Mobile Publisher for Salesforce App

Last Updated:2026.02.25

CONTENTS

Mobile Publisher for Salesforce App

Use Mobile Publisher to create a customized and branded version of the Salesforce mobile app. When your users can identify the app with your brand, they're more likely to use it, which increases adoption.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

Your organization must license Salesforce Mobile Publisher to use the feature. Contact your Salesforce sales rep for more information.

Get Started
Learn everything about the Mobile Publisher program, such as how it works, what information to gather to distribute your app to users, and the assets required to create your branded app.

App Distribution
Choose how you want to get your app into the hands of your users.

Build Your Branded App
Create a Mobile Publisher project, enter details about your app, and upload your custom branded assets to Salesforce.

Test and Submit the App
Request a binary and submit for your branded app to the application stores for approval.

Maintain and Update the App
Learn about the types of maintenance and how you can update your app.

Best Practices and Troubleshooting
This information helps you implement recommendations and troubleshoot when using Mobile Publisher.

Get Started

Learn everything about the Mobile Publisher program, such as how it works, what information to gather to distribute your app to users, and the assets required to create your branded app.

Note For additional assistance with Mobile Publisher initial preparation, setup, configuration, submission, and Apple and Google approval tips and processes, see our Mobile Publisher Pre-Submission Checklist.

How It Works
Let's look at the overall process so you know what to expect when creating your branded app with Mobile Publisher.

Requirements for Mobile Publisher for Salesforce App
Mobile Publisher for Salesforce App is supported on various mobile platforms, with minimum operating system requirements.

Prepare Your App's Information
Preparing your branded assets is the most important part of this process. Let's discuss some of the branding details so you can get started with your Mobile Publisher project.

See Also

YouTube: How to Create Pixel-Perfect Mobile Applications with Salesforce Mobile Publisher

How It Works

Let's look at the overall process so you know what to expect when creating your branded app with Mobile Publisher.

Here are the basic steps:

- Sign up for the Mobile Publisher program. Contact your Salesforce sales rep for more information.
- Choose the Salesforce mobile app to brand (only one app can be branded per org).
- Start a new Mobile Publisher project.
- Design your branded assets and upload them along with your app information to Salesforce.
- Receive and thoroughly test the beta version of your branded app. For example, test the beta version to make sure your branding looks correct on devices with different screen sizes.
- Set up your preferred type of app distribution (Google or Apple).
- Approve your branded app and Salesforce submits it to Google and Apple. The store approval process can take from one day to two weeks depending on the app.
- See your branded app listed in Google Play and the App Store.

After your app is available to download from Google Play and the App Store, maintenance is a breeze. If your branding changes in the future, you can make an unlimited number of edits to your assets with

Mobile Publisher. And when Salesforce releases a new version of the mobile app, the updates to your branded app are seamless.

Requirements for Mobile Publisher for Salesforce App

Mobile Publisher for Salesforce App is supported on various mobile platforms, with minimum operating system requirements.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Platform Requirements

Mobile Publisher for Salesforce App is available on mobile devices that meet these mobile platform requirements.

Operating System and Version Requirements
Android 11.0 or later, Android WebView 90.0 or later
iOS 17.0 or later

Mobile Devices Used for Testing

Platform	**Phones**	**Tablets**
Android	• Pixel 10 • Pixel 9 / 9 Pro • Pixel 7 / 7 Pro • Samsung Galaxy S24 Ultra • Samsung Galaxy S23/23 Ultra • Samsung Galaxy S22/22 Ultra • Samsung Galaxy S21 • Samsung Galaxy S20 / S20+	• Samsung Galaxy Tab S6 • Samsung Galaxy Tab S7 • Samsung Galaxy Tab A (8 inch)

Platform	Phones	Tablets
iOS	• iPhone 17 Pro / Pro Max • iPhone 17 / iPhone 17 Air • iPhone 16 Pro / Pro Max • iPhone 16 / 16 Plus • iPhone 15 Pro / Pro Max • iPhone 15 / 15 Plus • iPhone 15 Pro / Pro Max • iPhone 14 Pro / Pro Max • iPhone 14 / 14 Plus • iPhone 13 Pro / Pro Max • iPhone 13 • iPhone 12 Pro / Pro Max • iPhone 12 • iPhone XR • iPhone 11 • iPhone SE	• iPad Pro (10.5 inch - 12.9 inch) • iPad Pro (9.7 inch, 6th generation and later) • iPad Air 3rd Gen • iPad Mini 5th Gen

Customers aren't blocked from using Mobile Publisher for Salesforce App on untested devices as long as they meet current platform requirements. Salesforce can't replicate some issues for customers using Mobile Publisher for Salesforce App on untested devices or due to manufacturer-specific customizations. The list of Salesforce-tested devices is subject to change, with or without advance notice.

Wireless Connection

A Wi-Fi® or cellular network connection is needed to communicate with Salesforce. For cellular connections, a 3G network or faster is required. For the best performance, we recommend using Wi-Fi or LTE.

Prepare Your App's Information

Preparing your branded assets is the most important part of this process. Let's discuss some of the branding details so you can get started with your Mobile Publisher project.

You can apply your company's branding to many of the elements in the Salesforce mobile app. Whether you choose to brand the Salesforce mobile app for Android or for iOS, we've got you covered on what you need to know.

Prepare Your App's Information for Android
Let's look at what elements you can apply your company's branding to in the Salesforce app for Android.

Prepare Your App's Information for iOS

Apply your company’s branding to these assets in the Salesforce app for iOS.

Prepare Your App’s Information for Android

Let’s look at what elements you can apply your company’s branding to in the Salesforce app for Android.

Here’s what you provide for your Mobile Publisher project to brand your own Salesforce app for Android:

Note All image assets are PNG format. In the following section, image-related specifications are provided for your convenience. Since specifications frequently change, consult Add preview assets to showcase your app in *Play Console Help* and Google Play icon design specifications in *Google Play Brand & marketing resources* for the latest image-formatting guidelines. See also Support different pixel densities in the *Android Developer Guide*.

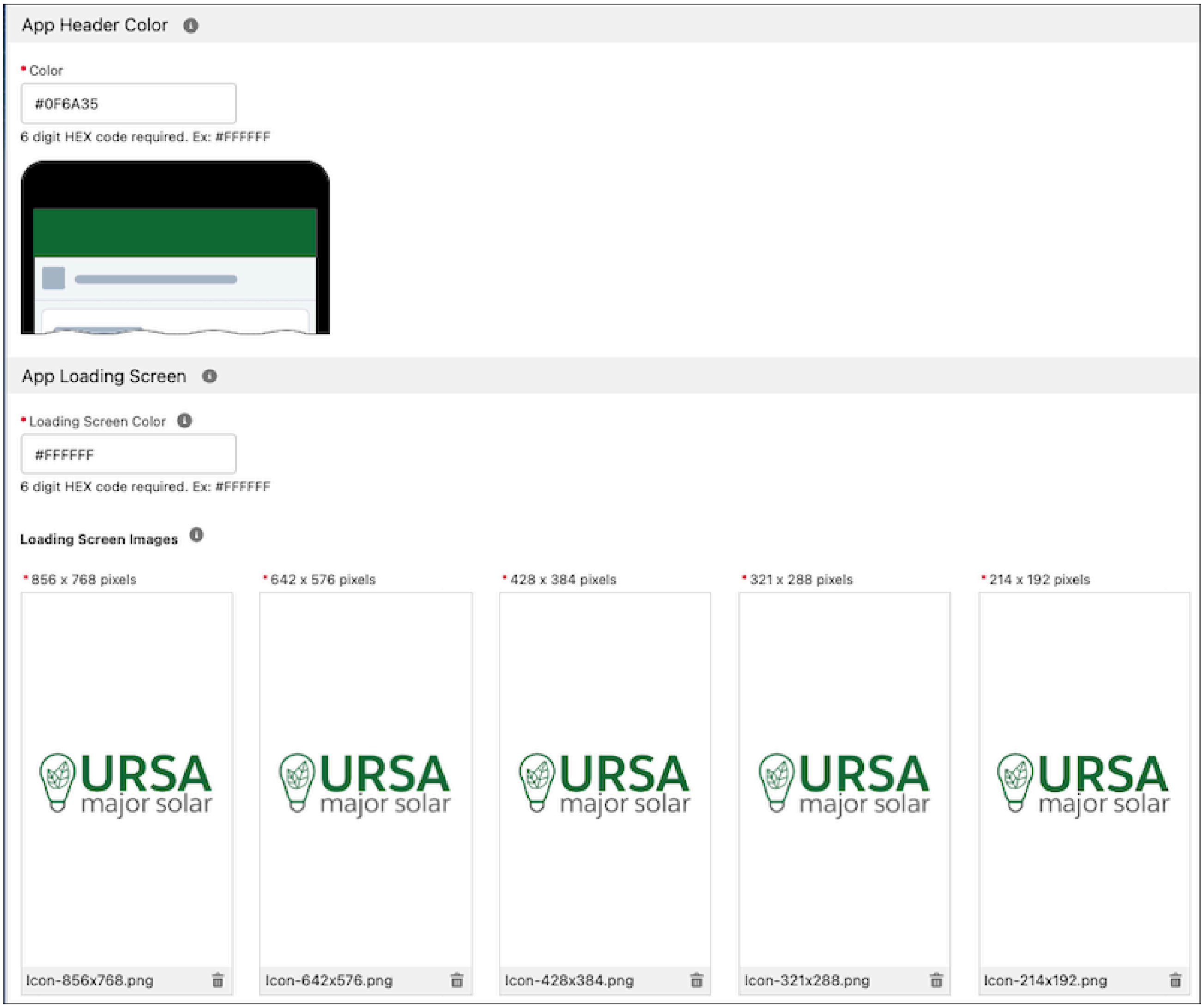
Field	Description
Product Details	
App Name	The name under the app icon on the device. This name is different from the Google Play Title. Max 12 characters.
App Help URL	A URL with help information for your app. Your users can access this link from the bottom of your app's Navigation Menu. If you leave this field blank, it defaults to the standard Salesforce Help URL.
Google Play Default Language	The default language to display for your listing on the Google Play Store.
Google Play Title	A specific name that isn’t too similar to existing app names on the Google Play Store. Max 30 characters.
Google Play Long Description	This description is for your listing on the Google Play Store. The best descriptions are concise, informative, and highlight the main features of your app. Max 4,000 characters.
Google Play Short Description	A quick description of your app on the Play Store app. The quick description expands to your app's full description. Max 80 characters.
Authorized Domains	
Domain URL	The default set of domains to log in to are production and sandbox. If you don’t add an extra

Field	Description
	domain, production is the default domain. The URL must begin with https://.
Domain Label	- English only - A-Z, a-z, 0–9 - Special characters: underscore (_), dash (-), period (.), and space ( ) - Max 20 characters
Beta Tester Information	
First Name	First name of beta tester.
Last Name	Family name of beta tester.
Email Address	Email address of beta tester. Your beta tester receives an email when the beta version of your app is ready to test. The beta tester needs a Gmail or Google Workspace account.
App Header Color	
Color	The background color for the header of the app. The Color format is 6 HEX digits. Must start with #. **See below for example*
App Loading Screen	
Loading Screen Color	The background color for the loading screen of the app. The Color format is 6 HEX digits. Must start with #. **See below for example*
Loading Screen Images: - 856 x 768 pixels - 642 x 576 pixels - 428 x 384 pixels - 321 x 288 pixels - 214 x 192 pixels	Image with the logo on the loading screen. 32-bit transparent PNG. **See below for example*
Icons	
Google Play Store Icon:	The Google Play Store icon shows up in your listing on the Google Play Store. Make your Google Play Store Icon the same design as your mobile device icons. Max 1,024 KB. 32-bit PNG with alpha

Field	Description
- 512 x 512 pixels	channel for transparency.
Mobile Device Icons: - 192 x 192 pixels - 144 x 144 pixels - 96 x 96 pixels - 72 x 72 pixels - 48 x 48 pixels - 36 x 36 pixels	The mobile device icons show up on the mobile device itself. Make your mobile device icons the same design as your Google Play Store Icon.
Mobile Device Icons - Round: - 192 x 192 pixels - 144 x 144 pixels - 96 x 96 pixels - 72 x 72 pixels - 48 x 48 pixels - 36 x 36 pixels	For devices running Android 7.1 or later. If you don't provide these icons, devices running Android 7.1 or later display the icons you've uploaded in the "Mobile Device Icons" section.
Push Notification Icons: - 48 x 48 pixels - 36 x 36 pixels - 24 x 24 pixels	Push notification icons display in several areas on an Android mobile device. Notifications provide short, timely, and relevant information about your app when it’s not in use, and the icon you upload is visible next to those notifications.
Salesforce External Client App	
Name	The name displays when your users log in to your app for the first time. If possible, make the name the same as your App Store name. 5–40 characters. English only.
Icon: - 128 x 128 pixels	The external client app icon shows up when your users log in to your app for the first time. Make this icon the same design as your app icons. PNG format and max 100 KB.
Google Play Screenshots	

Field	Description
Screenshot	Screenshots show up on your Google Play Store listing. The order of screenshots is the same in the store listing. 24-bit PNG (no alpha). 2–8 screenshots. Minimum 320 pixels. Maximum 3,840 pixels. The maximum dimension of your screenshot can't be more than twice as long as the minimum dimension. If you don't have access to your branded app to take screenshots, use the Publisher Playground app. Avoid using a mobile browser for screenshots, as it can lead to your app being rejected.
Google Play Feature Graphic	
Feature Graphic: • 1024 x 500 pixels	To be featured anywhere in Google Play, a feature graphic is required. The feature graphic shows up at the top of your store listing in the Play Store app. 24-bit PNG (no alpha).
Google Play Promo Video	
Promo Video URL	The promo video lets users know about current features of your app. • Use an individual video's YouTube URL, and not a YouTube playlist or channel URL • Don't use an age-restricted video • Use the full YouTube URL instead of a shortened URL. For example, use https://www.youtube.com/watch?v=yourvideoid instead of https://youtu.be/yourvideoid • For full accessibility, add closed captions and descriptive audio to your video
Country Availability	
Country Availability	Choose the countries where you want your app to be available. Default is all countries.

**Example*



App Header Color

* Color

#0F6A35

6 digit HEX code required. Ex: #FFFFFF

App Loading Screen

* Loading Screen Color

#FFFFFF

6 digit HEX code required. Ex: #FFFFFF

Loading Screen Images

* 856 x 768 pixels	* 642 x 576 pixels	* 428 x 384 pixels	* 321 x 288 pixels	* 214 x 192 pixels
Icon-856x768.png	Icon-642x576.png	Icon-428x384.png	Icon-321x288.png	Icon-214x192.png

Prepare Your App’s Information for iOS

Apply your company’s branding to these assets in the Salesforce app for iOS.

To brand your Salesforce app for iOS, provide these elements for your Mobile Publisher project.

- Image assets in PNG format.
- Screenshots captured by the Publisher Playground App. Don’t use a mobile browser for screenshots because it can lead to your app being rejected by Apple.
- Images and screenshots can’t include Alpha channels or transparencies. To remove the Alpha channel on a Mac device, use the Preview app to duplicate the image, deselect the Alpha option, and save the image.

Field	Description
Product Details	

Field	Description
App Name	The name under the app icon on the device. This name is different from the App Store name. Max 12 characters.
App Help URL	A URL with help information for your app. Your users can access this link from the bottom of your app's Navigation Menu. If you leave this field blank, it defaults to the standard Salesforce Help URL.
App Store Default Language	The default language to display for your listing on the App Store.
App Store Name	The App Store name must include the app name. Choose a name that isn’t too similar to existing app names. Max 30 characters.
App Store Description	The description for your app's listing on the App Store. The best descriptions are concise, informative, and highlight the main features of your app. Max 4,000 characters. If your app uses the Fully Managed distribution method, emojis aren’t supported in the App Store description.
App Store Keywords	Keywords to make your app are easy to find in search results. Max 100 characters total. Separate each keyword with a comma.
App Store Subtitle	A one-line summary of your app that displays under your app name throughout the App Store in iOS 11 and later. Max 30 characters.
App Store Promotional Text	Informs your users of current features. The text appears at the top of the app description in the App Store. Max 170 characters. If your app uses the Fully Managed distribution method, emojis aren’t supported in the App Store promotional text.
App Store Support URL	A URL with support information for your app. It appears on the App Store listing. If you leave this field blank, it defaults to the standard Salesforce Support URL.

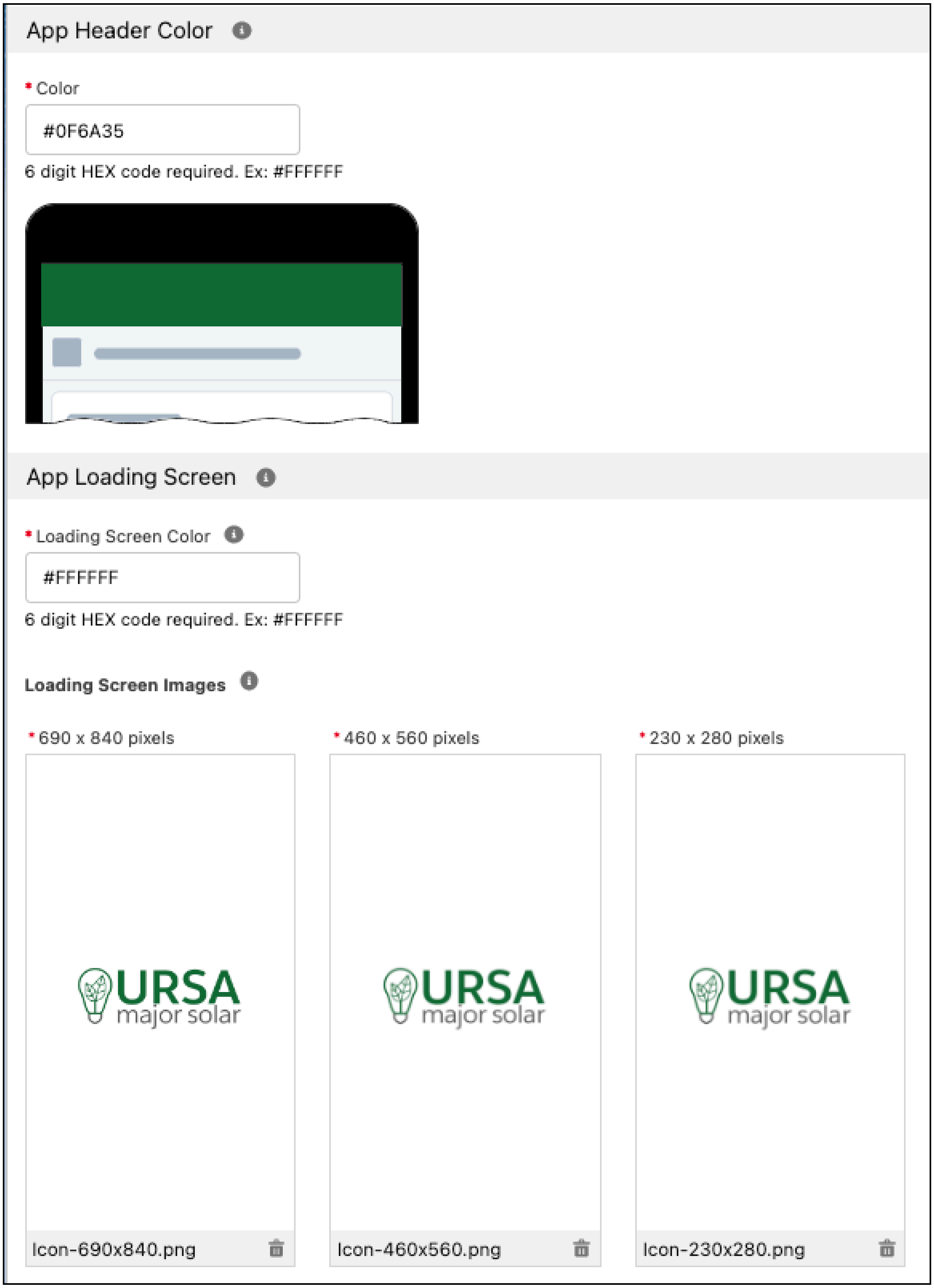
Field	Description
App Store Marketing URL	A URL with marketing information for your app. It appears on the App Store listing. If you leave this field blank, it defaults to the standard Salesforce Marketing URL.
Authorized Domains	
Domain URL	The default domains to log in to are production and sandbox. If you don't add an extra domain, production is the default domain. URL must begin with https://.
Domain Label	• English only • A-Z, a-z, 0–9 • Special characters: underscore (_), dash (-), period (.), and space ( ) • Max 20 characters
Beta Tester Information	
First Name	First name of beta tester.
Last Name	Family name of beta tester.
Email Address	Email address of beta tester. Your beta tester receives an email when the beta version of your app is ready to test.
App Header Color	
Color	The background color for the header of the app. Color format is 6 HEX digits. Must start with #. See the Examples section for images.
App Loading Screen	
Loading Screen Color	The background color for the loading screen of the app. Color format is 6 HEX digits. Must start with #. See the Examples section for images.
Loading Screen Images: • 690 x 840 pixels • 460 x 560 pixels • 230 x 280 pixels	Image with the logo on the loading screen. 32-bit transparent PNG. See the Examples section for images.

Field	Description
Icons	
App Store Icon: • 1024 x 1024 pixels	The App Store icon appears in your app's listing on the App Store. Use the same design as your mobile device icons.
Mobile Device Icons: • 180 x 180 pixels • 167 x 167 pixels • 152 x 152 pixels • 120 x 120 pixels • 87 x 87 pixels • 80 x 80 pixels • 76 x 76 pixels • 60 x 60 pixels • 58 x 58 pixels • 40 x 40 pixels • 29 x 29 pixels • 20 x 20 pixels	The mobile device icons appear on the mobile device itself. Use the same design as your App Store icon. • Flattened PNG format • Square icon corners • No transparency or alpha channel • Minimum 72 DPI resolution • RGB color space
Salesforce External Client App	
Name	The name appears when your users log in to your app for the first time. If possible, use the same name as your App Store name. 5–40 characters. English only.
Icon: • 128 x 128 pixels	The external client app icon appears when your users log in to your app for the first time. Use the same design as your app icons. The corners of the icon aren't rounded automatically. Apply rounded corners to the icon manually. PNG format and max 100 KB.
App Store Screenshots	
iPhone 6.7" Screenshots: • 1290 x 2796 pixels	• Use screenshots of your app's user experience. Don't use screenshots of your logo or loading screen. • Upload screenshots as they are, with no modifications. The rounded corners are

Field	Description
	automatically applied after the screenshots are upload. - If you add text descriptions to your screenshots, keep text away from the corners to avoid cropping. - Leave the home indicator in the screenshot, if it's visible. - Avoid placing screenshots within a hardware device image. - The order of screenshots in the App Store listing reflects the order of the screenshots in the Mobile Publisher iOS Setup page. Requirements: - 6.7 inch (17.01 cm) display screenshots - Flattened PNG format - No transparency or alpha channel - Minimum 72 DPI resolution - RGB color space - Max five screenshots Refer to Submit your iOS apps to the App Store.
iPhone 6.5” Screenshots: - 1242 x 2688 pixels	- Use screenshots of your app's user experience. Don't use screenshots of your logo or loading screen. - Upload screenshots as they are, with no modifications. The rounded corners are automatically applied after the screenshots are upload. - If you add text descriptions to your screenshots, keep text away from the corners to avoid cropping. - Leave the home indicator in the screenshot, if it's visible. - Avoid placing screenshots within a hardware device image. - The order of screenshots in the App Store listing reflects the order of the screenshots in the Mobile Publisher iOS Setup page.

Field	Description
	Requirements: - 6.5 inch (16.5 cm) display screenshots - Flattened PNG format - No transparency or alpha channel - Minimum 72 DPI resolution - RGB color space - Max five screenshots Refer to Submit your iOS apps to the App Store.
iPhone 5.5” Screenshots: - 1242 x 2208 pixels	Requirements: - 5.5 inch (13.97 cm) display screenshots - Flattened PNG format - No transparency or alpha channel - Minimum 72 DPI - RGB color space - Max five screenshots
iPad Screenshots: - Portrait: 2048 x 2732 pixels - Landscape: 2732 x 2048 pixels	One size screenshot for all versions of the iPad. Requirements: - 12.9 inch (32.77 cm) display screenshots - Flattened PNG format - No transparency or alpha channel - Minimum 72 DPI - RGB color space - Max five screenshots
Country Availability	
Country Availability	Select the countries where you want your app to be available. Default is all countries.

Examples



App Header Color

* Color

#0F6A35

6 digit HEX code required. Ex: #FFFFFF

App Loading Screen

* Loading Screen Color

#FFFFFF

6 digit HEX code required. Ex: #FFFFFF

Loading Screen Images

* 690 x 840 pixels

Icon-690x840.png

* 460 x 560 pixels

Icon-460x560.png

* 230 x 280 pixels

Icon-230x280.png

App Distribution

Choose how you want to get your app into the hands of your users.

Deciding on how you want to manage the release and distribution of your app is an important step. Mobile Publisher offers App Distribution Methods and App Distribution Types to help your org manage the release and distribution of your app, your way.

Watch the video: https://play.vidyard.com/ChnGsudDBC75AWfBd87MkA

Your apps can be distributed to end users either publicly or privately via different channels per platform according to Apple and Google guidelines.

As an app developer, Salesforce complies with Apple and Google distribution guidelines.

- App Distribution Methods:
  - Binary Handoff: Salesforce builds the app binary and provides it to you. You handle the resigning process of the binary, submitting the store listing information, and managing the release of the app.
  - Binary Upload: Salesforce builds and uploads the app to your store account. You submit the store listing information (screenshots, description, and so on) and manage the release of the app.
  - Fully Managed: Salesforce builds, releases, and manages the lifecycle of the app.
- App Distribution Types:
  - Public Distribution: Your branded app is published via your developer account to the public Apple App Store or Google Play.
  - Private Distribution: You manage distribution of your branded app via your developer account privately. Your branded app is only accessible to your end users privately, and isn't publicly available on the Apple App Store or Google Play.

App Distribution Methods

Choose whether you want Salesforce to manage the release of your branded app, or if you want to manage the release of the app on your own.

App Distribution Type Considerations

Choose the right distribution type based on your use case and Apple and Google guidelines. As an app developer, Salesforce complies with Apple and Google distribution guidelines, and violating the guidelines can cause Apple or Google to reject your branded app.

Set Up Distribution and External Client Apps

Your branded app is published via your developer account either publicly or privately.

Set Up a New External Client App (ECA) in Mobile Publisher

Start setting up an external client app (ECA) for your mobile app after completing the Set Up the Mobile App step in Mobile Publisher. Don't create your external client app in the External Client App manager (accessed through Setup). Create your ECA in Mobile Publisher instead.

Change Distribution Types

Consider the impact of changing the distribution type for your branded app.

Change Distribution Account

You may have to change your distribution account if you need to change the developer account associated with your distribution type.

App Distribution Methods

Choose whether you want Salesforce to manage the release of your branded app, or if you want to manage the release of the app on your own.

Choose your App Distribution Method before choosing your App Distribution Type.

