

Adobe® Marketing Cloud

# **Chromecast SDK 1.x for Marketing Cloud Solutions**

## Table of Contents

<b>CHROMECAST SDK 1.X FOR MARKETING CLOUD SOLUTIONS .....</b>	<b>3</b>
<i>Adobe Mobile Services .....</i>	<i>3</i>
GETTING STARTED.....	4
ANALYTICS.....	5
<i>Track App States .....</i>	<i>5</i>
<i>Track App Actions.....</i>	<i>5</i>
MARKETING CLOUD.....	6
<i>Marketing Cloud Visitor ID Configuration.....</i>	<i>6</i>
<i>Marketing Cloud Visitor ID Service Methods.....</i>	<i>6</i>
AUDIENCE MANAGER .....	7
<i>Audience Manager Methods.....</i>	<i>7</i>
VIDEO ANALYTICS VIA MEDIA HEARTBEATS .....	8
<i>Integrating media heartbeats APIs .....</i>	<i>8</i>
OPT-OUT AND PRIVACY SETTINGS .....	13
DEBUG LOGGING.....	13
USING BLOODHOUND TO TEST CHROMECAST APPLICATIONS .....	14
CONTACT AND LEGAL INFORMATION.....	15
<i>Help &amp; Technical Support .....</i>	<i>15</i>
<i>Service, Capability &amp; Billing.....</i>	<i>15</i>
<i>Feedback.....</i>	<i>15</i>
<i>Legal.....</i>	<i>15</i>

# Chromecast SDK 1.x for Marketing Cloud Solutions

Chromecast SDK 1.x for Marketing Cloud Solutions lets you measure Chromecast applications written in JavaScript, leverage and collect audience data through audience management, and measure video engagement through Video heartbeats.

## Adobe Mobile Services

Adobe Mobile services provides a new UI that brings together mobile marketing capabilities for mobile applications from across the Adobe Marketing Cloud. Initially, the Mobile service provides seamless integration of app analytics and targeting capabilities from the Adobe Analytics and Adobe Target solutions. Learn more at [Adobe Mobile Services documentation](#).

# Getting Started

Information to help you get starting with the Chromecast SDK for Marketing Cloud Solutions. This section assumes that you have configured a report suite through Adobe Mobile Services to collect app data.

Adobe Mobile services is the primary reporting interface for mobile app analytics and targeting. After you complete the configuration steps you can download a configuration file (ADBMobileConfig) that is pre-configured with your data collection server, report suite, and many other settings.

## Get the SDK

After unzipping the AdobeMobileLibrary-1.\*-Chromecast.zip download file, you'll have the following software components:

- *adbmobil-chromecast.min.js*: Library file to be included in your Chromecast app **index.html**

Add the library file to your **index.html** and create the ADBMobileConfig global variable as follows:

```
<script>
var ADBMobileConfig = {
  "marketingCloud": {
    "org": "06074954519E38710A490D16@AdobeOrg"
  },
  "target": {
    "clientCode": "",
    "timeout": 5
  },
  "audienceManager": {
    "server": "mobileservices.demdex.net"
  },
  "analytics": {
    "rsids": "mobile1figueroa-dev",
    "server": "obumobile1.sc.omtrdc.net",
    "ssl": false,
    "offlineEnabled": true,
    "charset": "UTF-8",
    "lifecycleTimeout": 300,
    "privacyDefault": "optedin",
    "batchLimit": 0,
    "timezone": "MDT",
    "timezoneOffset": -360,
    "referrerTimeout": 0,
    "poi": []
  }
};
</script>
<script type="text/javascript" src="script/lib/adbmobil-chromecast.min.js"></script>
```

# Analytics

Information to help you use the Chromecast SDK with Adobe Analytics.

## Track App States

States are the different screens or views in your application.

Each time a new state is displayed in your application (for example, when a user navigates from the home page to the video details screen), you should send a *trackState* call. In Chromecast, *trackState* is typically called each time a new screen is loaded.

```
ADBMobile.analytics.trackState("State Name", {})
```

The "State Name" is reported in the View State variable in Adobe Mobile services, and a view is recorded for each *trackState* call. In other Analytics interfaces, View State is reported as Page Name and state views is reported as page views.

In addition to the "State Name", you can send additional context data with each track action call:

```
dictionary = {}  
dictionary["myapp.login.LoginStatus"] = "logged in"  
ADBMobile.analytics.trackState("Home Screen", dictionary)
```

Context data values must be mapped to custom variables in Adobe Mobile services.

States are typically viewed using a pathing report so you can see how users navigate your app, and which states are viewed most.

## Track App Actions

Actions are the events that occur in your app that you want to measure.

