

i-mobile for SP Application 広告表示SDK Unity プラグイン 設定資料



2015年6月版

目次

1. 概要
2. アプリケーションの登録、広告スポットの作成
3. サンプルアプリケーションの使用方法
4. プラグイン適用手順
 - 4.1. プラグイン適用手順1 (iOS、Android共通)
 - 4.2-a. プラグイン適用手順2-a (iOS)
 - 4.2-b. プラグイン適用手順2-b (Android)
5. 広告の表示方法
 - 5.1. 全画面広告の表示方法 (全画面広告共通)
 - 全画面広告の表示方法 (起動時に広告を表示する)
 - 全画面広告の表示方法 (広告を表示する向きを設定)
 - 5.2. インライン広告の表示方法 (インライン広告共通)
 - インライン広告の表示方法 (表示・非表示の切り替え)
 - インライン広告の表示方法 (デバイスの横幅に合わせて拡大表示)
6. 広告表示の状態を通知するイベント
 - 6.1. 広告表示の状態を通知するイベントを受け取る方法
 - 6.2. 広告表示の状態を通知するイベントの種類
7. その他
 - ・ AndroidアプリケーションのPlayerSettingsのDefaultOrientationに「AutoRotation」を指定している場合に、広告表示がおかしくなる

1. 概要

- はじめに
この資料はUnityアプリに広告を表示するために必要な、Unity用のプラグインの設定方法を記載した技術資料です。

- 対象となるサービス



- 対象となる読者
Unity アプリ開発者

- 対象プラットフォーム

iOS iOS向けにビルドする場合、imobile_for_SP_app_iOS_SDK_2.0.5 以上が別途必要です

Android Android向けにビルドする場合、imobile_for_SP_app_android_SDK_2.0.5 以上が別途必要です

※可能な限り最新版のSDKを使用してください。

- 広告表示方式

| | |
|-------------------|---|
| 全画面広告 | インタースティシャル広告 / ウォール広告 / テキストポップアップ広告 |
| インライン広告 | バナー広告 [320x50] / ビッグバナー広告 [320x100] / ミディウムレクタングル [300x250] [Androidのみ] アイコン広告 [57x57] ※タイトル表示時は[75x75] |
| iPad向け インライン広告 | バナー広告 [468x60] / ビッグバナー広告 [768x90] / スカイスクレイパー広告 [120x600] ワイドスカイスクレイパー広告 [160x600] / スクエア(小)広告 [200x200] / スクエア広告(250x250) レクタングル広告[300x250] / レクタングル(大)広告[336x280] / ハーフページ [300x600] |

2. アプリケーションの登録、広告スポットの作成

アプリケーションに広告を表示するためには、「アプリケーションの登録」と「広告スポットの作成」が必要です。

アプリケーションの登録、および広告スポットの作成方法についての手順は、各IMobileSdkAdsに同梱されている広告表示SDK 設定資料の「管理画面の設定」を、ご参照ください。

【iOS】

SDK ダウンロードページの

「i-mobile 広告表示SDK iOS版設定資料.pdf」の「管理画面の設定」を、ご参照ください。

※ PDF については、以下のURL からアクセス可能です。

https://sppartner.i-mobile.co.jp/downloads/sdk/i-mobile_SDK_manual_for_iOS.pdf

【Android】

SDK ダウンロードページの

「i-mobile 広告表示SDK Android版設定資料.pdf」の「管理画面の設定」を、ご参照ください。

※ PDF については、以下のURL からアクセス可能です。

https://sppartner.i-mobile.co.jp/downloads/sdk/i-mobile_SDK_manual_for_Android.pdf

