GUIDES

How to transfer an app to your client's account

Without republishing, without losing your reviews, and without asking anyone to reinstall. Updated for 2026.

What this is about

It happens all the time: a business's app is published under the developer account of whoever built it. Nobody looks at that while the relationship is working. When it ends, the client finds out that their app — the one they paid for — lives in someone else's account.

The good news is that **both stores have an official process for this**. It's called an app transfer, and it moves the app from one developer account to another while keeping the listing, the reviews and the installed users. Nothing gets republished and nobody has to download anything again.

The bad news is that the process has hard requirements and several ways to get stuck. This guide covers both stores, what carries over, what you lose, and what to check before you start. If you're still deciding whose name the accounts should go under, start here.

Transferring is not handing over the code

These are two different things and it's worth keeping them apart. **Transferring the app** moves the store listing between developer accounts. **Handing over the project** means the source code, the signing certificates and access to the infrastructure.

You can do one without the other, but if the goal is for the client to stand on their own they need both: without the code they can't ship an update, and without the account they can't ship anything at all.

What carries over and what doesn't

What	Google Play	App Store
Reviews and ratings	Carry over	Carry over
Installed users	Keep the app; updates reach them normally	Keep the app; updates reach them normally
Store listing	Carries over (copy, screenshots, category)	Carries over
App identifier	Carries over	The Bundle ID carries over and can no longer be changed
Subscriptions	Subscription data transfers	Transfers, but you must hand over the shared secret
Sales and earnings reports	Don't transfer; they stay in the original account	Prior history stays with the original account
Analytics	User statistics transfer	The sender loses access to App Analytics
Test groups and promo codes	Don't transfer: they have to be rebuilt	TestFlight must be emptied before transferring

The rule of thumb: **what the public sees carries over, what is commercial history stays put.** Download any reports you want to keep before starting, because you won't get them back afterwards.

When you won't be able to transfer it

Check this first, because any one of these stops the process before it starts. On the App Store:

- ✕ **The app was never released.** It needs at least one version live on the store.
- ✕ **It's in review or waiting to be released.** If the status is *Waiting for Review*, *In Review*, *Accepted*, *Pending Developer Release* or *Pending Apple Release*, you have to wait it out.
- ✕ **It has an active pre-order** in any country or region.
- ✕ **There are in-app purchases sharing an identifier** with another app in the receiving account. Product IDs can't be duplicated.
- ✕ **It's an Apple Arcade app**: those can't be transferred.
- ✕ **It's a Mac app sharing an App Group container** with other Mac apps in the same account.
- ✕ **Either account has unaccepted agreements** or a pending change of ownership.

And on Google Play

- ✕ **It's a private app** built with the managed Google Play iFrame: those can't be moved to another account.
- ✕ **The receiving account has no active payments profile** and the app is paid or has in-app purchases.
- ✕ **Either account has unresolved policy violations.**

Google Play

Step by step

The account that currently holds the app starts the process, and the receiving account approves it. Both have to be active.

1

Download anything you want to keep

Sales reports, earnings reports and bulk review exports. They stay with the original account after the transfer, but it's worth having them on hand.

This step isn't required for the transfer itself. It's just the last convenient moment to do it.

2

Get the transaction ID for both accounts

It's the identifier for the USD 25 developer registration fee. Find it in the payment email or in your Google Payments activity by searching for "developer registration fee".

This is where people get stuck: **you have to trim the beginning of the identifier**. If it starts with something like "0.G.", or has digits before the words "token" or "Registration", drop that first part and paste only the rest.

3

Sort out the connected services

If the app uses Firebase, Google Analytics or AdMob, update the permissions and links before transferring: those settings don't travel with the app.

If the app is paid or has in-app purchases, make sure the receiving account already has an active payments profile. Without it the request won't go through.

4

Submit the request

From the account that holds the app, fill in the transfer form in Play Console with both accounts' details and the transaction identifiers.

5

Have the receiving account approve it

The owner of the receiving account has to review and accept the request. Nothing happens until they do.

6

Wait for Google's review

The support team reviews and replies to transfer requests **within 2 business days**.

App Store

Step by step

On Apple, the **Account Holder** of each team starts and accepts the transfer: no other role can do it, not even an admin.

Before you start, go back over the blockers above. Apple won't let you begin a transfer while the app is in review — and a review can appear mid-process if someone uploads a build.

1 Empty TestFlight

Turn off beta testing, remove every build and tester, and clear the test information fields.

If the app uses Xcode Cloud, you also have to delete that data before transferring, from the corresponding tab in the app's settings.

2 Save what doesn't travel

The app disappears from your account once the transfer completes. Write down your bundle information and download whatever history you need.

Sales history from before the transfer stays accessible to you, but **you lose App Analytics entirely**: it goes to the receiving account.

3 If there are subscriptions, hand over the shared secret

Generate the app-specific shared secret and share it with the recipient before starting. Without it, subscription validation breaks on the server side.

Once the transfer is done, the recipient should generate a fresh one and update their servers.

4 If you use Sign in with Apple, generate the transfer identifiers

You need to generate a transfer identifier for every user in your database before moving the app. Skip it and those accounts can no longer sign in.

There's no comfortable way back from this one: it's the step that causes the most trouble in apps with Apple social login.

5 Start the transfer

The Account Holder of the account that owns the app starts the handover from App Store Connect and picks the destination team.

6 Have the receiving Account Holder accept it

On the other side, the owner of the receiving account accepts the transfer. The app stays available on the store throughout.

7 Rebuild whatever was tied to the old account

On the recipient's side now: generate new push notification credentials (APNs) and update your servers, and create a new Merchant ID if the app uses Apple Pay.

> The Apple Pay Merchant ID **does not transfer**. Transactions keep working with the original certificates, but you have to create a new one before shipping the next update.

How long it takes

Stage	Google Play	App Store
Prep (cleanup, gathering data)	1–2 hours	2–4 hours with subscriptions or Apple login
Approval from the other account	Up to the two of you	Up to the two of you
Store review	Up to 2 business days	Usually completes the same day
Rebuilding credentials and services	1 hour	2–3 hours

Checklist before you press the button

- ✓ The receiving account **already exists, is paid for and verified**. You don't open it on the spot.
- ✓ The app isn't in review or half-way through a release.
- ✓ You downloaded the reports you want to keep.
- ✓ If there are subscriptions, the shared secret has been handed over.
- ✓ If there's Sign in with Apple, the transfer identifiers are generated.
- ✓ TestFlight is empty and Xcode Cloud is clean.
- ✓ You have both Google transaction identifiers ready, already trimmed.
- ✓ You've agreed with the other side on **who hands over the source code and when**, which is the other half of this.

What if it can't be transferred?

There's always the long road: publish the app again from the client's account and take the old one down. It works, but **you lose the reviews, the ratings and the download history**, and existing users don't get the update — they have to install the new app by hand.

That's a real cost, especially for an app with a reputation built up. Which is why it pays to check the blockers early: almost all of them — a pending review, an active pre-order, a duplicated purchase identifier — are solved by waiting or changing something. They're rarely final.

We wrote this guide because we believe **every digital tool we build should be under the client's control from day one**. The full reasoning is in this note.