



UNIFIED DIGITAL MEASUREMENT

Application Tag for iOS

Implementation Guide

Document version: 3.1.0 (released March 12, 2015)

FOR FURTHER INFORMATION, PLEASE CONTACT:

comScore, Inc.

Tag Support

+1 866 276 6972

tagsupport@comscore.com

Contents

Implementation Quick Start	3
Introduction	4
Tagging	4
Application Tag	4
Intended Use	4
Checklist	5
Implementing the Application Tag	6
Include the Library	6
Import the Library in Your Project	7
Set the Application Context	7
Set Configuration Parameters	8
The <code>CustomerC2</code> Parameter	8
The <code>PublisherSecret</code> Parameter	8
The <code>AppName</code> Parameter	8
User Experience Measurement	9
Auto-Update of Usage Time	9
Updating an Existing Implementation	11
Testing the Implementation	12
Configuring Your Test Environment	12
Reviewing Collected Data	12
Simple Test Scenario	13
Frequently Asked Questions	14

Implementation Quick Start

This section contains abridged implementation instructions for advanced users.

1. If you are updating an existing implementation, then please refer to [Updating an Existing Implementation on page 11](#) to see if there are any relevant steps mentioned.
2. Include the *comScore* folder – which contains the file *libcomScore.a* and folder *headers* – in your project as a *group* and include *libcomScore.a* in the *Link Binary with Libraries* list in your project settings (see [Include the Library on page 6](#)).
3. Make sure your project includes *SystemConfiguration.framework* and *Security.framework* (see [Checklist on page 5](#)).
4. Import the `CSCComScore.h` headers in your code (see [Import the Library in Your Project on page 7](#)).
5. Set the application context by including the following call inside `application didFinishLaunchingWithOptions` (see [Set the Application Context on page 7](#)):
`[CSCComScore setAppContext];`
6. Set library configuration parameters by calling the following API methods immediately after setting the application context (see [Set Configuration Parameters on page 8](#)).
 - Configure your assigned comScore Client ID:
`[CSCComScore setCustomerC2:@"your_client_ID"];`
 - Configure your assigned comScore Publisher Secret:
`[CSCComScore setPublisherSecret:@"your_publishersecret_string"];`
7. Implement calls to `[CSCComScore onUxActive]` and `[CSCComScore onUxInactive]` for measurement of application usage time measurement (see [User Experience Measurement on page 9](#)).
8. Enable the auto-update feature to improve measurement of usage times by calling the `enableAutoUpdate` method (see [Auto-Update of Usage Time on page 9](#)).

After successfully following these steps you will have implemented the comScore Application Tag. Your application should now send measurements whenever it is active. Please refer to [Testing the Implementation on page 12](#) for instructions on testing the implementation.

Please inform your comScore account representative of the collected application name (see [The AppName Parameter on page 8](#)) to ensure proper classification for Mobile Audience reporting.

Introduction

Tagging

A *tag* is a piece of scripting or markup that is placed on a website or another web based content asset. Tags are sometimes referred to as tracking pixels or beacons. They are used to measure the consumption of digital content by an end-user. *Tagging* is the process of adding a comScore tag – also called *measurement code* or *SDK*⁽¹⁾ – to your digital content. Each time a tagged piece of content – e.g., a web page, a video stream or an application view in a (mobile) application – is used by an end-user, the tag sends data via an HTTP request to comScore's data collection servers.

Application Tag

The comScore Application Tag provides (Mobile) Audience Measurement. It is designed to accurately capture and report on usage measurements for applications across all popular platforms. comScore will report reach and launches by measuring the application start event. The data collected through this method will be reported in comScore Direct and syndicated ranking reports.

Intended Use

These instructions and the SDK library are intended to be used with iOS applications that are developed in native Objective C code and compiled with Apple Xcode.

If your application is developed in another programming language then please inform your comScore account representative or contact comScore Tag Support (contact details can be found on the front page of this document). Your client representative and comScore Tag Support can confirm whether or not the comScore Application Tag SDK can be used with your development environment.

⁽¹⁾ SDK is an acronym for Software Development Kit which is typically a set of tools and documentation for the creation of software.

