Marketing Cloud MobilePush Implementation Guide

Salesforce, Winter '24

'24

CONTENTS

MARKETING CLOUD MOBILEPUSH IMPLEMENTATION GUIDE

Marketing Cloud MobilePush provides capabilities to send push notifications and inbox messages, including deals or discounts, to drive increased app usage. Additional features include geofence and beacon messages and app analytics, such as push received, push open, and time in app.

Currently, MobilePush supports native implementations for Android and iOS and offers a plugin for Cordova. Developer documentation for our SDKs can be found here.

Who This Guide Is For

The MobilePush Implementation Guide is designed to guide customers and developers through the key methodologies and steps that have been proven successful in large-scale global implementations by our own Marketing Cloud subject matter experts.

Before You Start

MobilePush is included in Corporate and Enterprise editions. These prerequisites are required before push notifications can be sent to devices from the Marketing Cloud.

Get Started in Marketing Cloud MobilePush

Marketing Cloud MobilePush is included in Corporate and Enterprise editions. These prerequisites are required before push notifications can be sent to devices from the Marketing Cloud.

For App Developers

Before you can use MobilePush, ask your app developer to follow these required steps to implement the MobilePush SDK.

1. Provision Android devices.
2. Provision iOS devices.
3. Create an app in MobilePush: Android and iOS. This step makes your app available in the MobilePush application. Also, MobilePush tells Google and Apple that MobilePush can send messages to your mobile app on users' devices.

   📝 **Note:** Note the app ID, access token, and the autogenerated app endpoint. Copy the app endpoint into your project using the latest SDK.

4. Integrate the MobilePush SDK in your app: Android and iOS. To use MobilePush, integrate your app with the MobilePush SDK.

For Marketers

After your app developer finishes implementing the MobilePush SDK, your mobile app appears in the MobilePush application. Now, immediately send a test push notification.

1. In the app switcher, hover over your name and click **Setup**.

2. Search for *MobilePush*.

3. Create an audience.

   a. Import contacts. The imported list has the contacts that you want in MobilePush to eventually send messages to.

   b. Create a filtered list of contacts. A filtered list includes the contacts that you want to send a specific message to.

4. Create and send a push notification.

5. Review message activity. Later, review the number of messages sent and the percent opened.

SET UP MARKETING CLOUD MOBILEPUSH

Once you have completed the prerequisites, connect your MobilePush app to the correct Marketing Cloud account and add provisioning information.

Add Provisioning Info to Your App

To connect your mobile app to the correct Marketing Cloud account, add the provisioning information to the MobilePush app.

> **Note:** Auth Keys point to the APNs production server by default. MobilePush does not currently provide an option for customers to point Auth Keys to the sandbox APNs server.

1. In the app switcher, hover over your name and click **Setup**.

2. Search for `MobilePush`.

3. To select the app to update, click the arrow by the app and select **Settings**.

4. You can update existing fields such as name, description, or the icon that represents your app.

5. The required app endpoint is automatically generated. Copy the app endpoint into your project using the latest Android or iOS SDK.

6. Enter Sending Services credentials:

   For iOS .p8 Auth Key:

   - Select the environment that your application is registered to send push notifications. Select either **Production** or **Development**.
   - Obtain these identifiers from the iOS Provisioning Portal:
     - Auth Key: unmodified .p8 file
     - Key ID: 10-character key ID associated with the .p8 file
     - Team ID: 10-character Team ID associated with your Apple developer account
     - App Bundle ID: ID for the app you're using with MobilePush

   - Upload all four credentials in MobilePush Administration.

     > **Tip:** You can use the same Auth Key, Key ID, and Team ID for multiple MobilePush apps, but your Bundle ID must be unique.

   - Save your changes.

   For iOS .p12 certificates:

   > **Note:** When APNs certificates expire for a specific app, MobilePush prevents you from creating and sending messages through that app. MobilePush also prevents selecting a message to use in Journey Builder or Automation Studio. If your app is provisioned with iOS and Android, sends to Android devices through apps with expired APNs certificates are not affected.

