IOS Native Unity3D Plugin

Anyone can comment

Overview

plugin provides the easy and flexible functionality of IOS native functions which are not available from Unity, including Game Center, in-app purchases, & native alerts and events.

Please read Plugin Setup Guide before using the plugin.

If you're new to IOS app development, please also read IOS Application Setup Guide.

Native Pop Up's

description of IOSRateUsPopUp, IOSDialog, IOSRateUsPopUp

IOS Rate Pop Up

Pop up creation:

```
IOSRateUsPopUp rate = IOSRateUsPopUp.Create();
```

Rate pop up will appear after this lines, if you want to listen rate pop up events you should add COMPLETE listener to it.

```
rate.addEventListener(BaseEvent.COMPLETE, onRatePopUpClose);
```

example of onRatePopUpClose function:

```
private void onRatePopUpClose(CEvent e) {
        (e.dispatcher as IOSRateUsPopUp)
            .removeEventListener(BaseEvent.COMPLETE, onRatePopUpClose);
        string result = e.data.ToString();
        IOSNative.showMessage("Result", result + " button pressed");
}
```

IOSDialogResult result can contain: `RATED,REMIND,DECLINED` of `IOSDialogResult` class.

IOS Dialog Pop Up

Creation:

```
IOSDialog dialog = IOSDialog.Create("Dialog Titile", "Dialog message");
```

Listeners:

```
dialog.addEventListener(BaseEvent.COMPLETE, onDialogClose);
```

onDialogClose function example:

```
private void onDialogClose(CEvent e) {

        //romoving listner
        (e.dispatcher as IOSDialog).removeEventListener(BaseEvent.COMPLETE,
onDialogClose);

        //parsing result
        switch((IOSDialogResult)e.data) {
        case IOSDialogResult.YES:
            Debug.Log ("Yes button pressed");
            break;
        case IOSDialogResult.NO:
            Debug.Log ("Yes button pressed");
            break;

        }

        string result = e.data.ToString();
        IOSNative.showMessage("Result", result + " button pressed");
  }
```

IOSDialogResult result can contain: `YES,NO` of `IOSDialogResult` class.

IOS Message Pop Up

Creation:

```
IOSMessage msg = IOSMessage.Create("Message Titile", "Message message");
```

Lisners:

```
msg.addEventListener(BaseEvent.COMPLETE, onMessageClose);
```

onDialogClose function example:

```
private void onMessageClose(CEvent e) {
        (e.dispatcher as IOSMessage).removeEventListener(BaseEvent.COMPLETE,
onMessageClose);
        IOSNative.showMessage("Result", "Message Closed");
 }
```

Game Center

Classes Documentation

GameCenterManager class.

Starts Game Center. Should be called on startup.

Triggers GAME_CENTER_PLAYER_AUTHENTICATED event

public static void init()

Show leaderboard UI by its ID

public static void showLeaderBoard(string leaderBoradrId)
public static void showLeaderBoard(string leaderBoradrId, GCBoardTimeSpan scope)

Show leaderboards

public static void showLeaderBoards()

Submit Score to leaderboard by it ID

public static void reportScore(int score, string leaderBoradrId)

Send leaderboard score request by leaderboard ID. Triggers GAME_CENTER_LEADER_BOARD_SCORE_LOADED event

public static void getScore(string leaderBoradrId, GCBoardTimeSpan scope, GCCollectionType collection)

Show Achievements UI

public static void showAchievements()

Reset All Achievements progress

public static void resetAchievements()

Submit Achievement progress

public static void submitAchievement(int percent, string achievementId)

public static void submitAchievement(int percent, string achievementId, bool isCompleteNotification)

Gets achievement progress by it ID

public static float getAchievementProgress(string id)

Gets leader board by id if it was previously loaded

public static GCLeaderBoard GetLeaderBoard(string id)

Gets player by id if it was previously loaded. All player are loaded automatically it they scores was loaded. If you got player score, you can get it data, but sometimes it can take some time to load, especially if player has avatar. See CustomLeaderBoardGUIExample scene, to find out more.

public static GameCenterPlayerTemplate GetPlayerById(string playerID)

Getters:

Achievements array

public static List<AchievementTemplate> achievements {get;}

Current player info

public static GameCenterPlayerTemplate player

Event dispatcher. Should be used for event listening

public static EventDispatcherBase dispatcher

Events:

Fires when local player is authed. Event does not contain any data.

GAME_CENTER_PLAYER_AUTHENTICATED

Fires when player has canceled authentication flow or any other error occurred during authentication process. Event does not contain any data.

GAME_CENTER_PLAYER_AUTHENTIFICATION_FAILED

Fires on leaderboard score loaded. Event data contains LeaderBoardScoreData.

GAME_CENTER_LEADER_BOARD_SCORE_LOADED

Fires on achievement progress. Event data contains AchievementTemplate.

GAME_CENTER_ACHIEVEMENT_PROGRESS

Fires when achievement data is loaded. Event does not contain any data

GAME_CENTER_ACHIEVEMENTS_LOADED

GameCenterMultiplayer class.

Start looking for a match. MATCH_STARTED event will be called as soon as mutch is found.

public void FindMatch(int minPlayers, int maxPlayers)

Sends data to all match players

public void SendDataToAll(byte[] buffer, int sendType)

Sends data to specified players

```
public void sendDataToPlayers(byte[] buffer, int sendType, params object[] players)
```

Disconnects from current match

```
public void disconnect()
```

Getters:

Current match data

```
public GameCenterMatchData match
```

Events:

Fires when new player has connected to game. Event contains player id as data

PLAYER_CONNECTED

Fires when new player has disconnected to game. Event contains player id as data

PLAYER_DISCONNECTED

Fires when player match started. Event contains GameCenterMatchData as data.

MATCH_STARTED

Fires when new data received. Event contains GameCenterDataPackage as data.

DATA_RECIVED

GameCenterDataPackage class.

Getters:

Contains player id

`public string playerID`

Contains package data

`public byte[] buffer`

GameCenterMatchData class.

Getters:

Match player ids

`public List<string> playerIDs`

GameCenterPlayerTemplate class.

Getters:

Current player id

`public string playerId`

player name

`public string displayName`

player alias

`public string alias`

player avatar. Can be null if player do not have avatar.

public Texture2D avatar

GCLeaderBoard class.

Getters:

leaderboard id

public string id

get scores by rank, scope and collection.

public GCScore GetScore(int rank, GCBoardTimeSpan scope, GCCollectionType collection)

AchievementTemplate class.

Achievement id

public string id;

Achievement progress

public float progress;

LeaderBoardScoreData class.

Contains leader boards id

```
public string leaderBoardId;

Leader board scores in different formats
public string leaderBoardScore;
public float GetFloatScore()
public int GetIntScore()
```

Initialization

You should call `GameCenterManager.init();` before you start working with GameCenterManager functions.

Show Leader Board GUI

```
string leaderBoardId =  "your.leaderbord.id.here";

GameCenterManager.showLeaderBoard(leaderBoardId);
```

Reporting score to Leader Board

```
int hiScore = 100;

string leaderBoardId =  "your.leaderbord.id.here";

GameCenterManager.reportScore(hiScore, leaderBoardId);
```

Achievements registration

I recommend to register all achievements in GameCenterManager. If you will skip this step GameCenterManager.achievements array will contain only achievements with reported progress.

Here is example of how to register your achievements. You should do this before GameCenterManager.init() call

```
GameCenterManager.registerAchievement (TEST_ACHIEVEMENT_1_ID);

GameCenterManager.registerAchievement (TEST_ACHIEVEMENT_2_ID);
```

Show achievements GUI

```
GameCenterManager.showAchievements();
```

Report achievement progress

```
GameCenterManager.submitAchievement(88.66f, TEST_ACHIEVEMENT_2_ID);
```

Report achievement progress without showing achievement complete message

```
GameCenterManager.submitAchievement(88.66f, TEST_ACHIEVEMENT_2_ID, false);
```

Game Center Achievements events

GAME_CENTER_ACHIEVEMENTS_LOADED - Will be called after achievements progress will be loaded from IOS Game Center server.

Example how to listen event:

```
GameCenterManager.dispatcher.addEventListener
(GameCenterManager.GAME_CENTER_ACHIEVEMENTS_LOADED, OnAchievementsLoaded);
```

```
private void OnAchievementsLoaded() {
        Debug.Log ("Achievemnts was loaded from IOS Game Center");

        foreach(AchievementTemplate tpl in GameCenterManager.achievements) {
            Debug.Log (tpl.id + ":  " + tpl.progres);
        }
    }
```

GAME_CENTER_ACHIEVEMENT_PROGRESS - Will be called after achievement progress has changed.

Example how to listen event:

```
GameCenterManager.dispatcher.addEventListener
(GameCenterManager.GAME_CENTER_ACHIEVEMENT_PROGRESS, OnAchievementProgress);

private void OnAchievementProgress(CEvent e) {
        Debug.Log ("OnAchievementProgress");

        AchievementTemplate tpl = e.data as AchievementTemplate;
        Debug.Log (tpl.id + ":  " + tpl.progres);
    }
```

GAME_CENTER_ACHIEVEMENTS_RESET - Will be called when your game resets a progress on achievements.

```
GameCenterManager.dispatcher.addEventListener
(GameCenterManager.GAME_CENTER_ACHIEVEMENTS_RESET, OnAchievementsReset);


private void OnAchievementsReset() {
     Debug.Log ("All  Achievemnts was reseted");
 }
```

In-App Purchases

Classes Documentation

IOSInAppPurchaseManager class.

Loads store data, should be used after all products are registred. Use ***addProductId*** *function to register product before loading store*

public void loadStore()

Registers product in store. Use this function before loading store

public static void addProductId(string productId)

Initialize purchase flow by product id. Triggers PRODUCT_BOUGHT or TRANSACTION_FAILED event

public static void buyProduct(string productId)

Restore all customer purchases. You will get PRODUCT_BOUGHT event for each already bought product.

public static void restorePurchases()

Initialize verification request for last purchased product by verification URL. If you are not using your own server, use SANDBOX_VERIFICATION_SERVER url for testing mode and APPLE_VERIFICATION_SERVER url for release mode

public void verifyLastPurchase(string url)

Getters:

List of products

public List<ProductTemplate> products

Events:

Fires when purchase flow is successfully finished.

Event data contains IOSStoreKitResponce

PRODUCT_BOUGHT

Fires when purchase flow fails.

Event data contains IOSStoreKitResponce

TRANSACTION_FAILED

Fires on verification server response.

Event data contains IOSStoreKitVerificationResponce

VERIFICATION_Responce

Fires when achievement's data loaded. Event does not contain any data

GAME_CENTER_ACHIEVEMENTS_LOADED

ProductTemplate class.

product id

public string id

product title

public string title

product description

public string description

product price string

public string price

product localized price string

public string localizedPrice

IOSStoreKitResponce class.

product id

public string productIdentifier;

transaction recipe

public float receipt;

error description

public string error;

IOSStoreKitVerificationResponce class.

response status

```
public string status;
```

transaction recipe

```
public float receipt;
```

original response JSON string, use it if you need additional data. Full response data description can be found here.

```
public string originalJSON;
```

Transactions Validation

How can I validate transactions that have already completed?

Non-consumables: Set aside the current receipts, perform a restore operation, and validate the new receipts.

To do this have a look on restorePurchases function of InAppPurchaseManager class

Consumables: If you have saved the receipts, either on the device or on your server, revalidate the receipts after implementing your mitigation strategy. If you have not saved the receipts, you cannot validate these past transactions; you should not take any action.

When you got PRODUCT_BOUGHT event from the InAppPurchaseManager class, it contains IOSStoreKitResponse as event data. From IOSStoreKitResponse you can get transaction recipe, send it to your server and validate transaction there.

Note: Apple's official recommendation to perform receipt validation is to connect to your own server, which then connects to Apple's servers to validate the receipts.

For a number of reasons, this is more secure than connecting to Apple directly.

If you do not want to use your server you can use apple server for transaction validation.

After you got PRODUCT_BOUGHT event call verifyLastPurchase(string url) function of InAppPurchaseManager class. It will launch verification request. You will get VERIFICATION_RESPONSE when request is complete. Event contains IOSStoreKitVerificationResponse data, with information about transaction from apple server. Learn more here

Warning: Use SANDBOX_VERIFICATION_SERVER url (https://sandbox.itunes.apple.com/verifyReceipt) during app testing and APPLE_VERIFICATION_SERVER url (https://buy.itunes.apple.com/verifyReceipt) on production.

