

# Plugin Cordova

Plugin Cordova Plugin Cordova 1.0.0 Beta Plugin Cordova 1.1.x

- Getting Started
  - Changelog
  - Download
- Integration
  - Basic Integration
  - Example
  - Manage GDPR compliancy
  - Advanced Integration
- Push Notifications
  - Enable Push Notifications
  - Prevent notifications tracking
- Tracking
  - Events
  - Using Analytics
  - Update Device Information
  - Manually Update Geolocation (Android only)
  - Enable/Disable Location Services (iOS)
  - Views
  - States
  - Subscription Tag
  - Static Lists
  - Tracking Facebook
- Deep Linking
  - Retrieve custom parameters from Push/In-App/Geofence/Beacons
  - Retrieve Push Notification payload
- In-App Notifications
  - Prevent In-App notification display
- Inbox
- Troubleshooting
  - Find your A4SID (or Device ID or Mobinaute ID)
  - Restrict SDK connection
  - Specific for Android
  - Specific for iOS

## Getting Started

Accengage SDK allows you to track the execution and display of In-App and to handle Push notifications of your application.

This Phonegap/Cordova documentation explains the various methods available and how to use them.

In order to understand more deeply the features of the SDK, please refer also to the [User Guide](#), as well as native [iOS](#) and [Android](#) document ations.

## Changelog

The full changelog is available here : [Changelog Cordova](#)

## Download

Current Plugin Version: 1.4.2 (November 2018)

- Android SDK Version: 3.8.3  
*including Google Play Services 15.0.1 (Base, Location, Ads), Firebase 17.0.0 (Core, Messaging), A4SSDK-Beacons (1.3.0) A4SSDK-GooglePlayServices (3.3.0) and Accengage plugin-firebase-messaging (2.5.0)*
- iOS SDK Version: 6.3.2
- Cordova CLI version required : 8.0.0
- Cordova Android platform version : 7.1.4
- Cordova iOS platform version : 4.5.4

Plugin installation:

```
cordova plugin add com.bma4s.sdk.plugins.cordova
```

Link : <https://www.npmjs.com/package/com.bma4s.sdk.plugins.cordova>

The Cordova plugin is now compatible with FCM.

- If you're already using FCM, you may see our documentation about [conflicts dependencies](#)
- If you're still using GCM, 2 ways are available :
  - [Migrate your GCM project to a FCM project](#) following our documentation
  - Or integrate our GCM plugin manually following this [documentation](#)

## Integration

### Basic Integration

Update the file **config/BMA4SCordovaConfig.json** with your PartnerID and you privateKey BEFORE preparing/building your app on Cordova.

### **BMA4SCordovaConfig** example:

Please **DO NOT** copy paste the content of this example (comments are not allowed inside a valid JSON file) Fields with \* means REQUIRED

```

{
  //General configuration
  "facebookAppId":"your-facebook-app-id", //Facebook application id

  //iOS Configuration
  //Base
  "ios_partnerId":"{ios_partnerId}", // * Accengage partnerId of your iOS App
  "ios_privateKey":"{ios_privateKey}", // * Accengage privateKey of your iOS App
  "ios_allowPushNotification":true, // Use iOS push notifications feature
  "ios_geoLocEnabled":true, // Use Geolocation
  "ios_beaconServicesEnabled":true, // Use Beacons feature
  "ios_geofenceServicesEnabled":true, // Use Geofences feature
  "ios_optin_data":true, // Use GDPR features
  "ios_shouldWaitForCustomParametersListener":true // Custom parameters

  //Android Configuration
  //Base
  "android_partnerId":"your-partner-id", // * Accengage partnerId of your Android
App
  "android_privateKey":"your-private-key", // * Accengage privateKey of your
Android App

  //Push (FCM)
  "android_senderId":"your-sender-id", // FCM sender id. Example : 123456
  "android_notifications_icon":"your-icon-name", // Icon (embedded in your app
into res/drawable- folders) to use for push notification into android notification
center
  "android_notifications_accent_color":"#ccc", // Accent color to use for
notification icon into android notification center (Android 5.0+)

  //GDPR
  "android_optin_data":true

  //Others
  "android_use_advertiser_id":true, // Use Google Advertiser Id
  "android_no_geoloc":false, // If true, deactivate Geolocation feature
  "android_activate_logging" : true // Display logs, please change it to false
before production
}

```

Avoid to simply name your notification icon, "icon" or "android". This could be a problem in case of replacement.

