

[StoreKit](#) / [In-App Purchase](#) / Supporting win-back offers in your app

## Article

# Supporting win-back offers in your app

Re-engage previous subscribers with a free or discounted offer for an auto-renewable subscription, for a specific duration.

## Overview

A win-back offer is a free or discounted offer for an auto-renewable subscription for eligible customers whose subscription is *churned*. A churned subscription is one that has an [expiration Date](#) in the past, and a [willAutoRenew](#) value of `false`. You set up win-back offers in App Store Connect, including the criteria that determines whether a customer is eligible. You can base eligibility on the duration of the customer's paid subscription, the time since the subscription expired, and the time that must pass between redeemed offers. You also set the discount type, and specify the regions or countries in which the offer is available.

### Related sessions from WWDC24

Session 10110: [Implement App Store Offers](#)

You can merchandise win-back offers in multiple ways. App Store Connect provides a direct link that you can share through your own channels, such as email. Customers can also discover and redeem win-back offers in the App Store, in their Apple account in the Subscription settings, and in your app.

Within your app, you can merchandise win-back offers with the win-back offer sheet, using StoreKit views, or by using other StoreKit APIs to fully customize your own merchandising.

Your app needs to handle win-back offers that customers redeem outside the app. By default, your app receives a completed purchase transaction when someone redeems the offer outside of the app. This workflow results when the Streamlined Purchasing option in App Store Connect is on, its default state. You can choose to turn off Streamlined Purchasing to enable the customer to

complete the purchase within your app. For more information, see [Handle win-back offers redeemed outside of your app](#) below.

## Configure win-back offers in App Store Connect

Set up win-back offers in App Store Connect by following the instructions in [Set up win-back offers](#). As you set up a win-back offer, consider the following:

- You create the win-back offer identifier. This alphanumeric string appears in transaction information and subscription renewal information.
- After you confirm the win-back offer details, App Store Connect creates a numeric win-back offer Apple ID for the offer, and a direct link. The direct link uses the offer's Apple ID, and has the format `https://apps.apple.com/win-back/<win-back offer Apple ID>`.
- Add an approved subscription image in App Store Connect. The App Store requires the image if you choose to promote the win-back offer on the App Store. The win-back offer sheet and the direct link also use the image, if it's available. For more information, see [Add or remove an image](#).

### Tip

Store the win-back offer identifier and its corresponding direct link on your server, for easy access later.

## Configure Streamlined Purchasing in App Store Connect

When you set up your win-back offer, you can also set the Streamlined Purchasing option in App Store Connect. This setting applies to all win-back offers and contingent pricing in your app. It affects how your app handles a win-back offer redemption, as follows:

- With Streamlined Purchasing on (default), the customer completes the win-back offer purchase outside of your app. Your app receives a completed purchase transaction in the [updates](#) sequence if it's running, and in the [all](#) or [currentEntitlements](#) sequences otherwise. Your app is responsible for unlocking the purchased product, as usual.
- With Streamlined Purchasing turned off, your app has the opportunity to perform processing related to the win-back offer before it enables the customer to complete the purchase. The workflow starts when the customer initiates the purchase outside of your app. The system prompts them to install the app if it isn't already installed. Your app receives a `__purchase` intent in the [intents](#) sequence, instead of a transaction, and must use the [PurchaseIntent](#) API to complete the purchase.

- If customers redeem a win-back offer through the manage subscriptions page in their Apple account, the redemption always behaves as if Streamline Purchasing is on, and your app receives a completed purchase transaction.

### Important

Before you can turn off the Streamlined Purchasing option in App Store Connect, your latest approved app binary must use the [PurchaseIntent](#) API.

For more information on configuring this setting in App Store Connect, see [Manage Streamlined Purchasing](#).

## Merchandise win-back offers in your app

StoreKit provides several ways to merchandise win-back offers to eligible customers in your app. By default, StoreKit displays a win-back offer sheet when it receives a [winBackOffer](#) message from the App Store. You can also merchandise the win-back offer using StoreKit views, or other StoreKit APIs to fully customize the experience.

For information on implementing each of these options, see [Merchandising win-back offers in your app](#).

## Merchandise win-back offers outside your app using the direct link

After you create a win-back offer, App Store Connect provides a direct link. Use the direct link to merchandise the win-back offer through your own channels.

Before you send a direct link to a customer, check whether the customer is eligible to redeem the offer by calling the [Get All Subscription Statuses](#) endpoint from your server. The [JWSRenewalInfoDecodedPayload](#) from the response contains the [eligibleWinBackOfferIds](#) array, which lists the eligible offers. Send the direct link to eligible customers only.

