Emory Mobile App Signing and Distribution

Status

This document is a draft.

Apps that pass the review process are eligible to be distributed in the Emory section of the Apple App Store and Google Play Store. This article proposes additions to the manner of signing and distribution used to upload a reviewed app to the store and to publish it. This only applies to apps that are meant for public distribution.

In order to safeguard the Emory brand, the current policy is to not share credentials or secrets with app vendors and developers, both internal to Emory and external. This means that the vendor or developer *is not* issued credentials to login to the Apple App Store Connect, Apple Developer Portal or the Google Play Console. The vendor or developer is not allowed to upload binaries to the store nor can they use Emory signing credentials, nor can they distribute an app for testing or submit it for publishing to production.

Instead, the vendor or developer provides Emory with a signed binary archive (SBA) of the app – an ipa file for iOS, an apk file for Android. The vendor or developer must use their own developer account(s) to build, sign and export the app to the binary archive format which is then delivered to Emory. Emory then resigns the app, then uploads to the store, then submits it for review and publishing.

Question for Apple

What is Apple's policy about sharing certs? To what risks does doing so expose the account holder?

The only alternative to providing an SBA is for the vendor or developer to provide the source of the project, Xcode or Android Studio usually, and to have Emory build the app. This not an ideal solution because automation (build scripts) and environmental (Xcode version, library versions) dependencies. It is useful for when the vendor or developer can't produce the signed binary archive because they lack the knowledge or are otherwise unwilling to do so.

The policy was put in place nearly 10 years ago and reflected in part the practical limitations of both stores ability to create a user that was limited to actions for a particular app or apps. In the present this is not the case: both app stores allow creating of users that are limited to a single app or set of apps.

This policy worked well for many years and did not need to be changed. This is because many of the developers were Emory employees and were bound to comply with the policy. The few 3rd party developed apps did not seem to mind complying with the policy. Now however, many more of the apps in the Emory stores are developed by 3rd parties. Many of those are unable or unwilling to provide a signed binary archive and are requesting access to the stores, specifically, the permission to distribute the app. The requests were denied based on concerns about protecting Emory's brand.

This started with the "Theater Emory" app, a, Emory branded version of a generic app by a vendor who claimed they could only distribute the app by uploading it to the store. Furthermore, they wanted to update the app dozens of times a week. The request was denied and the theater department dropped the vendor and did not replace it.

Then, the Yomingo vendor would not provide the SBA and demanded that they should control the distribution and be able to "pull the app from the store" for reasons (such as non-payment) that are normally resolved contractually, not by relinquishing control over distribution. We nearly convinced them to change their minds but in the end the app owner dropped them and is in search of a new vendor.

Another class of vendor that is quite popular now but nearly non-existent only a few years back is the DIY vendor. This is exemplified by the case of the CBCT department who wanted to use Good Barber, a DIY vendor, to build their simple app that played MP3s of guided meditation. In this case, Good Barber's policy conflicted with Emory's. Just like Yomingo, Good Barber wanted to be the one who controlled distribution/publishing.

There are other mobile app vendors and developers that have no desire to distribute their apps but want access to the Emory accounts simply do not understand how to build SBAs, or, have some reason why they can't or won't. Instead they want access to the store to do the upload.

For instance, Softura, a vendor responsible for the HeathMindr, eP / P@H, and SMART apps, tried for a while to build the SBAs for eP and P@H but quit after several of requests for new ones due to technical difficulties we had resigning and uploading the SBA to App Store. First, they handed over the source, however, we were unable to build the SBAs from the source. So they requested and have been granted access to upload directly to the store (for Apple only) to solve the problem once and for all. The have been granted 3 account with Developer role permissions with access to certificates, identifiers and profiles. This makes it is possible for them to upload the app to the store but *not* to distribute it.

Question for Apple

What does checking "Access to Certificates, Identifiers & Profiles" on the new user screen do when they have access to only one app? Does it allow them to view all objects in the developer account or just those that are used by the app? Does it allow the user to create objects or just view and download them?

If you do not check this box does that mean you must export the signing cert and create a provisioning profile for the app and give it to the developer who must then configure Xcode to do manual signing by using them?

