

Index

Space Index

0-9 ... 0	A ... 10	B ... 15	C ... 29	D ... 16	E ... 5
F ... 4	G ... 7	H ... 1	I ... 105	J ... 0	K ... 3
L ... 10	M ... 9	N ... 3	O ... 5	P ... 17	Q ... 0
R ... 132	S ... 33	T ... 18	U ... 6	V ... 2	W ... 4
X ... 2	Y ... 0	Z ... 2	!@#\$... 0		

0-9

B

Best Practices

Performance Best Practice Rationale | Benefits Optimize your channel code to provide smooth transitions and animations Je

blocky transitions erode the seamlessness of the Roku user experience and feel unfinished or “broken” to users. Minimize p

Best Practices for Data Management

Table of Contents Thread Ownership of Nodes Each node is owned by the thread which created it, which might or might not Render thread. However, if a node interacts with the Render thread, the Render thread will automatically take ownership of it

Best Practices for Roku Pay

Table of Contents This list of best practices for Roku Pay is a good checklist to review your implementation. Most advice is to implement, and will help optimize your user acquisition funnel, provide better feedback to your customer base, and prevent

Bookmarking Playback Position

Table of contents: Overview When it comes to streaming media, content bookmarking refers to the idea of recording a user's position on the content in your channel so that later on they will be able to continue watching from precisely where they left

BrightScript Component Reference

BrightScript Components

Two new BrightScript components, `roSGScreen` and `roSGNode`, have been defined to allow our SceneGraph technology to scripting. Currently, a fairly strict ordering must be used in BrightScript to create a SceneGraph screen and set up its Scene

BrightScript Language Reference

Roku BrightScript is a powerful scripting language that makes it easy and quick to build media and networked applications for embedded devices. The language has integrated support for BrightScript Components, a library of lightweight components.

BrightScript Profiler

BrightScript Profiler is a tool created for collecting and analyzing channel metrics that can be used to determine where performance improvements and efficiencies can be made in the channel. Table of Contents The BrightScript profiler provides the following

BrightScript Support

Several BrightScript functions and components cannot be used in SceneGraph component scripts. Many of the BrightScript components that cannot be used provide duplicate rendering functionality as SceneGraph nodes, and cannot be used for their intended purpose. You should

BrightScript/XML Markup Equivalence

Because the SceneGraph API includes BrightScript objects and interfaces, you have the flexibility to create SceneGraph application screens using a variety of methods. You can create the screens entirely in XML markup in the `<children>` element of the `Screen`

BrightScriptDoc

Table of Contents Overview BrightScriptDoc is the BrightScript language documentation generator for inline code comments that operates inside Roku's Eclipse plug-in, see Roku Plug-in for Eclipse IDE <https://sdkdocs.roku.com/display/sdkdoc/Roku+Plug-in+for+Eclipse+IDE>

Building a channel using SceneGraph Developer Extensions

This tutorial is for building a SceneGraph Developer Extensions (SGDEX) based channel. Developers who want to build the channel, or are interested in moving their existing channel to RSG, benefit from this guide. SGDEX includes the following

BusySpinner

Table of Contents Extends: `Poster` Description The `BusySpinner` node class is a simple widget that displays a continuously rotating bitmap. Since the `BusySpinner` node class uses an internal `Poster` node instance, the busy spinner bitmap can be specified

Button

Table of Contents Extends: `Group` Description The `Button` node is a simple widget that generates a `buttonSelected` event when the user selects it. The button can display a label and/or an icon, as well as a background image. Fields are provided to customize

ButtonGroup

Table of Contents Extends: `LayoutGroup` Description The `ButtonGroup` node class manages the layout, visual attributes, and management of a vertical list of `Button` nodes. When the `ButtonGroup` node has focus, it sets the key focus on a single one of the

D

[Debugging SceneGraph Applications](#)

Table of Contents [Special SceneGraph Debugging Commands Available since firmware version 7.2](#) You can use special de
commands to debug SceneGraph applications. These commands can be invoked in the debug server from telnet port 8080
comma

[Debugging Your Application](#)

Testing Roku Channels involves using a debug console and access to a variety of ports. The debug console provides a win
runtime environment and provides features such as crash logs, stack-traces and much more. Table of Contents [Nested und](#)

[Deep Linking](#)

Table of Contents [Overview of deep linking](#) Deep linking lets other parts of the system launch a particular piece of video con
parameters passed to the channel at launch time, rather than launching the home page for the channel. To enable the de

[Defining SceneGraph Components](#)

A Roku SceneGraph application consists of one or more SceneGraph components. These SceneGraph components are de XML files, which consist of a required <component> XML element, that contains the other possible XML elements to fully de compo

Deprecated APIs

APIs that have been deprecated should not be used. If you use an API after it is deprecated, your channel may not pass cer <https://developer.roku.com/develop/channel-store/certification>. Table of Contents By removing these components from your

Designing for Device Capabilities

Every public channel on the Roku platform is certified on all currently-supported Roku models before being published. The li supported devices includes a wide range of product classes, from our entry-level set-top-boxes to 4K HDR TVs, and everyth

Developer Guide

Table of Contents

Developer Settings

Developer Settings contains a host of essential tools for Roku channel development: Additional references on this page: Ins "side-loading" channels Utilities to rekey a Roku device and take screenshots Packaging channels for publication to the C

Developer Setup Guide

Table of Contents Overview The first step to building channels on the Roku Platform is setting up a development environme guide, we will cover the essential steps to enable Developer Settings on a Roku device. Steps: Setup your Roku device to e

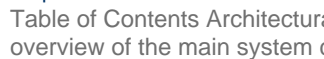
Developer Tools

Pages under this section:

Developing SceneGraph Applications

Table of Contents Channel packages should include all the XML components they define in a top level directory named cor (the top level directory in the channel application package contains the manifest file, the source directory, and so forth). Whe

Development Environment Overview

Table of Contents Architectural Overview  Architecture Block Diagram The diagram above provides a overview of the main system components for the Roku Streaming Player platform. Developer applications are written using t

Dialog

Table of Contents Extends: Group Description The Dialog node class defines a modal pop-up dialog used to present the use information requiring their immediate attention. Setting the dialog field of the current Scene node to a Dialog node causes th

Dialog Nodes

This section describes the various SceneGraph dialog node classes available to Roku developers.

