Unity SDK Setup

The Seeds Unity SDK

(source code)

Asset Packages (Contains SDK, Documentation, Script Samples):

- Unity 5.x
- Unity 4.x

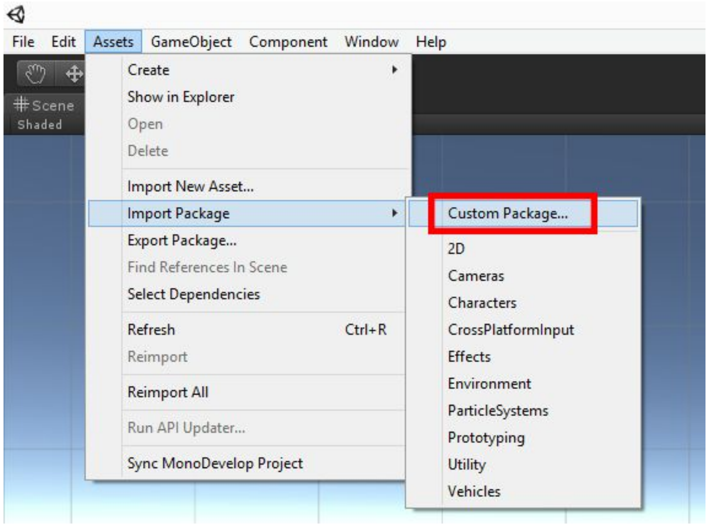
New! iOS support now available.

The Seeds Unity Package is a wrapper for our Android and iOS SDKs which are themselves built with production-tested open source components including Countly Android SDK.

Installation Guide

1. Import the Seeds Unity SDK into Your Project

From the Unity editor "Assets" menu, select "Import Package" => "Custom Package..." and select the package that you downloaded.



(Note: if you get an error saying assets/plugins or other folders are mapped to something else. Try the following workaround: 1) rename the folders with errors 2) import the Seeds package 3) rename the original folders back to their original names.)

2. Configure the Seeds prefab

Select the Seeds prefab in the Project view in Assets/Seeds. If you have an app_key, enter it into the App_Key field in the inspector. Otherwise, use "test" as a default value until your app_key is assigned. Make sure that the "Auto Initialize" check box is checked.