Question for Google

What is a draft app? Does the vendor/developer need any keys from Emory to put one in the play store? How do I promote a draft to production?

Mind you, this all happened in less than one year. There is no sign that the pace will slow. We have to change our policies sooner than later in order to keep from delaying the release of approved apps.

The EHC Patient Portal app, HealtheLife by Cerner will be the first app in which we allow a 3rd party to publish the app. Several vendors have asked for this permission in the past yet none had the clout to demand it like Cerner and The Cerner app now sets a precedent whereby we can be legitimately challenged by a 3rd party who wishes to have the same access as Cerner.

Here are the proposed new supported roles that allow delegation of certain tasks to the vendor or developer while still allowing Emory to protect its brand. Both Apple and Google support these roles.

Developer Role

The developer role will allow the user to access the Emory developer account and upload apps designated by Emory (and only designated apps) directly into the store. The developer user can not distribute not publish apps.

Apple App Store Settings for Developer Role

App Features

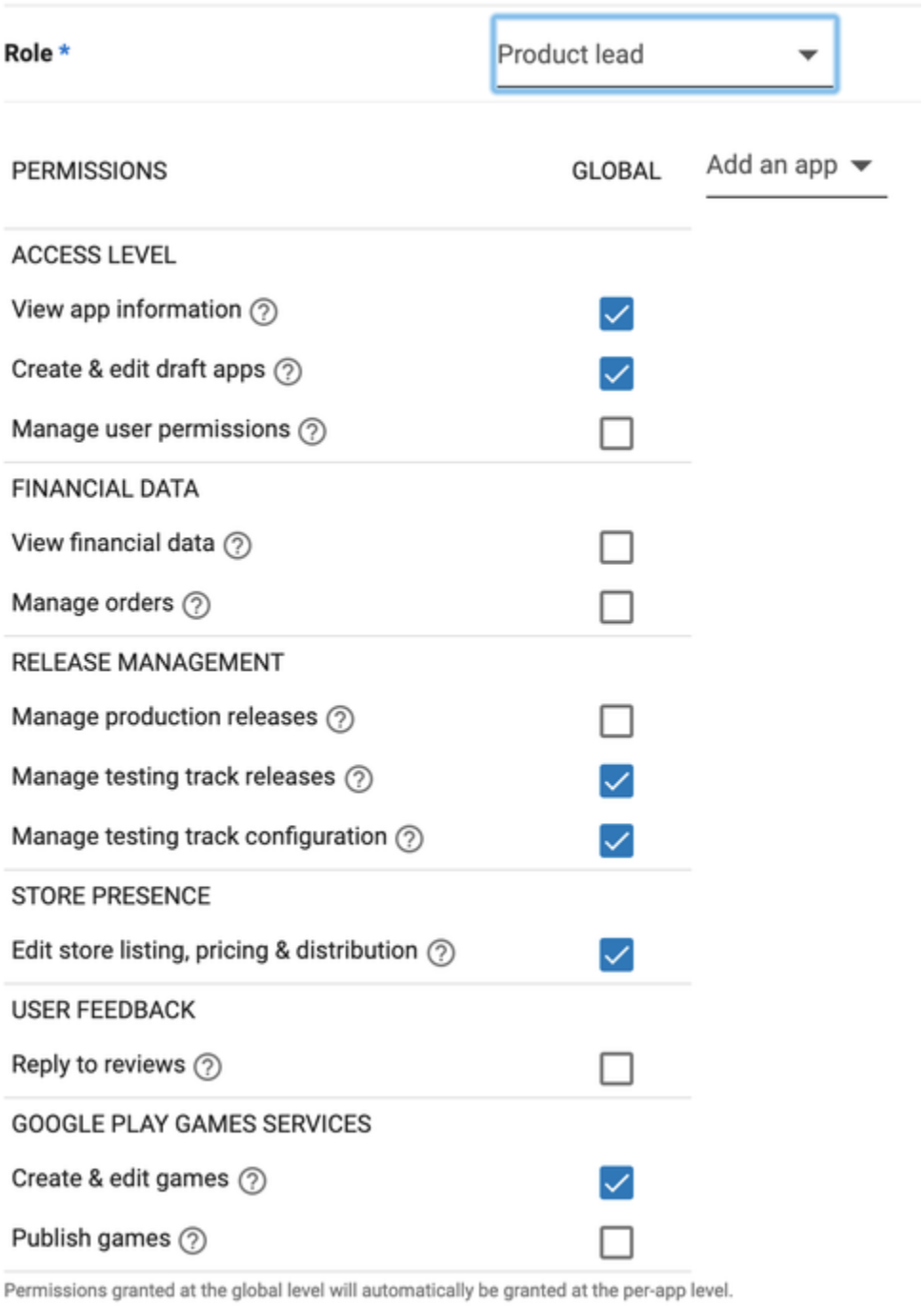
- Create In-App Purchases
- Edit App Store Details (Read Only)
- Edit In-App Purchases (Read Only)
- Manage Game Center
- Manage TestFlight Builds (Read Only)
- Manage TestFlight Testers (Internal Only)
- Reply to and Edit Responses to Customer Reviews (Read Only)
- Upload Builds