   - In the APNs Client field, click Choose File and upload the APNs certificate provided when you created your app in the iOS Provisioning Portal.
   - Enter the APNs certificate password, if applicable.
   - Enter the Bundle ID.
   - Save your changes.

For Google

- In the Android Client field, enter the server key from your Firebase console in the API Key field.

7. Enter or update optional settings.

8. After you populate the required fields for all applicable application platforms, click **Save**.

9. Review the information you provided and check for any potential errors.

10. Click **Finish**.

The MobilePush Setup page shows a message that the process is completed successfully with a summary of your app information.

MARKETING CLOUD MOBILEPUSH USE CASES

Use cases expand Marketing Cloud MobilePush's functionality and the reach of your messages. For example, use lists and tags to segment your audience and send to contacts with the same attribute values.

Marketing Cloud has several products available. Each use case has suggestions for which combination of products are required in order to implement the solution for a specific use case. However, for all use cases described in this document, MobilePush will need to be configured for your account.

Implement Tags in Marketing Cloud MobilePush

To help better segment your contacts into more precise audiences, use tags in Marketing Cloud MobilePush.

🛑 **Important:** The tag limit per device is 100.

Tags hold additional information about your subscribers and how they use your app. You can assign values to contacts interested in hiking, fashion, or sports.

For example, Northern Trail Outfitters (NTO) can create their app to assign tags to a contact record based on a preferences section. That section can ask the contact to indicate their interest in several different topics, such as hiking, boating, or fishing. After the app assigns those values to a contact record, NTO can use those values when creating filtered lists.

Ensure that your app developer includes the proper tag terms in your app via the Journey Builder for Apps SDK.

Create a Filtered List Using Tags

Select which tag values to filter a list by when you create a filtered list in MobilePush, just like you select attributes or data extension values. For example, select the tag hiking to send hiking-related messages to contacts. You can combine tags to send messages to combined groups of contacts. For example, send the same message to fishing and boating audiences who could be interested in the same content.

Send Message Using Tags Via API

After you implement tags in Marketing Cloud MobilePush, authenticate your app and use the messageTag REST API call to send messages based on tag values.

👁 **Example:**

```
POST /push/v1/messageTag/ODM6MTE0OjA/send HTTP/1.1
Authorization: Bearer YOUR_API_TOKEN_VALUE
Content-Type: application/json
Host: www.exacttargetapis.com
Connection: close
User-Agent: Paw/2.2.2 (Macintosh; OS X/10.10.4) GCDHTTPRequest
Content-Length: 83
{"InclusionTags":["Hiking"],
"Override":"true",
"MessageText":"Hiking Only Message"}
```

Who Receives These Messages?

If contacts in Marketing Cloud MobilePush meet certain conditions, they receive push notifications .

- At least one device for the contact matches filtered list values including any applicable tags
- Contact's status is Active in Marketing Cloud
- Contact enabled notifications in the app on their device
- Contact opted to receive notifications for preferences on that mobile device

Review these example cases on how the MobilePush makes an API call using the hiking tag to reach subscribers.

Device	Subscriber	Opted In	Status	Notifications	Tags	Will Receive?
Tablet 1	john@example.com	Yes	Active	Enabled	• Hiking • Boating • Fishing	Yes
Phone 1	john@example.com	Yes	Active	Enabled	• Boating • Fishing	No
Tablet 2	angela@example.com	Yes	Active	Enabled	• Hiking	Yes
Phone 2	angela@example.com	No	Deleted	Disabled	• Hiking	No

Include Only Opted-in Contacts in a Filtered List

Just because a contact is active doesn't mean that they have opted in to receive Marketing Cloud MobilePush notifications.

Contacts receive a push message if they satisfy these conditions.

