

# orxOBJECT structure

## Summary

```
[ObjectTemplate]
Alpha           = <float>
AngularVelocity = <float>
AnimationFrequency = <float>
AnimationSet    = AnimationSetTemplate
AutoScroll      = x | y | both
BlendMode       = alpha | multiply | add | none
Body            = BodyTemplate
ChildList       = ObjectTemplate1 # ObjectTemplate2
ChildJointList  = JointTemplate1 # JointTemplate2
Clock           = ClockTemplate
Color           = <vector>
DepthScale      = <bool>
Graphic         = GraphicTemplate
Group           = <string>
Flip            = x | y | both
FXList          = FXTemplate1 # FXTemplate2
LifeTime        = <float> | fx | sound | spawn | track | child
ParentCamera    = CameraTemplate
Pivot           = center(+truncate|round) | left | right | top | bottom | <Vector>
Position        = <vector>
Repeat          = <vector>
Rotation        = <float>
Speed           = <vector>
Scale           = <vector> | <float>
ShaderList      = ShaderTemplate1 # ShaderTemplate2
TrackList       = TimeLineTrackTemplate1 # TimeLineTrackTemplate2
SoundList       = SoundTemplate1 # SoundTemplate2
Spawner         = SpawnerTemplate
Smoothing       = <bool>
UseParentSpace  = <bool> | none | position | scale | both
UseRelativeSpeed = <bool>
```

## Details

Here's a list of the available properties for an orxOBJECT structure:

- **Alpha**: Opacity of the object. Its value should be between 0.0 and 1.0. By default, its value is 1.0 (fully opaque).
- **AngularVelocity**: Initial angular velocity for this object, in degrees/second. By default its value is 0°/s.
- **AnimationFrequency**: Relative frequency at which animations should be played on this object. This acts as a multiplier on the original frequency of played [animations](#). By default its value is 1.0.

- **AnimationSet**: Config name of the animation set to use for this object (see [orxANIM section](#)). If none is provided, no animation will be available for this object. In order to use an **AnimationSet**, a valid [orxGRAPHIC](#) has to be provided for the **Graphic** property.
- **AutoScroll**: Defines if this object should have an automated differential scrolling rendering depending on its relative depth (Z axis) to the camera. It can be defined for the X axis or the Y axis or even both at the same time. If none is provided, no differential scrolling will be used for this object (ie. default plain rendering position).
- **BlendMode**: Defines which [blend mode](#) will be used when rendering this object. By default its value is `alpha`, which means the alpha channel will be used as opacity value. This rendering parameter can be overridden by the current rendered [graphic object](#).
- **Body**: Config name of the [orxBODY](#) to use for this object. This gives physical properties <sup>1)</sup> to the object. If none is provided, the object won't have any physical properties (ie. no collision detection).
- **ChildList**: List of objects (config names) to create along this object. They'll be linked as children which means they'll adapt to the object's current position, rotation and scale (they'll 'follow' it). Children will automatically be deleted when the current object (ie. their parent) is deleted. Up to 255 children can be created. If no value is provided for this property, no children will be created.
- **ChildJointList**: Optional list of joints to use to link the children to the parent. If defined, its indices **must** match those of the **ChildList** property. Defaults to nothing.
- **Clock**: Associates the corresponding clock and creates it if needed. If no value is provided, no clock will be associated (updates will only depend on the main (aka core) clock). Associated clocks are useful to achieve time localized stretching (only for the corresponding object). When an object is time stretched by an associated clock, all its properties will behave accordingly (spawners, FXs, physics and sound pitch).
- **Color**: Defines a color (ie. tint) to apply on this object when rendered. This color will override any color defined in the current rendered [orxGRAPHIC](#) for this object. If none is provided, the color of the current [orxGRAPHIC](#) will be used if available, the white color (ie. no tint) will be used otherwise.
- **DepthScale**: Defines if the object should be *visually* (and not physically) scaled when rendered, depending on its relative depth (Z axis) to the camera. If you want to achieve a physically correct display of an object with differential scrolling, this property should be set to `true` **AND** **AutoScroll** set to `both`. If not you might get visual gaps at rendering. However, if you don't use [orxBODY](#) for this object, it doesn't matter. By default its value is `false`.
- **Flip**: Defines if the object should be flipped when rendered <sup>2)</sup>. It can be flipped on either X axis or Y axis or on both axes at the same time. If no value is provided, no flipping will occur.
- **FXList**: Specifies a list of [orxFX](#) to apply on this object when created. Up to 4 FXs can be defined. If none is provided, no [orxFX](#) will be applied.
- **Graphic**: Config name of the [orxGRAPHIC](#) associated to this object. When an object is animated, the current frame's graphic will be displayed instead. If none is provided, the object will exist but won't be visible (ie. nothing will be displayed for it by the renderer).
- **Group**: GroupId from the [GroupList](#) for the object to be rendered into (like a layer).
- **LifeTime**: If a float value is specified, the life time of this object in seconds. 0 means it'll be instantly deleted. Its default value is -1.0, which means the object has an infinite life time. If a literal is used, the object's life will end when the condition is fulfilled. Multiple literals can be defined at once, in which case the object will be deleted when all of their associated conditions are fulfilled. `child` keeps the object alive as long as there are owned children; `fx` keeps the object alive as long as there are FXs; `sound` keeps the object alive as long as there are sounds;