In-Apps Initialization

Define your product ids

```
public const string SMALL_PACK     =  "your.in.app.purchase.id1";

public const string NC_PACK     =  "your.in.app.purchase.id2";
```

Put them to the store request:

```
InAppPurchaseManager.instance.addProductId(SMALL_PACK);

InAppPurchaseManager.instance.addProductId(NC_PACK);
```

Listeners:

```
InAppPurchaseManager.instance.addEventListener(InAppPurchaseManager.PRODUCT_B
OUGHT, onProductBought);

InAppPurchaseManager.instance.addEventListener(InAppPurchaseManager.TRANSACTI
ON_FAILED, onTransactionFailed);
```

Load Store Data

```
InAppPurchaseManager.instance.loadStore();
```

onProductBought function example:

```
 private static void onProductBought(CEvent e) {

         string productId = (e.data as
 IOSStoreKitResponse).productIdentifier;
         Debug.Log("STORE KIT GOT BUY: " + productId);

         IOSNative.showMessage("Success", "product " + productId + " is
 purchased");
 }
```

onTransactionFailed function example:

```
private static void onTransactionFailed() {
      IOSNative.showMessage("Fail", "Transaction was failed");
}
```

iCloud

Classes Documentation

iCloudManager class.

Initialize iCloud. Best practice to call this on app startup. You will recive CLOUD_INITIALIZED or CLOUD_INITIALIZE_FAILED event as result.

public void init()

Sets string value for specified key

public void setString(string key, string val)

Sets float value for specified key

public void setFloat(string key, float val)

Sets bytes data for specified key

public void setData(string key, byte[] val)

Retrieves data for specified key. You will get CLOUD_DATA_RECEIVE event as result

public void requestDataForKey(string key)

Events:

Fires on successful iCloud initialization. Event does not contain any data.

CLOUD_INITIALIZED

Fires if iCloud initialization failed. Event does not contain any data.

CLOUD_INITIALIZE_FAILED

Fires if Cloud data change detected. Event does not contain any data.

CLOUD_DATA_CHANGED

Fires when requested cloud data received. Event contains iCloudData class as data

CLOUD_DATA_RECEIVE

iCloudData class.

Getters:

Data key

public string key

String representation of data

public string stringValue

Float representation of data

public string floatValue

Bytes representation of data

public string bytesValue

TRUE if key has empty data

public string IsEmpty

Other features

---

iAd App Network

iAdBannerController class.

Initialize iAd banner with ahchor. Triggers AD_LOADED, AD_VIEW_LOADED or FAIL_TO_RECEIVE_AD events.

public iAdBanner CreateAdBanner (TextAnchor anchor)

Initialize iAd banner with ahchor. Triggers AD_LOADED, AD_VIEW_LOADED or FAIL_TO_RECEIVE_AD events.

public iAdBanner CreateAdBanner (int x, int y)

Destroying banner by it's id.

public void DestroyBanner(int id)

Function will start interstitial ad request, and will show interstitial ad as soon as it's loaded.

public void StartInterstitialAd()

Function will start interstitial ad request.

public void LoadInterstitialAd()

Shows interstitials ad if it was pervoistly loaded.

public void ShowInterstitialAd()

Events:

Fires when receiving interstitials ad is failed.
INTERSTITIAL_DID_FAIL_WITH_ERROR

Fires if interstitials ad successfully loaded.
INTERSTITIAL_AD_DID_LOAD

Fires on interstitials ad will load.
INTERSTITIAL_AD_WILL_LOAD

Fires on interstitials ad view finish
INTERSTITIAL_AD_ACTION_DID_FINISH

GoogleMobileAdBanner(AdMob) interface.

API methods:

Hide ad banner
public void HideAd()

Show ad banner (only if it was hided by **HideAd** *function)*
public void ShowAd()

Get / Set:

Banner id
int id {get;}

true if banner was loaded
bool IsLoaded {get;}

true if banner currently on screen
bool IsOnScreen {get;}

Defines show or not banner when it’s loaded.
bool ShowOnLoad{get; set;}

Events:

Called when an ad is received
AD_LOADED

Called when an ad request failed
FAIL_TO_RECEIVE_AD

Called when the user is about to return to the application after clicking on an ad.
AD_VIEW_FINISHED

Social Sharing

IOSSocialManager class.

Post to twitter

```
public void TwitterPost(string text)
```

Post to twitter with image

```
public void TwitterPost(string text, Texture2D texture)
```

Post to facebook

```
public void FacebookPost(string text)
```

Post to facebook with image

```
public void FacebookPost(string text, Texture2D texture)
```

Local and Push Notifications

You do not need any additional set up to use **local** notification. However **push** notifications requires a lot more set up actions. Please follow instructions to find out how to setup and test push notification using this plugin.

IOSNotificationController class.

Schedule simple local notification. ***time*** *- seconds befor notification will be fired.* ***message*** *- notification message.*

Note: Notification will be fired only if app in background or closed.

public void ScheduleNotification(int time, string message)

Register application for receiving push notifications.

public void RegisterForRemoteNotifications(RemoteNotificationType notificationTypes)

Shows notification banner with specified title and message.

public void ShowNotificationBanner (string title, string messgae)

Events:

Fires when device token for push notification is received. Event contains IOSNotificationDeviceToken as data.

DEVICE_TOKEN_RECIVED

IOSNotificationDeviceToken class.

Getters:

String representation of device token

```
public string tokenString
```

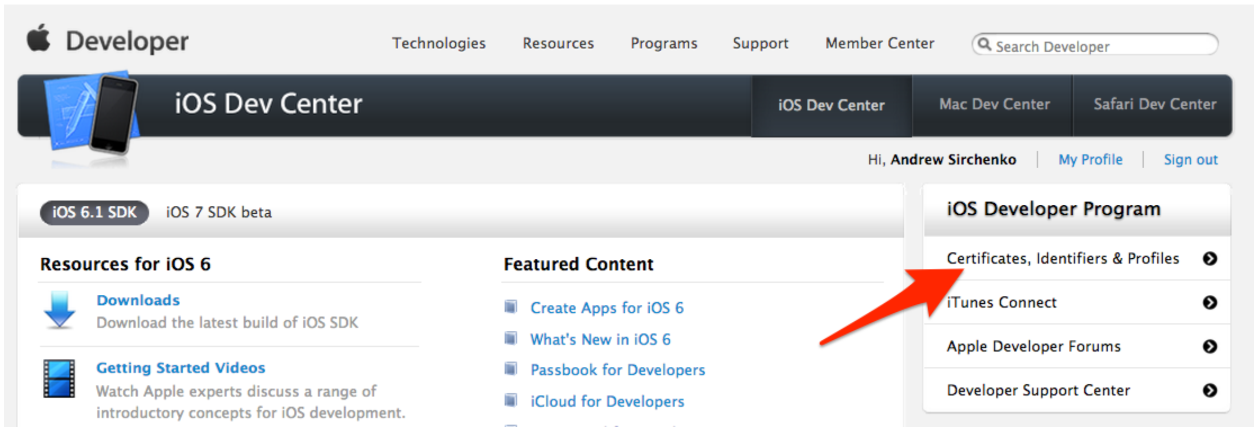
Byte representation of device token

```
public byte[] tokenBytes
```

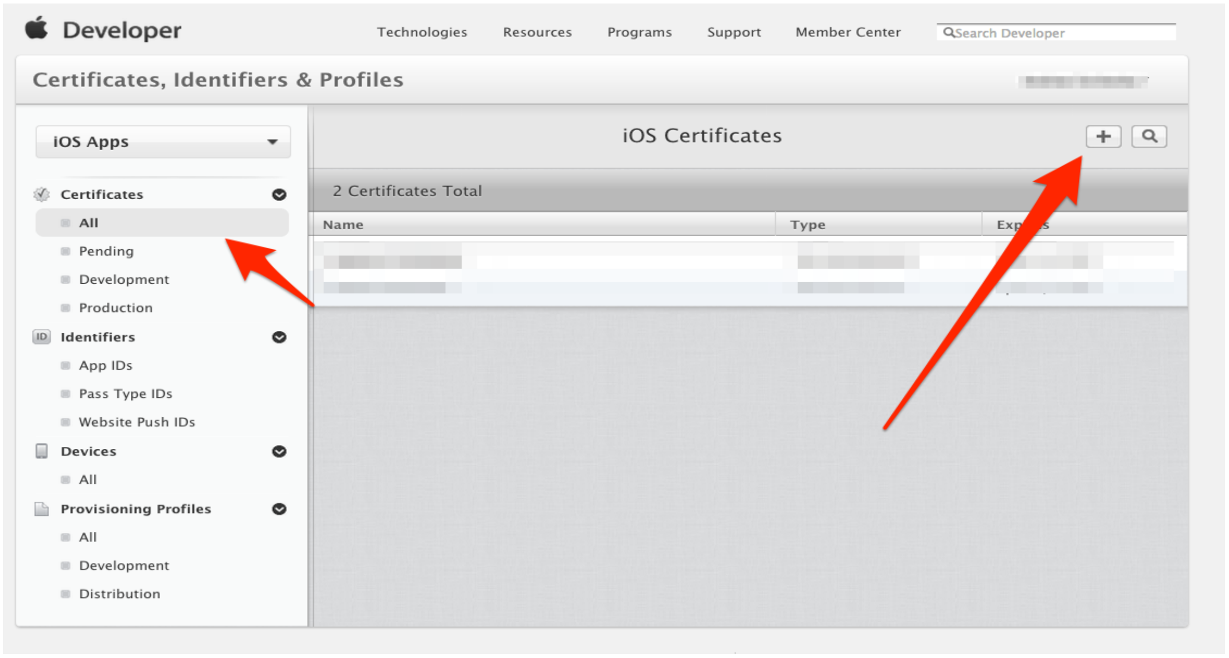
Guide's

Creating Certificate and Provocation profile

Point your browser to the IOS Developer website. Login to your developer account and navigate to the provision center.

