**FACEBOOK PROPRIETARY AND CONFIDENTIAL**

**FB Install Tracking**

**iOS**

**Basic flow**

When there is an installation event happening for a 3rd-party app, analytics partners (via their SDKs on this 3rd-party app) fetch fb-specific mobile cookie on UIPasteboard. Then they ping back to an fb server-side endpoint with this information, so that fb can log this installation.

Later on, fb will also provide another server-side endpoint that analytics partners can call in order to know which of these installations were actually from fb and the detailed information associated with these installations like organic vs. paid, which part of fb (newsfeed, notification, bookmarks and etc.), and so on. The spec of this endpoint is not in this doc, but we will provide it later when we finalize it.

**Fetching Mobile cookie**

The mobile cookie is placed in a UIPasteboard named "fb\_app\_attribution".  Below is a code snippet to fetch this id from the named UIPasteboard.

UIPasteboard \*pb = [UIPasteboard

    pasteboardWithName:@"fb\_app\_attribution"

    create:NO];

if (pb) {

  NSString \*attributionID = pb.string;

  // Do something with attributionID here

}

**Mobile cookie format**

The mobile cookie is created by FB iOS app using CFUUIDCreateString and is of typical 128-bit UUID string representation (<http://en.wikipedia.org/wiki/Universally_unique_identifier>). This is an implementation detail though and could change in future. It is not derived from any user or device attributes.

**Ping-back endpoint**

Analytics partners first register an app on fb (<https://developers.facebook.com/docs/appsonfacebook/tutorial/>, only need to create a dummy app very fast without caring about all the fields) and then obtain app access token (<https://developers.facebook.com/docs/authentication/applications/>). Afterwards, they can start to make graph API calls to the endpoint mentioned below. They should ping back to this endpoint from their servers because they need to pass their app access token in order for fb to authenticate them and it will be unsafe for them if they put their app access token at client side.

API spec:

POST /{3rd\_party\_app\_id}/activities

Parameters:

* access\_token: the analytics partner's app access token, so that we know which partner to tie the distribution to
* event: MOBILE\_APP\_INSTALL
* attribution: mobile\_cookie from the user's device

Return: true or false to indicate whether the post was successful or not.

**Android**

We append referrer param to all the 3rd-party apps' URLs inside FB Android app, so that they will know the traffic is from fb, and they should also ping back to fb about the installation events. URL format,

market://details?id=<package>&referrer=  
  urlencode(  
    utm\_source=apps.facebook.com&  
    utm\_campaign=fb4a&  
    utm\_content=  
      urlencode(  
        {"app":<app-id>,"t":<timestamp>}  
      )  
  )

**Ping-back endpoint**

To ping back to fb, analytics partners follow the same endpoint and API interface as mentioned above for iOS. And the only difference here is that, they do not include "attribution" field in the API call.

**Caveat**

This approach for Android tracking will only last for a couple of months, and later we will migrate to use the same “pasteboard” approach for iOS as mentioned above. The purpose is to standardize the tracking mechanism across different mobile platforms, and provide consistent and frictionless adoption experiences for our partners.

**FB Re-engagement Referrals**

**iOS**

If the user has your application installed, and has authenticated your app with Facebook, here is the URL we open:

fb[app id]://authorize#expires\_in=0&access\_token=[token]&target\_url=[Linked URL]

If the app is installed, but the user has not authenticated it with Facebook, the URL format is:

fb[app id]://authorize#target\_url=[Linked URL]

**Android**

For Android referrals to installed apps, the intent would be **'com.facebook.application.[applicationId]**' and the 'access\_token' is in the extras bundle of the intent, if the user is already authorized to the app.

​