Developer Features

- Purchase and submit Technical Support Incidents
- Download beta software
- Eligible for other membership benefits

Google Play Store Setting for Developer Role

This is the "Product Lead" role. Note that "global" access is shown but the user would really be restricted to one or more apps.



App Manager Role

The App Manager role will allow the user to access the Emory developer account to both upload and distribute designated apps.

Apple App Store Settings for App Manager Role

App Features

- Create Apps and Submit Versions
- Create In-App Purchases
- Edit App Pricing and Availability
- Edit App Store Details
- Edit In-App Purchases
- Manage Game Center
- Manage Leaderboard Scores
- Manage Promo Codes and Promo Art
- Manage TestFlight Builds
- Manage TestFlight Testers
- Reply to and Edit Responses to Customer Reviews (Read Only)
- Reset App Summary Rating
- Submit In-App Purchases
- Upload Builds

Provider Features

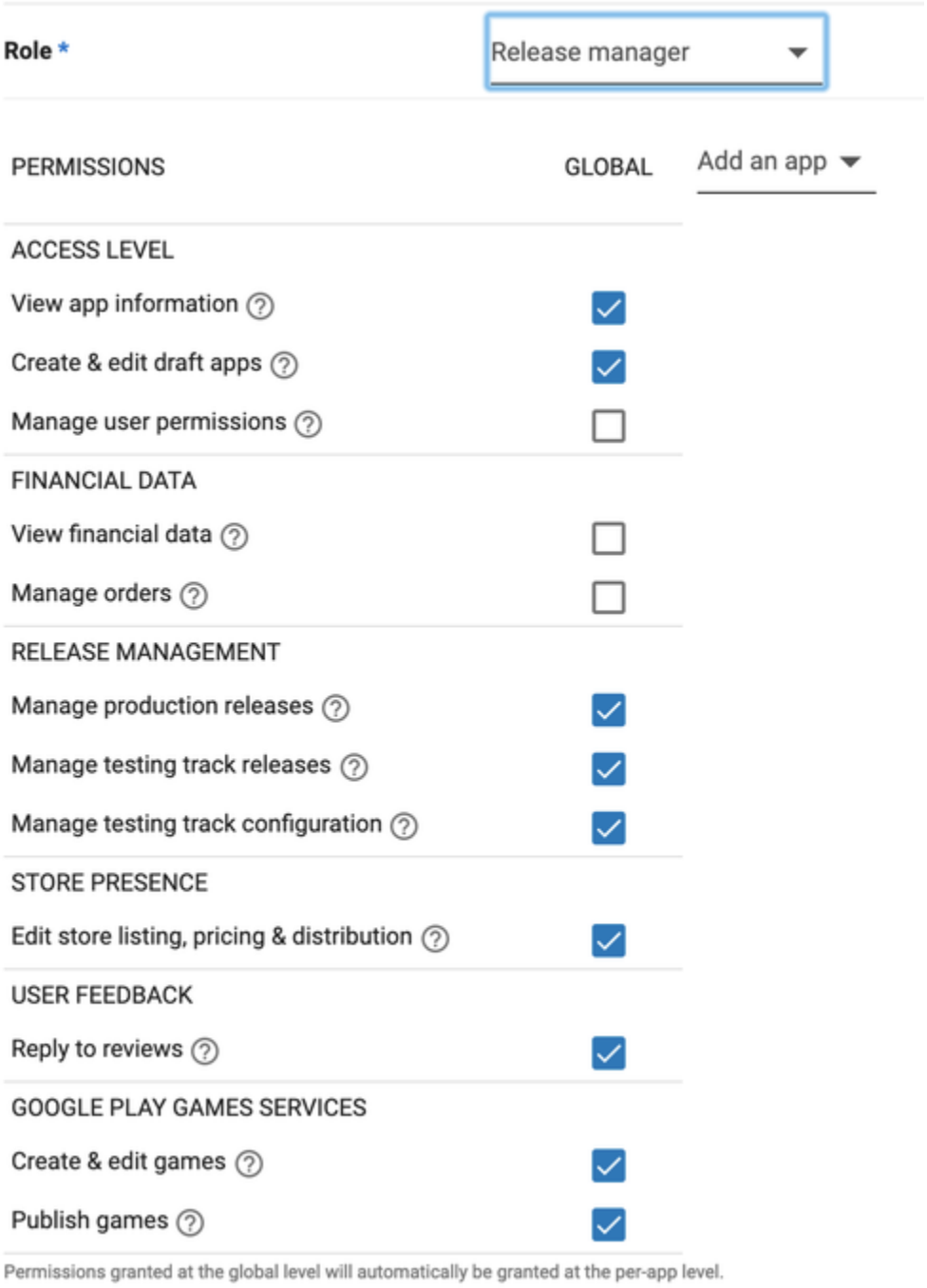
- Manage App Access
- Manage Sandbox Testers
- Manage Users and Roles

