

Mobile: What Are iOS Campaign Callbacks and How Are They Used?

Brief Overview

As of WalkMe Mobile SDK version 1.6.0 the WalkMe Mobile SDK supports campaign callbacks.

Callbacks are a programmatic way for an app to register to WalkMe Campaign events. By implementing the Campaign Callbacks interface, apps can use the data transferred through the callback objects (including Campaign and user information) and utilize it for the app's purposes, such as passing that information to any analytics or CRM system unrelated to WalkMe, or updating any app or user settings according to the user interaction with the Campaign.

Use Cases

Campaign callbacks can be used for the following and many more:

- Track user onboarding from any analytics system
- Generate opportunities in a CRM based on user engagement with WalkMe Shoutout - i.e. "Are you interested in a free demo? Yes / No"
- Integrate with existing support systems to track and compare WalkThru usage/completion against the top support ticket topics

Below is a description of all the required steps in order to implement campaign callbacks.

Using Campaign Callbacks

To be able to register to campaign callbacks, first implement this protocol:

```
/**
 * Interface definition for a callback to be invoked in Campaign actions.
 */
@protocol WMCampaignInfoDelegate <NSObject>

/**
 * Called after campaign was dismissed.
 *
 * @param campaignInfo The dismissed campaign info.
 */
```

```
- (void)campaignDidDismiss:(WMCampaignInfo *)campaignInfo;
```

```
@end
```

After implementing the protocol, use the `setCampaignInfoDelegate` method to register to the campaign callback events:

```
/**
 * Register a delegate to campaign events
 *
 * @param delegate The delegate
 *
 */
+ (void)setCampaignInfoDelegate:(id<WMCampaignInfoDelegate>)delegate;
```

The callback return object will be of class `WMCampaignInfo`, which includes the following information:

Item	Retrieved By	Description
Campaign CTA	campaignCta	The campaign CTA that the app user has interacted with. For all CTAs other than the custom one, the CTA returned will be: <code>abbi://cta</code> , for example: abbi://ok or abbi://never
Campaign ID	campaignId	The unique ID assigned to the campaign
Campaign CTA ID	campaignCtaId	The unique ID for the Campaign CTA
User Data	userData	An object holding information on the user who interacted with the campaign. See full description of the object below.
Campaign Data	campaignData	A placeholder object added for possible future use of additional data appended to the campaign. Survey submission data is populated here.

```
@interface WMCampaignInfo : NSObject
```

```
@property (nonatomic, strong) NSString* campaignCta;
@property (nonatomic, strong) NSString* campaignCtaId;
@property (nonatomic, strong) NSString* campaignId;
@property (nonatomic, strong) WMUserData* userData;
@property (nonatomic, strong) NSDictionary* campaignData;
```

```
@end
```

As mentioned above, WMCampaignInfo includes the object WMUserData, which is described below:

Item	Retrieved By	Description
Public User Attributes	*userAttributes	All of the public user attributes that were set for the app user by the moment the user interacted with the campaign. The list will always include the attribute key and its value, e.g. { "lead_id" : "12345" , "account_type" : "family" }
Private User Attributes	*privateUserAttributes	All of the private user attributes that were set for the app user by the moment the user interacted with the campaign. The list will always include the attribute key and its value, e.g. { "lead_id" : "12345" , "account_type" : "family" }
Session Duration	sessionDuration	How long was the user session at the moment the user made the campaign interaction
iOS Version	systemVersion	The user's OS version
Device Unique ID	deviceId	The device unique ID
Device Model	deviceModel	The user's device model
Device Orientation	deviceOrientation	The device orientation at the moment the user interacted with the campaign
App Version	appVersion	The app version the user is using
App Name	appName	The app name
Locale	locale	The user's device locale

SDK Version	sdkVer	The SDK version integrated with the app the user is using
Session ID	sessionId	The unique session ID generated by WalkMe
Push Notifications Status	isPushEnabled	Has the user enabled push notification for the app (true / false). If the app does not feature push notifications - the value will be "false".
Device Timezone	timezone	The user timezone as set on the device.
Network	network	The network type the user is using: WiFi / Cellular / Offline
System Name	systemName	The user's OS name - in this case will always be "iOS"
Current user timestamp	timestamp	The user's current timestamp

This is the structure for the WMUserData object:

```
@interface WMUserData : NSObject

/**
 * User Attributes in current session.
 */
@property (nonatomic,strong) NSDictionary *userAttributes;

/**
 * Private User Attributes in current session.
 */
@property (nonatomic,strong) NSDictionary *privateUserAttributes;

/**
 * User Session Duration in second.
 */
@property (nonatomic,strong) NSNumber* sessionDuration;

/**
 * User's iOS version
```

```
*/
@property (nonatomic,strong) NSString* systemVersion;

/**
 * Device Unique ID
 */
@property (nonatomic,strong) NSString* deviceId;

/**
 * Device Model.
 */
@property (nonatomic,strong) NSString* deviceModel;

/**
 * Device Orientation PRT_REG / LSL.
 */
@property (nonatomic,strong) NSString* deviceOrientation;

/**
 * The application version.
 */
@property (nonatomic,strong) NSString* appVersion;

/**
 * The application name.
 */
@property (nonatomic,strong) NSString* appName;

/**
 * Locale language for user's device.
 */
@property (nonatomic,strong) NSString* locale;

/**
 * WalkMe SDK version.
 */
@property (nonatomic,strong) NSString* sdkVer;

/**
 * Session unique id.
 */
@property (nonatomic,strong) NSString* sessionId;
```

```
/**
 * If Push notification approved for user.
 */
@property (nonatomic,strong) NSString* isPushEnabled;

/**
 * User device timezone.
 */
@property (nonatomic,strong) NSString* timezone;

/**
 * User current network ( WIFI / 3G ).
 */
@property (nonatomic,strong) NSString* network;

/**
 * User OS System Name.
 */
@property (nonatomic,strong) NSString* systemName;

/**
 * Current Timestamp for this User.
 */
@property (nonatomic,strong) NSString* timestamp;

@end
```