Checklist

Before adding the comScore Application Tag SDK to your project, please complete the following checklist:

- Familiarize yourself with the contents of this document.
If you are updating an existing implementation, then please refer to [✎ Updating an Existing Implementation on page 11](#) to see if there are any relevant steps mentioned.
- Make sure you have your comScore Client ID.
 - The Client ID is also known as the *C2 value*. This is a number of at least 7 digits, provided by comScore.
 - You can find your Client ID in the comScore Direct interface (see inset below).
- Make sure you have a *Publisher Secret*, which is supplied by comScore.
 - The *Publisher Secret* is a text string used to obfuscate the ADID⁽²⁾ when measurements are sent to comScore servers. The ADID is an anonymous unique value that is generated by the Application Tag library and is persistent only between applications of the same publisher.
 - It is the same for all of your applications, but unique to you.
 - This is required for security, and to protect the privacy of your application's users.
 - You can find the Publisher Secret in the comScore Direct interface (see inset below).
- Make sure your project includes *SystemConfiguration.framework* and *Security.framework*.
 - If you are targeting versions of iOS older than iOS 6.0, mark *AdSupport.framework* as "Optional" in your project's target options (*Build Phases > Link Binary With Libraries*).
- Make sure your project includes the *comScore* folder which contains the file *libcomScore.a* and *headers* folder..
 - This folder is included in the comScore Application Tag SDK.
 - You can find the comScore Application Tag SDK in the comScore Direct interface (see inset below).



Your comScore Client ID and Publisher Secret can be found in the comScore Direct interface.

1. Use a web browser to go to [✎ http://direct.comscore.com/clients/MobileApp.aspx](http://direct.comscore.com/clients/MobileApp.aspx).
2. Log in to comScore Direct with your client login ID and password if you are prompted to.
If you do not have a comScore login and already are a comScore client then please contact your client service representative.
If you are not yet a comScore client please sign up for a client login ID and password through comScore Direct.
3. Confirm you are on the "Mobile App" tab and click on "Get Tag".
4. The Client ID – or, C2 value – and Publisher Secret are shown in the popup.

Please contact your client service representative or comScore Tag Support with any questions (contact details can be found on the front page of this document).

⁽²⁾ ADID is an acronym for *Anonymous Device Identifier* which is generated by the comScore Application Tag library.

Implementing the Application Tag



Please implement the comScore Application Tag library according to the instructions given in this document.

Please do not implement the Application Tag library on a separate thread or with a delay. Please call the library's API methods and assign any event handlers as instructed.

The library needs to be able to register application lifecycle events before those events are triggered for correct measurement of application start, foreground to background transitions and application usage times. The provided implementation instructions facilitate this.

If you have doubts or concerns regarding these instructions then please contact your comScore account representative or contact comScore Tag Support to discuss your doubts and/or concerns.

If you are updating an existing implementation from comScore Application Tag for iOS version **2.1305.08 or lower** then you likely have to address the items mentioned in [✎ Updating an Existing Implementation on page 11](#).

