

Roku Advertising Framework

Table of Contents

- Overview
 - Advantages
 - Features
 - Nielsen DAR Agreement
 - Revenue Share
 - Inventory Split
 - Revenue Split
 - Integrating RAF In Your Channel
 - Sample Channels
 - Release Notes
 - Version 2.5 – 05/2018
 - Version 2.4 – 03/2018
 - Version 2.3 – 10/2017
 - Version 2.2 – 07/2017
 - Version 2.1 – 05/2017
 - Version 2.0 – 03/2017
 - Version 1.9 – 11/2016
 - Version 1.8 – 10/2016
 - Version 1.7 – 06/2016
 - Version 1.6 – 03/2016
 - Version 1.5 – 12/2015
 - Version 1.4 – 10/2015
-

Overview

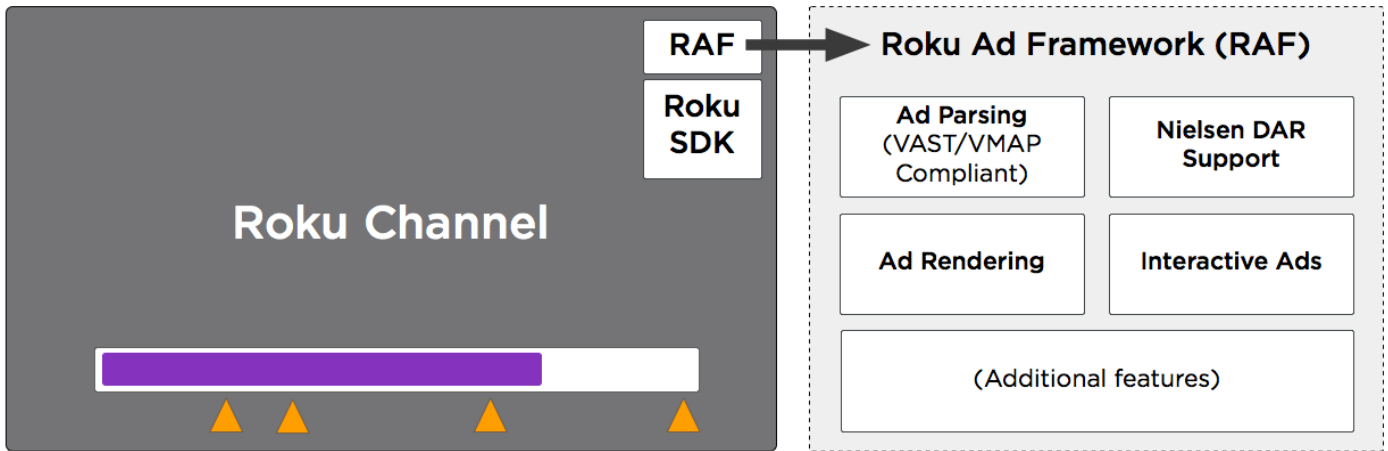
The Roku Advertising Framework (RAF) is a universal video ad solution integrated directly into the core Roku SDK as a common library. RAF is required for channels that include ads, see [Certification](#).

RAF natively integrates baseline and advanced advertising capabilities, including:

- IAB¹ VAST² 2.0/3.0 processing
- IAB VMAP³ playlist management
- Interactive ad units
- Roku's new privacy-friendly device ID ("RIDA")⁴
- Nielsen DAR Support - a comprehensive, next-day view of your ad's online and mobile audience in a way comparable to the Nielsen TV ratings

Roku Developers

The Roku Ad Framework (RAF)



Other benefits of the framework include:

- Allows developers to continue using their preferred ad server (e.g. Freewheel, DFP)
- Automatically updates without rebuilding/resubmitting your app
- Standardizes client-side ad insertion across all apps

The Roku Advertising Framework is intended to provide advanced advertising fulfillment and rendering capabilities to applications. The library supports a variety of ad services, and rendering of both video ads and interactive ads. By deploying as a common library, we reduce the overhead of implementation and maintenance for individual channel application developers, and provide a consistent user experience for all applications. The library is designed to support multiple use cases, providing applications with flexibility to control as little or as much of the ad rendering process as desired.

