

# **nend SDK β 設定マニュアル**

iOS ver1.3.2

2012年9月20日

更新履歴	更新内容
iOS_ver 1.2.0 2011/8/16	–
iOS_ver 1.2.1 2011/8/24	シュミレーター向け不具合の修正
iOS_ver 1.2.2 2012/3/27	デバイス ID (UDID) 取得の停止 サイズ変更 (320×48) から (320×50) の View を生成するように変更
iOS_ver 1.2.3 2012/4/11	広告取得時に不必要なメモリをリリースしない不具合の修正
iOS_ver 1.3.0 2012/6/25	広告受信成功通知の追加 広告受信エラー通知の追加 定期ロード中断の追加 定期ロード再開の追加
iOS_ver 1.3.1 2012/8/31	WebView タイプの場合ローテーションが管理されない不具合修正
iOS_ver 1.3.2 2012/9/20	iOS SDK 6, iPhone 5 (armv7s) 対応

## 目次

◆nendSDK iOS について.....	4
◆組み込み方法.....	5
○プロジェクトへの nendSDK 追加.....	5
○「Security Framework」の追加.....	6
○ビルド.....	7
◆NADView の内容.....	8
○メソッド.....	8
○プロパティ.....	8
○Delegate.....	8
◆実装.....	9
○ヘッダファイル.....	9
(1) delegate 準拠.....	9
○実装ファイル.....	10
(2)NADView 生成.....	10
(3)apiKey,spotID のセット.....	10
(4) デリゲートオブジェクトの指定.....	11
(5) RootViewController の指定.....	11
(6) 広告のロード.....	11
(7) Delegate 通知.....	12
(8) 定期ロードの管理.....	13
(9) リリース.....	14
◆検証.....	15
◆よくある質問.....	16

## nendSDK 導入のおおまかな流れ

① nend の管理画面でアプリ登録、広告枠登録を行います。

※ 本マニュアルは広告枠登録後、「apiKey」「spotID」を発行しSDKを入手している前提で説明を行います。

※ 広告枠を登録していない場合には、別紙「管理画面マニュアル」PDF をご参照の上、ご申請ください。

※ 広告枠を申請後、広告枠の管理＞広告枠＞SDK＞「SDK をダウンロード」で SDK を入手できます。

② 本マニュアルに従って nendSDK をアプリに組み込みます。

※ 本マニュアルは、iOS 用 nendSDK 導入に関するマニュアルとなります。

## ◆nendSDK iOS について

### ファイル構成

NendSDK\_iOS.zip

NendAd/	SDK フォルダ
libNendAd.a	ライブラリ
NADView.h	ヘッダファイル
NADView_readme.txt	ライセンス文等
Samples/	サンプルソースフォルダ
NendSDKTestARC/	Single View Application, ARC, 画像回転時の実装例
NendSDKTestTabBase/	Tabbed Application, アニメーション表示を利用した実装例
nendSDK1.3.2_manual.pdf	本マニュアル