Dialogs Markup

Table of Contents Showing a modal dialog in a Roku SceneGraph Scene node is simple: Create a dialog node object Config dialog node object Set the Scene node dialog field to the dialog node object The dialog is dismissed if the user presses the l

Downloading Server Content

Table of Contents Various types of content material can be downloaded from your server to a SceneGraph application. Man renderable nodes include a uri field that allows you specify the URI of a graphical image file or other content on your server.

F

Fast Video Start

Table of Contents Available since firmware version 7.2 Fast Video Start is a user interface design technique for VOD chann reduces the amount of time apparent to the user that the video stream buffers before actual playback begins. All digital vid

File System

Table of Contents Application storage <https://sdkdocs.roku.com/display/sdkdoc/File+System#FileSystem-Applicationstorage> Pathnames <https://sdkdocs.roku.com/display/sdkdoc/File+System#FileSystem-Pathnames> Examples <https://sdkdocs.roku.com/display/sdkdoc/F>

FloatFieldInterpolator

Table of Contents Extends: Node Description The FloatFieldInterpolator node class specifies a keyframe animation sequenc applied to a floating point field of a node (such as, an opacity, width or height value.) All field interpolators include a se

Font

Table of Contents Extends: Node Description The Font node class specifies the font to be used by a Label node, or any oth

that render text. Nodes that use fonts include a field that stores a Font node. The font to use is specified by creating a F

H

Handling Application Events

Table of Contents Observer Callback Models Firmware v7.5 introduces a fundamental change in the observer callback model from a queued or deferred model to a more intuitive and expected recursive callback model. Queued Callback Model In firm

J**L****Label**

Table of Contents Extends: Group Description The Label node class is used to display a string of text. The Label node class has many options including: Specifying the color of the font Specifying the font to be used by specifying a TrueType/OpenType

LabelList

Table of Contents Extends: ArrayGrid Also See Related Page: List and Grids Markup - LabelList Markup Description The LabelList node class is a simple list class that can be used to display a list of items. Each item can include a text string and an option

LabelList Markup

Example Applications: LabelListExample.zip

<https://sdkdocs.roku.com/download/attachments/4262893/LabelListExample.zip?version=2&modificationDate=1471018798>

[LabelListFocusStyleExample.zip https://sdkdocs.roku.com/download/attachments/4262893/L](https://sdkdocs.roku.com/download/attachments/4262893/LabelListFocusStyleExample.zip)

Layout/Group Nodes

The following node classes allow you group and arrange sets of renderable node classes.

Layout/Groups Markup

Table of Contents There are a few Roku SceneGraph node classes that group screen sub-elements together to allow them to be used as a group: ButtonGroup LayoutGroup ButtonGroup Markup Example Application: ButtonGroupExample.zip Node Class Reference Button

LayoutGroup

Table of Contents Extends: Group Description The LayoutGroup node class manages the position of its child nodes by arranging them in a row from left to right (horizontal layout), or in a column from top to bottom (vertical layout). Fields provide options

List and Grid Nodes

The following are the list and grid node classes included in the SceneGraph API.

ListPanel

Table of Contents Extends: Panel Description The ListPanel node class allows you to easily create a Panel that adheres to the standard layout and behavior for panels that contain a LabelList or MarkupList node. The ListPanel provides the following functional

Lists and Grids Markup

Overview The lists and grids node classes are the foundation of a Roku SceneGraph user interface. You want to carefully select and configure a list or grid node class, and integrate it into your user interface design. You have the following list and grid

Loading and Running Your Application

Table of Contents Enabling Development Mode on your box Before you are able to load a development application to your Roku device, you must enable development mode on your Roku Streaming device. Development mode can be enabled only on a linked device

N**Node**

Table of Contents Description The abstract base class of all SceneGraph nodes and the equivalent of the BrightScript roSG

component. See `roSGNode` for supported interfaces. Node class objects do not draw anything and are skipped in the render

Node Field Observers

All node and component fields can have observers attached to them. These observers continuously monitor the state of the field, and if the field changes, a specified callback function is triggered to perform an action in response to the field st

Non-Certified Channels

The Roku platform supports both public and non-certified channel. Public channels are published in the Roku Channel Store discoverable from the Search feature on the Roku home page. Non-certified channel, however, are used for QA and beta te

P

Packaging Roku Channels

Publishing on the Roku Channel Store requires several core items — such as source code, images, and fonts — to be "packaged". This enables developers to publish channels while keeping all intellectual property safely encrypted. The process of "packaging"

Panel

Table of Contents Extends: Group Description The Panel node is used to create sliding panels for channel UI, similar to those on the Roku OS home screen, Fields Field Type Default Use `panelSize` string narrow Write-Only Specifies one of the default panel

