the image pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the image exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- **rotation** - type: FLOAT.
 - angle in degrees that the image should be rotated. Positive values will rotate clockwise, negative values will rotate counterclockwise.
- **rotationOrigin** - type: NORMALIZED_PAIR.
 - Point around which the image will be rotated. Defaults to 0.5 0.5.
- **path** - type: PATH.
 - Path to the image file. Most common extensions are supported (including .jpg, .png, and unanimated .gif).
- **default** - type: PATH.
 - Path to default image file. Default image will be displayed when selected game does not have an image.
- **tile** - type: BOOLEAN.
 - If true, the image will be tiled instead of stretched to fit its size. Useful for backgrounds.
- **color** - type: COLOR.
 - Multiply each pixel's color by this color. For example, an all-white image with `<color>FF0000</color>` would become completely red. You can also control the transparency of an image with `<color>FFFFFFAA</color>` - keeping all the pixels their normal color and only affecting the alpha channel.
- **visible** - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- **zIndex** - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- **reflexion** - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : table reflexion effect. First item is top position alpha, second is bottom alpha.
- **reflexionOnFrame`** - type: BOOLEAN
 - BATOCERA 5.24 : When the image is scaled maxSize, tells if the table reflexion effect alignment is done on the bottom of the physical image, or on the bottom of the frame (according to pos/size)
- **minSize** - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : The image will be resized so that it fits the entire rectangle and maintains its aspect ratio. Some parts of the image may be hidden. Used for tiles.
- **colorEnd** - type: COLOR.
 - BATOCERA 5.24 : End color for drawing the image with a gradient.
- **gradientType** - type: STRING.
 - BATOCERA 5.24 : Direction of the gradient. Values can be 'horizontal' or 'vertical'. Default is 'vertical'
- **horizontalAlignment** - type: STRING.
 - BATOCERA 5.24 : If the image is scaled maxSize, tells the image horizontal alignment.

Values can be left, right, and center. Default is center.

- `verticalAlignment` - type: `STRING`.
 - BATOCERA 5.24 : If the image is scaled `maxSize`, tells the image vertical alignment. Values can be top, bottom, and center. Default is center.
- `flipX` - type: `BOOLEAN`.
 - BATOCERA 5.24 : If true, the image will be flipped horizontally.
- `flipY` - type: `BOOLEAN`.
 - BATOCERA 5.24 : If true, the image will be flipped vertically.
- `visible` - type: `BOOLEAN`.
 - BATOCERA 5.24 : Tells if the element is visible or hidden.
- `x`, `'y'`, `'w'`, `'h'` - type: `FLOAT`.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.
- `'scale'` - type: `FLOAT`.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- `scaleOrigin` - type: `NORMALIZED_PAIR`.
 - BATOCERA 29 : Point around which the image scale will be applied.
- `'storyboard'` - type: `STORYBOARD`.
 - BATOCERA 29 : See storyboard documentation.

imagegrid

- `pos` - type: `NORMALIZED_PAIR`.
- `size` - type: `NORMALIZED_PAIR`.
 - The size of the grid. Take care the selected tile can go out of the grid size, so don't position the grid too close to another element or the screen border.
- `margin` - type: `NORMALIZED_PAIR`. Margin between tiles.
- `gameImage` - type: `PATH`.
 - The default image used for games which doesn't have an image.
- `folderImage` - type: `PATH`.
 - The default image used for folders which doesn't have an image.
- `scrollDirection` - type: `STRING`.
 - `vertical` by default, can also be set to `horizontal`. Not that in `horizontal` mod, the tiles are ordered from top to bottom, then from left to right.
- `zIndex` - type: `FLOAT`.
 - BATOCERA 5.24: z-index value for component. Components will be rendered in order of z-index value from low to high.
- `padding` - type: `NORMALIZED_RECT`.
 - BATOCERA 5.24 : Padding for displaying tiles.
- `autoLayout` - type: `NORMALIZED_PAIR`.
 - BATOCERA 5.24 : Number of column and rows in the grid (integer values).
- `autoLayoutSelectedZoom` - type: `FLOAT`.
 - BATOCERA 5.24: Zoom factor to apply when a tile is selected.
- `imageSource` - type: `STRING`.
 - BATOCERA 5.24: Selects the image to display. `thumbnail` by default, can also be set to `image` or `marquee`.
- `showVideoAtDelay` - type: `FLOAT`.
 - BATOCERA 5.24: delay in milliseconds to display video, when the tile is selected.

