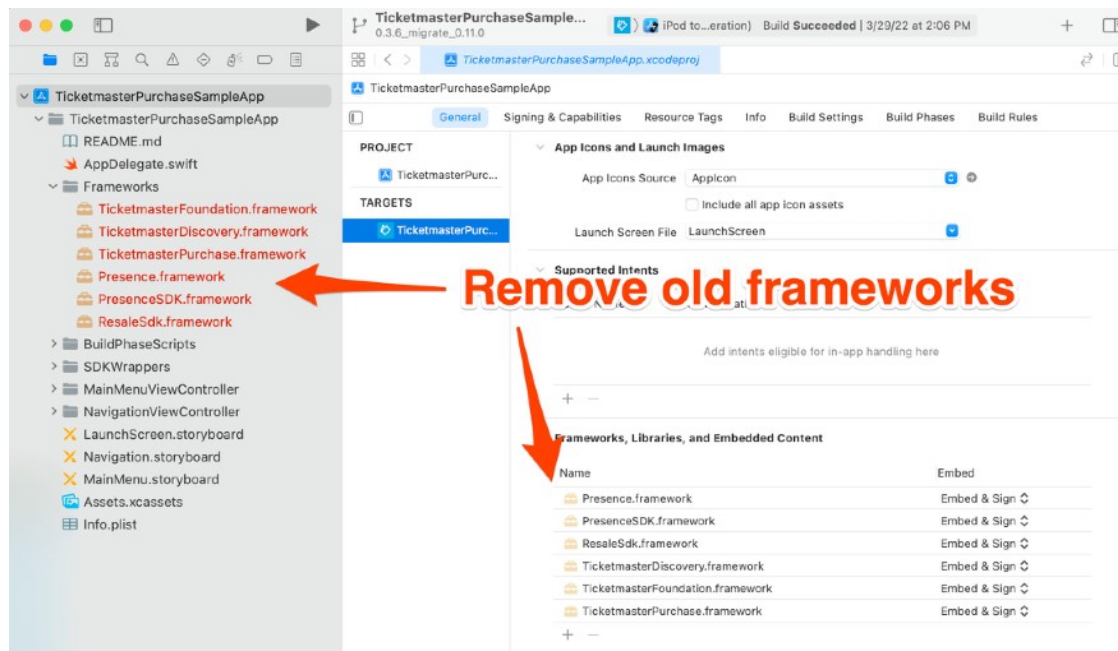


# Purchase SDK 0.3.6 to 0.11.0 Migration Guide

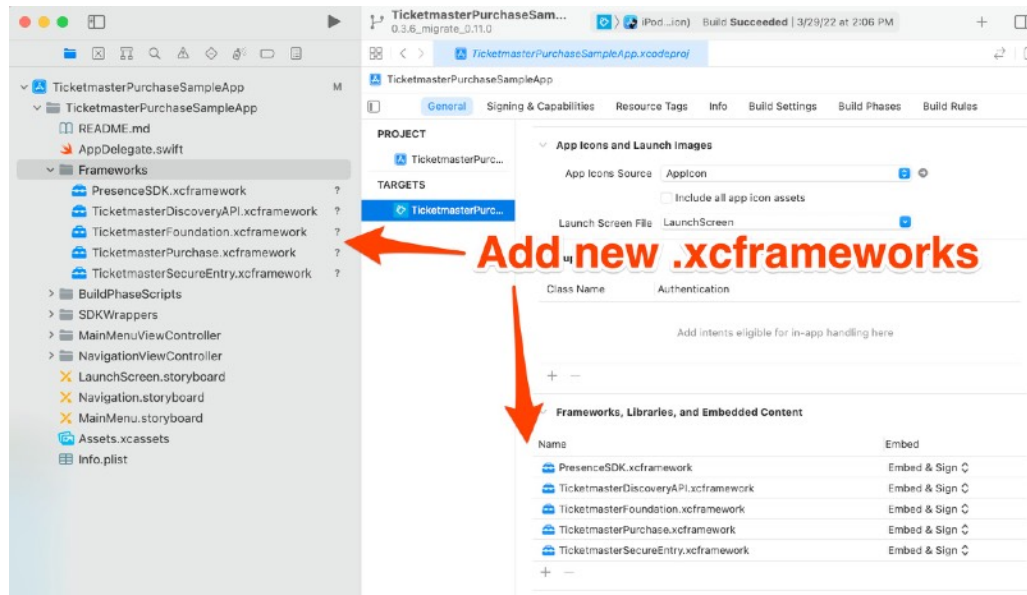
## 1. Remove old .framework files

1. Delete all of these Ticketmaster frameworks from your Project repo file system:
  1. **TicketmasterFoundation.framework**
  2. **TicketmasterDiscovery.framework**
  3. **TicketmasterPurchase.framework**
  4. **Presence.framework**
  5. **PresenceSDK.framework**
  6. **iosExperienceSDK.framework**
  7. **ResaleSdk.framework**
2. Delete all of these same frameworks from your Project file as well:



## 2. Add new .xcframework files

1. Copy the required Ticketmaster xcframeworks
  1. Required for both Presence SDK and Purchase SDK:
    1. **TicketmasterFoundation**
    2. **TicketmasterSecureEntry**
    3. **PresenceSDK**
  2. If using Purchase SDK, also add:
    1. **TicketmasterDiscoveryAPI**
    2. **TicketmasterPurchase**
  3. If using Discovery SDK, also add:
    1. **TicketmasterPrePurchase**
2. Please make sure that:
  1. Frameworks appear in the Project Navigator list
  2. Frameworks appear in the Framework list as “Embed & Sign”



### 3. Make Code Changes

If you are only using the Presence SDK, you are already done.