Then call **cordova prepare** or/and **cordova build**.

Some functions require a function as a callback.

In this doc, this function is always named "theCallback" and can be any JS function.

This callback function always takes only one parameter which can be : the result OR an error.

If an error occurs, you will always get a JSONObject like this one : {"error":"The message of the error"}

## Example

You can have a list of all available SDK methods by looking at the following file : **your-cordova-app/plugins/com.bma4s.sdk.plugins.cordova/example/example.html**

If you want to use this file, please copy the content into **your-cordova-app/www/index.html** and do not forget to copy our stylesheet too (**example/css/example.css**).

If you still need more information (about methods compatibilities for instance), you can look our JS code at : **your-cordova-app/plugins/com.bma4s.sdk.plugins.cordova/www/BMA4S.js**

## Manage GDPR compliancy

There are two steps for sending user opt-in for data collection to Accengage.

First, there is a parameter that you can override in order to let us know about user opt-in or opt-out for data collection. This parameter is *android\_optin\_data* for Android (*ios\_optin\_data* for iOS). In your json file set it to *true* if you plan to send user opt-in to Accengage :

### Android

#### BMA4SCordovaConfig.json

```
"android_optin_data":true
```

### iOS

#### BMA4SCordovaConfig.json

```
"ios_optin_data":true
```

If the parameter "*android\_optin\_data*" (or *ios\_optin\_data* in case of iOS) is set to "false" or any other value different from "true", our SDK will automatically be launched without the need of calling the "setOptinData" method.

Then, you **MUST** call the method *setOptinData* before any other methods from the Accengage SDK for letting us know the opt-in state of the user.

There are 2 states the user can be in : *TRUE* or *FALSE*.

#### source.js

```
BMA4S.setOptinData(true)
```

If it is *TRUE*, the SDK will start normally. Otherwise it will not start and no data will be collected.

Furthermore, you can block or allow the collect of an user's geolocation by calling the method *setOptinGeoloc*.

Like the Optin Data, there are 2 states : *TRUE* or *FALSE*

### source.js

```
BMA4S.setOptinGeoloc(true)
```

Of course, if setOptinData is set to false, the SDK will not be running, and then the state of setOptinGeoloc will not be used.

## Advanced Integration

Please see platform related (iOS, Android) documentation for more information (methods usage, advanced integration...)

## Push Notifications

### Enable Push Notifications

#### Android

As of April 10, 2018, Google has deprecated Google Cloud Messaging (GCM). This gateway will no longer be available to broadcast Push Notifications to your users as of April 11, 2019.

We highly recommend you to migrate your GCM project to Firebase Cloud Messaging (FCM).

However if you don't plan to migrate yet, you have to integrate manually our GCM plugin from Cordova 1.4.2.

To do so, please follow this [documentation](#).

In order to receive push notifications you need to create a Firebase project first or import your GCM project into Firebase. You can do it either by using **Firebase Assistant** in Android Studio or by using **Firebase console**. Check [this page](#) for more details.

Once a new firebase project and an Android app of this project are created, you need to move the **google-services.json** file into your Android app module root directory.

Then add your own sender Id to the **config/BMA4SCordovaConfig.json** file:

```
"android_senderId": "your-sender-id", // FCM sender id. Example : 123456
```

Please see [Android Documentation](#) for more information.

We recommend to use the same version of firebase as our plugin but if you are using or want to use a different one, you may have some trouble with dependencies conflicts.

You should see the part of the doc about [dependencies conflicts](#) then.