Developer Features

- Purchase and submit Technical Support Incidents
- Download beta software
- Eligible for other membership benefits

Google Play Store Setting for App Manager Role

This is the "Release Manager" role. Note that "global" access is shown but the user would really be restricted to one or more apps.



Workflow

Publishing

The account holder is not allowed to publish the app unless granted permission under special circumstances such as the Cerner HealtheLife app. In general the task of publishing the app is reserved for the MARDT, specifically an admin for the Emory's Apple Developer's Account(s) and Google Play Store Console Account(s).

Risk Assessment and Mitigation

Mitigation

- Required developer to request continued access every (6?) months or else the account will be deleted.
- Require developer to acknowledge agreement
- Delete account
- Revoke certificate
    - Prohibits Xcode automatic signing and upload of app to App Store

Benefits Analysis

Shorter turnaround without middleman involved. App Store upload errors and review rejections are handled much more quickly.

The Manager can create and manage the store profile directly and no longer needs to fill out a store template. Can resolved screenshots, other graphic and miscellaneous problems more easily.

Developer/Manager Agreement

Do Nots

- Do not use account for any other purpose than to upload or distribute apps for which the account holder is so authorized.
- Don't do anything to get Emory's account suspended.
- Don't do anything that might be permissible by virtual of the role assignment but not is not permissible by contract. For instance, don't distribute updates without permission or make the app unavailable.
- Do not create users without permission.
- Do not create certificates without permission.
- For each permission granted, indicate if and how it will be used.
- Don't share secrets
- Safeguard secrets
- Rotate secrets often
- Discard unused and expired secrets
- Report the compromise of secrets as soon as discovered.
- Use 2FA

Can Dos

The vendor or developer must sign a special agreement to obtain a Developer or Admin role for one of more of their apps.

Onboarding for Client Managed Apps

MARDT Admin does this

- create app id
- create app on the app and/or play store.

Client does this:

1. Client requests access to Emory store accounts.
2. Invitations are accepted.

For iOS apps, the MARDT Admin does this:

- create provisioning profile
- create other certs or keys as needed
- create other developer resources as needed
- provide apple signing cert to app developer
- set app to "manual" release

For iOS apps, the Client does this:

- if using Xcode
    - change team to 'Emory University'
    - archive the app
    - export or upload app using manually managed signing downloading profile if needed

Assumptions

- Apple Apps store with App Store certs can't be installed on a device from any source other than the App Store. This is unlike In-house (enterprise certs) which can be installed from multiple sources.