Distribution Feature	Fully Managed	Binary Upload	Binary Handoff
App Build	Salesforce builds the app binary.	Salesforce builds the app binary.	Salesforce builds the app binary (.ipa or .aab) and delivers it to you.
Code Signing	Salesforce handles all code signing.	Salesforce handles all code signing.	**You have full control.** Your organization re-signs the app binary with your own distribution certificates and private keys.
App Store Listing & Metadata	Salesforce manages all store listing information (screenshots, etc.).	Salesforce manages all store listing information (descriptions, etc.).	**You have full control.** You submit and manage all store listing information.
Upload to App Stores	Salesforce uploads the signed app binary to the store.	Salesforce uploads the signed app binary to your store account.	**You have full control.** Your organization is responsible for uploading the re-signed binary to the app stores.
Submission for Review	Salesforce submits the app for final review and handles the release lifecycle.	**You have full control.** You're responsible for submitting the app for review.	**You have full control.** You're responsible for submitting the app for review.
Release Management	Salesforce manages the release of the app to your users after Apple/Google has approved it.	**You have full control.** This allows you to use advanced app store features like choosing your own release schedule and performing staged rollouts.	You have maximum control over the entire submission and release process.
Best For...	Organizations that are	Organizations that are	Organizations with

	not familiar with the app stores and prefer Salesforce to handle the entire process.	familiar with the app stores and want flexibility and control over their app's store listing, release schedule, and testing of new versions..	specific security protocols that mandate in-house code signing or established internal processes that require handling the final app submission themselves.

Binary Handoff Distribution Method

If you choose Binary Handoff, Salesforce builds the application binary file (.ipa for iOS or .aab for Android) and delivers it directly to you. Your organization is then responsible for re-signing the binary with your own signing certificates and submitting the app to the Apple App Store and/or Google Play Store. This method provides the maximum level of control over the final app signing and submission process. It’s ideal for orgs with specific security protocols or established internal processes for distributing mobile applications that require them to handle the final signing and upload to the app stores themselves.

Binary Upload Distribution Method

If you choose Binary Upload, you’re responsible for submitting App Store listing information, submitting the app for review, and releasing the app to your users. After you invite Salesforce to your App Store’s accounts and grant access to the credentials, Salesforce uploads the app to your account. Salesforce then manages your app branding assets and app updates.

This distribution method is ideal for orgs who already have apps in the app stores and are familiar with how the app stores work. You can take advantage of App Store features like staged releases.

Fully Managed Distribution Method

If you choose Fully Managed, Salesforce builds and uploads your app to your App Store’s accounts after you grant access to the credentials. Salesforce then manages your App Store listing information, app branding assets, app updates, and releasing the app to your users.

This distribution method is ideal for orgs that aren’t familiar with the app stores and its complexities.

Advantages of Binary Upload Distribution Method and Binary Handoff Distribution Methods
If your org is familiar with how the app stores work, the Binary Upload Distribution Method has many advantages for managing the release of your app on your end.

Advantages of Binary Upload Distribution Method and Binary Handoff Distribution Methods

If your org is familiar with how the app stores work, the Binary Upload Distribution Method has many advantages for managing the release of your app on your end.

The Binary Upload Distribution Method gives you the flexibility to take advantage of features offered by the app stores.

iOS

- Choose your own release schedule that aligns with your practices and processes.
- Localization support for store metadata.
- Staged release rollout for your app.
- Quick access to the latest improvements and features on the App Store.
- Upload up to 10 screenshots for your App Store listing.
- Publish your app in the correct app category in the App Store.
- Distribute your app on Macs with Apple Silicon. See iPhone and iPad Apps on Apple Silicon Macs (Apple) for more information.
- Manage the last compatible version of your app. You can make the last compatible of your app available to users who don't have a device that's compatible with the latest version of your app.
- Unpublish your app at your own discretion.

Android

- Choose your own release schedule that aligns with your practices and processes.
- Localization support for store metadata.
- Pre-launch your app to identify any potential issues such as stability, performance, accessibility, and security. See Pre-launch reports (Google) for more information.
- Staged release rollout for your app.
- Do a pre-registration of the app for specific user base. See Pre-registration (Google) for more information.
- Perform different types of testing (open vs closed).
- Quick access to the latest improvements and features on the Play Store.
- Upload up to 8 screenshots for your Play Store listing.
- Unpublish your app at your own discretion.
- Manage when certain updates (not in the binary) go live in your app. See Publishing overview (Google) for more information.
- Target your app for specific mobile carrier (AT&T, Verizon, and so on).
- Take advantage of App Actions on using Actions on Google. See App Actions overview (Google) for more information.
- Exclude specific devices that you don't want your app to support.
- Display links to your app in Google searches. See App Indexing on Google Search (Google) for more information.
- Create custom store listings to appeal to a specific audience. See Create custom store listings to target specific countries (Google) for more information.

- Run A/B tests on your store listing. See Store listing experiments (Google) for more information.

Binary Handoff

The Binary Handoff distribution method offers an even greater level of control than Binary Upload. In addition to the advantages listed above, Binary Handoff gives your organization direct control over the app's signing identity. This is a critical requirement for companies whose security policies mandate that all application code signing is performed in-house using their own private keys. By re-signing the app yourself, you maintain end-to-end control over the cryptographic integrity and submission of your mobile app.

App Distribution Type Considerations

Choose the right distribution type based on your use case and Apple and Google guidelines. As an app developer, Salesforce complies with Apple and Google distribution guidelines, and violating the guidelines can cause Apple or Google to reject your branded app.

Note You receive a beta app for testing, whether you choose private or public distribution. For Android private distribution, customers are responsible for distributing pre-release versions of the app to trusted test users because Salesforce doesn’t provide a beta app.

App Store Distribution Guidelines

- You can distribute Business to Business (B2B) and Business to Employee (B2E) only with Apple Business Manager (ABM).
- You need an Apple Business Manager (ABM) ID for each country that you want to privately distribute your app in. For example, if you want to distribute your app in India, you must have a valid Apple Business Manager (ABM) ID in that country.
- You can distribute Business to Consumer (B2C) only on the public App Store.
- To ensure that there are no delays or surprises in the end-user experience when it’s live, read the distribution offerings. You can’t use the same app name for any app that's live that you want to change the distribution type for. For example, if you have an app with a private distribution type, you can’t use the same app name if you change the distribution type to public. For more information, see the App Store guidelines.

Google Play Store Distribution Guidelines

- You can distribute B2C, B2B, and B2E apps to various distribution types based on the customer case.
- You can change distribution channels from private to public or from public to private for existing apps.
- If you have a private distribution, always use a G-Suite with your organization domain. Access to the app is available only to users who are part of that G-Suite organization. For example, if your organization’s name is AYZ Inc., use a G-Suite with the same domain and app users who are part of the organization. For more information, your Google Play Store admin can review Google’s Mobile Device Management (MDM) and Enterprise Mobility Management (EMM) documentation, or contact your EMM provider.

- There's no beta app provided for Android private distribution. If you don't use G-Suite and Google for work, refer to Distribute private apps.

  Important For Android private distribution, the beta status is misleading and not related to providing a beta app to testers. To make the app available for testing, the app status must be in live status.

Customer-Managed Distribution Type Attributes

Operating System	Private Distribution	Public Distribution
Android	• Customer Developer Account: Required • App Distribution: Customer-Managed Google Play for Work (customer is responsible for beta testing) • Salesforce Required Access to Customer Developer Account: Yes	• Customer Developer Account: Required • App Distribution: Customer Public Google Play account • Salesforce Require Access to Customer Developer Account: Yes
iOS	• Customer Developer Account: Required • App availability: Customer-Managed Private on App Store account • Salesforce Required Access to Customer Developer Account: Yes	• Customer Developer Account: Required • App availability: Customer Public on App Store account • Salesforce Required Access to Customer Developer Account: Yes

Set Up Distribution and External Client Apps

Your branded app is published via your developer account either publicly or privately.

Set Up Public Distribution
Set up your branded app to be published through your developer account to the public Apple App Store or Google Play.

Set Up Private Distribution
You manage distribution of your branded app through your developer account privately. Your branded app is only accessible to your end users privately, and isn't publicly available on the Apple App Store or Google Play.

Unlisted Option for iOS App Distribution
Learn about unlisted app distribution, an alternative to the traditional public and private app

distribution options. Unlisted app distribution is sometimes appropriate for apps that were rejected from App Store public distribution.

Set Up Public Distribution

Set up your branded app to be published through your developer account to the public Apple App Store or Google Play.

Set Up Public Distribution for Android
Let’s get your branded app ready for distribution on the public Google Play store.

Set Up Public Distribution for iOS
Let’s get your branded app ready for distribution on the public App Store.

Set Up Public Distribution for Android

Let’s get your branded app ready for distribution on the public Google Play store.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Publisher needs certain pieces of information from your Google Play Developer account to distribute your branded app.

Note If you haven’t signed up for a Google Play Developer account, click here to sign up.

1. **Choose Distribution Type**
   Start the public distribution process by selecting your distribution type.
2. **Create a Placeholder App**
   Create a placeholder app in your new Google Play Console account.
3. **Mobile Publisher Best Practices for App Content Declarations**
   After you create a placeholder app, you complete Google Play’s required App Content declarations to publish your Android app. When you publish a Mobile Publisher Android app, follow best practices specific to Mobile Publisher for the Privacy policy, App access, Data safety, and Financial features declarations.
4. **Update Test User Credentials**
   Google reviews apps by logging in as a user to check the functionality. Update the test user credentials

in your Google Play Console account to avoid any app rejections.

5. **Create Service Account User and Find Your Google Play Developer ID**
   To manage the distribution of your branded app with your developer account, Mobile Publisher needs your Service Account User and Google Play Developer ID info.
6. **Grant Permission to Distribute Your Branded App**
   Grant Salesforce permission to distribute your branded app.
7. **Register Your Firebase App to Enable Mobile Publisher Android Push Notifications**
   To set up push notifications for your Android app, provide Mobile Publisher with information from a Google Firebase project associated with your app. To get the required information, register your Android app and generate a private key. Then, submit the information to your app's Setup for Mobile Publisher project.
8. **Enter Your Information into Your Mobile Publisher Project for Android**
   Now that we have all the information we need, we can go back to the Mobile Publisher and fill out the necessary fields.

Choose Distribution Type

Start the public distribution process by selecting your distribution type.

REQUIRED EDITIONS

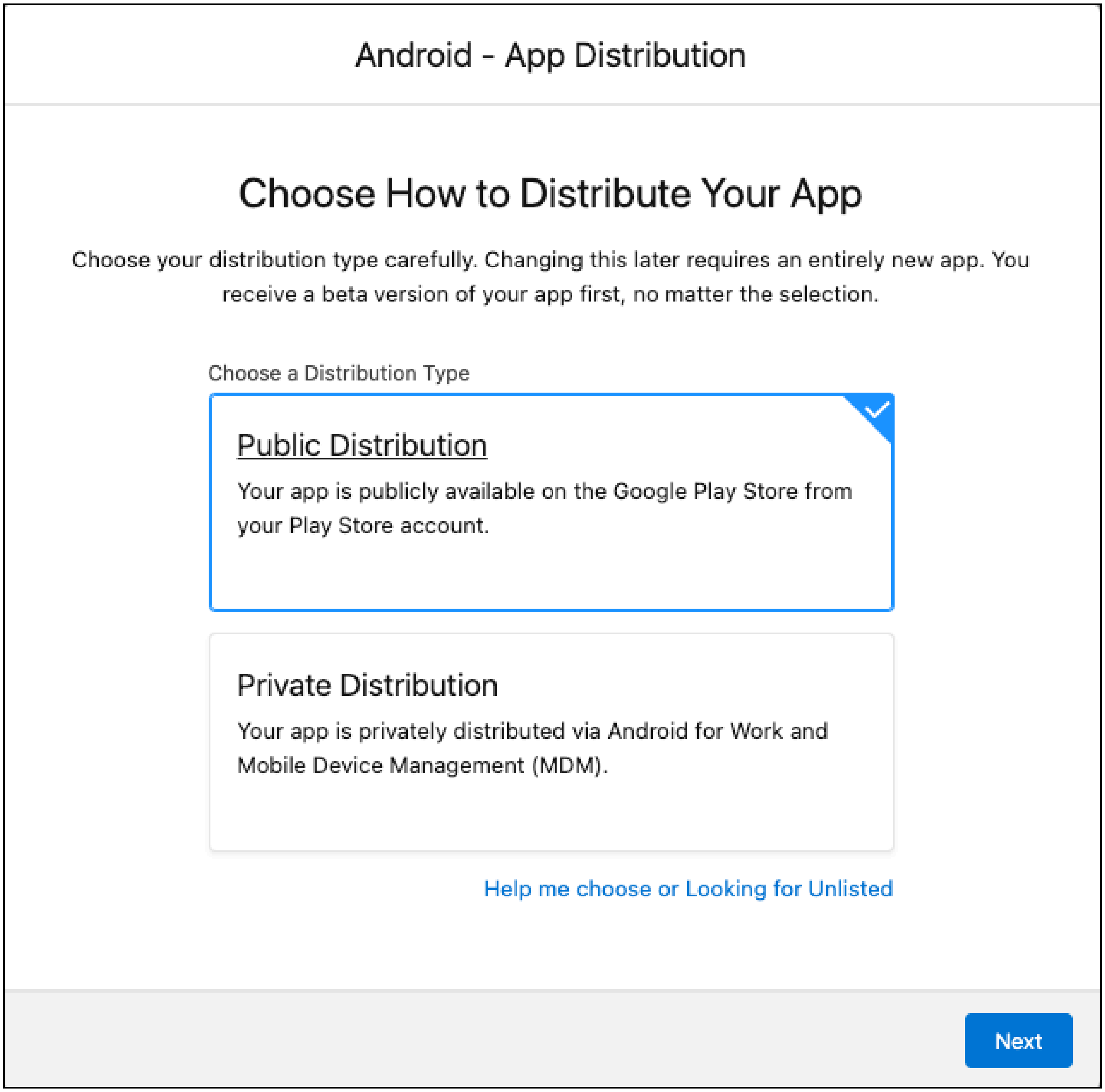
Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Watch the video: https://play.vidyard.com/ChnGsudDBC75AWfBd87MkA

First, we must choose the correct distribution type for your branded app.

1. Choose **Public Distribution** from the App Distribution menu and select **Next**.



Android - App Distribution

Choose How to Distribute Your App

Choose your distribution type carefully. Changing this later requires an entirely new app. You receive a beta version of your app first, no matter the selection.

Choose a Distribution Type

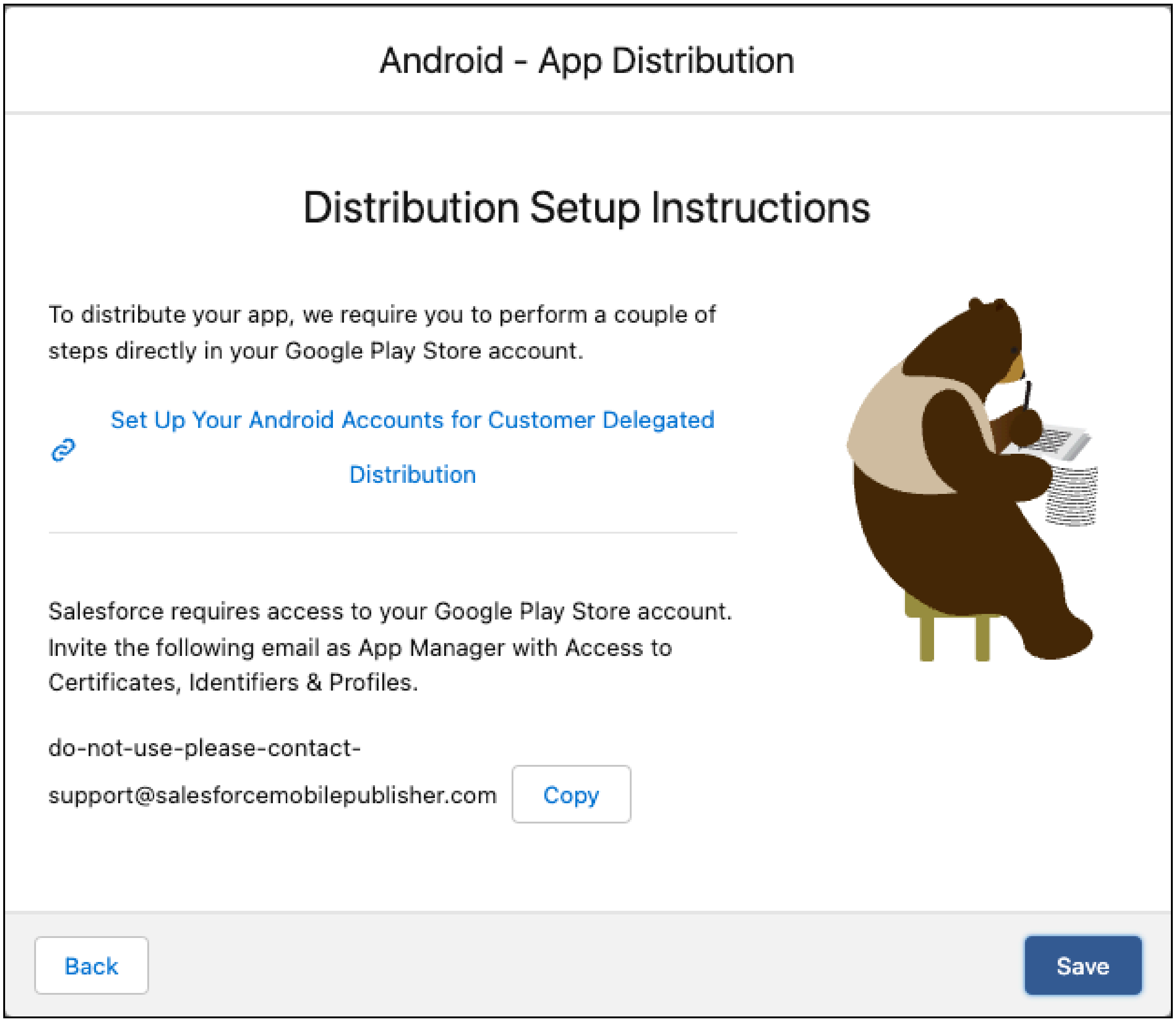
☑ Public Distribution

Your app is publicly available on the Google Play Store from your Play Store account.

Private Distribution

Your app is privately distributed via Android for Work and Mobile Device Management (MDM).

2. Select **Copy** to copy the Salesforce Developer Email to your clipboard. You'll need this email address when you log into your Google Play Developer account.



Android - App Distribution

Distribution Setup Instructions

To distribute your app, we require you to perform a couple of steps directly in your Google Play Store account.

Set Up Your Android Accounts for Customer Delegated Distribution

Salesforce requires access to your Google Play Store account. Invite the following email as App Manager with Access to Certificates, Identifiers & Profiles.

do-not-use-please-contact-support@salesforcemobilepublisher.com

Create a Placeholder App

Create a placeholder app in your new Google Play Console account.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Before you begin, make sure that you have a verified Google Play developer account for an organization.

Note Mobile Publisher can't submit your app for review until your organization is successfully verified by Google. Make sure to follow Google's requirements for verifying your organization. If your app requires advertiser verification, see Document requirements for advertiser verification in *Google Play Console Help*.

Watch the video: https://play.vidyard.com/GaqayoPk3sSbZ5LUjpNQYh

1. Log in to your Google Play Console with your Google Play Developer account.
2. Select **Create App**.
3. Enter the App Name, which must match the name entered in the **Google Play Title** field in the metadata form.
4. In the App or Game category, select **App**.
5. In the Free or Paid category, select **Free**.
6. Review and accept all the required Google Developer Declaration policies, and then select **Create App**.

Mobile Publisher Best Practices for App Content Declarations

After you create a placeholder app, you complete Google Play's required App Content declarations to publish your Android app. When you publish a Mobile Publisher Android app, follow best practices specific to Mobile Publisher for the Privacy policy, App access, Data safety, and Financial features declarations.

Watch the video: https://play.vidyard.com/GaqayoPk3sSbZ5LUjpNQYh

To find your app's declarations in the Google Play console, scroll to the Policy and programs section in the left navigation pane. Then, click **App content**. Complete the declarations on the **Needs attention** tab.

Declaration: Privacy policy

Enter information about your app's privacy policy. Make sure to link to your app's privacy policy. Don't enter the link to the Salesforce privacy policy.

Note Google requires that your app's privacy policy contains specific information about your app and is linked from your app. See Set Up Your Android App for Google's Privacy Policy Requirements.

Declaration: App access

Enter information about how Google reviewers can access your Mobile Publisher app, including your app's log-in instructions and details. See Update Test User Credentials.

Declaration: Data safety

Enter information about how your app collects data. To learn about the baseline disclosures for your

Mobile Publisher Android app, see the Google Data Safety reference for Mobile Publisher apps.

Note If your app doesn't collect any data from your users, select **Device or other IDs** in the Data safety declaration, and use the Google Data Safety reference for Mobile Publisher apps to answer the questions.

For instructions on filling out the Data safety form, see Provide information for Google Play's Data safety section in the *Google Play Console Help*.

Declaration: Financial features

If you created your Mobile Publisher Android app before August 2023, make sure to return to the Google Play console and complete your app's Financial features declaration. See Complete and submit the declaration in the *Google Play Console Help*.

Important To publish your Android app on Google Play, make sure to also complete the declarations for Ads, Content rating, Target audience, News apps, COVID-19 contact tracing and status apps, and Government apps. See the Google Play Console Help.

Update Test User Credentials

Google reviews apps by logging in as a user to check the functionality. Update the test user credentials in your Google Play Console account to avoid any app rejections.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Log in to your Google Play Console with your Google Play Developer account before completing these steps.

Watch the video: https://play.vidyard.com/GaqayoPk3sSbZ5LUjpNQYh

1. Open the Google Play Console, and go to the App content page.

   Note If you've previously added instructions for app access and you want to make changes, select **Manage** instead of **Start**.

2. Under **App access**, select **Start**.

3. Select the **All or some functionality is restricted** button.
4. Click **+ Add new instructions** and provide your access details such as the username and password.
5. In the **Any other instructions** field let Google know if there's anything special or unique about your login mechanisms.

   Note If you want Google to review your sandbox site, and you've entered your sandbox URL in Setup for Mobile Publisher, provide the Google reviewers with instructions on how to switch your app to the sandbox testing mode.

6. Enter the word *`Production`* or *`Sandbox`* for the Experience Cloud site you're providing credentials for.
7. If there are any other pages in your Experience Cloud site where login is required, provide the navigation path to it.
8. If applicable, include information such as one-time password, multi-factor authentication, or logins with more than two fields.

   Note To avoid an app rejection due to sign-in issues, we recommend that you disable two-factor authentication and single sign-on (SSO) for the test user. We also recommend that you specify a range of IP addresses for the test user. To specify the IP address range, clone the Salesforce user profile for the Google Reviewer test user, and then add the login IP range from 1.1.1.1 to 255.255.255.255 for that profile. Defining the IP range limits the change to only this user without affecting other users. See Restrict Login IP Ranges in Profiles for more information on adding a range of IP addresses.

9. Save your changes.

Create Service Account User and Find Your Google Play Developer ID

To manage the distribution of your branded app with your developer account, Mobile Publisher needs your Service Account User and Google Play Developer ID info.

Complete these steps from your Google Play Console.

Important You must be the Account Owner of your Google Cloud Platform account to perform the following steps.

Watch the video: https://play.vidyard.com/v63U4rHBcvRUGtY7s4Hmon

1. Log in to your **Google Cloud Console** (https://console.cloud.google.com/) using the same email associated with your **Google Play Console** account.
2. In the search bar, search for **Create a Project**, fill in the **Project name**, and click **Create**.

   Note You can skip creating a new project if you want to reuse an existing project from your Google Cloud Console.

3. Search for **APIs and Services** in the search box, choose **Library** from the left-hand navigation bar and search for **Google Play Android Developer API**.
4. Select **Google Play Android Developer API** and click **Enable**.
5. After it's enabled, from the home page, search for **Service Accounts** in the search bar.
6. At the top of the page, click **+ Create Service Account**.

7. Enter the Service Account Name. This field automatically populates the Service Account ID.
8. Click **Create and Continue**.
9. In the **Grant this service account access to project** section, select the role as **Service Accounts | Service Account User**, then click **Continue**.
10. Leave the optional **Grant user access to this service account** section blank and click **Done**.
11. Now that a new service account has been created, click the three dots under **Actions** and select **Manage Keys**.
12. Click **Add Key | Create New Key** then select **JSON** and choose **Create**. This action saves the JSON onto your computer (you must upload this file later to the Mobile Publisher Setup).
13. From the left panel navigation menu, click **Service Accounts**, then click the newly created service account.
14. Copy the email address of the Service Account User you created to your clipboard. The email appears under the **Email** field under the **Details** tab. The email address must end in .iam.gserviceaccount.com.

Now go back to the Google Play Console, and from the left panel navigation menu, click **Account Details** and copy the 19-Digit ID from the **Developer Account ID** field. Alternatively, you can also find your Google Play Developer ID in the URL of your browser address bar.

- In your browser address bar, look for `https://play.google.com/console/u/0/developers/<19 Digit ID>/app-list`.
- Copy the set of 19 digits (your Google Play Developer Account ID) after `developers/` to your clipboard.

Note The private key that you download from the Google Play Console is a different JSON file than the Firebase admin SDK private key that you download when you set up push notifications.

Grant Permission to Distribute Your Branded App

Grant Salesforce permission to distribute your branded app.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Complete these steps after logging into the Google Play Console with your Google Play Developer account.

Watch the video: https://play.vidyard.com/v63U4rHBcvRUGtY7s4Hmon

Invite the Salesforce Developer ID

1. Navigate to **Users and Permissions** and click **Invite New Users**.
2. Enter the Salesforce Developer Email you copied from your clipboard and don’t set an expiration date.

   Note Salesforce provides the Salesforce Developer Email address when you select an app distribution type. To view the email address, in Salesforce, go to **Setup | Mobile Publisher** and open your Android draft. In the App Distribution section, click the **Choose App Distribution Type** link. Select **Public** or **Private** (you can change this selection before you submit the draft) and click **Next**. The App Distribution dialog displays the email address beside the Copy button.

3. Under **App Permissions**, click **Add App** and select the placeholder app you created earlier, then click **Apply**.
4. In the pop-up window, select the following permissions:
   a. View app information (read-only)
   b. View financial data
   c. Manage orders and subscriptions
   d. Edit and delete draft apps
   e. Release to production, exclude devices, and use Play App Signing
   f. Release apps to testing tracks
   g. Manage testing tracks and edit tester lists
   h. Manage store presence
   i. Reply to reviews
5. Click **Invite User**.

Invite the Salesforce Service Account

1. Navigate to **Users and Permissions** and click **Invite New Users**.
2. Enter the **Service Account User** you created from Find Your Google Play Developer and Service Account Key Info and don’t set an expiration date.
3. Under **App Permissions**, click **Add App** and select the placeholder app you created earlier, then click **Apply**.
4. In the pop-up window, select the following permissions:
   - View app information (read-only)
   - View financial data
   - Manage orders and subscriptions
   - Edit and delete draft apps
   - Release to production, exclude devices, and use Play App Signing
   - Release apps to testing tracks
   - Manage testing tracks and edit tester lists
   - Manage store presence
   - Reply to reviews
5. Click **Invite User**.

Register Your Firebase App to Enable Mobile Publisher Android Push Notifications

To set up push notifications for your Android app, provide Mobile Publisher with information from a Google Firebase project associated with your app. To get the required information, register your Android app and generate a private key. Then, submit the information to your app's Setup for Mobile Publisher project.

Watch the video: https://play.vidyard.com/MdJFfAC6Zw31ncGjPJ6Rjf

Note If you submitted your Mobile Publisher Android app before November 2023, don't create a new Google Firebase project. Confirm that you've completed steps 4 through 6 with your existing Firebase project. You must complete those steps to receive a new Android app build.