### Integrate our GCM plugin manually

Since our Cordova Plugin is compatible with FCM, if you want to continue to use GCM, you'll need to make some change on the file at **pl**

## atforms/android/com.bma4s.sdk.plugins.cordova/hellocordova-build.gradle.

Please find below an example of what to do :

```
// We recommend to comment the code instead of remove it.
// It will be much easier to just uncomment the lines when you'll need to switch
on FCM later.

dependencies {
    implementation 'com.ad4screen.sdk:A4SSDK:3.8.3'
    implementation 'com.ad4screen.sdk:A4SSDK-Plugin-Beacons:1.3.0'
    implementation ('com.ad4screen.sdk:plugin-play-services:3.3.0') {
        // Comment the following line to use our GCM plugin
        //exclude group: 'com.ad4screen.sdk', module: 'plugin-play-services-push'
    }
    // Comment the following line to remove our Firebase plugin
    //implementation 'com.ad4screen.sdk:plugin-firebase-messaging:2.5.0'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    implementation 'com.android.support:multidex:1.0.3'
}
// Comment the following lines to allow the build without google-services.json
file, which are needed for Firebase
// cdvPluginPostBuildExtras.push({
//   apply plugin: com.google.gms.google.services.GoogleServicesPlugin
// })
```

## iOS

To enable push notifications for iOS, edit your **config/BMA4SCordovaConfig.json** file with the following line:

```
"ios_allowPushNotification":true, // Use iOS push notifications feature
```

To enable multiple language support for interactive notifications, add the language localization to your project:

▼ Localizations	⇅	Array	(6 items)
Item 0		String	es ⇅
Item 1		String	ru ⇅
Item 2		String	Italian ⇅
Item 3		String	German ⇅
Item 4		String	French ⇅
Item 5		String	English ⇅

## iOS 10 Rich Notifications

The latest version of the plugin supports iOS 10 Rich Notifications. However, to be able to receive these notifications you'll need to add the corresponding extension and the AccengageExtension framework manually to your project.

Please see our [iOS Media attachments Documentation](#) for the required steps and more information on this topic.

## Prevent notifications tracking

You can disable/enable the tracking of notifications by using the following method :

```
BMA4S.setPushNotificationLocked(true); // or false to reenale it.
```

You can get the lock status by using the following method :

```
BMA4S.isPushNotificationLocked(callback)
```

For more informations, please consult [iOS native documentation](#).

This won't affect In-App messages tracking.

## Tracking

### Events

#### Important

To use the additional parameters of the custom events for targeting purposes, they must be formatted as a valid `JSON` string. Only strings are accepted as key. For the value, only strings, numbers and booleans are accepted. Use a timestamp to represent a date.

If you want to track a custom event use the `trackEvent` method. Events types below 1000 are reserved for the library internal usage. You can use custom event types starting from 1001.

```
var eventParams = {
  "string-type-field": "value",
  "bool-type-field": true,
  "number-type-field": 7201,
  "date-type-field": Math.floor(Date.now() / 1000)
};
BMA4S.trackEvent(1001, JSON.stringify(eventParams));
```

#### Note

Each event needs to be defined in the [Accengage User Interface](#). Go to Settings > Advanced Settings, and add a new event of type Custom with the code used in `trackEvent` method. The label is only used for a display purpose.

### Custom events (iOS only)

Event API has been updated. You can now use this method for example :

```
var date = new Date('January 1, 2015 12:00:00');
BMA4S.trackCustomEvent(1000, {'Param1':2, 'Param2':'Start', 'Date': date});
```

## Using Analytics

You can track specific events like “Add to Cart”, “Purchase” and “Lead”. Here is how to use each of these events.

### Lead

If you want to send a Lead, you can add to your code:

```
BMA4S.trackLead('Lead Label','Lead Value')
```

### Add to Cart

If you want to track an Add to Cart event, add to your code:

```
BMA4S.trackAddToCart('CartId', 'ArticleID', 'Label', 'Category', 'Currency', 12.30, 1)
```

**Note:** Currency should be a valid 3 letters [ISO4217](#) currency (EUR,USD,..) and the 2 last arguments are price and quantity.

### Purchase

If you want to track a Purchase event, you can add to your code:

```
BMA4S.trackPurchase('Purchase ID', 'Purchase Currency', 12.30)
```

**Note:** Currency should be a valid 3 letters [ISO4217](#) currency (EUR,USD,..) and the last argument is the total price of this purchase.

## Update Device Information

You can create a device profile for each device in order to qualify the profile (for example, registering whether the user is opt in for or out of some categories of notifications). A device profile is a set of key/value that are uploaded to Accengage server. In order to update information about a device profile, add the following lines to your code:

```
//App starting, update device info
var opencount++; //integer value in database scheme
var pushOptin = 1; //integer value in database scheme
BMA4S.updateDeviceInfo('firstname','Jack"); //string value in database scheme
BMA4S.updateDeviceInfo('age',"21"); //integer value in database scheme
BMA4S.updateDeviceInfo('openCount', opencount.toString())
BMA4S.updateDeviceInfo('pushOptin', pushOptin.toString());
```

Replace “key” and “value” with the segmentation key name and the value you want to set.