spawn keeps the object alive as long as spawner isn't depleted (based on TotalObject); track keeps the object alive as long as there are tracks

- **ParentCamera**: Config name of the [orxCAMERA](#) that will be used as parent. This is very useful to place objects in the camera space for UI purposes, for example. If none is provided, the object won't have any parent except if it was created through a [ChildList](#) of another object.
- **Pivot**: Normally this is defined in the Object's Graphic section. However, setting Pivot here overrides graphic's value if present; truncate and round will adjust pivot values if they are not integers; z is ignored for 2D graphics;
- **Position**: Initial position for this object, in pixels. If an object has a parent, this value will be considered in parent's space instead than in world space. By default its value is (0.0, 0.0, 0.0).
- **Repeat**: Specifies if the object should be repeated (so as to achieve a tiling effect) when rendered. The vector specifies the number of times the object will be repeated on X and Y axes (Z being ignored). If none is provided, the value specified for the current [orxGRAPHIC](#) will be used. Please notice that the repeat property does not change the object's size, if you wish the object to be drawn several times (e. g.: you want to render the ground of your game, by having the same graphic several times) you need to set the [Scale](#) property as well.
- **Rotation**: Initial 2D rotation for this object, in degrees. If an object has a parent, this value will be considered in parent's space instead than in world space. By default its value is 0.0.
- **Speed**: Initial speed for this object, in pixels/second. Even objects without an [orxBODY](#) can use speed. By default its value is (0.0, 0.0, 0.0).
- **Scale**: Initial scale for this object (Z component being ignored). By default its value is 1.0. (ie. (1.0, 1.0, 0.0) )
- **ShaderList**: List of [orxSHADER](#) to apply when this object will be rendered. The shader is applied after the object's rendering but object without any Graphic can also use them. Up to 4 shaders can be defined. If none is provided, no shader will be applied when the object is rendered.
- **SoundList**: List of [orxSOUND](#) to apply on the object at its creation. They can be both music or sound. Up to 4 sounds can be defined. If none is provided, no sound will be created with this object.
- **Spawner**: Config name of the [orxSPAWNER](#) that will be created and linked to this object. By default no spawner will be created.
- **Smoothing**: Specifies if the graphic object should be rendered with antialiasing or not. This property is overridden by the current [orxGRAPHIC](#) used for this object. By default its value is default which means the [global display setting value](#) will be used.
- **TrackList**: Timeline tracks will be played immediately. Up to 16 timeline tracks can be defined. [Command Module notes](#) is one way to use TrackList.
- **UseParentSpace**: If set to true or both and the object has a valid parent [orxCAMERA](#)<sup>3)</sup>, its position and scale will be considered in parent's space, ie. values between 0.0 and 1.0 defines parent's size on each axis<sup>4)</sup>. Only one of the two parent property can be considered by using the value position or rotation. Its value is true by default.
- **UseRelativeSpeed**: If set to true, the value of the Speed property will be considered relatively to the current object's rotation and scale. By default its value is false.

## Latest config settings for the Development Version

We endeavor to keep the config properties on this page up to date as often as possible. For up to the minute config information for the latest version of Orx, check the most recent published at:

[CreationTemplate.ini](#) and

## [SettingsTemplate.ini](#)

Additionally these files can be found under your orx source tree in the `orx/code/bin` folder.

- 1) collision detection/dynamics through the use of orx's internal physics plugin
- 2) it only affects the visual, physical property, ie. its body, won't be affected
- 3) this won't affect objects created through another object's `ChildList`
- 4) mainly used for easiness of placing UI objects relative to camera that will remain resolution independent

From: <https://orx-project.org/wiki/> - **Orx Learning**

Permanent link: [https://orx-project.org/wiki/en/orx/config/settings\\_structure/orxobject?rev=1559132817](https://orx-project.org/wiki/en/orx/config/settings_structure/orxobject?rev=1559132817)

Last update: **2025/09/30 17:26 (6 months ago)**