¹The International Advertising Bureau is an organization comprised of 650+ media and technology companies. The IAB evaluates and recommends standards and practices and conducts research on interactive advertising.

²Video Ad Serving Template is an IAB specification for a universal XML schema for serving ads to digital video players.

³Video Multiple Ad Playlist is an IAB specification for an XML template that video content owners can use to describe the structure for ad inventory insertion. It is commonly used in conjunction with VAST to structure ads into ad pods.

⁴[RIDA] The Roku ID for Advertising is a device identifier to track activity for development and marketing purposes. It is designed to generally follow the guidelines established for the IDFA (Identifier for Advertising) used by other platforms. The RIDA limits disclosure of users' identifying information and allows the ability to opt-out of remarketing or reset the ID at any time. The RIDA ID must only be passed if "limit ad tracking" is not set in the Roku Settings UI. See [GetAdvertisingId\(\)](#), [IsAdIdTrackingDisabled\(\)](#), and [GetPublisherId\(\)](#) for details.

Advantages

The Roku Advertising Framework (RAF) positions the publisher for the future of video advertising on the Roku platform. RAF exists as a universal ad structure across all channels and Roku devices. All channels that have integrated RAF will benefit from Roku's development of new ad features.

- RAF eases the inherent complexities of enabling a Roku channel for video advertising, while providing a robust and powerful feature set. From the ground up, RAF is designed for simple integration.
- Implementation of the framework is mutually beneficial to all involved parties, so will grow the AVOD ecosystem of consumers, advertisers, and channels:
 - Consumers – Better ad experience (e.g., interactivity, frequency capping) means significantly increased consumption of free content on AVOD channels.

- Advertisers – RAF addresses long-standing obstacles for advertisers who want to buy media on the Roku platform.
- Channels – Superior tools for additional possibilities to monetize publisher inventory.
- New features added to RAF will not require further integration work, as the functionality resides within the Roku OS.
- RAF and all further RAF releases are certified to function on the Roku platform, and will receive full support from Roku.
- RAF is fully compatible with any VAST 2.0 ad server, and replaces the existing VAST library code.

Features

- Automatic appending of key targeting parameters in the ad request (Roku ID for Advertising, content genre, display dimensions, etc.)
- IAB VAST 2.0/3.0 parsing
- IAB VMAP parsing
- FreeWheel SmartXML⁵ parsing
- Interactive ad rendering
- Audience measurement

VAST 2.0 Feature	Supported	VAST 3.0 Feature	Supported	VMAP Feature	Supported
Wrapper and Inline Ads	Y	Ad pods ⁶ via sequence attribute	Y	Ad pods playlists	Y
Tracking Events	Y	Extended error tracking	Y	Tracking events	Y
Linear Ads	Y	VAST tracking macros	Y	'repeatAfter' AdBreak attribute	N
ClickThrough (interactive ads)	Y	Ad "buffer" selection	Y	Extension elements	N
Companion Ads	"image/jpeg", "image/png" (parsing only, not rendered)	Skippable linear ads	N		
MediaFile	"video/mp4", "video/x-mp4", "application/x-mpegurl", "application/json"	OBA ⁷ industry icon	N		
Non-Linear Ads	N				
Extension elements	N				

⁵FreeWheel's proprietary standard combines ad payload and ad scheduling in a single XML file.

⁶In the context of ad delivery, a pod is a sequence of ads that is rendered consecutively during a single ad break. An ad pod may also consist of a single ad.

⁷Online Behavioral Advertising refers to the practice of collecting information about online activity to target ads based on relevance.