3. サンプルアプリケーションの使用方法

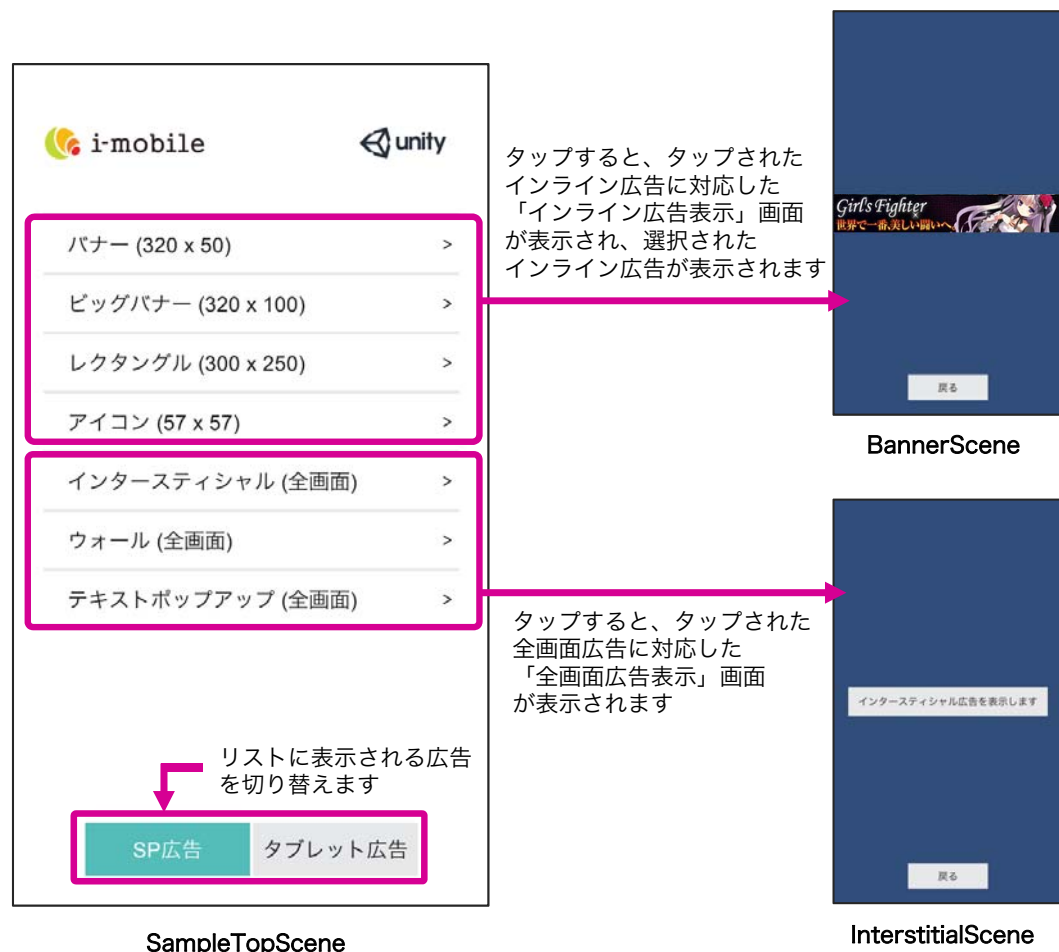
サンプルアプリケーションプロジェクトをUnityで読み込みビルド後に実行してください。

※ サンプルプロジェクトはUnity4.6.0以降での動作を想定しております。それ以前のバージョンでは動作確認を行っておりません。

※ iOSの場合は、ビルド後にSDKの設定等の手順が必要となります。

「i-mobile 広告表示SDK iOS版設定資料.pdf」 (https://sppartner.i-mobile.co.jp/downloads/sdk/i-mobile_SDK_manual_for_iOS.pdf)の

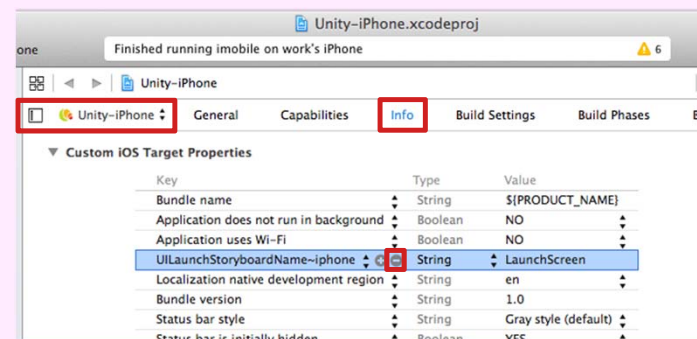
「XcodeによるSDKの設定」を、ご参照ください。



Xcodeでの実行時に例外が発生する場合

>> You are using Unity iPhone Basic. You are not allowed to remove the Unity splash screen from your game