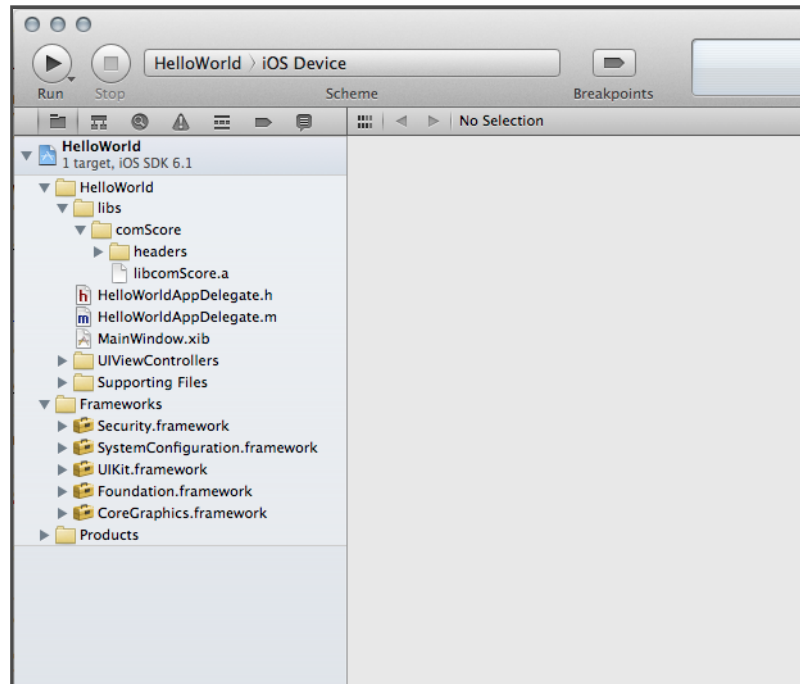
Include the Library

To begin, the *comScore* library must be available in your project.

1. Drag and drop the *comScore* folder – which contains the file *libcomScore.a* and folder *headers* – into your project.
2. Add the library for linking:
 1. Go to the *Build Phases* panel of your project's target options.
 2. Expand *Link Binary with Libraries* to reveal the list of frameworks and libraries that will be linked when building a binary.
 3. Drag and drop *libcomScore.a* from the *Project Navigator* into the list of frameworks and libraries beneath *Link Binary with Libraries*. This will also automatically cause the *comScore* folder to be added to the *Library Search Paths* in the *Build Settings* panel of your project's target options.
3. Add the native *SystemConfiguration.framework* and *Security.framework*.

Previous versions of this document included instruction to add the native *AdSupport.framework* in your project. If your application does not contain advertisements and your project does not need the *AdSupport.framework* for other reasons, then you can safely remove the *AdSupport.framework*. The comScore library does not require the *AdSupport.framework*.

The image below shows an example of what the Xcode *Project Navigator* bar could look like after performing these steps.



Import the Library in Your Project

Include the comScore library headers with a statement like:

```
#import "CSCoScore.h"
```

If you are upgrading from a previous Application Tag SDK implementation then you might have to change the import statement. Please verify this in your project code's import statements. Previous versions required the framework to be imported, instead of only the headers.

Set the Application Context

You must inform the comScore library of the *Application Context* by using the `setAppContext` method of the Application Tag library API. This has to be done **before** calling any of the other comScore library API methods, including those that define configuration parameter settings for the Application Tag library.

Please add the following statement inside `application didFinishLaunchingWithOptions`:

```
[CSCoScore setAppContext];
```

Please include the call to the `setAppContext` as well as any statements with Application Tag library configuration parameter settings inside `application didFinishLaunchingWithOptions`. The configuration parameters are discussed in [Set Configuration Parameters on this page](#). Please also do not implement these initialization calls on a separate thread or with a delay.

The Application Tag library relies on application lifecycle events. If you do not follow these instructions then the Application Tag library will not be able to register application lifecycle events before those events are triggered for correct measurement of application starts and usage times.

Set Configuration Parameters

Please use the API methods described in this section to set the configuration parameters of the Application Tag library. Please make sure to call these API methods immediately after setting the application context.

Previous versions of the library also supported the use of a settings file named `comScore.plist` for setting configuration parameters. Support for this functionality has been deprecated in favor of the library API methods.

The CustomerC2 Parameter

The *Customer C2* is your comScore Client ID. It is a number with at least 7 digits. You can set the `CustomerC2` parameter as follows⁽³⁾:

```
[CSCComScore setCustomerC2:@"1000001"];
```

The PublisherSecret Parameter

The *Publisher Secret* is a comScore-supplied string used to obfuscate the ADID⁽⁴⁾ when measurements are sent to comScore servers. It is the same for all of your applications, but unique to you, and is required to protect the privacy of your application's users. You can set the `PublisherSecret` parameter as follows:

```
[CSCComScore setPublisherSecret:@"your_publishersecret_string"];
```

The AppName Parameter

By default the library sets the reported application name from your app's *Info.plist* application bundle name (`CFBundleName`).

The collected application name is used in the classification for Mobile Audience reporting. Should you want to override the automatically collected value, then you can set the *App Name* manually with your preferred application name by using the following code statement.

⁽³⁾ The examples in this document use Client ID value `1000001` which is an example value. Please make sure you use your assigned comScore Client ID instead.