- `scrollSound` - type: `STRING`.
 - BATOCERA 5.24: Path of the sound file to play when the grid selection changes.
- `centerSelection` - type: `BOOL`.
 - BATOCERA 5.24: If true the selection will always be centered in the grid.
- `scrollLoop` - type: `BOOL`.
 - BATOCERA 5.24: If true the grid will loop through elements. If false, when the grid has no more elements, it displays no more items.
- `animateSelection` - type: `BOOL`.
 - BATOCERA 5.24: If true the selection under the image will be moved in an animation.
- `x`, `'y'`, `'w'`, `'h'` - type: `FLOAT`.
 - BATOCERA 29 : Tells the x, y position, width or height of the element.

gridtile

- `size` - type: `NORMALIZED_PAIR`.
 - The size of the default gridtile is used to calculate how many tiles can fit in the imagegrid. If not explicitly set, the size of the selected gridtile is equal the size of the default gridtile * 1.2
- `padding` - type: `NORMALIZED_PAIR`.
 - The padding around the gridtile content. Default 16 16. If not explicitly set, the selected tile padding will be equal to the default tile padding.
- `imageColor` - type: `COLOR`.
 - The default tile image color and selected tile image color have no influence on each others.
- `backgroundImage` - type: `PATH`.
 - If not explicitly set, the selected tile background image will be the same as the default tile background image.
- `backgroundCornerSize` - type: `NORMALIZED_PAIR`.
 - The corner size of the ninepatch used for the tile background. Default is 16 16.
- `backgroundColor` - type: `COLOR`.
 - A shortcut to define both the center color and edge color at the same time. The default tile background color and selected tile background color have no influence on each others.
- `backgroundCenterColor` - type: `COLOR`.
 - Set the color of the center part of the ninepatch. The default tile background center color and selected tile background center color have no influence on each others.
- `backgroundEdgeColor` - type: `COLOR`.
 - Set the color of the edge parts of the ninepatch. The default tile background edge color and selected tile background edge color have no influence on each others.
- `selectionMode` - type: `STRING`.
 - BATOCERA 5.24: Selects if the background is over the full tile or only the image. `full` by default, can also be set to `image`.
- `imageSizeMode` - type: `STRING`.
 - BATOCERA 5.24: Selects the image sizing mode. `maxSize` by default, can also be set to `minSize` (outer zoom) or `size` (stretch).
- `reflexion` - type: `NORMALIZED_PAIR`.
 - BATOCERA 5.24: table reflexion effect. First item is top position alpha, second is bottom alpha.

video

Can be created as an extra.

- **pos** - type: NORMALIZED_PAIR.
 - If only one axis is specified (and the other is zero), the other will be automatically calculated in accordance with the video's aspect ratio.
- **size** - type: NORMALIZED_PAIR.
 - If only one axis is specified (and the other is zero), the other will be automatically calculated in accordance with the video's aspect ratio.
- **maxSize** - type: NORMALIZED_PAIR.
 - The video will be resized as large as possible so that it fits within this size and maintains its aspect ratio. Use this instead of size when you don't know what kind of video you're using so it doesn't get grossly oversized on one axis (e.g. with a game's video metadata).
- **origin** - type: NORMALIZED_PAIR.
 - Where on the image pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the image exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- **rotation** - type: FLOAT.
 - angle in degrees that the text should be rotated. Positive values will rotate clockwise, negative values will rotate counterclockwise.
- **rotationOrigin** - type: NORMALIZED_PAIR.
 - Point around which the text will be rotated. Defaults to 0.5 0.5.
- **delay** - type: FLOAT. Default is false.
 - Delay in seconds before video will start playing.
- **default** - type: PATH.
 - Path to default video file. Default video will be played when selected game does not have a video.
- **showSnapshotNoVideo** - type: BOOLEAN
 - If true, image will be shown when selected game does not have a video and no default video is configured.
- **showSnapshotDelay** - type: BOOLEAN
 - If true, playing of video will be delayed for delayed seconds, when game is selected.
- **visible** - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- **zIndex** - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- **path** - type: PATH.
 - BATOCERA 5.24 : Path to video file if video is an extra.
- **visible** - type: BOOLEAN.
 - BATOCERA 5.24 : Tells if the element is visible or hidden.
- **snapshotSource** - type: STRING.
 - BATOCERA 5.24 : Defines which image to show during delay. Values can be image, thumbnail, and marquee.
- **effect** - type: STRING.
 - BATOCERA 5.24 : Defines the transition in effect when the video is first loaded. Values can be slideRight, size, bump, or none. The default is bump.
- **x, 'y', 'w', 'h'** - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.