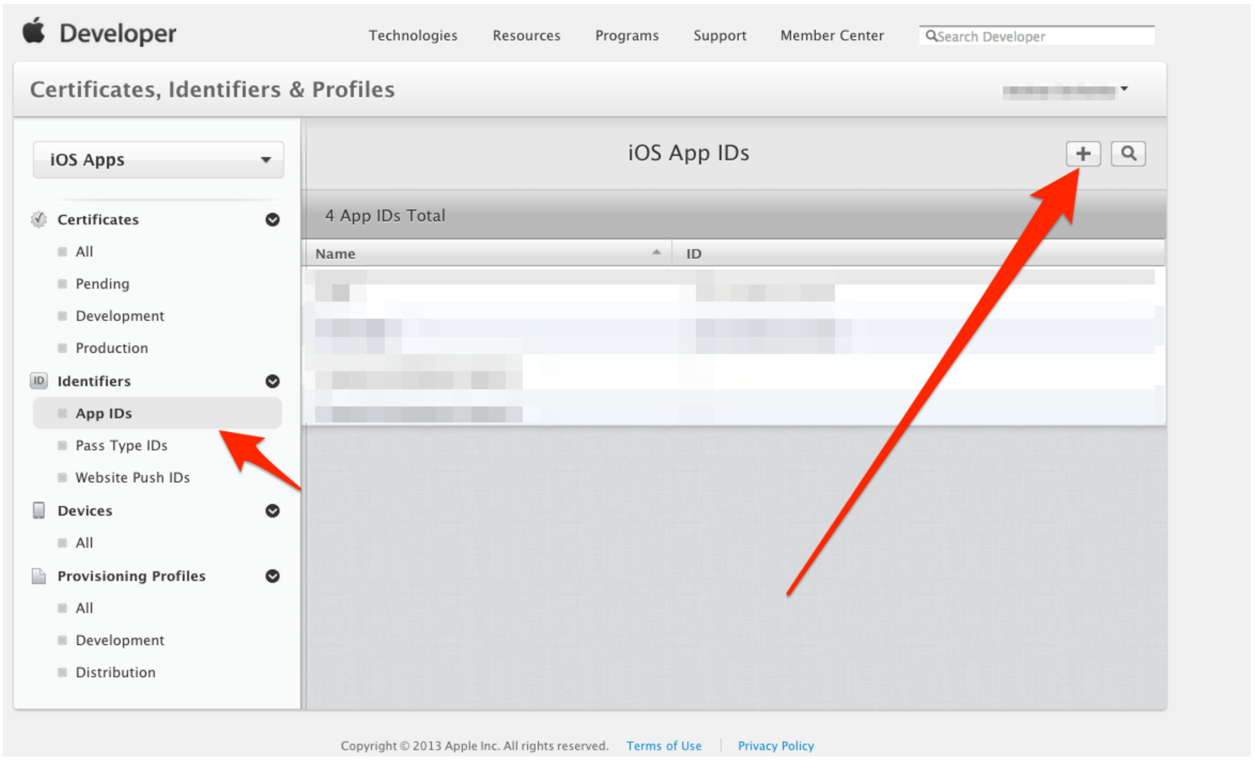


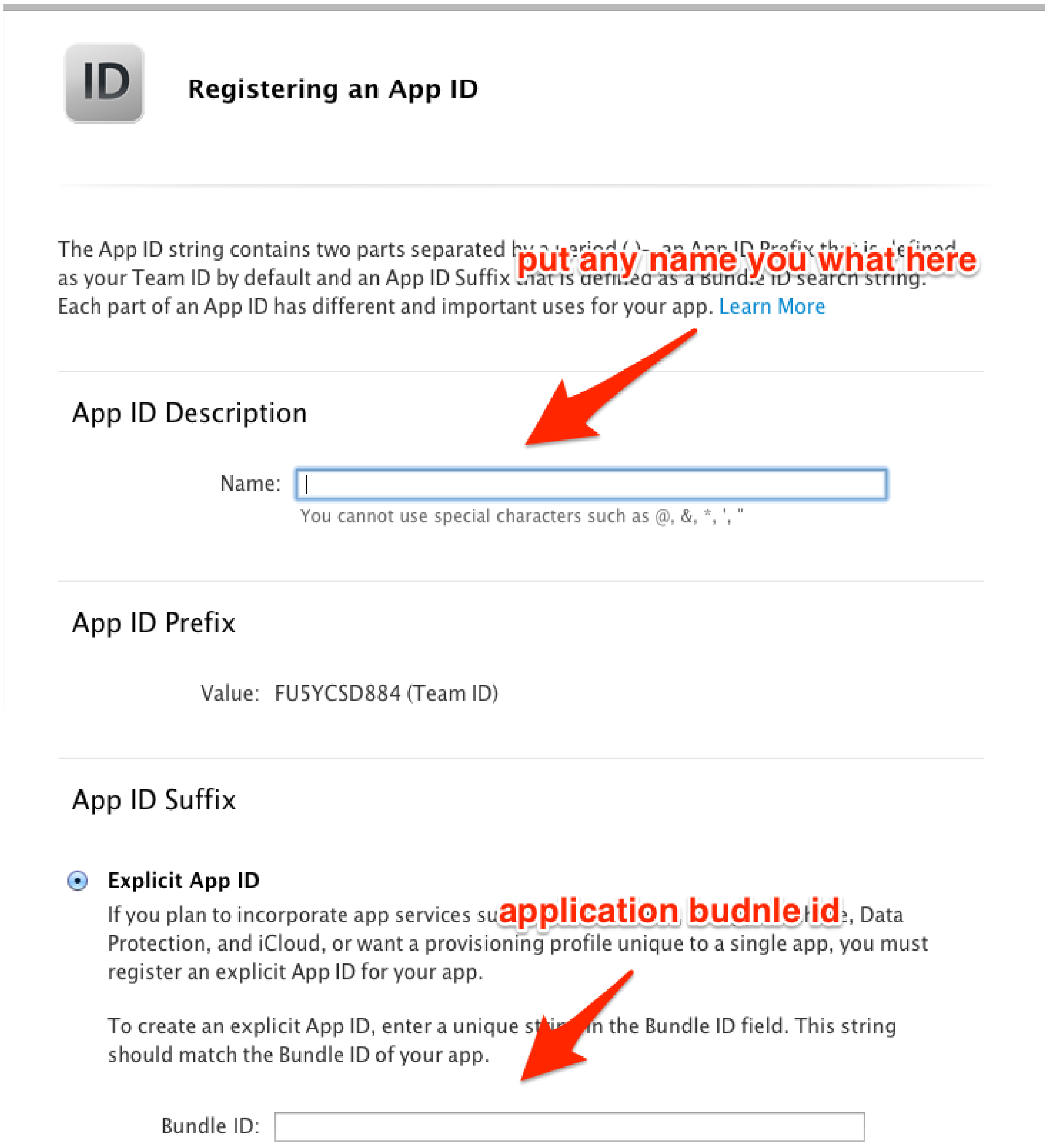
Open certificates tab. If you do not have developer certificate yet, press "+" and follow instructions to create one.



Download and click on downloaded certificate to install it to your keychain.

Next create app id (bundle id) of your application.





Registering an App ID

The App ID string contains two parts separated by a period (.)—an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. Learn More

App ID Description

Name: |

You cannot use special characters such as @, &, *, ', "

App ID Prefix

Value: FU5YCSD884 (Team ID)

App ID Suffix

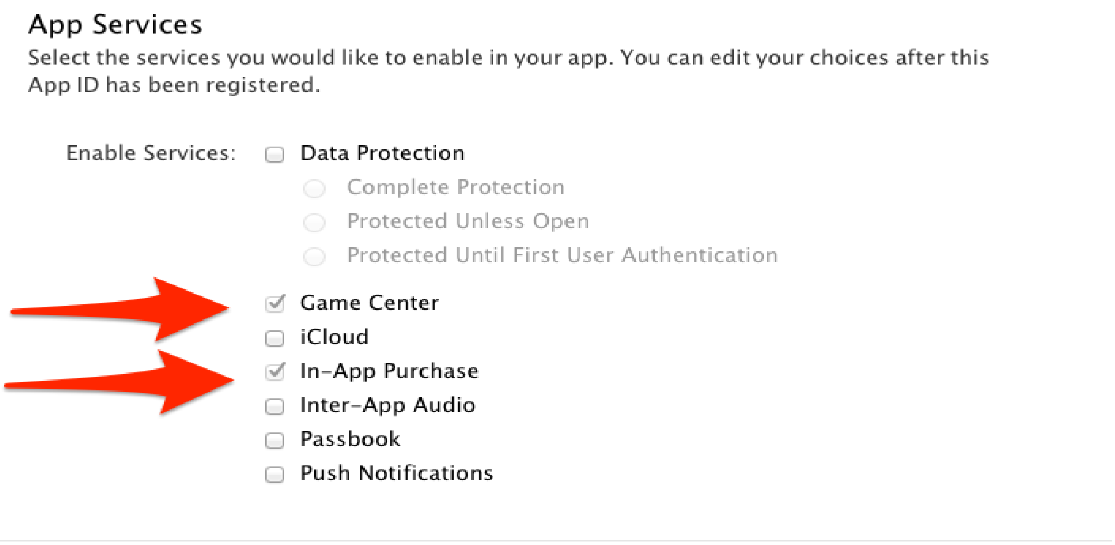
● **Explicit App ID**

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID:

Make sure that Game Center and In-App purchases are selected.



App Services

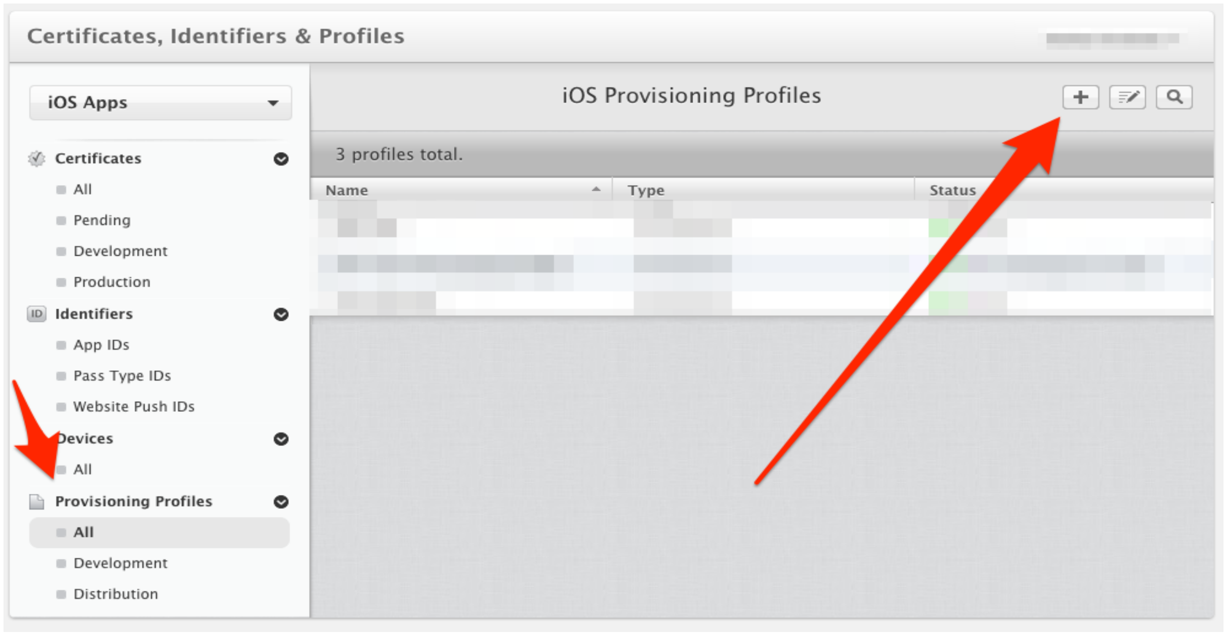
Select the services you would like to enable in your app. You can edit your choices after this App ID has been registered.

Enable Services:

- [ ] Data Protection
  - ○ Complete Protection
  - ○ Protected Unless Open
  - ○ Protected Until First User Authentication
- [✓] Game Center
- [ ] iCloud
- [✓] In-App Purchase
- [ ] Inter-App Audio
- [ ] Passbook
- [ ] Push Notifications

Go to the Devices tab and register your test devices.

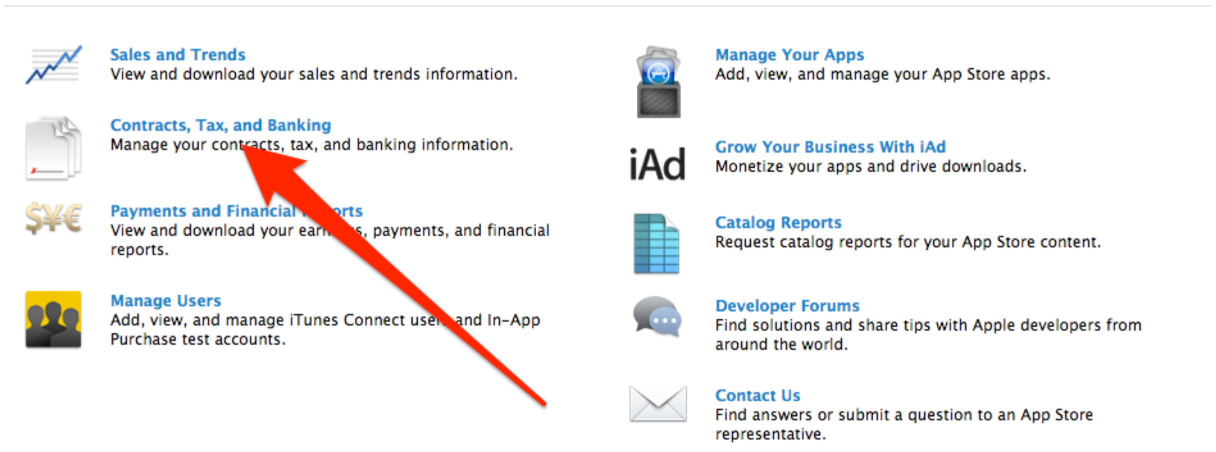
Next switch to the “**Provision Profiles**” and add profile for your game. Download and click on downloaded profile to install it to your keychain.

