

Cross-Platform Audience Measurement Guide

This implementation guide provides steps for integrating a cross-platform measurement feature enabling you to directly measure *unduplicated reach* across your online, mobile web and mobile app properties. Unduplicated reach reports let you understand audience visitation patterns for unique users as they visit your online website, mobile website or mobile app – powerful intelligence for your feature / content development strategy, and for your prospective advertisers.

Please note that unduplicated reach reports are in beta and are currently under development. You will receive a notification when unduplicated reach reports become available. If you have a mobile app, you'll be able to view reports displaying visit frequency for your logged in users.

Privacy Protection: How Quantcast Handles User Identifiers

To enable unduplicated cross-platform audience measurement, a stable user identifier is needed – one that is consistently used across your online and mobile app properties.

Please note that Quantcast does not use any Personally Identifiable Information (PII). Quantcast's code immediately applies a one-way hash to your user identifier passed before it is transmitted back to Quantcast's servers. As a result, Quantcast will never see the actual value of your user identifier.

Although Quantcast applies a one-way hash to your user identifier, it is strongly recommended that you do not use any PII as the user identifier passed to Quantcast. Examples of PII include the user's actual name, login name, password, phone number, email address, or any other information that could potentially identify your user.

Integrating Unduplicated Cross-Platform Audience Measurement

Measuring Your Online and Mobile Web Audiences

To pass Quantcast a user identifier, you must add a `uid` parameter to the `_qevents` variable of the Quantcast tag. The `uid` parameter should be reported on all page views where the user identifier is known, not just on the user login page.

Normally, the tag that is generated for your properties would look similar to:

```
<!-- Quantcast Tag -->
<script type="text/javascript">
var _qevents = _qevents || [];

(function() {
var elem = document.createElement('script');
elem.src = (document.location.protocol == "https:" ? "https://secure" : "http://edge")
+ ".quantserve.com/quant.js";
elem.async = true;
elem.type = "text/javascript";
var scpt = document.getElementsByTagName('script')[0];
scpt.parentNode.insertBefore(elem, scpt);
})();

_qevents.push({
```

```

qacct:"p-your-pcode-here"
});
</script>

<noscript>
<div style="display:none;">

</div>
</noscript>
<!-- End Quantcast tag -->

```

To add the `uid` parameter, update the `_qevents` variable to:

```

_qevents.push({
qacct:"p-your-pcode-here",
uid:"your-user-identifier"
});

```

Replace the `"your-user-identifier"` string with a string representation of your user identifier. If you provide **audience segment labels** in your tag, the `_qevents` variable would be set up like this:

```

_qevents.push({
qacct:"p-your-pcode-here",
labels:"labelNumberOne,labelNumberTwo",
uid:"your-user-identifier"
});

```

Note: in all cases above, you must replace `p-your-pcode-here` with the p-value generated from your Quantcast account.

Measuring Your Mobile App Audiences

To add your mobile app audience to your cross-platform audience measurement, you first need to integrate the Quantcast Measure for Mobile Apps SDK into each mobile app. Instructions can be found on the [Quantcast website](#).

Once the Quantcast SDK has been integrated in your mobile app, you will need to pass the SDK your user identifier for the user currently using the app.

iOS Apps

To pass the Quantcast iOS SDK the user identifier at the time of the app launch, place the following in your `UIApplication` delegate's `application:didFinishLaunchingWithOptions:` method:

```
[[QuantcastMeasurement sharedInstance] beginMeasurementSessionWithAPIKey:@"your-api-key-here",
userIdentifier:userIdentifierStrOrNil labels:nil];
```

Replace "your-api-key-here" with your Quantcast API Key, which can be generated in your Quantcast account homepage on [Quantcast website](#). The `userIdentifierStrOrNil` parameter is a string (passed as an `NSString` object) that uniquely identifies an individual user, such as an account login. This parameter may be nil if your app does not have a user identifier available at the time the app launches.

If the current user identifier needs to be updated after the app launches, you can update the user identifier using the following method call:

```
[[QuantcastMeasurement sharedInstance] recordUserIdentifier:userIdentifierStrOrNil
withLabels:nil];
```

The current user identifier is passed as a `NSString` in the `userIdentifierStrOrNil` argument. If you need to indicate that no user is currently logged into your app, you should pass `nil` for the `userIdentifierStrOrNil` argument.

Android Apps

In the `onCreate()` method of every Activity in your project, place the following to initialize the measurement service:

```
QuantcastClient.beginSessionWithApiKeyAndWithUserId(this, "your-api-key-here",
userIdentifier);
```

Replace "your-api-key-here" with your Quantcast API Key string, which can be generated in your Quantcast account homepage on the [Quantcast website](#).

The `userIdentifier` parameter is a String representation of your user identifier. This parameter may be null if the user identifier is not known at the time the `onCreate()` method is called.

If your app's active user identifier changes later in the app's life cycle, you can update the user identifier using the following method call:

```
QuantcastClient.recordUserIdentifier(userIdentifier);
```

If you need to indicate that no user is currently logged into your app, pass null for the `userIdentifier` parameter.

Additional Help

If you require any additional help in implementing this feature, please email mobilesupport@quantcast.com with any questions you may have.