

Effective in 2026, to align with our trunk stable development model and ensure platform stability for the ecosystem, we will publish source code to AOSP in Q2 and Q4. For building and contributing to AOSP, use `android-latest-release`. The `android-latest-release` manifest branch will always reference the most recent release pushed to AOSP. For more information, see [Changes to AOSP](/docs/whatsnew/site-updates#aosp-changes) (/docs/whatsnew/site-updates#aosp-changes).

Configure and handle update ownership for apps

When an app is installed by a store or installer, the store or installer is considered the "installer of record" meaning the last installer of the app. Prior to Android 14, Android allowed another store or app installer to become the installer of record and to update the app without notifying the user.

In Android 14, the initial installer of an app can declare itself "update owner" and own updates to the app. If another installer attempts to update the app, the user is given the opportunity to approve the new update before it proceeds.

Opt packages in to update ownership

To declare that a store or installer owns an app package, include the `update-ownership` tag in your `sysconfig` XML for each package as follows:

```
<update-ownership package="com.example.application" installer="com.example.installer"
```

In this example, `com.example.application` is the app package to be owned and `com.example.installer` is the owner of the package. When a package is opted in to update ownership, other privileged stores or installers have to [handle update owner and obtain user consent](#) (#handle) to update the app.

Opt packages out of ownership changes

You can have your store or installer opt a subset of packages out of update owner changes by providing a denylist in the APK. By including a package in this list, no store or installer can request update ownership of the package.

To opt packages out of being updatable by another store or installer:

1. Include the following property in the original store or installer's `AndroidManifest.xml` file:

```
<application ...>
  <property android:name="android.app.PROPERTY_LEGACY_UPDATE_OWNERSHIP_DENYLIST"
            android:resource="@xml/legacyOwnershipDenylist" />
</application>
```

This example references an XML denylist called `legacyOwnershipDenylist`.

2. Create a denylist as raw XML resource with the following format:

```
<deny-ownership>com.example.app1</deny-ownership>
<deny-ownership>com.example.app2</deny-ownership>
```

If a store or installer requests ownership of a package on a denylist, ownership won't be granted and the package is still installed but won't be owned by any installer. Further, regardless of the installer, an app on a denylist can't be owned by anyone.

The set of packages in this list can change through an update to the installer APK that provides the list. Any ownership that is set for a package that is subsequently introduced to a deny list is cleared when the installer is updated. As such, subsequent updates of the app package on the deny list won't require user interaction.

Handle update owner and obtain user consent

With Android 14, even if a store or app installer has the `android.permission.INSTALL_PACKAGES` permission, it still needs to handle the `STATUS_PENDING_USER_ACTION`

(https://developer.android.com/reference/android/content/pm/PackageInstaller#STATUS_PENDING_USER_ACTION)

state if it wants to update an app whose updates are owned by another store or installer.

The `InstallAPKSessionApi.java`

(<https://android.googlesource.com/platform/development/+android17-release/samples/ApiDemos/src/com/example/android/apis/content/InstallApkSessionApi.java>)

sample app also shows how to handle `STATUS_PENDING_USER_ACTION`.

Establish ownership for preloaded apps

Preloaded apps aren't typically owned by a specific installer. Instead preloaded apps are assigned a new owner using system configuration as shown in [Opt packages in to update ownership \(#optin\)](#)

.

Content and code samples on this page are subject to the licenses described in the [Content License \(/license\)](#). Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates.

Last updated 2026-06-17 UTC.