Nielsen DAR Agreement

Roku Ads are integrated with Nielsen audience measurement and is required for all Ad-supported channels. When integrating the Roku Ad Framework, you acknowledge and agree to the following:

- (i) that turning on these features will cause device and content viewing information provided by your Roku Channel to the Roku SDK to be automatically appended to Nielsen digital ad ratings ("DAR") beacons received in VAST responses and sent to Nielsen;
- (ii) Nielsen may have personally identifiable information ("PII") on certain end users of the Roku platform;
- (iii) these features will enable third party advertisers, agencies and media resellers (including Roku's ad sales team if Roku is a reseller of your ad inventory) to serve VAST responses with Nielsen DAR campaign beacons into your channel, and receive reports on the performance of those DAR campaigns; and
- (iv) if you are not a Nielsen DAR customer, you may not receive reports on DAR campaigns. If you choose to turn on the Nielsen audience

measurement features, you hereby represent and warrant:

- (i) that you will notify your users of the occurrence of audience measurement;
- (ii) you have and will maintain a legally adequate privacy policy;
- (iii) you have and will maintain all necessary rights from Nielsen and consents from users to use the Nielsen audience measurement features; and
- (iv) your use of the Nielsen audience measurement features will comply with all applicable laws, rules and regulations.

In the event that Roku runs an ad campaign on your channel, Roku may, in its sole discretion, provide to you or your ad agency, as applicable, excerpts of the Nielsen DAR that relate to the delivery and performance of advertisements on your ad inventory. You hereby agree to

- (i) maintain the confidentiality of any Nielsen DAR reports provided to you by Roku; and
- (ii) maintain in the Nielsen DAR reports any sourcing and copyright information provided by Nielsen.

YOU AGREE YOU WILL NOT USE THE NIELSEN AUDIENCE MEASUREMENT SOFTWARE AND FEATURES IN CONNECTION WITH CONTENT OR CHANNELS DIRECTED TOWARD CHILDREN OR IN CONNECTION WITH USERS KNOWN TO BE CHILDREN. If Roku discovers or determines in its sole discretion that you are using the Nielsen audience measurement software and features in connection with content or channels directed toward children or with users known to be children, Roku reserves the right to disable or otherwise limit functionality.

YOU MAY NOT ENABLE THE NIELSEN AUDIENCE MEASUREMENT FEATURES IF YOU DO NOT AGREE TO ABOVE. PLEASE CONTACT ROKU OR NIELSEN FOR FURTHER INFORMATION.

Revenue Share

A channel publisher can have the Roku Ad sales team sell ads on the channel on their behalf. In such cases, Roku and the channel publisher can agree to share the advertising revenue in one of two ways:

- inventory split, where the total amount of ad inventory for the channel is split by an agreed-upon percentage, with Roku and the channel publisher each selling their respective percentage of the inventory
- revenue split, where all of the ad inventory for the channel is sold by Roku, with an agreed-upon percentage of the revenue returned to the channel publisher by Roku

When implementing RAF for your channel, it is important to set the URL to the correct ad server according to the revenue share agreement between Roku and your channel publisher, if it exists. The ad server URL is set as the argument to `setAdUrl()`.

Inventory Split

For an inventory split agreement, set the URL in `setAdUrl()` to the ad server used by your channel for those ads sold by your channel publisher. For the percentage of inventory served by Roku, please contact advertising@roku.com to request a VAST tag.

Revenue Split

Since by default, any ads served without an ad server URL will be served by a Roku ad server, you can omit the URL argument, or the `setAdUrl()` call entirely, for revenue split agreements.

Integrating RAF In Your Channel

For steps to getting started, use cases, and the full API reference, see [Integrating the Roku Advertising Framework](#).

