

ifDeviceInfo

Implemented By

- roDeviceInfo

Supported Methods

- Device Properties
 - GetModel() as String
 - GetModelDisplayName() as String
 - GetModelType() as String
 - GetModelDetails() as Object
 - GetFriendlyName() as String
 - GetVersion() as String
 - GetDeviceUniqueId() as String
 - GetAdvertisingId() as String
 - GetRIDA()
 - IsAdIdTrackingDisabled() as Boolean
 - IsRIDADisabled() as Boolean
 - GetClientTrackingId() as String
 - GetChannelClientId() as String
 - GetUserCountryCode() as String
 - GetRandomUUID() as String
 - GetTimeZone() as String
 - HasFeature(feature as String) as Boolean
 - GetCurrentLocale() as String
 - GetCountryCode() as String
 - TimeSinceLastKeypress() as Integer
 - GetDrmlInfo() as Object
 - GetDrmlInfoEx() as Object
 - GetCaptionsMode() as String
 - SetCaptionsMode(mode As String) as Boolean
 - GetCaptionsOption(option as String) as String
 - GetClockFormat() as Integer
 - EnableAppFocusEvent(enable as Boolean)
 - EnableScreensaverExitedEvent(enable as Boolean)
 - IsHDMIConnected() as Boolean
 - HasFeature("energy_star_compliant") as Boolean
 - EnableLowGeneralMemoryEvent(enabled as Boolean)
 - GetGeneralMemoryLevel() as String
- Network Info
 - GetLinkStatus() as Boolean
 - EnableLinkStatusEvent(enable as Boolean) as Boolean
 - GetConnectionType() as String
 - GetExternalIp() as String
 - GetIPAddr() as Object
 - GetConnectionInfo() as Object
- Video Info
 - GetDisplayType() as String
 - GetDisplayMode() as String
 - GetDisplayAspectRatio() as String
 - GetDisplaySize() as Object
 - GetVideoMode() as String
 - GetDisplayProperties() as Object

- `GetSupportedGraphicsResolutions()` as Object
- `CanDecodeVideo(video_format as Object)` as Object
- `GetUIResolution()` as Object
- `GetGraphicsPlatform()` as String
- `GetVideoDecodeInfo()` as Object
- Audio Info
 - `GetAudioOutputChannel()` as String
 - `GetAudioDecodeInfo()` as Object
 - `CanDecodeAudio(audio_format as Object)` as Object
 - `GetSoundEffectsVolume()` as Integer
 - `IsAudioGuideEnabled()`
 - `EnableAudioGuideChangedEvent(enable as Boolean)`

Device Properties

GetModel() as String

Returns the model name for the Roku Streaming Player device running the script. This is a five-character alphanumeric string; for example, "3050X". Please see [Roku Models and Features](#) of the Developer Guide for a list of the current and classic models.

GetModelDisplayName() as String

Returns the model display name for the Roku Streaming Player device running the script (for example, "Roku 2 XD").

GetModelType() as String

Returns a string describing what type of device it is. For future compatibility, the caller should by default assume "STB" when anything other than described value is returned. Current possible values are:

"STB" Set-top box type device.
"TV" Roku TV type device.

GetModelDetails() as Object

Returns an associative array containing more information about the device model. The following keys are defined:

`VendorName` - string describing model vendor
`ModelNumber` - string describing model number

GetFriendlyName() as String

Returns a string describing the device that may be used for network device selection. The string may be a user-assigned device name or a description of the device such as model name and/or serial number. The string is subject to change and should not be used as a persistent key or ID.

GetVersion() as String

Returns the version number of the Roku Streaming Player firmware running on the device. This is a 13 character string; for example "034.08E01185A". The third through sixth characters are the major/minor version number ("4.08") and the ninth through twelfth are the build

number ("1185").

GetDeviceUniqueID() as String

Returns the serial number of the unit running the script. This is a 12 character alphanumeric string. While this value was traditionally used to uniquely identify a device, we now recommend using the publisher ID for this purpose. See [GetClientTrackingId](#) for more details.