- Their status is active in Marketing Cloud.
- They meet the criteria for the list that you're sending to.

An active contact is not automatically opted in to receive notifications for your app. For example, if you build a filtered list where **status = active** and **platform = iPhone OS**, the initial list includes contacts who won't receive the message because they're not opted in. MobilePush doesn't send to devices that have opted out. Therefore, the number of active contacts may not be the same as your sendable contact list.

To ensure that you only include the contacts who are opted in to receive your app's messages, build your filtered list using the following information:

- Opt In Status: Opted In
- Platform: [Include either iPhone OS or Android OS]
- Application: [Your app name]
- Status: Active

Set the Subscriber Key or Contact Key

Contact key, also known as subscriber key, is the unique identifier used to aggregate a contact's devices within Marketing Cloud. Set the contact key to a specific value provided by your customer or to another unique identifier for the contact. You can use a mobile number, customer number, Salesforce ContactID, or another value.

Refer to the iOS and Android SDK documentation for specific details on how to set the contact key.

Use Exclusion Lists and Data Extensions Lists in MobilePush

To exclude certain devices from receiving push notifications from Marketing Cloud MobilePush, use exclusion lists.

📝 **Note:** Suppression lists are not supported for MobilePush and can still send notifications to the device.

Use an existing list as an exclusion list when you send your push notification.

1. Create a data extension called PushSendLogDE.

   📝 **Note:** To enable the PushSendLog template in your account, contact your Marketing Cloud admin.

   a. In addition to the existing template fields, create the following fields: ContactKey or SubscriberKey, and Platform.

2. Create an exclusion data extension to devices you want to exclude.

3. Link both of the data extensions using Data Designer in Contact Builder.

   a. Create an attribute group.

   b. Click **Link Data Extensions** and select the PushSendLogDE.

   c. To complete the linking process, select contact key from Customer Data and SubscriberKey from PushSendLogDE and click **Save**.

   d. Repeat this process for the ExclusionDE.

   e. Insert this AMPscript into the PushSendLogDE or ExclusionDE at send time. `InsertData("ExclusionDE", "SubscriberID",SubscriberID, "trackingId", @guid, "messageId", internalMessageId, "deviceId", deviceId, "sendDate", NOW());`

   f. Create an exclusion list from this ExclusionDE, and drag it to the Exclusion List bucket. To apply exclusions at send time, refresh this list before every send. If you are using automation, include Refresh Activity before Send activity.

Use the MobilePush SDK to Send a Location Message to Selected MobilePush Users

Marketing Cloud MobilePush can trigger location messages when users enter or exit a geofence or come within range of a beacon. Since MobilePush does not offer an interface within Marketing Cloud to choose an existing audience list or data extension eligible for receiving a location message, use the SDK to segment location message recipients.

With the SDK, you can segment recipients of a location message.

- Show the message if the user's my store location is within the geofence region.
- Show the message if the user has items in the app shopping cart and if the message has an "abandonedCart":"true" custom key.
- Show the message only when the store is open based on the app's store location list and the region and local time.
- Don't show messages if the user isn't logged in to the app.

- Show specific messages based on the user's customer profile. For example, customers who are coffee lovers get messages about coffee, but customers who prefer pastry receive a message to "add a coffee" when they're in the shop.

Implementation

1. Define which user segments should receive the location messages based on your marketing goals.

2. Create a location message.

3. In the CustomKeys field, enter a key that the app uses to match to the user profile. Example, loverType = coffee.

4. Ask your app developer to review the SDK instructions to suppress messages that do not match the value within custom keys.

📝 **Note:** See the iOS and Android SDK documentation for additional instructions.

Use Deep Linking in MobilePush

To take a user directly to a particular page in the app instead of loading the app, use deep linking in Marketing Cloud MobilePush. Users can be linked from external apps, push notifications, or an email link. Deep linking brings users to specific screens that require action rather than going to the home screen and then navigating.