PanelSet

Table of Contents Extends: Group Description The PanelSet node provides the panel sliding behavior seen in the Roku home screen. The PanelSet node manages the position of a set of child Panel nodes, and implements the left and right panel sliding beha

ParallelAnimation

Table of Contents Extends: AnimationBase Description The ParallelAnimation node class allows you to specify that a set of animations should occur simultaneously. The children of a ParallelAnimation node specify the set of animations to be executed. Note

ParentalControlPinPad

Available since firmware version 8. Description ParentalControlPinPad is a variant of the PinPad component. See <https://sdkdocs.roku.com/display/sdkdoc/PinPad>, although it does have a few key differences: The pin, pinLength, and security fields are made priv

Performance FAQ

Table of Contents To find exactly what you are looking for, you may use your browser "FIND" button to locate an item on this page. Question Answer Can I have a background video playing in my channel navigation screen? Yes. A common channel design

PinDialog

Table of Contents Extends: Dialog Description The PinDialog node class is a special type of Dialog node that prompts the user to enter a numeric string. The PinDialog node class includes an internal PinPad node to allow the user to input a numeric value,

PinPad

Table of Contents Extends: Group Description The PinPad node is a simple widget that allows you to enter a fixed length numeric string. The string that is entered is displayed in a set of boxes above the PinPad node keyboard, with each box displaying a

Playing Videos

Table of Contents Playing any type of video requires just one SceneGraph node class: Video. To play a video, you must first ensure the files to be served to the Roku Player: the video files must have been encoded in one of the formats supported by a Roku

Poster

Table of Contents Extends: Group Description The Poster node class draws an image with the top/left corner located at the node local coordinate system. Because the Poster node class extends the Group node class, a Poster node can have child

PosterGrid

Table of Contents Extends: ArrayGrid Also See Related Page: List and Grids Markup - PosterGrid Markup Description The f

node is a simple grid class that can be used to display two-dimensional grids of posters. In addition to the poster, each ite

PosterGrid Markup

Example Application: PosterGridExample.zip Node Class Reference: PosterGrid The PosterGrid node class shows a grid of images that allow a user to select a content item or option from the grid. The postergridscene.xml component file in PosterG

Prioritizing authenticated channels in Roku Search

Overview The Roku Event Dispatcher (RED) is a simple library that allows channels to share in-channel user behaviors and Roku. The event type "Roku_Authenticated" is documented below. New use cases may be introduced in the future as the lib

Processing Power

Overview If your channel is experiencing low frame rate or laggy transitions, Processing Power is probably the culprit. The d CPU between the low end and high end Roku device is large, and will likely only continue to grow as new models are re

Program Statements

Table of Contents DIM name (dim1, dim2, ..., dimK) DIM ("dimension") is a statement that provides a short cut to creating ro objects. It sets variable name to type "roArray", and creates Arrays of Arrays as needed for multi-dimensional arrays. The di

ProgressDialog

Table of Contents Extends: Dialog Description The ProgressDialog node class is a special type of Dialog node that includes title region and a spinning icon as the body of the dialog. The ProgressDialog node class uses a BusySpinner node to dis

Publishing Roku Channels

Once a Roku channel has been designed, developed and thoroughly tested - the final step is publishing on the Roku platform following guide goes over the channel distribution models available on the Roku platform and how to:

R

RadioButtonList

Table of Contents Extends: LabelList Also See Related Page: List and Grids Markup - RadioButtonList Markup Description ` RadioButtonList node class is a simple list class that can be used to display a list of mutually exclusive radio buttons. Each i

RadioButtonList Markup

Example Application: RadioButtonListExample.zip Node Class Reference: RadioButtonList Overview The RadioButtonList n provides a list where a single list item can be exclusively selected by pressing the OK remote key when the item is focused.

Rectangle

Table of Contents Extends: Group Description The Rectangle node class draws a solid color rectangle with the top/left corner rectangle drawn at the origin of the node local coordinate system. Because the Rectangle node class extends the Group no

Remote Control Buttons

The Roku remote is unique to our platform, with a specific set of buttons each triggering different commands. Some buttons, dPad, are likely familiar to any user or developer. Other buttons, like the Instant Replay or Star buttons, might seem a l

Remote Control Events

The SceneGraph architecture supports a notion of remote control key focus. At any time, any node in the SceneGraph node be assigned the remote control key focus. The node with key focus is unique, so when focus is assigned to a node, the curr

Renderable Node Markup

Table of Contents How to Create and Display Node Classes on the Screen As described in SceneGraph XML Guide, there : basic renderable node classes (node classes that draw on the display screen): Rectangle Label Poster Rectangle Markup E Appl

Renderable Nodes

This section contains details of the fundamental SceneGraph node classes that can render to the screen. The rendered scre elements of all SceneGraph API node classes are derived from these fundamental renderable node classes.