3. Display the Seeds promo interstitial

Use the Seeds instance to load:

```
Seeds.Instance.RequestInAppMessage();
```

and show:

```
Seeds.Instance.ShowInAppMessage();
```

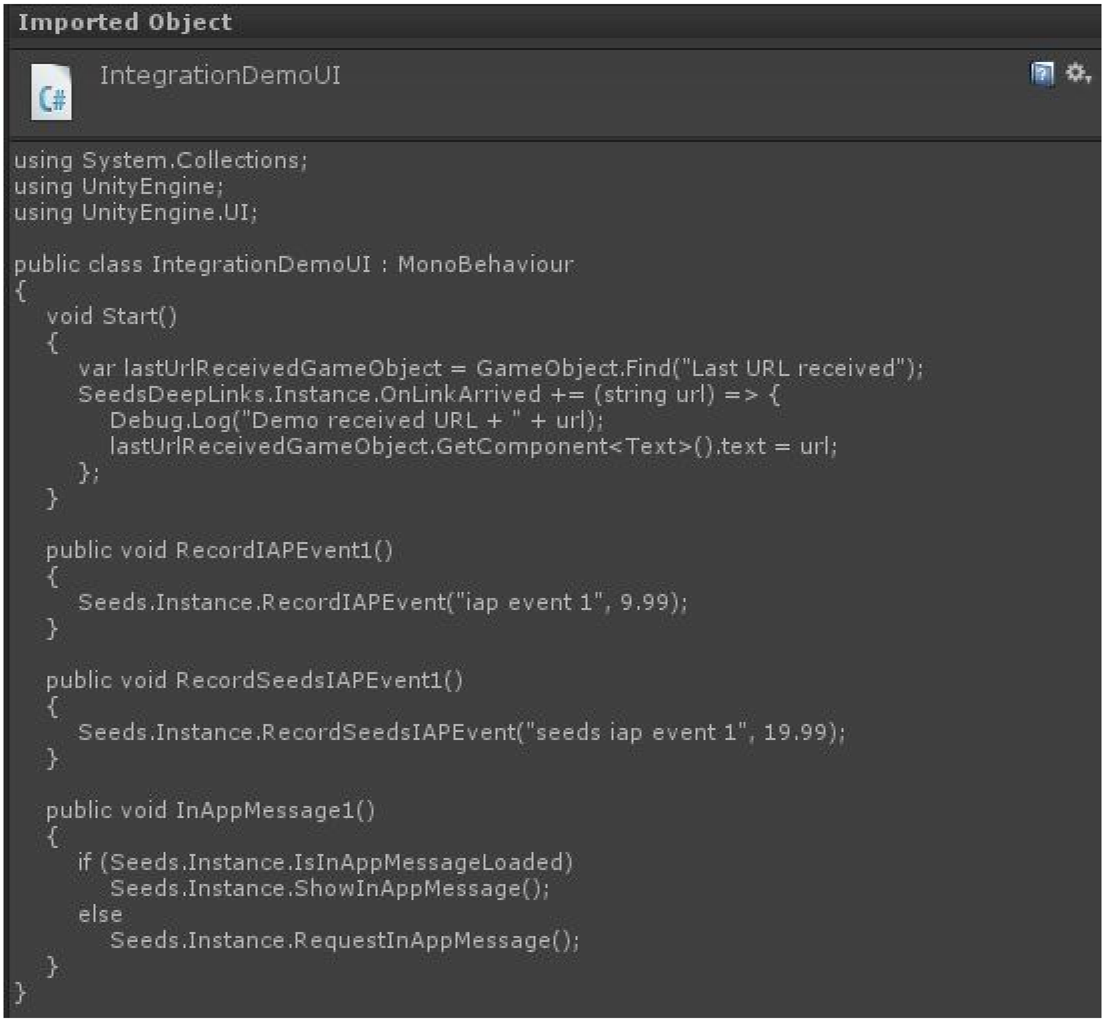
the interstitial promo in your code.

You may wish to add a helper method to your activity to accomplish these functions, e.g.:

```
public void ShowOrLoadSeedsPromo()
    {
        if (Seeds.Instance.IsInAppMessageLoaded)
            Seeds.Instance.ShowInAppMessage();
        else
```

```
            Seeds.Instance.RequestInAppMessage();
    }
```

**note: feel free to reference the demo script included in the asset package, shown below



```
using System.Collections;
using UnityEngine;
using UnityEngine.UI;

public class IntegrationDemoUI : MonoBehaviour
{
    void Start()
    {
        var lastUrlReceivedGameObject = GameObject.Find("Last URL received");
        SeedsDeepLinks.Instance.OnLinkArrived += (string url) => {
            Debug.Log("Demo received URL + " + url);
            lastUrlReceivedGameObject.GetComponent<Text>().text = url;
        };
    }

    public void RecordIAPEvent1()
    {
        Seeds.Instance.RecordIAPEvent("iap event 1", 9.99);
    }

    public void RecordSeedsIAPEvent1()
    {
        Seeds.Instance.RecordSeedsIAPEvent("seeds iap event 1", 19.99);
    }

    public void InAppMessage1()
    {
        if (Seeds.Instance.IsInAppMessageLoaded)
            Seeds.Instance.ShowInAppMessage();
        else
            Seeds.Instance.RequestInAppMessage();
    }
}
```

4. Track the item purchase

In your item store code, please include the following tracking code after a purchase of the Seeds-promoted item:

```
Seeds.Instance.RecordSeedsIAPEvent("ITEM", PRICE);
```

and for regular non-Seeds-promoted items:

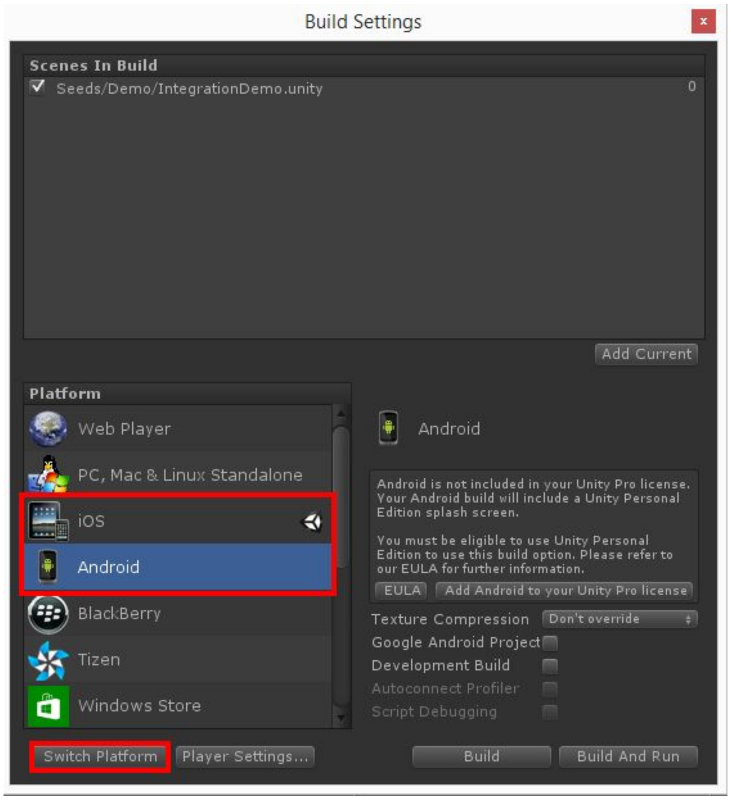
```
Seeds.Instance.RecordIAPEvent("ITEM", PRICE);
```

where ITEM is a string that is the name or SKU of the item and PRICE is a double representing the price of the item.

5 Setting up Deep Links

1) Ensure that you are working with the correct build settings

In order to configure Deep Links, your designated platform must be set to either Android or iOS. If your build settings are already set to the iOS or Android platforms, you can move on to 2) Configure. Otherwise, you can change your target platform from the Build Settings menu. Select iOS or Android, and then click the Switch Platform button at the bottom left of the window..



☑ Seeds/Demo/IntegrationDemo.unity 0

Android is not included in your Unity Pro license. Your Android build will include a Unity Personal Edition splash screen.

You must be eligible to use Unity Personal Edition to use this build option. Please refer to our EULA for further information.

Texture Compression: Don't override
Google Android Project □
Development Build □
Autoconnect Profiler □
Script Debugging □

