Cornell University – iOS App Store

The Cornell Center for Technology Enterprise & Commercialization (CCTEC) owns and funds Cornell University’s Apple iOS AppStore agreement. Apple policy only allows 1 AppStore agreement per DUNS number. CCTEC’s Chidori Boeheim (cb472@cornell.edu) retains full administrative, legal, and billing responsibility for Cornell’s agreement.

Requests for submitting a new App begin with a discussion of the purpose and intellectual property involved with CCTEC. Depending on the App’s origin, many additional steps, including invention disclosure may be necessary. If transfer of proceeds from paid apps or in-app purchases is required, additional steps are required. After CCTEC believes process is on track, technical discussions with SAS-IT may begin for the submission process. Before final submission, confirmation with CCTEC is required. CCTEC has an agreement to review the Cornell branding present in the App with University Communications.

iTunes Connect

Submission of new, updated, or changes to the ‘Cornell University’ iOS AppStore are maintained centrally through iTunes Connect https://itunesconnect.apple.com. CCTEC has delegated access to a “holding” Apple ID setup by SAS-IT, sas-mobile-dev@cornell.edu. SAS-IT/Eric Grysko (emg33@cornell.edu) manages submissions and updates.

Submission Overview

The following process has been used successfully with several departments and vendors building apps for the Cornell University community. Unlike the clients of some vendors, Cornell will and does not share the distribution certificate/key/provisioning profile, iTunes Connect access, or provide admin or agent access to our account. Instead Cornell utilizes a process to re-codesign a fully compiled archive of the built product. Since it's an already a compiled archive, there is no source code/IP disclosed. Additionally since it is only a resign, there's no scenario where bugs or changes can be introduced into the App.

Overview

1. Department/Vendor identifies a developer with an Apple ID (email address) who will be responsible for signing the App. A "generic developer" is recommended for vendors.
2. Cornell invites the new developer to the 'Cornell University' team with a developer role.
3. Developer creates a certificate request submitted to the Cornell team.
4. Cornell approves the vendor developer certificate request
5. While likely unnecessary, if developers needs a couple device UDID added to our team, this can be done.
6. In parallel, a discussion of the needs for a new 'App ID' (explicit or wildcard) is necessary. This includes agreeing to the Bundle ID (suffix), for example edu.cornell.birds.livingbird. Additionally review of “application services” is necessary for the App ID, for example In-App Purchases or Apple Push Notifications.
7. Cornell creates App ID and adds developer certificate to the appropriate development provisioning profile.
8. Vendor developer (refreshes) the team thru Xcode settings to get proper provisioning profiles pulled down to their development environment
9. In the developer' XCode project, info.plist (or similar) -- add a reference to an Entitlements.plist. In the entitlements.plist you're basically looking to set get-task-allow to false, and a few other things: see http://stackoverflow.com/a/9727388 for more details.
10. Developer adjust signing (for release) to sign with the Cornell University dev prov profile
11. Build to archive with a Release configuration.
12. Vendor can test the entitlements in the archive...
   ```
   codesign -d --entitlements - appname
datetime.xarchive/Products/Applications/targetnamehere.app
   ```
   And then look for get-task-allow false in the output xml. While it's not the only item that matters, it's the trickiest to get correct.
13. Developer compresses .xarchive as a .zip and transmits to Cornell
14. In parallel, Cornell works with the department/vendor to compile any necessary app metadata (keywords, scriptions, support URLs, screenshots, etc).
15. Cornell can verify the build and upload to iTunes Connect.

For any future app updates, the same process is followed.