GetDeviceUniqueID() is deprecated and will be removed from the firmware in the Spring 2019 OS update. **Developers must update their channels to use the 12-character alphanumeric unique identifier returned by [GetChannelClientID\(\)](#).**

GetAdvertisingId() as String

Returns a unique identifier of the unit running the script. The string returned is a Universally Unique Identifier (UUID) as specified in IETF-RFC 4122 with 36 characters (32 alphanumeric characters and four hyphens). The characters are grouped in the form 8-4-4-4-12, for example "123e4567-e89b-12d3-a456-426655440000". This identifier is persistent but can be reset by the user from the device's Settings menu or by performing a factory reset on the device. If the user has disabled Ad ID tracking from the settings menu, then this identifier should not be used for targeted advertising. [IsAdIdTrackingDisabled\(\)](#) should be called to check if the user has disabled Ad ID tracking.

GetAdvertisingId() is deprecated and will be removed from the firmware in the Spring 2019 OS update. **Developers must update their channels to use the replacement API [GetRIDA\(\)](#) to get the unique identifier.**

GetRIDA()

Available since firmware 8.1

[GetRIDA\(\)](#) returns a unique identifier of the unit running the script. The string returned is a Universally Unique Identifier (UUID). This identifier is persistent but can be reset by the user from the device's Settings menu or by performing a factory reset on the device.

If the user has set "Limit ad tracking" (RIDA is disabled) from the Settings menu, then this identifier should not be used for targeted advertising.

Additionally, if the viewer's country is an EU member country, any data collection must be compliant with the EU General Data Protection Regulation.

[IsRIDADisabled\(\)](#) should be called to check if the user has disabled RIDA tracking and [GetUserCountryCode\(\)](#) should be called to check the user's country.

IsAdIdTrackingDisabled() as Boolean

Returns True if the user has disabled Ad Id tracking by selecting "Limit ad tracking" from the Roku Settings menu. If Ad Id tracking is disabled, the identifier returned by [GetAdvertisingId\(\)](#) should not be used for Ad targeting.

[IsAdIdTrackingDisabled\(\)](#) is deprecated and will be removed from the firmware in the Spring 2019 OS update. **Developers must update their channels to use the replacement API [IsRIDADisabled\(\)](#) to know the Ad Id tracking status.**

IsRIDADisabled() as Boolean

Available since firmware 8.1

[IsRIDADisabled\(\)](#) returns True if the user has disabled RIDA tracking by selecting "Limit tracking" from the Roku Settings menu.

GetClientTrackingId() as String

Returns a unique identifier of the unit running the script. This identifier is different across channels so each channel will get a different identifier when calling this function. This id is persistent and cannot be reset; It is, therefore, the recommended alternative to the ESN returned by `GetDeviceUniqueId()`. One example of how this value can be used by content partners is to manage or identify devices linked to their content services. Prior to firmware version 7.6, this function was known as `GetPublisherId()`.

`IsGetClientTrackingId()` is deprecated and will be removed from the firmware in the Spring 2019 OS update. **Developers must update their channels to use the replacement API `GetChannelClientId()` to get the unique identifier.**

GetChannelClientId() as String

Available since firmware 8.1

`GetChannelClientId()` returns a unique identifier of the unit running the script. This identifier is different across channels so each channel will get a different identifier when calling this function.

The ID is persistent and cannot be reset. It is, therefore, the recommended alternative to the ESN returned by `GetDeviceUniqueId()`.

This value can be used to manage or identify devices linked to the channel's content services.

GetUserCountryCode() as String

Available since firmware 8.1

To determine the country associated with a user's Roku account, a new method `GetUserCountryCode()` as `String` was added to `roDeviceInfo`. Typically, the value returned will be an ISO 3166-1 (2-letter) country code representing the country.