Reserved Words

Like other programming languages, BrightScript has a number of reserved words that have specific meanings to the Roku p These words cannot be used for other purposes in your channels. These words include: And Box CreateObject Dim Each E

roAppInfo

roAppInfo returns information about the application. Supported Interfaces ifAppInfo Description roAppInfo retrieves the deve which can be useful during development. It also retrieves manifest values, such as the title and version number, avoiding

roAppManager

The Application Manager APIs set application level attributes, which mostly affect the look-and-feel of the application. The user interface styles gives each application a consistent look-and-feel, but it's often desirable to customize attributes such as colors and fonts.

roArray

An array stores an indexed collection of BrightScript objects. Each entry of an array can be a different type, or they may all be the same type. Supported Interfaces `ifArray` `ifArrayGet` `ifArraySet` `ifEnum` `ifArrayJoin` `ifArraySort` Description An `roArray` is a collection of objects.

roAssociativeArray

An associative array (also known as a map, dictionary or hash table) allows objects to be associated with string keys. Associative arrays are built into the language. They can be accessed implicitly by using the dot or bracket operators, or by calling `fun`.

roAudioGuide

Available since firmware version 7.5 This component is only available on the following devices: Roku Streaming Stick (3600), Roku Express (3700X) and Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku device.

roAudioMetadata

The `roAudioMetadata` component provides developers access to audio file metadata included in many audio files. This component allows developers to use some audiofiles to deliver the information needed to fill out an `roSpringboard` screen without passing the info in a separate `roAudioResource`.

roAudioPlayer

The Audio Player object provides the ability to setup the playing of a series of audio streams. The object accepts an array of meta-data objects, describing the audio and providing url's for accessing each stream. The component understands the following meta-data objects:

roAudioPlayerEvent

The `roAudioPlayer` sends the `roAudioPlayerEvent` with the following predicates that indicate its valid event types: `isListItemSelected` as Boolean A stream has been selected to start playing. `GetIndex()` as Integer Returns index of audio stream. `isStatusMessage` as Boolean A status message has been received.

roAudioResource

The `roAudioResource` allows .wav files to be cached to memory and quickly played at any time. Supported Interfaces `ifAudioResource` Description `roAudioResource` is intended to support short audio clips which need to be played with very little latency. The `roAudioResource` component provides the ability to cache audio files to memory and play them quickly.

roBitmap

The `roBitmap` component contains image data and provides an interface (`ifDraw2D`) for drawing. Bitmaps can be used for a variety of purposes, such as for sprites, compositing, or as double buffers. Supported Interfaces `ifDraw2D` Description `roBitmaps` store image data and provide an interface for drawing.

roBoolean

Object equivalent for intrinsic type Boolean. Supported Interfaces `ifBoolean` `ifToStr` Description This is useful in the following situations: When an object is needed, instead of an intrinsic value. For example, "roList" maintains a list of objects. If a predicate is needed, instead of a Boolean value.

roByteArray

The byte array component is used to contain and manipulate an arbitrary array of bytes. Supported Interfaces `ifByteArray` `ifArrayGet` `ifArraySet` `ifEnum` Description This object contains functions to convert strings to or from a byte array, as well as functions to manipulate the array.

roCaptionRenderer

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roCaptionRendererEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roCECStatusEvent

Available since firmware version 8. Description `roCECStatusEvent` determines active source status for set boxes. Channels to the `roCECStatusEvent` will be notified when the active-source status of the device changes per the CEC message traffic.

roChannelStore

Supported Interfaces `ifChannelStore` `ifSetMessagePort` <http://sdkdocs.roku.com/display/sdkdoc/ifSetMessagePort> `ifGetMessagePort` <http://sdkdocs.roku.com/display/sdkdoc/ifGetMessagePort> Supported Events `roChannelStoreEvent` Description The `roChannelStore` component provides a way to store channel information.

roChannelStoreEvent

The `roChannelStore` sends an `roChannelStoreEvent` in response to a call to any of several `Get*` methods in `ifChannelStore`. The following predicates indicate its valid event types: `isRequestSucceeded()` as Boolean The previous `Get` request has completed successfully.

roCodeRegistrationScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roCodeRegistrationScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roCompositor

The roCompositor allows the composition and animation of multiple roBitmaps and roRegions. Supported Interfaces ifComp Description The roCompositor can create and manage roSprites in a z-ordered list. The sprites can be of arbitrary size and c

roDataGramSocket

The roDataGramSocket component enables Brightscript apps to send and receive UDP packets. The interface is modeled o works much like standard Berkeley sockets. Supported Interfaces ifSocket ifSocketAsync ifSocketStatus ifSocketOption ifSocketCastOpti

roDateTime

The roDateTime provides an interface to obtain the current date/time for the player and manipulate date/times. Supported In ifDateTime Note: some firmware versions may implement ifDateTime as ifRoDateTime. Description This component provide

roDeviceInfo

The roDeviceInfo component provides an interface to obtain attributes about the device. Supported Interfaces ifDeviceInfo ifSetMessagePort ifGetMessagePort Supported Events roDeviceInfoEvent Description These attributes are not changeable script, b

roDeviceInfoEvent

Available since Roku OS 8. The roDeviceInfo component sends the roDeviceInfoEvent with the following predicates that ind valid event types: isStatusMessage() as Boolean Device status has changed. GetInfo() as Object Returns an roAssociative#

roDouble

Object equivalent for intrinsic type 'Double'. Supported Interfaces ifDouble ifToStr Description roDouble is a legacy object na corresponding to the intrinsic Double object. Applications should use Double literal values and/or Double-typed variables d

roEVPcCipher

The EVP Cipher component provides an interface to the OpenSSL EVP library of symmetric cipher commands. The EVP lib provides a high-level interface to cryptographic functions to implement digital "envelopes". Supported Interfaces ifEVPcCiphe

roEVPDigest

The EVP Digest component provides an interface to the OpenSSL EVP library of message digest algorithms. The EVP libra a high-level interface to cryptographic hash functions. Supported Interfaces ifEVPDigest Description roEVPDigest processes

roFileSystem

The roFileSystem component implements common filesystem inspection and modification routines. Supported Interfaces ifFil ifSetMessagePort ifGetMessagePort Supported Events roFileSystemEvent Description All paths are matched case-insensiti

roFileSystemEvent

The roFileSystem component sends the roFileSystemEvent with the following predicates that indicate its valid event types: isStorageDeviceAdded() as Boolean A storage device was inserted in the USB port. GetMessage() as String Returns volum added

roFloat

Object equivalent for intrinsic type 'Float'. Supported Interfaces ifFloat ifToStr Description This is useful in the following situa object exposes the ifFloat interface, that object can be used in any expression that expects an intrinsic val

roFont

roFont represents a particular font, from a font-family (eg. Arial), with a particular pixel size (e.g 20), and a particular boldnes italicness. Supported Interfaces ifFont Description roFont is used in conjunction with roFontRegistry to create and ma

roFontMetrics (deprecated)