If the [eligibleWinBackOfferIds](#) array has multiple entries, the customer is eligible for more than one offer. The App Store sets the order of the win-back offer IDs with the best offer first. The order takes into account the available offers you configure in App Store Connect for the customer's most recent subscription in the subscription group.

## Handle offers redeemed through the win-back offer sheet

When customers redeem a win-back offer using the win-back offer sheet, StoreKit treats this transaction the same as a purchase that occurs outside of the app. Specifically, the app receives the transaction in the updates sequence of Transaction. Be sure your app creates a task to iterate through updates as soon as it launches. For more information and sample code, see updates.

## Handle win-back offers redeemed outside of your app

Customers can discover and redeem win-back offers outside of your app in several ways: through the direct link that you merchandise, in the App Store, and when they manage their subscriptions in their Apple account. How your app handles the redemption depends on whether the Streamlined Purchasing setting is on or off in App Store Connect.

If Streamlined Purchasing is on (the default setting), customers complete the win-back offer purchase outside of your app. Your app receives the completed purchase transaction the next time they open the app. Treat this transaction the same as any purchase that completes outside of your app, and unlock the purchased content. Specifically:

- To get the transactions that occurred in the past, including while your app wasn't running, iterate the all or currentEntitlements sequences on Transaction when the app launches.
- To observe transactions that occur outside of the app while the app is running, iterate updates on Transaction. Create a Task to iterate through the transactions from the listener as soon as your app launches. For more information and sample code, see updates.

The transaction entitles the customer to the auto-renewable subscription. Your app unlocks the purchased content and finishes the transaction with finish().

If you turn off Streamlined Purchasing in App Store Connect, your app receives a purchase intent instead of a completed transaction. When a customer initiates a redemption outside your app, with Streamlined Purchasing off, the flow is as follows:

1. The system opens your app if it isn't already running. StoreKit doesn't display the win-back offer sheet when a purchase intent is in progress.
2. StoreKit sends the win-back offer information to the intents asynchronous sequence of PurchaseIntent.
3. Your app has the opportunity to perform any actions related to the purchase before it prompts the customer to complete the purchase. For example, your app might ask the customer to log in.
4. Your app calls the purchase(options:) method on the product of the PurchaseIntent to enable the customer to complete the purchase. Be sure to add the win-back offer to the purchase options.

The following example checks the `offer` property of `PurchaseIntent` to detect whether the purchase intent is for a win-back offer, and adds the offer to the purchase options:

```
// Prepare the purchase options.
var purchaseOptions: Set<Product.PurchaseOption> = []

// Add the win-back offer to the purchase.
if let offer = purchaseIntent.offer, offer.type == .winBack {
    purchaseOptions.insert(.winBackOffer(offer))
}

// Present the purchase.
let result = try await purchaseIntent.product.purchase(options: purchaseOptions)
```

Finally, follow the same workflow your app uses for any successful purchase, such as unlocking the purchased content and finishing the transaction with `finish()`.

#### Note

If your app uses the `PurchaseIntent` API but doesn't explicitly add support for win-back offers using `offer`, StoreKit adds the win-back offer to the purchase options on devices running iOS 18 and later and visionOS 2 and later.

## Identify subscriptions purchased with a win-back offer

When customers successfully redeem subscription offers, including win-back offers, the transaction information contains the offer details. Find those details in the app and on your server, as follows:

- In your app, check the `Transaction` property, `offer`. An offer type value of `winback` indicates the customer redeemed a win-back offer.
- On your server, check the `JWSTransactionDecodedPayload` that you receive in `App Store Server Notifications V2` notifications or `App Store Server API` responses. The `offerIdentifier` represents the offer that's applied to the transaction. An `offerType` of 4 indicates a win-back offer.

## Testing win-back offers

You can test win-back offers early in your development using StoreKit Testing in Xcode, and in the sandbox environment. For more information, see [Testing win-back offers in Xcode](#) and [Testing](#)

## See Also

### Subscription offers and offer codes

#### Merchandising win-back offers in your app

Present win-back offers to eligible customers in your app with the win-back offer sheet or by implementing custom merchandising.

#### Supporting subscription offer codes in your app

Provide subscription service for customers who redeem offer codes through the App Store or within your app.

`struct SubscriptionOffer`

Information about a subscription offer that you configure in App Store Connect.

`struct OfferType`

The types of offers for auto-renewable subscriptions.