Note: If the channel owner entered into an additional agreement to have the channel published to a curated [Roku Powered Channel Store](#) instead of the user country, a *Roku Powered Channel Store Identifier* will instead be returned:

Roku Powered Channel Store	Roku Powered Channel Store Identifier	Country
Econet	Econet	ZW
Globe	globe	PH
PLDT	PLDT	PH
Sky Germany	skyde	DE
Sky Spain	skyes	ES
Sky Ireland	skyie	IE
Sky Italy	skyit	IT
Sky UK	skyuk	UK
Telstra	Telstra	AU

A future enhancement to `getUserCountryCode()` will return the 2-letter country code instead of the Roku Powered Channel Store Identifier. We, therefore, recommend that channels utilize both to avoid having to update later.

GetRandomUUID() as String

Returns a randomly generated unique identifier. The string returned is a Universally Unique Identifier (UUID) version 4 as specified in IETF-RFC 4122 with 36 characters (32 alphanumeric characters and four hyphens). The characters are grouped in the form 8-4-4-4-12, for example "123e4567-e89b-12d3-a456-426655440000". Each time this function is called, a different identifier is returned.

GetTimeZone() as String

Returns a string representing the user's current system time zone setting. Current possible values are:

- "US/Puerto Rico-Virgin Islands"
- "US/Guam"
- "US/Samoa"
- "US/Hawaii"
- "US/Aleutian"
- "US/Alaska"
- "US/Pacific"
- "US/Arizona"
- "US/Mountain"
- "US/Central"
- "US/Eastern"
- "Canada/Pacific"
- "Canada/Mountain"
- "Canada/Central Standard"
- "Canada/Central"
- "Canada/Eastern"
- "Canada/Atlantic"
- "Canada/Newfoundland"
- "Europe/Iceland"
- "Europe/Ireland"
- "Europe/United Kingdom"
- "Europe/Portugal"
- "Europe/Central European Time"
- "Europe/Greece/Finland"

HasFeature(feature as String) as Boolean

Returns true if the current device/firmware supports the passed in feature string.

Valid features to query for are:

- "5.1_surround_sound"
- "can_output_5.1_surround_sound"
- "sd_only_hardware"
- "usb_hardware"
- "sdcard_hardware"
- "ethernet_hardware"
- "gaming_hardware"

The "1080p_hardware" argument has been deprecated for firmware version 7.0. Channel applications should use the `GetVideoMode()` and `CanDecodeVideo()` functions instead.

GetCurrentLocale() as String

Returns a string representing the current locale based on the user's language setting. The string is an ISO 639-1 (2-letter) language code followed by an underscore and a ISO 3166-1 (2-letter) country code. Current possible values are:

"en_US"	US English
"en_GB"	British English
"fr_CA"	Canadian French
"es_ES"	International Spanish
"de_DE"	German
"it_IT"	Italian

GetCountryCode() as String

Returns a value that designates the Roku Channel Store associated with a user's Roku account. Typically, the value returned will be an ISO 3166-1 (2-letter) country code representing the country. Alternatively, if the channel owner entered into an additional agreement to have the channel published to a curated [Roku Powered Channel Store](#) instead of the user country, then a *Roku Powered Channel Store Identifier* will instead be returned.

Current possible values are:

Value	Country	Roku Powered Channel Store (if applicable)
"CA"	Canada	
"Econet"	Zimbabwe	Econet
"FR"	France	
"GB"	Great Britain and Northern Ireland	
"globe"	Philippines	Globe
"IE"	Republic of Ireland	
"MX"	Mexico	
"OT"	Rest of World	
"PLDT"	Philippines	PLDT
"Telstra"	Australia	Telstra
"skyde"	Germany	Sky Germany
"skyes"	Spain	Sky Spain

"skyie"	Ireland	Sky Ireland
"skyit"	Italy	Sky Italy
"skyuk"	United Kingdom	Sky UK
"US"	United States	

This does not necessarily match the physical location of the device, nor does it necessarily match the last two letters of the current locale string.

TimeSinceLastKeypress() as Integer

Returns the number of seconds since the last remote keypress was received.