This component is deprecated. Developers should use roFont methods (GetOneLineHeight and GetOneLineWidth). The roF object allows you to get display size information for a specific font returned by the roFontRegistry.Get() method. Supported I

roFontRegistry

The roFontRegistry object allows you to create roFont objects, either using the default font or using fonts in TrueType or Op packaged with your application. Supported Interfaces ifFontRegistry Description This object is created with no parame

roFunction

Object equivalent for intrinsic type Function. Supported Interfaces ifFunction ifToStr

roGridScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roGridScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roHdmiHotPlugEvent

The roHdmiStatus sends the roHdmiHotPlugEvent with the following predicates that indicate its valid event types: isHdmiHotPlug Boolean An HDMI hot plug has occurred. GetMessage() as String Returns the string "HdmiHotPlug". GetIndex() as Integer 1

roHdmiStatus

The HDMI status component provides an interface to the current HDMI operational status. Supported Interfaces ifHdmiStatus Description This component allows you to query the status of the HDMI connection for the device through ifHdmiStatus

roHdmiStatusEvent

The roHdmiStatus sends the roHdmiStatusEvent with the following predicates that indicate its valid event types: isHdmiStatusEvent Boolean An HDMI status event has occurred. GetMessage() as String Returns the string "HdmiHotPlug". GetIndex() as Integer 1

roHMAC

The HMAC component provides an interface to the OpenSSL HMAC functions. These functions generate a Message Authentication Code (MAC) based on a key, in such a way that no one without the key could plausibly generate the MAC. HMAC uses a digital

roHttpAgent

Available since firmware version 7.2 Supported Interfaces ifHttpAgent Description All Scene Graph nodes can use the roHttpAgent component to support cookies, custom HTTP headers, and support secure HTTP file transfer protocols, such as passing certificates

rolmageCanvas

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

rolmageCanvasEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

rolmageMetadata

The rolmageMetadata component provides developers access to image file metadata included in many .jpg EXIF headers. Supported Interfaces ifImageMetadata Description rolmageMetadata currently only works with local file URLs. This object is created with

rolnput

An rolnput object can be used to receive events sent from a network client using the External Control Protocol (ECP), as defined in the External Control API. Supported Interfaces ifInInput Supported Events rolnputEvent Description See External Control Service

rolnputEvent

The rolnput component sends the rolnputEvent with the following predicates that indicate its valid event types: isInput() as Boolean An input event was received. GetInfo() as Object Returns an roAssociativeArray describing the input event.

rolnt

Object equivalent for intrinsic type Integer. Supported Interfaces ifInt ifIntOps ifToStr Description This is useful in the following situations: When an object is needed, instead of an intrinsic value. For example, "roList" maintains a list of objects. I

rolnvalid

Object equivalent for intrinsic type 'Invalid'. Supported Interfaces ifToStr

roKeyboardScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roKeyboardScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

Roku Advanced Layout Editor

The Roku Advanced Layout Editor (RALE) <http://devtools.web.roku.com/roku-advanced-layout-editor/> is a tool that lets developers dynamically layout the UI elements of a channel for quick prototyping and design purposes. Changes made in the

Roku Advertising Framework

Table of Contents Overview The Roku Advertising Framework (RAF) is a universal video ad solution integrated directly into Roku SDK as a common library. RAF is required for channels that include ads, see Certification <https://developer.roku.com/>

Roku Analytics Component

Available since firmware version 8. Table of Contents Overview Roku analytics component is a component library which implements Google Analytics, Omniture, and Ooyala Analytics, amongst others. The library creates a simple method for using SceneGraph

Roku Channel Manifest

Table of Contents Root Level The root level of all Roku Channels must contain a manifest file (pkg:/manifest <http://pkg/manifest>) containing important attributes for the application. The manifest file must be UTF-8 encoded. The attributes in the manifest

Roku Federated Identity Single Sign-On

Table of Contents Introduction This specification defines a mechanism and protocols for the Roku Federated Identity SSO (FID) service, whereby a Channel Partner can authorize a Roku service to share a federated identity unique to that channel in order

Roku OS Release Notes

Table of Contents SDK Update Release Notes: version 9 - 10/23/2018 Roku OS 9 introduces a new manifest flag which adds improvements to the memory footprint and launch times of Roku SceneGraph (RSG) channels. All RSG channels should set "rsg_version"

Roku Pay and In-Channel Purchasing

Table of Contents Overview of Roku Pay The core services of Roku Pay are the billing of customers for subscriptions and the management of invoicing. You can bill customers within your Roku channel for a subscription or prior to installing your channel

Roku Plugin for Eclipse IDE

Table of Contents Overview of the Roku Plug-In The Roku plug-in for Eclipse gives you the power to quickly create your channel and deploy it to a device for testing. The Roku plug-in for the Eclipse IDE offers channel development in BrightScript, including

