

# Scene

## Table of Contents

- [Description](#)
- [Fields](#)

**Extends:** [Group](#)

## Description

The **Scene** node class serves as the root of a SceneGraph node tree. Every **roSGScreen** object must have a **Scene** node, or a node that derives from the **Scene** node class as its root, including an XML markup component that extends the **Scene** node class or subclass. That node must be created using the **roSGScreen** `createScene()` function, with an argument that is a string of the name of the **Scene** node object created. For example:

```
screen = CreateObject("roSGScreen")
scene = screen.CreateScene("Scene")
```

While it is technically possible to have more than one scene per channel, we recommend you only have one **roSGScreen** and one **Scene** node. Child nodes of the scene can be treated as different "scenes" where you can then implement transitions between them.

## Fields

Field	Type	Default	Use
backgroundURI	uri	invalid	Specifies a graphic image file to be used for the <b>Scene</b> node background.
backgroundColor	color	0x000000FF	Specifies a background color for the scene. This color is only used if the <code>backgroundURI</code> field is set to an empty string. For example:  <code>scene.backgroundColor="0xEB1010FF"</code> <code>scene.backgroundUri = ""</code>
dialog	node	invalid	Setting this field to a node extended from a <b>Dialog</b> node causes the dialog to be displayed
backExitsScene	Boolean	true	If true, a <b>Back</b> key press causes the scene to exit, back to the last user-focused item. If false, a <b>Back</b> key press does not cause the scene to exit. In order for the <b>Back</b> key to cause the scene to exit, the remote control focus must be explicitly set on the scene, or a child of the scene, using the <code>ifSGNodeFocus</code> interface <code>setFocus(true)</code> function. A <b>Home</b> key press always causes the scene to exit.

currentDesignResolution	assocarray		<p><i>This field is available in firmware 7.7 and later.</i></p> <p>This read-only field is set when the Scene is initialized. It indicates which of a channel's design resolutions (per <a href="#">manifest's ui_resolutions</a> value) is being used, based on the player model and connected display type. Previously, a developer could deduce the same information by using both <a href="#">roDeviceInfo.GetUIResolution()</a> and <a href="#">roAppInfo.getValue("ui_resolutions")</a>. This new field simplifies the process.</p> <p>The field is set to an AA with two numeric-valued keys — <i>width</i> and <i>height</i> — as well as a string value indicating the current design <i>resolution</i> ("HD", "FHD" or "SD"). E.g.</p> <pre>Brightscript Debugger&gt; ? myNode.getScene().currentDesignResolution &lt;Component: roAssociativeArray&gt; = {   height: 720   resolution: "HD"   width: 1280 }</pre>
-------------------------	------------	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------