The keys and values must match Accengage user information field names and values to allow information to be correctly updated.  
But please note that they **MUST** be sent in String format when calling updateDeviceInfo() method !

## New API to Update Device Information

A new API is available in order to update device information in a more structured manner.

```
BMA4S.updateDeviceInformation('set', 'artist_name', 'Hercules');
BMA4S.updateDeviceInformation('delete', 'artist_name');
BMA4S.updateDeviceInformation('increment', 'artist_albums', 1);
BMA4S.updateDeviceInformation('decrement', 'artist_albums', 1);
```

## Date Format

You can send a date using the updateDeviceInfo method.

Your date has to be in the following format: "yyyy-MM-dd HH:mm:ss zzz"

## Manually Update Geolocation (Android only)

Geolocation is automatically updated and sent to our servers if "**android\_no\_geoloc**" is false inside **BMA4SCordovaConfig.json**

If you want to manually update the current location of the device, you can call:

```
BMA4S.updateGeolocation(latitude, longitude, altitude, accuracy);
//Example : BMA4S.updateGeolocation(40.12, 2.12, 20, 50)
```

## Enable/Disable Location Services (iOS)

You can enable geolocation, geofence and beacon services at any time by calling:

```
BMA4S.setGeoLocEnabled(true);
BMA4S.setBeaconServicesEnabled(true);
BMA4S.setGeofenceServicesEnabled(true);
```

To disable geolocation, geofence and beacon services, simply use the following:

```
BMA4S.setGeoLocEnabled(false);
BMA4S.setBeaconServicesEnabled(false);
BMA4S.setGeofenceServicesEnabled(false);
```

## Views

If you want to tag each Fragment/Tabs/Activity and be able to target it, you can use the following code as soon as a view is displayed:

```
BMA4S.setView('DashBoard')
```

Where your-view is the name of your view.

## States

In Accengage Interface, you can “Declare a State” (Applications Advanced Parameters). States are like browser cookies which can be used to trigger Accengage In-App notifications. You can target one or more states and combine them with Event or ViewTag.

Example: When event 5000 is triggered and state “search” contains “pizz\*”, display a specific In-App.

The following section explains how to declare certain states with the Accengage SDK.

### Put a State

You can obtain the name of the state from the Accengage “value” field in Applications Advanced Parameters States section.

In order to put a state, write this code:

```
BMA4S.putState('search', 'myValue')
```

Where "search" is the state name and “myValue” is the value you want to put for this state.

## Subscription Tag

Subscription tag is a new API used to mark user interest for a particular topic with a category, an id and optional parameters :

BMA4S.setDeviceTag(category, id, params), where params is a json containing some optional parameters for the Tag.

Example :

```
BMA4S.setDeviceTag('Football', 'Brésil', {'club_league_1':'corinthians',  
'club_league_2':'figueirense'});
```

A device tag can contain up to 5 parameters and all parameter key should be different. Parameter value should be type of String, Number, Boolean or Date.

As mentioned with the name of the method, setDeviceTag is an update of the Subscription Tag, it will override then the previous information.

You can set an expiration date for your Subscription Tag by adding the parameter "exp" to the Device Tag where the value is a timestamp for the expiration date :

```
BMA4S.setDeviceTag('Football', 'Brésil', {'club_league_1':'corinthians',  
'club_league_2':'figueirense', 'exp':'1543512073'});
```

You can unsubscribe a user to a particular topic with the following code :

Example :

```
BMA4S.deleteDeviceTag('Football', 'Brésil');
```

### Important

Please note that this feature is only available through Accengage's API

For more information please contact our Support Team at <http://ticket.accengage.com>

## Static Lists

First of all, to use a Static List, you need to create it on the Accengage User Interface.

### Add the current device to a Static List

In order to add the current device to a Static List you have to provide a JSONArray containing JSONObjects including each a listId and an optional expireAt expiration date

```
BMA4S.subscribeToLists([{'listId':'sport'},{'listId':'politic',  
expireAt:151646777000}]);
```

### Remove the current device from a Static List

In order to remove the current device to a Static List you have to provide a JSONArray containing JSONObjects including each a listId

```
BMA4S.unsubscribeFromLists([{'listId':'sport'}]);
```

### Get subscription status

You have 2 ways to get the subscription status of the current device.