⁽⁴⁾ ADID is an acronym for *Anonymous Device Identifier* which is generated by the Application Tag library.


```
[CSCoScore setAppName:@"My Application Name"];
```

Please inform your comScore account representative of the collected application name to ensure proper classification for Mobile Audience reporting.

User Experience Measurement

If your application provides a user experience – like the playback of music or video – then please implement calls to the following Application Tag library API methods:

onUxActive

This API method needs to be called whenever your application **starts** providing the user experience (i.e., audio and/or video).

onUxInactive

This API method needs to be called whenever your application **stops** providing the user experience.

For the correct measurement of usage time during background activity this is particularly important. For example, to notify the Application Tag library that playback of audiovisual content has started:

```
[CSCoScore onUxActive];
```

Auto-Update of Usage Time

The Application Tag library contains an auto-update feature that will update the application usage times on regular intervals. This feature is turned off by default.

To get the best possible measurement of application usage time please enable the auto-update feature and implement calls to the `onUxActive` and `onUxInactive` methods.

You can enable the auto-update feature by including a call to the `enableAutoUpdate` method as part of the initialization settings. The method accepts two parameters:

interval

An integer interval value in seconds. This parameter has a default value of `60`, which is also the minimum value. If you provide a value lower than `60` for the interval then the library will use the default value.

foregroundOnly

A boolean flag which will allow auto-update to be active while the application is in the background. The default value of the parameter is `YES` which means the auto-update feature is *only* active while the application is in the foreground.

If your application is capable of delivering a user experience while the application is in the background then please make sure you allow auto-update to be active while the application is in the background by setting the `foregroundOnly` flag to `NO`.

Please note the following when you enable the auto-update feature and allow it to be active while the application is in the background:

The auto-update feature will only activate in the background if the Application Tag library determines your application is *active* too. This relies on calls to the `onUxActive` method (see [User Experience Measurement on page 9](#)). Similarly, when you call `onUxInactive` the auto-update feature will automatically deactivate while the application is in the background.

Some examples of activating auto-update:

```
21. [CSComScore enableAutoUpdate:120 foregroundOnly:NO]; // Interval at 2 minutes during fore- and background.  
22. [CSComScore enableAutoUpdate:300 foregroundOnly:YES]; // Interval at 5 minutes, only during foreground.
```

If you have followed the implementation instructions in this chapter successfully then the Application Tag library will be collecting data for your application. The Application Tag library will send a measurement whenever your application is started.

Updating an Existing Implementation

Remove the Streaming Tag Library

This only applies if your application project includes the comScore Streaming Tag.

As of version 3.1412.05 of the Application Tag library the Streaming Tag classes are delivered as part of the Application Tag library.

The Streaming Tag classes are no longer delivered in their own *StreamSense* folder (which contains the *libStreamSense.a* library and *headers* folder). Please remove this folder and its contents from your project when updating.

Change Import Statements

The library is no longer distributed as a framework. As a result the import statements have slightly changed. Please check the import statements in your project against the instructions mentioned in [✎ Import the Library in Your Project on page 7](#). Please also remove any remaining *comScore* frameworks from your project.

Remove Settings File

The use of the *comScore.plist* settings file has been deprecated in favor of using API methods to set the library configuration. With the next major version update of the library any support for the settings file will be removed. If your application project uses this settings file then please update your implementation to use the library API methods instead.

Remove Calls to Start Method

From version 2.1306.20 the comScore Application Tag library will automatically detect the application start. If you are upgrading an existing Application Tag implementation then your application could contain a statement containing a call to the *Start* API method. For example:

```
| [CSCoMScore Start];
```

You will likely find that statement in the *application didFinishLaunchingWithOptions* method. Please **remove** that statement from your application.

Testing the Implementation

As you test your comScore-tagged application internally, comScore servers will collect the measurements. For immediate testing, you can use any tool you would normally use to look at HTTP traffic coming from your application. As long as your chosen tool can show you the query string parameters it will be sufficient for testing your Application Tag library implementation. Some often used tools are HTTP proxies like *Charles Proxy* and *Fiddler*.