Roku SDK Documentation

Welcome to the Roku Software Development Kit (SDK) documentation. This section of our documentation is for programmers who intend to write a custom channel and deploy it to the Channel Store. If you are not a programmer but a content provider or publisher

Roku Search

Table of Contents Roku Search provides another way to convert Roku users into customers of your VOD channel, besides the Channel Store. Roku Search is located on the main menu of the Roku home-screen, and allows Roku users to search for a

Roku's Master UI

While developers have the ability to build and design their channel to meet nearly any aesthetics they prefer, users will still be experiencing your Channel within the Roku UI experience. This Roku Master UI has its own set of conventions and system components

roList

Supported Interfaces `ifList` `ifArray` `ifArrayGet` `ifArraySet` `ifEnum` `ifListToArray` Description The list object implements the interfaces `ifArray`, `ifEnum` and therefore can behave like an array that can dynamically add members. The array operator `[]` can

roListScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roListScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roLocalization

The `roLocalization` object provides functions to assist in localization. This component is available beginning with firmware version 7.0. Supported Interfaces `ifLocalization` Description This object provides functions to assist in localization. It is created

roLongInteger

Object equivalent for intrinsic type `LongInteger`. This is available in firmware 7.0 or later. Supported Interfaces `ifLongInteger` `ifToLongInteger` Description `roLongInteger` is the object name corresponding to the intrinsic `LongInteger` object.

roMessageDialog

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roMessageDialogEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roMessagePort

A Message Port is the place messages (events) are sent. Supported Interfaces `ifMessagePort` Description When using BrightScript you would not call these functions directly. Instead, use the "Wait" BrightScript statement. This object is created with no parameters.

roMicrophone

Available since firmware version 7.6 Supported Interfaces `ifMicrophone` `ifSetMessagePort` `ifGetMessagePort` Supported Events `roMicrophoneEvent` Description The `roMicrophone` API allows channel applications to receive audio data from the user's microphone-supply.

roMicrophoneEvent

Available since firmware version 7.6 The `roMicrophone` component sends the `roMicrophoneEvent` with the following predicates to indicate its valid event types: `IsRecordingInfo()` as Boolean True when the microphone is open. `GetInfo()` as Object Returns microphone-supply.

roOneLineDialog

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roOneLineDialogEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roParagraphScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roParagraphScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roPath

The `roPath` component provides developers an easy way to create valid file system paths. Supported Interfaces `ifPath` `ifString` Description The `roPath` component is a convenience class that implements `ifString` while providing additional validation and validation.

roPinEntryDialog

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roPinEntryDialogEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roPosterScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roPosterScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roRegex

The `roRegex` component provides the powerful regular expression processing of the PCRE library to Brightscript strings. Supported Interfaces `ifRegex` Description Please see the PCRE documentation (<http://www.pcre.org/> <http://www.pcre.org/>) for documentation.

roRegion

The `roRegion` component is used to represent a subsection of a bitmap. Supported Interfaces `ifRegion` Description The region is specified with an x,y, width, and height as well as a time field for use with animated sprites and a wrap field which causes the region to wrap.

roRegistry

The Registry is an area of non-volatile storage where a small number of persistent settings can be stored. Supported Interfaces `ifRegistry` Description The Registry provides a means for an application to write and read small amounts of data such as settings.

roRegistrySection

A Registry Section enables the organization of settings within the registry. Different registry sections may have their own key same name. In other words, key names are scoped within the registry section to which they belong. Supported Interface

roRSA

The RSA component provides an interface to the OpenSSL RSA library of signing algorithms. The RSA component is available in firmware version 4.7 and later. Supported Interfaces ifRSA Description This component can be used to sign/verify using RS

roScreen

The roScreen component provides a full screen drawing surface that can be stacked and that you can receive input events from. Supported Interfaces ifScreen ifDraw2D ifSetMessagePort ifGetMessagePort Supported Events roUniversalControlEvent DeviceKey

roSearchHistory

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSearchScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSearchScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSGNode

Table of Contents The roSGNode object is the BrightScript equivalent of SceneGraph XML file node creation. Supported Interfaces ifAssociativeArray ifSGNodeChildren ifSGNodeField ifSGNodeDict ifSGNodeFocus ifSGNodeBoundingRect ifSGNodeHttpASupp

roSGNodeEvent

An roSGNode object sends roSGNodeEvent messages to a specified port when changes occur in nodes. An roSGNodeEvent is sent as the argument of field observer callback functions. Both of these cases allow a SceneGraph application to respond to

roSGScreen

Table of Contents The roSGScreen object is a SceneGraph canvas that displays the contents of a SceneGraph Scene node. Supported Interfaces ifRoSGScreen Supported Events roSGScreenEvent <https://sdkdocs.roku.com/display/sdkdoc/roSGScreen+Description>

roSGScreenEvent

roSGScreenEvents are events sent to a scene graph roSGScreen by the framework. Other than when notifying the channel's BrightScript thread that the screen is being closed, and thus that the channel should be terminated, channels do not generate

roSlideShow

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSlideShowEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSocketAddress

The roSocketAddress is used by the roStreamSocket and roDataGramSocket components for TCP and UDP traffic respectively. Supported Interfaces ifSocketAddress Description This object is created without any arguments: CreateObject("roSocketAddress") Methods

roSocketEvent

An roStreamSocket or roDataGramSocket object sends the roSocketEvent to indicate a change in the status of the socket. You must enable specific event notifications via the notify methods of ifSocketAsync. GetSocketID() as Integer Returns the ID of

roSpringboardScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSpringboardScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roSprite