1. Log in to the Google Firebase Console with the Google Play account associated with your Mobile Publisher app.
2. Create a Firebase project.
   a. In the Google Firebase Console, click **Create a project**.
   b. Click **Add Firebase to Google Cloud project**.
   c. In the **Project name** field, select the Google Cloud project where you previously created a service account user.
   d. Follow the prompts to create a project.
3. Assign the project's Firebase admin role to your Salesforce Developer ID.
   a. In a separate browser window or tab, open the Google Play Console, and then click **Users and permissions**. In the **Users** tab, copy your Salesforce Developer ID.
      The Salesforce Developer ID is in the format of an 8-character alphanumeric username and the `@salesforcemobilepublisher.com` domain, similar to `1a2b3c4d@salesforcemobilepublisher.com`. The Salesforce Developer ID is the same user that you invited to your Google Play account when you granted Salesforce permission to distribute your branded app.
   b. Return to the browser window or tab showing your Firebase project. In the Firebase project, click the cog icon next to Project Overview, and then click **Project settings**.
   c. Select the **Users and permissions** tab.
   d. In the email address field, add the Salesforce Developer ID that you copied from the Google Play Console.
   e. In the **Role(s)** field, select **Assign Firebase role(s)**, and select the **Admin** column of the Firebase row.
4. Register your Android app on Firebase and download its config file.
   a. In a separate browser window or tab, open your app's Setup for Mobile Publisher project. In the App Distribution section, copy the value of the Store Identifier field.
      The identifier name format is similar to:
      `com.mysalesforce.mycommunity.C00AB0000000CDefGHI.A0BCD00000001ABcDEF`
   b. Return to the browser window or tab showing your Firebase project. In the Firebase project, click the cog icon next to Project Overview, and then click **Project settings**.
   c. In the **Your apps** section, click **Add app**, and then click the Android icon.
   d. In the **Android package name** field, enter the store identifier that you copied from your app's Setup

for Mobile Publisher project.

> **Note** If you get an error message in the Firebase registration workflow that states there's already an app with that package name, close the registration window. Confirm that you see your app's store identifier in the list of apps, and then click the link to download `google-services.json`. Continue to step 7.

e. In the **App nickname** field, enter your app's name.
We recommend that you use the same app name that's entered in the Product Details section of your Setup for Mobile Publisher project.

f. Click **Register app**.

g. Click **Download google-services.json**, and then click **Next**. Note the location of the downloaded file, because you need it later.
Don't change the settings under Firebase SDK.

h. Click **Next**, and then click **Continue to console**.

5. Generate and download a private key for your Firebase service account.

a. Click the cog icon next to Project Overview, and then click **Project settings**.

b. Select the **Service accounts** tab, and then click **Generate new private key**.

c. Download the JSON file that contains the private key. Note the location of the downloaded file, because you need it later.
The file name format for this private key is similar to `project-name-firebase-adminsdk-a1bcd-a1234bc5678.json`.

> **Note** The private key for your Firebase service account is a different JSON file than the Google Cloud Platform service account key that you upload when you add a Google Play developer account to your Mobile Publisher project. Make sure that the private key you're downloading to set up push notifications contains `firebase-adminsdk` in the JSON file name.

6. Add the Firebase information to your app's Setup for Mobile Publisher project.

a. From Setup, enter *`Mobile Publisher`* in the Quick Find box, then select **Mobile Publisher**.

b. Open your app's Android project, and then scroll to the Push Notifications section.

c. In the **Firebase Admin SDK Private Key** field, upload or drop the private key that you generated for your Firebase service account.
The file name format for this private key is similar to *`project-name-firebase-adminsdk-a1bcd-a1234bc5678.json`*.

d. In the **Firebase Config File** field, upload or drop the *`google-services.json`* file that you downloaded when you registered your app on Firebase.

Enter Your Information into Your Mobile Publisher Project for Android

Now that we have all the information we need, we can go back to the Mobile Publisher and fill out the necessary fields.

REQUIRED EDITIONS

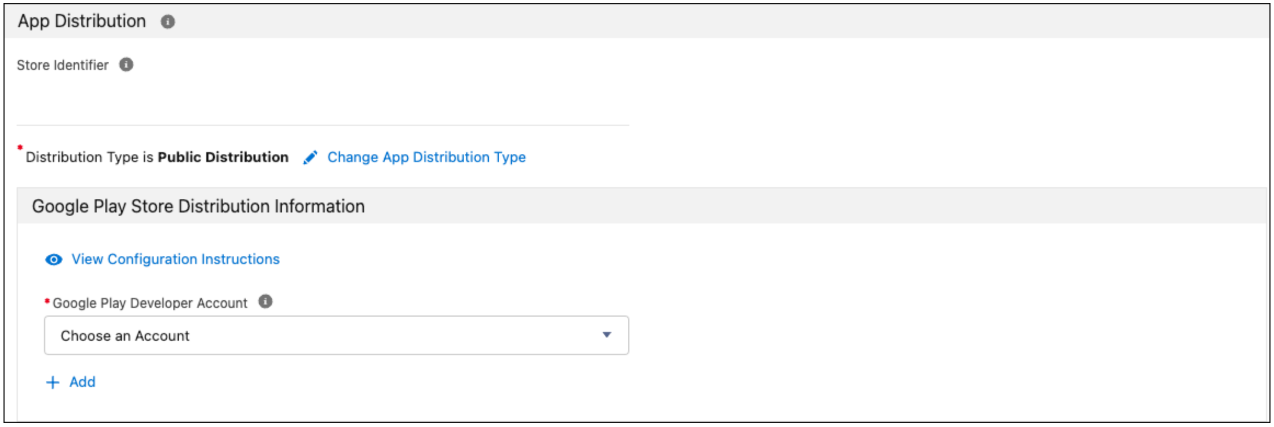
Setup for Mobile Publisher available in: Lightning Experience

Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Watch the video: https://play.vidyard.com/xPmTwXujNiAs7DuB7QwHZH

1. In Mobile Publisher, go to the App Distribution section.



App Distribution

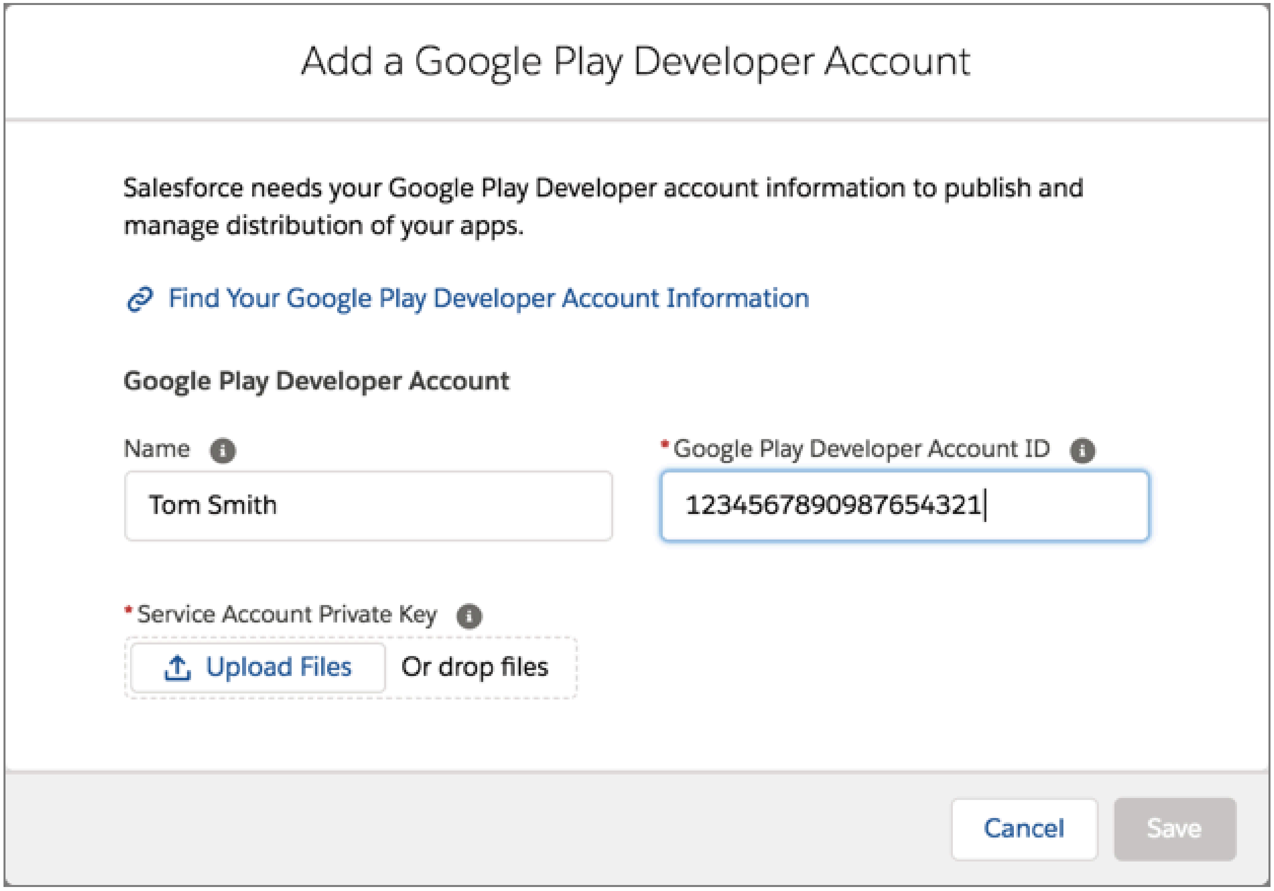
Store Identifier

*Distribution Type is **Public Distribution**

Google Play Store Distribution Information

*Google Play Developer Account: Choose an Account

2. To enter your Google Play Developer Account information, click **Add**.



Add a Google Play Developer Account

Salesforce needs your Google Play Developer account information to publish and manage distribution of your apps.

Google Play Developer Account

Name: Tom Smith

*Google Play Developer Account ID: 1234567890987654321

*Service Account Private Key

- Choose the contents of the Name field. This field is for Salesforce internal use only and can’t be

changed after you save.
- Google Play Developer Account ID is the set of numbers that you copied to your clipboard from the URL of your browser address bar.
- Service Account Private Key is the JSON file that you saved to your computer from your Google Play Console.

3. Save your work.
4. Under Push Notifications, enter the information from your Google Firebase Console. See Register Your Firebase App to Enable Mobile Publisher Android Push Notifications for more information.

If you filled out the fields correctly, you're ready to distribute your branded app to your users.

Set Up Public Distribution for iOS

Let's get your branded app ready for distribution on the public App Store.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Publisher needs certain pieces of information from your Apple Developer account to distribute your branded app. To participate in the Mobile Publisher program for iOS apps, you must have an Apple Developer Program account enrolled as an organization. See the Enrolling as an Organization section in What You Need To Enroll.

Important To distribute your Mobile Publisher app using the Fully Managed or Binary Upload distribution methods, your Apple Developer account must have an iOS distribution certificate available for your app. By default, Mobile Publisher generates an iOS distribution certificate from your Apple Developer account. However, if you've reached the limit for iOS distribution certificates, you must make one available to distribute your app. Before you submit your app, verify that your Apple Developer account has an iOS distribution certificate available. To verify, log in to your Apple Developer account, go to the **Certificates, Identifiers & Profiles** page, and click **+** to add a new certificate. If you get an error that there are no iOS distribution certificates available, then you've reached the limit and you must make a certificate available for your Mobile Publisher app. If you don't get an error, you can cancel the new certificate and proceed to submit your app to Mobile Publisher.

1. **Choose Distribution Type**
   First, we must choose the correct distribution type for your branded app.

2. Register Salesforce Bundle ID
   Register your Salesforce bundle ID with your Apple developer account.
3. Create a Placeholder App with Privacy Questions and Age Ratings
   Create a placeholder app in your App Store Connect account.
4. Update Test User Credentials
   Update the test user credentials in your App Store Connect account to avoid app rejections.
5. Invite Salesforce to Your App Store Connect Account
   To distribute your branded app, grant Salesforce with App Manager access to your App Store Connect.
6. Record Your Apple Developer Team Account Info
   Mobile Publisher needs your Apple developer team name and ID to distribute your branded app.
7. Enter Your Apple Developer Information into Your Mobile Publisher Project
   After your prep work, return to your Mobile Publisher project and enter your Apple Developer Team Account information.

Choose Distribution Type

First, we must choose the correct distribution type for your branded app.

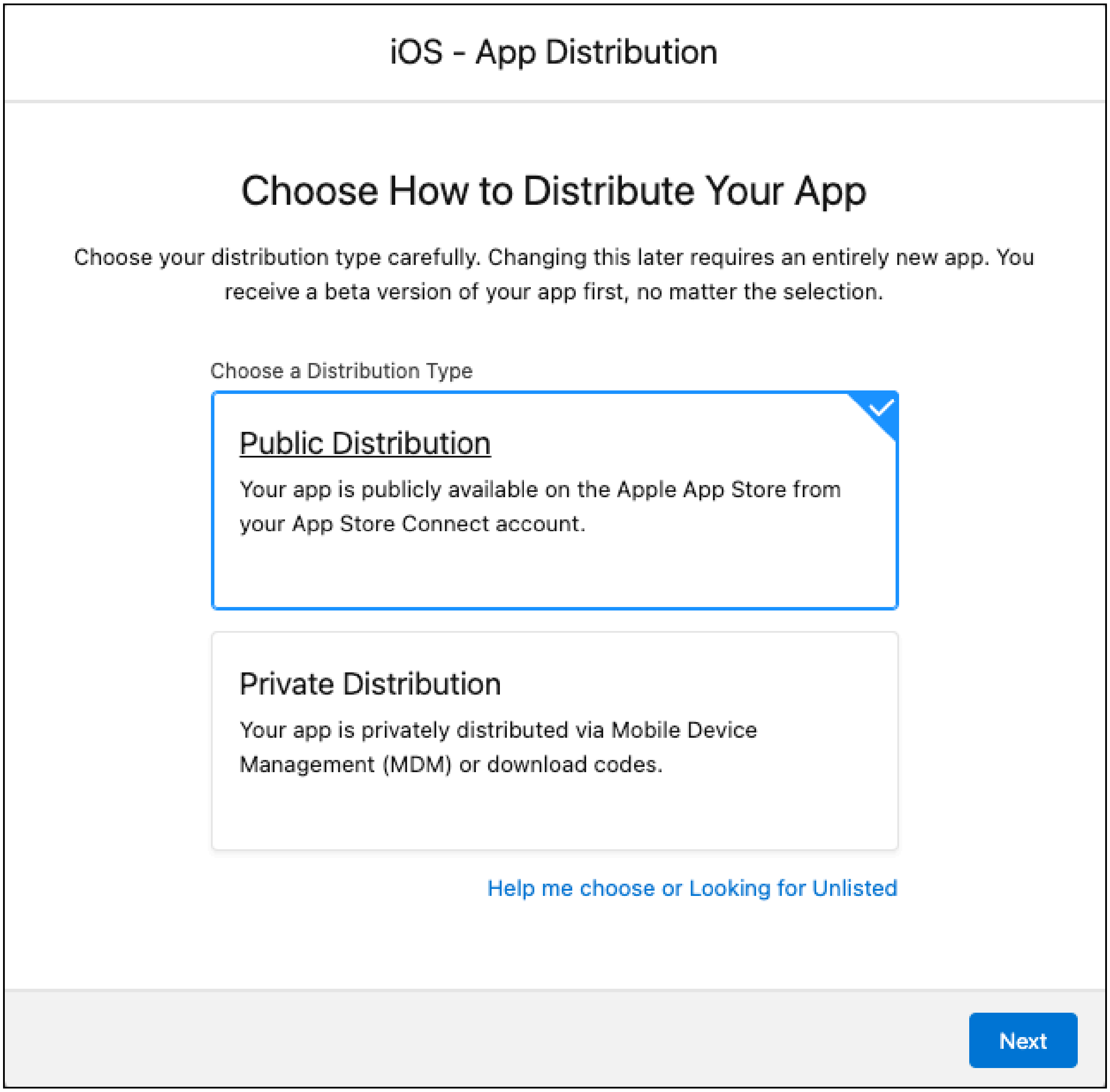
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Watch the video: https://play.vidyard.com/ChnGsudDBC75AWfBd87MkA

1. Select **Public Distribution** from the App Distribution menu, and click **Next**.



iOS - App Distribution

Choose How to Distribute Your App

Choose your distribution type carefully. Changing this later requires an entirely new app. You receive a beta version of your app first, no matter the selection.

Choose a Distribution Type

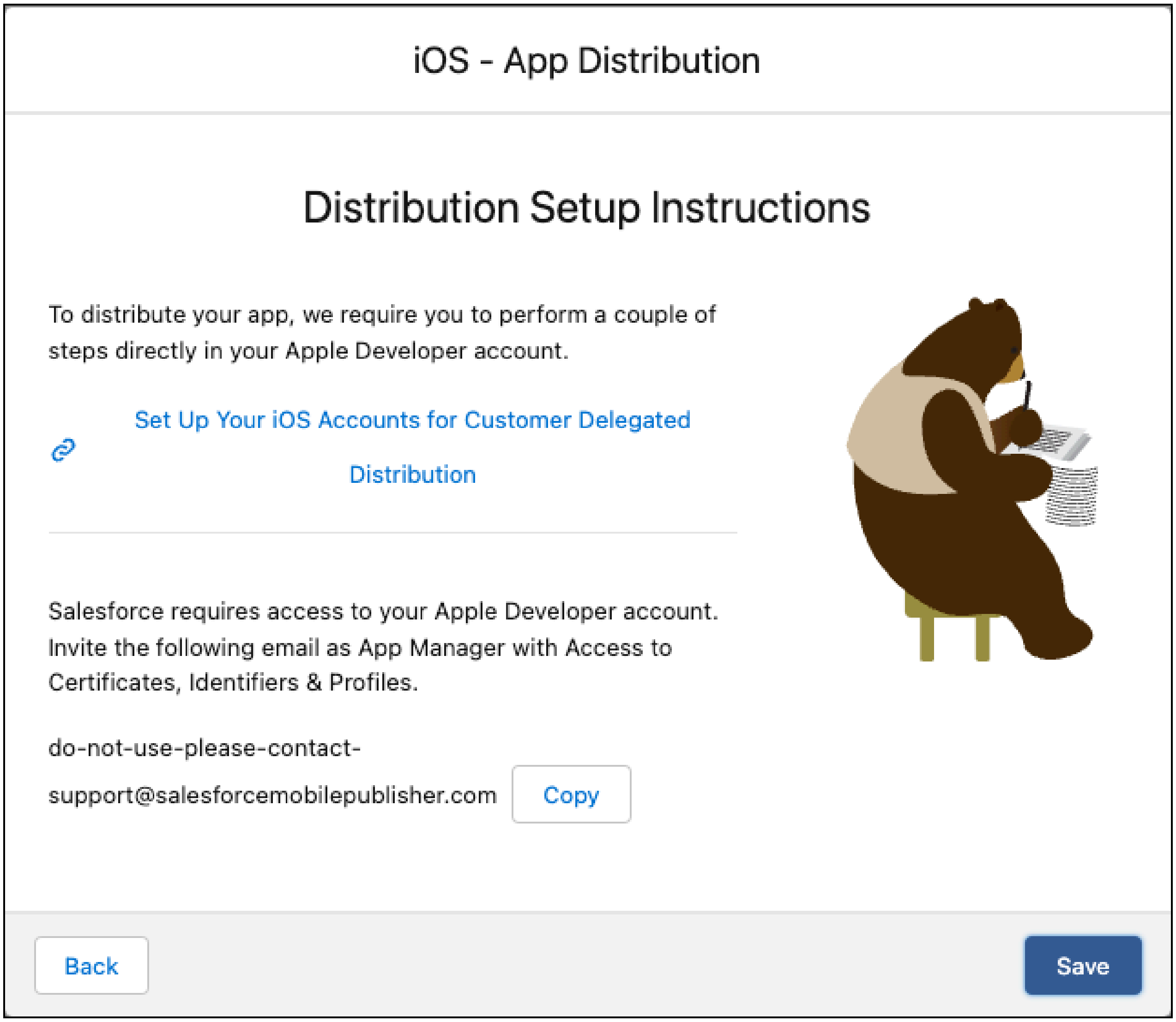
Public Distribution

Your app is publicly available on the Apple App Store from your App Store Connect account.

Private Distribution

Your app is privately distributed via Mobile Device Management (MDM) or download codes.

2. Click **Copy** to copy and save the Salesforce Developer Email onto your computer for future reference.



iOS - App Distribution

Distribution Setup Instructions

To distribute your app, we require you to perform a couple of steps directly in your Apple Developer account.

Set Up Your iOS Accounts for Customer Delegated Distribution

Salesforce requires access to your Apple Developer account. Invite the following email as App Manager with Access to Certificates, Identifiers & Profiles.

do-not-use-please-contact-support@salesforcemobilepublisher.com

You're going to need the Salesforce Developer Email address when you log into your Apple Developer account to grant Salesforce access to distribute your app.

Register Salesforce Bundle ID

Register your Salesforce bundle ID with your Apple developer account.

Watch the video: https://play.vidyard.com/X4gnXRX2i7AxiaAu4PL7Gb

1. To register your Salesforce Bundle ID, go to https://developer.apple.com/account and log into your developer account.
2. Click **Certificates, IDs & Profiles**, then click **Identifiers**.
3. To register a new ID, click **+**, then select **App IDs**.
4. On the Select a type page, click **App**.
5. Enter `Salesforce Mobile Publisher` for the description.
6. Select the App ID prefix.
   a. If you see **Prefix Listed (Team ID)** without a dropdown in the App ID Prefix section, your App ID

Prefix is set.

   b. If you see **Prefix Listed** in the App ID Prefix section with a dropdown to the right, select the App ID Prefix with **(Team ID)** listed next to it from the dropdown.

7. For the **Bundle ID**, enter the value of the `Store Identifier` field in your app's Setup for Mobile Publisher project.
8. Click **Continue**.
9. Click **Register**.
10. Click **Done**.

Create a Placeholder App with Privacy Questions and Age Ratings

Create a placeholder app in your App Store Connect account.

After logging into https://appstoreconnect.apple.com, complete these steps.

Watch the video: https://play.vidyard.com/R5Bck1mjCmHNM2oo1RhbhD

Important Apple has introduced new requirements for some content rights information, including more granular age ratings and new ratings questions. You must update your age ratings and answer the new questions by January 26, 2026, in order to continue submitting app updates. These requirements are also enforced for apps that are privately distributed.

1. Click **My Apps**.
2. Click **+**, then **New App**.
3. Set the platform to iOS, and fill out the required fields.
   a. Fill out the App Name, which must match the name entered in the `App Store Name` field in the metadata form.
   b. Set the Primary Language to the same language that you entered in the Mobile Publisher submission.
   c. Select the value in the `Store Identifier` field presented in Mobile Publisher as the Bundle ID.
   d. Enter *`Salesforce-MM-DD-YYYY`* for the SKU.
      An example for Salesforce-MM-DD-YYYY is Salesforce-07-01-2020.
4. For the User Access, select **Full Access**.
5. Click **Create**.
6. Select the placeholder app that you created.
7. In the sidebar, select **App Privacy**.
8. Click **Get Started**.
9. Specify data that your app collects from your company or from your third-party partners.
   To learn more about the baseline disclosures for your Experience Cloud Mobile Publisher iOS app, see App Privacy - Experience Cloud App.
   To learn about the baseline disclosures for your Salesforce Mobile Publisher iOS app, see App Privacy - Salesforce App.
10. After you answer the app questions, click **Publish** in the top-right corner of the privacy page.
11. In the Distribution tab, select the App Information section.
12. Select **Edit** next to **App Age Ratings** and answer all the seven steps listed. Generally, the default

answer is No to most of the questions, but your app's needs can be different. We recommend you pay special attention to the Capabilities and the Wellness or Medical sections.

13. Select **Done**.

Update Test User Credentials

Update the test user credentials in your App Store Connect account to avoid app rejections.

Before you begin, make sure that you have the correct credentials for end-user testing by Apple.

- To test a sandbox site in your Mobile Publisher configuration, have your sandbox site credentials ready.
- To test a production site in your Mobile Publisher configuration, have your production site credentials ready.
- If you have no sandbox site in your Mobile Publisher configuration, have your production site credentials ready.

Watch the video: https://play.vidyard.com/R5Bck1mjCmHNM2oo1RhbhD

1. Log in to your App Store Connect account.
2. Navigate to your app.
3. Click the **App Store** tab, then go to App Review Information.
4. Select **Sign-in required**, then enter the end-user test credentials for your Experience Cloud site.
5. Save your changes.
6. Click the **TestFlight** tab.
7. From General Information, go to Test Information and then Beta App Review Information.
8. Under Sign-in required, enter the credentials for end-user testing by Apple. To get your sandbox site tested, enter the sandbox site test credentials. To get your production site tested, or if you don't have a sandbox site in your Mobile Publisher configuration, enter the production site test credentials.
   To avoid an app rejection due to sign-in issues, we also recommend that you disable two-factor authentication and single sign-on (SSO) for the test user.
9. Specify a range of IP addresses for the test user. To specify the IP address range, clone the Salesforce user profile for the Apple Reviewer test user, and then add the login IP range from 1.1.1.1 to 255.255.255.255 for that profile. Defining the IP range limits the change to only this user without affecting other users.
   See Restrict Login IP Ranges in Profiles for more information on adding a range of IP addresses.
10. Save your changes.

When your app is submitted in beta, Mobile Publisher populates the Notes field in the App Store tab and the Review Notes field in the TestFlight tab with sign-in information for testing. To avoid app rejections, review the sign-in information and make sure that it's up to date and works with the credentials that you entered.

If you have a sandbox site in your Mobile Publisher configuration but you want your production site tested, make sure to update the Notes and Reviewer Notes fields with the sign-in information for your production site.

With every Mobile Publisher for Experience Cloud app update, Apple attempts to log in to the app as part of the approval process. To prevent delays in app delivery, make sure that:

- The test user credentials are always valid. While your app is in beta, keep the test user credentials updated in the App Store tab and the TestFlight tab.
- The reviewer notes always describe the most up-to-date sign-in information. The sign-in information must always work with the test user credentials that you enter.

Invite Salesforce to Your App Store Connect Account

To distribute your branded app, grant Salesforce with App Manager access to your App Store Connect.

Complete these steps after logging into https://appstoreconnect.apple.com.

Watch the video: https://play.vidyard.com/ZZHqmoDTRfBorTNZJpQHW5

1. Navigate to Users and Access.
2. To add a user, click **+** to add a user.
3. Fill out these fields:
   a. Enter `Publisher` for the First Name.
   b. Enter `Salesforce` for the Last Name.
   c. Enter the Salesforce Developer Email address for Email.

      Note Salesforce provides the Salesforce Developer Email address when you select an app distribution type. To view the email address, in Salesforce, go to **Setup | Mobile Publisher** and open your iOS or Android draft. In the App Distribution section, click the **Choose App Distribution Type** link. Select **Public** or **Private** (you can change this selection before you submit the draft) and click **Next**. The App Distribution dialog displays the email address beside the Copy button.
4. Select the **App Manager** role.
5. Check the **Access to Certificates, Identifiers & Profiles** check box.
6. Under Apps, select the Salesforce Mobile Publisher app you created earlier.
7. Click **Invite**.

Important Remind your Apple developer store owner or admins to not remove the Mobile Publisher Developer ID user from the account or organization.

Record Your Apple Developer Team Account Info

Mobile Publisher needs your Apple developer team name and ID to distribute your branded app.

When prompted to "record" a value or ID, be sure to note the value or ID exactly as written, and save it for later. These values and IDs are needed for the final step of setting up delegated public or private distribution for iOS.

Watch the video: https://play.vidyard.com/d8zyxUAEhDCpvbndRV2PFS

1. Go to https://developer.apple.com/account and log in to your Apple Developer account.
2. Navigate to the **Membership details** page.
3. Record the value in the **Entity Name** field for use in a later step.
4. Go to https://appstoreconnect.apple.com/login and log in to your App Store account.
5. Navigate to the **Users and Access** page.
6. Select the **Integrations** tab.
7. Select **App Store Connect API**, and then select the **Team Keys** tab. The Issuer ID appears. Record the Issuer ID for use in a later step
8. Create an API key.
   a. Click **+**.
   b. Name the key Mobile Publisher.
   c. In the Access field, select **App Manager**.
9. The newly generated key is called the Apple Private Key. Record the Key ID for use in a later step.
10. Refresh the webpage.
11. Hover over the **Apple Private Key** and download it. This key can only be downloaded at this time.