Configuring Your Test Environment

These instructions are generic and also apply to other tools commonly used to review HTTP traffic from (mobile) devices.

1. Setup your tool of choice to be able to review HTTP traffic from your device.
2. Setup your device so its HTTP traffic can be reviewed by your chosen tool.
3. To make it easier to review the collected data, configure your chosen tool to filter HTTP traffic:
 - Configure the filter to use hostname b.scorecardresearch.com.

If you see an HTTP request going to this hostname when you *cold start*⁽⁵⁾ the application this means the comScore Application Tag SDK is sending measurement data.

Reviewing Collected Data

The collected data will be present on the URL of the HTTP requests going to comScore's servers as query string parameters. There are a number of key query string parameters you can check to confirm the application is tagged correctly. These parameters are:

Parameter	Description / Comments
c1	Fixed value 19 , indicating the collected data is sent from an application
c2	The comScore client ID
ns_ap_an	The application name as it will show up in reporting ⁽⁶⁾
ns_ap_sv	The comScore Application Tag library version
ns_ap_cs	Number of cold starts of the application
ns_ap_dft	Foreground duration – in milliseconds – of previous sessions ⁽⁷⁾
ns_ap_dbt	Background active duration – in milliseconds – of previous sessions ⁽⁸⁾

⁽⁵⁾ A *cold* start is defined as starting the application while the application was not running in the foreground or background.

⁽⁶⁾ Please reach out to your comScore account team to ensure this value is properly classified for Mobile Audience reporting.

⁽⁷⁾ The first time the application is cold started after installing it on a device the value of [ns_ap_dft](#) will be close to [0](#). The second time the application is cold started the value of [ns_ap_dft](#) will be the time the application was running in the foreground in the previous session.

⁽⁸⁾ This is only relevant to applications that can be active when put in the background or minimized, like (streaming) music players or map/navigation applications. For [ns_ap_dbt](#) to have a non-zero value the calls for [User Experience Measurement on page 9](#) must be implemented.

Simple Test Scenario

Please execute the following simple test scenario to confirm application cold starts and foreground duration are collected properly:

1. Cold start the application.
 - An HTTP request to comScore's servers should appear. Please check the value of parameter `ns_ap_dft` which should have a low value (usually well below 300).
 - Keep the application in the foreground for 15 seconds.
2. Return to the device / operating system's home screen, i.e., on an iOS device press the home button.
 - Wait for 30 seconds.
3. Bring the application to the foreground.
 - Keep the application in the foreground for 45 seconds.
4. Return to the device / operating system's home screen.
 - Wait for 20 seconds.
5. Terminate the application while it is in the background.
6. Wait 50 seconds.
7. Cold start the application again.
 - An HTTP request to comScore's servers should appear. Please check the value of parameter `ns_ap_dft` which should be close to 60000 if you've followed the timings in this scenario.

Once you have verified the SDK is correctly implemented you must resubmit your application to the iTunes App Store for approval.

Frequently Asked Questions

How can I validate my application measurement is working as intended?

There are two ways to validate your application measurements. You could test the measurements yourself by following the [instructions on page 12](#).

In addition, you could start your application at least 10 times, and contact comScore Tag Support (contact details can be found on the front page of this document) with your client ID and PublisherSecret. We will confirm whether or not we have received your application measurements within two business days.

Do I have to re-submit my app?

Yes. Once you have tested and confirmed your implementation of the comScore Application Tag library, resubmit your application. If the changes you have made since your last application update are only related to the comScore Application Tag library then your applications should pass review in a timely manner.

Will adding the comScore library slow down the application?

No. The code in the comScore Application Tag library is extremely lightweight and will not affect the performance of your application.

How can I change my application's name that is reported to comScore?

The reported application name defaults to the value mentioned in section [The AppName Parameter](#). Before you call any of the comScore Application Tag library's notification methods, you can change your reported application name by setting the `AppName` configuration parameter (see [The AppName Parameter on page 8](#)).

Please reach out to your comScore account team to ensure this value is properly classified for Mobile Audience reporting.

Will comScore receive measurements if my application user is not connected to the internet?

Yes. If device has no internet connectivity, all the measurements are being collected. They will be received by comScore when internet connectivity is enabled again.