GetDrmInfo() as Object

This function is available in firmware 7.0 and above.

Returns an associative array with the supported DRM system and features. For example, a device that supports PlayReady inside a trusted environment with secure stop returns:

```
{"playready": "tee;ss"}
```

The values for the PlayReady key above are:

- tee indicates the core DRM system runs in a Trusted Execution Environment
- ss indicates the DRM system supports secure stop

GetDrmInfo() is deprecated and will be removed from the firmware in the Spring 2019 OS update. **Developers must update their channels to use the replacement API [GetDrmInfoEx\(\)](#) to return the supported DRM system and features.**

GetDrmInfoEx() as Object

Available since firmware version 8.1

A new API, `GetDrmInfoEx()`, returns an associative array with the supported DRM system and features. The return value looks like the following:

Note: tee here indicates the core DRM system runs in a Trusted Execution Environment.

```

PlayReady : {
  multikey: false
  securestop: true
  tee: false
  version: "2.5"
}
Widevine : {
  multikey: true
  securestop: false
  tee: false
  version: "3.5"
}
Verimatrix : {
  multikey: false
  securestop: false
  tee: false
  version: "3.7"
}
AdobeAccess : {
  multikey: false
  securestop: false
  tee: false
  version: "2.0"
}
    
```

GetCaptionsMode() as String

This function returns the current global setting for the Mode property. In other words, this function is used to determine whether global captions are turned on or off, or are in instant replay mode. The possible return values are those listed for the Mode property:

Property	Possible Values
Mode	On Off Instant replay Note: On a Roku TV, when the user selects On Mute this function will return On when the TV is muted and Off when it is not muted.

SetCaptionsMode(mode As String) as Boolean

This function sets the current global setting for the Mode property. In other words, this function is used to determine whether global captions are turned on or off, or are in instant replay mode or enabled onMute. The possible parameter values are those listed for the Mode property. The function returns true if the mode was successfully set.

Property	Possible Values
----------	-----------------

Mode	On Off Instant replay When mute (Roku tv only)
-------------	---

GetCaptionsOption(option as String) as String

This function returns the current value of the specified global setting property. The value returned is one of the possible values for the specified property. The option argument is passed to the GetCaptionsOption() function can be one of the following:

Property to Query	GetCaptionsOption() Argument
Text style	Text/Font
Text edge effect	Text/Effect
Text size	Text/Size
Text color	Text/Color
Text opacity	Text/Opacity
Background Color	Background/Color
Background Opacity	Background/Opacity
Window Color	Window/Color
Window Opacity	Window/Opacity

The possible values for each property are:

Property	Possible Values
Text style	Default Serif Fixed Width Serif Proportional Sans Serif Fixed Width Sans Serif Proportional Casual Cursive Small Caps
Text edge effect	Default None Raised Depressed Uniform Drop shadow (left) Drop shadow (right)
Text size	Default Large Medium Small

Text color	Default White Black Red Green Blue Yellow Magenta Cyan
Text opacity	Default 25% 75% 100%
Background Color	Default White Black Red Green Blue Yellow Magenta Cyan
Background Opacity	Default Off 25% 75% 100%
Window Color	Default White Black Red Green Blue Yellow Magenta Cyan
Window Opacity	Default Off 25% 75% 100%

Here is some sample BrightScript code demonstrating how to use the global closed caption setting APIs.

```

di = CreateObject("roDeviceInfo")
mode = di.GetCaptionsMode()

print "Font=";di.GetCaptionsOption("Text/Font")
print "Color=";di.GetCaptionsOption("Text/Color")
print "Size=";di.GetCaptionsOption("Text/Size")
print "Effect=";di.GetCaptionsOption("Text/Effect")
print "Opacity=";di.GetCaptionsOption("Text/Opacity")
print "Background Color=";di.GetCaptionsOption("Background/Color")
print "Background Opacity=";di.GetCaptionsOption("Background/Opacity")
print "Window Color=";di.GetCaptionsOption("Window/Color")
print "Window Opacity=";di.GetCaptionsOption("Window/Opacity")

```

GetClockFormat() as Integer

GetClockFormat lets the channel query whether the system settings for Time (Setting --> System --> Time) is set to a 12 or 24-hour format. The API returns 0 for the 12-hour format and 1 for the 24-hour format.