Enter Your Apple Developer Information into Your Mobile Publisher Project

After your prep work, return to your Mobile Publisher project and enter your Apple Developer Team Account information.

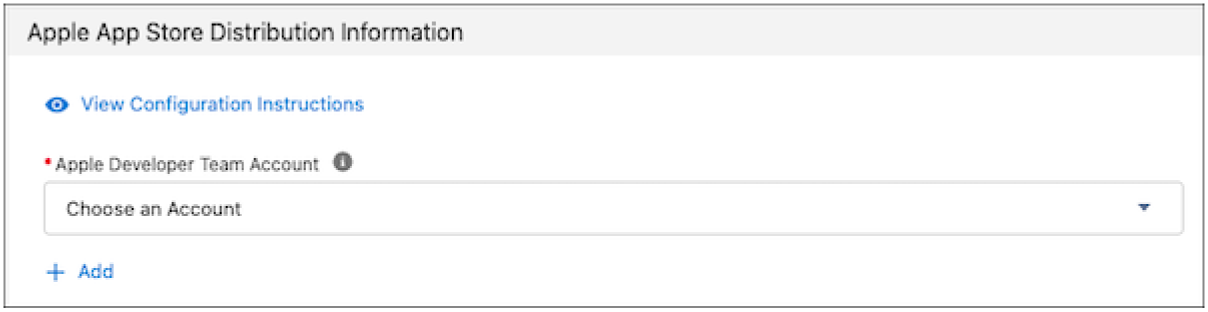
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Watch the video: https://play.vidyard.com/d8zyxUAEhDCpvbndRV2PFS

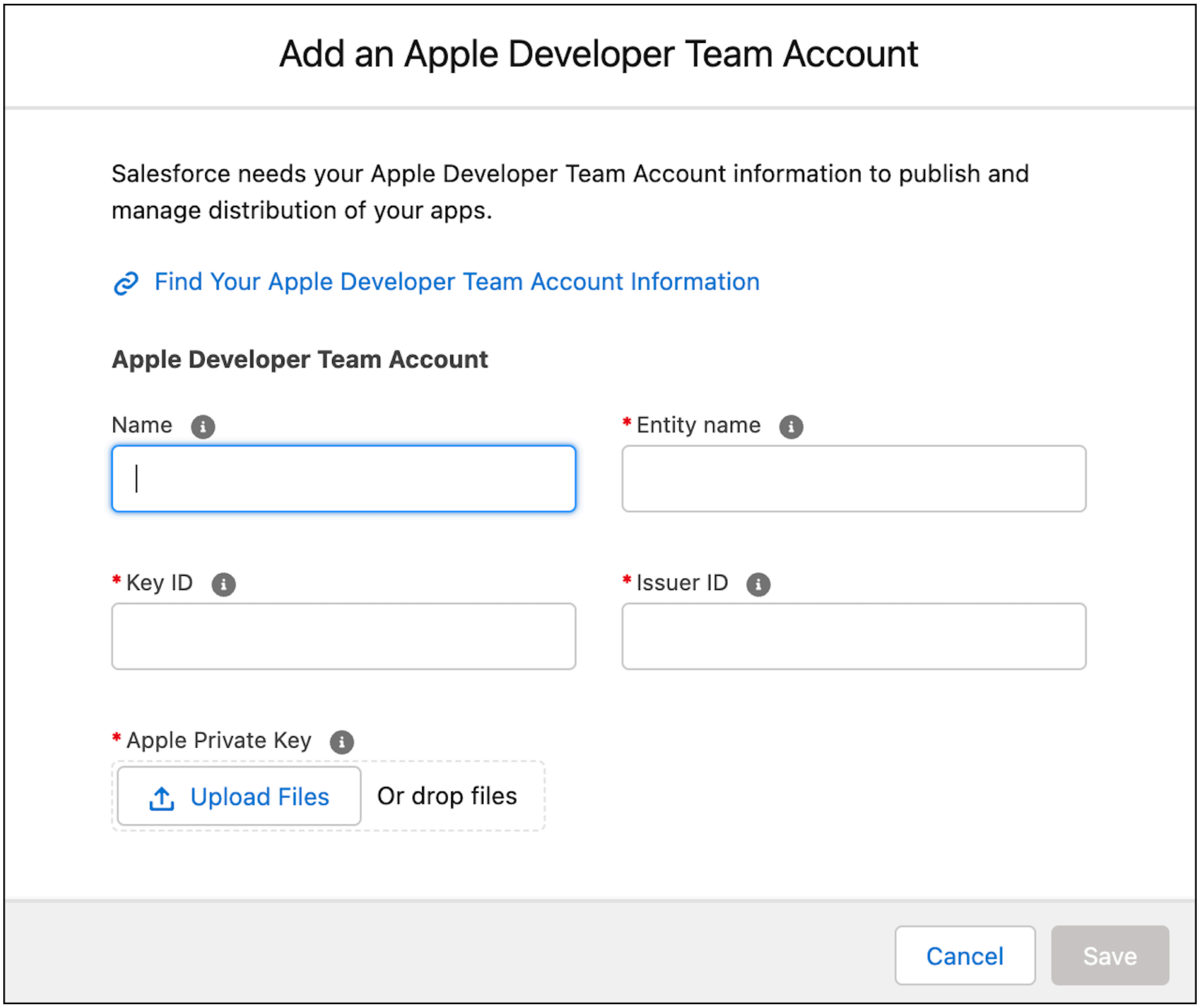
1. In the Mobile Publisher, go to the App Distribution section.



Apple App Store Distribution Information

View Configuration Instructions
* Apple Developer Team Account
Choose an Account
+ Add

2. Click **Add** to enter your Apple Developer Team Account information.



Add an Apple Developer Team Account

Salesforce needs your Apple Developer Team Account information to publish and manage distribution of your apps.

Find Your Apple Developer Team Account Information

Apple Developer Team Account

Name	* Entity name
* Key ID	* Issuer ID
* Apple Private Key	

3. You recorded these values earlier, on the Record Your Apple Developer Team Account Info page.
   - Name is chosen by you for internal use only.
   - Entity Name is the name from your Apple Developer Account.
   - Issuer ID is the ID from your App Store Connect.
   - Key ID is the ID from your App Store Connect.
   - Apple Private Key is the key downloaded from App Store Connect.
4. Save your changes.

If all fields are correct, you can distribute your branded app to your users.

Set Up Private Distribution

You manage distribution of your branded app through your developer account privately. Your branded app is only accessible to your end users privately, and isn't publicly available on the Apple App Store or Google Play.

Create a Binary Handoff Project for the Salesforce Mobile App

Use the binary handoff distribution method to have Salesforce build your app and provide you with the binary file (.ipa or .aab) for manual signing and distribution. This method is ideal for organizations with security protocols that require in-house code signing or specific internal distribution processes. Use the binary handoff distribution when you're certain you don't want to change the distribution method, because you can't change it to fully managed or binary upload.

Set Up Private Distribution for Android
Get your branded app ready for distribution on the private Google Play store.

Set Up Delegated Private Distribution for iOS
Let's get your branded app ready for distribution on the private Apple App store.

Create a Binary Handoff Project for the Salesforce Mobile App

Use the binary handoff distribution method to have Salesforce build your app and provide you with the binary file (.ipa or .aab) for manual signing and distribution. This method is ideal for organizations with security protocols that require in-house code signing or specific internal distribution processes. Use the binary handoff distribution when you're certain you don't want to change the distribution method, because you can't change it to fully managed or binary upload.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important After you request the app binary, the distribution method is locked to Binary Handoff and can't be changed.

When you use Binary Handoff, Salesforce builds the application binary and delivers it to you, but your organization maintains full control over the signing identity and store submission.

1. From Setup, in the Quick Find box, enter Mobile Publisher, and then select Mobile Publisher.
2. Click Create New App.
3. Select **External Client App** and click **Next**.
4. Select **Experience Builder Site** as the app type and click **Next**.
5. Enter a project name and click **Next**.

   Important You can't change the project name after you save it. The project name is only seen internally, not by the app users.

6. Click Start for the platform (iOS or Android) you want to configure.
7. In the Set Up the Mobile App section, click Start.
8. Click Change App Distribution Method, select Binary Handoff, and click Done.

9. **Important** For Android apps, register your app in Firebase using the Customer Store Identifier field in the Setup UI to obtain the google-services.json file. If your app uses deep links, supply the SHA-256 value in the setup form. You can find the value in your Google Play console, in **Test & Release** | **App Integrity** | **App Signing** | **Play App Signing** | **Settings**.

   Complete the Customer Store Identifier and all other required fields on the form.
10. Click Submit.
11. In the Set Up an External Client App section, click **Start** to configure and deploy your ECA to production.

    Note To test your app in a sandbox, deploy the ECA to the sandbox org after deploying to production. During ECA setup, you can also turn on push notifications by adding your Apple APNs key (.p8 file) or Google Firebase Admin SDK JSON file.
12. From the main Mobile Publisher Setup page, click Request Binary. Salesforce builds the app binary and delivers it to you.

After receiving the binary, you're responsible for re-signing it with your own signing certificates and private keys and submitting it to the app stores.

Set Up Private Distribution for Android

Get your branded app ready for distribution on the private Google Play store.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Publisher needs certain information from your Google Play Developer account to distribute your branded app.

- If you don’t have a Google Play Developer account, click here to sign up.
- Always use a G-Suite with your organization domain. Access to the app is available only to users who are part of that G-Suite organization. For example, if your organization’s name is AYZ Inc., use a G-Suite with the same domain and app users who are part of the organization.
- There’s no beta app provided for Android private distribution. If you don’t use G-Suite and Google for work, refer to Distribute Private apps.

1. **Mobile Publisher Best Practices for App Content Declarations**
   After you create a placeholder app, you complete Google Play’s required App Content declarations to

publish your Android app. When you publish a Mobile Publisher Android app, follow best practices specific to Mobile Publisher for the Privacy policy, App access, Data safety, and Financial features declarations.

2. Grant Permission to Distribute Your Branded App
   Grant Salesforce permission to distribute your branded app.
3. Create Service Account User and Find Your Google Play Developer ID
   To manage the distribution your branded app with your developer account, Mobile Publisher needs your Service Account User and Google Play Developer ID info.
4. Register Your Firebase App to Enable Mobile Publisher Android Push Notifications
   To set up push notifications for your Android app, provide Mobile Publisher with information from a Google Firebase project associated with your app. To get the required information, register your Android app and generate a private key. Then, submit the information to your app's Setup for Mobile Publisher project.
5. Find Your Managed Google Play Organization Info
   To distribute your branded app, Mobile Publisher needs your Managed Google Play Organization info.
6. Enter Your Information into Your Mobile Publisher Project for Android
   Now that we have all the information we need, we can go back to the Mobile Publisher and fill out the necessary fields.

Mobile Publisher Best Practices for App Content Declarations

After you create a placeholder app, you complete Google Play's required App Content declarations to publish your Android app. When you publish a Mobile Publisher Android app, follow best practices specific to Mobile Publisher for the Privacy policy, App access, Data safety, and Financial features declarations.

Watch the video: https://play.vidyard.com/GaqayoPk3sSbZ5LUjpNQYh

To find your app's declarations in the Google Play console, scroll to the Policy and programs section in the left navigation pane. Then, click **App content**. Complete the declarations on the **Needs attention** tab.

Declaration: Privacy policy

Enter information about your app's privacy policy. Make sure to link to your app's privacy policy. Don't enter the link to the Salesforce privacy policy.

Note Google requires that your app's privacy policy contains specific information about your app and is linked from your app. See Set Up Your Android App for Google's Privacy Policy Requirements.

Declaration: App access

Enter information about how Google reviewers can access your Mobile Publisher app, including your app's log-in instructions and details. See Update Test User Credentials.

Declaration: Data safety

Enter information about how your app collects data. To learn about the baseline disclosures for your Mobile Publisher Android app, see the Google Data Safety reference for Mobile Publisher apps.

Note If your app doesn't collect any data from your users, select **Device or other IDs** in the Data safety declaration, and use the Google Data Safety reference for Mobile Publisher apps to answer the questions.

For instructions on filling out the Data safety form, see Provide information for Google Play's Data safety section in the *Google Play Console Help*.

Declaration: Financial features

If you created your Mobile Publisher Android app before August 2023, make sure to return to the Google Play console and complete your app's Financial features declaration. See Complete and submit the declaration in the *Google Play Console Help*.

Important To publish your Android app on Google Play, make sure to also complete the declarations for Ads, Content rating, Target audience, News apps, COVID-19 contact tracing and status apps, and Government apps. See the Google Play Console Help.

Grant Permission to Distribute Your Branded App

Grant Salesforce permission to distribute your branded app.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Complete these steps after logging into the Google Play Console with your Google Play Developer account.

Watch the video: https://play.vidyard.com/v63U4rHBcvRUGtY7s4Hmon

Invite the Salesforce Developer ID

1. Navigate to **Users and Permissions** and click **Invite New Users**.

2. Enter the Salesforce Developer Email you copied from your clipboard and don't set an expiration date.

   Note Salesforce provides the Salesforce Developer Email address when you select an app distribution type. To view the email address, in Salesforce, go to **Setup | Mobile Publisher** and open your Android draft. In the App Distribution section, click the **Choose App Distribution Type** link. Select **Public** or **Private** (you can change this selection before you submit the draft) and click **Next**. The App Distribution dialog displays the email address beside the Copy button.

3. Under **App Permissions**, click **Add App** and select the placeholder app you created earlier, then click **Apply**.
4. In the pop-up window, select the following permissions:
   a. View app information (read-only)
   b. View financial data
   c. Manage orders and subscriptions
   d. Edit and delete draft apps
   e. Release to production, exclude devices, and use Play App Signing
   f. Release apps to testing tracks
   g. Manage testing tracks and edit tester lists
   h. Manage store presence
   i. Reply to reviews
5. Click **Invite User**.

Create Service Account User and Find Your Google Play Developer ID

To manage the distribution your branded app with your developer account, Mobile Publisher needs your Service Account User and Google Play Developer ID info.

Complete these steps from your Google Play Console.

Important You must be the Account Owner of your Google Cloud Platform account to perform the following steps.

Watch the video: https://play.vidyard.com/v63U4rHBcvRUGtY7s4Hmon

1. Log in to your **Google Cloud Console** (https://console.cloud.google.com/) using the same email associated with your **Google Play Console** account.
2. In the search bar, search for **Create a Project**, fill in the **Project name**, and click **Create**.

   Note You can skip creating a new project if you want to reuse an existing project from your Google Cloud Console.

3. Search for **APIs and Services** in the search box, choose **Library** from the left-hand navigation bar and search for **Google Play Android Developer API**.
4. Select **Google Play Android Developer API** and click **Enable**.
5. After it‘s enabled, from the home page, search for **Service Accounts** in the search bar.
6. At the top of the page, click **+ Create Service Account**.
7. Enter the Service Account Name. This field automatically populates the Service Account ID.
8. Click **Create and Continue**.

9. In the **Grant this service account access to project** section, select the role as **Service Accounts | Service Account User**, then click **Continue**.
10. Leave the optional **Grant user access to this service account** section blank and click **Done**.
11. Now that a new service account has been created, click the three dots under **Actions** and select **Manage Keys**.
12. Click **Add Key | Create New Key** then select **JSON** and choose **Create**. This action saves the JSON onto your computer (you must upload this file later to the Mobile Publisher Setup).
13. From the left panel navigation menu, click **Service Accounts**, then click the newly created service account.
14. Copy the email address of the Service Account User you created to your clipboard. The email appears under the **Email** field under the **Details** tab. The email address must end in .iam.gserviceaccount.com.

Now go back to the Google Play Console, and from the left panel navigation menu, click **Account Details** and copy the 19-Digit ID from the **Developer Account ID** field. Alternatively, you can also find your Google Play Developer ID in the URL of your browser address bar.

- In your browser address bar, look for `https://play.google.com/console/u/0/developers/<19 Digit ID>/app-list`.
- Copy the set of 19 digits (your Google Play Developer Account ID) after `developers/` to your clipboard.

Note The private key that you download from the Google Play Console is a different JSON file than the Firebase admin SDK private key that you download when you set up push notifications.

Register Your Firebase App to Enable Mobile Publisher Android Push Notifications

To set up push notifications for your Android app, provide Mobile Publisher with information from a Google Firebase project associated with your app. To get the required information, register your Android app and generate a private key. Then, submit the information to your app's Setup for Mobile Publisher project.

Watch the video: https://play.vidyard.com/MdJFfAC6Zw31ncGjPJ6Rjf

Note If you submitted your Mobile Publisher Android app before November 2023, don't create a new Google Firebase project. Confirm that you've completed steps 4 through 6 with your existing Firebase project. You must complete those steps to receive a new Android app build.

1. Log in to the Google Firebase Console with the Google Play account associated with your Mobile Publisher app.
2. Create a Firebase project.
   a. In the Google Firebase Console, click **Create a project**.
   b. Click **Add Firebase to Google Cloud project**.
   c. In the **Project name** field, select the Google Cloud project where you previously created a service account user.
   d. Follow the prompts to create a project.

3. Assign the project’s Firebase admin role to your Salesforce Developer ID.
    a. In a separate browser window or tab, open the Google Play Console, and then click **Users and permissions**. In the **Users** tab, copy your Salesforce Developer ID.
    The Salesforce Developer ID is in the format of an 8-character alphanumeric username and the *`@salesforcemobilepublisher.com`* domain, similar to *`1a2b3c4d@salesforcemobilepublisher.com`*. The Salesforce Developer ID is the same user that you invited to your Google Play account when you granted Salesforce permission to distribute your branded app.
    b. Return to the browser window or tab showing your Firebase project. In the Firebase project, click the cog icon next to Project Overview, and then click **Project settings**.
    c. Select the **Users and permissions** tab.
    d. In the email address field, add the Salesforce Developer ID that you copied from the Google Play Console.
    e. In the **Role(s)** field, select **Assign Firebase role(s)**, and select the **Admin** column of the Firebase row.
4. Register your Android app on Firebase and download its config file.
    a. In a separate browser window or tab, open your app’s Setup for Mobile Publisher project. In the App Distribution section, copy the value of the Store Identifier field.
    The identifier name format is similar to: *`com.mysalesforce.mycommunity.C00AB0000000CDefGHI.A0BCD00000001ABcDEF`*
    b. Return to the browser window or tab showing your Firebase project. In the Firebase project, click the cog icon next to Project Overview, and then click **Project settings**.
    c. In the **Your apps** section, click **Add app**, and then click the Android icon.
    d. In the **Android package name** field, enter the store identifier that you copied from your app's Setup for Mobile Publisher project.

    Note If you get an error message in the Firebase registration workflow that states there's already an app with that package name, close the registration window. Confirm that you see your app's store identifier in the list of apps, and then click the link to download `google-services.json`. Continue to step 7.

    e. In the **App nickname** field, enter your app’s name.
    We recommend that you use the same app name that’s entered in the Product Details section of your Setup for Mobile Publisher project.
    f. Click **Register app**.
    g. Click **Download google-services.json**, and then click **Next**. Note the location of the downloaded file, because you need it later.
    Don’t change the settings under Firebase SDK.
    h. Click **Next**, and then click **Continue to console**.
5. Generate and download a private key for your Firebase service account.
    a. Click the cog icon next to Project Overview, and then click **Project settings**.
    b. Select the **Service accounts** tab, and then click **Generate new private key**.
    c. Download the JSON file that contains the private key. Note the location of the downloaded file, because you need it later.
    The file name format for this private key is similar to `project-name-firebase-adminsdk-a1bcd-a1234bc5678.json`.

> **Note** The private key for your Firebase service account is a different JSON file than the Google Cloud Platform service account key that you upload when you add a Google Play developer account to your Mobile Publisher project. Make sure that the private key you're downloading to set up push notifications contains `firebase-adminsdk` in the JSON file name.

6. Add the Firebase information to your app's Setup for Mobile Publisher project.
   a. From Setup, enter *`Mobile Publisher`* in the Quick Find box, then select **Mobile Publisher**.
   b. Open your app's Android project, and then scroll to the Push Notifications section.
   c. In the **Firebase Admin SDK Private Key** field, upload or drop the private key that you generated for your Firebase service account.
      The file name format for this private key is similar to *`project-name-firebase-adminsdk-a1bcd-a1234bc5678.json`*.
   d. In the **Firebase Config File** field, upload or drop the *`google-services.json`* file that you downloaded when you registered your app on Firebase.

Find Your Managed Google Play Organization Info

To distribute your branded app, Mobile Publisher needs your Managed Google Play Organization info.

If you choose to distribute your app via the Private distribution type, gather certain pieces of info to properly distribute your app.

1. Navigate to your Google Play for Work account and log in.
2. Go to Settings.
3. Copy the following info:

   Organization Name is the name of your organization that can be found in your Managed Google Play Organization Account.

   Organization ID can be found in your Managed Google Play Organization Account. There's a maximum of 10 characters and it must begin with LC0 or C0.

Enter Your Information into Your Mobile Publisher Project for Android

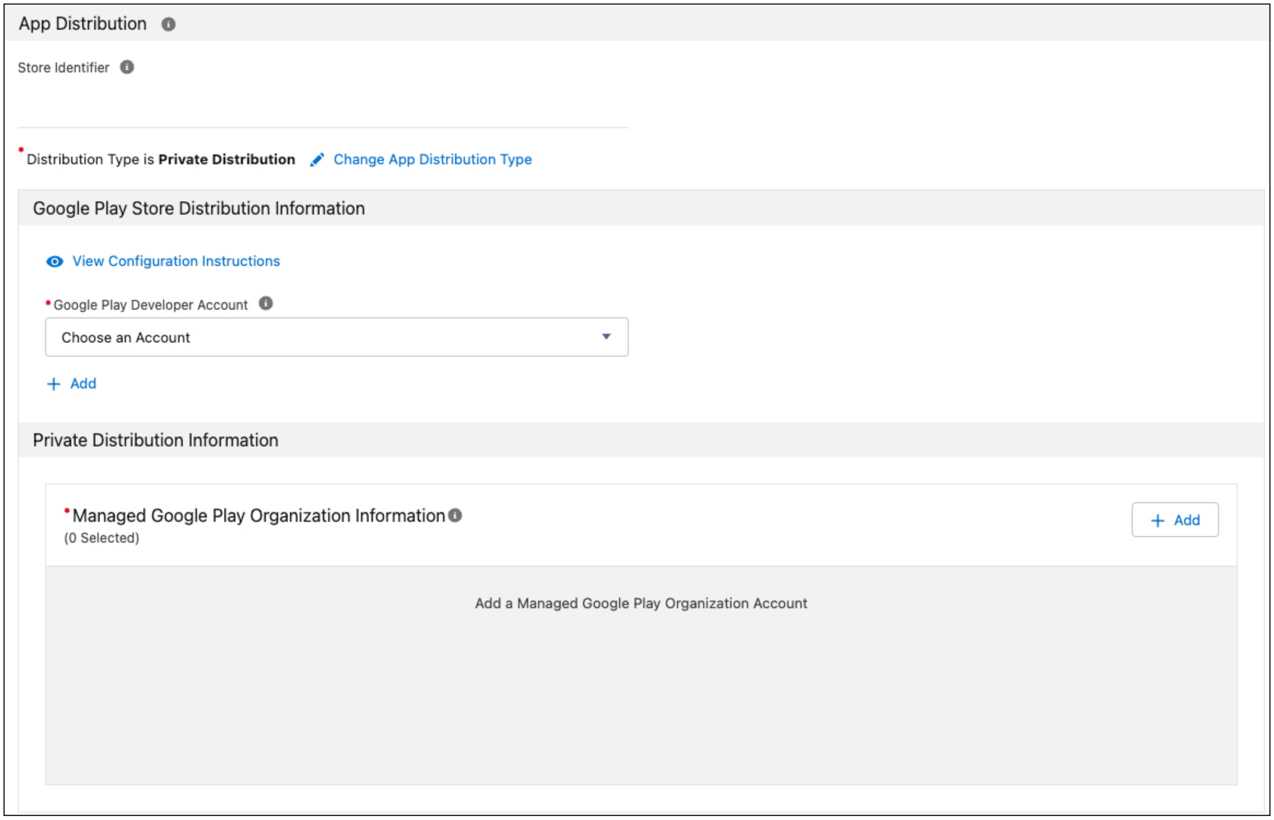
Now that we have all the information we need, we can go back to the Mobile Publisher and fill out the necessary fields.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

1. In the Mobile Publisher, go to the App Distribution section.



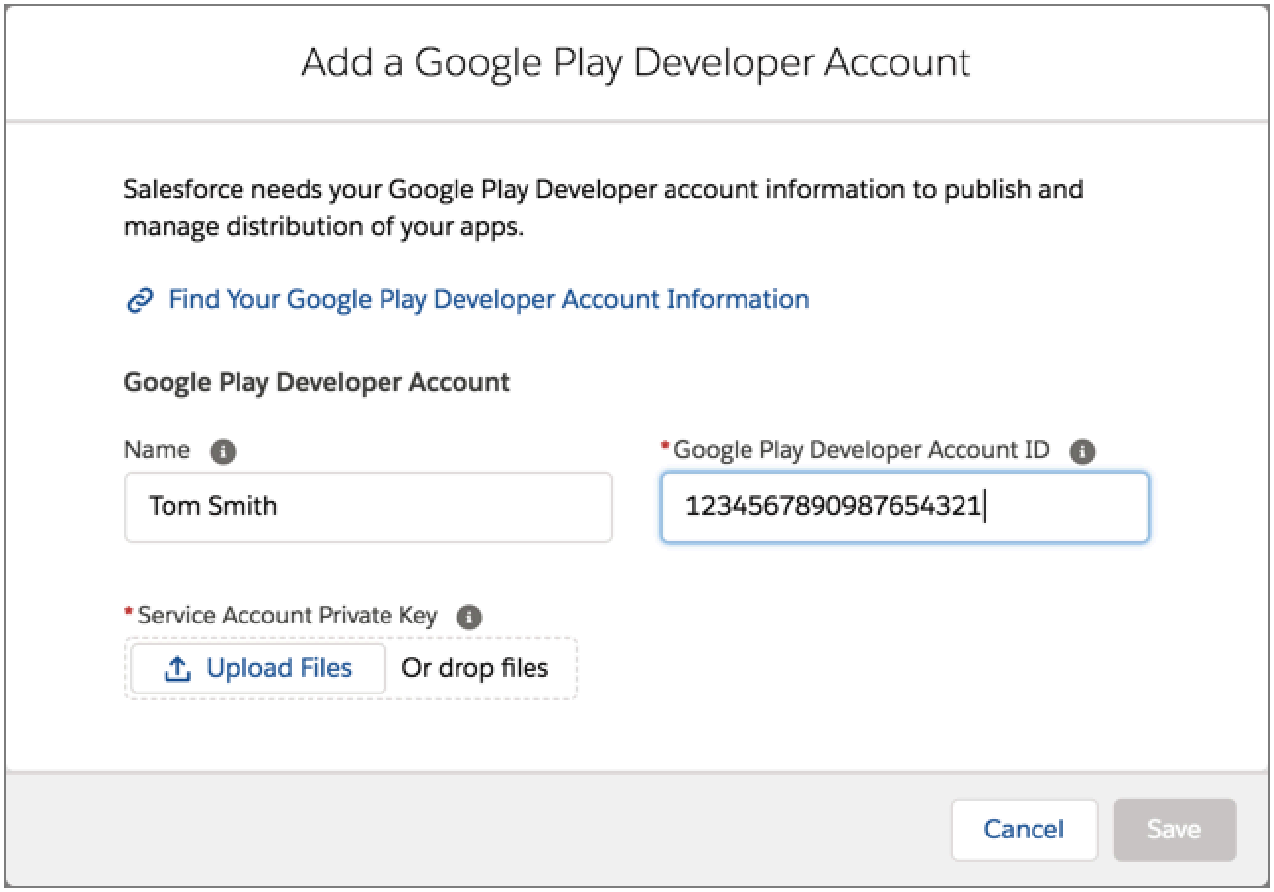
Distribution Type is **Private Distribution**

Google Play Developer Account: Choose an Account

Managed Google Play Organization Information (0 Selected)

App Distribution
Store Identifier
Change App Distribution Type
Google Play Store Distribution Information
View Configuration Instructions
Add
Private Distribution Information
Add
Add a Managed Google Play Organization Account

2. To enter your Google Play Developer Account information, click **Add**.



Add a Google Play Developer Account

Salesforce needs your Google Play Developer account information to publish and manage distribution of your apps.