Each action has one or more corresponding metrics that are incremented each time the event occurs. For example, you might send a *trackAction* call for each new subscription, each time a content is rated, or each time a level is completed. Actions are not tracked automatically, you must call *trackAction* when an event you want to track occurs, and then map the action to a custom event.

```
ADBMobile.analytics.trackAction("myapp.ActionName", {})
```

In addition to the action name, you can send additional context data with each track action call:

```
dictionary = {}  
dictionary["myapp.social.SocialSource"] = "Twitter"  
ADBMobile().trackAction("myapp.SocialShare", dictionary)
```

# Marketing Cloud

Information to help you use the Chromecast SDK with the Adobe Marketing Cloud.

## Marketing Cloud Visitor ID Configuration

The Marketing Cloud visitor ID service provides a universal visitor ID across Marketing Cloud solutions. The visitor ID service is required by Video heartbeat and future Marketing Cloud integrations.

Verify that ADBMobileConfig contains the marketingCloudorg:

```
"marketingCloud" : {
  "org": "YOUR-MCORG-ID"
}
```

*Note: Marketing Cloud organization IDs uniquely identify each client company in the Adobe Marketing Cloud, and are similar to the following value: 016D5C175213CCA80A490D05@AdobeOrg. Be sure to include @AdobeOrg.*

After configuration, a Marketing Cloud visitor ID is generated and is included on all hits. Other visitor IDs (custom and automatically-generated) continue to be sent with each hit.

## Marketing Cloud Visitor ID Service Methods

Marketing Cloud visitor ID methods are prefixed with "visitor."

Method	Description
getMarketingCloudID	Retrieves the Marketing Cloud visitor ID from the visitor ID service. <i>ADBMobile.visitor.getMarketingCloudID()</i>
syncIdentifiers	Along with the Marketing Cloud visitor ID, you can set additional customer IDs to associate with each visitor. The Visitor API accepts multiple Customer IDs for the same visitor, along with a customer type identifier to separate the scope of the different customer IDs. This method corresponds to setCustomerIDs in the JavaScript library. Ex: <i>identifiers = {}</i> <i>identifiers["idType"] = "idValue"</i> <i>ADBMobile.visitor.syncIdentifiers(identifiers)</i>

# Audience Manager

Information to help you send signals and retrieve visitor segments from audience manager.

## Audience Manager Methods

Send signals and retrieve visitor segments from audience management.

Method	Description
getVisitorProfile	Returns the visitor profile that was most recently obtained. Returns an empty object if no signal has been submitted yet. <i>ADBMobile.audienceManager.getVisitorProfile()</i>
getDpid	Returns the current DPID. <i>ADBMobile.audienceManager.getDpid()</i>
getDpuuid	Returns the current DPUUID. <i>ADBMobile.audienceManager.getDpuuid()</i>
setDpidAndDpuuid	Sets the DPID and DPUUID. If DPID and DPUUID are set, they will be sent with each signal. <i>ADBMobile.audienceManager.setDpidAndDpuuid("myDpid", "myDpuuid")</i>
submitSignal	Sends audience management a signal with traits <i>ADBMobile.audienceManager.submitSignal()</i>

# Video Analytics via Media heartbeats

Information to instrument a media (video/audio) player with the media module of Adobe Mobile SDK.

The APIs belonging to media heartbeats enable real-time dashboards and other media reporting capabilities. This guide is intended for a media integration engineer who has an understanding of the APIs and workflow of the media player being instrumented. Implementing these APIs requires that your media player provide the following:

- An API to subscribe to player events. The media heartbeat requires that you call a set of simple APIs when events occur in your player.
- An API or class that provides player information, such as media name and play-head position.

## Integrating media heartbeats APIs

Integrating media analytics real-time media tracking into a media player requires including Adobe Mobile SDK, instantiating and configuring the media heartbeats instance, listening to media player events and using appropriate media heartbeats APIs in your project. Therefore the steps for a developer integrating the media heartbeats APIs are as follows:

### Initial Setup

1. Acquire the required Adobe Mobile SDK and add it into your project: The SDK contains the media heartbeats module.
2. Provide the configuration for heartbeats through ADBMobileConfig. The new media module, self configures itself through ADBMobileConfig.

## Approaches to tracking Media during a Playback Session

*Using Tracking APIs:* This way of integration involves handling player events and calling relevant media heartbeats APIs from the application. The APIs documentation and this guide should be used to learn the details on all the media heartbeats APIs and their respective life cycle.

## Configure Media heartbeats

Once the application has acquired the Adobe Mobile SDK, the first step is to configure the media heartbeats.