This function is available in firmware 8.0 and above.

EnableAppFocusEvent(enable as Boolean)

Channels can get notified when a system overlay event (such as the [confirm partner button HUD](#) or the caption control overlay) is displayed. This notification gives the channel the opportunity to do any processing they may want to when the channel loses or regains focus. The default is the overlay event display is disabled (false).

This function is available in firmware 8.0 and above.

EnableScreensaverExitedEvent(enable as Boolean)

Enables (true) or disables (false) sending an [roDeviceInfoEvent](#) when a user has exited the screensaver. The default is disabled (false).

To receive events, you must have first called [SetMessagePort](#) on the [roDeviceInfo](#) object specifying the message port that is to receive the events.

This function is available in firmware 7.5 and above.

IsHDMIConnected() as Boolean

The [IsHDMIConnected\(\)](#) function is deprecated in firmware version 7.0. Channel applications should use the [ifHdmiStatus](#) interface functions instead.

Returns true if an HDMI connection to a TV has been detected.

HasFeature("energy_star_compliant") as Boolean

Available since firmware version 8.1

To determine if a Roku powered TV is Energy Star Compliant use the `roDeviceInfo.HasFeature("energy_star_compliant")`. The type boolean API returns True if the Roku TV is Energy Star Compliant and False otherwise.

EnableLowGeneralMemoryEvent(enabled as Boolean)

Available since firmware version 8.1

If enabled, requests the OS to send a `roDeviceInfoEvent` with `generalMemoryLevel` field set.

GetGeneralMemoryLevel() as String

Available since firmware version 8.1

Returns "normal", "low", or "critical" depending on the general memory levels for the application.

Network Info

GetLinkStatus() as Boolean

Returns true if the player seems to have an active network connection.

EnableLinkStatusEvent(enable as Boolean) as Boolean

Enables or disables sending an `roDeviceInfoEvent` when the network connection status changes. The default is disabled.

To receive events, you must have first called `SetMessagePort` on the `roDeviceInfo` object specifying the message port that is to receive the events.

GetConnectionType() as String

If the unit is connected via WiFi, returns the string "WiFiConnection". If the unit is connected via a wired connection, returns "WiredConnection". If the unit is not connected, returns an empty string.

GetExternalIp() as String

Returns the external IP address of the Roku player. This is the address seen by the Internet and all other computers outside your local network.

GetIPAddr() as Object

Returns `roAssociativeArray`. Each key in the AA is the name of a network interface and the value is the IP-address of the interface. Normally there will be only one interface in the AA.

Provides a way for your application to get the local IP address of the Roku box. This can be used in conjunction with the ECP (see the External Control Protocol Guide) command "**launch**" (or "**install**" for uninstalled channels) to start a different channel from the current channel.

GetConnectionInfo() as Object

Returns an Associative Array with these entries:

name	value
type	Same as the value returned from <code>GetConnectionType()</code>
name	Name of the connection interface
ip	IP Address used by the connection
mac	MAC address of the connection's hardware
ssid	The SSID of the Access Point (present only if type = "WiFiConnection")
gateway	IP Address of the connection gateway (usually the router)
dns.0	IP Address of first DNS server associated with the connection
dns.1	IP Address of the second DNS server, if any (Similarly for any subsequent DNS servers)

This function is available in firmware 6.1 and above.

Video Info

GetDisplayType() as String

Returns the text corresponding to the button selection in the Player Info Settings/Display Type page. Either "HDTV", "4:3 standard", or "16:9 anamorphic".