Find Your Google Play Developer Account Information

Google Play Developer Account

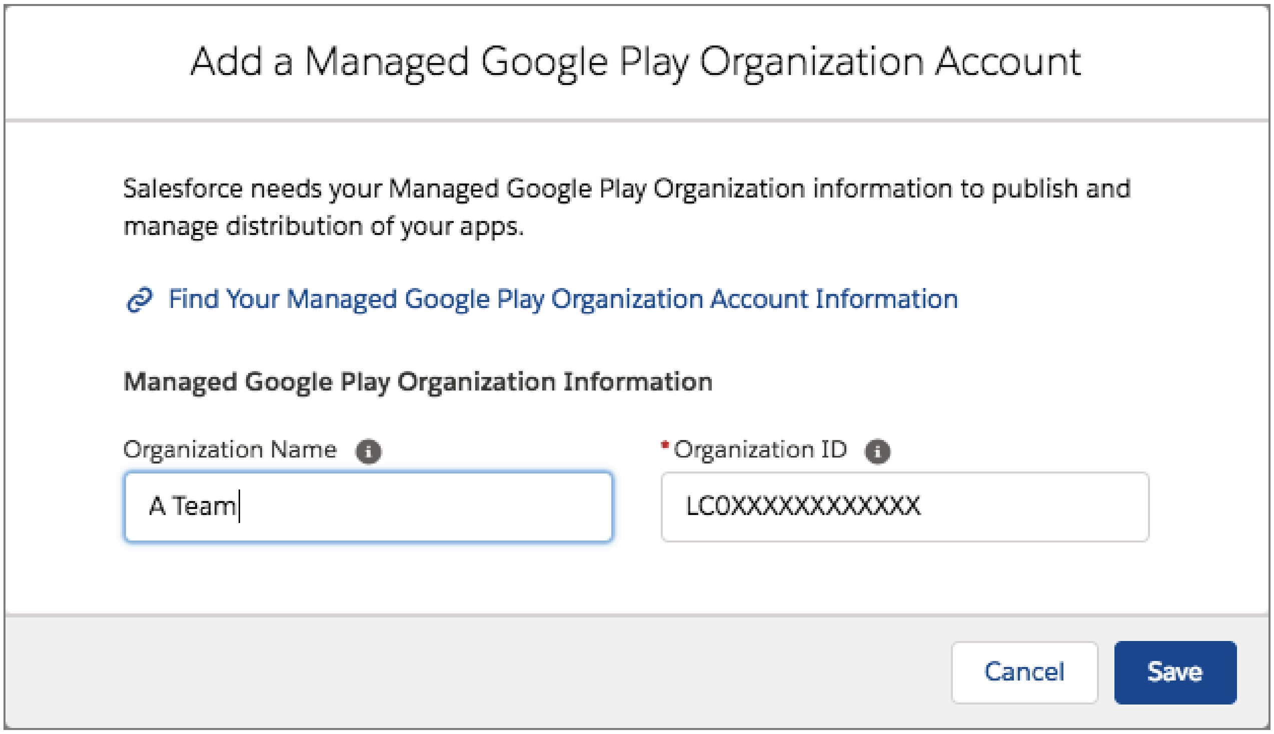
Name: Tom Smith

*Google Play Developer Account ID: 12345678909876543210

*Service Account Private Key

- Name is chosen by you and is for Salesforce internal use only (the name can't be changed after you save).
- Google Play Developer Account ID is the set of numbers you copied to your clipboard from the URL of your browser address bar.
- Service Account Private Key is the JSON file you saved to your computer from your Google Play Console.

3. Click **Save**.
4. Under Push Notifications, enter the information from your Google Firebase Console. See Register Your Firebase App to Enable Mobile Publisher Android Push Notifications for more information.
5. In the Private Distribution Information section, click **Add** to enter your Managed Google Play Organization information.



Add a Managed Google Play Organization Account

Salesforce needs your Managed Google Play Organization information to publish and manage distribution of your apps.

Find Your Managed Google Play Organization Account Information

Managed Google Play Organization Information

Organization Name: A Team

*Organization ID: LC0XXXXXXXXXXX

- Organization Name is the name you copied from your Google Play for Work account.
- Organization ID is the ID you copied from your Google Play for Work account that begins with LC0 or C0.

6. Click **Save**.

If you filled out the fields correctly, you’re ready to distribute your branded app to your users.

Set Up Delegated Private Distribution for iOS

Let’s get your branded app ready for distribution on the private Apple App store.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Publisher needs certain pieces of information from your Apple Developer account to distribute your branded app.

Note If you haven’t signed up for an Apple Developer account, click here to sign up.

Important Salesforce requires one Distribution Signing Certificate and one push token from your Apple Developer account to distribute and maintain your branded app. To participate in the Mobile Publisher program for iOS Delegate apps, you must have an Apple Developer Program account enrolled as an Organization. See Enrolling as an Organization at https://developer.apple.com/programs/enroll/.

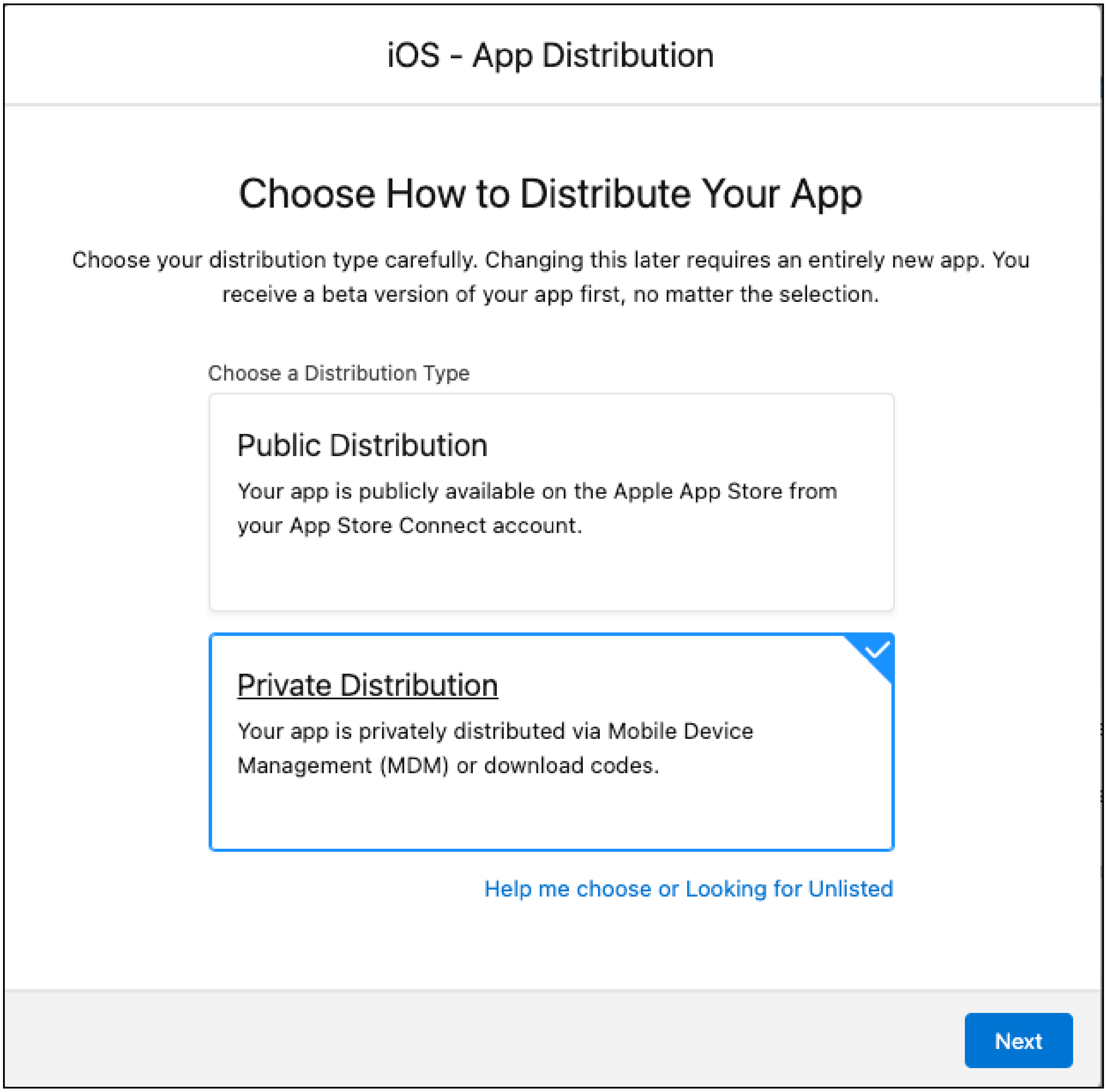
1. **Choose Distribution Type**
   First, we must choose the correct distribution type for your branded app.
2. **Register Salesforce Bundle ID**
   Register your Salesforce bundle ID with your Apple developer account.
3. **Find Your Apple Deployment Programs Account Info**
   Before you can distribute your branded app, add your Apple Development Programs Account credentials to Mobile Publisher.
4. **Create a Placeholder App with Privacy Questions and Age Ratings**
   Create a placeholder app in your App Store Connect account.
5. **Invite Salesforce to Your App Store Connect Account**
   To distribute your branded app, grant Salesforce with App Manager access to your App Store Connect.
6. **Record Your Apple Developer Team Account Info**
   Mobile Publisher needs your Apple developer team name and ID to distribute your branded app.
7. **Enter Your Apple Developer Information into Your Mobile Publisher Project**
   Now that we've done all our prep work, we can go back to your Mobile Publisher project and enter your Apple Developer Team Account information.

Choose Distribution Type

First, we must choose the correct distribution type for your branded app.

Watch the video: https://play.vidyard.com/ChnGsudDBC75AWfBd87MkA

1. Select **Private Distribution** from the App Distribution menu, and click **Next**.



iOS - App Distribution

Choose How to Distribute Your App

Choose your distribution type carefully. Changing this later requires an entirely new app. You receive a beta version of your app first, no matter the selection.

Choose a Distribution Type

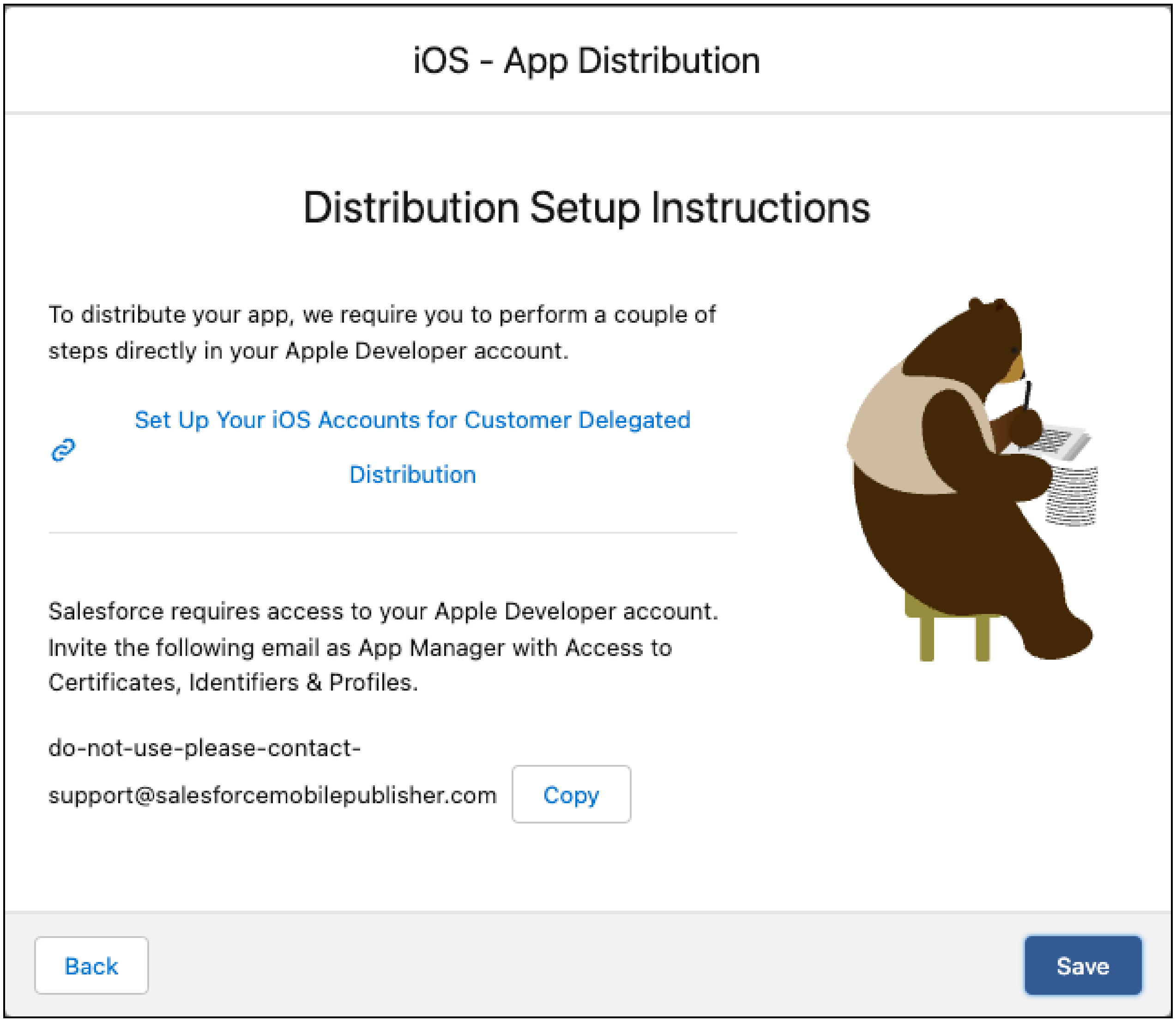
Public Distribution

Your app is publicly available on the Apple App Store from your App Store Connect account.

Private Distribution ✓

Your app is privately distributed via Mobile Device Management (MDM) or download codes.

2. Click **Copy** to copy and save the Salesforce Developer Email onto your computer for future reference.



iOS - App Distribution

Distribution Setup Instructions

To distribute your app, we require you to perform a couple of steps directly in your Apple Developer account.

Set Up Your iOS Accounts for Customer Delegated Distribution

Salesforce requires access to your Apple Developer account. Invite the following email as App Manager with Access to Certificates, Identifiers & Profiles.

do-not-use-please-contact-support@salesforcemobilepublisher.com

You're going to need the Salesforce Developer Email address when you log into your Apple Developer account to grant Salesforce access to distribute your app.

Register Salesforce Bundle ID

Register your Salesforce bundle ID with your Apple developer account.

Watch the video: https://play.vidyard.com/X4gnXRX2i7AxiaAu4PL7Gb

1. To register your Salesforce Bundle ID, go to https://developer.apple.com/account and log into your developer account.
2. Click **Certificates, IDs & Profiles**, then click **Identifiers**.
3. To register a new ID, click **+**, then select **App IDs**.
4. On the Select a type page, click **App**.
5. Enter `Salesforce Mobile Publisher` for the description.
6. Select the App ID prefix.
   a. If you see **Prefix Listed (Team ID)** without a dropdown in the App ID Prefix section, your App ID

Prefix is set.

b. If you see **Prefix Listed** in the App ID Prefix section with a dropdown to the right, select the App ID Prefix with **(Team ID)** listed next to it from the dropdown.

7. For the **Bundle ID**, enter the value of the `Store Identifier` field in your app's Setup for Mobile Publisher project.
8. Click **Continue**.
9. Click **Register**.
10. Click **Done**.

Find Your Apple Deployment Programs Account Info

Before you can distribute your branded app, add your Apple Development Programs Account credentials to Mobile Publisher.

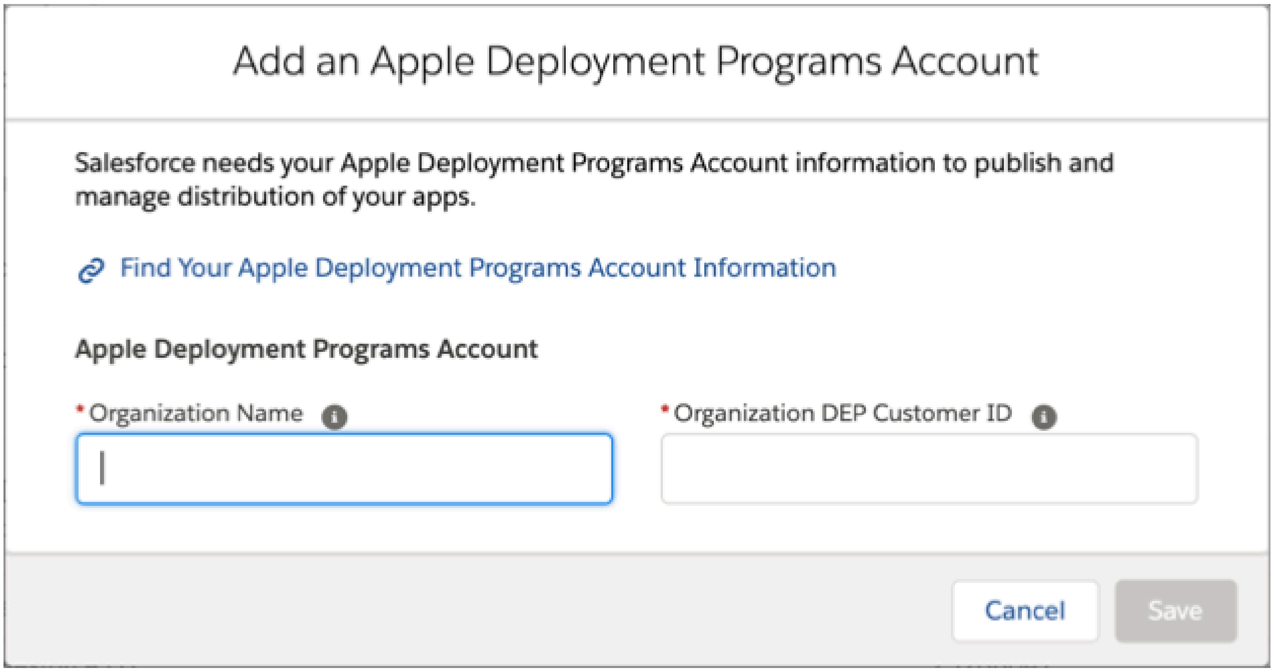
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important You need an Apple Business Manager (ABM) ID for each country that you want to privately distribute your app in. For example, if you want to distribute your app in India, you must have a valid Apple Business Manager (ABM) ID in that country.

1. In the Private Distribution Information section, click **+ Add** to enter your Apple Deployment Programs Account.



Add an Apple Deployment Programs Account

Salesforce needs your Apple Deployment Programs Account information to publish and manage distribution of your apps.

Find Your Apple Deployment Programs Account Information

Apple Deployment Programs Account

- Organization Name is the registered name of your Apple Business Manager.
- Organization DEP Customer ID is the Organization ID listed in your Apple Business Manager. Sometimes the Organization DEP Customer ID is referred to as Organization ID.

2. To find the Organization Name and Organization DEP Customer ID:
   a. Go to https://business.apple.com and log into your Apple Business Manager account as a user with the Administrator role.
   b. Click your name in the bottom-left sidebar.
   c. Navigate to Preferences and select **Enrollment Information**.
   d. Under the Organization Info section, locate your Organization Name and Organization DEP Customer ID (Organization ID).
3. Enter your Organization Name and Organization DEP Customer ID in the Add an Apple Deployment Programs Account page.
4. Save your changes.

Create a Placeholder App with Privacy Questions and Age Ratings

Create a placeholder app in your App Store Connect account.

Now, let's create a placeholder app in your App Store Connect account. Complete these steps after logging into https://appstoreconnect.apple.com.

Watch the video: https://play.vidyard.com/R5Bck1mjCmHNM2oo1RhbhD

Important Apple has introduced new requirements for some content rights information, including more granular age ratings and new ratings questions. You must update your age ratings and answer the new questions by January 26, 2026, in order to continue submitting app updates. These requirements are also enforced for apps that are privately distributed.

1. Click **My Apps**.
2. Click **+**, then **New App**.
3. Set the platform to iOS, and fill out the required fields:
   a. Fill out the App Name (must match the name entered in the `App Store Name` field in the metadata form).
   b. Set the Primary Language to **English (U.S.)**.
   c. Select the value in the `Store Identifier` field presented in Mobile Publisher as the Bundle ID.
   d. Enter **Salesforce-MM-DD-YYYY** for the SKU.
   An example for **Salesforce-MM-DD-YYYY** is Salesforce-07-01-2020.
4. Select **Full Access** for the User Access.
5. Click **Create**.
6. Select the placeholder app you created.
7. In the sidebar, select **App Privacy**.
8. Click **Get Started**.
9. Specify if you or your third-party partners collect data from your app.

   For Experience Cloud app, see App Privacy - Experience Cloud App to learn about the baseline disclosures regarding your Experience Cloud Mobile Publisher iOS app.

   For Salesforce app, see App Privacy - Salesforce App to learn about the baseline disclosures regarding your Salesforce Mobile Publisher iOS app.

10. In the Distribution tab, select the App Information section.
11. Select **Edit** next to **App Age Ratings** and answer all the seven steps listed. Generally, the default answer is No to most of the questions, but your app's needs can be different. We recommend you pay special attention to the Capabilities and the Wellness or Medical sections.
12. Navigate to the TestFlight tab.
13. Select **Test Information**.
14. Navigate to Beta App Review Information.
15. In the Sign-In Information section, enter your test or demo user credentials.
    a. The test or demo user credentials must be able to log in to the sandbox Experience Cloud site URL you submitted.
       - If you're providing user credentials for your production environment, enter "PRODUCTION" in the Review Notes field in App Store Connect.
       - If you're providing user credentials for your sandbox environment, enter "SANDBOX" in the Review Notes field in App Store Connect.
    b. Disable two-factor authentication for this user. You can specify a range of IP addresses for this particular user. To specify the IP address range, clone the Salesforce user profile for the Apple Reviewer test user, and add the login IP range from 1.1.1.1 to 255.255.255.255 for that profile. Defining the IP range limits the change to only to this user without affecting other users.
    See Restrict Login IP Ranges in Profiles for more information on adding a range of IP addresses.

    With every Mobile Publisher for Experience Cloud app update, Apple attempts to log in to the app as part of the approval process. Make sure the user credentials are always valid to prevent delays in app delivery.

16. Click **Save**.

Invite Salesforce to Your App Store Connect Account

To distribute your branded app, grant Salesforce with App Manager access to your App Store Connect.

Complete these steps after logging into https://appstoreconnect.apple.com.

Watch the video: https://play.vidyard.com/ZZHqmoDTRfBorTNZJpQHW5

1. Navigate to Users and Access.
2. To add a user, click **+**.
3. Fill out these fields:
   a. Enter `Publisher` for the First Name.
   b. Enter `Salesforce` for the Last Name.
   c. Enter the Salesforce Developer Email address for Email.

      Note Salesforce provides the Salesforce Developer Email address when you select an app distribution type. To view the email address, in Salesforce, go to **Setup | Mobile Publisher** and open your iOS or Android draft. In the App Distribution section, click the **Choose App Distribution Type** link. Select **Public** or **Private** (you can change this selection before you submit the draft) and click **Next**. The App Distribution dialog displays the email address beside the Copy button.
4. Select the **App Manager** role.
5. Check the **Access to Certificates, Identifiers & Profiles** check box.
6. Under Apps, select the Salesforce Mobile Publisher app you created earlier.
7. Click **Invite**.

Record Your Apple Developer Team Account Info

Mobile Publisher needs your Apple developer team name and ID to distribute your branded app.

Important When prompted to "record" a value or ID, be sure to note the value or ID exactly as written, and save it for later. These values and IDs are needed for the final step of setting up public or private distribution for iOS.

Watch the video: https://play.vidyard.com/d8zyxUAEhDCpvbndRV2PFS

1. Go to https://developer.apple.com/account and log in to your Apple Developer account.
2. Navigate to the **Membership** page.
3. Record the value in the **Team Name** field for use in a later step.
4. Go to https://appstoreconnect.apple.com/login and log in to your App Store account.
5. Navigate to the **Users and Access** page.
6. Select the **Integrations** tab.
7. Select **App Store Connect API**, and then select the **Team Keys** tab. The Issuer ID appears. Record the Issuer ID for use in a later step

8. Create an API key.
   a. Click **+**.
   b. Name the key Mobile Publisher.
   c. In the Access field, select **App Manager**.
9. The newly generated key is called the Apple Private Key. Record the Key ID for use in a later step.
10. Refresh the webpage.
11. Hover over the **Apple Private Key** and download it. This key can only be downloaded at this time.

Enter Your Apple Developer Information into Your Mobile Publisher Project

Now that we've done all our prep work, we can go back to your Mobile Publisher project and enter your Apple Developer Team Account information.

