

iPhone 用 SDK 組み込み手順 ご説明資料

2013/07/17 version 1.2 r03

修正履歴

#	日付	version	release	修正内容
1	2013/01/14	1.0	01	初版。
2	2013/03/28	1.1	02	1-1 ファイルをプロジェクトに追加：一部記載修正
3	2013/07/17	1.2	03	3 アクセス解析組み込み手順追加

目次

1	ライブラリの組み込み手順.....	- 4 -
1-1	ファイルをプロジェクトに追加	- 4 -
1-2	DimageSDK.plist を作成.....	- 7 -
1-3	ソースの編集.....	- 9 -
1-4	動作確認.....	- 9 -
1-5	注意事項.....	- 9 -
2	LTV 組み込み手順.....	- 10 -
2-1	ファイルをプロジェクトに追加	- 10 -
2-5	ソースの編集.....	- 10 -
2-6	動作確認.....	- 13 -
2-7	注意事項.....	- 13 -
3	アクセス解析組み込み手順.....	- 13 -
3-1	ソースの編集.....	- 13 -
	動作確認.....	- 14 -

1 ライブラリの組み込み手順

下記の手順により、Dimage SDK を組み込んでいただくことができます。

組み込みますと、ユーザがアプリストアからアプリケーションをダウンロードし

初めてアプリを起動した際に一度だけブラウザが起動され

指定のページを表示できるようになります。

このタイミングで成果計測が行われます。

起動がオンライン環境でなかった場合にはブラウザは開かれず、次の起動時にリトライされます。

iOS 3.2 以降に対応しております。

1-1 ファイルをプロジェクトに追加

Xcode プロジェクトにファイルを追加

Xcode のメニューで「プロジェクト」 「プロジェクトに追加」を選択します。

解凍した SDK フォルダから以下のファイルを選択します。

iPhone シミュレータで動作させる場合：libDimageSDK_v01.1_dev.a

iPhone 実機で動作させる場合：libDimageSDK_v01.1_sim.a

ダイアログが表示されますので「ディスティネーショングループのフォルダに項目をコピーする」にチェックをした状態で追加します。



Finder などから直接 Xcode へドラッグ＆ドロップする方法ですと、

元のファイルの置き場所がリンカオプションの `LIBRARY_SEARCH_PATHS` に追加されてしまいますので

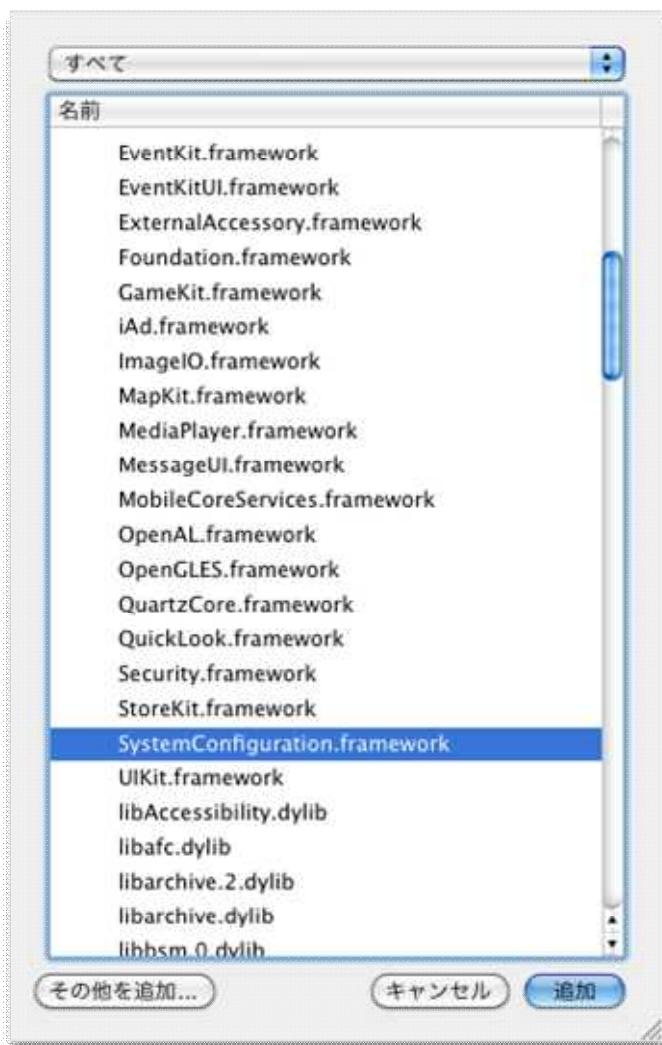
ご注意ください。

同様の手順で AdManager.h を追加します。

：フレームワークを追加

Xcode 内の「Frameworks」を選択して右クリックし、「追加」 「既存のフレームワーク」を選択します。