#### Get subscription status of selected lists

Retrieve the status of all wanted lists, you will need to pass a function in order to retrieve the result (here it's theCallback) :

```
BMA4S.getSubscriptionStatusForLists([{'listId':'sport'}], theCallback)
```

#### Get all subscriptions of the current device

Retrieve the status of all lists, you will need to pass a function in order to retrieve the result (here it's theCallback) :

```
BMA4S.getListOfSubscriptions(theCallback)
```

As a result, you will get a JSONArray containing some JSONObjects :

```
[{  
  "listId":"sport",  
  "name":"Sport list",  
  "expireAt": 0, // timestamp in ms or 0 if no expiration date  
  "status":"subscribed" //can be subscribed or unsubscribed  
}]
```

## Tracking Facebook

Add your Facebook AppID in **config/BMA4SCordovaConfig.json**:

```
"facebookAppId": "your-facebook-app-id", //Facebook application id
```

This will enable Facebook Tracking inside your app, no other step is needed on your behalf.

## Deep Linking

### Retrieve custom parameters from Push/In-App/Geofence/Beacons

If you want to retrieve custom parameters from Push, In-App or Inbox features, call:

```
BMA4S.setCustomParametersListener(your-callback-function)
```

Each time a **Push/In-App** is displayed or clicked, you will receive a response like this one:

#### Android

```
{
  "action": "CLICKED",
  "categories": [
    "PUSH_NOTIFICATIONS"
  ],
  "params": {
    "a4stitle": "",
    "a4sid": "251_530143_16632_3892",
    "collapse_key": "16632",
    "a4scancel": "Cancel",
    "a4scontent": "Rich push with webview",
    "a4sforeground": "y",
    "a4spopup": "y",
    "from": "255322792147",
    "a4surl": "http://google.fr",
    "my-key": "settings",
    "a4sok": "Let's see",
    "a4ssysid": "16632"
  }
}
```

#### iOS

```
{
  "params": {
    "notificationParams": {
      "keyCustomParam1": "valueCustomParam1",
      "keyCustomParam2": "valueCustomParam2"
    },
    "notificationId": "1_2345678_12345678_12345678"
  },
  "action": "CLICKED",
  "categories": [PUSH_NOTIFICATIONS]
}
```

---

## Retrieve Push Notification payload

If you want to retrieve the entire push payload, use the following method:

```
BMA4S.setPushNotificationListener(your-callback-function)
```

Here's a payload example for iOS and Android:

### Android

```
{
  "action": "CLICKED",
  "categories": [
    "PUSH_NOTIFICATIONS"
  ],
  "params": {
    "a4stitle": "",
    "a4sid": "251_530143_16632_3892",
    "collapse_key": "16632",
    "a4scancel": "Cancel",
    "a4scontent": "Rich push with webview",
    "a4sforeground": "y",
    "a4spopup": "y",
    "from": "255322792147",
    "a4surl": "http://google.fr",
    "my-key": "settings",
    "a4sok": "Let's see",
    "a4ssysid": "16632"
  }
}
```

### iOS

```

{
  "params": {
    "a4sd": 1,
    "a4surl": "https://www.google.fr",
    "acc-atts": [
      {
        "att-id": "",
        "att-type": "",
        "att-url": "http://sound.mp3"
      }
    ],
    "aps": {
      "alert": {
        "body": "Demo",
        "title": "push"
      },
      "category": "acc_dismiss_track",
      "mutable-content": 1,
      "sound": "default",
      "thread-id": "demo"
    },
    "notificationId": "6_102478119_100930905_100082542"
  },
  "action": "CLICKED",
  "categories": ["PUSH_NOTIFICATIONS "]
}

```

**action** can contain:

CLICKED	DISPLAYED
CLOSED	TRIGGER

**categories** can contain:

PUSH_NOTIFICATIONS	INAPP_NOTIFICATIONS
BEACON_NOTIFICATIONS	GEOFENCE_NOTIFICATIONS

Only our Android SDK is broadcasting BEACON\_NOTIFICATIONS and GEOFENCE\_NOTIFICATIONS information to third party developer.

