Article

Supporting business model changes by using the app transaction

Access the app transaction to determine when a customer purchased an app and the features to which they're entitled.

Overview

The `originalAppVersion` property indicates the app version that the customer purchased. If you change your business model from a paid app to a free app that offers in-app purchases, use this property to learn whether the customer purchased your app before you changed the business model. Then, use that information to determine your customers' entitlement to features that were available in the paid app.

Provide features to all customers

For example, an app that is a paid app in version 1 has premium features available to everyone who buys it. For version 2, the developer changes the business model, making it a free app that offers in-app purchases. Version 2 of the app has the same premium features, but now they're available as in-app purchases.

In version 2, the developer wants to continue to provide the premium features to customers who purchased version 1. To do so, the app performs the following steps:

1. The app's code includes a constant that indicates the version the business model changed; that constant is `"2"` in this example.

2. The app compares the `originalAppVersion` value with the constant. If the customer purchased the app before the business model changed, the app determines that they're entitled to the premium features.

3. The app also checks the `currentEntitlements` sequence and delivers any in-app purchases the customers may have made.

Obtain an app transaction

An app that performs these steps ensures that paid customers can access the premium features that were included with the app they purchased.

The code examples below demonstrate how an app obtains an `AppTransaction`, compares the `originalAppVersion` with a constant that represents a specific app version, and then determines the customer’s entitlements.

Here’s how it looks in macOS:

```
do {
    // Get the `appTransaction`.
    let shared = try await AppTransaction.shared
    if case .verified(let appTransaction) = shared {
        // Hard-code the major version number in which the app's business model char
        let newBusinessModelMajorVersion = "2"

        // Get the major version number of the version the customer originally purch
        let versionComponents = appTransaction.originalAppVersion.split(separator: '
        let originalMajorVersion = versionComponents[0]

        if originalMajorVersion < newBusinessModelMajorVersion {
            // This customer purchased the app before the business model changed.
            // Deliver content that they're entitled to based on their app purchase.
        }
        else {
            // This person purchased the app after the business model changed.
        }
    }
}
catch {
    // Handle errors.
}



// Iterate over any other products they purchased.
for await result in Transaction.currentEntitlements {
    if case .verified(let transaction) = result {
        // Deliver the content based on their current entitlements.
    }
}
```

Here’s an example of the code for an app in iOS, tvOS, watchOS, and visionOS:

```
do {
    // Get the `appTransaction`.
    let shared = try await AppTransaction.shared
    if case .verified(let appTransaction) = shared {
        // Hard-code the `CFBundleVersion` number in which the app's business model
        let newBusinessModelVersion = "2"

        // Compare this number with the version number the person originally purchas
        if appTransaction.originalAppVersion < newBusinessModelVersion {
            // Handle a case in which a person purchased the app before the business
            // Deliver content that they're entitled to based on their app purchase.
        }
        else {
            // This person purchased the app after the business model changed.
        }
    }
}
catch {
    // Handle errors.
}

// Iterate over any other products they purchased.
for await result in Transaction.currentEntitlements {
    if case .verified(let transaction) = result {
    // Deliver the content based on their current entitlements.
    }
}
```

Related sessions from WWDC22

Session 10007: What’s new with in-app purchase

See Also

App transaction

```
struct AppTransaction
```

Information that represents the customer’s purchase of the app, cryptographically signed by the App Store.