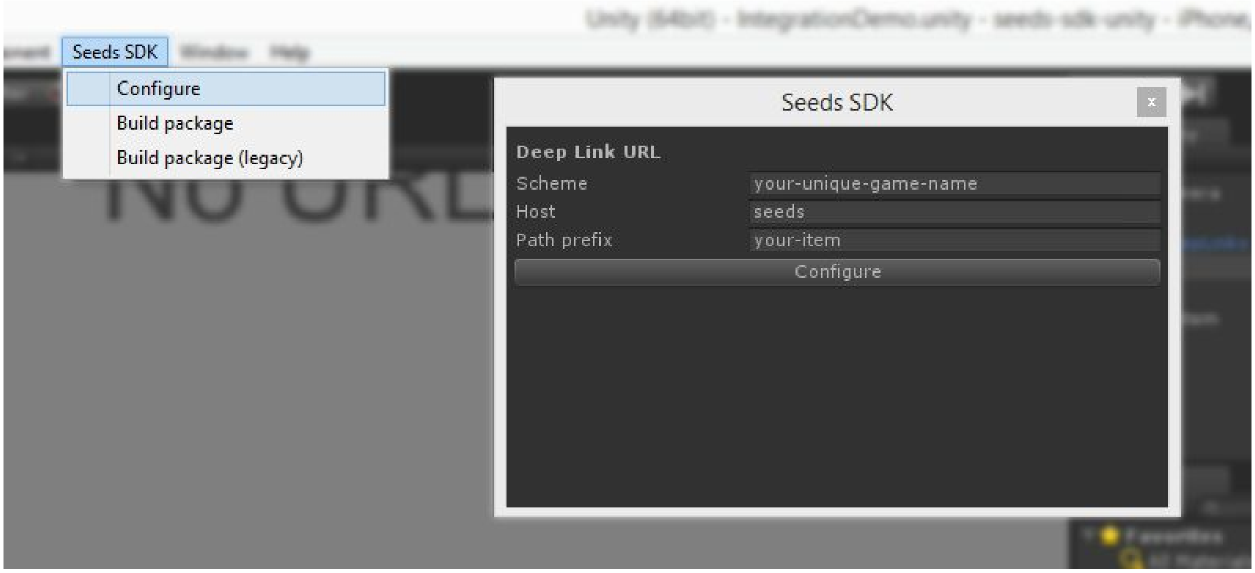
2) Configure

From the Seeds SDK Menu => select ‘Configure’* and input the deep link URI that will link from the ad to the in-app purchase item.

Scheme, Host, and Path Prefix correspond to these parts of the URI:

scheme://host/path-prefix

e.g. `your-unique-game-name://seeds/your-item'



Deep Link URL

Scheme your-unique-game-name

Host seeds

Path prefix your-item

2) Insert code snippet

Insert this code in the class where you can display your in-app item for purchase:

```
SeedsDeepLinks.Instance.OnLinkArrived += (string url) => {
    // code to display item
};
```

(see the demo app for example code)

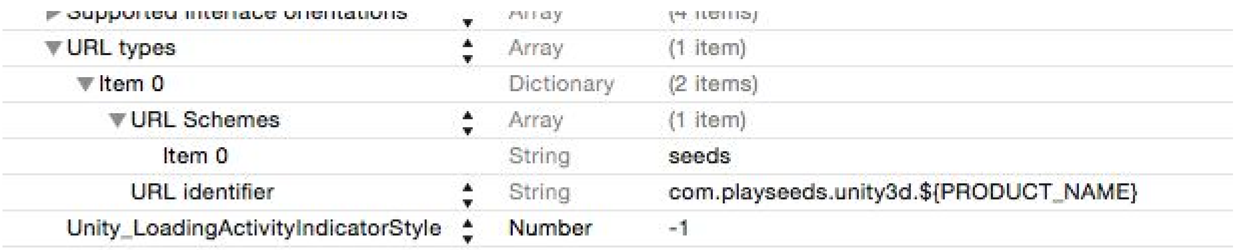
Additional steps for iOS Builds

Building for iOS will export code for Xcode. In Xcode,

1. add the Objc flag
2. Please follow the instructions in

   https://developer.apple.com/library/mac/qa/qa1490/_index.html
3. (Note: please just add it, do not replace existing flags)
4. Add CoreData and CoreTelephony framework references
5. add deep link info in Xcode:

- In the Project Navigator view, select the Info.plist file.
- Right-click (or control-click) in the file and select Add Row from the context menu.
- Scroll down and select the URL Types section.
- Expand the URL Types field and then expand the Item 0 field
- Use a unique String for the URL identifier, using the Bundle Identifier is a good choice
- Click the + next to Item 0 and add a URL Schemes field.
- In Item 0 under URL Schemes add the scheme that you input in the Configure panel in Unity, e.g. your-unique-game-name



▸ Supported interface orientations	Array	(4 items)
▾ URL types	Array	(1 item)
▾ Item 0	Dictionary	(2 items)
▾ URL Schemes	Array	(1 item)
Item 0	String	seeds
URL identifier	String	com.playseeds.unity3d.${PRODUCT_NAME}
Unity_LoadingActivityIndicatorStyle	Number	-1