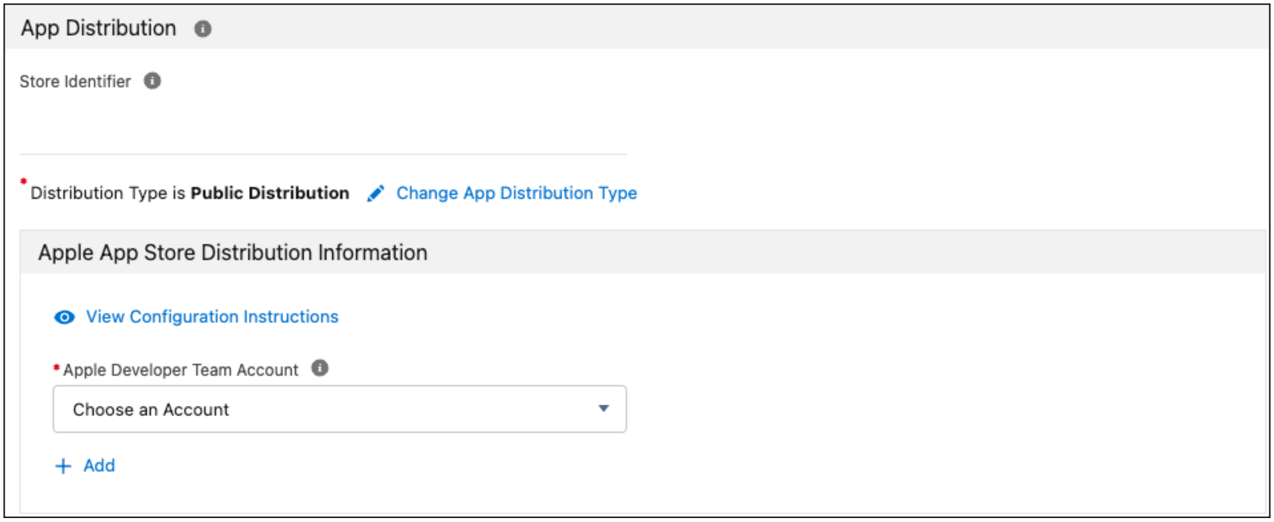
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Watch the video: https://play.vidyard.com/d8zyxUAEhDCpvbndRV2PFS

1. In the Mobile Publisher, go to the App Distribution section.



App Distribution

Store Identifier

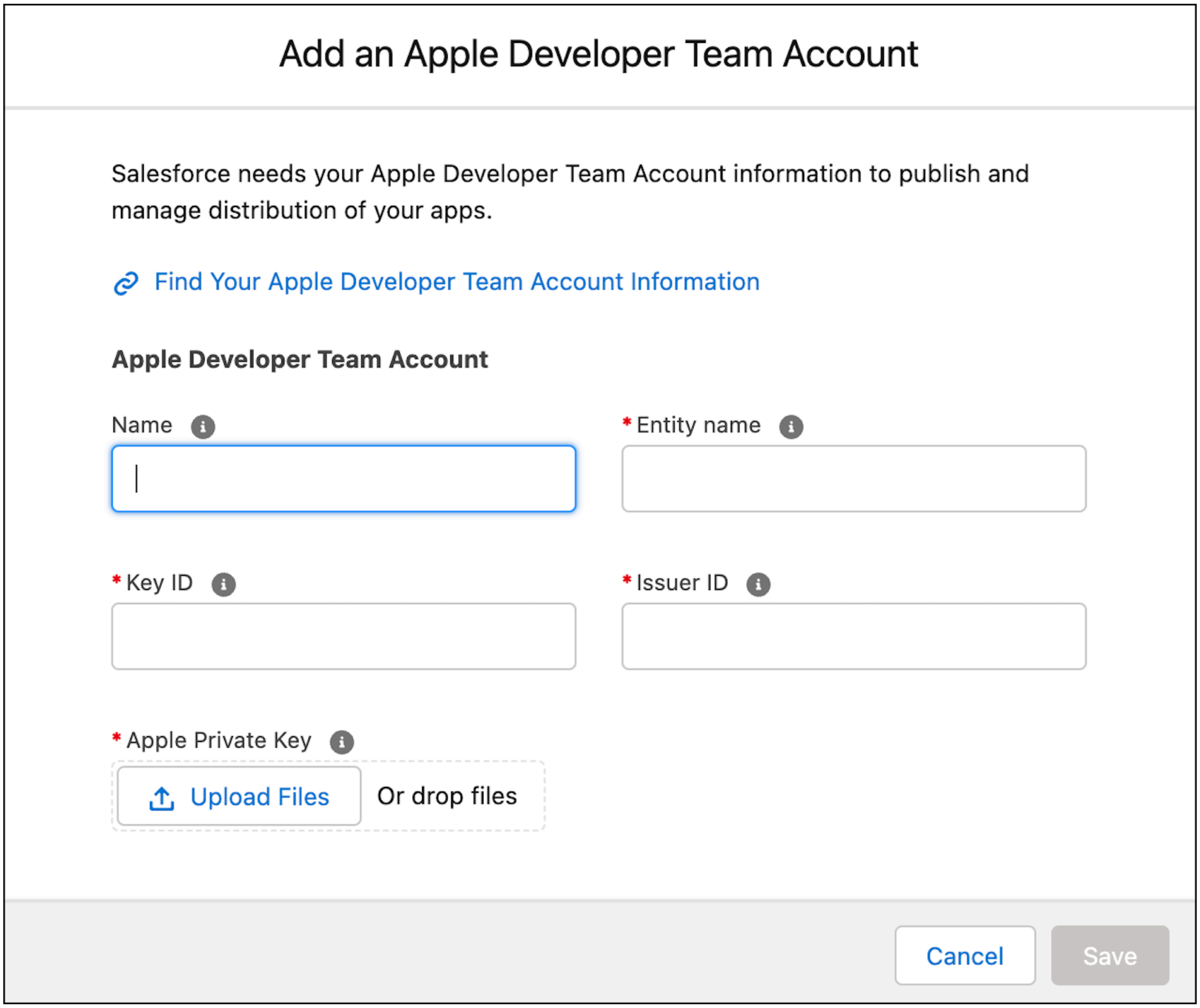
*Distribution Type is **Public Distribution**

Apple App Store Distribution Information

*Apple Developer Team Account

Choose an Account

2. To enter your Apple Developer Team Account information, click **Add**.



Add an Apple Developer Team Account

Salesforce needs your Apple Developer Team Account information to publish and manage distribution of your apps.

- Name is chosen by you for Salesforce internal use only to differentiate between multiple accounts.
- Team Name is the name from your Apple Developer Account.
- Issuer ID is the ID from your App Store Connect.
- Key ID is the ID from your App Store Connect.

3. Click **Save**.

If you filled out the fields correctly, you’re ready to distribute your branded app to your users.

Unlisted Option for iOS App Distribution

Learn about unlisted app distribution, an alternative to the traditional public and private app distribution options. Unlisted app distribution is sometimes appropriate for apps that were rejected from App Store public distribution.

Apple provides an unlisted app distribution option for app developers to distribute apps through the App Store that aren’t suitable for broad public consumption. With unlisted app distribution, you distribute your app to an audience of your choosing through a direct URL link. Your unlisted app isn’t searchable in the App Store, nor is it visible in any App Store categories or listings. Keep in mind that anyone with the download link can download your app. Consider this information carefully if it’s important that your app is available only to members of your organization.

If Apple rejected your organization’s public iOS app because of section 3.2 of Apple’s Review Guidelines, the unlisted app distribution option is worth considering. You provide your full-featured application to your target iOS users, without worrying about making your app relevant to the broader public or needing additional expertise in managing a private distribution.

Note The unlisted option can’t be selected as your distribution type in the Mobile Publisher project setup. Apple only provides public and private options for initial app submissions, and the Mobile Publisher UI reflects these two options.

Unlisted vs. Private Distribution

To learn more about the many differences between public and private distribution, see App Distribution Type Considerations. In this section, we list some advantages of the unlisted distribution option compared to private distribution.

- Less overhead: Private distribution requires an Apple Business Manager (ABM) ID for every country that you want to privately distribute your app in. Unlisted distribution has no such requirement.
- Simple to change from public to unlisted: When you have an approved private app on the App Store, it’s difficult to convert it back to public or unlisted distribution. Switching from a public distribution to unlisted is simpler.
- Ease of use: Private distribution requires use of ABM, setting up MDM policies, and managed Apple IDs. Unlisted apps are accessed through a simple and shareable link, and your users can easily download your app onto their iOS device.

Distribute My App as Unlisted

To distribute your app as unlisted, complete your app testing and handle any outstanding rejections raised by the Apple reviewer.

After Apple rejects the final app submission for 3.2 guidelines, and Mobile Publisher confirms that it’s OK to submit the request for unlisted, you can submit a request to Apple to submit your app as unlisted (Apple Developer sign-in required).

For more information about Apple’s distribution options, including the unlisted option, refer to this Explore unlisted app distribution video.

See Also

Unlisted App Distribution (Apple)

Set Up a New External Client App (ECA) in Mobile Publisher

Start setting up an external client app (ECA) for your mobile app after completing the Set Up the Mobile

App step in Mobile Publisher. Don’t create your external client app in the External Client App manager (accessed through Setup). Create your ECA in Mobile Publisher instead.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important Important: If you are creating an ECA to test it in a sandbox, don’t create the ECA directly in the sandbox. Create the ECA in the production environment first, and then deploy it to your sandbox.

Click **Start** to set up the External Client App, and fill in the fields.

1. Name
2. Description
3. Contact Email
4. Contact Phone
5. Icon (PNG file no larger than 100 KB. The image must be 128x128 pixels, and have a 1:1 aspect ratio.)
6. Push Notifications. If enabled, add the following fields for iOS:
   - Authentication for Push Notifications
   - Signing Key or Certificate
   - Key ID (ID of your App Store Connect API key)
   - Team ID (Team ID listed in your company’s Apple Developer account)
7. Push Notifications. If enabled, for Android upload the Firebase Admin SDK private key from the Google Firebase project associated with your app.
8. Click **Save** and **Deploy**. Select a target for the ECA deployment: production or sandbox. The first deployment must be to the production org. After the ECA is deployed in production, you can then deploy to a sandbox org for testing purposes.
9. After you select a target, the Deployed To field shows where the ECA version is available. A Live status means that the app is live in production. The Deployed status, when coupled with a sandbox entry in the Deployed To column, indicates that the app is deployed in a sandbox environment.

Note

- If you create a new sandbox, or your existing sandbox is refreshed after the ECA is deployed to the production org, the sandboxes get the same version of the ECA as the production org. You don’t need to explicitly deploy the ECA in the sandbox.
- If you need to change the name or icon of your ECA, you must create a new version and redeploy

it.

Manage External Client App Versions in Mobile Publisher
After you set up your external client app, you can see its settings in the External Client App Settings tab.

Manage External Client App Versions in Mobile Publisher

After you set up your external client app, you can see its settings in the External Client App Settings tab.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

1. Click **View** to see the ECA's detail page and version history. The version history includes where the ECA is deployed.
2. After an ECA is deployed, you can't edit it. Click **+ New Version** to create and deploy a new version of the app. If a draft version already exists, this card will be replaced with a link to your draft.
3. Clicking **+ New Version** takes you back to the ECA setup form, pre-filled with the details from your last version. The version number is automatically updated.
4. To view older versions, click **View more**.

Note You can only have one ECA version in an org.

Change Distribution Types

Consider the impact of changing the distribution type for your branded app.

You can change the distribution type for your branded app. If you do change your distribution type, make sure that the right distribution type matches your use case. We recommend that you refer to Apple and Google's app distribution guidelines.

Change Distribution Type for iOS
You can change the distribution type for your iOS app.

Change Distribution Types for Android

You can change the distribution type for your Android app.

Change Distribution Type for iOS

You can change the distribution type for your iOS app.

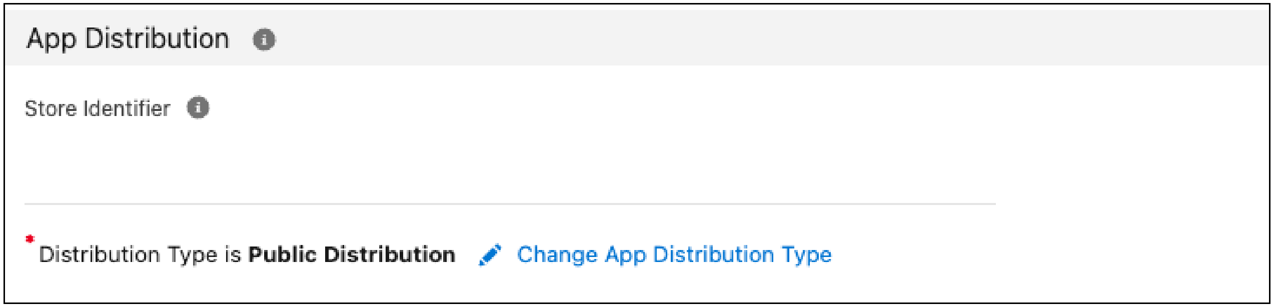
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important Changing the distribution type for iOS requires a new managed package to be installed and updates to the Bundle ID. If you must change the distribution type for your app, it's best to do so before your app is live. You can't use the same app name for any app that's live that you want to change the distribution type for. See App Distribution Type Considerations before changing the distribution type for your app.

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Choose the app you want to change the distribution type for.
3. Go to the App Distribution section, and select **Change App Distribution Type**.



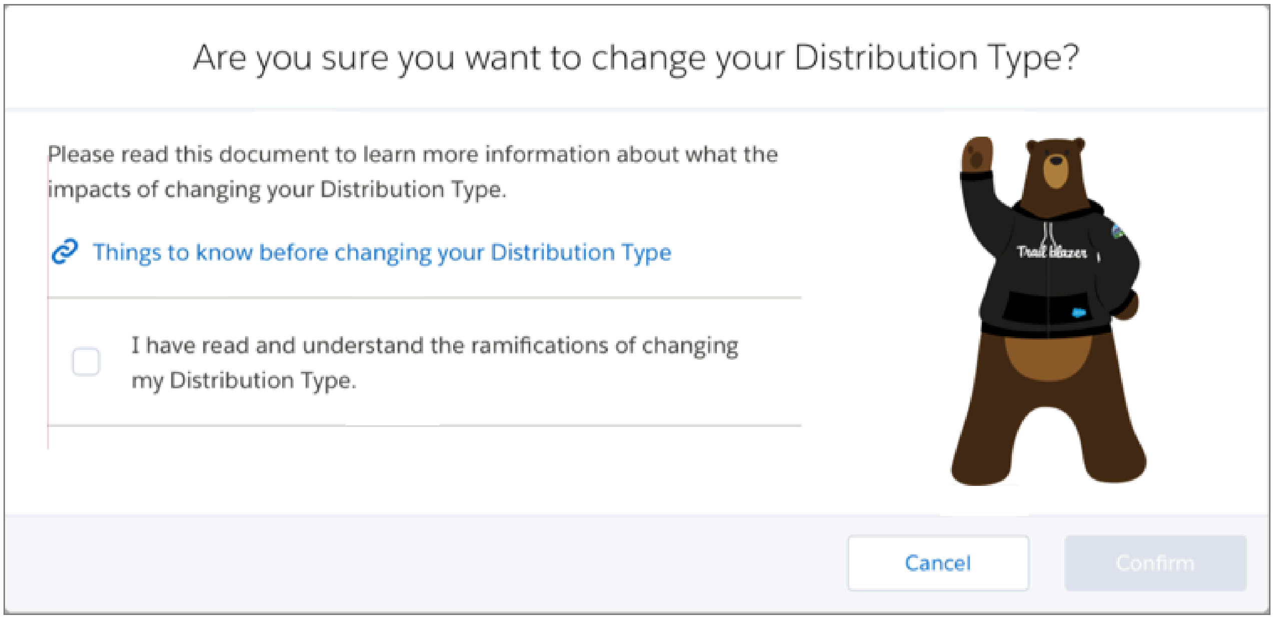
App Distribution

Store Identifier

* Distribution Type is **Public Distribution**

Note You can change the distribution type for your existing iOS app only if the app isn't live. If you want to change the distribution type of your live iOS app, create and submit a new app. Then, ask users to uninstall the existing app and install the new app.

4. Read about the impact of changing your distribution type and click **Confirm**.



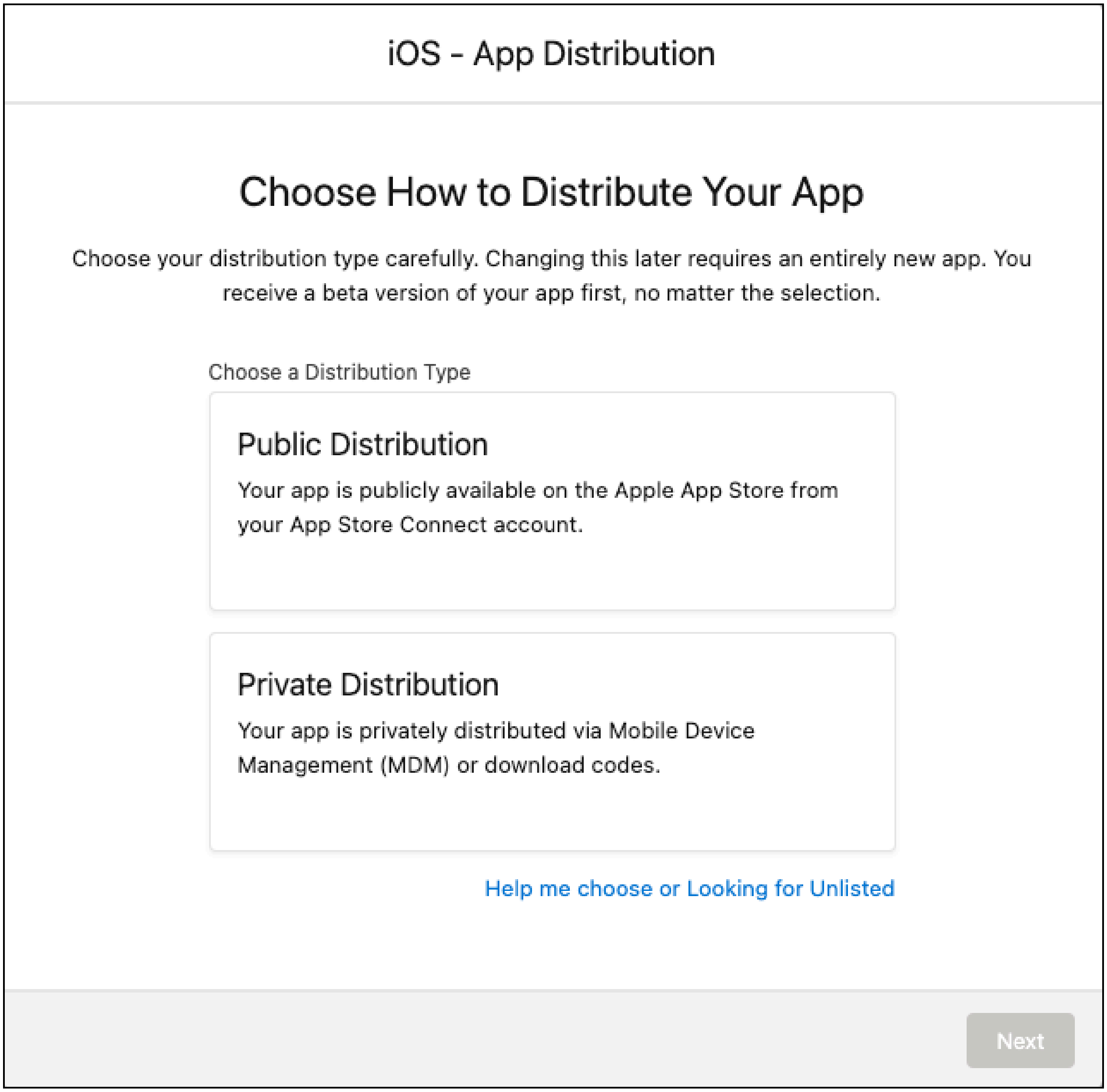
Are you sure you want to change your Distribution Type?

Please read this document to learn more information about what the impacts of changing your Distribution Type.

Things to know before changing your Distribution Type

□ I have read and understand the ramifications of changing my Distribution Type.

5. Choose your new distribution type and click **Save**.



iOS - App Distribution

Choose How to Distribute Your App

Choose your distribution type carefully. Changing this later requires an entirely new app. You receive a beta version of your app first, no matter the selection.

Choose a Distribution Type

Public Distribution

Your app is publicly available on the Apple App Store from your App Store Connect account.

Private Distribution

Your app is privately distributed via Mobile Device Management (MDM) or download codes.

Change Distribution Types for Android

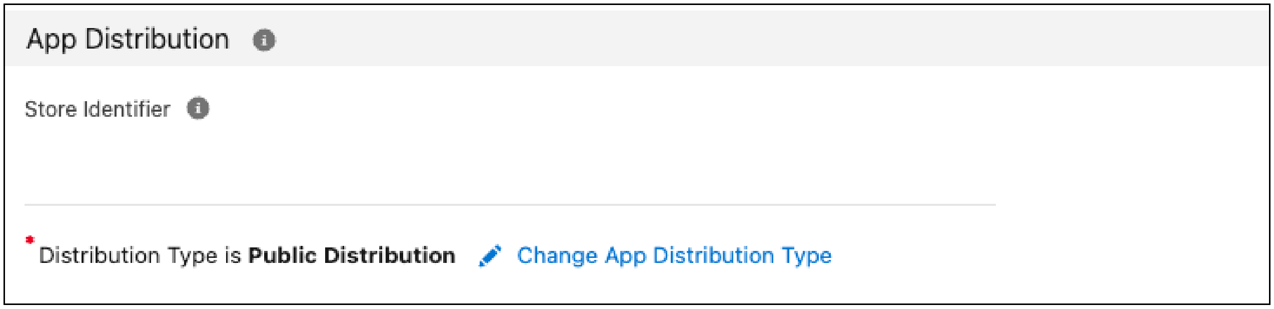
You can change the distribution type for your Android app.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Choose the app you want to change the distribution type for.
3. Go to the App Distribution section, and select **Change App Distribution Type**.

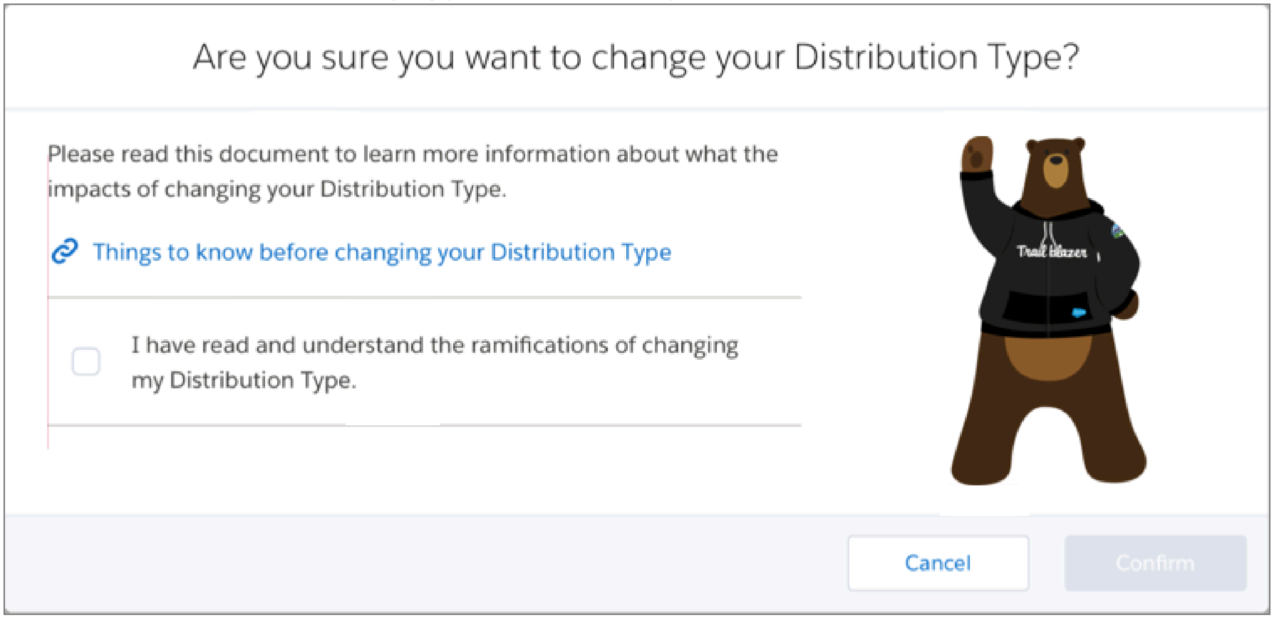


App Distribution

Store Identifier

* Distribution Type is **Public Distribution** Change App Distribution Type

4. Read about the impact of changing your distribution type and click **Confirm**.



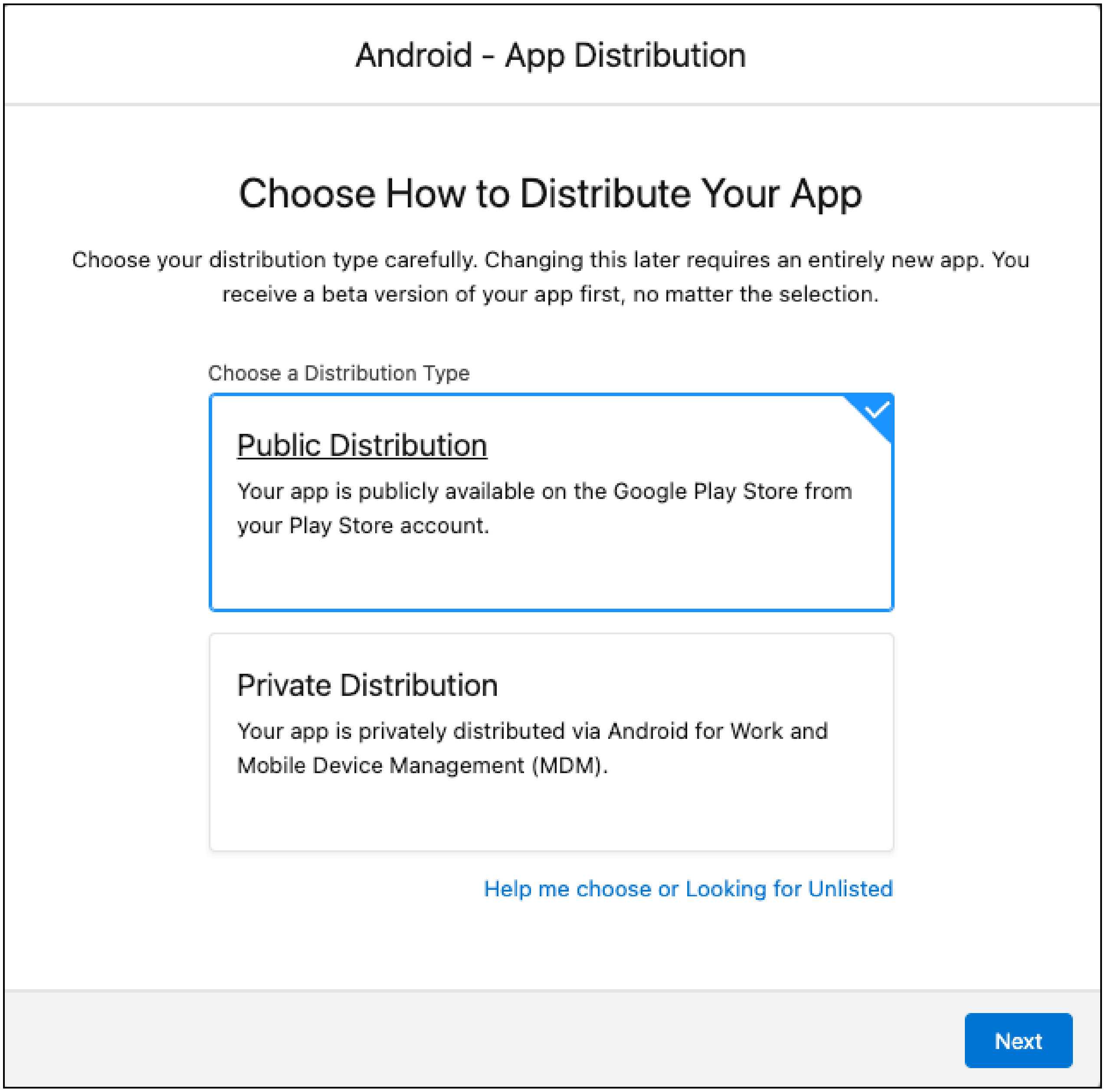
Are you sure you want to change your Distribution Type?

Please read this document to learn more information about what the impacts of changing your Distribution Type.

Things to know before changing your Distribution Type

□ I have read and understand the ramifications of changing my Distribution Type.

5. Choose your new distribution type and click **Save**.



Android - App Distribution

Choose How to Distribute Your App

Choose your distribution type carefully. Changing this later requires an entirely new app. You receive a beta version of your app first, no matter the selection.

Choose a Distribution Type

☑ Public Distribution

Your app is publicly available on the Google Play Store from your Play Store account.

Private Distribution

Your app is privately distributed via Android for Work and Mobile Device Management (MDM).

Change Distribution Account

You may have to change your distribution account if you need to change the developer account associated with your distribution type.

This can also happen if there are errors with your developer account details.

Changing your distribution account requires a change in account ownership. The Salesforce support team will reach out to you with any required information from either Apple or Google.

Change Distribution Account for iOS
Here’s how to change your distribution account for iOS.

Change Distribution Account for Android

Here’s how to change your distribution account for Android.