All parameters including your custom parameters are inside params (here we have a custom param named 'my-key')

## Geofence/Beacons

For a **Geofence/Beacon**, you will receive a response like this one :

```
{
  "action": "TRIGGER",
  "categories": [
    "GEOFENCE_NOTIFICATIONS"
  ],
  "params": {
    "com.ad4screen.sdk.extra.GEOFENCE_PAYLOAD": {
      "transition": 1,
      "ids": [
        "1921",
        "4282"
      ],
      "triggeringLocation": "{\"latitude\":48.8699786,\"longitude\":2.3347858,\"altitude\":0,\"accuracy\":26.370008392334,\"bearing\":0,\"provider\":\"fused\",\"speed\":0,\"time\":1454670928159}"
    }
  }
}
```

Here, we have entered 2 geofences.

**transition :**

- 1 for enter
- 0 for exit

**For a beacon, the key will be**

```
com.ad4screen.sdk.extra.BEACON_PAYLOAD
```

instead of

```
com.ad4screen.sdk.extra.GEOFENCE_PAYLOAD
```

More details can be found on [Android#RetrievingBeaconinformationwithaBroadcastReceiver](#)

## Stop listening for custom parameters

If you don't want to retrieve custom parameters anymore, please call:

```
BMA4S.removeCustomParametersListener()
```

# In-App Notifications

## Prevent In-App notification display

If you don't want to display any inapp messages temporarily, you can use:

```
BMA4S.setInAppDisplayLocked(true)
```

This function allows you to enable or disable any In-App notification display. By default, its value is false which means that InApp display is enabled.

Once Inapp is locked, no In-App visual element will be displayed until unlocked. Please use this function with care!

## Inbox

Inbox is not yet available in our Cordova Plugin and must be implemented the native way

For iOS, please see [iOS Documentation](#).

For Android, please see [Android Documentation](#).

## Troubleshooting

### Find your A4SID (or Device ID or Mobinaute ID)

If you need to get the Accengage Device ID (or "Mobinaute Id") of the user, use the following method:

```
BMA4S.getIdfv(theCallback)
```

### Restrict SDK connection

You can temporarily restrict the connections in order to tell our SDK to stop flushing network requests.

```
BMA4S.setRestrictedConnection(true)
```

All requests will be locked until you use `setRestrictedConnection(false)`. In order to know if the connection is restricted for our SDK, please use:

```
BMA4S.isRestrictedConnection(theCallback)
```

### Specific for Android

## Activate Logs

To enable logs in the console, you can modify the following value of BMA4SCordovaConfig.json :

```
"android_activate_logging":true
```

You will have to rebuild your app in order to see the logs

## Dependencies conflicts

If you encounter conflicts regarding Google Play Services versions and Firebase versions, you have two possibilities :

- use the same versions as our plugin ones
- override the versions of our plugin

Please find below an example for the second point :

You'll have to modify the two following files.

### platforms/android/com.bma4s.sdk.plugins.cordova/hellocordova-build.gradle

```
dependencies {
    implementation 'com.ad4screen.sdk:A4SSDK:x.y.z'
    implementation ('com.ad4screen.sdk:plugin-play-services:x.y.z') {
        // Just exclude google play services from our plugin, we will override it in
        build-extras.gradle
        exclude group: 'com.google.android.gms'
    }
    implementation ('com.ad4screen.sdk:plugin-firebase-messaging:x.y.z'){
        // Just exclude firebase from our plugin, we will override it in
        build-extras.gradle
        exclude group: 'com.google.firebase'
    }
}
```

### platforms/android/app/build-extras.gradle

```
dependencies {
    // In order to override the google play services versions, you'll have to add the
    following lines
    implementation 'com.google.android.gms:play-services-base:x.y.z'
    implementation 'com.google.android.gms:play-services-gcm:x.y.z'
    implementation 'com.google.android.gms:play-services-location:x.y.z'
    implementation 'com.google.android.gms:play-services-ads:x.y.z'

    // In order to override the version of firebase, you'll have to add the following
    line
    implementation "com.google.firebase:firebase-messaging:x.y.z"
}
```

## Others

Please see the [Android Troubleshooting](#) and [Android Optional Code](#).