表示されたダイアログから「SystemConfiguration.framework」を選択して追加します。



1-2 DimageSDK.plist を作成

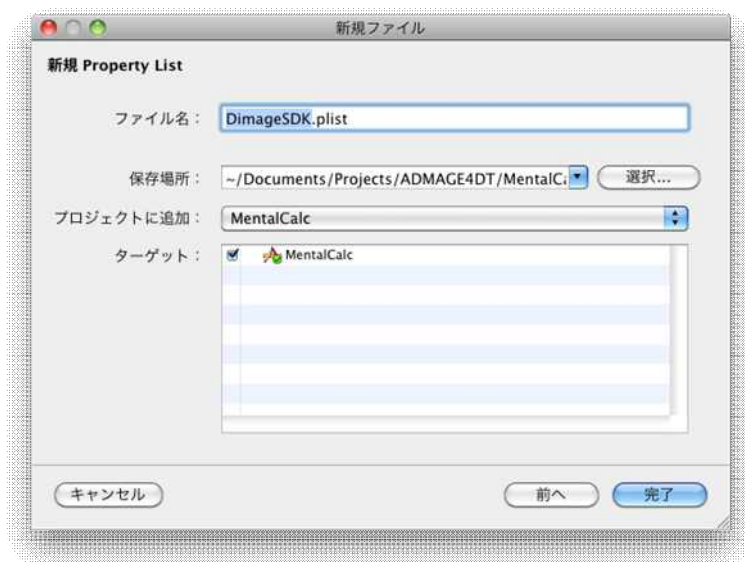
DimageSDK.plist を作成。

Xcode のメニューで「ファイル」 「新規ファイル」を選択します。

表示されたダイアログで「Mac OS X」 「Resource」 「Property List」を選択してプロパティリストを作成します。



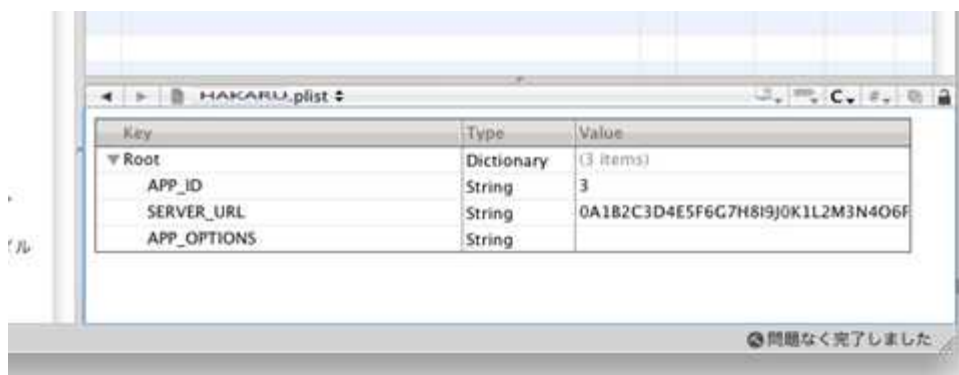
名前は DimageSDK.plist とします。



： DimageSDK.plist へ Key を追加。

作成した DimageSDK.plist を選択し、次の Key を追加します。

#	キー名	タイプ	値
1	APP_ID	String	AppTizer 管理画面より確認出来ますので、その値を入力してください(AppTizer 基本操作マニュアル『SDK 導入時「アプリ ID」確認方法』参照)
2	SERVER_URL	String	01a716bda7458e77e4cc58136efdc8c5236b508b581d93f0f609180d5ddb01a6ba 上記の文字列を設定ください。
3	APP_OPTIONS	String	空白のままにしてください



1-3 ソースの編集

：ヘッダファイルの#import を追加。

UIViewController を継承したクラスなど、アプリの起動時の処理があるクラスの.m ファイルに、次のように追加します。

```
#import "AdManager.h"
```

：アプリ起動時のソースに以下を追加。

次のように追加します。

```
[[DimageManager sharedManager] sendConversionWithStartPage:@" http://          "];
```

sendConversionWithStartPage 関数の引数は URL になります。起動時に表示させたいページの URL をご指定ください。

オンライン環境下での起動時に 1 度だけブラウザが開き、指定 URL のページが表示されるようになります。

ブラウザに表示するページが特に無い場合は

以下のように文字列 " default " を渡してください。

```
[[DimageManager sharedManager] sendConversionWithStartPage:@" default "];
```

デフォルトのページはアプリを特定せず汎用的にご使用いただくため、デザイン・内容ともに非常にシンプルなものとなっております。独自の thanks ページをご作成いただくことを推奨しております。