Change Distribution Account for iOS

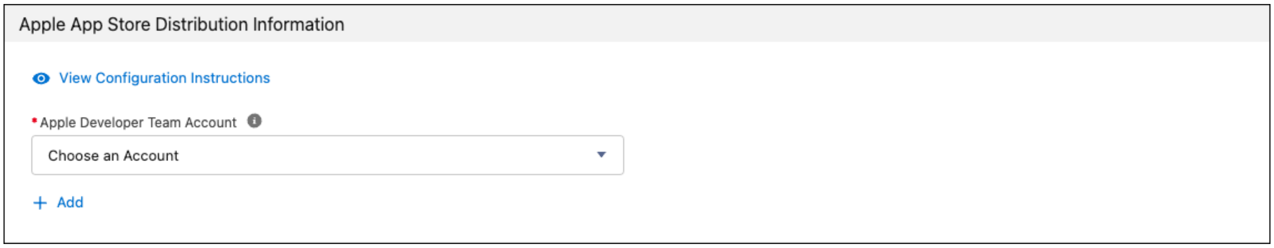
Here’s how to change your distribution account for iOS.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Choose the app you want to change the distribution type for.
3. Go to the Customer Distribution Information section, and click **Add**.

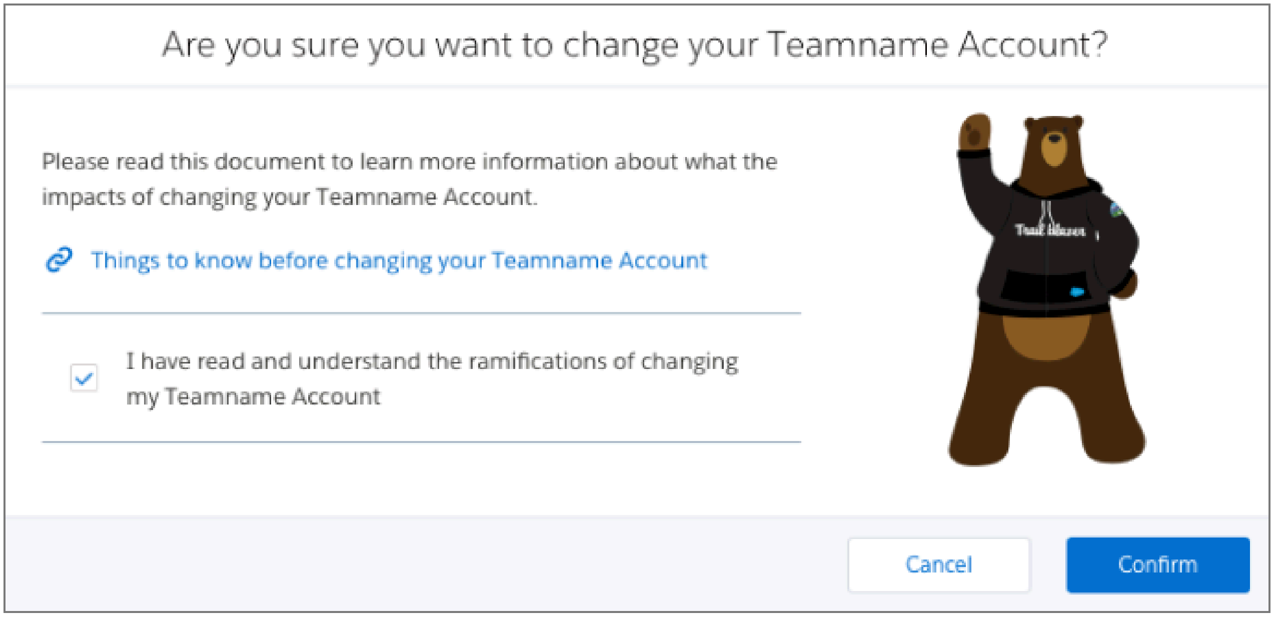


Apple App Store Distribution Information

View Configuration Instructions

*Apple Developer Team Account: Choose an Account

4. Read about the impact of changing your account and click **Confirm**.



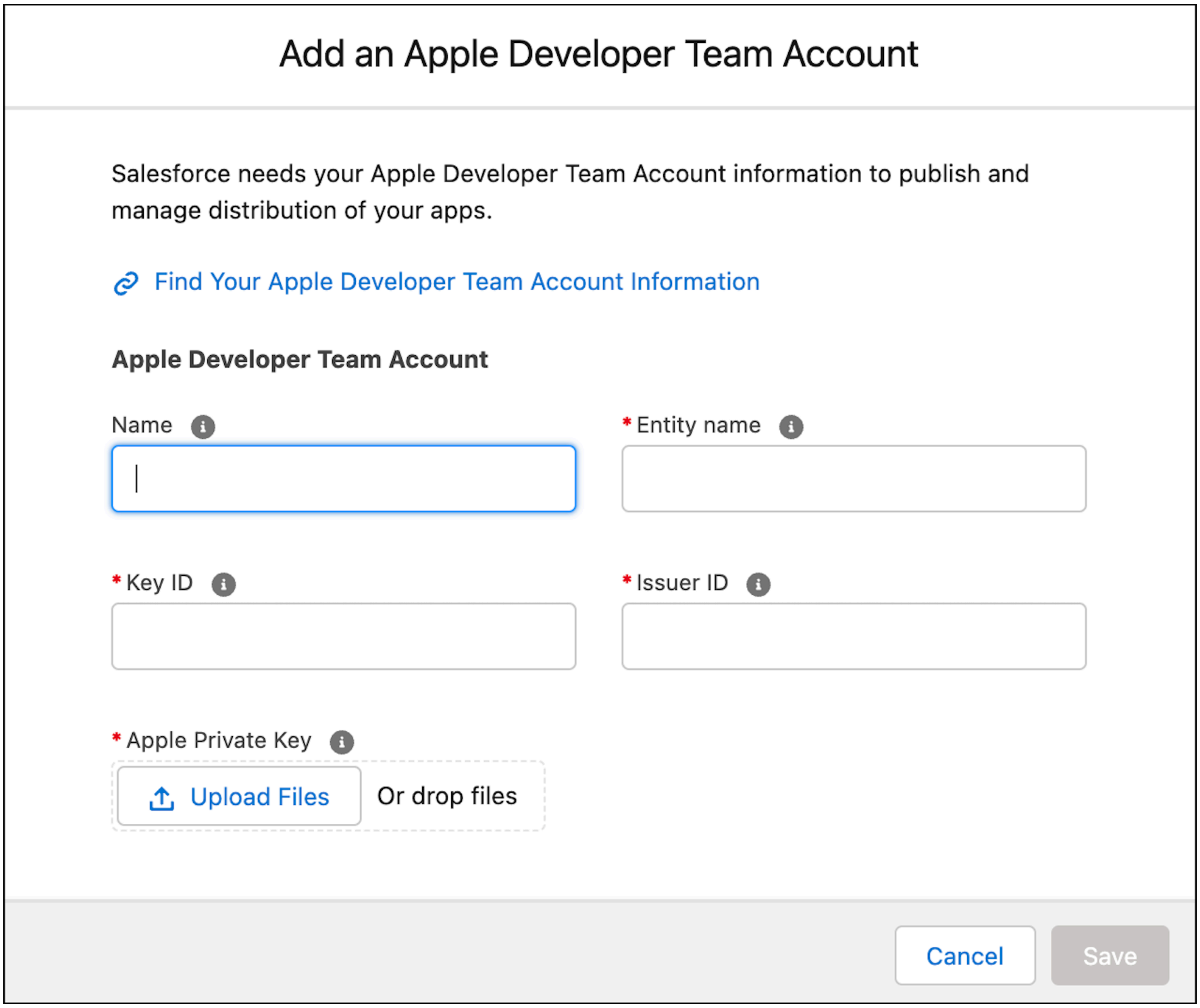
Are you sure you want to change your Teamname Account?

Please read this document to learn more information about what the impacts of changing your Teamname Account.

Things to know before changing your Teamname Account

☑ I have read and understand the ramifications of changing my Teamname Account

5. Add your new Apple Developer Team Account.



Add an Apple Developer Team Account

Salesforce needs your Apple Developer Team Account information to publish and manage distribution of your apps.

Find Your Apple Developer Team Account Information

Apple Developer Team Account

Name

* Entity name

* Key ID

* Issuer ID

* Apple Private Key

- Team Name is the name from your Apple Developer Account.
- Team ID is the ID from your Apple Developer Account.

6. Click **Save**.

The new Apple Developer Team Account is selected by default.

Change Distribution Account for Android

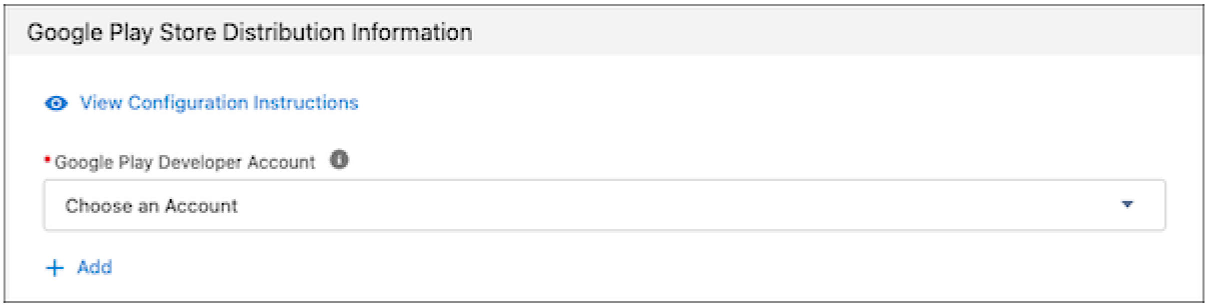
Here's how to change your distribution account for Android.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

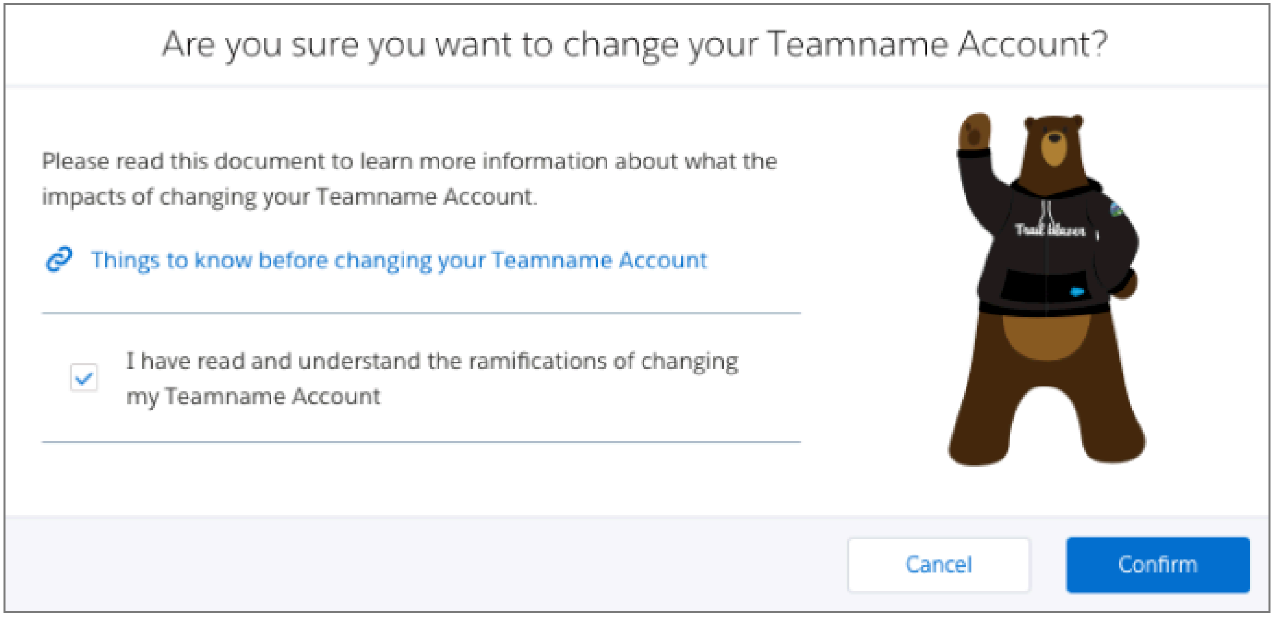
1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Choose the app you want to change the distribution type for.
3. Go to the Customer Distribution Information section, and click **Add**.



Google Play Store Distribution Information

* Google Play Developer Account

4. Read about the impact of changing your account and click **Confirm**.

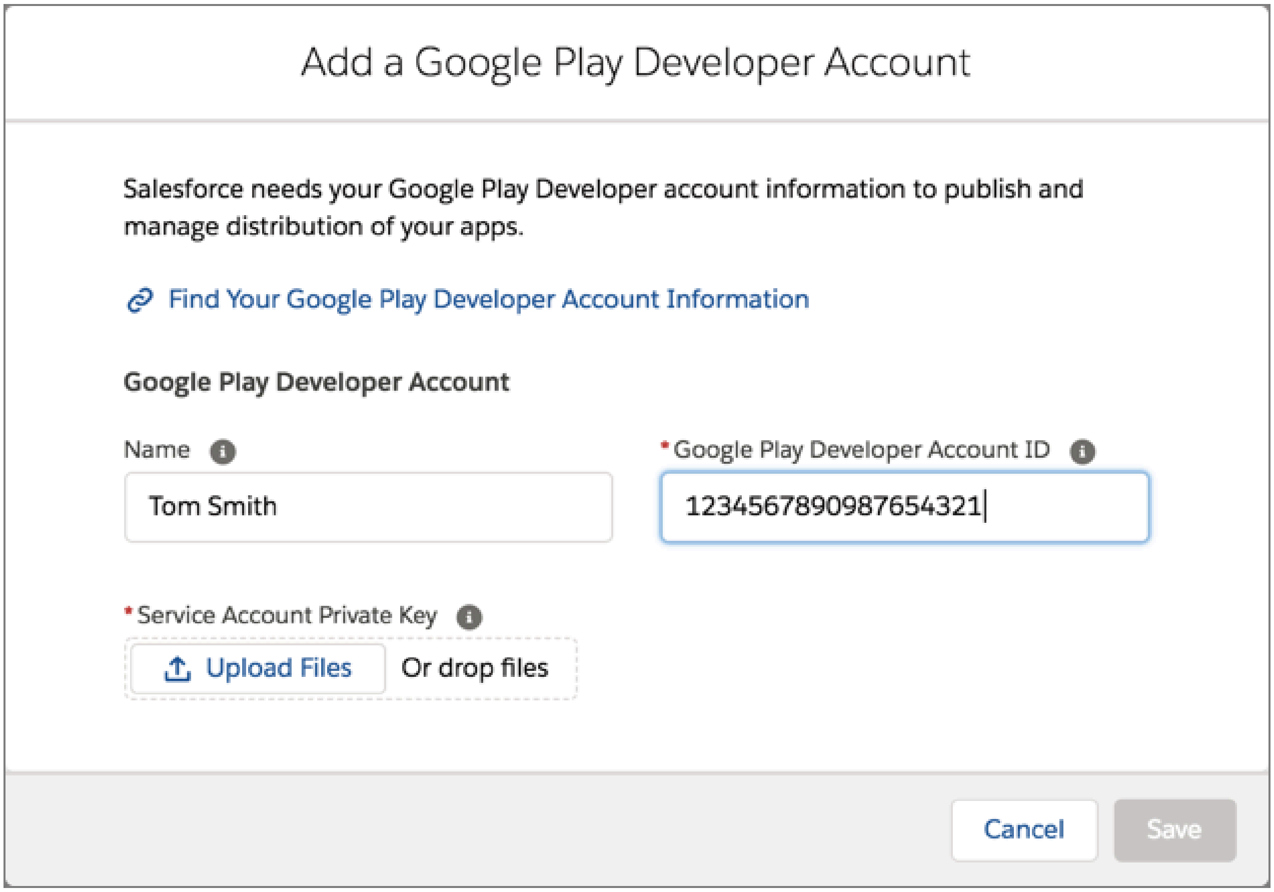


Are you sure you want to change your Teamname Account?

Please read this document to learn more information about what the impacts of changing your Teamname Account.

☑ I have read and understand the ramifications of changing my Teamname Account

5. Add your new Google Play Developer Account.



Add a Google Play Developer Account

Salesforce needs your Google Play Developer account information to publish and manage distribution of your apps.

Find Your Google Play Developer Account Information

Google Play Developer Account

Name: Tom Smith

*Google Play Developer Account ID: 12345678909876543211

*Service Account Private Key

- Name is chosen by you and is for Salesforce internal use only (the name can’t be changed after you save).
- Google Play Developer Account ID is the set of numbers you copied to your clipboard from the URL of your browser address bar.
- Service Account Private Key is the JSON file you saved to your computer from your Google Play Console.

6. Click **Save**.

The new Google Play Developer Account is selected by default.

Build Your Branded App

Create a Mobile Publisher project, enter details about your app, and upload your custom branded assets to Salesforce.

Create a Mobile Publisher Project for Salesforce App
Start a Mobile Publisher project for Salesforce app and manage both Android and iOS branded apps.

Enable Biometric ID App Unlock for iOS and Android
When using Mobile Publisher, you can apply biometric ID credentials to make your environment more secure. To use biometric ID credentials, your device must have a device passcode or PIN enabled and FaceID (for iOS) or Android Security and Privacy | Biometrics.

Pre-Authorize User App Access Through External Client App Policies
Configure who can use your app by defining which users are pre-authorized. Users who are pre-authorized can bypass the Allow or Deny permission pop-up when accessing the app.

Push Notifications
Keep your users in the know with timely notifications sent to their mobile devices. Create custom push notifications using Notification Builder or use Marketing Cloud Engagement tools like Automation Studio or Journey Builder.

Enable Cross-Website Tracking (iOS)
With iOS 14, Apple placed additional tracking restrictions on third-party applications. These tracking restrictions can prevent your custom web content from appearing in your branded app.

Create a Mobile Publisher Project for Salesforce App

Start a Mobile Publisher project for Salesforce app and manage both Android and iOS branded apps.

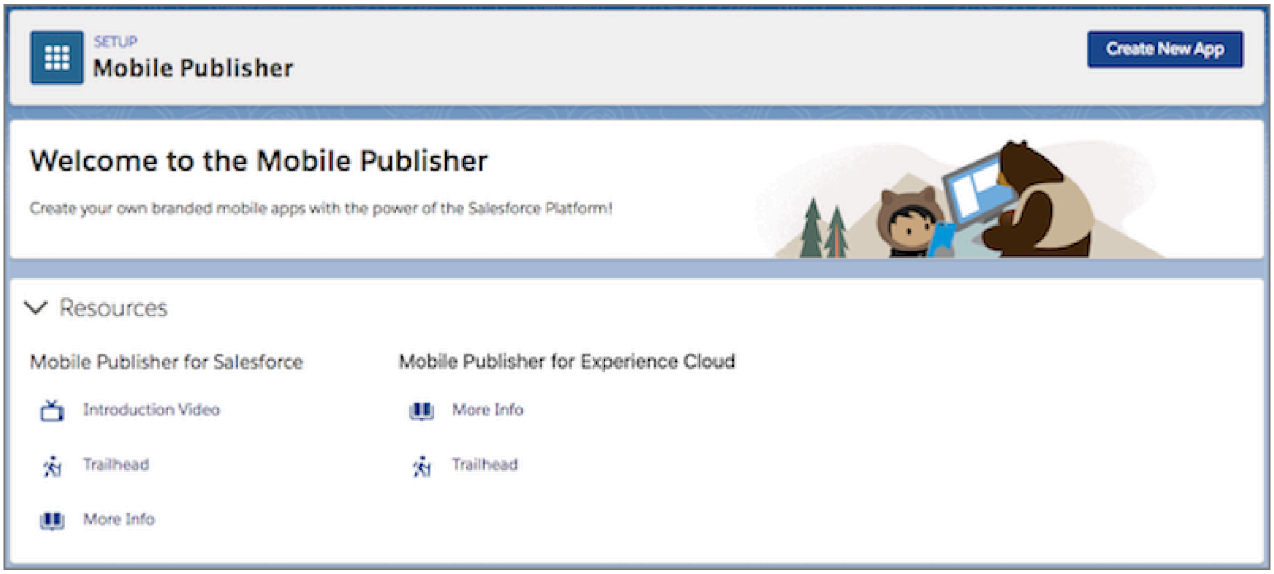
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Note Your organization must license Salesforce Mobile Publisher in order to use the feature. Contact your Salesforce sales rep for more information.

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.

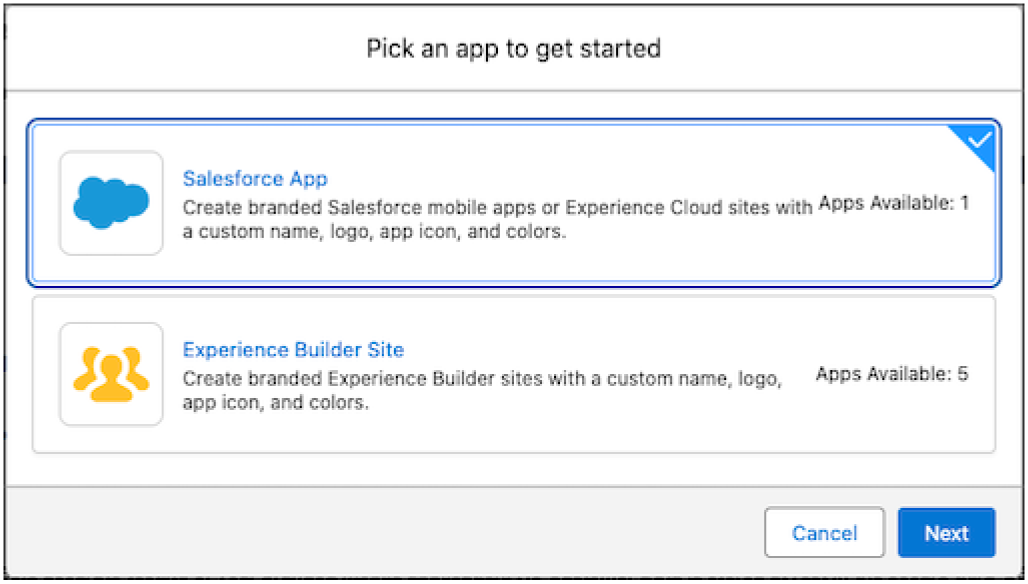


Welcome to the Mobile Publisher

Create your own branded mobile apps with the power of the Salesforce Platform!

Resources
Mobile Publisher for Salesforce
Introduction Video
Trailhead
More Info
Mobile Publisher for Experience Cloud
More Info
Trailhead

2. Click **Create New App**.
3. Choose the Salesforce mobile app to brand (only one app can be branded per org).

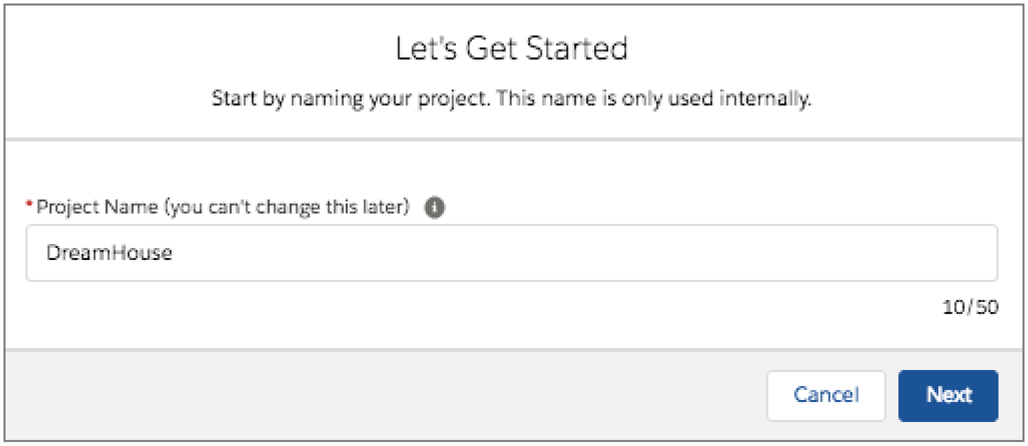


Pick an app to get started

Salesforce App
Create branded Salesforce mobile apps or Experience Cloud sites with a custom name, logo, app icon, and colors. Apps Available: 1

Experience Builder Site
Create branded Experience Builder sites with a custom name, logo, app icon, and colors. Apps Available: 5

4. Enter a name for the Mobile Publisher project. You can't edit the project name after you set it. This name is for internal reference only; it doesn't display publicly in Google Play or the App Store.



Let's Get Started

Start by naming your project. This name is only used internally.

*Project Name (you can't change this later)
DreamHouse
10/50

5. Click **Next**.

When you're done creating the Mobile Publisher project, you can manage the Android and iOS apps that are part of your project.

Each application store–Google Play and the App Store–requires a different set of information in order to publish an app.

Enter Information and Upload Assets for Your App
Customize your app's appearance, including app icon, loading page logo, and header background color, and much more so that the app matches your company's branding.

Enter Information and Upload Assets for Your App

Customize your app's appearance, including app icon, loading page logo, and header background color, and much more so that the app matches your company's branding.

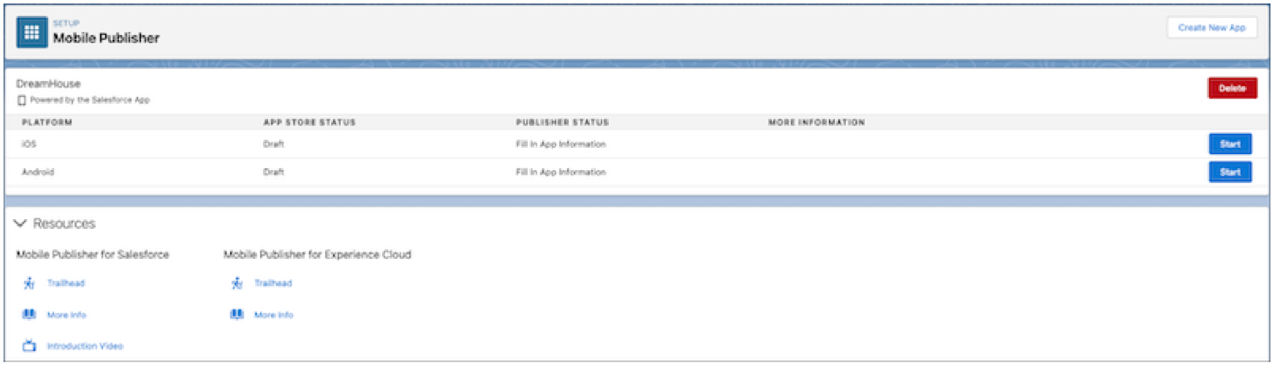
REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Note Mobile Publisher isn't available in Salesforce Setup until your organization licenses the feature. Contact your Salesforce sales rep for more information.

1. On the Mobile Publisher page, click **Start** for the iOS app.



PLATFORM	APP STORE STATUS	PUBLISHER STATUS	MORE INFORMATION	
iOS	Draft	Fill In App Information		Start
Android	Draft	Fill In App Information		Start

Resources
Mobile Publisher for Salesforce
Trailhead
More Info
Introduction Video
Mobile Publisher for Experience Cloud
Trailhead
More Info

2. Fill in all the necessary fields. For more information about the app distribution choices, see Set Up Distribution.
3. If you're confused about a certain field, hover your cursor over the info bubble to see helpful tips and guidance.
4. When you're done entering all the information, click **Submit**.
5. Click **Submit** again to confirm that you want to submit the form.
   The submission goes to the Mobile Publisher team for review.

The app is now a draft, and you can see the status of the app on the iOS page.

If your company wants an Android version of the app, repeat the same steps for Android.

Enable Biometric ID App Unlock for iOS and Android

When using Mobile Publisher, you can apply biometric ID credentials to make your environment more secure. To use biometric ID credentials, your device must have a device passcode or PIN enabled and

FaceID (for iOS) or Android Security and Privacy | Biometrics.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Mobile Publisher apps support biometric ID app unlock using face or fingerprint credentials when the biometric authentication method is available on the Android or iOS device. Depending on the mobile device, Mobile Publisher also supports Swipe and PIN unlock.