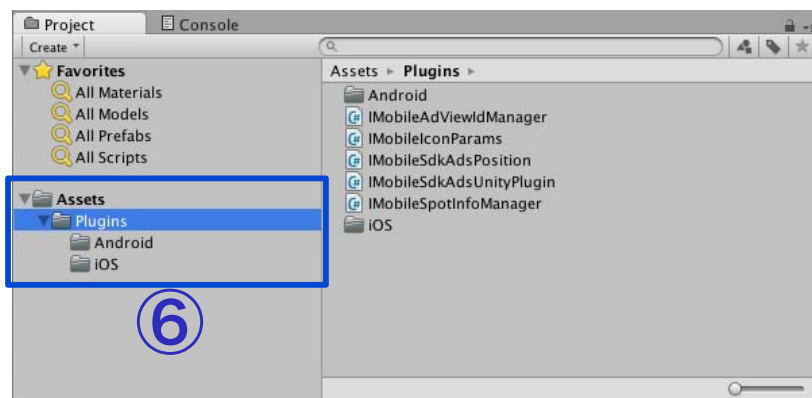
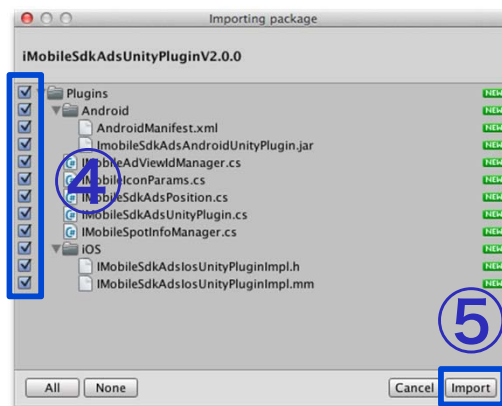
という例外が発生し、サンプルアプリケーションが実行できない場合は、Custom iOS Target Properties から、<< UILaunchStoryboardName~iphone >> という項目を削除したうえで、再度お試しください。



4.1. プラグイン適用手順 (iOS、Android共通)

プラグインの適用手順について説明します。

- ① プラグインを組み込むUnityプロジェクトを開きます。
- ② 「iMobileSdkAdsUnityPlugin_2.x.x.unitypackage」をダブルクリックします。
- ③ importing package ダイアログが開きます。
- ④ 全てのファイルにチェックが入っていることを確認します。
- ⑤ Importボタンをクリックします。
- ⑥ 正常にImportが完了すると、ProjectタブのAssets以下にPluginsフォルダが展開されます。



以上でプラグインの適用は完了となります。

4.2a. プラグイン適用手順 (iOS)

本手順は、iOS向けにアプリをビルドする際に必要となる手順です。(Androidアプリのみビルドする場合は不要です。)

- ① i-mobile管理画面から『imobile_for_SP_app_iOS_SDK_2.x.x.zip』をダウンロードします。
(本手順の中でSDKに同梱されているファイルが必要になります。)
- ② UnityからXcode用のプロジェクトをビルドし、Xcodeを起動します。
- ③ 必要なframeworkをプロジェクトに適用します。
※ 詳細な適用手順については、SDKダウンロードページの『i-mobile 広告表示SDK iOS版設定資料.pdf』の「3. アプリの設定 - 1. XcodeによるSDKの設定2」から「3. アプリの設定 - 1. XcodeによるSDKの設定3」を参照してください。
また、適用に必要な各種ファイルも、同ファイルに同梱されています。

本手順は、最初にXcodeでビルドする前に一度だけ行う手順です。
以降は、UnityからXcode用のプロジェクトを出力する際、表示されるWarningに対して、「Append」を選択することにより、上記の手順で設定した内容を維持したまま、Xcode用のプロジェクトが生成されます。



※ここで、「Replace」を選択した場合、上記の手順で設定した内容は全て消えてしまうため、ご注意ください。

以上が、iOS向けにアプリをビルドする際に必要となる手順です。

4.2b. プラグイン適用手順(Android)

本手順は、Android向けにアプリをビルドする際に必要となる手順です。(iOSアプリのみビルドする場合は不要です。)

- ① i-mobile管理画面から『imobile_for_SP_app_android_SDK_x.x.x.zip』をダウンロードします。SDKに同梱されている『imobileSdkAds.jar』をAssets/Plugins/Androidにドラッグアンドドロップします。

本バージョンより、SDKを動作させるために、Google Play Servicesの適用が必要になりました。Google Play Servicesを適用するためには、下記操作を行い、ビルドする必要があります。