Supported Interfaces ifSprite Description The roSprite object cannot be created directly with a CreateObject() call. It must be created with a managing roCompositor object. This association is implicitly created by creating an roSprite object with the

roStreamSocket

The roStreamSocket component enables BrightScript apps to accept and connect to TCP streams as well as send and receive data with them. The interface is modeled on and works much like standard Berkeley sockets. Supported Interfaces ifSocketConn

roString

Object equivalent for intrinsic type 'String'. Supported Interfaces ifString ifStringOps ifToStr Description This is useful in the following situations: When an object is needed, instead of an intrinsic value. For example, "roList" maintains a list of obj

roSystemLog

The roSystemLog component enables the application to receive events from the Roku Streaming Player that are intended for logging errors and trends, rather than trigger a response to a user action. Table of Contents Supported Interfaces ifSystemLog Desc

roSystemLogEvent

roSystemLogEvents are sent when enabled via roSystemLog. roSystemLogEvent has the following method: GetInfo() as Object Returns an AssociativeArray containing information describing the event. All event AAs have the following base keys: LogTy

roTextScreen

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roTextScreenEvent

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during certification. Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

roTextToSpeech

Please note this component is only available on the following devices: Roku Streaming Stick (3600X), Roku Express (3700X), Roku Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS 7.2 and later.

roTextToSpeechEvent

Please note this component is only available on the following devices: Roku Streaming Stick (3600X), Roku Express (3700X), Roku Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS 7.2 and later.

roTextureManager

The Texture Manager provides a set of API's for managing an roBitmap cache. Supported Interfaces ifTextureManager
<http://sdkdocs.roku.com/display/rokdocs/ifSetMessagePort> ifSetMessagePort
<http://sdkdocs.roku.com/display/rokdocs/ifGetMessagePort> ifGetMessagePort

roTextureRequest

An roTextureRequest is used to make requests to the roTextureManager. Supported Interfaces ifTextureRequest ifHttpAger
 Description An roTextureRequest object is created using the CreateObject() method and passing it a URI string.
 CreateObject("roTextureRequest", {uri: "http://example.com/image.png"})

roTextureRequestEvent

The roTextureManager sends the roTextureRequestEvent after completing a request. GetId() as Integer Returns the unique request ID. GetState() as Integer Returns the state of the request. See ifTextureRequest.GetState() for the list of states. Get

roTimespan

The Timespan object provides an interface to a simple timer for tracking the duration of activities. It's useful for tracking how long an activity has taken or if a specified time has elapsed from some starting event. Supported Interfaces ifTimespan Description

roUniversalControlEvent

The roScreen object sends the roUniversalControlEvent with the following related methods. If an app constrains the events to just the roUniversalControlEvent, the app will work with any controller. The GetID(), GetChar(), GetKey(), and IsPress()

roUrlEvent

The roUrlTransfer component sends the roUrlEvent with the following methods: GetInt() as Integer Returns the type of event. The following event types are currently defined: 1 transfer complete 2 transfer started. Headers are available for suitable protocols.

roUrlTransfer

A roUrlTransfer object transfers data to or from remote servers specified by URLs. It can perform mutual authentication with

server. Supported Interfaces [ifUrlTransfer](#) [ifHttpAgent](#) [ifSetMessagePort](#) [ifGetMessagePort](#) Events Supported [roUrlEvent](#) De

[roVideoPlayer](#)

The [roVideoPlayer](#) component implements a video player with more programmatic control, but less user control than the [roVideoPlayerEvent](#) component. Supported Interfaces [ifVideoPlayer](#) [ifHttpAgent](#) [ifSetMessagePort](#) [ifGetMessagePort](#) Supported Events [roVideo](#)

[roVideoPlayerEvent](#)

The [roVideoPlayer](#) sends the [roVideoPlayerEvent](#) with the following predicates that indicate its valid event types: [isStreamS](#) Boolean The video stream has started playing. [GetIndex\(\)](#) as Integer Returns the number of seconds from play to start str

[roVideoScreen](#)

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during cer Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

[roVideoScreenEvent](#)

This component is deprecated. Beginning July 1st, 2017, any new channels using this component will be rejected during cer Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

[RowList](#)

Table of Contents Extends: [ArrayGrid](#) Also See Related Page: [List and Grids Markup - RowList Markup](#) [RowList Overview](#) T node class provides a vertically-scrollable list, containing rows of independent horizontally-scrollable individual items. Each

[RowList Markup](#)

Example Application: [RowListExample.zip](#) Node Class Reference: [RowList](#) The [RowList](#) node class incorporates characteris lists and grids, and allows the same type of custom appearance and behavior of the list items (and the list rows) as in the

[roXMLElement](#)

[roXMLElement](#) is used to contain an XML tree. Supported Interfaces [ifXMLElement](#) Description Example: `<tag1>this is some text</tag1>` Would parse such that: Name = "tag1" Attributes = invalid Body = [roString](#) with "this is some text" Example:

[roXMLList](#)

Contains a list of [roXML](#) objects. Supported Interfaces [ifList](#) [ifListToArray](#) <https://sdkdocs.roku.com/display/sdkdoc/ifListToAr> [ifXMLList](#) Description Normally [roXMLList](#) objects are not created via [CreateObject\(\)](#), but are returned from various [ifXMLElement](#)

[Runtime Functions](#)

Table of Contents [CreateObject\(classname as String, \[optional parameters\]\) as Object](#) Creates a BrightScript Component of classname specified. Return invalid if the object creation fails. Some Objects have optional parameters in their constructor th