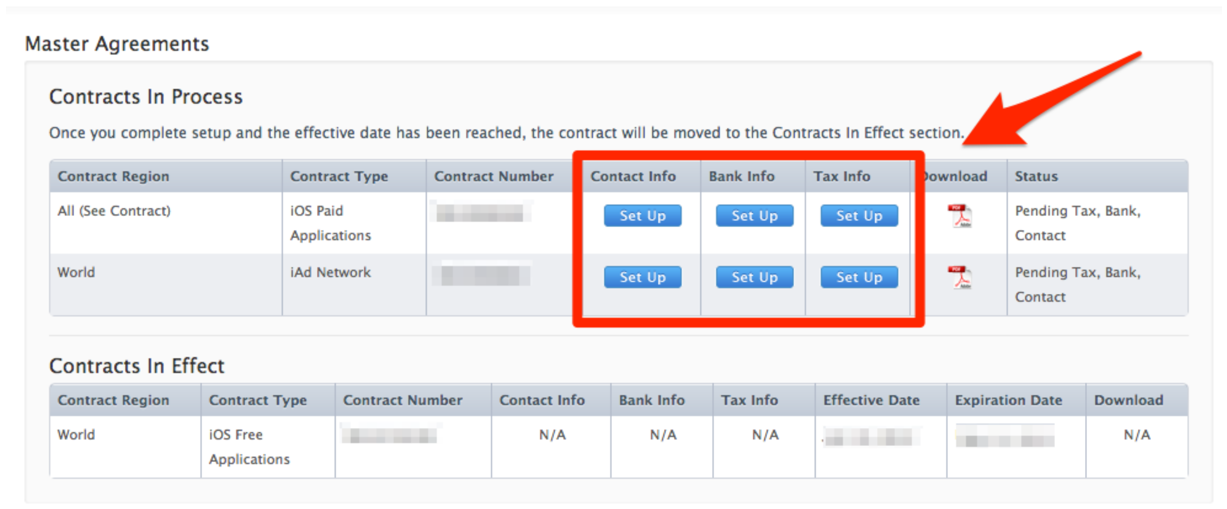


Creating iTunes app

Point your browser to [iTunes Connect]() and login to your developer account.

Select


Sales and Trends
View and download your sales and trends information.
Contracts, Tax, and Banking
Manage your contracts, tax, and banking information.
Payments and Financial Reports
View and download your earnings, payments, and financial reports.
Manage Users
Add, view, and manage iTunes Connect users and In-App Purchase test accounts.
Manage Your Apps
Add, view, and manage your App Store apps.
iAd
Grow Your Business With iAd
Monetize your apps and drive downloads.
Catalog Reports
Request catalog reports for your App Store content.
Developer Forums
Find solutions and share tips with Apple developers from around the world.
Contact Us
Find answers or submit a question to an App Store representative.



Master Agreements

Contracts In Process

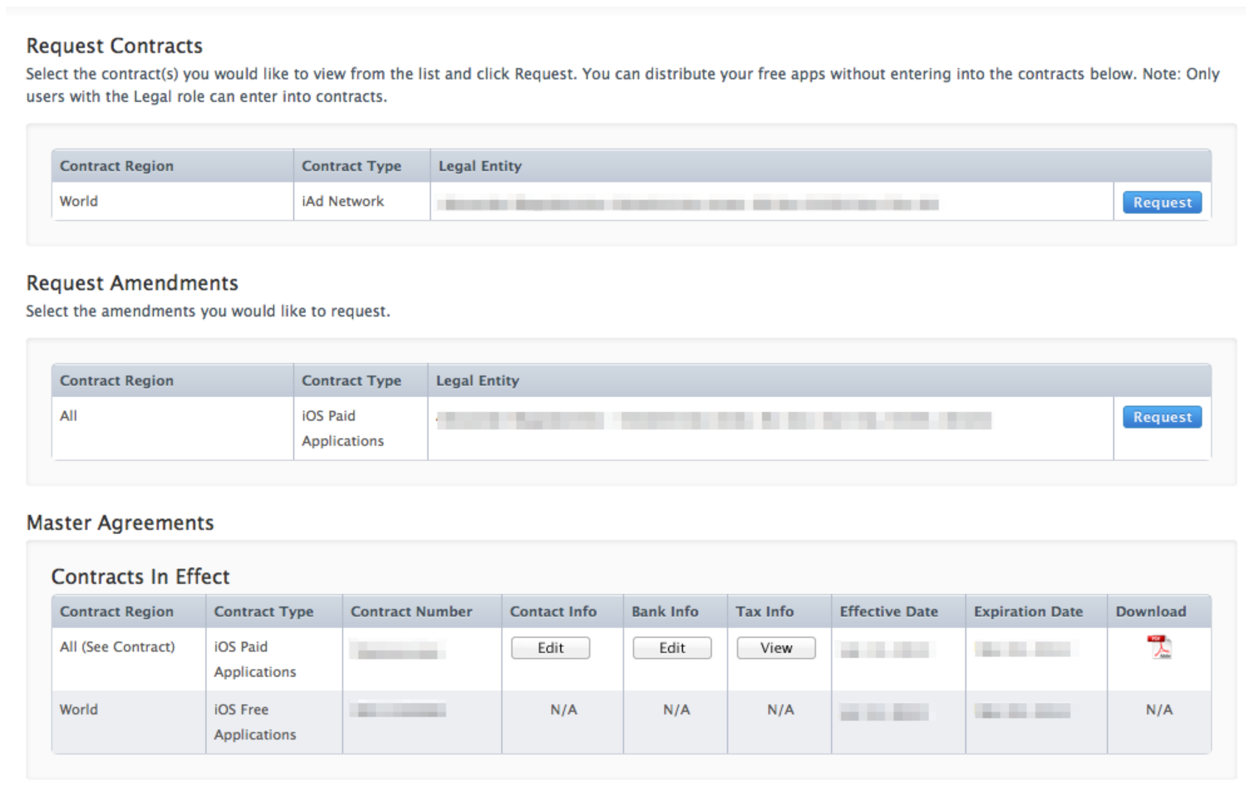
Once you complete setup and the effective date has been reached, the contract will be moved to the Contracts In Effect section.

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Download	Status
All (See Contract)	iOS Paid Applications	[redacted]	Set Up	Set Up	Set Up		Pending Tax, Bank, Contact
World	iAd Network	[redacted]	Set Up	Set Up	Set Up		Pending Tax, Bank, Contact

Contracts In Effect

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Effective Date	Expiration Date	Download
World	iOS Free Applications	[redacted]	N/A	N/A	N/A	[redacted]	[redacted]	N/A

It should look similar to this when you're done.



Request Contracts

Select the contract(s) you would like to view from the list and click Request. You can distribute your free apps without entering into the contracts below. Note: Only users with the Legal role can enter into contracts.

Contract Region	Contract Type	Legal Entity	
World	iAd Network	[redacted]	Request

Request Amendments

Select the amendments you would like to request.

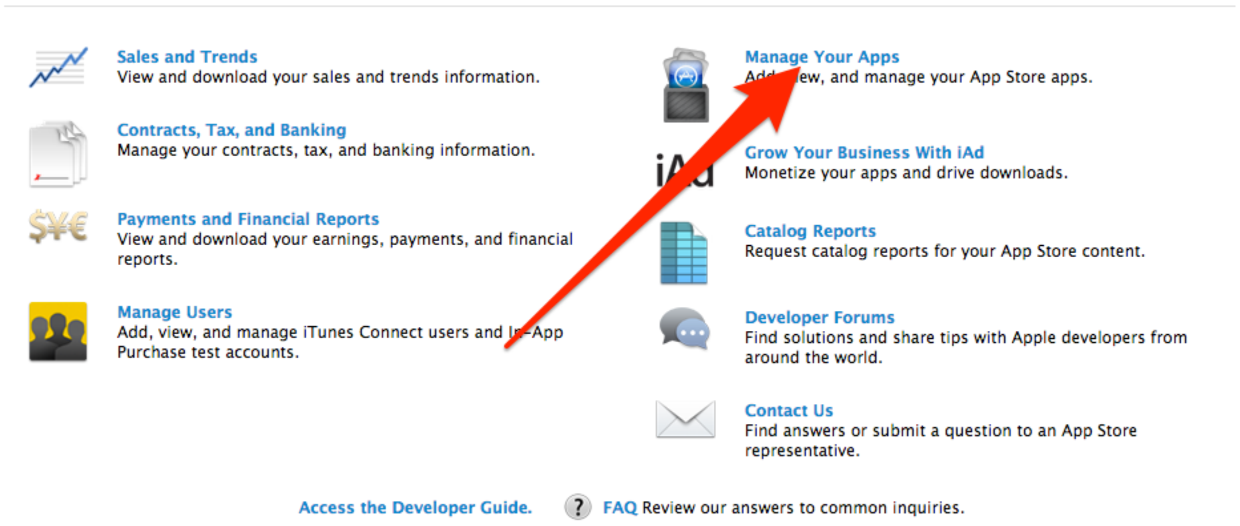
Contract Region	Contract Type	Legal Entity	
All	iOS Paid Applications	[redacted]	Request

Master Agreements

Contracts In Effect

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Effective Date	Expiration Date	Download
All (See Contract)	iOS Paid Applications	[redacted]	Edit	Edit	View	[redacted]	[redacted]	
World	iOS Free Applications	[redacted]	N/A	N/A	N/A	[redacted]	[redacted]	N/A

Go back to the web page. And select "**Manage your apps**"



Sales and Trends
View and download your sales and trends information.

Contracts, Tax, and Banking
Manage your contracts, tax, and banking information.

Payments and Financial Reports
View and download your earnings, payments, and financial reports.

Manage Users
Add, view, and manage iTunes Connect users and In-App Purchase test accounts.

Manage Your Apps
Add, view, and manage your App Store apps.

Grow Your Business With iAd
Monetize your apps and drive downloads.

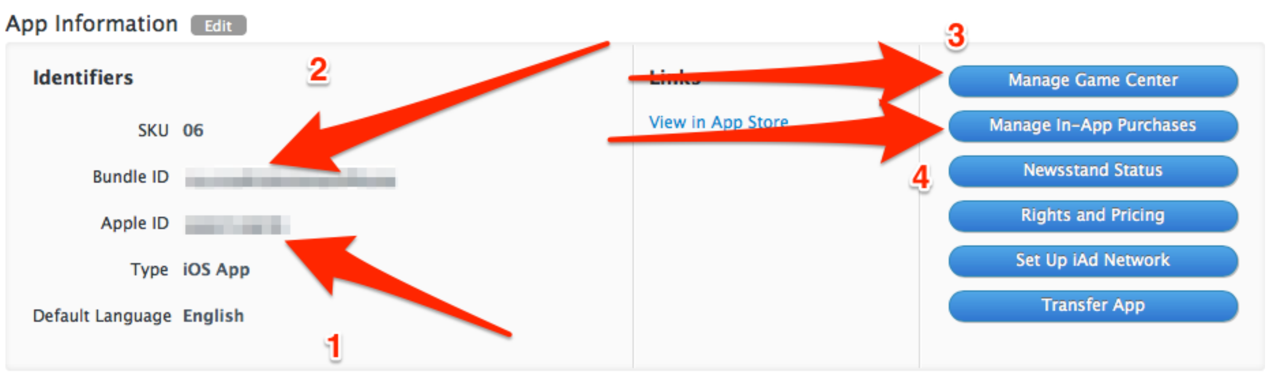
Catalog Reports
Request catalog reports for your App Store content.

Developer Forums
Find solutions and share tips with Apple developers from around the world.

Contact Us
Find answers or submit a question to an App Store representative.

Access the Developer Guide. FAQ Review our answers to common inquiries.

Create new app, and fill all the required fields. After app is created you can get necessary app info to provide it to the plugin and create In-App purchases and Game Center.


App Information Edit

Identifiers

SKU	06
Bundle ID	[redacted]
Apple ID	[redacted]
Type	iOS App
Default Language	English

Links

View in App Store

1) Your App id. Required for plugin to work correctly.