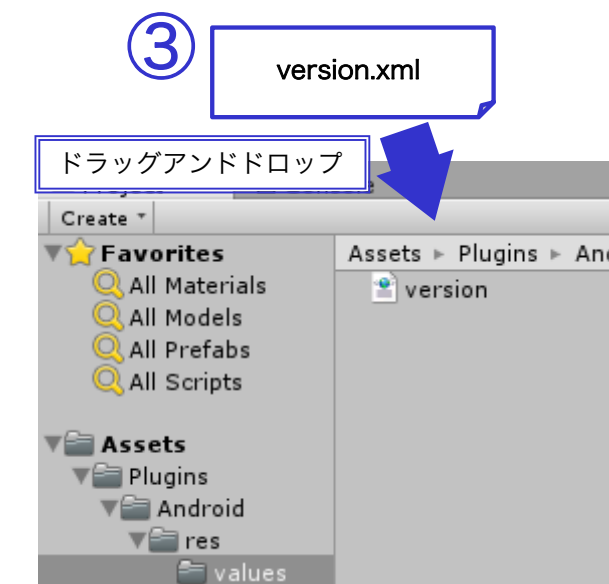
- ② Assets/Plugins/Androidに『google-play-services.jar』をドラッグアンドドロップします。
 - ※ 『google-play-services.jar』は以下のフォルダにあります。
“ANDROID_HOME”/extras/google/google_play_services/libproject/google-play-services_lib/libs
 - ※ Google Play Servicesのインストール方法等は、SDKダウンロードページの『i-mobile 広告表示SDK Android版設定資料.pdf』の「よくある質問と回答」の「■Google Play Servicesのインストール方法」をご参照ください。
- ③ Assets/Plugins/Android/res/valuesに、Google Play Servicesのversion.xmlを追加します。
 - ※Google Play Servicesのversion.xmlは以下のフォルダにあります。
“ANDROID_HOME”/extras/google/google_play_services/libproject/google-play-services_lib/res/values

独自のAndroidManifest.xmlを使用する場合は、別途以下の対応が必要です。
AndroidManifest.xmlを開きます

- ④ AndroidManifest.xmlに以下を追記します。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<application>
  <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version" />
</application>
```

以上が、Android向けにアプリをビルドする際に必要となる手順です。



5.1. 全画面広告の表示方法 (全画面広告共通)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

■ 広告の取得開始

[設定方法]

MonoBehaviourの「void Awake()」などのメソッドで広告表示を登録するためのメソッド
「IMobileSdkAdsUnityPlugin.registerFullScreen()」と開始するためのメソッド
「IMobileSdkAdsUnityPlugin.start()」を実行することにより、広告の取得が開始されます。

[記述例]

```
void Awake () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.registerFullScreen(publisherId, mediald, spotId);  
        IMobileSdkAdsUnityPlugin.start(spotId);  
    #endif  
}
```

複数回「registerFullScreen ()」や、「start ()」を呼び出しても問題ございません。
「IMobileSdkAdsUnityPlugin.registerFullScreen()」は、imobile_for_Interstitial_Unity_SDK_1.0.0から定義されているメソッド
「IMobileSdkAdsUnityPlugin.register ()」と同等のメソッドです。既に「IMobileSdkAdsUnityPlugin.register ()」で記述されている場合、
書き換える必要はございません。

■ 広告の表示

[設定方法]

MonoBehaviourの「void OnGUI()」、「void Update()」などのメソッドで広告表示を行うためのメソッド
「IMobileSdkAdsUnityPlugin.show()」を呼び出すことにより、広告が表示されます。

[記述例]

```
void OnGUI () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.show(spotId);  
    #endif  
}
```

参考:メソッドの引数

| 引数 | 設定値 |
|--------------------|---------------------------------|
| publisherId | i-mobileの管理画面より割り当てられたパブリッシャーID |
| mediald | i-mobileの管理画面より割り当てられたメディアID |
| spotId | i-mobileの管理画面より割り当てられたスポットID |

5.1. 全画面広告の表示方法 (起動時に広告を表示する)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

※ この表示方法は、テキストポップアップ広告のみ使用することを推奨します。

■ 広告の取得開始

[設定方法]

Cocos2dx::AppDelegateの「void applicationDidFinishLaunching ()」などで、
広告表示を登録するためのメソッド「IMobileCocos2dxModule::registerSpotFullScreen ()」を呼び出し後、
広告の取得が開始されます。
広告表示を行うためのメソッド「IMobileCocos2dxModule::show()」を呼び出すことにより、広告が表示されます。

[記述例]

```
void Awake () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.registerFullScreen(publisherId, mediald, spotId);  
        IMobileSdkAdsUnityPlugin.start(spotId);  
        IMobileSdkAdsUnityPlugin.show(spotId);  
    #endif  
}
```

参考:メソッドの引数

| 引数 | 設定値 |
|--------------------|---------------------------------|
| <i>publisherId</i> | i-mobileの管理画面より割り当てられたパブリッシャーID |
| <i>mediald</i> | i-mobileの管理画面より割り当てられたメディアID |
| <i>spotId</i> | i-mobileの管理画面より割り当てられたスポットID |

※ インターstitial広告やウォール広告についても同様の実装は可能ですが、
テキストポップアップ広告と比べ、広告の取得に時間がかかるため、多くの場合、
「show」呼び出し時に 広告表示準備が完了していないため、広告の表示がされません。
また、テキストポップアップ広告については、「show」が呼び出されてから約2秒以内に
広告の取得および表示ができない場合は、広告の表示をキャンセルします。

5.1. 全画面広告の表示方法 (広告を表示する向きを設定)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

■ 広告を表示する向きの設定

[設定方法]

広告の表示向きを設定するためのメソッド「IMobileSdkAdsUnityPlugin.setAdOrientation()」を呼び出すことで、広告の表示向きを指定することができます。

※この設定は必須ではありません。

設定をしなない場合は、デバイスの向きに合わせて広告が表示されます(AUTO設定)。

[記述例]