### 対応環境

#### 実機端末

iPhone 3GS 以上 iPhone 5、iOS 4.3 以上 iOS 6 が動作保障対象となります。

#### 開発環境

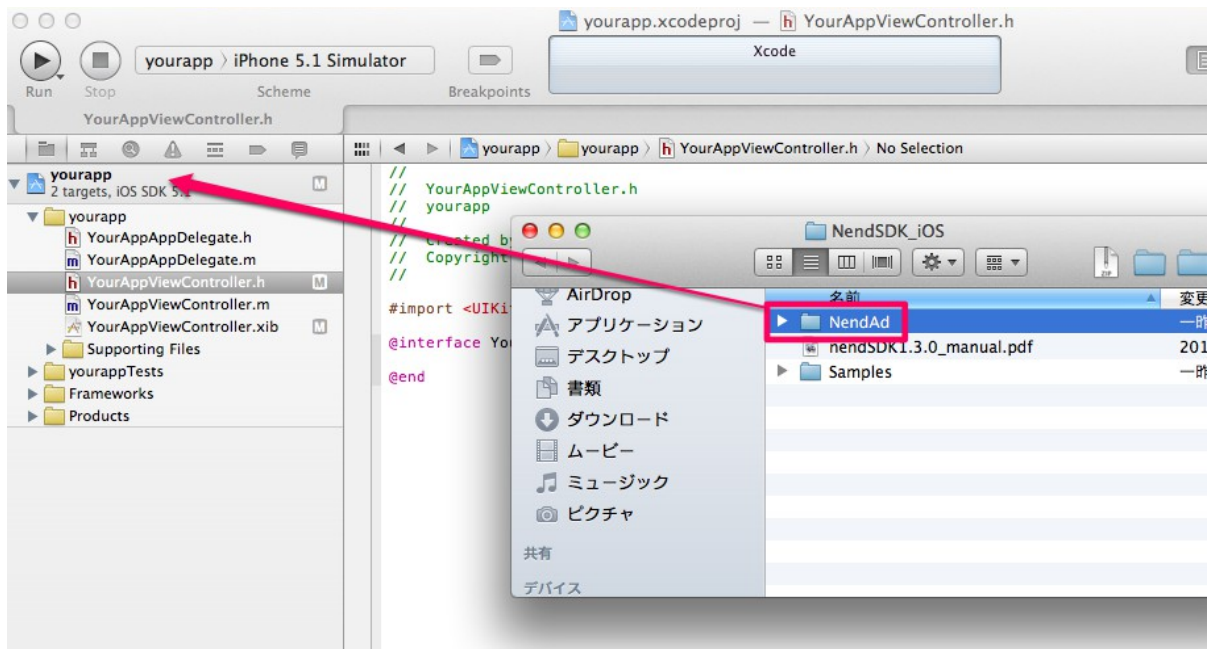
Xcode4.1& iOS SDK4.3 以上、Xcode 4.5 & iOS SDK 6 が対象です。