GetDisplayMode() as String

Returns the configured graphics layer resolution: "480i" or "480p" (if the `ui_resolutions` manifest entry includes `sd` as a supported resolution, otherwise "720p" is returned), "720p", or "1080p" (if the `ui_resolutions` manifest file entry includes `hd` as a supported resolution, otherwise "720p" is returned).

GetDisplayAspectRatio() as String

Returns "4x3" or "16x9"

GetDisplaySize() as Object

Returns an `roAssociativeArray` with keys "w" and "h" that contain the values for the screen width and height respectively, either 720 and 480, or 1280 and 720. Example: { w:1280, h:720 }.

GetVideoMode() as String

Returns a string representing the video playback resolution. The possible strings are:

String	Resolution	Aspect Ratio	Refresh Rate	Bit Depth
"480i"	720x480	4:3	60 Hz	8 Bit

"480p"	720x480	4:3	60 Hz	8 Bit
"576i25"	720x576	4:3	25 Hz	8 Bit
"576p50"	720x576	4:3	50 Hz	8 Bit
"576p60"	720x576	4:3	60 Hz	8 Bit
"720p50"	1280x720	16:9	50 Hz	8 Bit
"720p"	1280x720	16:9	60 Hz	8 Bit
"1080i50"	1920x1080	16:9	50 Hz	8 Bit
"1080i"	1920x1080	16:9	60 Hz	8 Bit
"1080p24"	1920x1080	16:9	24 Hz	8 Bit
"1080p25"	1920x1080	16:9	25 Hz	8 Bit
"1080p30"	1920x1080	16:9	30 Hz	8 Bit
"1080p50"	1920x1080	16:9	50 Hz	8 Bit
"1080p"	1920x1080	16:9	60 Hz	8 Bit
"2160p24"	3840x2160	16:9	24 Hz	8 Bit
"2160p25"	3840x2160	16:9	25 Hz	8 Bit
"2160p30"	3840x2160	16:9	30 Hz	8 Bit
"2160p50"	3840x2160	16:9	50 Hz	8 Bit
"2160p60"	3840x2160	16:9	60 Hz	8 Bit
"2160p24b10"	3840x2160	16:9	24 Hz	10 Bit
"2160p25b10"	3840x2160	16:9	25 Hz	10 Bit
"2160p30b10"	3840x2160	16:9	30 Hz	10 Bit
"2160p50b10"	3840x2160	16:9	50 Hz	10 Bit
"2160p60b10"	3840x2160	16:9	60 Hz	10 Bit

getDisplayProperties() as Object

Returns an `roAssociativeArray` with the following key/value pairs:

Key	Type	Value
Width	integer	Physical width of the attached display in centimeters
Height	integer	Physical height of the attached display in centimeters
Internal	Boolean	Set to true if the display is part of the Roku Player (such as a Roku TV), false otherwise
Hdr10	Boolean	Set to true if the attached display supports HDR 10, false otherwise
DolbyVision	Boolean	Set to true if the attached display supports Dolby Vision, false otherwise

This function is available in firmware 7.0 and above.

GetSupportedGraphicsResolutions() as Object

Return the list of supported graphics resolutions as a list of roAssociative arrays. Each array has the following keys:

Key	Type	Value
width	integer	The pixel width of the supported resolution
height	integer	The pixel height of the supported resolution
name	string	Either SD, HD, or FHD
ui	boolean	True if this resolution is the current Roku UI resolution
preferred	boolean	True if this is the preferred UI resolution, i.e., if this is the optimal resolution for the device.

This function is available in firmware 7.0 and above.

CanDecodeVideo(video_format as Object) as Object

Checks if the Roku Player can decode and play a video format specified as an associative array, and returns an associative array that includes a Boolean value indicating if the video format can be played, and the closest video format supported by the Roku Player.