```
void Awake () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.registerFullScreen(publisherId, mediald, spotId);  
        IMobileSdkAdsUnityPlugin.start(spotId);  
        // 広告の表示向きを指定 以下の場合は、常に横向きで広告が表示されます  
        IMobileSdkAdsUnityPlugin.setAdOrientation(IMobileSdkAdsUnityPlugin.ImobileSdkAdsAdOrientation.IMOBILESDKADS_AD_ORIENTATION_LANDSCAPE);  
    #endif  
}
```

※記載する場所は、IMobileSdkAdsUnityPlugin.show()を呼び出す前であれば、どこに記載しても問題ありません。

参考:メソッドの引数

| 引数 | 設定値 | |
|-----------------------------------|---|-------------------|
| publisherId | i-mobileの管理画面より割り当てられたパブリッシャーID | |
| mediald | i-mobileの管理画面より割り当てられたメディアID | |
| spotId | i-mobileの管理画面より割り当てられたスポットID | |
| ImobileSdkAdsAdOrientation | 広告の表示向き | |
| | IMOBILESDKADS_AD_ORIENTATION_AUTO | デバイスの向きに合わせて広告を表示 |
| | IMOBILESDKADS_AD_ORIENTATION_PORTRAIT | 縦向きで広告を表示 |
| | IMOBILESDKADS_AD_ORIENTATION_LANDSCAPE | 横向きで広告を表示 |

5.2. インライン広告の表示方法 (インライン広告共通)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

■ 広告の取得と表示

[設定方法]

MonoBehaviourの「void Awake()」などのメソッドで広告表示を登録するためのメソッド

「IMobileSdkAdsUnityPlugin.registerInline ()」、開始するためのメソッド「IMobileSdkAdsUnityPlugin.start()」、表示するためのメソッド「IMobileSdkAdsUnityPlugin.show()」を実行することにより、 広告の取得と表示が行われます。

[記述例]