See the documentation on customizing notification handling for Android and iOS and Android URL handling for further instruction.

Use Multiple Providers for Push Notifications

Implementing a push notification provider's SDK in addition to the Marketing Cloud MobilePush SDK is not recommended. While multiple push SDKs can be integrated into a single app, this practice can cause issues. Results are not guaranteed.

See our iOS and Android SDK documentation for considerations.

MOBILEPUSH FAQS

Frequently asked questions about Marketing Cloud MobilePush. Question types include Implementation, Product, and Troubleshooting. For more information, see the MobilePush help documentation.

Implementation FAQs

Frequently asked implementation questions about Marketing Cloud MobilePush.

How do I add and update MobilePush attributes using the MobilePush SDK?

See the iOS and Android SDK documentation for details on how to use the SDK to update and add attributes.

Can I use REST APIs to update data extensions or refresh lists?

Yes. REST API routes can add or update data extensions and refresh a contact list.

Can I use REST APIs to register devices with MobilePush?

REST APIs are used to register devices, but this is not the preferred method with MobilePush. Use the native SDK for device registration because it opens up access to other features. However, push notifications can be delivered with REST API.

What happens when I enable location services?

When you enable location services, ask your app developer to request the required permissions from your customer. When you enable location services with the consumer's permission, the SDK gets the user's location when appropriate for proper SDK operation.

What data are provided to the Marketing Cloud when I enable location services or beacons?

We use latitude and longitude data to determine which messages to display, but we only store information on which messages displayed to a given user and which geofences may have triggered that information. We do not store a user's location information as part of this process. For MobilePush iOS SDK, location must be enabled in the SDK and by the device user. MobilePush Android SDK requires the app developer to enable location in their configuration. The SDK is built to minimize the data burden on the device. We use the same location call that is used for location services to get a list of nearby beacons.

Can I use exclusion lists or suppression lists to exclude contacts from a send?

Use exclusion lists to exclude contacts from a send. Suppression lists are not supported in MobilePush.

Are there installation services or accelerators for MobilePush?

There are no dedicated installation services for MobilePush. Use the help documentation and the SDK installation instructions to get started.

As of April 2019, the MobilePush accelerator is no longer available.

How big are the MobilePush SDKs?

With your app size in mind, the MobilePush SDK team works to keep the SDK size small each release. Because the SDK size changes with each release, it is best to download the SDK and review its size every release.

Can I use multiple push notification providers in my app?

Implementing a push notification provider's SDK in addition to the Marketing Cloud MobilePush SDK is not recommended. While multiple push SDKs can be integrated into a single app, this practice can cause issues. Results are not guaranteed.

See our iOS and Android SDK documentation for considerations.

Can I use a third-party plugin, such as React Native, Cordova, or Xamarin?

Marketing Cloud supports the use of React Native and Cordova via a provided plug-in project, but doesn't offer plugins for other hybrid mobile platforms as a standard integration with MobilePush. We strongly encourage using our native SDKs for iOS and Android to integrate your mobile app with Marketing Cloud. Any integration using a hybrid mobile platform other than Cordova is considered a custom implementation and does not meet the requirements for standard or premier support.

MobilePush FAQs

Frequently asked product questions about Marketing Cloud MobilePush.

How is a contact related to a device?

A contact can have multiple mobile devices, but a device can't have multiple contacts. If a new contact logs in to a device, Marketing Cloud associates the device with the new contact key. This switch typically takes 15 minutes.

How does a user opt in or out of receiving messages?

For iOS, your app developer sets up your app to prompt the contact to opt in to notifications. For Android, notifications are enabled by default. To use location messages, prompt contacts to enable location tracking.

To opt in, the contact downloads your app and enables notifications. To enable notifications, the contact can Use notifications on your iPhone or iPad or Control notifications on Android.

If a contact deletes your app or disables notifications, Apple or Google opts out the contact.