However, if you are also using the Purchase SDK, the newest version has some changes:

1. Presence.framework is now named TicketmasterSecureEntry.xcframework
2. TicketmasterPurchase.xcframework now requires TicketmasterDiscoveryAPI.xcframework
3. TMPurchaseNavigationController.userNavigationDelegate has been split into two delegates:
  1. TMPurchaseNavigationController.userAnalyticsDelegate
  2. TMPurchaseNavigationController.navigationDelegate
4. PresenceSDK now directly supports Purchase OAuth delegate. Setting this directly is highly recommended:
  1. **`purchaseNavigationController.oauthDelegate = PSDK.shared.oauthProvider`**
5. TicketmasterDiscovery.framework is now named TicketmasterPrePurchase.xcframework
6. iosExperienceSDK.framework and ResaleSdk.framework are no longer used

At this point you have two choices:

1. Re-integrate the SDKs using the latest guides:
  1. Quick Start Guide
  2. Purchase Sample App
  3. Presence Demo App
2. Modify your existing code to compile with the new changes shown in the following sections

Note that while most of the Purchase delegate methods have the same overall form, there are changes to types. For example:

```
TMPurchaseViewControllerUserNavigationDelegate {  
    func purchaseViewController(_ purchaseViewController: TMPurchaseViewController,  
        didEndTicketSelectionFor event: TMEvent,  
        because reason: TMEndTicketSelectionReason)
```

Is changed to:

```
TMPurchaseUserAnalyticsDelegate {  
    func purchaseNavigationController(_ purchaseNavigationController: TMPurchaseNavigationController,  
        didEndTicketSelectionFor event: DiscoveryEvent,  
        because reason: TMEndTicketSelectionReason)
```

There are the occasional, slight changes to method form as well. For example:

```
TMPurchaseViewControllerSharingDelegate {  
    func getSharingMessage(for purchaseViewController: TMPurchaseViewController,  
        event: TMEvent, completion: @escaping (String) -> Void)
```

Is changed to:

```
TMPurchaseSharingDelegate {  
    func purchaseNavigationController(_ purchaseNavigationController: TMPurchaseNavigationController,  
        sharingMessageFor event: DiscoveryEvent, completion: @escaping (String) -> Void)
```

It is recommended that you re-implement all protocol delegate methods just to be safe.

### 3. PresenceSDK Changes

The old names have been deprecated, but are still available for use:

PresenceSDKView

=> PresenceView

PSDK.shared.start(presenceSDKView, ...)

=> PSDK.shared.start(presenceView, ...)

## 4. TicketmasterPurchase Changes (if used)

TicketmasterPurchase now requires TicketmasterDiscoveryAPI for certain delegates:

TMPurchase	=>	TMPurchase.shared
TMPurchase.isDebugMode	=>	TMPurchase.shared.useDebugAnalytics
TMPurchaseViewController	=>	TMPurchaseNavigationController
TMPurchaseViewController	=>	TMPurchaseNavigationController
TMPurchaseViewController.oAuthDelegate	=>	TMPurchaseNavigationController.oauthDelegate
TMPurchaseViewControllerOAuthDelegate	=>	TicketmasterOAuthDelegate protocol requires: import TicketmasterFoundation
TMPurchaseViewController.userNavigationDelegate	=>	TMPurchaseNavigationController.userAnalyticsDelegate
	=>	TMPurchaseNavigationController.navigationDelegate
TMPurchaseViewControllerUserNavigationDelegate	=>	TMPurchaseUserAnalyticsDelegate
	=>	TMPurchaseNavigationDelegate requires: import TicketmasterDiscoveryAPI
TMPurchaseViewControllerSharingDelegate	=>	TMPurchaseSharingDelegate requires: import TicketmasterDiscoveryAPI
TMUserToken	=>	TicketmasterOAuthToken requires: import TicketmasterFoundation
TMOrder	=>	TMPurchaseOrder

PresenceSDK now directly supports Purchase OAuth delegate. Setting this directly is highly recommended:

**`purchaseNavigationController.oauthDelegate = PSDK.shared.oauthProvider`**

## 5. TicketmasterDiscovery Changes

TicketmasterDiscovery has been renamed to TicketmasterPrePurchase:

import TicketmasterDiscovery	=>	import TicketmasterPrePurchase
TMDiscovery	=>	TMPrePurchase.shared
TMDiscoveryViewController	=>	TMPrePurchaseViewController
TMDiscoveryNavigationDelegate	=>	TMPrePurchaseNavigationDelegate
TMDiscoveryAnalyticsDelegate	=>	TMPrePurchaseAnalyticsDelegate
DiscoveryPage	=>	PrePurchasePage