The global variable used to configure Adobe Mobile has an exclusive key for media heartbeats with the name "mediaHeartbeat". This is where the configuration parameters for the media heartbeats belong. An example below demonstrates it. Also, a sample ADBMobileConfig implementation is provided with the package (under samples), settings for which must be obtained from an Adobe representative.

```
var ADBMobileConfig = {
  "marketingCloud": {
    "org": "06074954519E38710A490D16@AdobeOrg"
  },
  "target": {
    "clientCode": "",
    "timeout": 5
  },
  "audienceManager": {
```



```

"server": "mobileservices.demdex.net"
},
"analytics": {
  "rsids": "mobile1figueroa-dev",
  "server": "obumobile1.sc.omtrdc.net",
  "ssl": false,
  "offlineEnabled": true,
  "charset": "UTF-8",
  "lifecycleTimeout": 300,
  "privacyDefault": "optedin",
  "batchLimit": 0,
  "timezone": "MDT",
  "timezoneOffset": -360,
  "referrerTimeout": 0,
  "poi": []
},
"mediaHeartbeat": {
  "server": "heartbeats.omtrdc.net",
  "publisher": "06074954519E38710A490D16@AdobeOrg",
  "channel": "test-channel-chromecast",
  "ssl": false,
  "ovp": "chromecast-player",
  "sdkVersion": "chromecast-sdk",
  "playerName": "Chromecast"
}
};

```

Config Parameter	Description
server	String representing the URL of the tracking endpoint on the backend side.
publisher	String representing the content publisher unique identifier.
channel	String representing the name of the content distribution channel.
ssl	Boolean representing whether ssl should be used for tracking calls
ovp	String representing the name of the video player provider
sdkVersion	String representing the current version of the app/sdk.
playerName	String representing the name of the player.

If *mediaHeartbeat* is incorrectly configured, media module (VHL) will go into disabled state and will not send any tracking calls.

## Media heartbeat methods

Track media using the following methods

Method	Description
trackLoad	Media playback tracking method to track Media Load, and set the current session active Ex: <i>' Create a media info object</i> <i>mediaInfo = new ADBMobile.media.MediaInfo()</i> <i>mediaInfo.id = "sample-media-id"</i> <i>mediaInfo.name = "VideoName"</i> <i>mediaInfo.playerName = "Chromecast sample player"</i> <i>mediaInfo.length = "600"</i> <i>mediaInfo.name = ADBMobile.media.STREAM_TYPE_VOD</i>

	<p><i>'Create context data if any</i>  <i>mediaContextData = {}</i>  <i>mediaContextData["cmk1"] = "cmv1"</i>  <i>mediaContextData["cmk2"] = "cmv2"</i></p> <p><i>ADBMobile.media.trackLoad(mediaInfo, mediaContextData)</i></p>
trackUnload	<p>Media playback tracking method to track Media Unload and deactivate current session</p> <p>Ex:  <i>ADBMobile.media.trackUnload()</i></p>
trackPlay	<p>Media playback tracking method to track Media Play</p> <p>Ex:  <i>ADBMobile.media.trackPlay()</i></p>
trackPause	<p>Media playback tracking method to track Media Pause</p> <p>Ex:  <i>ADBMobile.media.trackPause()</i></p>
trackComplete	<p>Media playback tracking method to track Media Complete</p> <p>Ex:  <i>ADBMobile.media.trackComplete()</i></p>
trackError	<p>Error tracking method to track Player Error</p> <p>Ex:  <i>ADBMobile.media.trackError(msg.GetMessage(),</i>  <i>ADBMobile.media.SOURCE_PLAYER)</i></p>
trackEvent	<p>Media tracking method to track events that do not belong to media lifecycle and are optional. Example: AD_START/AD_COMPLETE, CHAPTER_START/CHAPTER_COMPLETE; Refer Events section for detailed list of events. This method takes three arguments: <i>event constant</i>, <i>event info</i>, and <i>context data</i> (send empty object if there is no context data)</p>
setDelegate	<p>Method to set delegate on media heartbeat to receive playhead position and QoSInfo. Media Heartbeat will invoke "getCurrentPlaybackTime" and "getQoSInfo" methods on this delegate to get playhead and QoS values respectively.</p> <p>Ex:  <pre>ADBMobile.media.setDelegate({   "getCurrentPlaybackTime": function() {     return playhead;   },   "getQoSInfo": function(){     var qosInfo = new ADBMobile.media.QoSInfo();     qosInfo.bitrate = player.getBitrate();     qosInfo.droppedFrames = player.getDroppedFrames();     return qosInfo;   } });</pre></p>

In addition to the above methods, the media module also provides constants to track media events

Constant	Description
BUFFER_START	EventType for Buffer Start
BUFFER_COMPLETE	EventType for Buffer Complete

