Native Toolkit

iOS / Android / Windows Phone 8 Plugin

First of all thanks for purchasing Native Toolkit. This Unity plugin allows easy access to native functionality on iOS, Android and Windows Phone 8 devices. It also contains a wrapper for the Facebook SDK to make it simpler to use (it is not intended as a replacement for the Facebook SDK however!). There are specific setup steps for each platform so please read this document carefully and study the demo scenes and scripts before using the plugin in your code.

Native Toolkit has been developed using Unity 4.6 and therefore this is the minimum version of Unity you should use it with. It has been tested to compile with the Unity IL2CPP backend and works great on 64-bit devices. Native Toolkit has been tested and is working with Unity 5.

There are demo scenes located in the Assets/Native Toolkit/Examples folder – one showing native functionality and the other demonstrating the Facebook wrapper. You will need to configure the Facebook SDK before the Facebook Scene will function correctly, please read on for how to do so.
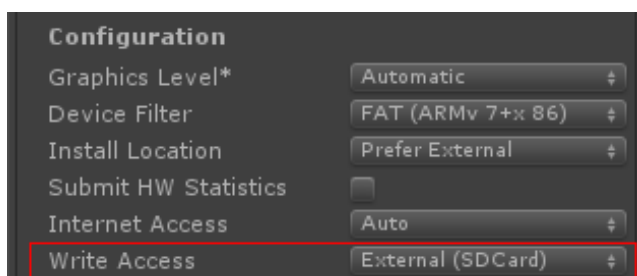
If you are looking for a scripting reference for Native Toolkit this can be found at:

http://secondfury.com/nativetoolkit/

It contains a description of each function, what it does, the parameters it takes and any callbacks and/or return values.

Android setup

If you are using Android, you will need to go into **Build Settings** and access the **Player Settings** panel. Change **Write Access** to **External (SDCard)** to allow your app to save images and screenshots to the users device.

If you are using the provided version of AndroidManifest.xml, this will already be pre-populated with the necessary permissions and activities required to run Native Toolkit and all of its functionality. If you are making the additions manually the following is required:

```
<activity android:name="com.secondfury.nativetoolkit.MainActivity"
android:configChanges="orientation|screenSize" />
<activity android:name="com.secondfury.nativetoolkit.LocalNotificationResult" />
<receiver android:name="com.secondfury.nativetoolkit.LocalNotification"/>
```