Sample Channels

The table below provides a number of samples channels to help you get started with your own RAF implementations:

Download File	SDK Version	Description
RAF Video Node Sample	SceneGraph	This sample demonstrates the use of RAF in a Video node with support for pre-, mid-, and post roll ads.
FullRAFSceneGraphSample.zip	SceneGraph	This sample shows how to add RAF to a channel, configure Nielsen, obtain ads and play them as pre-, mid- and post-rolls. It also shows how to import Ads from non-standard feed (neither VMAP, VAST or SMartXML.)
CustomBufferScreenSceneGraphSample.zip	SceneGraph	This sample shows how to change default ad buffering screen background, title and description. It also shows how to create a completely custom buffering screen.
FullRAFSDK1Sample.zip	Legacy SDK	This sample shows how to add RAF to a channel, configure Nielsen, obtain ads and play them as pre-, mid- and post-rolls. It also shows how to import Ads from non-standard feed (neither VMAP, VAST or SMartXML.) This sample also contains examples of custom buffering screens and stitched ads.
CustomBufferingScreenSDK1Sample.zip	Legacy SDK	This sample shows how to change default ad buffering screen background, title and description. It also shows how to create a completely custom buffering screen.
ServerStitchedAdSDK1Sample.zip	Legacy SDK	This sample demonstrates the interactive stitched ads feature. It shows how to configure RAF with interactive ads and process interactive ad events.

For the full RAF integration cases, the samples:

1. Initializes RAF
2. Turns on Nielsen tracking, and configures it with genre, program ID, and content parameters
3. Configures the ad URL using URL macros (See [URL Parameter Macros in Integrating the Roku Advertising Framework](#)) and the `setAdURL()` method
4. Gets the ads using `getAds()` (VAST feed), and renders them using `showAds()`

For non-standard ad responses, ads are imported from a non-standard feed, neither VMAP, VAST or SmartXML. RAF is configured as if for a standard feed, with backfill ads disabled and extra debug output enabled. Ads are parsed from local JSON file, then formatted as an ad pods array, and imported into RAF using the `importAds()` method. After that, the sample checks for particular ads to play by passing video playback events to the RAF `getAds()` method inside event loop. If any ads were returned from `getAds()`, they are rendered using the RAF `showAds()` method. For the Scene Graph example, after `importAds()`, the sample checks for particular ads to play by passing fake video events created with `createPlayPosMsg()` to the RAF `getAds()` method before event-loop (preroll ads) and inside it (midroll/postroll ads). If any ads were returned from `getAds()`, they are rendered using the RAF `showAds()` method.

Release Notes

Version 2.5 – 05/2018

- Major rework of RAF's diagnostic output to [BrightScript console](#)
 - Warning messages (prefixed with "[RAF.err]") are always printed for known potential problems. Note that these are just additional diagnostics - they do not change the library's behavior, as compared to previous versions.
 - Substantially more information is printed when in `setDebugOutput(true)` mode: method call arguments and return values, [URL](#)

- [macros](#) expansion, ad XML/parsed, etc.
- New interactive templates by BrightLine/Innovid
- Bug fixes
- RAF 2.5 is deployed to devices with Roku OS 8.0 and above

Version 2.4 – 03/2018

- New feature: JIT ("Just In Time") ad resolution for VMAP, SmartXML to reduce overhead incurred by prefetching all ad pods before content playback starts
- New feature: RIA ("Roku Interactive Ads") to allow rendering of Roku interactive ad overlays for OTT content (previously only available for ACR on linear content)
- BrightLine bug fixes and performance improvements
- Innovid bug fixes and new templates ("User Satisfaction Survey" and "Skippable" interactive ads)

Version 2.3 – 10/2017

- Add support for BrightLine interactive ads in SSAI+RSG use case
- New interactive ad templates (Innovid)
- Implement ad buffering limit
- Add support for tracking beacons with HTTP HTTPS redirects
- General performance improvements and bug fixes
- RAF 2.3 available in Roku OS 7.7 and above

Version 2.2 – 07/2017

Features:

- Added a native RSG renderer for Brightline interactive ads
- Enabled the firing of tracking events on empty ad breaks (SmartXML and VMAP; relevant to FreeWheel forecasting)

Bug Fixes:

- Fixed the autoscaling of interactive ads for FHD-only RSG apps on a HD UI device
- Fixed an error when the ad response is invalid XML
- Improved the RIDA hashing when "limit ad tracking" is set
- Improved the draining of pending beacons cache, to benefit low memory devices
- Enhanced the handling of non-standard view sizes (RSG)
- Fixed various minor issues