SEEK_START	EventType for Seek Start
SEEK_COMPLETE	EventType for Seek Complete
BITRATE_CHANGE	EventType for Bitrate change
CHAPTER_START	EventType for Chapter Start
CHAPTER_COMPLETE	EventType for Chapter Complete
CHAPTER_SKIP	EventType for Chapter skip
AD_BREAK_START	EventType for AdBreak Start
AD_BREAK_COMPLETE	EventType for AdBreak Complete
AD_BREAK_SKIP	EventType for AdBreak Skip
AD_START	EventType for Ad Start
AD_COMPLETE	EventType for Ad Complete
AD_SKIP	EventType for Ad Skip
STREAM_TYPE_LIVE	Constant for Stream Type LIVE
STREAM_TYPE_VOD	Constant for Stream Type VOD
SOURCE_APPLICATION	Constant for Error source being Application
SOURCE_PLAYER	Constant for Error source being Player

There is also convenience methods as described below for creating various info objects sent through the media heartbeat API methods. Please refer to the table below

Method	Description
MediaInfo	This method returns an initialized Media Information object Ex: <i>mediaInfo = new ADBMobile.media.MediaInfo()</i>
AdInfo	This method returns initialized Ad Information object Ex: <i>adInfo = new ADBMobile.media.AdInfo()</i>
ChapterInfo	This method returns initialized Chapter Information object Ex: <i>chapterInfo = new ADBMobile.media.ChapterInfo()</i>
AdBreakInfo	This method returns initialized AdBreak Information object Ex: <i>adBreakInfo = new ADBMobile.media.AdBreakInfo()</i>
QoSInfo	This method returns initialized QoS Information object Ex:

<code>QoSInfo = new ADBMobile.media.QoSInfo()</code>
--

Once the application developer is familiar with all the above APIs, integrating media heartbeats with the application media player can be achieved by calling raw tracking APIs directly.

### **Using Media Tracking APIs**

App developers can use the media tracking APIs described above directly to track media life cycle and ad/chapter events. Please refer to the sample app (to be shipped soon with later versions) for example implementation on implementing these APIs.

## Opt-Out and Privacy Settings

You can control whether or not Analytics data is sent on a specific device using the following settings:

- *privacyDefault* setting in *ADBMobileConfig*. This controls the initial setting that persists until it is changed in code.
- *ADBMobile.config.setPrivacyStatus()* method. After the privacy setting is changed using this method, the change is permanent until it is changed again using this method, or the app is completely uninstalled and re-installed.

*Note: Media heartbeat tracking calls are also disabled if the privacy status is set to opt-out*

**Set this to *ADBMobile.config.PRIVACY\_STATUS\_OPT\_IN*/*ADBMobile.config.PRIVACY\_STATUS\_OPT\_OUT* if user wants to opt-in/opt-out**

*ADBMobile.config.setPrivacyStatus(ADBMobile.config.PRIVACY\_STATUS\_OPT\_OUT)*

**This method will return the current value for privacy status constant (*ADBMobile.config.PRIVACY\_STATUS\_OPT\_IN* or *ADBMobile.config.PRIVACY\_STATUS\_OPT\_OUT*)**

*ADBMobile.config.getPrivacyStatus()*

## Debug Logging

*ADBMobile* library provides debug logging through the *setDebugLogging* method. Debug logging should be set to false for all the production apps.

*ADBMobile.config.setDebugLogging(true)*

## Using Bloodhound to Test Chromecast Applications

During application development, Bloodhound lets you view server calls locally, and optionally forward the data to Adobe collection servers.

Bloodhound can be downloaded from any app configuration page in Adobe Mobile services.

[Bloodhound 3 Beta for Mac documentation](#)

[Bloodhound 2 for Windows documentation](#)

## Contact and Legal Information

Information to help you contact Adobe and to understand the legal issues concerning your use of this product and documentation.

### Help & Technical Support

The Adobe Marketing Cloud Customer Care team is here to assist you and provides a number of mechanisms by which they can be engaged:

- [Check the Marketing Cloud help pages for advice, tips, and FAQs](#)
- [Ask us a quick question on Twitter @AdobeMktgCare](#)
- [Log an incident in our customer portal](#)
- [Contact the Customer Care team directly](#)
- [Check availability and status of Marketing Cloud Solutions](#)

### Service, Capability & Billing

Dependent on your solution configuration, some options described in this documentation might not be available to you. As each account is unique, please refer to your contract for pricing, due dates, terms, and conditions. If you would like to add to or otherwise change your service level, or if you have questions regarding your current service, please contact your Account Manager.

### Feedback

We welcome any suggestions or feedback regarding this solution. Enhancement ideas and suggestions for the Analytics suite [can be added to our Customer Idea Exchange](#).

### Legal

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Published by Adobe Systems Incorporated.

[Terms of Use](#) | [Privacy Center](#)

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. A trademark symbol (®, ™, etc.) denotes an Adobe trademark.

All third-party trademarks are the property of their respective owners. Updated Information/Additional Third Party Code. Information available at <http://www.adobe.com/go/thirdparty>.