```
void Awake () {
    #if UNITY_IPHONE || UNITY_ANDROID
        IMobileSdkAdsUnityPlugin.registerInline(publisherId, mediaId, spotId);
        IMobileSdkAdsUnityPlugin.start(spotId);
        // 表示位置を座標で指定する場合
        IMobileSdkAdsUnityPlugin.show(spotId, IMobileSdkAdsUnityPlugin.AdType.BANNER, left, top);
        // 表示位置を定数パラメータで指定する場合 以下の場合は、画面中央下部に表示されます
        IMobileSdkAdsUnityPlugin.show(spotId, IMobileSdkAdsUnityPlugin.AdType.BANNER,
            IMobileSdkAdsUnityPlugin.AdAlignPosition.CENTER, IMobileSdkAdsUnityPlugin.AdValignPosition.BOTTOM);
    #endif
}
```

参考:メソッドの引数

| 引数 | 設定値 | | | | | | |
|--|---|-------------|-----------|---------------|------------|---------------|-----------|
| publisherId | i-mobileの管理画面より割り当てられたパブリッシャーID | | | | | | |
| mediaId | i-mobileの管理画面より割り当てられたメディアID | | | | | | |
| spotId | i-mobileの管理画面より割り当てられたスポットID | | | | | | |
| IMobileSdkAdsUnityPlugin.AdType | 表示する広告種別をIMobileSdkAdsUnityPlugin.AdTypeで指定 ※IMobileSdkAdsUnityPlugin.AdTypeについては、「4.2. 広告の表示方法2 (インライン広告共通 AdType)」を、ご参照ください。 | | | | | | |
| left | 表示する位置のX座標 | | | | | | |
| top | 表示する位置のY座標 | | | | | | |
| AdAlignPosition | 表示する位置の横位置 <table border="1"><tr><td>LEFT</td><td>左寄せで広告を表示</td></tr><tr><td>CENTER</td><td>中央寄せで広告を表示</td></tr><tr><td>RIGHT</td><td>右寄せで広告を表示</td></tr></table> | LEFT | 左寄せで広告を表示 | CENTER | 中央寄せで広告を表示 | RIGHT | 右寄せで広告を表示 |
| LEFT | 左寄せで広告を表示 | | | | | | |
| CENTER | 中央寄せで広告を表示 | | | | | | |
| RIGHT | 右寄せで広告を表示 | | | | | | |
| AdValignPosition | 表示する位置の縦位置 <table border="1"><tr><td>TOP</td><td>上寄せで広告を表示</td></tr><tr><td>MIDDLE</td><td>中央寄せで広告を表示</td></tr><tr><td>BOTTOM</td><td>下寄せで広告を表示</td></tr></table> | TOP | 上寄せで広告を表示 | MIDDLE | 中央寄せで広告を表示 | BOTTOM | 下寄せで広告を表示 |
| TOP | 上寄せで広告を表示 | | | | | | |
| MIDDLE | 中央寄せで広告を表示 | | | | | | |
| BOTTOM | 下寄せで広告を表示 | | | | | | |

5.2. インライン広告の表示方法 (インライン広告共通 AdType)

■ IMobileSdkAdsUnityPlugin.AdTypeについて

インライン広告を表示する場合は、表示する広告の種別をshowメソッドに渡します。
広告種別は列挙型(IMobileSdkAdsUnityPlugin.AdType)で指定します。

指定できる広告種別は下記の通りです。

| IMobileCocos2dxModule::AdType | |
|-------------------------------|---|
| ICON | アイコン ※ Androidでは表示されますが、iOSでは表示されません |
| BANNER | バナー |
| BIG_BANNER | ビッグバナー |
| MEDIUM_RECTANGLE | ミディアムレクタングル |
| TABLET_BANNER | タブレットバナー (※iPadのみ) |
| TABLET_BIG_BANNER | タブレットビッグバナー (※iPadのみ) |
| BIG_RECTANGLE | ビッグレクタングル (※iPadのみ) |
| SKYSCRAPER | スカイクレイパー (※iPadのみ) |
| WIDE_SKYSCRAPER | ワイドスカイクレイパー (※iPadのみ) |
| SQUARE | スクウェア (※iPadのみ) |
| SMALL_SQUARE | スモールスクエア (※iPadのみ) |
| HALFPAGE | ハーフページ (※iPadのみ) |

[記述例]

```
void Awake () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.registerInline(publisherId, mediaid, spotid);  
        IMobileSdkAdsUnityPlugin.start(spotid);  
        IMobileSdkAdsUnityPlugin.show(spotid, IMobileSdkAdsUnityPlugin.AdType.BANNER, left, top);  
    #endif  
}
```

5.2. インライン広告の表示方法 (表示・非表示の切り替え)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

■ 広告の表示・非表示を切り替える

以下のソースのように、showメソッドの戻り値として渡されるadViewIdを第一引数、falseを第二引数に指定し、setVisibilityメソッドを呼び出すことで広告を一時的に非表示にすることができます。

また、setVisibilityメソッドの第二引数にtrueを指定することで、非表示にした広告を再表示できます。

[記述例]

```
int adViewId = IMobileSdkAdsUnityPlugin.show (DefinelmobileSdkAdsID.bbSpotid,
                                             IMobileSdkAdsUnityPlugin.AdType.BigBanner,
                                             IMobileSdkAdsUnityPlugin.AdAlignPosition.Center,
                                             IMobileSdkAdsUnityPlugin.AdValignPosition.Bottom);

// 広告を非表示にします
IMobileSdkAdsUnityPlugin.setVisibility(adViewId, false);

// 非表示にした広告を再表示します
IMobileSdkAdsUnityPlugin.setVisibility(adViewId, true);
```

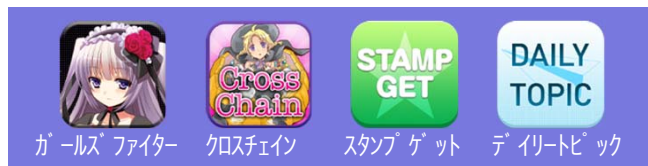
参考：メソッドの引数

| 引数 | 設定値 |
|----------|--------------------------|
| adViewId | Showメソッドの戻り値として渡される固有のID |

5.2. インライン広告の表示方法 (アイコン広告の表示形式カスタマイズ)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

- アイコン広告の表示形式カスタマイズ ※ アイコン広告は、Androidでのみ動作します
アイコン広告の場合、前述の組み込み方法で組み込みを行った場合、以下の様に表示されます。



この表示形式は、IMobileIconParamsにパラメータを設定し、showメソッドへ渡すことにより変更することができます。設定可能なパラメータの詳細については、IMobileSdkAdsに同梱されている『i-mobile for SP Application 広告表示SDK Version2 iOS版.pdf』の「3. アプリの設定 - 2. 広告の表示方法5 (アイコンの表示形式のカスタマイズ)」をご参照ください。
※Androidの場合も、同様のパラメータが使用できます。

[記述例]