それ以外の環境では正常に動作しない場合があります。

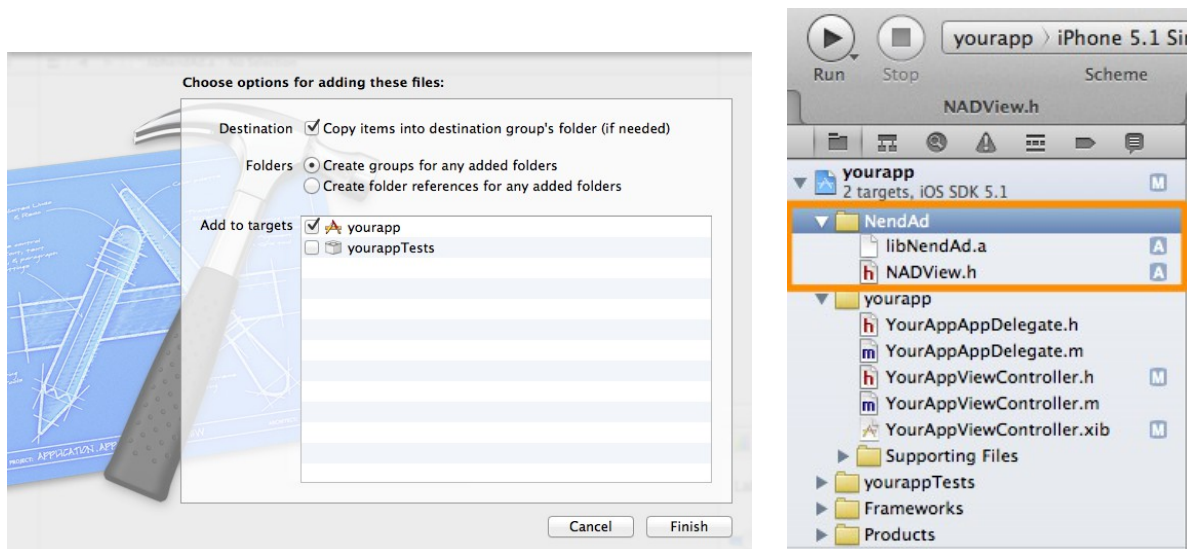
## ◆組み込み方法

### ○プロジェクトへの nendSDK 追加

Xcode 上で対象プロジェクトに「NendAd」フォルダごとドラッグ & ドロップします。



下記、左図ダイアログの表示で、必要に応じて任意の設定を選び「Finish」をクリックします。



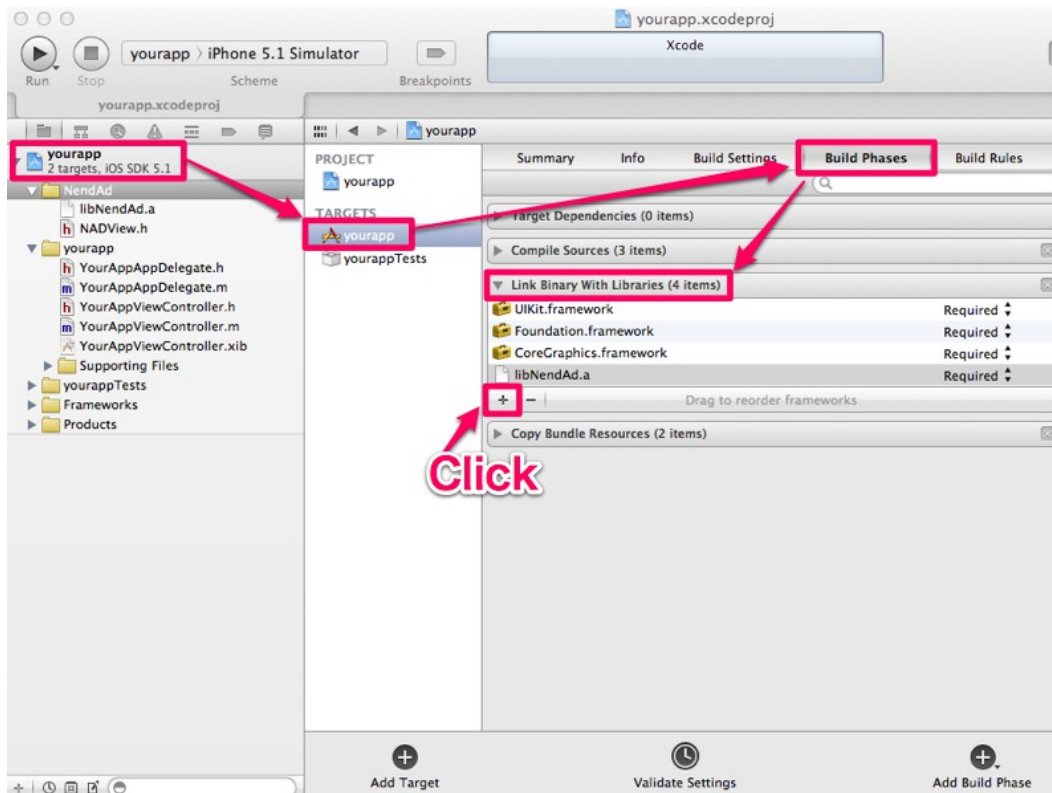
追加が完了すると Xcode のファイルリストに右図のように表示されます。