- ``scale`` - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- `scaleOrigin` - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied.
- ``storyboard`` - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

text

Can be created as an extra.

- `pos` - type: NORMALIZED_PAIR.
- `size` - type: NORMALIZED_PAIR.
 - Possible combinations:
 - `0 0` - automatically size so text fits on one line (expanding horizontally).
 - `w 0` - automatically wrap text so it doesn't go beyond `w` (expanding vertically).
 - `w h` - works like a "text box." If `h` is non-zero and `h` $\leq$ `fontSize` (implying it should be a single line of text), text that goes beyond `w` will be truncated with an elipses (...).
- `origin` - type: NORMALIZED_PAIR.
 - Where on the component `pos` refers to. For example, an origin of `0.5 0.5` and a `pos` of `0.5 0.5` would place the component exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- `rotation` - type: FLOAT.
 - angle in degrees that the text should be rotated. Positive values will rotate clockwise, negative values will rotate counterclockwise.
- `rotationOrigin` - type: NORMALIZED_PAIR.
 - Point around which the text will be rotated. Defaults to `0.5 0.5`.
- `text` - type: STRING.
- `color` - type: COLOR.
- `backgroundColor` - type: COLOR;
- `fontPath` - type: PATH.
 - Path to a truetype font (.ttf).
- `fontSize` - type: FLOAT.
 - Size of the font as a percentage of screen height (e.g. for a value of `0.1`, the text's height would be 10% of the screen height).
- `alignment` - type: STRING.
 - Valid values are "left", "center", or "right". Controls alignment on the X axis. "center" will also align vertically.
- `forceUppercase` - type: BOOLEAN. Draw text in uppercase.
- `lineSpacing` - type: FLOAT. Controls the space between lines (as a multiple of font height). Default is 1.5.
- `visible` - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- `zIndex` - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- `glowColor` - type: COLOR;
 - BATOCERA 5.24 : Defines the color of the glow around the text.
- `glowSize` - type: FLOAT.

- BATOCERA 5.24 : Defines the size of the glow around the text.
- glowOffset - type: FLOAT.
 - BATOCERA 5.24 : Defines the decal of the glow around the text (used for shadow effects).
- reflexion - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : table reflexion effect. First item is top position alpha, second is bottom alpha.
- reflexionOnFrame - type: BOOLEAN
 - BATOCERA 5.24 : When the image is scaled maxSize, tells if the table reflexion effect alignment is done on the bottom of the physical image, or on the bottom of the frame (according to pos/size)
- padding - type: NORMALIZED_RECT.
 - BATOCERA 5.24 : Padding for displaying text.
- visible - type: BOOLEAN.
 - BATOCERA 5.24 : Tells if the element is visible or hidden.
- x, 'y', 'w', 'h' - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.
- 'scale' - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- scaleOrigin - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied.
- 'storyboard' - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