When you enable biometric ID app unlock, users are prompted for their biometric credentials when they open the app with a cold start. A cold start is when the user opens the app for the first time or opens the app after closing the app entirely (swiping up to quit the app). If the user has no security settings enabled, they're forced to log out of the app.

Important Remember to wrap attribute values in quotation marks.

1. From Setup, in the Quick Find box, enter `External Client App Manager`, and then select **External Client App Manager**.
2. Select the external client app for the mobile app that you want to enable User Opt-In Biometric Login for.

   Tip The name of the external client app is the name that you specified in the corresponding Mobile Publisher project.

3. Create a custom attribute that enables User Opt-In Biometric Login.
   a. In the Custom Attributes section, click the plus sign.
   b. For the attribute key, enter `ENABLE_OPT_IN_BIOMETRIC_LOGIN`.
   c. For the attribute value, enter `"TRUE"`.

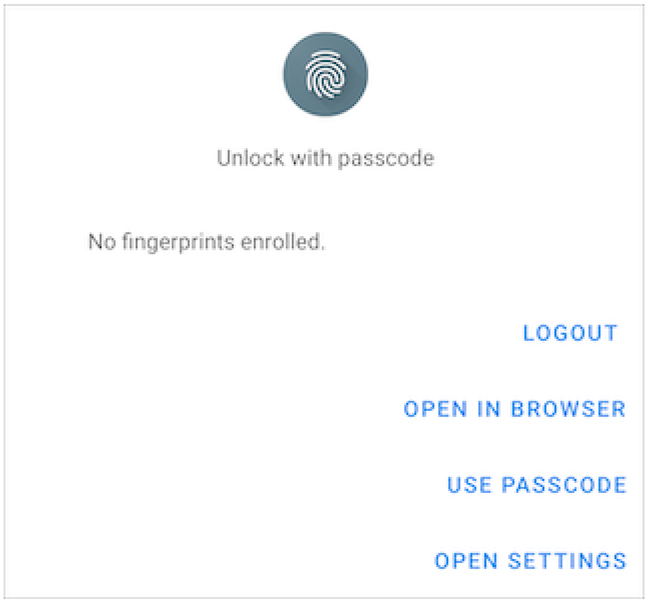
   Note If you previously enabled Biometric ID App Unlock, you don't need to remove the ENABLE_BIOMETRICS_UNLOCK custom attribute. You can keep the ENABLE_BIOMETRICS_UNLOCK custom attribute along with the new User Opt-In Biometric Login custom attributes.

4. Create another custom attribute so that users can fall back to entering a secondary authentication method such as a passcode or pattern after a certain number of failed biometric login attempts.
   a. In the Custom Attributes section, click the plus sign.
   b. For the attribute key, enter `ENABLE_BIOMETRIC_LOGIN_FALLBACK`.
   c. For the attribute value, enter `"TRUE"`.
5. Create another custom attribute that sets the timeout value for biometric login.

6. Save your changes.

After you enable biometric ID app unlock, your app users must configure their devices to enable unlocking apps with biometric ID credentials in order to use this login method.

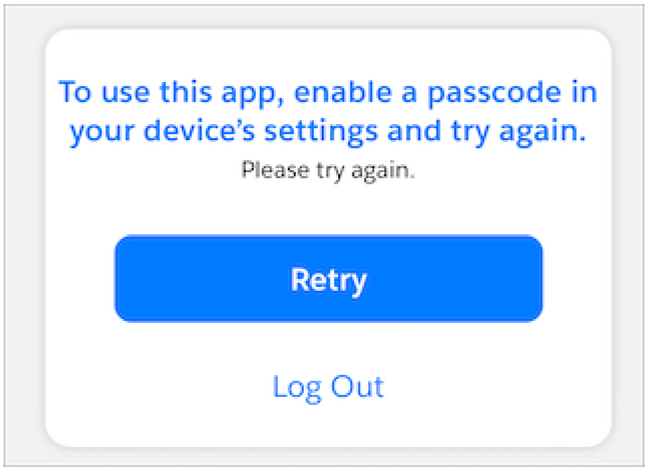
If Android app users don't configure their biometric ID credentials, they're prompted to proceed with one of these options: log out, log in on a browser, log in with a passcode, or to open their device's settings.



Unlock with passcode

No fingerprints enrolled.

If iOS app users don't configure their biometric ID credentials, they're prompted to log in with a passcode. If the iOS app user doesn't have biometric ID app unlock or passcode unlock configured on their device, the user is prompted to enable a passcode.



To use this app, enable a passcode in your device's settings and try again.

Please try again.

Set Time Value for Biometric Login

By default, users must reenter biometric ID credentials when the app is in the background for more than 15 minutes. You can set your own time value requirement by adding a custom attribute.

Set Time Value for Biometric Login

By default, users must reenter biometric ID credentials when the app is in the background for more than 15 minutes. You can set your own time value requirement by adding a custom attribute.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important Remember to wrap attribute values in quotation marks.

1. From Setup, enter *`Mobile Publisher`* in the Quick Find box, then select **Mobile Publisher**.
2. In the External Client Setting tab, select the external client app you want to change the biometric ID unlock time value for.
3. In the Custom Attributes section, click the plus sign.
4. For the attribute key, enter *`REQUIRE_BIOMETRICS_AFTER`*. For the attribute value, enter the value in minutes wrapped in quotation marks.
5. Click **Save**.

Pre-Authorize User App Access Through External Client App Policies

Configure who can use your app by defining which users are pre-authorized. Users who are pre-authorized can bypass the Allow or Deny permission pop-up when accessing the app.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Important Almost all Mobile Publisher apps were moved to the External Client App framework in the Summer '25 release. If your app is one of a few still using Connected Apps, refer to the help documentation for any customizations you need to make.

Admin approved users who are pre-authorized allows only users with the associated profile or permission set to access the app without first authorizing it. After selecting this option, manage profiles for the app by editing each profile's External Client App Access list. Or manage permission sets for the app by editing each permission set's External Client App list.

Warning If you switch from All Users may self-authorize to Admin-approved users are pre-authorized, anyone using the app loses access, unless a user's permission authorizes the external client app specifically. In addition, if users have the Use Any API Client permission, they can access any external client app–even if its Permitted Users setting is set to Admin-approved users are pre-authorized. Be careful when using the Use Any API Client permission. As the name implies, you're giving up your control over authorization.

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. In the External Client App Settings tab, select **Policy Configuration** for your app and click **Edit**.
3. Under OAuth policies, click the **Permitted Users** dropdown and select **Admin approved users are pre-authorized**.
4. Click **Save**.

Push Notifications

Keep your users in the know with timely notifications sent to their mobile devices. Create custom push notifications using Notification Builder or use Marketing Cloud Engagement tools like Automation Studio or Journey Builder.

Create Custom Notifications for Your App
Use Notification Builder to create custom push notifications to send to your app's users.

Create Custom Notifications for Your App

Use Notification Builder to create custom push notifications to send to your app's users.

REQUIRED EDITIONS

Available in: Experience Cloud sites accessed through Lightning Experience in Enterprise, Performance, Unlimited, and Developer editions where Mobile Publisher is enabled.

To create a custom notification for your app, create a custom notification, and select your Mobile Publisher app as its delivery channel. For more information, see Manage Your Notifications with Notification Builder.

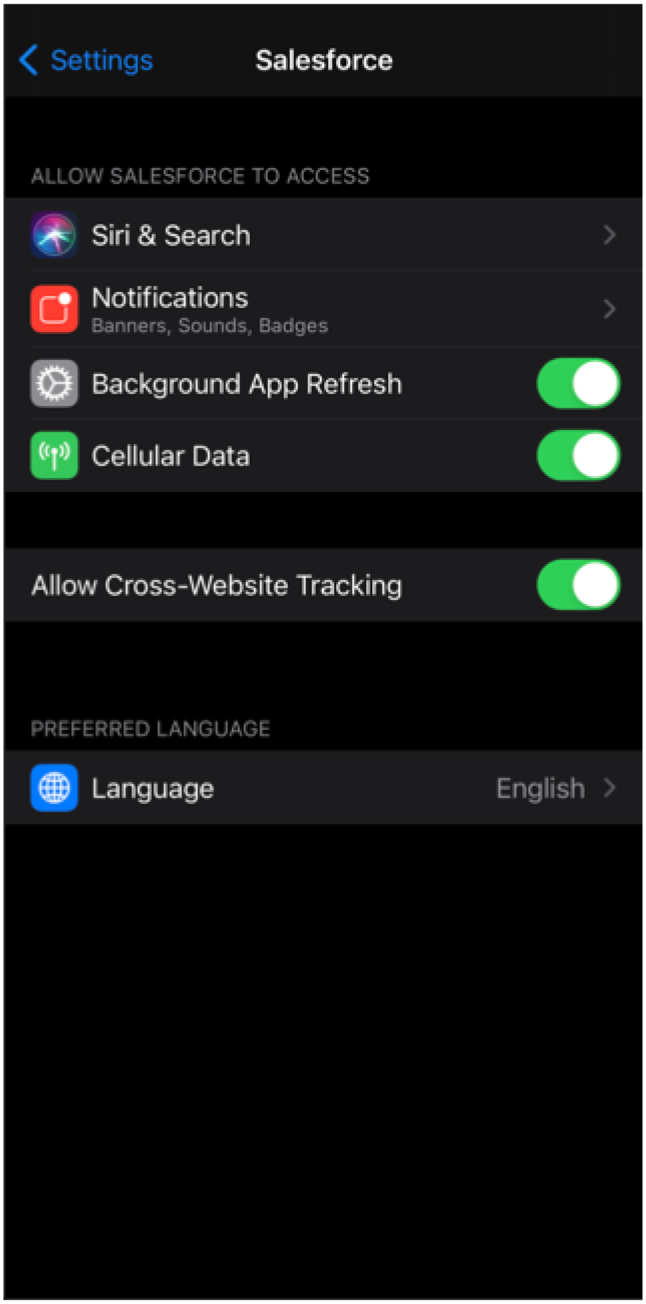
Enable Cross-Website Tracking (iOS)

With iOS 14, Apple placed additional tracking restrictions on third-party applications. These tracking restrictions can prevent your custom web content from appearing in your branded app.

These configurations can be affected.

- Visualforce pages
- Third-party content rendered via Canvas, or custom Lightning components using <iframe>
- Third-party applications using cookies, sessionStorage, or localStorage

To unblock third-party content for your branded app on iOS, go to **Settings | Your App Name**, and turn on **Allow Cross-Website Tracking**.



< Settings Salesforce

ALLOW SALESFORCE TO ACCESS

Siri & Search >

Notifications >
Banners, Sounds, Badges

Background App Refresh ●

Cellular Data ●

Allow Cross-Website Tracking ●

PREFERRED LANGUAGE

Language English >

Test and Submit the App

Request a binary and submit for your branded app to the application stores for approval.

Request a Binary of Your App
Google and Apple provide programs that give your organization the ability to thoroughly test your Android and iOS apps before publishing them.

Test with the UI Test Automation Model (UTAM) for Your App
Testing is paramount in getting your Mobile Publisher apps working as intended. Salesforce already uses UTAM to speed up UI testing, and we want you to have the tools available to thoroughly test your

apps.

Submit the App for Approval
After testing of the beta version of your app is complete, the next step is to submit the app to the application stores for approval.

Request a Binary of Your App

Google and Apple provide programs that give your organization the ability to thoroughly test your Android and iOS apps before publishing them.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

Note Mobile Publisher frequently checks to ensure that the Terms and Conditions are up to date in the stores. If the Terms and Conditions require acceptance, Mobile Publisher sends notifications. To avoid delays with your app going live, keep your Terms and Agreements up to date.

1. From Setup, in the Quick Find box, enter `Mobile Publisher`, and then select **Mobile Publisher**.
2. Click **Continue** next to the listing for the iOS app.
3. Click **Request Binary.**.
4. Click **Done**.
5. Repeat the same steps for the Android app.

   Important For Android private distribution, the BETA status is misleading and not related to providing a beta app to testers. To make the app available for testing, the app status needs to be in LIVE status.

After Mobile Publisher creates the beta versions, we submit them to the App Store and Google Play for approval. This approval process is for the beta version only, and the app isn’t listed in the application stores.

When the beta is available for testing, Mobile Publisher notifies the admin and the designated beta testers. It can take time before Apple or Google approves the apps (Salesforce can’t give a realistic estimate on Apple or Google’s review timeline).

Test with the UI Test Automation Model (UTAM) for Your App

Testing is paramount in getting your Mobile Publisher apps working as intended. Salesforce already uses UTAM to speed up UI testing, and we want you to have the tools available to thoroughly test your apps.

Salesforce has created a UI Test Automation Model (UTAM) to support test automation of Mobile Publisher apps. UTAM is based on the popular Page Object model design pattern commonly used in UI tests.

To understand how to use UTAM to generate page objects for native pages or components, and run a test on iOS and Android platforms, consult either the Java or the JavaScript mobile setup guide.

Examples for Java are in the utam-java-recipes repo. Examples for JavaScript are in the utam-js-recipes. Each repo's `README` file explains how to set up necessary tools and where to find examples of mobile tests.

After you set up the necessary tools and are ready to test, here are resources to test your Mobile Publisher apps.

- Mobile Publisher Sample Test: utam-java-recipes (github)
- UTAM Page Objects for the Publisher Playground App: salesforce-pageobjects (Native Package Manager)

See Also

UI Test Automation Model: Learn about UTAM for Mobile
Lightning Web Components Developer Guide: Install Mobile Extensions
Github: Mobile Publisher Sample Test
Native Package Manager (npm): UTAM Page Objects for the Publisher Playground App

Submit the App for Approval

After testing of the beta version of your app is complete, the next step is to submit the app to the application stores for approval.

REQUIRED EDITIONS

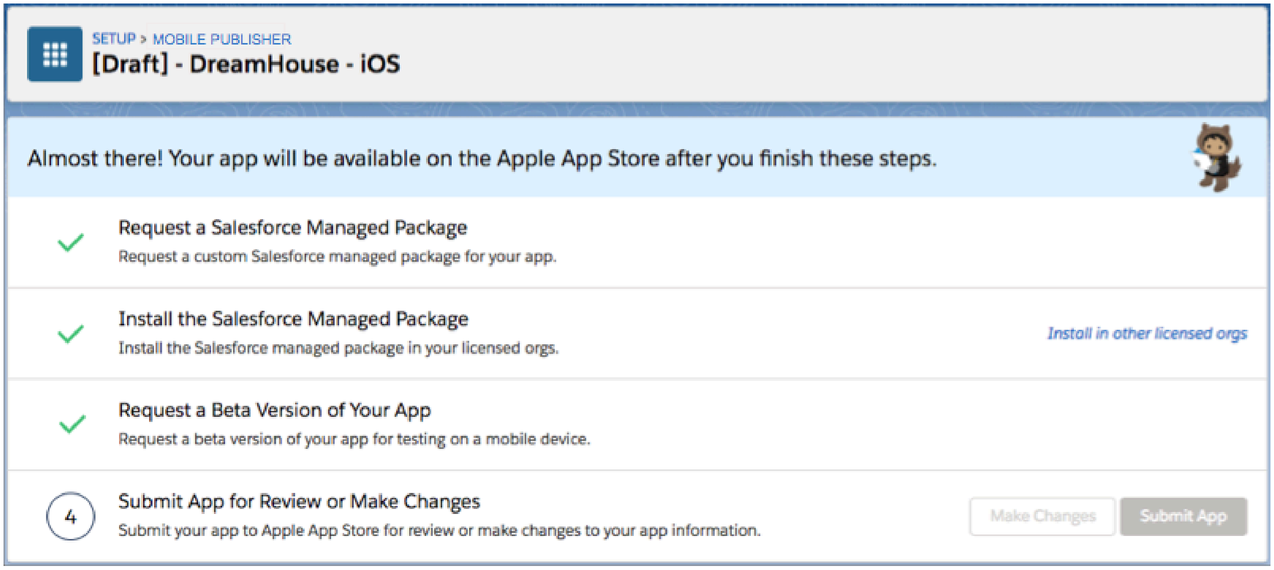
Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

When you submit your app for approval, Salesforce sends the final apps to Google Play and the App

Store. Google and Apple review the apps to make sure they adhere to the application store guidelines. Although you're about to submit the apps in Salesforce, they aren't immediately available in Google Play and the App Store.

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Click **Continue** next to the listing for the iOS app.



SETUP > MOBILE PUBLISHER

[Draft] - DreamHouse - iOS

Almost there! Your app will be available on the Apple App Store after you finish these steps.

✓ Request a Salesforce Managed Package
Request a custom Salesforce managed package for your app.

✓ Install the Salesforce Managed Package
Install the Salesforce managed package in your licensed orgs.
Install in other licensed orgs

✓ Request a Beta Version of Your App
Request a beta version of your app for testing on a mobile device.

4 Submit App for Review or Make Changes
Submit your app to Apple App Store for review or make changes to your app information.
Make Changes | Submit App

Note To make edits, click the **Make Changes** button. On the App Information page, you can upload revised brand assets or change the values in the fields. Then follow the same steps again: Request a new managed package, install it, and request a beta to verify the changes. The **Make Changes** button is available only after a beta app or live app is delivered.

3. Click **Submit App**.
4. To confirm that you're ready to submit the apps, select the checkbox.
5. Click **Submit App**.

   Salesforce submits the app to the application store. It can take anywhere from 1 day to 2 weeks for the store to review and approve your app. During that time, you can't make any changes to the app.

6. Repeat the same steps for the Android app.

Important For Android private distribution, the BETA status is misleading and not related to providing a beta app to testers. To make the app available for testing, the app status must be in LIVE status.

If there's an issue during the approval process, Google and Apple can reject the app. If the app is rejected, Salesforce emails you and lets you know exactly how to resolve any issues.

Note For apps distributed with the Binary Upload distribution method, we recommend that you release app updates to your users in a staggered rollout when the update includes any new features. A staggered rollout gives you more flexibility in pausing or rolling back an app update if any issues are discovered in the early phases of the release. See Release a version update in phases in *Apple*

Developer Documentation and Release app updates with staged rollouts in *Play Console Help*. You can stagger your app update rollout by the percentage of app users. For example, you can start rolling out the update to 1% of users and then 5% of users. If no issues are discovered, you can incrementally increase the percentage. We also recommend against rolling out an app update before a weekend or holiday, so that any issues can be addressed quickly.

Maintain and Update the App

Learn about the types of maintenance and how you can update your app.

Types of Maintenance
There are two types of maintenance: updates that Salesforce initiates and updates that your organization initiates.

Create a New Version of Your Salesforce App
When you want to edit the visual appearance of an app that's already been published, you have to create a version of the app. For example, you can create a version of your app if you want to change the visual appearance with a new app icon and branding assets.

Types of Maintenance

There are two types of maintenance: updates that Salesforce initiates and updates that your organization initiates.

Note Mobile Publisher frequently checks to ensure that the Terms and Conditions are up to date in the stores. If the Terms and Conditions require acceptance, Mobile Publisher sends notifications. To avoid delays with your app going live, keep your Terms and Agreements up to date.

Salesforce Initiated Updates

Sometimes Salesforce updates your app. Here are the most common reasons.

- Salesforce releases a new version of the Salesforce mobile app or a new version of the Mobile Publisher app for Experience Cloud.
- If Apple or Google changes their app submission form, Salesforce updates the mySalesforce form to reflect the change and then rolls out a release.

Sometimes Salesforce resubmits your app to Google and Apple. If Salesforce updates your apps for any reason, we send you an email in advance and provide plenty of information and instructions.

Your Updates

You can update your branded apps to:

- Change the branding. For example, if you change your company's logo or if you have a new design of your app icon.
- Change the distribution type and distribution account
- To edit the visual appearance of an app that's already live on the application store, you have to create a version of it.

Create a New Version of Your Salesforce App

When you want to edit the visual appearance of an app that's already been published, you have to create a version of the app. For example, you can create a version of your app if you want to change the visual appearance with a new app icon and branding assets.

REQUIRED EDITIONS

Setup for Mobile Publisher available in: Lightning Experience
Setup for Mobile Publisher available in: production only (not sandbox)
Available in Lightning Experience in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED	
To create and modify a Mobile Publisher project:	Manage mySalesforce Apps

1. From Setup, enter `Mobile Publisher` in the Quick Find box, then select **Mobile Publisher**.
2. Click **Continue** next to the listing for the iOS or Android app for which you want to create a version.
3. Click **Create New Version**.

   Salesforce creates a draft of the app and copies over the existing information and assets to the new draft version.

4. Update the necessary fields or upload revised branded assets.
5. Click **Submit**.

Request a binary of the app and make sure you or your tester thoroughly reviews the updates. If everything looks good, see Submit the App for Approval to get your updated branded app published.

Best Practices and Troubleshooting

This information helps you implement recommendations and troubleshoot when using Mobile

Publisher.

- Setup for Mobile Publisher is available in Lightning Experience only. However, admins can switch to Lightning Experience from Salesforce Classic to manage the branded app lifecycle.

Recommendations

- Use a sandbox or UAT environment for testing.
- Use standard components for logout.
- Only enable the Manage mySalesforce App permission for admins or specific required profiles.

Authentication
This information helps you troubleshoot authentication issues when using Mobile Publisher.

De-list or Remove Apps from App Stores
Learn how to temporarily remove apps from app stores or decommission apps that are no longer being used.

Most Common Reasons for App Store Rejections
Here's some suggestions to help prevent your iOS app from getting rejected by Apple reviewers.

Known Issues and Limitations for Mobile Publisher
Learn about known issues for Mobile Publisher.

Authentication

This information helps you troubleshoot authentication issues when using Mobile Publisher.

- If you use SAML authentication, configure it to use HTTP POST to avoid RelayState cut off.
- Use the default login page if you have only one auth provider.
- Don't enforce IP restrictions unless you're required to not allow access to the Experience Cloud site from outside networks or LTE.
- For mobile apps, the ideal setting for refresh token policy is to never expire. This setting provides the best experience for your users.
- Use the "SsoUrl" defined in ".well-known/auth-configuration" to trigger a login request in Experience Cloud sites.
- If there are missing or incomplete CA (Certificate Authority) intermediate certificates, Android users could experience messages stating the Experience Cloud site is unable to load. Some Android devices don't have trusted intermediate certificates. To resolve this issue, the certificates must be installed and trusted or the certificate chain must be uploaded to Salesforce and associated with the custom domain.
  - Qualys SSL Labs offers a web-based free service (https://www.ssllabs.com/ssltest/analyze.html) that can quickly analyze custom domain URLs and check for incomplete certificate chains.
  - For more information about certificate chains, see Merge a complete certificate chain for custom HTTPS domains.
- If you use reCAPTCHA for your login flow you must specify the login page to open in an in-app browser. Also make sure to use a supported mobile browser (Google). For more information on setting up reCAPTCHA, see Set Up ReCaptcha.

De-list or Remove Apps from App Stores

Learn how to temporarily remove apps from app stores or decommission apps that are no longer being used.

Temporarily Remove Mobile Publisher Apps from Sale
When an app has been delisted or temporarily removed from sale, current users aren't affected. However, new users can't search or download the delisted app.

Decommission a Mobile Publisher App
Decommissioning an app means that the app no longer is offered through Mobile Publisher.

Temporarily Remove Mobile Publisher Apps from Sale

When an app has been delisted or temporarily removed from sale, current users aren't affected. However, new users can't search or download the delisted app.

The delisted apps can be added back for sale, and aren't deleted from the respective app stores.

For Apple apps, log in to your App Store Connect account to delist or temporarily remove an app from sale.

1. From My Apps, select the app that you want to remove. The page opens with the App Store tab selected.
2. In the sidebar under General, click **App Information**.
3. Scroll to the Additional Information section, then click **Remove App**.

   Note If you don't see Remove App, make sure that you're an Account Holder or Admin and that your app is in a state that allows removal.
4. In the dialog that appears, click **Remove**.

The app can still be accessed by going to the upper right dropdown menu where it shows All Statuses and switching it to Removed Apps.

Important When an app is delisted or removed from sale, the App Store Name becomes available to other Developers to use. If the app name you used is no longer available, the app can't be restored. If another app is preventing you from using your trademark as an app name on the App Store, you can submit a claim with Apple here: App Name Dispute.

See Also

App Store Connect Help: Remove an app from the App Store
Play Console Help: Update or unpublish your app

Google Apps

Log in to your Google Play Console to delist or temporarily remove an app from sale.

1. Open the Play Console.
2. Select the app you want to remove.
3. Go to **Release | Setup | Advanced settings**.
4. On the App Availability tab, select **Unpublish**.

Decommission a Mobile Publisher App

Decommissioning an app means that the app no longer is offered through Mobile Publisher.

To decommission an app, open a support case with Mobile Support. To open a support case, see Create a Support Case.

Include these details in your support case.

1. App name as it appears on the app stores.
2. Org ID of the Salesforce org that the app is associated with.
3. State your request to decommission the app. To continue to use the app name, specify that the listing is to remain in the store. Otherwise, Mobile Publisher removes the iOS app from sale or unpublishes the Android app.
   Most customers don't use their Mobile Publisher app after it's decommissioned. A decommissioned app is removed from the live listing and doesn't appear in the stores, but it isn't permanently removed from your store. Some customers want to use their own binary post-decommissioning. If specified, Mobile Publisher doesn't remove the app from sale or unpublish it. For example, the app's iOS bundle Id and the app remain live in the store. You can't use duplicate names or transfer a current app to another one.

The Support team coordinates with the Mobile Publisher team to permanently decommission your app.

Most Common Reasons for App Store Rejections

Here's some suggestions to help prevent your iOS app from getting rejected by Apple reviewers.

- Section 3.2 of Apple's App Store Review Guidelines is a commonly cited reason for App Store rejection. If section 3.2 was the reason for your app's rejection, consider Apple's unlisted app distribution option.
  For more information on the unlisted app distribution option, see Unlisted Option for iOS App Distribution.
- Ensure the credentials that you provided for Apple reviewers to test with are always valid. For more information, see step 13 in Create a Placeholder App.
- Ensure that the screenshots you're submitting for iPhone and iPad show the app in use.
  - Ensure that the screenshot for the device is taken on that specific device. For example, don't use a screenshot for the 12.9 inch (32.766 cm) iPad that was taken on an iPhone and stretched out.
  - Ensure that the screenshots are from the actual branded app or from the Publisher Playground app. Don't take screenshots of your org in a mobile web browser, such as Safari, Chrome, or Firefox.
- Ensure that you're not entering in other mobile services names, such as Apple, Google, or Android, in

the store descriptions, app name, or screenshots.
  - Ensure that the content within the app doesn't contain any mobile services names such as Apple, Google, or Android.
- Ensure that you're not entering in non-production ready terms, such as beta or test, in your store description, app name, or screenshots.
- If your app is going to be published to the App Store publicly, don't enter in audience-specific language in your store description, app name, or screenshots. An example of specific language is employee, customer, or internal.
- If your app is available in China, make sure to enter the app's Internet Content Provider (ICP) Filing Number in your App Store Connect account. For more information, see the "Availability in China mainland" section of App information in *App Store Connect Help*.

See Also

App Store Review Guidelines (Apple)

Known Issues and Limitations for Mobile Publisher

Learn about known issues for Mobile Publisher.

- You can't identify and send email notifications to Mobile Publisher admins. Email notifications go to all your system admins.
- You can create hyperlink formula fields for iOS and Android. But Android doesn't support:
  - `HYPERLINK("https://salesforce.com", "Salesforce")`
  - `HYPERLINK("https://salesforce.com", "Salesforce", "_blank")`

  Android does support:
  - `HYPERLINK("https://salesforce.com", "Salesforce", "_self")`

  See Tips for Working with Hyperlink Formula Fields for more information.