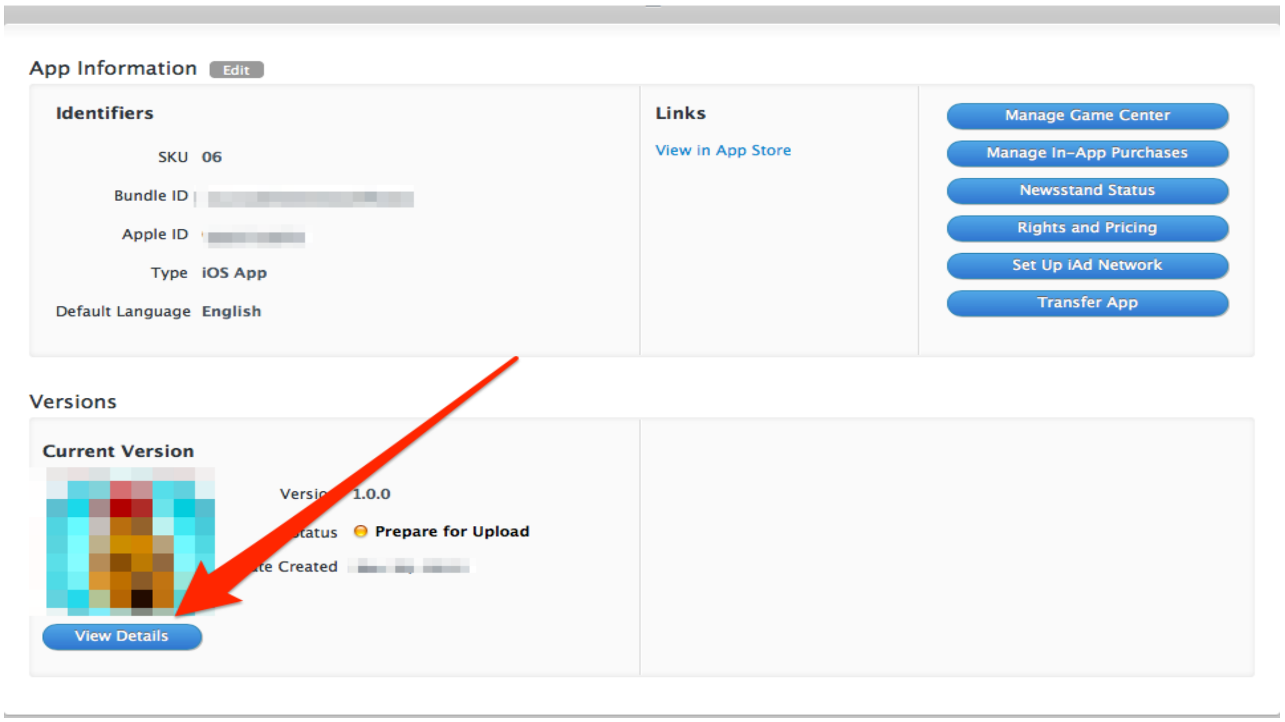
2) Your App Bundle ID. Required for plugin to work correctly.

3) Manage Game Center

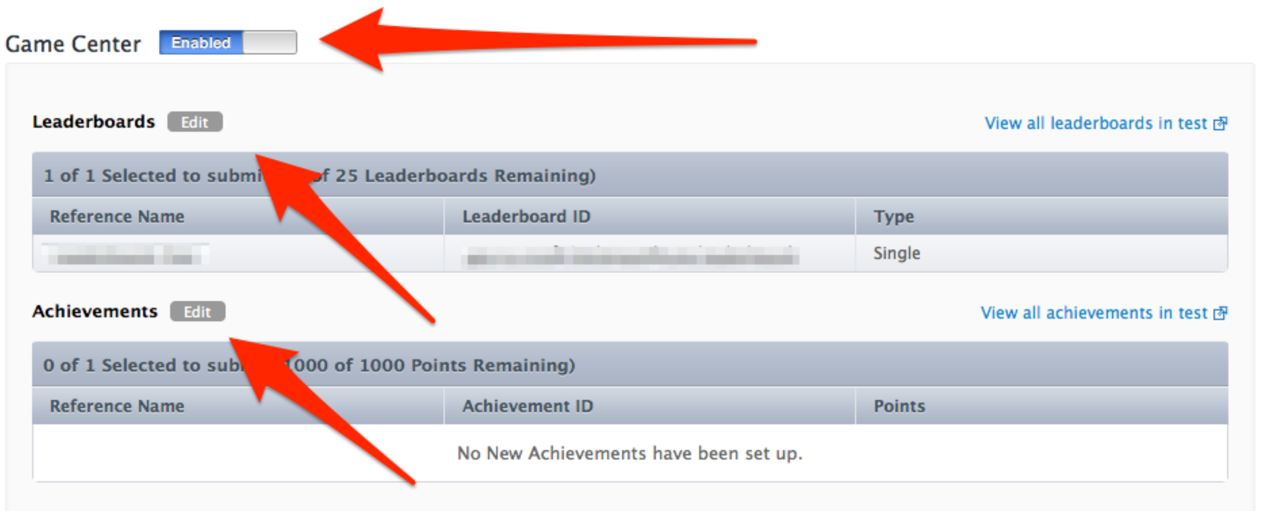
4) Manage In-App Purchases

Manage Game Center

Press **Manage Game Center** button and create all necessary leaderboards and achievements for your game. Then go back to the created app and press **View Details** button.



Enable Game Center support for your game and add leaderboards and achievements you want to test by pressing **Edit** button.



Game Center Enabled

Leaderboards Edit View all leaderboards in test

1 of 1 Selected to submi f 25 Leaderboards Remaining)

Reference Name	Leaderboard ID	Type
[redacted]	[redacted]	Single

Achievements Edit View all achievements in test

0 of 1 Selected to sub 000 of 1000 Points Remaining)

Reference Name	Achievement ID	Points
	No New Achievements have been set up.	

Game Center Coding Guidelines

To enable Game Center you should call `GameCenterManager.init();` function on your app start up. You can also call it later if you do not want your user to log in on startup.

This function will start authentication flow. If user is logged out from Game Center hi will see Game Center loging in window. If user logs in successfully you will got GAME_CENTER_PLAYER_AUTHENTICATED event. If authentication flow has been canceled or any other error occurred you will get GAME_CENTER_PLAYER_AUTHENTIFICATION_FAILED.

Note: If you got GAME_CENTER_PLAYER_AUTHENTIFICATION_FAILED it generally means that user does not want to use your app with Game Center. So probably it will be wise to not call `GameCenterManager.init()` function on startup any more, and just notify user that Game Center is disabled. And add button “enable Game Center” for user to be able start using Game Center with your app again.

Note: Every time when your app goes background, and restored from background state, user authentication will be renewed. With means you will again got GAME_CENTER_PLAYER_AUTHENTICATED or GAME_CENTER_PLAYER_AUTHENTIFICATION_FAILED event.

Warning: Do not use any Game Center function until you sure that user is successfully logged in (you will get GAME_CENTER_PLAYER_AUTHENTICATED event as soon it happens)

Warning: If you see pop up window "Game Center is disabled" this means that you trying to use Game Center when user is not authed. And plugin tries to fix this. But if you got it in the first place this mean you trying to call Game Center function before you got GAME_CENTER_PLAYER_AUTHENTICATED event with is wrong.

Troubleshooting

If you've cancelled game center sign in enough times, the OS disables game center for that game. Prior to iOS7, you could manually sign in again using the game center app, and when you launch your game again it would sign in.

However, in iOS7, it appears that when the OS disables game center for your game, it's completely disabled. I've found that using **Settings > General > Reset > Reset All Settings** will restore game center functionality for games that have been disabled in this manner.

More Information

Copied answer from here: http://stackoverflow.com/a/19291001/1321401 - a bit different question caused by the same issue.

This issue appears to happen in sandbox only.

GameCenter shows prompt for login only a few times (3-5), if all prompts were canceled - GameCenter becomes disabled for this app.

In iOS 7 there is a problem: GameCenter disabled state remains even if you login into the GameCenter app itself - **application will still receive kind of this error:**

```
error: Error Domain=GKErrorDomain Code=2 GkErrorCanceled: the requested
operation has been canceled or disabled by the user
```

There are two workarounds:

First:

1. Log out of Game Center through the Settings app.
2. Reset all settings
3. After the device rebooted, launch app with development build. After launch the GC Login viewController pops up.

Second (this one helped me):

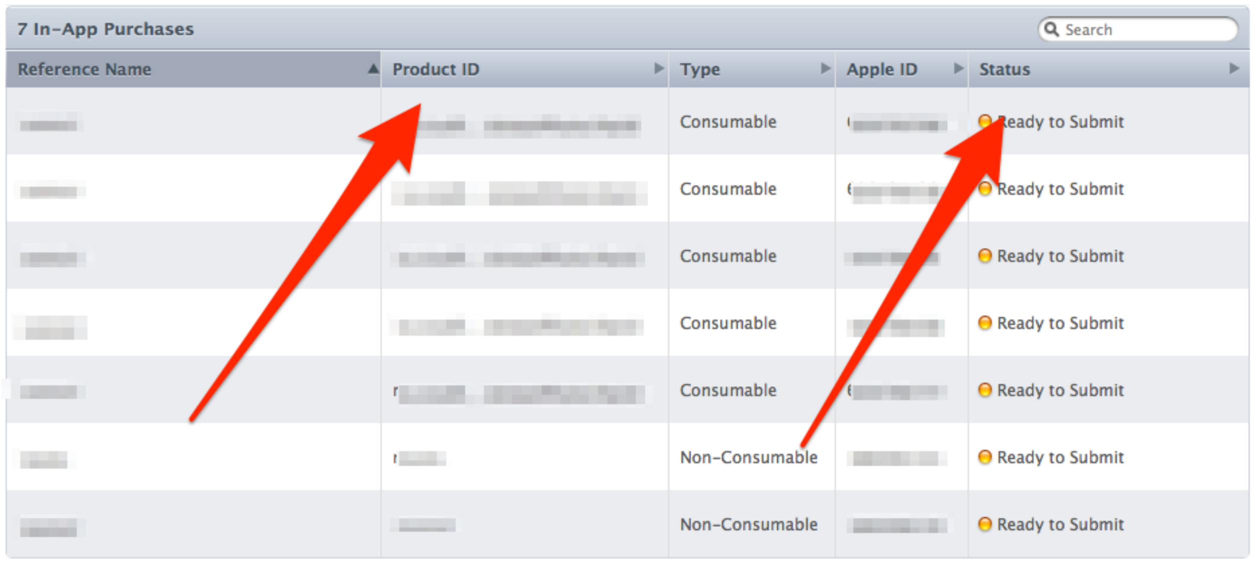
1. "Erase All Content And Settings".
2. Set up as a new device - do not restore from any backups.

Both workarounds are not a viable option for production - I hope there will be a fix from Apple soon.

Most of the info is from to this discussion.

Manage In-App Purchases

Press **Manage In-App Purchases** and create all necessary products for your app.



7 In-App Purchases

Reference Name	Product ID	Type	Apple ID	Status
[redacted]	[redacted]	Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Non-Consumable	[redacted]	● Ready to Submit
[redacted]	[redacted]	Non-Consumable	[redacted]	● Ready to Submit

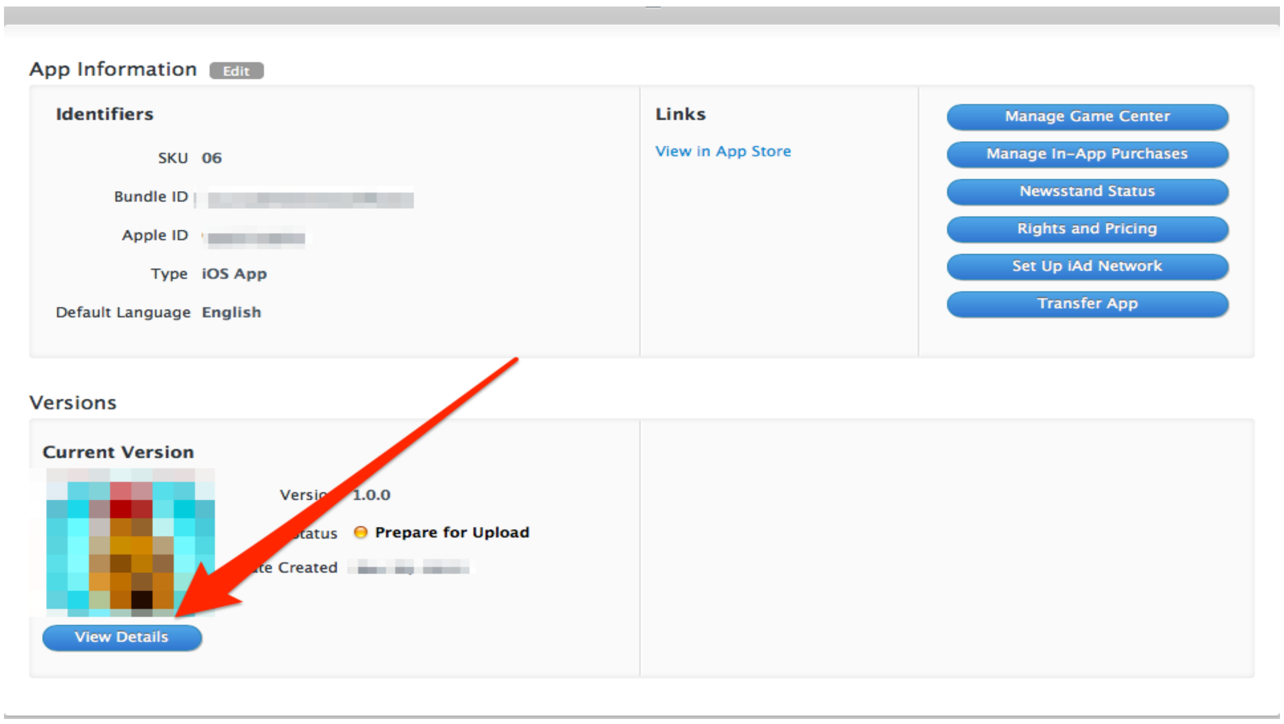
You will have similar table of in-apps when you finish creating your products.