textlist

- pos - type: NORMALIZED_PAIR.
- size - type: NORMALIZED_PAIR.
- origin - type: NORMALIZED_PAIR.
 - Where on the component pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the component exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- selectorColor - type: COLOR.
 - Color of the "selector bar."
- selectorImagePath - type: PATH.
 - Path to image to render in place of "selector bar."
- selectorImageTile - type: BOOLEAN.
 - If true, the selector image will be tiled instead of stretched to fit its size.
- selectorHeight - type: FLOAT.
 - Height of the "selector bar".
- selectorOffsetY - type: FLOAT.
 - Allows moving of the "selector bar" up or down from its computed position. Useful for fine tuning the position of the "selector bar" relative to the text.
- selectedColor - type: COLOR.
 - Color of the highlighted entry text.
- primaryColor - type: COLOR.
 - Primary color; what this means depends on the text list. For example, for game lists, it is the color of a game.
- secondaryColor - type: COLOR.
 - Secondary color; what this means depends on the text list. For example, for game lists, it

is the color of a folder.

- `fontPath` - type: PATH.
- `fontSize` - type: FLOAT.
- `scrollSound` - type: PATH.
 - Sound that is played when the list is scrolled.
- `alignment` - type: STRING.
 - Valid values are "left", "center", or "right". Controls alignment on the X axis.
- `horizontalMargin` - type: FLOAT.
 - Horizontal offset for text from the alignment point. If `alignment` is "left", offsets the text to the right. If `alignment` is "right", offsets text to the left. No effect if `alignment` is "center". Given as a percentage of the element's parent's width (same unit as `size`'s X value).
- `forceUppercase` - type: BOOLEAN. Draw text in uppercase.
- `lineSpacing` - type: FLOAT. Controls the space between lines (as a multiple of font height). Default is 1.5.
- `zIndex` - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- `selectorColorEnd` - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : Bottom color for the gradient of the "selector bar."
- `selectorGradientType` - type: STRING.
 - BATOCERA 5.24 : Type of gradient to apply : horizontal or vertical. Default is vertical
- `x`, `'y'`, `'w'`, `'h'` - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.
- `'scale'` - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- `scaleOrigin` - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied.
- `'storyboard'` - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

ninepatch

Can be created as an extra.

- `pos` - type: NORMALIZED_PAIR.
- `size` - type: NORMALIZED_PAIR.
- `path` - type: PATH.
- `visible` - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- `zIndex` - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- `color` - type: COLOR.
 - BATOCERA 5.24 : Fill color (assigns center & edge color)
- `centerColor` - type: COLOR.
 - BATOCERA 5.24 : Color of the center.
- `edgeColor` - type: COLOR.

- BATOCERA 5.24 : Color of the edge.
- `cornerSize` - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : Size of the corners. Default is 32 32.
- `x`, `y`, `w`, `h` - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.
- `scale` - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- `scaleOrigin` - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied.
- `storyboard` - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

EmulationStation borrows the concept of “nine patches” from Android (or “9-Slices”). Currently the implementation is very simple and hard-coded to only use 48x48px images (16x16px for each “patch”). Check the `data/resources` directory for some examples (`button.png`, `frame.png`).

rating

- `pos` - type: NORMALIZED_PAIR.
- `size` - type: NORMALIZED_PAIR.
 - Only one value is actually used. The other value should be zero. (e.g. specify width OR height, but not both. This is done to maintain the aspect ratio.)
- `origin` - type: NORMALIZED_PAIR.
 - Where on the component `pos` refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the component exactly in the middle of the screen. If the “POSITION” and “SIZE” attributes are themable, “ORIGIN” is implied.
- `rotation` - type: FLOAT.
 - angle in degrees that the rating should be rotated. Positive values will rotate clockwise, negative values will rotate counterclockwise.
- `rotationOrigin` - type: NORMALIZED_PAIR.
 - Point around which the rating will be rotated. Defaults to 0.5 0.5.
- `filledPath` - type: PATH.
 - Path to the “filled star” image. Image must be square (width equals height).
- `unfilledPath` - type: PATH.
 - Path to the “unfilled star” image. Image must be square (width equals height).
- `color` - type: COLOR.
 - Multiply each pixel’s color by this color. For example, an all-white image with `<color>FF0000</color>` would become completely red. You can also control the transparency of an image with `<color>FFFFFFAA</color>` - keeping all the pixels their normal color and only affecting the alpha channel.
- `visible` - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- `zIndex` - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- `x`, `y`, `w`, `h` - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used

instead of pos/size.

