

[StoreKit](#) / [PurchaseIntent](#)

Structure




PurchaseIntent

An instance that emits purchase intents, which indicate that the customer initiated a purchase outside of your app, for your app to complete.

iOS 16.4+ | iPadOS 16.4+ | Mac Catalyst 16.4+ | macOS 14.4+

```
struct PurchaseIntent
```

Mentioned in

-  [Supporting win-back offers in your app](#)
-  [Supporting promoted In-App Purchases in your app](#)
-  [Promoting In-App Purchases](#)

Overview

Using `PurchaseIntent` is required in the following cases:

- If you promote in-app purchases on the App Store
- If you turn off the Streamlined Purchasing setting in App Store Connect and use win-back offers or contingent pricing

You set up these offers and settings in App Store Connect. For more information, see [Promote in-app purchases](#), [Set up win-back offers](#), and [Manage Streamlined Purchasing](#).

Important

To enable promoted in-app purchases, your app must use either [PurchaseIntent](#) (starting in iOS 16.4) or [paymentQueue\(_:shouldAddStorePayment:for:\)](#) (starting in iOS 11). Don't use both at the same time. If necessary, use conditional compilation to identify the OS version the app is running in. For more information, see [Running code on a specific platform or OS version](#).

When a customer selects a promoted product on the App Store or a win-back offer outside of your app, it initiates a purchase that they complete in your app. If they don't have your app installed, the system prompts them to download it. After the app downloads, the transaction continues in the app, which receives a purchase intent from the [intents](#) sequence.

The purchase intent identifies the [product](#) the customer selected. To enable customers to complete the purchase, call [purchase\(options:\)](#) on this product instance. The following example code receives the purchase intent, and calls a method to complete the purchase workflow:

```
func purchaseProduct(_ product: Product) async {
    // Complete the purchase workflow.
    do {
        try await product.purchase()
    }
    catch {
        // Add your error handling here.
    }
    // Add your remaining purchase workflow here.
}

for await purchaseIntent in PurchaseIntent.intents {
    // Complete the purchase workflow.
    await purchaseProduct(purchaseIntent.product)
}
```

For more information, see [Supporting promoted In-App Purchases in your app](#) and [Supporting win-back offers in your app](#).

In Mac apps built with Mac Catalyst, this instance doesn't emit purchase intents for promoted in-app purchases because the Mac App Store doesn't support this feature.

Note

PurchaseIntent and promoted in-app purchases aren't available to compatible iPad or iPhone apps running in visionOS.

Topics

Identifying the product

```
var id: Product.ID
```

The product identifier of the In-App Purchase that the customer selects to purchase outside of the app.

```
let product: Product
```

The product information of the In-App Purchase the customer selects to purchase outside of the app.

Getting purchase intents

```
static var intents: PurchaseIntent.PurchaseIntents
```

The asynchronous sequence that emits a purchase intent when customers initiate a purchase outside of the app.

```
struct PurchaseIntents
```

An asynchronous sequence of purchase intents for purchases that customers initiate outside of the app.

Identifying the offer

```
let offer: Product.SubscriptionOffer?
```

The subscription offer that the customer redeems outside of your app.



Relationships

Conforms To

Copyable
Equatable
Identifiable
Sendable
SendableMetatype

See Also

Promoted In-App Purchases

-  Supporting promoted In-App Purchases in your app
- Display promoted In-App Purchases on your product page and handle purchases that users initiate on the App Store.
- `struct PromotionInfo`
- Information about a promoted In-App Purchase that customizes its order and visibility on the device.
-  Testing promoted In-App Purchases
- Test your In-App Purchases before making your app available in the App Store.