The following permissions should also be added to the manifest. If you are not using a certain feature of Native Toolkit, it's fine to remove that particular permission:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.VIBRATE"/>
```

Windows Phone 8 setup

First of all if you are building for WP8, please make sure your platform is set to Windows Phone 8 on the **Build Settings** panel. To use any of the photo/media functionality you must ensure the capability **ID_CAP_MEDIALIB_PHOTO** is specified in the app manifest. To use the GPS/Location functionality the capability **ID_CAP_LOCATION** needs to be specified.

After building your app you can find the manifest at `<BUILDPATH>/Properties/WMAppManifest.xml`

An extra step for **Unity 5** users is that the import settings must be changed on the plugin .dll files:

Assets/Plugins/NativeToolkitWP8.dll (platforms: all except WP8Player, CPU: Any CPU, OS: Any OS)

Assets/Plugins/WP8/NativeToolkitWP8.dll (platforms: WP8Player only, CPU: Any CPU, Don't process: Unchecked, Placeholder: Assets/Plugins/NativeToolkitWP8.dll)

iOS setup

No additional setup steps are required, but sometimes Xcode may throw errors due to Packages not being included (this can happen when you upgrade Xcode). To automatically include used packages, make sure you have Enable Modules set to "Yes" under Xcode Build Settings.

▼ Apple LLVM 6.1 - Language - Modules	
Setting	Unity-iPhone
Allow Non-modular Includes In Framework Modules	No ◇
▶ Enable Modules (C and Objective-C)	Yes ◇
Link Frameworks Automatically	Yes ◇

Native Toolkit

Facebook Setup

Native Toolkit provides a wrapper for the Facebook SDK which makes it simpler to use. To keep the plugin as lightweight as possible, the unitypackage that you have download from the Unity Asset Store does not contain the Facebook SDK.

Therefore to use Facebook you need to download the SDK, extract it to your project and then setup your Facebook App. It's a relatively straightforward procedure but it's always best to follow one of these installation guides:

https://developers.facebook.com/docs/unity/getting-started/ios

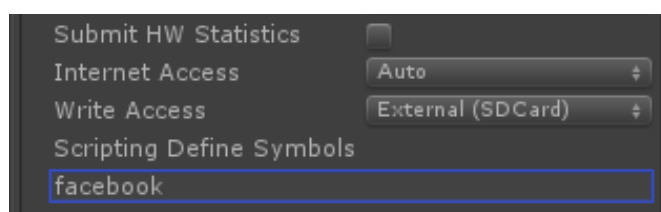https://developers.facebook.com/docs/unity/getting-started/android

Native Toolkit has been tested with the latest version of the Facebook SDK. Once you have completed the integration process, you can use all the functionality shown in the Native Toolkit Scripting API (link at the start of this document).

Remember – FacebookInit must be called before executing any other Facebook functions, and a user must be logged in (using FacebookLogin) before any status/image shares.

If you are having problems getting Facebook to work within your app, it's worth checking a few things:

- Make sure the App name and Id has been entered into Facebook Settings, found under **Facebook → Edit Settings** in top bar.

- Make sure you have entered the correct bundle ID on your Facebook Developers App page, and for Android, the correct package name, keyhash (found under **Facebook → Edit Settings**) and class (which should be com.facebook.unity.FBUnityDeepLinkingActivity).

- You should also ensure that the word **facebook** is specified in your Scripting Define Symbols in Player Settings**.** If you end up removing the Facebook SDK from your project please go ahead and remove this, otherwise you will see errors.

- On Android devices, be sure you have added the necessary Facebook Additions to the AndroidManifest.xml. These can be automatically added by pressing the "Regenerate Android Manifest" button under **Facebook → Edit Settings,** or alternatively they are listed below for you to copy/paste into the manifest.

```
<activity android:name="com.facebook.unity.FBUnityLoginActivity" android:screenOrientation="portrait"
android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|navigation|screenLayout|screenSize|
smallestScreenSize|uiMode|touchscreen"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"></activity>
<activity android:name="com.facebook.LoginActivity" android:screenOrientation="portrait"
android:configChanges="keyboardHidden"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"></activity>
<activity android:name="com.facebook.unity.FBUnityDeepLinkingActivity" android:exported="true"></activity>
<meta-data android:name="com.facebook.sdk.ApplicationId" android:value="\ 1533459673590952" />
<activity android:name="com.facebook.unity.FBUnityDialogsActivity" android:screenOrientation="portrait"
android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|navigation|screenLayout|screenSize|
smallestScreenSize|uiMode|touchscreen"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"></activity>
```

A point to note is that the Facebook SDK is not used on Windows Phone 8 devices, as it has built in dialogs for status and image sharing, which Native Toolkit will utilise on this platform.

Support

If you're experiencing difficulties with the plugin, we recommend you take a good look through the demo scenes and scripts, and also the scripting reference, the URL of which is above. If this doesn't resolve your problems you can contact us at the below email address, quoting your Unity Asset Store invoice number:

contact@secondfury.com

Unfortunately we will not be able to answer support emails that do not provide a Unity Asset Store invoice number, due to an inability to verify the purchase. Also please note that we do assume a certain level of programming ability – and that we are not providing a C# apprenticeship by email :)

If you have any requests or suggestions about how to improve the plugin, we'd be very happy to hear from you!

And finally if you like the plugin and it has helped you with your project, it would be great if you could review it on the Asset Store, it's always good to help out a fellow dev!