- `scale` - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- scaleOrigin - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied
- `storyboard` - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

datetime

- pos - type: NORMALIZED_PAIR.
- size - type: NORMALIZED_PAIR.
 - Possible combinations:
 - 0 0 - automatically size so text fits on one line (expanding horizontally).
 - w 0 - automatically wrap text so it doesn't go beyond w (expanding vertically).
 - w h - works like a "text box." If h is non-zero and $h \leftarrow \text{fontSize}$ (implying it should be a single line of text), text that goes beyond w will be truncated with an elipses (...).
- origin - type: NORMALIZED_PAIR.
 - Where on the component pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the component exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- rotation - type: FLOAT.
 - angle in degrees that the text should be rotated. Positive values will rotate clockwise, negative values will rotate counterclockwise.
- rotationOrigin - type: NORMALIZED_PAIR.
 - Point around which the text will be rotated. Defaults to 0.5 0.5.
- color - type: COLOR.
- backgroundColor - type: COLOR;
- fontPath - type: PATH.
 - Path to a truetype font (.ttf).
- fontSize - type: FLOAT.
 - Size of the font as a percentage of screen height (e.g. for a value of 0.1, the text's height would be 10% of the screen height).
- alignment - type: STRING.
 - Valid values are "left", "center", or "right". Controls alignment on the X axis. "center" will also align vertically.
- forceUppercase - type: BOOLEAN. Draw text in uppercase.
- lineSpacing - type: FLOAT. Controls the space between lines (as a multiple of font height). Default is 1.5.
- visible - type: BOOLEAN.
 - If true, component will be rendered, otherwise rendering will be skipped. Can be used to hide elements from a particular view.
- zIndex - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- displayRelative - type: BOOLEAN. Renders the datetime as a a relative string (ex: 'x days ago')
- format - type: STRING. Specifies format for rendering datetime.
 - %Y: The year, including the century (1900)
 - %m: The month number [01,12]

- %d: The day of the month [01,31]
- %H: The hour (24-hour clock) [00,23]
- %M: The minute [00,59]
- %S: The second [00,59]
- x, 'y', 'w', 'h' - type: FLOAT.
 - BATOCERA 29 : Respectively x, y position, width or height of the element. Can be used instead of pos/size.
- `scale` - type: FLOAT.
 - BATOCERA 29 : Scale of the element. 0 to 1. Default is 1.
- scaleOrigin - type: NORMALIZED_PAIR.
 - BATOCERA 29 : Point around which the image scale will be applied.
- `storyboard` - type: STORYBOARD.
 - BATOCERA 29 : See storyboard documentation.

sound

- path - type: PATH.
 - Path to the sound file. Only .wav files are currently supported.

helpsystem

- pos - type: NORMALIZED_PAIR. Default is "0.012 0.9515"
- origin - type: NORMALIZED_PAIR.
 - Where on the component pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the component exactly in the middle of the screen.
- textColor - type: COLOR. Default is 777777FF.
- iconColor - type: COLOR. Default is 777777FF.
- fontPath - type: PATH.
- fontSize - type: FLOAT.

carousel

- type - type: STRING.
 - Sets the scoll direction of the carousel.
 - Accepted values are "horizontal", "vertical", "horizontal_wheel" or "vertical_wheel".
 - Default is "horizontal".
- size - type: NORMALIZED_PAIR. Default is "1 0.2325"
- pos - type: NORMALIZED_PAIR. Default is "0 0.38375".
- origin - type: NORMALIZED_PAIR.
 - Where on the carousel pos refers to. For example, an origin of 0.5 0.5 and a pos of 0.5 0.5 would place the carousel exactly in the middle of the screen. If the "POSITION" and "SIZE" attributes are themable, "ORIGIN" is implied.
- color - type: COLOR.
 - Controls the color of the carousel background.
 - Default is FFFFFFFD8
- logoSize - type: NORMALIZED_PAIR. Default is "0.25 0.155"
- logoScale - type: FLOAT.
 - Selected logo is increased in size by this scale