Make sure that you have uploaded screenshot for each in-app product, and all of them has **Ready to Submit** status.

Product ID column displace id of your product. You will manage your products in code by this ids.

Not enough just create products, you also have to add them to your app.

Go back to the created app and press **View Details** button.


App Information Edit

Identifiers

SKU	06
Bundle ID	
Apple ID	
Type	iOS App
Default Language	English

Links

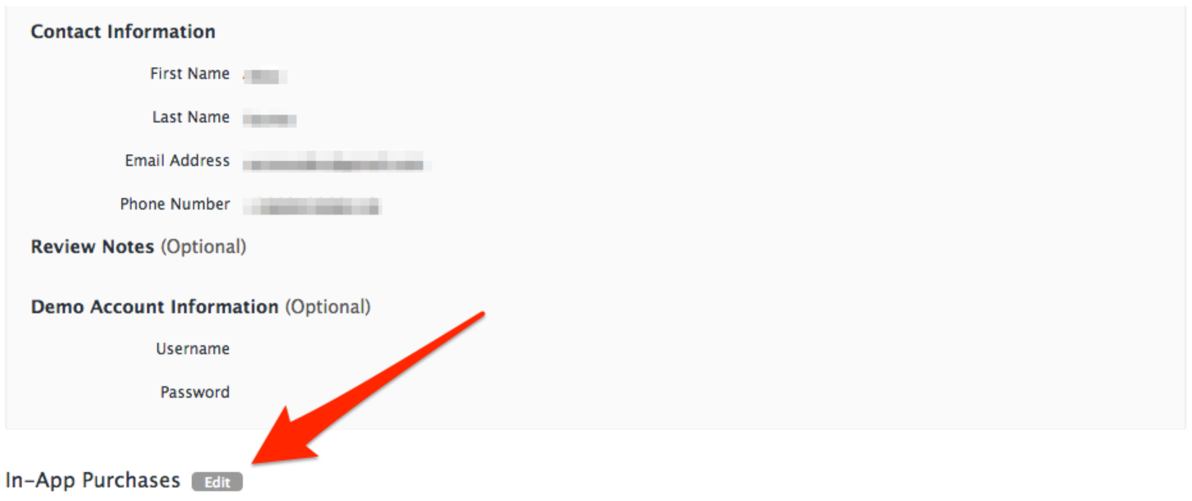
View in App Store

Versions

Current Version

Version	1.0.0
Status	Prepare for Upload
Date Created	

Press **Edit** button near In-App Purchases


Contact Information

First Name	
Last Name	
Email Address	
Phone Number	

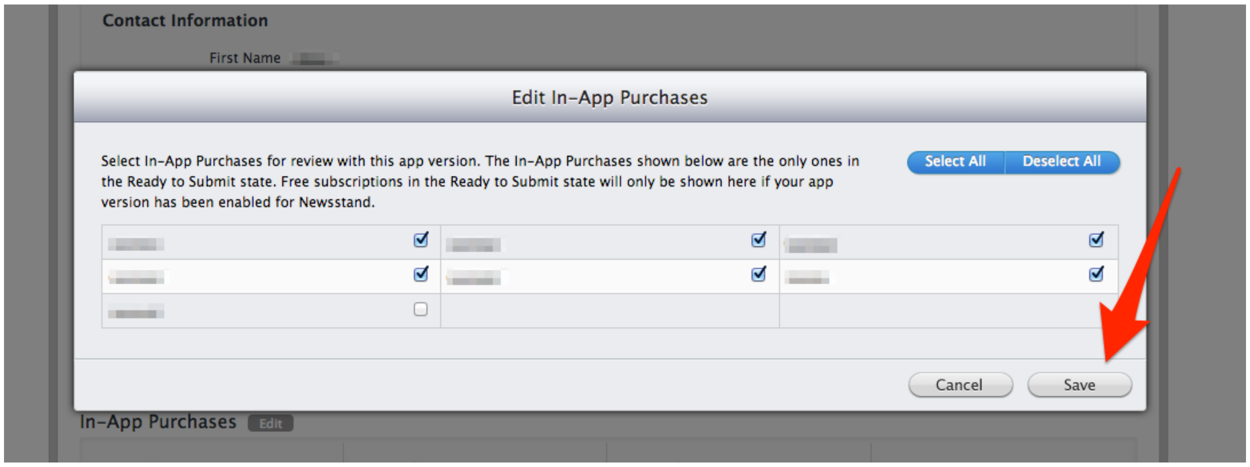
Review Notes (Optional)

Demo Account Information (Optional)

Username	
Password	

In-App Purchases Edit

Select all products you want to be used with your app, and press **Save** button.


Contact Information

First Name [redacted]

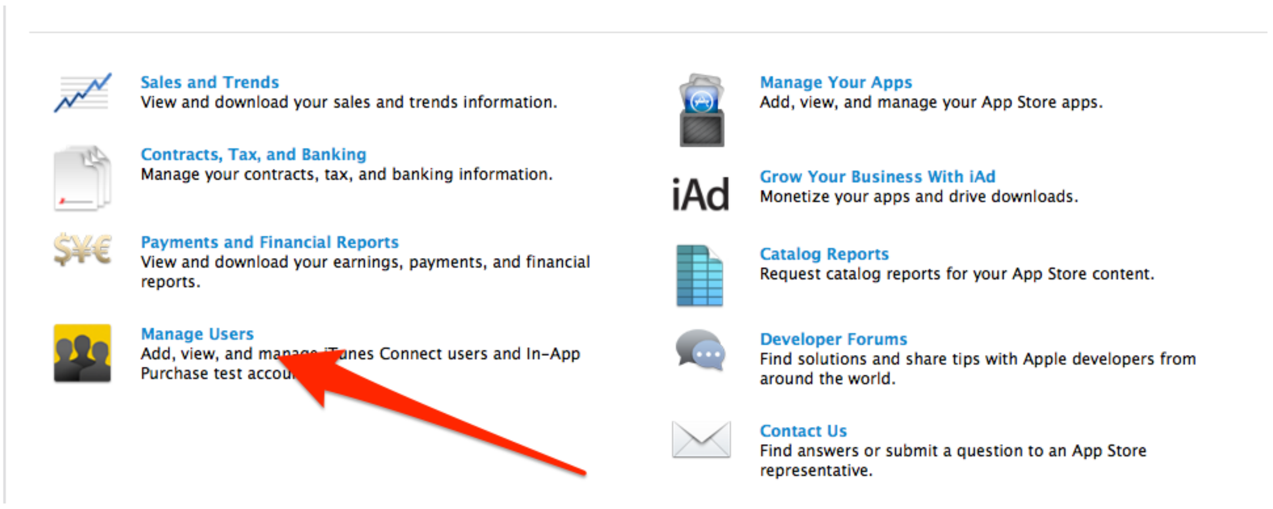
Edit In-App Purchases

Select In-App Purchases for review with this app version. The In-App Purchases shown below are the only ones in the Ready to Submit state. Free subscriptions in the Ready to Submit state will only be shown here if your app version has been enabled for Newsstand.

[redacted]	✓	[redacted]	✓	[redacted]	✓
[redacted]	✓	[redacted]	✓	[redacted]	✓
[redacted]	☐				

To be able to Test In-Apps you have to create test account and log out from your real account on your device.

Go back to the ITunes Connect main page. Then Manage Users → Test User. Fill all required fields and create new test user.


Sales and Trends
View and download your sales and trends information.

Contracts, Tax, and Banking
Manage your contracts, tax, and banking information.

Payments and Financial Reports
View and download your earnings, payments, and financial reports.

Manage Users
Add, view, and manage iTunes Connect users and In-App Purchase test accou

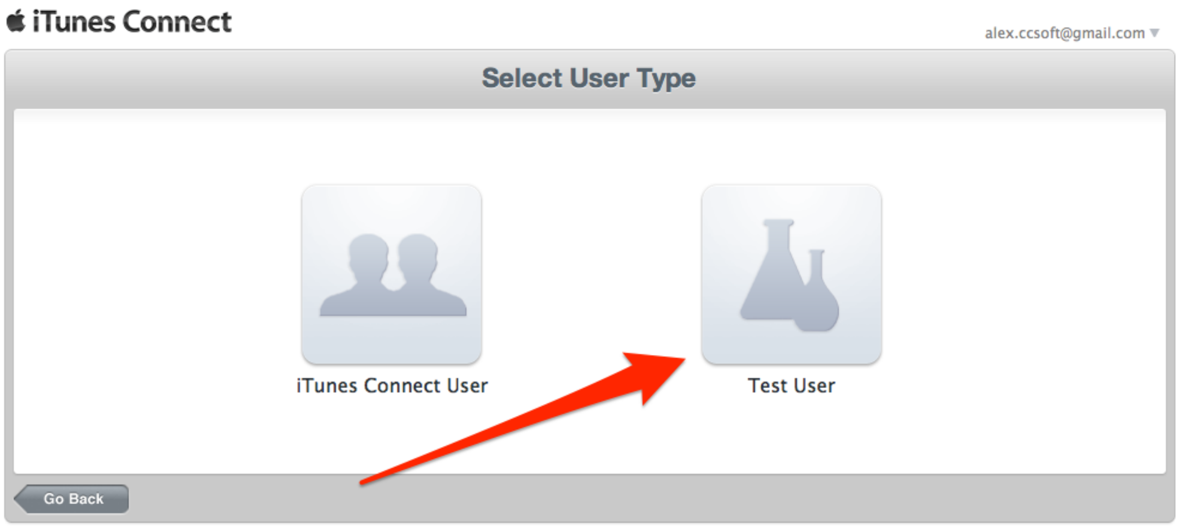
Manage Your Apps
Add, view, and manage your App Store apps.

Grow Your Business With iAd
Monetize your apps and drive downloads.

Catalog Reports
Request catalog reports for your App Store content.

Developer Forums
Find solutions and share tips with Apple developers from around the world.

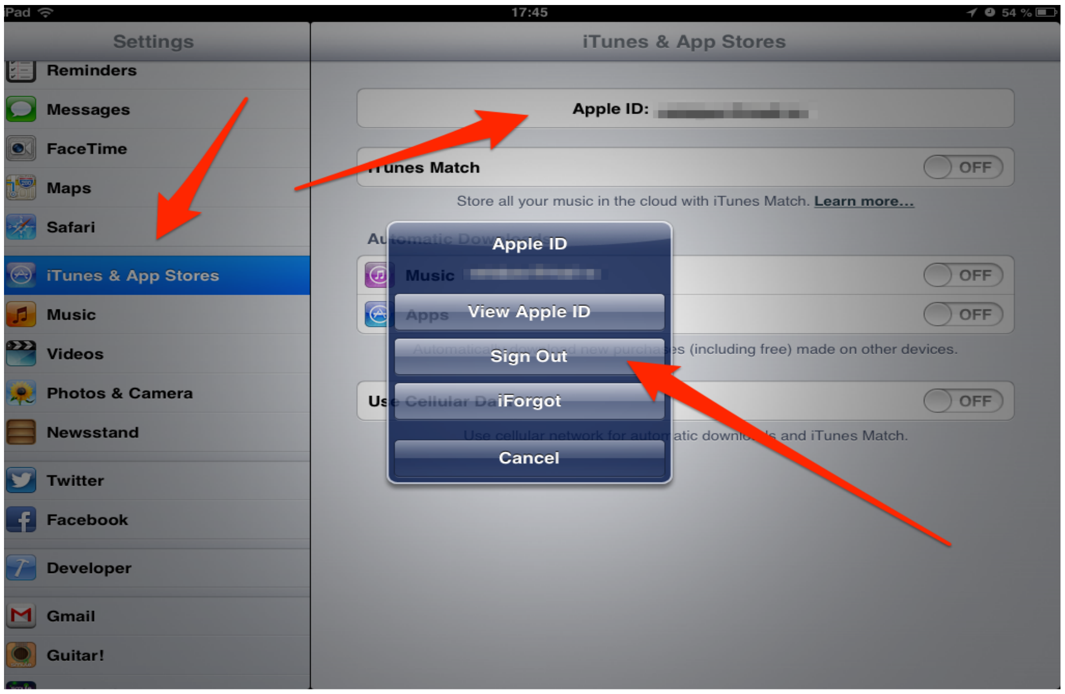
Contact Us
Find answers or submit a question to an App Store representative.



Make sure that you sign out from your real account before testing in-apps on device.