Why implement the SDK when using MobilePush?

Although you can use MobilePush by implementing only APIs, the SDK gives you access to more features. If you use the APIs without the SDK, these issues can arise.

- Tracking opt-ins and opt-outs or changes made to the device at any level isn't reflected in Marketing Cloud.
- You can't use rich push, inbox messaging, location-based messaging, and in-app messaging.
- Device analytics are unavailable.

How does device registration work with the MobilePush SDK?

The SDK registers a device when the SDK is initialized. If a login is required, you might want to delay registering a device until the login is completed and the user is a known contact. When a contactKey becomes known, a new contact record is created, and the deviceID of the unknown contact is reassigned to a known contact. See the iOS and Android SDK documentation for more information.

If a user has multiple devices, how do I send to only one of them?

You can target a specific device using device-level attributes combined with AMPscript. For example, you want to target only iOS devices or devices that haven't enabled location sharing. Identify these characteristics using default attributes, such as platform, or customized tags or attributes. When you create a message, you can provide AMPscript that generates a message only to the intended device of a given user. For example:

```
%%[IF Platform == 'iPhone OS' THEN]%%Hello from Marketing Cloud!%%[ENDIF]%%
```

What constitutes a bounce message?

Because the Apple Push Notification Service (APNS) doesn't guarantee send-time feedback, iOS devices don't have the concept of a bounce message. Instead, APNS provides a feedback service that Marketing Cloud calls every 12 hours to determine if there are devices to opt out.

Android provides send-time feedback. A message is flagged as bounced when Firebase Cloud Messaging (FCM) responds with a NotRegistered error. For Android devices, if a message is sent unsuccessfully to a device three times, the device is considered opted out. If a device re-registers after opt-out, the bounces are cleared and messages are sent again.

If a user's device is turned off, is that treated as a bounce or is the push delivered in the background?

If a user's device is off, the push is undelivered and enters a queue with the messaging provider. The push is delivered after the device is turned on. The length that a message stays in this queue isn't explicitly guaranteed with APNS or FCM.

When a device is off, out of the country, or in a location with no service, pushes that expire from the queue don't flag those devices as unregistered, opted out, or otherwise undeliverable. However, the messages are purged from the queue when they expire.

If a user disables push notifications with notification settings or uninstalls the app, does that get pushed back to the operating system vendor and treated as a bounce?

If a user uninstalls the app, the message bounces.

If push notifications are disabled at the operating system level but the app remains installed, the next registration payload from the app registers the opt-out.

On iOS, disabling push notifications doesn't unregister the device. Instead, push notification payloads are received but aren't delivered. The device doesn't appear in APNS feedback as a result of disabling push notifications.

On Android, opting out at the operating system level unregisters the app with FCM, and push messages bounce.

How does Marketing Cloud behave when MobilePush records are imported and then a user opts in using the mobile app?

When a user opts in to a mobile app with the SDK installed, imported contacts are overwritten.

For example, if the device token hasn't changed:

- Contact ABC is imported with device token XYZ.
- The user logs in to the mobile app, and a registration call is triggered via the SDK for device token XYZ.
- A new contact EFG is created with a deviceID 567 (generated by the SDK).
- The imported contact ABC is disassociated with device token XYZ. The FCM and APNS tokens are reassigned to the SDK-identified contact EFG.

How long do I wait until see my attributes in Marketing Cloud?

If the device was provisioned and registered correctly, allow up to 15 minutes for the attributes to be processed and appear in Marketing Cloud.

Does MobilePush use encryption in the SDK?

Yes. AES 256 is used to encrypt the database used in the SDK.

What are device IDs and system/device tokens?

A device ID, sometimes referred to as an installation ID, is a globally unique identifier (GUID) that the SDK randomly generates each time an app is installed on a device. The device ID isn't used as a persistent unique identifier for the physical device. A system/device token is generated by Google or Apple when the device registers to receive push notifications.