- Default is 1.2
- **logoRotation** - type: FLOAT.
 - Angle in degrees that the logos should be rotated. Value should be positive.
 - Default is 7.5
 - This property only applies when type is "horizontal_wheel" or "vertical_wheel".
- **logoRotationOrigin** - type: NORMALIZED_PAIR.
 - Point around which the logos will be rotated. Defaults to -5 0.5.
 - This property only applies when type is "horizontal_wheel" or "vertical_wheel".
- **logoAlignment** - type: STRING.
 - Sets the alignment of the logos relative to the carousel.
 - Accepted values are "top", "bottom" or "center" when type is "horizontal" or "horizontal_wheel".
 - Accepted values are "left", "right" or "center" when type is "vertical" or "vertical_wheel".
 - Default is "center"
- **maxLogoCount** - type: FLOAT.
 - Sets the number of logos to display in the carousel.
 - Default is 3
- **zIndex** - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.
- **colorEnd** - type: COLOR.
 - BATOCERA 5.24 : Color for the end of gradient
- **gradientType** - type: STRING.
 - BATOCERA 5.24 : Sets the gradient direction. Accepted values are "horizontal" and "vertical".
- **logoPos** - type: NORMALIZED_PAIR.
 - BATOCERA 5.24 : Set the logo position if it is not centered.
- **systemInfoDelay** - type: FLOAT.
 - BATOCERA 5.24 : Sets the amount of time before displaying game count information. Default is 1500

menuText & menuTextSmall

BATOCERA 5.24

- **color** - type: COLOR.
 - Default is 777777FF
- **fontPath** - type: PATH.
 - Path to a truetype font (.ttf).
- **fontSize** - type: FLOAT.
 - Size of the font as a percentage of screen height (e.g. for a value of 0.1, the text's height would be 10% of the screen height). Default is 0.085 for menutitle, 0.045 for menutext and 0.035 for menufooter and menutextsmall.
- **separatorColor** - type: COLOR.
 - Default is C6C7C6FF. Color of lines that separates menu entries.
- **selectedColor** - type: COLOR.
 - Default is FFFFFFFF. Color of text for selected menu entry.
- **selectorColor** - type: COLOR.

- Default is 878787FF. Color of the selector bar.
- selectorColorEnd - type: NORMALIZED_PAIR.
 - Bottom color for the gradient of the "selector bar."

menuSwitch

BATOCERA 5.24 - replace the switch image in menus

- pathOn - type: PATH.
 - Image path when the switch is on
- pathOff - type: PATH.
 - Image path when the switch is off

menuButton

BATOCERA 5.24 - replace the button images in menus

- path - type: PATH.
 - Image path when the button is not selected
- filledPath - type: PATH.
 - Image path when the button is selected.

menuTextEdit

BATOCERA 5.24 - replace the textbox background images in OSK

- inactive - type: PATH.
 - Image path when the textbox is not selected
- active - type: PATH.
 - Image path when the textbox is selected.

menulcons

BATOCERA 5.24 - Add images to menuitems

- * - type: PATH.
 - Element name must be the image name in the list : iconSystem, iconUpdates, iconControllers, iconGames, iconUI, iconSound, iconNetwork, iconScraper, iconAdvanced, iconQuit, iconRetroachievements, iconKodi, iconRestart, iconShutdown, iconFastShutdown

controllerActivity

BATOCERA 5.24 - controllerActivity element must be defined in screen view

- pos - type: NORMALIZED_PAIR.

- **size** - type: NORMALIZED_PAIR.
 - Only one value is actually used. The other value should be zero. (e.g. specify width OR height, but not both. This is done to maintain the aspect ratio.)
- **itemSpacing** - type: FLOAT.
 - Space between controller images
- **horizontalAlignment** - type: STRING.
 - Image alignment - left, right, center.
- **imagePath** - type: PATH.
 - Image to use to represent a joystick.
- **color** - type: COLOR.
 - Default color.
- **activityColor** - type: COLOR.
 - Color when the joystick has activity
- **hotkeyColor** - type: COLOR.
 - Color when the joystick hotkey is pressed.
- **visible** - type: BOOLEAN.
 - Show/Hides the controllers
- **zIndex** - type: FLOAT.
 - z-index value for component. Components will be rendered in order of z-index value from low to high.

