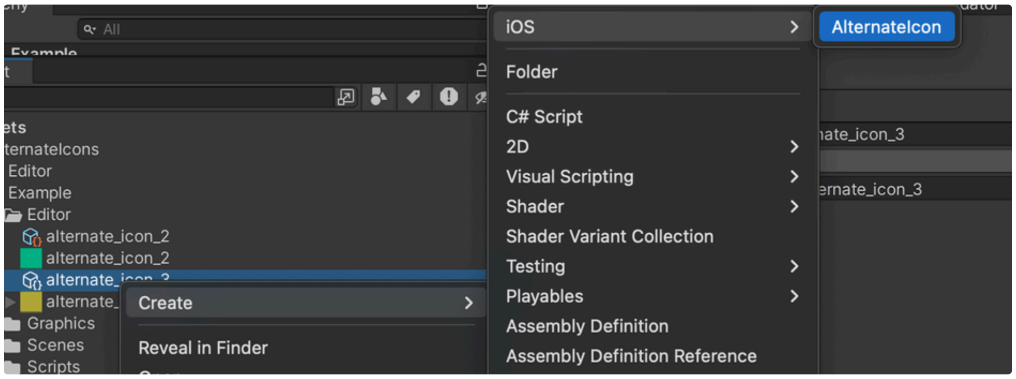
Alternate Icons for iOS

Adding alternate app icons to your app allows users to customize their home screen with an app icon that fits their style or you can test new app icon without app release in app store.

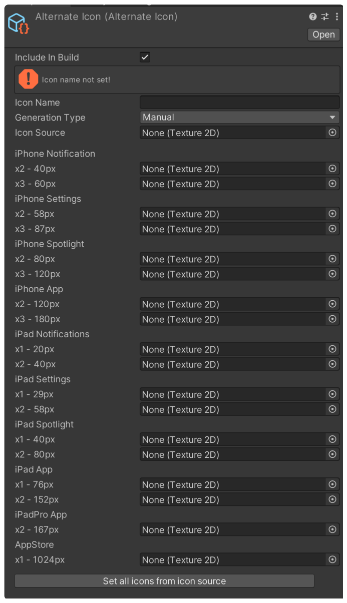
Setup

For add alternate app icons to your build you need to create **ScriptableObject** with type **AlternateIcon,** with contain information about icon in any **Editor** folder.
You can use context menu for creation **Create → iOS → AlternateIcon**



Recommendation: change name of created asset to icon name for simple navigation and understanding.

AlternateIcon asset:



Alternate Icon (Alternate Icon)

Open

Include In Build	✓
Icon name not set!	
Icon Name	
Generation Type	Manual
Icon Source	None (Texture 2D)
iPhone Notification	
x2 - 40px	None (Texture 2D)
x3 - 60px	None (Texture 2D)
iPhone Settings	
x2 - 58px	None (Texture 2D)
x3 - 87px	None (Texture 2D)
iPhone Spotlight	
x2 - 80px	None (Texture 2D)
x3 - 120px	None (Texture 2D)
iPhone App	
x2 - 120px	None (Texture 2D)
x3 - 180px	None (Texture 2D)
iPad Notifications	
x1 - 20px	None (Texture 2D)
x2 - 40px	None (Texture 2D)
iPad Settings	
x1 - 29px	None (Texture 2D)
x2 - 58px	None (Texture 2D)
iPad Spotlight	
x1 - 40px	None (Texture 2D)
x2 - 80px	None (Texture 2D)
iPad App	
x1 - 76px	None (Texture 2D)
x2 - 152px	None (Texture 2D)
iPadPro App	
x2 - 167px	None (Texture 2D)
AppStore	
x1 - 1024px	None (Texture 2D)

Set all icons from icon source

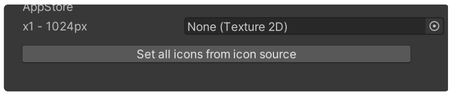
Include In Build - flag **true** if need include this app icon in build and **false** if need ignore this icon.
Icon Name - name to use alternate icon in AppStore or for change icon from app in runtime.

Generation Type - type of icon generation:

1. **Manual -** manual setup icon for each iOS source, for example: **iPhoneNotification** icons.
2. **Auto -** icon will be setup app for all iOS sources automatic.

