

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

Article

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.

Overview

If a customer is eligible for a win-back offer, StoreKit provides several ways to for your app to merchandise the offer. By default, StoreKit automatically displays a win-back offer sheet to eligible customers when the app launches. You can choose to suppress (or delay) this sheet using the [Message](#) API, present the win-back offer using [StoreKit views](#), or use StoreKit APIs and customize your UI.

Related sessions from WWDC24

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

For information on configuring win-back offers, handling redeemed offers, and more, see [Supporting win-back offers in your app](#).

Present the offer with the win-back offer sheet

By default, when StoreKit receives a [winBackOffer](#) message from the App Store, it displays a win-back offer sheet when the app launches. You don't need any code to choose this default behavior.

You may choose to delay this message, or suppress it if you customize the way your app presents win-back offers. Listen for messages using the [messages](#) asynchronous sequence when your app

launches. Intercept the win-back offer message, which has a [winBackOffer](#) reason. To suppress the message, ignore it by doing nothing.

The following example code suppresses the win-back offer redemption message, and displays all other App Store messages immediately.

```
struct MessageExampleView: View {
    @Environment(\.displayStoreKitMessage) private var displayStoreMessage

    var body: some View {
        MyContentView()
            .task {
                for await message in StoreKit.Message.messages {
                    if message.reason != .winBackOffer {
                        // Ask the system to display messages now.
                        try? displayStoreMessage(message)
                    }
                }
            }
    }
}
```

For more information about App Store messages, see [Message](#).

Note

Redeeming an offer in the win-back offer sheet results in a transaction that StoreKit treats the same as transactions that occur outside of the app; your app observes the transaction in the [updates](#) sequence.

Present win-back offers in StoreKit views

StoreKit views automatically support displaying pricing details in the [SubscriptionStoreView](#) for all subscription offers the customer is eligible for, including win-back offers. The StoreKit views fetch the offer metadata and set up the purchase for you. If a customer is eligible for multiple offers, use the view modifier [preferredSubscriptionOffer\(_:\)](#) to control which offer to present.

The [preferredSubscriptionOffer\(_:\)](#) modifier takes a single parameter: a function that StoreKit calls for each subscription in the group. In this function, StoreKit provides the product object, the subscription information, and a list of all offers the customer is eligible for. Use this information and your own logic to decide which offer to present, and return its [Product](#)

`.SubscriptionOffer` object. If you return `nil` and the customer is eligible for an introductory offer, the app always displays the introductory offer in the view.

Customize a win-back offer experience

If you choose to fully control the in-app merchandising experience, use the following APIs to get information about the customer's eligibility for win-back offers and the offer details:

- Check the `eligibleWinBackOfferIDs` in the `Product.SubscriptionInfo.RenewalInfo` to get the list of the product's win-back offers the customer is eligible for. If the customer is eligible for multiple offers, the App Store sorts this array with the best offer first.
- Get the win-back offer details from the `winBackOffers` array in `Product.SubscriptionInfo`, which contains all the available win-back offers that you configure in App Store Connect. Compare the IDs from the `eligibleWinBackOfferIDs` array to the `id` of the `Product.SubscriptionOffer` in the `winBackOffers` array.

To add a win-back offer to a purchase, include the win-back offer the customer chooses in the purchase options, `purchase(options:)`.

See Also

Subscription offers and offer codes

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.

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.