```
void Awake () {  
    #if UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.registerInline (publisherId, mediaId, spotId);  
        IMobileSdkAdsUnityPlugin.start(spotId);  
        var iconParams = new IMobileIconParams ();  
        iconParams.iconNumber = 6; // アイコンの数を6個に  
        iconParams.iconTitleFontColor = "#000000"; // タイトルの色を黒に  
        iconParams.iconTitleShadowEnable = false; // タイトルの影を非表示に  
        IMobileSdkAdsUnityPlugin.show(spotId, IMobileSdkAdsUnityPlugin.AdType.ICON, left, top, iconParams); // 設定したスタイル渡して表示  
    #endif  
}
```

参考:メソッドの引数

| 引数 | 設定値 |
|--|---|
| publisherId | i-mobileの管理画面より割り当てられたパブリッシャーID |
| mediaId | i-mobileの管理画面より割り当てられたメディアID |
| spotId | i-mobileの管理画面より割り当てられたスポットID |
| IMobileSdkAdsUnityPlugin.AdType | 表示する広告種別をIMobileSdkAdsUnityPlugin.AdTypeで指定 ※IMobileSdkAdsUnityPlugin.AdTypeについては、「4.2. 広告の表示方法2 (インライン広告共通 AdType)」を、ご参照ください。 |
| left | 表示位置のX座標 |
| top | 表示位置のY座標 |
| iconParams | アイコンの表示形式 |

5.2. インライン広告の表示方法 (デバイスの横幅に合わせて拡大表示)

以下は、「4.2. プラグイン適用手順」まで完了していることを前提とします。

■ デバイスの横幅に合わせて拡大表示

バナー広告(320 x 50)、ビックバナー広告(320 x 100)は、デバイスの幅が320のデバイスに合わせた広告サイズですが、最近では、より大きなサイズのデバイスも増えてきており、従来のバナー広告(320 x 50)、ビックバナー広告(320 x 100)を表示した場合、大きく余白ができてしまいます。



横幅320
のデバイス



横幅375
のデバイス

375px - 320px = 55px 分の余白ができます

showメソッドのSizeAdjust引数に「true」を指定した場合、バナーを横幅いっぱいまで拡大表示し、余白ができない様に表示することができます。

※拡大されるサイズは、デバイスサイズの短い方のサイズまでです。

```
// 広告をデバイスサイズに合わせて表示します
IMobileSdkAdsUnityPlugin.show( spotid, IMobileSdkAdsUnityPlugin.AdType.BANNER,
                                IMobileSdkAdsUnityPlugin.AdAlignPosition.Center,
                                IMobileSdkAdsUnityPlugin.AdValignPosition.Bottom, true); // 座標指定の場合でも同様にSizeAdjust引数が使用できます
```



横幅375
のデバイス

320x50のバナーが375x59 [50 × (375 ÷ 320) ※端数は切り上げ]に拡大され、余白無しで表示されます
※座標指定する場合は、Y座標も拡大されることを考慮した値にする必要があります。

6.1. 広告表示の状態を通知するイベントを受け取る方法

以下は、「5. 広告の表示方法」まで完了していることを前提とします。

■ 広告表示の状態通知イベントを受け取るオブジェクトの登録

[設定方法]

「IMobileSdkAdsUnityPlugin.addObserver()」メソッドを使用して、広告表示の状態通知イベントを受け取るオブジェクトを登録することにより、各種状態通知イベントを受け取ることができます。

[記述例]

```
void Awake () {  
    #if UNITY_EDITOR  
    #elif UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.addObserver(gameObjectName);  
    #endif  
}
```

※addObserverメソッド実行後、直ぐに通知が始まります。

基本的に、使用する箇所に制限はありませんが、「start()」メソッドの呼び出し前に設定することにより、発生する全てのイベントを受け取ることができます。(受け取りたいイベントに合った適切な場所で呼び出してください。)

■ 広告表示の状態通知イベントを受け取るオブジェクトの解除

[設定方法]

「IMobileSdkAdsUnityPlugin.removeObserver()」を呼び出すことにより、登録した状態通知イベントを受けるオブジェクトを解除することができます。

[記述例]

