

Quantcast Measure SDK for Roku

This implementation guide provides steps for integrating the Quantcast Measure for Roku SDK, so you can take advantage of valuable, actionable insights:

- **Know Your Audience** - Quantcast uses direct measurement and machine learning to build accurate and detailed demographic profiles.
- **Compare and Compete** - Gauge user loyalty by analyzing visit frequency, retention and upgrades over time.
- **Attract Advertising** – Attract advertisers by showcasing your most powerful data points using a trusted source.

If you have any implementation questions, please email mobilesupport@quantcast.com. We're here to help.

Integrating Quantcast Measure for Roku

To integrate Quantcast's SDK into your Roku channel, we recommend using Eclipse, though any text editor will do. The Quantcast SDK supports channels built for Brightscript 2.6 and later.

Download and Set Up the SDK

There are two ways to get the SDK. You can download it directly from the Quantcast website, or you can use GitHub. If you download the file from our site, unzip the file, and add the Quantcast-Roku-Measurement folder into your projects source folder.

SDK Integration

The simplest way to integrate the Quantcast SDK requires only a single startup call. This call synchronously handles the SDK startup process and should finish in less than half a second. If you would like to start the SDK in parallel with other application events please see [SDK Integration Within Main Runloop](#).

```
quantcast = getQuantcastInstance()    quantcast.QuickStartQuantcast("<Insert your API Key Here>")
```

Replace "<Insert your API Key Here>" with your Quantcast API Key. The API Key can be found in the file "api-key.txt" in your Quantcast SDK folder. All your API keys can also be found on your Quantcast dashboard: <https://www.quantcast.com/user/resources?listtype=apps>. For more information about how and when to use the API Key, read [Understanding the API Key](#).

The `userIdentifier` is an optional parameter that accepts a string that uniquely identifies an individual user, such as an account login. Passing this information allows Quantcast to provide reports on your combined audience across all your properties: online, mobile web and mobile app. Please see the [Combined Web/App Audiences](#) section for more information. This parameter may be INVALID or omitted.

The `segmentLabels` parameter is used to create Audience Segments. This parameter may be INVALID or omitted. Learn more in the [Audience Labels](#) section.

(optional) Understanding the API Key

The API key is used as the basic reporting entity for Quantcast Measure. The same API Key can be used across multiple apps (i.e. AppName Free / AppName Paid) and/or app platforms (i.e. iOS / Android). For all

apps under each unique API Key, Quantcast will report the aggregate audience among them all, and also identify/report on the individual app versions.

Compile and Test

You're now ready to test your integration. Build and run your project. Quantcast Measure will record activities and events from your Roku, as long as you quit your app properly (as opposed to killing the debug process while the app is running). After finishing an app session, you will see your session recorded in your [Quantcast Measure dashboard](#) the following day. Questions? Please email us at mobilesupport@quantcast.com.

Congratulations! Now that you've completed basic integration, explore how you can enable powerful features to understand your audience and track usage of your app.

- Read about [User Privacy](#) disclosure and options.
- Learn about event tracking, which can be implemented by calling [LogEvent](#)
- Learn about Audience Segments, which you implement via [Labels](#).
- If you have a web or mobile property, get a combined view of your [mobile app and web audiences](#).

User Privacy

Privacy Notification

Quantcast believes in informing users of how their data is being used. We recommend that you disclose in your privacy policy that you use Quantcast to understand your audiences. You may link to Quantcast's privacy policy: <https://www.quantcast.com/privacy>.

User Opt-Out

You can give users the option to opt out of Quantcast Measure by providing an opt out screen. This should be accomplished with a switch or button in your channel's settings screen with the title "Measurement Options" or "Privacy". When a user taps the button you provide, pass the opt out preference to the SDK by calling:

```
quantcast.SetOptOut(TRUE)
```

Note: when a user opts out of Quantcast Measure, the SDK immediately stops transmitting information to or from the user's device and deletes any cached information that may have retained.

Optional Code Integrations

SDK Integration Within Main Runloop

You might want to manually handle the SDK startup process in parallel with other task your application might be doing. These steps are much more complicated and should only be used by developers with a deep understanding of roMessagePorts and the Roku's lifecycle. This should not be used with the `QuickStartQuantcast` call.

1. In your channel's Main Sub, as soon as possible in the code start up the Quantcast by calling:

```
quantcast = getQuantcastInstance()  
quantcast.StartQuantcast("<Insert your API Key Here>", messagePort, userIdentifier, Se
```

Replace "*<Insert your API Key Here>*" with your Quantcast API Key. The API Key can be found in

the file “api-key.txt” in your Quantcast SDK folder. All your API keys can also be found on your Quantcast dashboard: <https://www.quantcast.com/user/resources?listtype=apps>. For more information about how and when to use the API Key, read [Understanding the API Key](#).

The `messagePort` variable is of type `roMessagePort` and should be the main message queue that is being checked in the channel's event loop.

The `userIdentifier` is an optional parameter that accepts a string that uniquely identifies an individual user, such as an account login. Passing this information allows Quantcast to provide reports on your combined audience across all your properties: online, mobile web and mobile app. Please see the [Combined Web/App Audiences](#) section for more information.

The `SegmentLabels` parameter is used to create Audience Segments. This parameter may be `INVALID` or omitted. Learn more in the [Audience Labels](#) section.