To do that open device settings, select iTunes & App Store, tab on your account and press Sign Out button.

Warning: Do not try to sign in as test user. Use your test user id and pass when your game ask for it.



In-App Purchases Coding Guidelines

Take a look on **PaymnetManagerExample** class. This is example of game payment manager you should create one for your game or modify this example to make it perfect fit for your game.

Take a look at **init** function. It should be called before your game starts

```
public static void init() {
        //adding your game products id's
        InAppPurchaseManager.instance.addProductId(SMALL_PACK);
        InAppPurchaseManager.instance.addProductId(NC_PACK);
        //signing on InAppPurchaseManager events
        InAppPurchaseManager.instance.addEventListener(InAppPurchaseManager.PRODUCT_BOUGHT,
onProductBought);
        InAppPurchaseManager.instance.addEventListener(InAppPurchaseManager.TRANSACTION_FAILED,
onTransactionFailed);
        InAppPurchaseManager.instance.addEventListener(InAppPurchaseManager.VERIFICATION_Responce,
onVerificationResponce);
        InAppPurchaseManager.instance.loadStore();
}
```

In init function you should register your products, add listeners for events you need (at least `PRODUCT_BOUGHT` events)

And then using `InAppPurchaseManager.instance.loadStore();` function send load store request to apple.

`InAppPurchaseManager.instance.loadStore();` function will trigger STORE_KIT_INITIALIZED event. You should wait for this event before calling other methods of `InAppPurchaseManager` class. As soon as you got it, you can be sure that store kit fully initialized and ready to work.

Here is buyItem function of PaymnetManagerExample

```
public static void buyItem(string productId) {

      InAppPurchaseManager.instance.buyProduct(productId);

}
```

It's really simple, it just receive productId (SKU) and starts purchase flow using method. But you can add some code like:

- Check is store kit initialized
- Probably suggest something to your customer before purchase
- Fade screen, draw preloader or pause the game util customer purchasing your stuff.

```
private static void onProductBought(CEvent e) {
    IOSStoreKitResponce Responce =  e.data as IOSStoreKitResponce;
    Debug.Log("STORE KIT GOT BUY: " + Responce.productIdentifier);
    Debug.Log("RECIPT: " + Responce.receipt);

    switch(Responce.productIdentifier) {
    case SMALL_PACK:
       //code for adding small game money amount here
       break;
    case NC_PACK:
       //code for unlocking cool item here
       break;

    }
    IOSNative.showMessage("Success", "product " + Responce.productIdentifier + " is purchased");
  }
```

This function demonstrate with data you will get when customer has bought the product. You can use switch or any other methods (Unlocker class for example) to unlock content for your customers or increasing customer balance with your game currency.

Sure you can remove all logs form this function, and add some screen unlock code, if screen was locked in buyItem method.

```
private static void OnTransactionFailed(CEvent e) {
    IOSStoreKitResponce Responce =  e.data as IOSStoreKitResponce;
    IOSNative.showMessage("Fail", Responce.error);
}
```

We were signed for TRANSACTION_FAILED event, so OnTransactionFailed function will be perfect fit to notify your user that something was wrong and ask him to try later, and add some screen unlock code, if screen was locked in buyItem method.

Note: If you will call restorePurchases You will get PRODUCT_BOUGHT event for each already bought product. Sou you do not have to implement any other additional logic for purchases restoring. But you should be ready that when you call restorePurchases methods, you can get bunch of PRODUCT_BOUGHT events.

iCloud SetUP

From the user’s perspective, iCloud is a simple feature that automatically makes their personal content available on all their devices. To make your app participate in this "magic," you need to design and implement your app somewhat differently, and for this you need to learn about your app's roles when it participates with iCloud.

These roles, and the specifics of your iCloud adoption process, depend on your app. You design how your app manages its data, so only you can decide which iCloud supporting technologies your app needs and which ones it does not.

This chapter gets you started with the fundamental elements of iCloud that all developers need to know.

First, Provision Your Development Devices

To start developing an iCloud app, you must have an appropriate device provisioning profile and app ID. If you don't already have these in place, learn about setting up a provisioning profile and app ID in "Provisioning Your App for Store Technologies" in *App Distribution Guide*.

iCloud Data Transfer Proceeds Automatically and Securely

When you adopt iCloud, the operating system initiates and manages uploading and downloading of data for the devices attached to an iCloud account. Your app does not directly communicate with iCloud servers and, in most cases, does not invoke upload or download of data. At a very high level, the process works as follows:

1. You configure your app to gain access to special local file system locations known as *ubiquity containers*.
2. You design your app to respond appropriately to changes in the availability of iCloud (such as if a user signs out of iCloud), and to changes in the locations

of files (because instances of your app on other devices can rename, move, duplicate, or delete files).

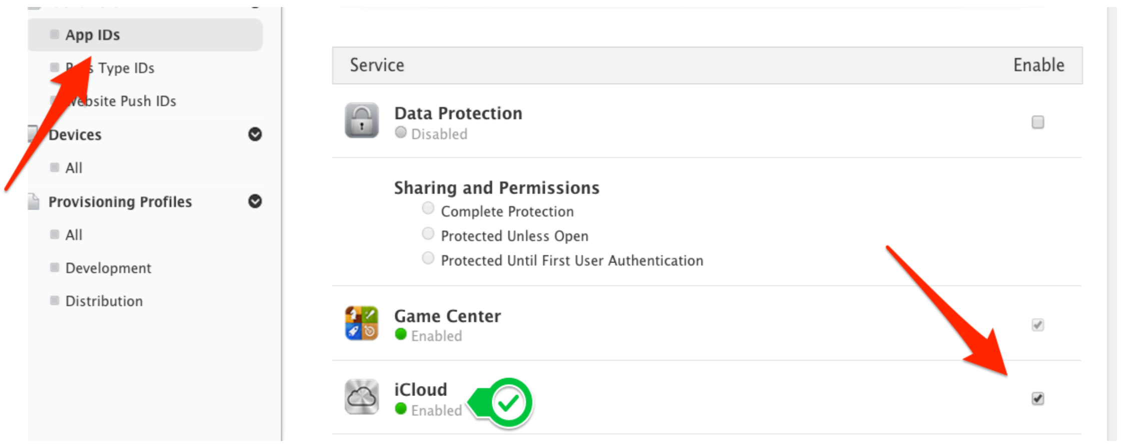
3. Your app reads and writes to its ubiquity containers using APIs that provide file coordination, as explained in “How iCloud Document Storage Works.”
4. The operating system automatically transfers data to and from iCloud as needed.

In iOS, there is an exception to automatic iCloud data transfer. For the first-time download of an iCloud-based document in iOS, your app actively requests the document. You learn about this process in “How iCloud Document Storage Works.”

iCloud secures user data with encryption in transit and on the iCloud servers, and by using secure tokens for authentication. For details, refer to iCloud security and privacy overview. Key-value storage employs the same security as iCloud uses for "Documents in the Cloud,” as it is described in that document.

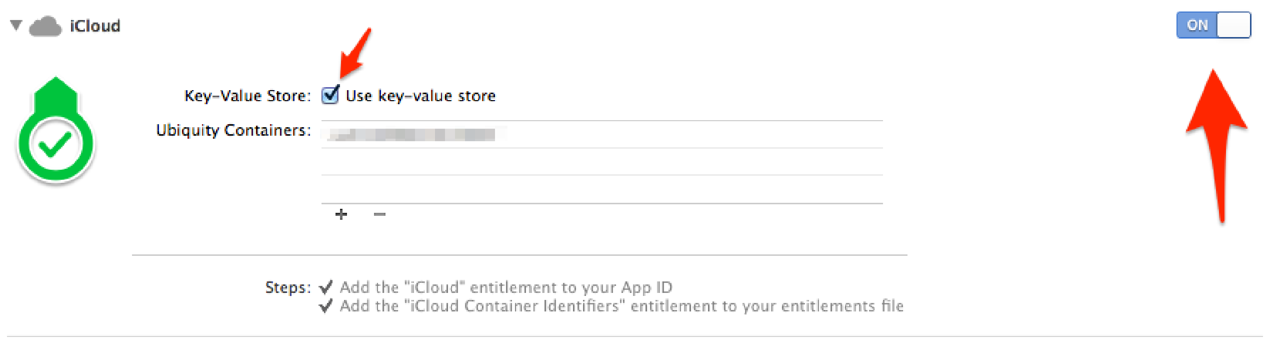
Set Up Steps.

1) Enable iCloud for your app id



Service	Enable
Data Protection ● Disabled	☐
Sharing and Permissions ○ Complete Protection ○ Protected Unless Open ○ Protected Until First User Authentication	
Game Center ● Enabled	✓
iCloud ● Enabled	✓

2) Enable iCloud inside XCode project



iCloud

ON

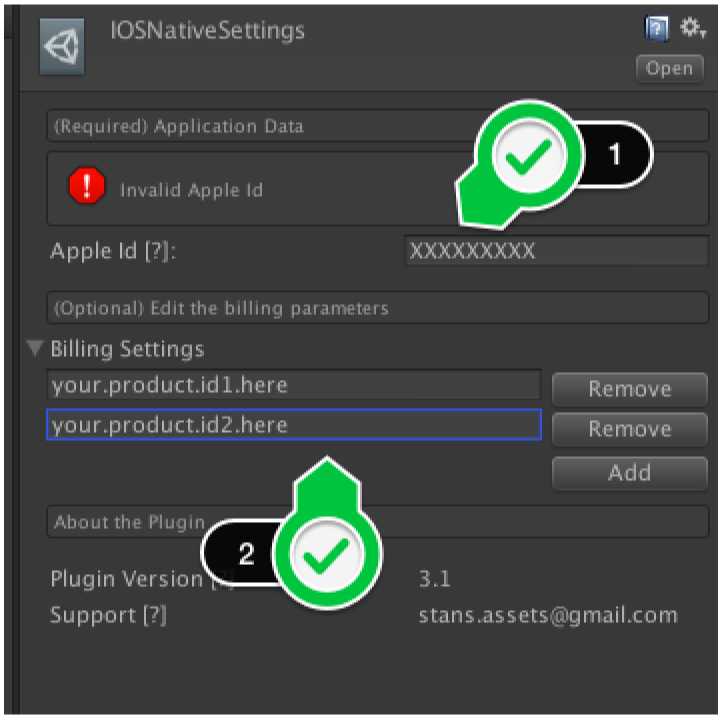
Key-Value Store: ☑ Use key-value store

Ubiquity Containers: [redacted]

Steps: ✓ Add the "iCloud" entitlement to your App ID
✓ Add the "iCloud Container Identifiers" entitlement to your entitlements file

Plugin Setup

SetUp in Unity



IOSNativeSettings

(Required) Application Data

Invalid Apple Id

Apple Id [?]: XXXXXXXXX

(Optional) Edit the billing parameters

Billing Settings

your.product.id1.here

your.product.id2.here

About the Plugin

Plugin Version [?] 3.1

Support [?] stans.assets@gmail.com

1) Set your game Apple ID.

2) If you have In-App purchases in your game you can add products id using this GUI. Or you can do this in code using `IOSInAppPurchaseManager.instance.addProductId` function.