## Specific for iOS

### Activate Logs

To enable logs in the console, you can use the method :

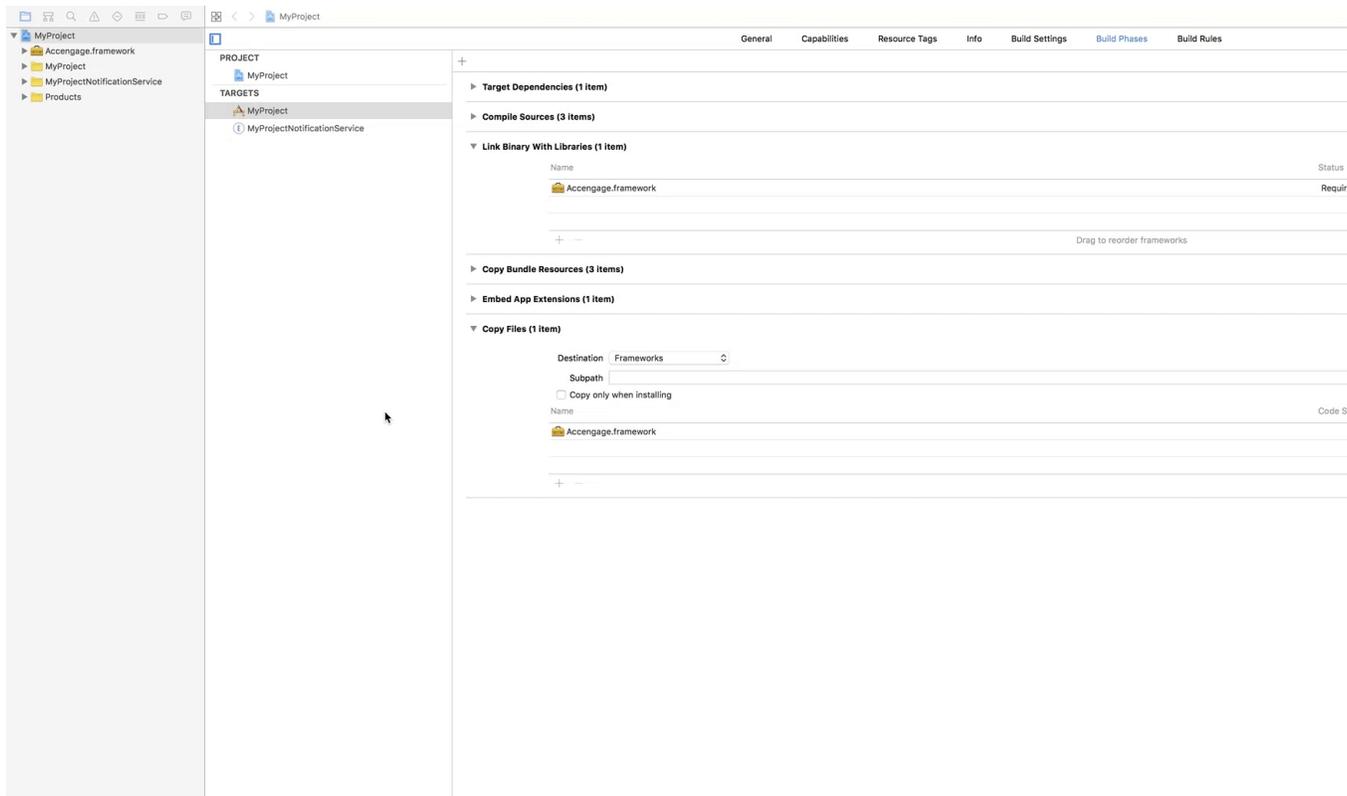
```
BMA4S.setDebugMode(true)
```

To enable logs in the Accengage logging application, please follow this [documentation](#).

### AppStore unsupported architectures

Accengage's frameworks contains architectures for iPhone simulators (x86\_64 and i386). If you try to submit an application that contains these architectures to AppStore, it will be rejected by Apple.

To fix that behaviour, go on Build Phases on XCode and add a new Run Script Phase. Then paste the content of [this script](#), in order to eliminate simulator architecture slices.



The order of the build phases is important. The Run Script phase must be added after (or placed below) the Copy Files phase.

## Others

Please see the [iOS Troubleshooting](#).