App ID Prefix Change and Keychain Access

This thread has been locked by a moderator.

455

DTS regularly receives questions about how to preserve keychain items across an App ID change, and so I thought I'd post a comprehensive answer here for the benefit of all.

If you have any questions or comments, or other creative solutions!, please start a new thread here on DevForums, tagging it with *Security* so that I see it.

Share and Enjoy
—
Quinn "The Eskimo!" @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

App ID Prefix Change and Keychain Access

The list of keychain access groups your app can access is determined by three entitlements. For the details, see Sharing Access to Keychain Items Among a Collection of Apps. If your app changes its App ID prefix, this list changes and you're likely to lose access to existing keychain items.

This situation crops up under two circumstances:

- When you migrate your app from using a unique App ID prefix to using your Team ID as its App ID prefix.
- When you transfer your app to another team.

In both cases you have to plan carefully for this change. If you only learn about the problem after you've made the change, consider undoing the change to give you time to come up with a plan before continuing.

Note On macOS, the information in this post only applies to the data protection keychain. For more information about the subtleties of the keychain on macOS, see On Mac Keychains.

For more about App ID prefix changes, see Technote 2311 Managing Multiple App ID Prefixes and QA1726 Resolving the Potential Loss of Keychain Access warning.

Migrate From a Unique App ID Prefix to Your Team ID

Historically each app was assigned its own App ID prefix. This is no longer the case. Best practice is for apps to use their Team ID as their App ID prefix. This enables multiple neat features, including keychain item sharing and pasteboard sharing.

If you have an app that uses a unique App ID prefix, consider migrating it to use your Team ID. This is a good thing in general, as long as you manage the migration process carefully.

Your app's keychain access group list is built from three entitlements:

- `keychain-access-groups`, see Keychain Access Groups Entitlement
- `application-identifier` (`com.apple.application-identifier` on macOS)
- `com.apple.security.application-groups`, see App Groups Entitlement

IMPORTANT A macOS app can't use an app group as a keychain access group.

The first two depend on the App ID prefix. If that changes, you lose access to any keychain items in those groups.

WARNING Think carefully before using the keychain to store secrets that are the only way to access irreplaceable user data. While the keychain is very reliable, there are situations where a keychain item can be lost and it's bad if it takes the user's data with it.

In some cases losing access to keychain items is not a big deal. For example, if your app uses the keychain to manage a single login credential, losing that is likely to be acceptable. The user can recover by logging in again.

In other cases losing access to keychain items is unacceptable. For example, your app might manage access to dozens of different servers, each with unique login credentials. Your users will be grumpy if you require them to log in to all those servers again.

In such situations you must carefully plan your migration. The key element here is the third item in the list above, the `com.apple.security.application-groups` entitlement. An app group is tied to your team, and so your app retains access to the corresponding keychain access group across an App ID change. This suggests the following approach:

1. Release a version of your app that moves keychain items from other keychain access groups to a keychain access group corresponding to an app group.
2. Give your users time to update to this new version, run it, and so move their keychain items.
3. When you're confident that the bulk of your users have done this, change your App ID prefix.

Be wary of the following caveats:

- This approach won't work on macOS because macOS apps can't use an app group as a keychain access group.
- It's hard to judge how long to wait at step 2.

Transfer Your App to Another Team

There is no supported way to maintain access to keychain items across an app transfer. This makes it critical that you plan the transfer carefully.

Note The approach described in the previous section doesn't work in this case because app groups are tied to a team.

There are three potential approaches here:

- Do nothing
- Do not transfer your app
- Get creative

Do Nothing

In this case the user loses all the secrets that your app stored in the keychain. This may be acceptable for certain apps. For example, if your app uses the keychain to manage a single login credential, losing that is likely to be acceptable. The user can recover by logging in again.

Do Not Transfer

Another option is to not transfer your app. Instead, ship a new version of the app from the new team and have the old app recommend that the user upgrade.

There are a number of advantages to this approach. The first is that there's absolutely no risk of losing any user data. The two apps are completely independent.

The second advantage is that the user can install both apps on their device at the same time. This opens up a variety of potential migration paths. For example, you might ship an update to the old app with an export feature that saves the user's state, including their secrets, to a suitably encrypted file, and then match that with an import facility on the new app.

Finally, this approach offers flexible timing. The user can complete their migration at their leisure.

However, there are a bunch of clouds to go with these silver linings:

- Your users might never migrate to the new app.
- If this is a paid app, or an app with in-app purchase, the user will have to buy things again.
- You lose the original app's history, ratings, reviews, and so on.

Get Creative

Finally, you could attempt something creative. For example, you might:

1. Publish a new version of the app that supports exporting the user's state, including the secrets.
2. Tell your users to do this, with a deadline.
3. Transfer the app and then, when the deadline expires, publish the new version with an import feature.

Frankly, this isn't very practical. The problem is with step 2: There's no good way to get all your users to do the export, and if they don't do it before the deadline there's no way to do it after.

Security App ID

Reply

Posted 4 months ago by eskimo

Add a Comment

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation Agreement.

Copyright © 2022 Apple Inc. All rights reserved. Terms of Use | Privacy Policy | License Agreements