For example, if the application wants to check if the Roku Player can play an AVC stream at high profile and level 4.2, it calls CanDecodeVideo() with the following video_format:

```
{Codec: "mpeg4 avc", Profile: "high", Level: "4.2"}
```

If the Roku Player cannot play that video format, it will return false, and return the closest video format it can play, with the changed fields, such as:

```
{"Result":false,"Updated": "level;profile", "Codec": "mpeg4 avc", "Profile": "main", "Level": "4.1"}
```

The return value shows the Roku Player cannot play requested video format, shows the updated keys of the requested video format (level and profile) that it can support, and the all the key values of the requested video format supported by the Roku Player. Format keys that are not provided by the caller are not taken into account and not updated. For example, calling CanDecodeVideo() with a format description that has only a codec key (such as {"codec": "vp9"}) will return that same format if the device can decode and play that codec at all, even if the decode capability is limited to one specific container, profile, and level.

The following are the keys of the requested video format and supported video format associative arrays:

Key	Type	Value	Required
Codec	string	Specifies the video codec: "mpeg2", "mpeg4 avc", "hevc", "vp9"	Required
Profile	string	Specifies the profile	Optional
Level	string	Specifies the level	Optional
Container	string	Specifies the container format: "mp4", "hls", "mkv", "ism", "dash", "ts"	Optional

Codec	Profile	Level
"mpeg2"	n/a	"main", "high"
"mpeg4 avc" ¹	"main", "high"	"4.1", "4.2"
"hevc" ¹	"main", "main 10"	"4.1", "5.0", "5.1"
"vp9" ¹	"profile 0", "profile 2"	"4.1", "5.0", "5.1"

¹ These codecs can provide additional display info (bit depth, refresh rate, display width, display height) if `codec`, `profile`, and `level` are passed as parameters.

This function is available in firmware 7.0 and above.

GetUIResolution() as Object

Returns an associative array describing the current UI resolution. The associative array contains the following key-value pairs:

Key	Values								
name	The possible values are: <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD</td> <td>standard definition (720x480 screen pixel dimensions)</td> </tr> <tr> <td>HD</td> <td>high-definition (1280x720 screen pixel dimensions)</td> </tr> <tr> <td>FHD</td> <td>full high-definition (1920x1080 screen pixel dimensions)</td> </tr> </tbody> </table>	Value	Meaning	SD	standard definition (720x480 screen pixel dimensions)	HD	high-definition (1280x720 screen pixel dimensions)	FHD	full high-definition (1920x1080 screen pixel dimensions)
Value	Meaning								
SD	standard definition (720x480 screen pixel dimensions)								
HD	high-definition (1280x720 screen pixel dimensions)								
FHD	full high-definition (1920x1080 screen pixel dimensions)								
width	The possible values are: <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>720</td> <td>standard definition screen pixel width</td> </tr> <tr> <td>1280</td> <td>high-definition screen pixel width</td> </tr> <tr> <td>1920</td> <td>full high-definition screen pixel width</td> </tr> </tbody> </table>	Value	Meaning	720	standard definition screen pixel width	1280	high-definition screen pixel width	1920	full high-definition screen pixel width
Value	Meaning								
720	standard definition screen pixel width								
1280	high-definition screen pixel width								
1920	full high-definition screen pixel width								
height	The possible values are: <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>480</td> <td>standard definition screen pixel height</td> </tr> <tr> <td>720</td> <td>high-definition screen pixel height</td> </tr> <tr> <td>1080</td> <td>full high-definition screen pixel height</td> </tr> </tbody> </table>	Value	Meaning	480	standard definition screen pixel height	720	high-definition screen pixel height	1080	full high-definition screen pixel height
Value	Meaning								
480	standard definition screen pixel height								
720	high-definition screen pixel height								
1080	full high-definition screen pixel height								

This function is available in firmware 7.0 and above.

GetGraphicsPlatform() as String

Returns a string specifying the device's graphics platform, either `opengl` or `directfb`.

This function is available in firmware 7.6 and above.

GetVideoDecodeInfo() as Object



This function is deprecated as of firmware version 7.0. Channel applications should use the `CanDecodeVideo()` function instead.

Returns an `roAssociativeArray` with the EDID (EIA.2FCEA-861) information describing the video display.