How do I know which devices are still active?

When you send a push notification, Google provides instant feedback regarding the status of the device.

Apple doesn't guarantee immediate send-time feedback. Apple has a feedback service that provides device status information. Marketing Cloud contacts this feedback service every 12 hours to obtain changes in device status.

What kind of data is captured when the analytics flag is enabled?

The SDK captures these analytic events and sends them back to Marketing Cloud. These flags are subject to change.

Analytic Event	Description
Open	Indicates that a message has been opened.
Display	Used when a local fence or beacon notification is displayed.
Time in App	Counts the time that an app is in the foreground.

Analytic Event	Description
Time in App Opens	Indicates that the app was opened.
Fence Entry	Indicates that a device crossed into a geofence.
Fence Exit	Indicates that a device left a geofence.
Time in Location	Captures the amount of time that a device is inside a geofence.
Beacon in Range	Captures that a device is within range of a beacon.
Time with Beacon in Range	Captures the amount of time that a device is within range of a beacon.
Inbox Download	Indicates that an inbox message has been downloaded to the device.
Inbox Open	Indicates that an inbox message has been opened.
Should Show Message	Reports value of 1/0 based on the true/false value return from shouldShowNotification() for Region Messages.

Troubleshooting FAQs

Frequently asked troubleshooting questions about Marketing Cloud MobilePush.

What information do I provide to Salesforce Customer Support?

Before contacting Salesforce Customer Support, check the list of known issues in MobilePush on the Salesforce Trailblazer Community. When you contact Salesforce Customer Support, include the following information in your request:

- MID for the account
- Name of the app
- Device logs
- Apple and Google security certificate information

Why don't I see my message in Journey Builder or Automation Studio?

Check to see if your APNS certificate is valid. APNS certificates are used to send push notifications from MobilePush to iOS devices. When APNS certificates expire for a specific app, MobilePush prevents you from creating and sending messages through that app. You are also prevented from selecting a message to use in Journey Builder or Automation Studio. If your app is provisioned with iOS and Android, sends to Android devices through apps with expired APNS certificates are not affected. Update your certificate in the iOS Provisioning Portal and upload the updated certificate in MobilePush Administration.

Why can't I upload Apple iOS certificates to the Marketing Cloud?

Ensure that your certificates are valid. Check to see if your certificate file contains both a development and production certificate. If both certificates are in the same file, the upload fails.

Why can't I send rich push notifications or add media to my push notification?

Rich push must be enabled for your account to see the Add Media option during message creation. To provision rich notifications, contact Marketing Cloud Support.

Why are devices not receiving push notifications?

- Ensure that devices have opted in to receive push notifications for that app.
- Ensure that the SDK is implemented correctly to register the devices with Marketing Cloud.
- Check the troubleshooting guides for Android and iOS under their respective Github repositories.

How does Marketing Cloud manage Device IDs?

Device ID is a globally unique identifier (GUID) that is randomly generated by the SDK each time an app is installed on a device. Sometimes, this ID is referred to as an Installation ID. Device ID is not used as or considered a persistent unique identifier for the physical device.

Device ID represents an instance of an app installation. The ID does not persist across app uninstall and install cycles. When a contact installs your app, the SDK creates a Device ID.

The SDK creates another Device ID when these events occur:

- The contact purchases a new mobile device and installs your app.
- The contact deletes the app and then later reinstalls your app.
- The contact does a factory-reset of the device and then reinstalls the app.

What happens when a device ID changes due to an uninstall?

By default, SDK registers a new device upon initialization. Since the user is not yet in a known state, the SDK generates a GUID, and a contact record is created for the updated device ID.

To prevent a new contact from being created, delay registering the contact until the appropriate time when they become known. However, depending on our user experience within the application, this approach could reduce the number of contacts that are captured and therefore reachable within Marketing Cloud. This approach is only recommended in applications where user login is required early in the user experience.