1-4 動作確認

Simulator と実機、それぞれで動作確認を行って下さい。

1-5 注意事項

成果通信は 1 度しか行われません。

ふたたび成果通信を行いたい場合は、当該アプリをアンインストール後再度インストールしてください。

2 LTV 組み込み手順

下記の手順により、LTV 計測処理を組み込んでいただくことができます。

組み込みますと、アプリ起動時成果後も継続的に成果計測をする事が可能になります。

手順 1 の libDimage ライブラリが追加されていることが前提になります。

2-1 ファイルをプロジェクトに追加

：Xcode プロジェクトにファイルを追加

1 . 1 と同様の手順で、解凍した SDK フォルダから以下のファイルをプロジェクトに追加します。

Ltv.h を追加します。

2-5 ソースの編集

：ヘッダファイルの#import を追加。

アプリ内で、Ltv 計測を行う予定の m ファイルに、次のように追加します。

```
#import "Ltv.h"
```

：LTV 処理のコードの追加

LTV の処理開始時に以下の様に追加してください。

```
DimageLtv *ltv = [[DimageLtv alloc] init];
```

・ アプリ内からの LTV 計測の場合 (広告主端末 ID 無し)

```
[ltv sendLtv:成果地点ID];
```

・ アプリ内からの LTV 計測の場合 (広告主端末 ID 有り)

```
[ltv sendLtv:成果地点ID :@" 広告主端末 ID "];
```

・ 上記パラメータの説明

#	パラメータ名	
1	成果地点 ID	AppTizer 管理画面より発行可能ですので、その値を入力してください (AppTizer 基本操作マニュアル「SDK 導入時 LTV 組み込み」成果地点 ID」 確認方法」参照)
2	広告主端末 ID	広告主様が管理しているユニークな識別子 (会員 ID など) を使用出来ます。指定 していただくと、広告主端末 ID を成果に含める事ができ、これを基準とした成果 計測が行えます。 退会計測する場合は必須です。

上記アプリ内成果時には、下記パラメータもオプションとして設定する事が可能です。

使用関数: [ltv addParam:@パラメータ名:@”任意の値”];

上記パラメータの説明

#	パラメータ名	
1	PARAM_SKU	Stock Keeping Unit(商品管理コード) 商品の在庫管理する際に使用してください
2	PARAM_PRICE	Price 売上額を管理する際に使用してください。
3	PARAM_OUT	Out 退会を管理する際に使用してください。 値は1を指定してください。 [広告主端末 ID]が必須です。 成果時に渡した [広告主端末 ID] を入れてください。