Icon Source - reference to icon texture

When selected **Generation Type = Manual,** you can see button: '**Set all icons from source'**



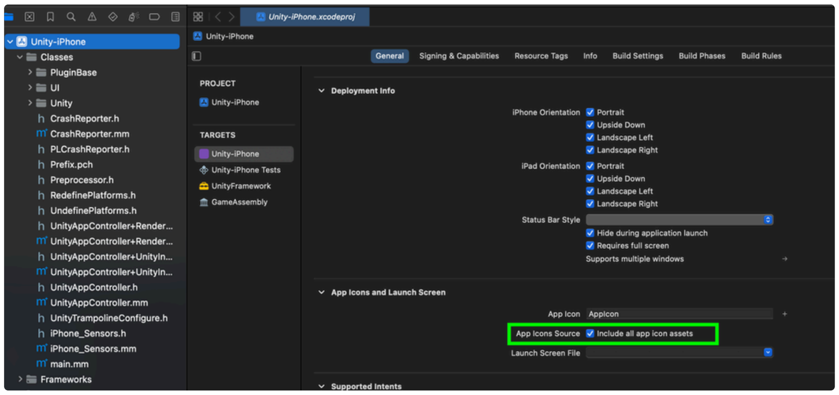
When this button pressed, all the services icon will be set as **Icon Source**.

Build 🔗

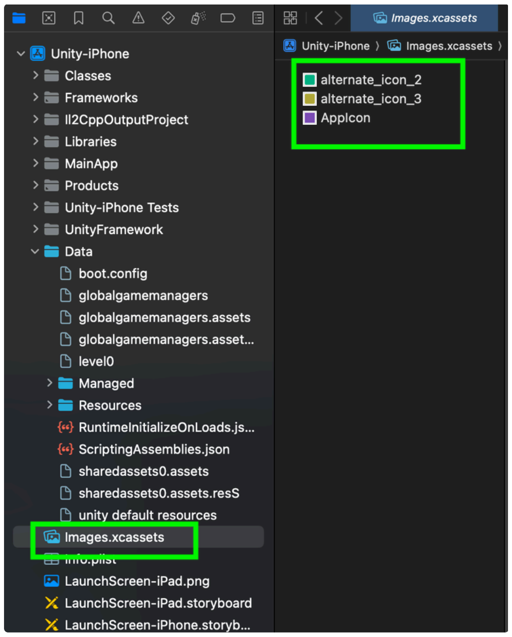
When you starting build your application, all your alternate icons will be found and add to Xcode project and be available for use.

In order to check whether icons were added to the Xcode project after the build:

1. In the **General** tab, the **Include all app icons assets** checkbox should be checked.



2. Select **Unity-Iphone** → **Data** → **Images** in the hierarchy and there should be assets of all the icons you added.



Runtime 🔗

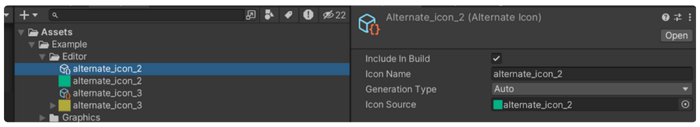
For change application icon in runtime use class **ApplicationIcon.**

Member	Description

static bool **IsSupportsAlternateIcons** { get; }	Returned **true** when device support alternate icons logic.
static string **AlternateIconName** { get; }	Returned current app icon name, if app icon set to default **AlternateIconName** returned **null**.
static bool **IsDefaultIconDisplayed** { get; }	Return **true** if currently the app is displaying the default icon
static void **SetAlternateIcon**(string iconName, ResultCallback onCompleted)	This method changes the icon the system displays for the app by icon name ⚠ your alternate icon must be included in build • **onCompleted** returned result of this operation, argument **isSuccess** will be **true** if app icon changed to alternative. When **onCompleted** returned **isSuccess** as **false** reason set in **errorMassage**.
static void **SetDefaultIcon**(ResultCallback onCompleted)	This method set the default icon the system displays for the app. • **onCompleted** returned result of this operation argument **isSuccess** will be **true** if app icon set to default successfully. When **onCompleted** returned **isSuccess** as **false** reason set in **errorMassage**.

Example:

1. Create **AlternateIcon** with name **alternate_icon_2** and set up his.



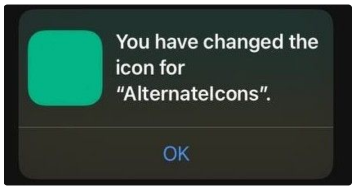
Alternate_icon_2 (Alternate Icon)

Include In Build ✓
Icon Name alternate_icon_2
Generation Type Auto
Icon Source alternate_icon_2

2. Create **TestController**, an add code below

```
    public class TestController : MonoBehaviour
    {
        private void Start()
        {
            ApplicationIcon.SetAlternateIcon("alternate_icon_2", (isSuccess, errorMessage) =>
            {
                Debug.Log(isSuccess
                    ? "Alternate icon is set successfully"
                    : $"Alternate icon set with error: {errorMessage}");
            });
        }
    }
```

3. Add **TestController** to your game object in Scene.
4. Build Unity and Xcode project.
5. Run application, and you will see a native message about successfully changed app icon


You have changed the icon for "AlternateIcons".