Set Up in XCode

You should include following frameworks to your IOS project

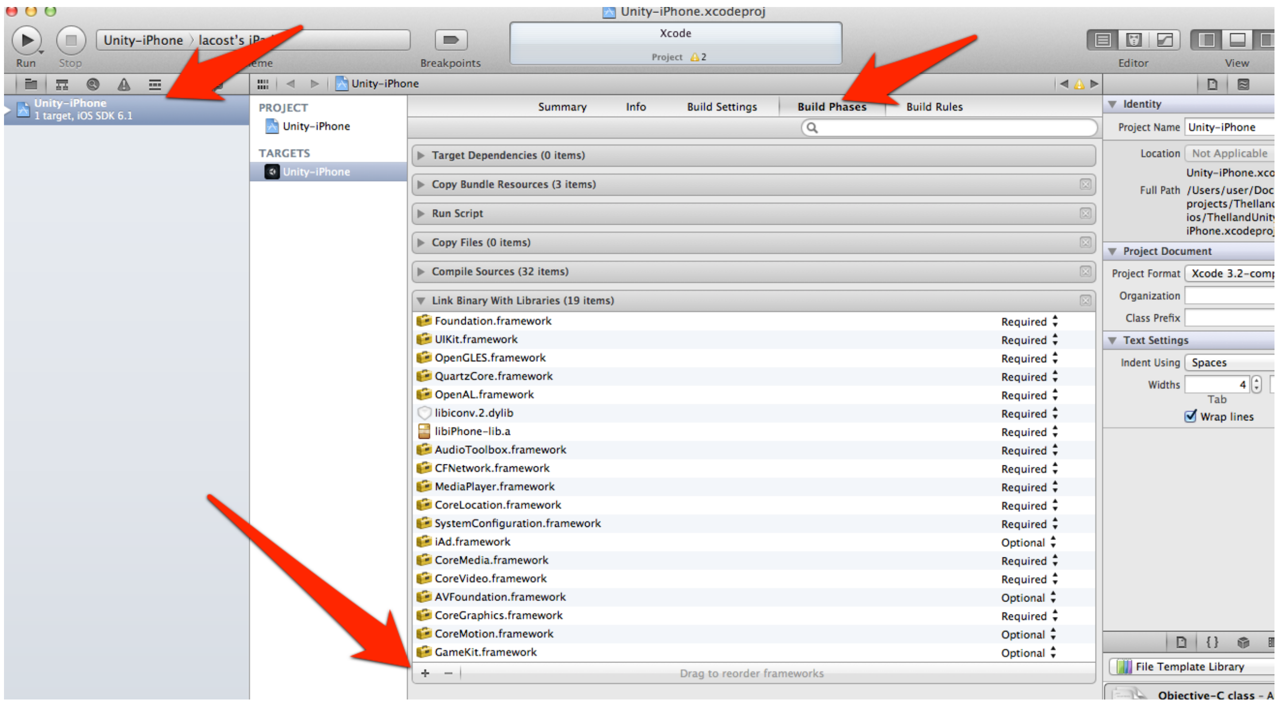
StoreKit.framework

Accounts.*framework*

Social.*framework*

Here is instruction how to do that.

Open your project. Click on The Solution Name (project setting will open). Click on "**Build Phase Tab**", C



Unity-iPhone
1 target, iOS SDK 6.1

PROJECT
Unity-iPhone

TARGETS
Unity-iPhone

Target Dependencies (0 items)
Copy Bundle Resources (3 items)
Run Script
Copy Files (0 items)
Compile Sources (32 items)
Link Binary With Libraries (19 items)

Foundation.framework	Required
UIKit.framework	Required
OpenGLES.framework	Required
QuartzCore.framework	Required
OpenAL.framework	Required
libiconv.2.dylib	Required
libiPhone-lib.a	Required
AudioToolbox.framework	Required
CFNetwork.framework	Required
MediaPlayer.framework	Required
CoreLocation.framework	Required
SystemConfiguration.framework	Required
iAd.framework	Optional
CoreMedia.framework	Required
CoreVideo.framework	Required
AVFoundation.framework	Optional
CoreGraphics.framework	Required
CoreMotion.framework	Optional
GameKit.framework	Optional

Identity

Project Name	Unity-iPhone
Location	Not Applicable
	Unity-iPhone.xco
Full Path	/Users/user/Doc projects/Thellan ios/ThellandUnit iPhone.xcodepro

Project Document

Project Format	Xcode 3.2-comp
Organization	
Class Prefix	

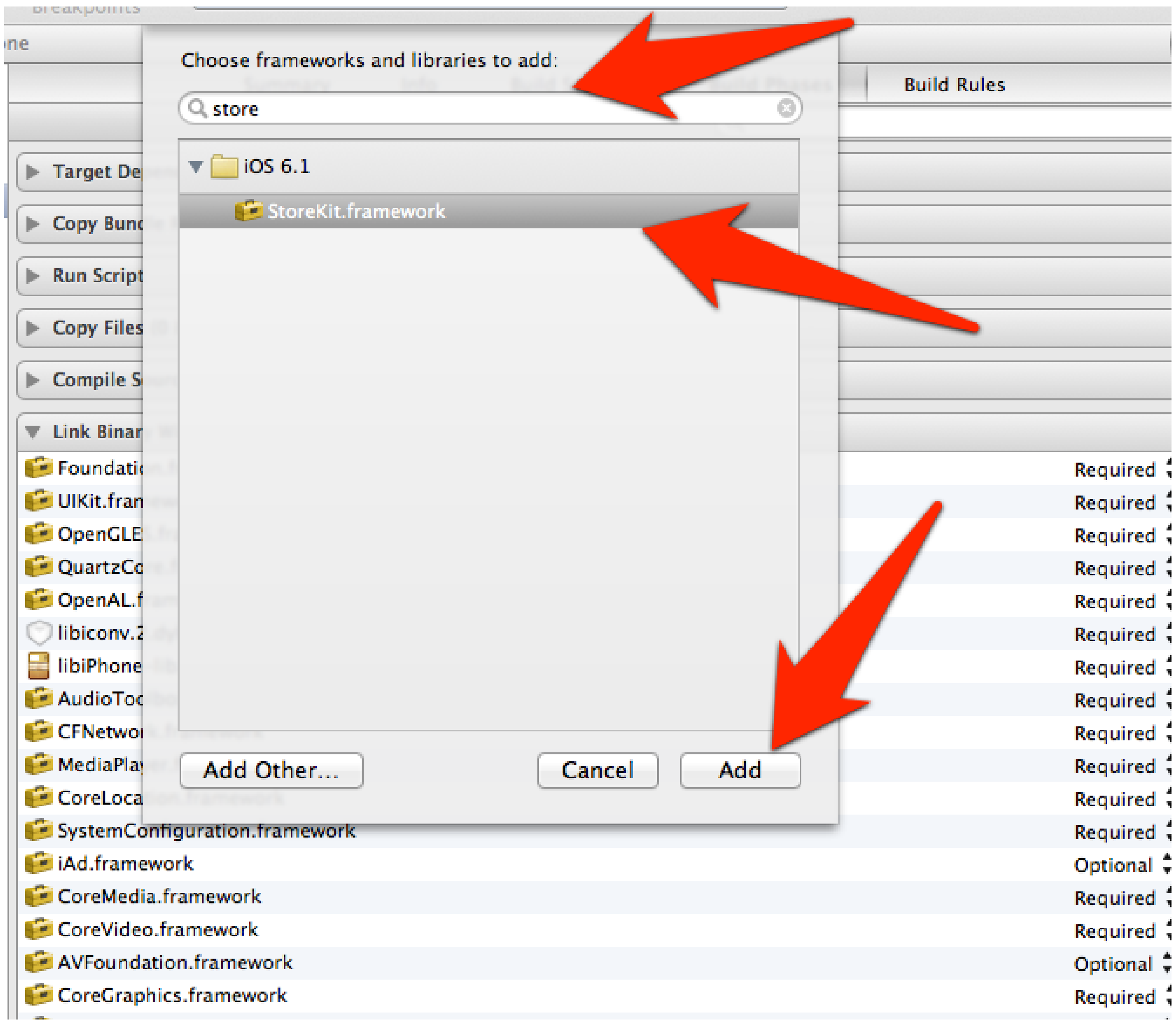
Text Settings

Indent Using	Spaces
Widths	4 Tab

☑ Wrap lines

click on "**+**" button on "Link Binary".

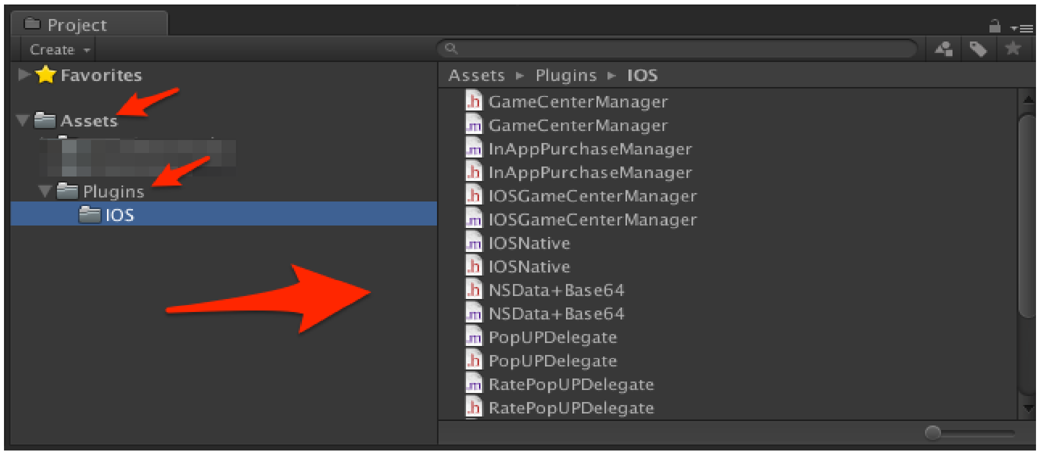
Write "store" on pop up search field. Choose Storekit framework, And press Add button. That's it.



Do the same for the rest of freamworks.

You need to make sure that extension “**Plugins**” folder is in your “**Asset”** folder. This is required as described in the Unity3D documentation “All files with extensions **.a**,**.m**,**.mm**,**.c**,**.cpp** located in the Assets/**Plugins/iOS** folder will be merged into the generated Xcode project ”

http://docs.unity3d.com/Documentation/Manual/PluginsForIOS.html

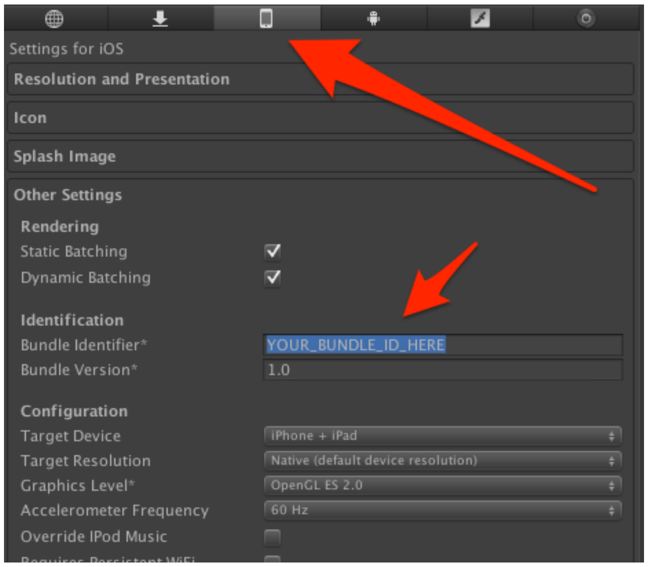


GameCenterManager.h
GameCenterManager.m
InAppPurchaseManager.m
InAppPurchaseManager.h
IOSGameCenterManager.h
IOSGameCenterManager.m
IOSNative.m
IOSNative.h
NSData+Base64.h
NSData+Base64.m
PopUPDelegate.m
PopUPDelegate.h
RatePopUPDelegate.m
RatePopUPDelegate.h

For correct plugin work you should provide all necessary information. Itunes app should be set-uped, bundle id should be correct. You should initialize IOS Native plugin on your app start up, and provide your **Apple Id.** You can do this using *Window->IOS Native->Edit Settings.* Or you can put your **Apple Id** in the **IOSNativeSettings** script. For Plugin initialization simply add this line in your game initialization script:

```
IOSNative.instance.Init();
```

Setting app bundle Id that also important step. You should set in under the build Setting menu. **File** → **Build** Settings, and press **Player Settings** Button.



Rendering
Static Batching ✓
Dynamic Batching ✓

Identification
Bundle Identifier* YOUR_BUNDLE_ID_HERE
Bundle Version* 1.0

Configuration
Target Device iPhone + iPad
Target Resolution Native (default device resolution)
Graphics Level* OpenGL ES 2.0
Accelerometer Frequency 60 Hz
Override IPod Music
Requires Persistent WiFi

If you not sure where to get **Apple ID** and **Bundle ID**, please follow IOS App Creation Guide.

Know Issues

Invalid Product IDs / StoreKit has not been initialized

This is common issue, just got thru this check list:

Make sure you can answer "Yes" to each of these questions:

- Have you enabled In-App Purchases for your App ID?
- Have you checked Cleared for Sale for your product?
- Have you submitted (and optionally rejected) your application binary?
- Does your project's .plist Bundle ID match your App ID?
- Have you generated and installed a new provisioning profile for the new App ID?
- Have you configured your project to code sign using this new provisioning profile?
- Are you building for iPhone OS 3.0 or above?
- Have you waited several hours since adding your product to iTunes Connect?
- Are your bank details active on iTunes Connect? (via Mark)
- Have you tried deleting the app from your device and reinstalling? (via Hector, S3B, Alex O,Joe, and Alberto)
- Is your device jailbroken? If so, you need to revert the jailbreak for IAP to work. (via oh my god,Roman, and xfze)