Animating elements using Storyboards

** Batocera 29

You can now create animations using storyboards on every extra element & some built-in elements.

Storyboard example

Here is a very simple example of an animation that animates an image position, scale & opacity :

```
<theme>
  <formatVersion>7</formatVersion>
  <view name="detailed">
    <image name="my_image" extra="true">
      <pos>0.5 0.5</pos>
      <origin>0.5 0.5</origin>
      <size>0.8 0.8</size>
      <path>./my_art/my_awesome_image.jpg</path>
      <storyboard>
        <animation property="y" from="0.05" begin="0" duration="350"
mode="easeOut"/>
        <animation property="opacity" from="0" to="1" duration="350"
mode="linear"/>
        <animation property="scale" from="0" to="1" duration="350"
mode="easeOut"/>
      </storyboard>
    </image>
  </view>
</theme>
```

```
</image>
</view>
</theme>
```

The 'storyboard' element

A storyboard is a set of animated properties. This element should contain child "animation" elements.

This element can be added to any common elements : image, video, text, datetime, rating, ninepatch, textlist, imagegrid.

Only exceptions are : sound, menu, carousel, helpsystem and splash elements.

Attributes

event - type: STRING - Optional. Name of the event the storyboard applies to. This value can be used in systemview's `<html><image name="logo"></html>` & `<html><text name="logoText"></html>` elements. In this case, value can be :

- "activate" : animation applies when a logo becomes active only
- "deactivate" : animation applies when the active logo deactivates
- If the value is empty or if the attribute is missing, the storyboard applies to all logos.

This value can be used in video elements : In this case, value can be :

- "staticImage" : animation applies only to the static image behind the video
- If the value is empty or if the attribute is missing, the storyboard applies to the video when it starts.

In those case, multiple storyboards, each one with a different event is possible.

- repeat - type: STRING or INTEGER - Optional. Tells if the storyboard repeats itself. If the value is empty or 1, there is no repeat. Value can also be "forever" or an integer value (number of times the storyboard is repeated). This value is ignored if any of the animation elements has a repeat="forever" attribute.

The 'animation' element

animation element must be a child of a storyboard.

This element is designed to set a list of properties to transform, given a timing, an acceleration curve, and a repeat information.

Attributes

- property - type: STRING - Required : Name of the property to animate.

Value is one of the property in the object (pos, size, x, y, w, h, scale, scaleOrigin, color, opacity, rotation, rotationOrigin, zIndex...) Note : It's not possible to animate STRING and BOOLEAN properties. Only FLOAT, NORMALIZED_PAIR, NORMALIZED_RECT and COLOR elements can be animated.

- from- type: DYNAMIC (uses the same type as the source property) - Optional. Tells the start value of the animation. If omitted, the value is taken in the control when the animation starts.
- to- type: DYNAMIC (uses the same type as the source property) - Optional. Tells the end value of the animation. If omitted, the value is taken in the control when the animation starts.
- begin- type: INTEGER - Optional. Tells when the animation should begin. This information is given in milliseconds. Default is 0 (immediate).
- duration- type: INTEGER - Optional. Duration of the animation in millisecond. If the value is 0, the To value will be set at "begin" of this animation. Default is 0.
- repeat - type: STRING or INTEGER - Optional. Tells if the animation repeats itself. If the value is empty or 1, the animation won't be repeated. Value can also be "forever" to loop infinitely or an integer value which will be the number of times the storyboard is repeated. Default is 1.
- autoReverse- type: BOOLEAN - Optional. Determines whether the animation plays in reverse after it completes a forward iteration. Default is false.
- mode- type: STRING - Optional. Defines the acceleration curve to apply. Default is "Linear". Value is one of :

Linear
EaseIn
EaseInCubic
EaseInQuint
EaseOut
EaseOutCubic
EaseOutQuint
EaseInOut
Bump

From:

<https://www.wiki.batocera.org/> - **Batocera.linux - Wiki**

Permanent link:

https://www.wiki.batocera.org/write_themes_for_emulationstation?rev=1645666045

Last update: **2022/02/24 01:27**