Version 2.1 – 05/2017

Features:

- Added support for comScore vCE campaign measurement service
- Introducing a generalized audience measurement API (see [enableAdMeasurements\(\)](#) for details)
- Support for a new TrueX SAB interactive ad template

Bug Fixes:

- Fix for a display resolution issue when a FHD-only RSG app was playing ad video on a HD UI device
- Miscellaneous other fixes

Version 2.0 – 03/2017

Features:

- Support for RSG apps to use RAF from Task node
- SceneGraph ad rendering support (video ads and Innovid interactive ads)
 - New `view` parameter for `showAds()` - this is required for all SceneGraph applications
- VAST 3.0 "ad buffet" support
- Extended companion ad tag parsing from VAST to allow multiple ad renderers for different companion creatives
- New interactive ad template support
- New `adCompleted` return value for `stitchedAdHandledEvent()`
- New `provider` member for `companionAds` metadata in `Ad Structure`

Bug Fixes:

- Fix in VAST parser to address problem with DFP waterfall containing invalid ads
- Multiple bug fixes to address ad rendering in both SDK1 and RSG apps built with different combinations of supported `ui_resolutions`

Version 1.9 – 11/2016**Features:**

- Freewheel SmartXML adReplica changes
 - Improve forecasting by only resolving ad requests for wrapped creative renditions that are placed into ad slots
 - Respect `replicaId` if specified in the `adReference` tag and a matching replica exists in the `creativeRenditions`, otherwise treat unwrapped renditions as alternate streams
- When Limit Ad Tracking is set by the user, use a new time-scoped ID that is cycled every 30 days to provide the benefits of frequency capping while still respecting the user's desire to avoid ad tracking
- Added an optional new parameter to the `setContentGenre()` API to indicate whether content is targeted for kids
- Added a new content macro, `ROKU_ADS_KIDS_CONTENT`, and modified default/backfill URLs to use this new macro
- Added a new API, `getNielsenContentData()`, that will return an encrypted N-RIDA parameter string for apps wishing to use Nielsen SDK for DCR measurements
- New BrightLine template

Bug Fixes:

- Eclipse plugin compatibility fixes
- Exit key handling fixes
- TrueX and BrightLine bug fixes and enhancements
- Added a fix for BrightLine ads to use cached ad position
- Fixed 3rd-party tags that used improperly-encoded URL fragments by URL-encoding fragment contents
- Modified garbage collection after interactive ad rendering to fix display issue with BrightLine ads

Version 1.8 – 10/2016**Features:**

- Add missing tracking events for plain video ads in server-stitched streams: Impression, Pause, Resume
- Add contextual info for Complete tracking event
- Add companion tracking metadata to Innovid ads, which do not explicitly have a `CompanionAd` tag to distinguish video ad tracking from microsite tracking
- Add 303 error tracking when wrapped VAST returns no ads
- New BrightLine templates
- Merged Innovid renderer changes, including modifications to tracking pixel logic
- New `creativeAdId` metadata field for ads

Bug Fixes:

- Fix crash when replacing RAF macros in URL containing query parameter values without a name
- Correct pod-specific tracking for ad pods in server-stitched streams: PodComplete, PodStart
- Disallow re-rendering of ad pod when pod cache has been updated while rendering the pod (e.g., for TrueX ads)
- Re-purpose "Expand" and "Collapse" ad tracking to refer to microsite interactions for Innovid ads, which do not generally have a separate

- CompanionAd tag in the VAST representation for these additional tracking events
- Ignore repIcald values when specified in SmartXML ad slots, since these always refer to the first replica
- Treat multiple renditions of wrapped ads in SmartXML as replicas
- Override any creative ID set from a wrapped ad with the creativeld attribute in SmartXML, since this is likely more meaningful to the app than the wrapped ID
- Numerous TrueX/BrightLine bug fixes and feature changes
- Track ad render position values to prevent spurious Complete/PodComplete events when exiting microsites (playback of stitched video can resume across ad boundary, resulting in extra tracking pixels being fired)

