Emory Mobile App Signing and Distribution

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.

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

Instead, the vendor or developer will provide 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 to build, sign and export the app to the binary archive format which is then delivered to Emory. Emory resigns the app, then uploads to the store, then submits it for review and publishing.

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 is a last resort because it can be problematic. 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.

This 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 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 or set of apps.

Another limitation that has been lifted was the ability to restrict a user from performing the distribution/publishing task because Emory wants to have full control over this and not delegate this task.

This policy worked well for many years and did not need to be changed. It did so because many of the developers were Emory employees and were bound to comply with the policy. Of the few that weren't they were willing to comply without question. The 3rd parties were the minority and did not seem to mind not being troubled in complying with the policy. Now however, at least half check this 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. This started with Theater Emory who could only distribute the app by uploading it to the store. Furthermore, they wanted to update the app dozens of times a week.

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.

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 many, many more mobile app vendors and developers and many of them 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. They have no interest in doing the distribution.

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. Instead the handed over the source, however, we are unable to build the SBAs from the source. They have requested access to upload directly to the store (for Apple only) to solve the problem once and for all. It seems like it is possible to create a user in a role that would allow them to upload the app to the store but they would not be able to distribute it.

Status
This document is a draft.

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 is a last resort because it can be problematic. 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.

This 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 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 or set of apps.

Another limitation that has been lifted was the ability to restrict a user from performing the distribution/publishing task because Emory wants to have full control over this and not delegate this task.

This policy worked well for many years and did not need to be changed. It did so because many of the developers were Emory employees and were bound to comply with the policy. Of the few that weren't they were willing to comply without question. The 3rd parties were the minority and did not seem to mind not being troubled in complying with the policy. Now however, at least half check this 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. This started with Theater Emory who could only distribute the app by uploading it to the store. Furthermore, they wanted to update the app dozens of times a week.

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.

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 many, many more mobile app vendors and developers and many of them 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. They have no interest in doing the distribution.

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. Instead the handed over the source, however, we are unable to build the SBAs from the source. They have requested access to upload directly to the store (for Apple only) to solve the problem once and for all. It seems like it is possible to create a user in a role that would allow them to upload the app to the store but they would not be able 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?
Mind you, this all happened in less than one year. There is no sign that the pace will slow. Already the HealtheLife (Patient Portal) vendor has (perhaps) stipulated in contract that they want access to the stores however the details are not presently clear. It is also unclear that EHC and Emory would grant them the ability to distribute the app due to the sensitive nature of the app. However, there are many other apps that are not of such a sensitive nature and it may be “ok” to allow the vendor or developer to distribute the app (e.g. CBCT and Theater Emory).

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 publish apps.

### Apple App Store Settings for Developer Role

**Roles**

- [ ] Admin
- [ ] Sales
- [ ] Customer Support
- [x] Developer
- [ ] Finance
- [ ] Access to Reports
- [ ] App Manager
- [ ] Marketing

**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.
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

<table>
<thead>
<tr>
<th>PERMISSIONS</th>
<th>GLOBAL</th>
<th>Add an app</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACCESS LEVEL</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View app information</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Create &amp; edit draft apps</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Manage user permissions</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FINANCIAL DATA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View financial data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage orders</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RELEASE MANAGEMENT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage production releases</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage testing track releases</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Manage testing track configuration</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td><strong>STORE PRESENCE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit store listing, pricing &amp; distribution</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td><strong>USER FEEDBACK</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reply to reviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GOOGLE PLAY GAMES SERVICES</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create &amp; edit games</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Publish games</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Permissions granted at the global level will automatically be granted at the per-app level.
Roles

☐ Admin
☐ Sales
☐ Customer Support
☐ Finance
☒ Developer
☐ Marketing
☐ Access to Reports
☒ App Manager

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.
Risk Assessment

Special Agreement

- Don’t share secrets
- Safeguard secrets
- Rotate secrets often
• Discard unused and expired secrets
• Do not use account for any other purpose than to upload or distribute apps for which the user is so authorized.
• Don't do anything to get Emory's account suspended.
• Don't do anything not allowed by the role but not by contract. For instance, don't distribute updates without permission or make the app unavailable.
• Report the compromise of secrets as soon as discovered.
• Use 2FA

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