See http://en.wikipedia.org/wiki/Extended_display_identification_data#EIA.2FCEA-861_extension_block for an explanation of the information returned.

Audio Info

GetAudioOutputChannel() as String

Returns a string representing the selected audio output ("Stereo" or "5.1 surround").

GetAudioDecodeInfo() as Object

Returns an `roAssociativeArray` with the EDID (EIA.2FCEA-861) audio decoder information for the device connected to the HDMI port (or the device itself for a Roku TV). Each audio decoder supported by the device is listed, with up to four numbers describing the decoder from the EDID SAD (Short Audio Descriptor). Each value is of the form "<number of channels>:<SAD1>:<SAD2>:<PassThru>:". For example, the name "DD+" may have the value "8:6:0:1" where there are 8 independent audio tracks (7.1 audio), 6 is the SAD1 byte, 0 is the SAD2 byte, and 1 is the binary value that indicates this is a pass-through audio device (not a Roku TV). The SAD1 and SAD2 bytes are interpreted differently for different codecs and more information about their values can be found here: http://en.wikipedia.org/wiki/Extended_display_identification_data#CEA_EDID_Timing_Extension_Version_3_data_format

Here is some sample BrightScript code demonstrating how to determine if the attached device supports Dolby Digital Plus audio:

```
di = CreateObject("roDeviceInfo")
audioDecoders = di.GetAudioDecodeInfo()

REM Check for surround sound codecs:
hasDolbyDigital = audioDecoders.doesexist("AC3")
hasDTS = audioDecoders.doesexist("DTS")
hasDDPlus = audioDecoders.doesexist("DD+")
```

Note that the definition of `hasFeature` ("5.1_surround_sound") has changed in firmware v6.1. In previous firmware revisions it returned true when the user set the system audio format to "Surround Sound". In firmware v6.1 and above, it returns true when any of the codecs in the `GetAudioDecodeInfo()` AA has more than 2 audio channels. Users devices will also be default to the "Auto Detect" system HDMI audio setting in v6.1.

CanDecodeAudio(audio_format as Object) as Object

Checks if the Roku Player can decode and play an audio format specified as an associative array, and returns an associative array that includes a Boolean value indicating if the audio format can be played, and the closest audio format supported by the Roku Player. The general format of the associative arrays for `CanDecodeAudio()` is similar to the parameter and return associative arrays used in `CanDecodeVideo()`.

The following are the keys of the requested audio format and supported audio format associative arrays:

Key	Type	Value	Required
-----	------	-------	----------

Codec	string	Specifies the audio codec: "aac", "ac3", "eac3", "alac", "flac", "mp2", "mp3", "vorbis", "wma", "wma pro", "dts".	Required
Profile	string	Specifies the profile.	Optional
ChCnt	integer	Specifies the number of audio channels.	Optional
SampleRate	integer	Specifies the sample rate.	Optional
BitRate	integer	Specifies the bit rate in Kbit/sec.	Optional
Container	string	Specifies the container format.	Optional

This function is available in firmware 7.0 and above.

GetSoundEffectsVolume() as Integer

Returns the user interface sounds effects volume as a percentage. A return value of 0 indicates that UI sound effects are muted, and a value of 100 indicates that they are set to the maximum volume level.

This function is available in firmware 7.0 and above.

IsAudioGuideEnabled()

Returns true if Audio Guide is enabled (on supported devices), otherwise false.

Audio Guide is available on: Roku Streaming Stick (3600X), Roku Express (3700X) and Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS version 7.5 and later.

This function is available in firmware 7.5 and above.

EnableAudioGuideChangedEvent(enable as Boolean)

Enables (true) or disables (false) sending an [roDeviceInfoEvent](#) when Audio Guide is enabled. The default is disabled (false).

To receive events, you must have first called `SetMessagePort` on the `roDeviceInfo` object specifying the message port that is to receive the events.

Audio Guide is available on: Roku Streaming Stick (3600X), Roku Express (3700X) and Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS version 7.5 and above.

This function is available in firmware 7.5 and above.