Version 1.7 – 06/2016

Features:

- New API: setContentMetaData(metadata): allows app to set information about the current content
- Added new HLS MIME type: "application/vnd.apple.mpegurl"
- Added "ai=Roku_Ads_App_ID" to default and backfill ad URLs' cust_params
- Changed macro value of Roku_Ads_Limit_Tracking to "1" or "0" instead of "true" or "false," to accommodate DFP's special LAT values
- Changed handling of invalid messages passed to the event handler for stitched ads to return either the cached ad data if an ad is currently being rendered, or Invalid if no ad is being rendered to accommodate apps that erroneously pass Invalid messages to the handler
- Prioritize MP4 over HLS ad creatives as HLS can take longer than the length of an ad to settle on an acceptable playback bitrate
- Add support for TrueX ad experience and parse new TrueX VAST extensions
- Invalidate rendering of current ad pod if pod cache has been updated
- Parse "special" wrapped URLs inside <asset> tag in SmartXML
- Numerous BrightLine changes to support rendering of choice cards, skip cards, managing ad pod cache when pods are skipped
- Modified backfill URL's slotname parameter to use Roku_Ads_App_ID as it was in v1.6
- Merged Innovid's latest code containing important tracking fixes

Bug Fixes:

- Fix to the BrightLine code to address crashes on some devices still running 7.0 FW
- Fix bug that caused lower ad fill rates for SmartXML responses that included erroneous or empty ad tags in a given ad pod
- Fix construction of generic tracking events for SmartXML when quartile events are not specified

Version 1.6 – 03/2016

Features:

- Interactive ads on Server Stitched Ads
- Support for DFP Waterfall
- Customize Buffer Screens - static image only
- Update LR tags to DFP tags
- Innovid- Extender
- New URL parameter macros: Roku_Ads_Limit_Tracking, Roku_Ads_App_Version, Roku_Ads_Lib_Version, Roku_Ads_Device_Model

Bug Fixes:

- Loading message was not updating correctly for preroll/midroll/postroll ads
- Pass the raw unchanged value in the Roku_Ads_Tracking_ID macro
- Macros were not expanded when held in the URL encoded section of the key/values
- Pre-roll ad in 1080p HD TV didn't display full screen
- BrightScript log is displaying "ERROR: Runtime: FOR EACH value is Invalid" when ad is playing fine

Version 1.5 – 12/2015

Features:

- Max URL transfer count bumped up from 40 to 300
- If Nielsen impressions contained prior values for parameters that should not be substituted due to whitelisting or ad server blacklisting, remove those values from the URL
- set maximum decode resolution on all rendered video ads to avoid memory issue due to buffering algorithm on lower end devices
- Ensure that a properly-handled exit key exits the main video render loop
- Add support for Freewheel "eventCallback"-style impression tracking

Bug Fixes:

- Issue with pressing "back" remote button on image canvas screen
- Fixed VMAP Bug where ad breaks with the same offset were ignored
- Ad Framework unable to parse response - Freewheel Promos
- Fix edge case bug in URL regularization with path parameters
- Fix "PodComplete" tracking sent when interactive ads are exited

Version 1.4 – 10/2015

Features:

- Ability for cross-promotion of channels/content
- Ability to install a channel from a video ad
- Ability to follow content on a channel from a video ad
- Integration of BrightLine Interactive Ads
- Integrate BrightLine Interactive ads to RAF
- Updates to Innovid library (Use "Up" key instead of "*" everywhere)
- SmartXML Parser Changes
- Support "slotImpression" beacon types
- Enhanced Support for quartile tracking events in all scenarios
- Additional attributes such as ad.Title, ad.Creativeld, ad.advertiser for VAST and FreeWheel ads
- String Localization for core UI strings

Bug Fixes:

- SetContentLength API for Nielsen beacons
- Midroll/postroll video playback issues on Roku TVs