2. In your channel's main event loop, usually found in the main screen object or `Main`, you must pass the SDK any messages that belong to it. This can be accomplished in two ways. The least verbose way is to pass messages to the Quantcast SDK and let the SDK tell you if the message was handled. This loop would look similar to

```
WHILE TRUE
    msg = wait(0, messagePort)
    IF NOT quantcast.HandleMsg(msg)
        <YOUR NORMAL EVENTLOOP CODE GOES HERE>
    END IF
END WHILE
```

The `HandleMsg` call will only check and handle messages belonging to the SDK, all other messages will be untouched. If you would like to check the messages manually before passing them to the SDK you can do that instead with the following block of code

```
WHILE TRUE
    msg = wait(0, messagePort)
    IF type(msg) = "roUrlEvent"
        IF quantcast.IsQuantcastMessage(msg.GetSourceIdentity())
            quantcast.HandleMsg(msg)
        END IF
        <YOUR NORMAL EVENTLOOP CODE GOES HERE>
    END IF
END WHILE
```

Please note that you should not be making a completely new "WHILE TRUE" loop just for Quantcast. The Quantcast calls should be added to the existing runloop.

3. Finally, after the main run loop has ended, you should call the `EndQuantcast` call. It is best to do this as soon as possible as this call tries to make a final data push to the server. It is OK if this is not called as will be in the case where the user exits via the Home button. `quantcast.EndQuantcast()` ##### Audience Labels ##### Use labels to create Audience Segments, or groups of users that share a common property or attribute. For instance, you can create an audience segment of users who purchase something in your channel. For each audience segment you create, Quantcast will track membership of the segment over time, and generate an audience report that includes their demographics. If you have implemented the same audience segments on your website(s), you will see a combined view of your web and channel audiences for each audience segment. Learn more about how to use audience segments, including how to create segment hierarchies using the dot notation, here: <https://www.quantcast.com/help/using-audience-segments>.

Labels can be set on most methods of the SDK and can be either a single string or an array of Strings. Labels are always the last parameter of a function and always optional. For example when you start the Channel you

can pass any known labels in the start function

```
initialLabels = ["sharer.onFB", "purchaser.ebook"]
quantcast.StartQuantcast("<_Insert your API Key Here_>", messagePort, "", initialLabels)
```

Here is an example that adds the label “sharer.firstShare” in addition to the labels you’ve already assigned (“sharer.onFB”, “purchaser.ebook”) via the start function. This example uses the `LogEvent(eventName, label)` method, which you can learn about under [Tracking App Events](#).

```
newLabel = "sharer.firstShare"
theEventStr = @"tweeted"
quantcast.LogEvent(theEventStr, newLabel)
```

All labels that are set during the course of a channel session will register a visit for that channel session. A session is started when a channel is launched. A session is defined as ended when a channel is closed or when a new session starts. In the example above, the session will register a visit on audience segments: “sharer.onFB”, “purchaser.ebook”, and “sharer.firstShare”. If the channel is then closed and re-opened, our servers will record a new channel session.

While there is no specific constraint on the intended use of the label dimension, it is not recommended that you use it to indicate discrete events; in these cases, use the `logEvent:withLabels:` method described under [Tracking Channel Events](#).

Tracking Channel Events

Quantcast Measure can be used to measure audiences that engage in certain activities within your channel. To log the occurrence of a channel event or activity, call the following method:

```
quantcast.LogEvent(theEventStr)
```

`theEventStr` is the string that is associated with the event you are logging. Hierarchical information can be indicated by using a left-to-right notation with a period as a separator. For example, logging one event named “button.left” and another named “button.right” will create three reportable items in Quantcast Measure: “button.left”, “button.right”, and “button”. There is no limit on the cardinality that this hierarchical scheme can create, though low-frequency events may not have an audience report due to the lack of a statistically significant population.

Combined Web/App Audiences

Quantcast Measure enables you to measure your combined web, mobile app, and livingroom audiences, allowing you to understand the differences and similarities of your audiences, or even the combined audiences of your different channels. To enable this feature, you will need to provide a user identifier, which Quantcast will always anonymize with a 1-way hash before it is transmitted from the user's device. This user identifier should also be provided for your website(s) and mobile apps; please see [Quantcast's web measurement documentation](#) for instructions.

Normally, your channels user identifier would be provided via the `StartQuantcast` method as described in the [Required Code Integration](#) section above. If the channel's active user identifier changes later in the app's life cycle, you can update the user identifier using the following method call:

```
quantcast.RecordUserIdentifier(userIdentifierStr)
```

The user identifier is passed in the `userIdentifierStr` argument.

Note that in all cases, the Quantcast Roku SDK will immediately 1-way hash the passed channel user

identifier, and return the hashed value for your reference. You do not need to take any action with the hashed value.

Trouble Shooting

Little or No App Traffic Showing Up In App's Profile On Quantcast.com Quantcast updates its website with your app's latest audience measurement data daily. If even after 1 day no data is showing up in your app's profile on quantcast.com, please check the following: * The `messagePort` passed to `startQuantcast` must be the exact messageport object that you call `wait(0,messageport)` and `HandleMsg`. * Be sure that the runloop calling `HandleMsg` is run at least a few times. Some application can have mutiple runloops in different screens so it is possible the runloop never has a chance to run. Quantcast should be added to the runloop that handles the main screen.

License

This Quantcast Measurement SDK is Copyright 2012-2014 Quantcast Corp. This SDK is licensed under the Quantcast Mobile App Measurement Terms of Service, found at [the Quantcast website here](#) (the "License"). You may not use this SDK unless (1) you sign up for an account at [Quantcast.com](#) and click your agreement to the License and (2) are in compliance with the License. See the License for the specific language governing permissions and limitations under the License. Unauthorized use of this file constitutes copyright infringement and violation of law.