```
void OnDestroy() {  
    #if UNITY_EDITOR  
    #elif UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.removeObserver(gameObjectName);  
    #endif  
}
```

参考:メソッドの引数

| 引数 | 設定値 |
|----------------|-------------------------------------|
| gameObjectName | 状態通知イベントを受け取りを解除するゲームオブジェクト名を設定します。 |

6.2. 広告表示の状態を通知するイベントの種類

以下は、「6.1.広告表示の状態を通知するイベントを受け取る方法」まで完了していることを前提とします。

■ 広告表示の状態を通知するイベントの種類

[イベントの種類と引数]

各種イベントは、以下のメソッドに通知されます。また、引数としてイベントの補足情報がCSV形式(カンマ区切り)で渡されます。

※実装していないメソッドに対しては、通知されません。

| メソッド名 | 内容 | 引数の1番目の値 | 引数の2番目の値 |
|----------------------------------|---|-----------------|--|
| imobileSdkAdsSpotDidReady(Value) | 広告の表示準備が完了した際に呼ばれます | イベントが発生したスポットID | ありません |
| imobileSdkAdsSpotDidFail(Value) | 広告の取得が失敗した際に呼ばれます | イベントが発生したスポットID | 1: 権限エラー(指定されたスポットが無効) 2: レスポンスエラー (サーバからのレスポンスの形式に問題がある等) 3: ネットワークエラー(ネットワークの利用ができない) 4: ネットワークエラー(タイムアウト等の通信エラー全般) 5: 不明なエラー(予期しないエラー) 6: 広告表示実行時に広告の準備ができていない(ロード中) 7: 広告切れ(配信できる広告がない) |
| imobileSdkAdsSpotDidShow(Value) | 広告が表示された際に呼ばれます | イベントが発生したスポットID | ありません |
| imobileSdkAdsSpotDidClick(Value) | 広告がクリックされた際に呼ばれます | イベントが発生したスポットID | ありません |
| imobileSdkAdsSpotDidClose(Value) | 広告が閉じられた際に呼ばれます (広告の表示をスキップした場合も呼ばれます) | イベントが発生したスポットID | ありません |

[記述例]

```
public class GameObject : MonoBehaviour {
    void imobileSdkAdsSpotDidReady (string value) {
        Debug.Log("imobileSdkAdsSpotDidReady:" + value);
    }
    void imobileSdkAdsSpotDidFail (string value) {
        Debug.Log("imobileSdkAdsSpotDidFail:" + value);
    }
    void imobileSdkAdsSpotDidShow (string value) {
        Debug.Log("imobileSdkAdsSpotDidShow:" + value);
    }
    void imobileSdkAdsSpotDidClick (string value) {
        Debug.Log("imobileSdkAdsSpotDidClick:" + value);
    }
    void imobileSdkAdsSpotDidClose (string value) {
        Debug.Log("imobileSdkAdsSpotDidClose:" + value);
    }
}
```

7. その他

- AndroidアプリケーションのPlayerSettingsのDefaultOrientationに「AutoRotation」を指定している場合に、広告表示がおかしくなる

【Android】

AutoRotationのAllowed Orientations for Auto Rotationの設定で、PortraitとLandscapeが混在している場合に、広告表示後に画面を回転させると、正しいレイアウトで広告を表示することができません。

もし、PortraitとLandscapeが混在したAutoRotation設定を行う場合は、以下の対応がUnity側のスクリプトにて別途必要になります。

- 広告表示の状態通知イベントを受け取るオブジェクトの登録(登録方法は、「5.1. 広告表示の状態を通知するイベントを受け取る方法」参照)
- showメソッドを呼び出す際に、画面モードを固定化するコードを追加

```
void OnGUI () {  
    #if UNITY_ANDROID  
        switch (Screen.orientation) {  
            case ScreenOrientation.Landscape:  
                Screen.orientation = ScreenOrientation.Landscape;  
                break;  
            case ScreenOrientation.Portrait:  
                Screen.orientation = ScreenOrientation.Portrait;  
                break;  
        }  
    #elif UNITY_IPHONE || UNITY_ANDROID  
        IMobileSdkAdsUnityPlugin.show(spotid);  
    #endif  
}
```

- 広告が閉じられた際、およびクリックされた際に、画面モードの固定化を解除するコードを追加

```
void imobileSdkAdsSpotDidClick (string value) {  
    #if UNITY_ANDROID  
        Screen.orientation = ScreenOrientation.AutoRotation;  
    #endif  
    Debug.Log("imobileSdkAdsSpotDidClick:" + value);  
}  
  
void imobileSdkAdsSpotDidClose (string value) {  
    #if UNITY_ANDROID  
        Screen.orientation = ScreenOrientation.AutoRotation;  
    #endif  
    Debug.Log("imobileSdkAdsSpotDidClose:" + value);  
}
```