T

[TargetGroup](#)

Available since firmware version 7.5 Table of Contents Extends: [Group](#) Description The [TargetGroup](#) node class associates rectangular regions that children of the group will occupy. Like [MarkupList](#), the [TargetGroup](#) has a content field containing th

[TargetList](#)

Available since firmware version 7.5 Table of Contents Extends: [TargetGroup](#) <https://sdkdocs.roku.com/display/sdkdoc/Targ> Description The [TargetList](#) node class adds useful functionality to the [TargetGroup](#) node by making is easy to set up lists an

[TargetSet](#)

Available since firmware version 7.5 Table of Contents Extends: [Node](#) <https://sdkdocs.roku.com/display/sdkdoc/node> Descri [TargetSet](#) node class is used to specify a set of target regions where items in a [TargetGroup](#) <https://sdkdocs.roku.com/displ>

[Task](#)

Table of Contents Extends: [Node](#) Description The [Task](#) node class allows you to specify a function to be spawned in a differ and run asynchronously with respect to both the scene rendering thread and the main application thread. A [Task](#) node also

[Testing In-Channel Purchasing](#)

Table of Contents Overview of Testing In-Channel Products Developers can test billing for an in-channel product on a side-l channel using the "Use for billing testing" feature, eliminating the need to create a private channel to see if billing produc

[Text to Speech](#)

Note: This feature is only available on the following devices: Roku Streaming Stick (3600X), Roku Express (3700X) and Exp (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS version : later.

[TextEditBox](#)

Table of Contents Extends: Group Description The `TextEditText` node class is intended to display a string of characters as they are typed. When focused, it displays a flashing cursor to indicate the text insertion position. `TextEditText` nodes are automatically

Texture Memory

Table of Contents Overview A common offense is simply using too much texture memory. It's important to realize that simply using a larger image does not necessarily ensure a higher quality end result. Rather, using unnecessarily large images will more often than not

The Roku Channel Developer Program

Table of Contents Roku offers content owners and publishers the ability to create their own channels on the Roku devices and make those channels available to Roku users. The Roku SDK documentation contains the necessary information for creating a Roku channel.

Theme attributes for views

SceneGraph Developer Extensions (SGDEX) support customizing elements in the views. Using the theme attributes `SetThemeAttributes` at the start of a channel in the `Show(args)` function. Do not set global theme attributes before opening each view.

Timer

Table of Contents Extends: Node Description The `Timer` node class generates an observable event after a specified amount of time has elapsed. Fields `Field Type Default Use control string "none"` Used to control the operation of the `Timer` node. Recognize

Top Development Tips for the Roku Platform

Make sure you capture all of the events sent by the `roVideoScreen` or you may miss important playability information. Know the control codes for special screens: Dump Core: Home 5x, Up, Rew 2x, FF 2x Debug Info on screen: Home 5x, Rew 3x, FF 2x

Transitioning to SceneGraph

The following summarizes how to transition from the older Roku API to SceneGraph applications. In general, SceneGraph handles events without requiring you to write custom event loops, and allows you to easily add custom event handlers by observing virtual events.

Trick Mode Support

Table of Contents Two types of trick mode support (FF/REW/SEEK) are provided. For developers who generate and publish Roku archives in the Roku BIF (Base Index Frame) file format, scene-based trick-mode using index frames will be supported. A sample

Tutorial Overview

Every ZIP file example in this tutorial is a complete application, in that it consists of at least a scene component (a component derived from a scene node class, such as `Scene`, or `OverhangPanelSetScene`), and possibly any other components needed for the application.

Typographic Nodes

The following node classes allow you to set the font characteristics of the text in a SceneGraph application, and display scrolling text.

Typography

The SceneGraph API includes the `Font` node class for specifying the font characteristics to be used by your application. Each class that renders text on the screen display includes a `font` field to set the font characteristics for the node. There is a

Typography Markup

Table of Contents Roku SceneGraph offers two special node classes to display text: `ScrollingLabel` `ScrollableText` `ScrollingLabelMarkup` Example Application: `ScrollingLabelExample.zip` Node Class Reference: `ScrollingLabel` The `ScrollingLabel` node class

V

Vector2DFieldInterpolator

Table of Contents Extends: Node Description `Vector2DFieldInterpolator` specifies a keyframe animation sequence to be applied to a pair of `Vector2D` fields of a node. Most typically, this is used to animate the (x,y) coordinates of a node's translation field. A

Video

Table of Contents Extends: Group Description The `Video` node class provides a controlled play of live or VOD video. The `Video` node includes a wide variety of internal nodes to support trick play, playback buffering indicators, and so forth. Playback buffer

X

XML Components

An XML-based markup language has been added to the SceneGraph API that allows new SceneGraph components to be defined consisting of a set of SceneGraph nodes loaded from a declarative description, with interactive or animated behaviors added.

XML Elements

The following are the component XML file elements used in SceneGraph applications.

Z

Z-Order/Parent-Child Markup

[Table of Contents](#) This section of the tutorial shows the sequence that several renderable nodes are drawn on the screen, a general concepts of inheritance of properties in child nodes from parent nodes. You'll see how to organize a node tree to achieve a desired Z-order.

ZoomRowList

Available since firmware version 9 Extends: [ArrayGrid](https://sdkdocs.roku.com/display/sdkdoc/ArrayGrid) Overview The ZoomRowList node allows a row of the Row-Row Grid to smoothly zoom up to a larger size when that row has focus. Rows in this node are rendered in the order they are defined in the XML file.