例: [ltv addParam:@ PARAM_PRICE:@”1000”];

その他、任意でパラメータを加える事も可能です。

[ltv addParameter:@任意のパラメータ名, @” 任意の値”];

[ltv addParameter:@任意のパラメータ名 2, @” 任意の値”];

1 同一パラメータ名を記述した場合は、後者が有効となります。

2 アンダースコア (“_”) をパラメータ名の先頭に記述しないでください。

・ ブラウザでの LTV 計測の場合

ブラウザで LTV 計測を行う場合、

アプリ外での計測になりますので、Cookie に記述しておく必要があります。

この為、ブラウザを起動する前のコードに以下の記述を追記してください。

アプリ外ブラウザの場合

```
[ltv ltvOpenBrowser:@" ブラウザで起動するURL "];
```

アプリ内ブラウザの場合

```
[ltv setLtvCookie];
```

アプリ外ブラウザ 組み込み例：

```
[ltv ltvOpenBrowser:@" http://          "];
```

アプリ内ブラウザ 組み込み例：

```
// Ltv Cookie 書き込み
[ltv setLtvCookie];

// URL 作成
NSURL *url = [NSURL URLWithString:@"http://          "];
NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];

// Webビューの生成
webView=[[self makeWebView:self.view.frame] retain];
[webView setDelegate:self];
[self.view addSubview:webView];

//WEB上のHTMLの読み込み
[webView loadRequest:request];
}
```

AppTizer 管理者より成果地点タグを通知致しますので、LTV 成果を発生させたいページに埋め込んでください。アプリ内ブラウザでそのページが表示された時に、成果通信が行われます。

成果タグ例

```

```

2-6 動作確認

Simulator と実機、それぞれで動作確認を行ってください。

2-7 注意事項

ブラウザでの LTV 計測の場合、

ブラウザを終了した段階で、セットした情報がリセットされますので、

内部で複数回ブラウザを立ち上げる場合は、その都度上記コードを記述する必要があります。

3 アクセス解析組み込み手順

下記の手順により、アクセス解析処理を組み込んでいただくことができます。

組み込みますと、アプリ起動数とページビュー計測をする事が可能になります。

手順 1 の libDimage ライブラリが追加されていることが前提になります。

3-1 ソースの編集

：Xcode プロジェクトにファイルを追加

1 . 1 と同様の手順で、解凍した SDK フォルダから以下のファイルをプロジェクトに追加します。

```
#import "AdManager.h"
```

：アクセス解析処理のコードの追加

以下、計測種類に応じて組み込んでください。

A: アプリ起動数計測

UIViewController を継承したクラスなど、アプリの起動時の処理があるクラスの.m ファイルに、次のように追加します。 **アクセス解析組み込み時はかならず起動数を計測ください。**

```
[[DimageManager sharedManager] track:@" 起動イベント ID "];
```

B: ページビュー計測

UIViewController を継承したクラスなど、各ページの起動時の処理があるクラスの.m ファイルに、次のように追加します。

```
[[DimageManager sharedManager] track:@" PV イベント ID "];
```

上記パラメータの説明

#	パラメータ名	説明
1	起動イベント ID	AppTizer 管理画面より発行可能ですので、その値を入力してください（AppTizer 基本操作マニュアル「SDK 導入時アクセス解析組み込み「アプリ起動タイプ ID」「ページビュータイプ ID」確認方法」参照）
2	PV イベント ID	AppTizer 管理画面より発行可能ですので、その値を入力してください（AppTizer 基本操作マニュアル「SDK 導入時アクセス解析組み込み「アプリ起動タイプ ID」「ページビュータイプ ID」確認方法」参照）

動作確認

Simulator と実機、それぞれで動作確認を行って下さい。