## ○「Security Framework」の追加

nendSDK の利用には Security.framework が必要です。

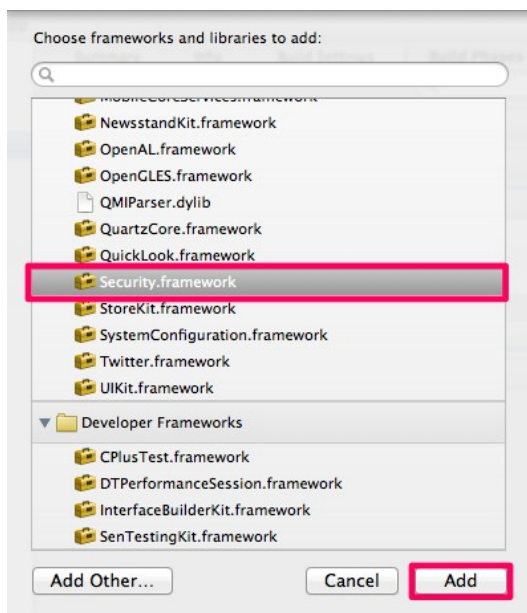
左側のプロジェクトナビゲーターから、プロジェクトをクリックして

TARGETS>Build Phases>Link Binary With Libraries 項目を開きます。



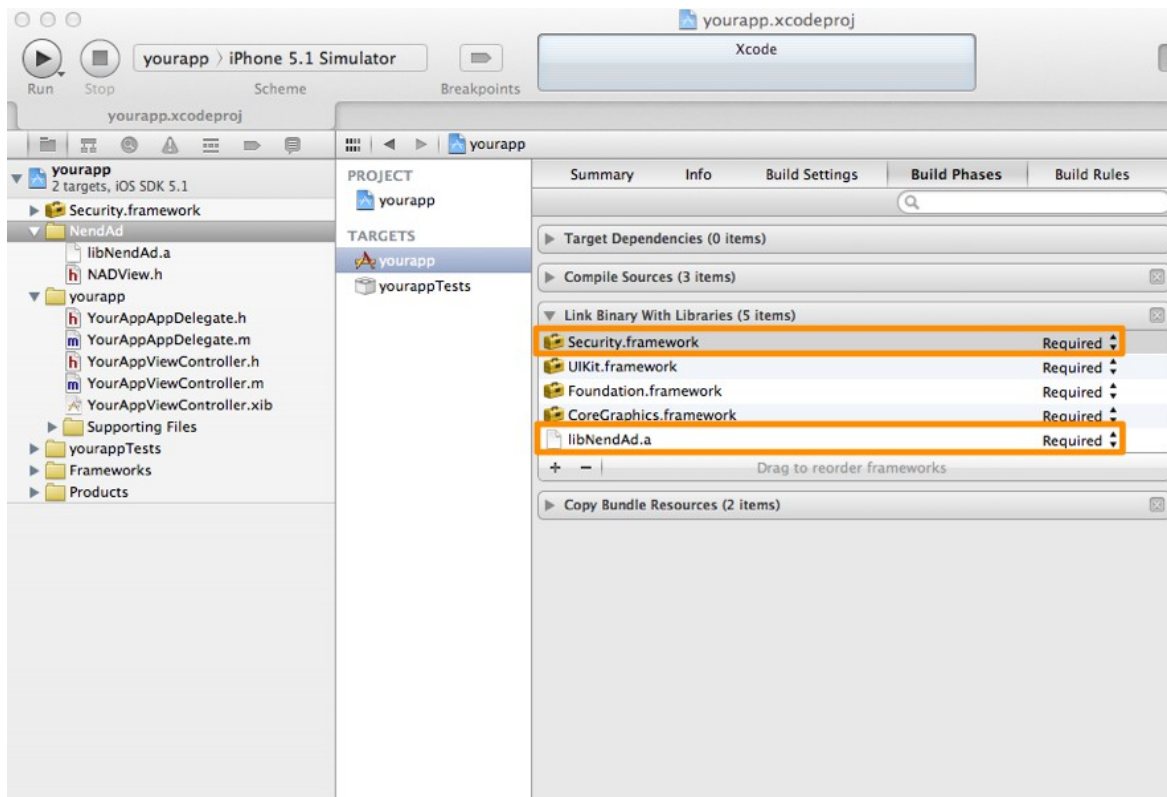
「+」をクリックして、追加できるライブラリの一覧を開き

Security.framework を選択して追加してください。



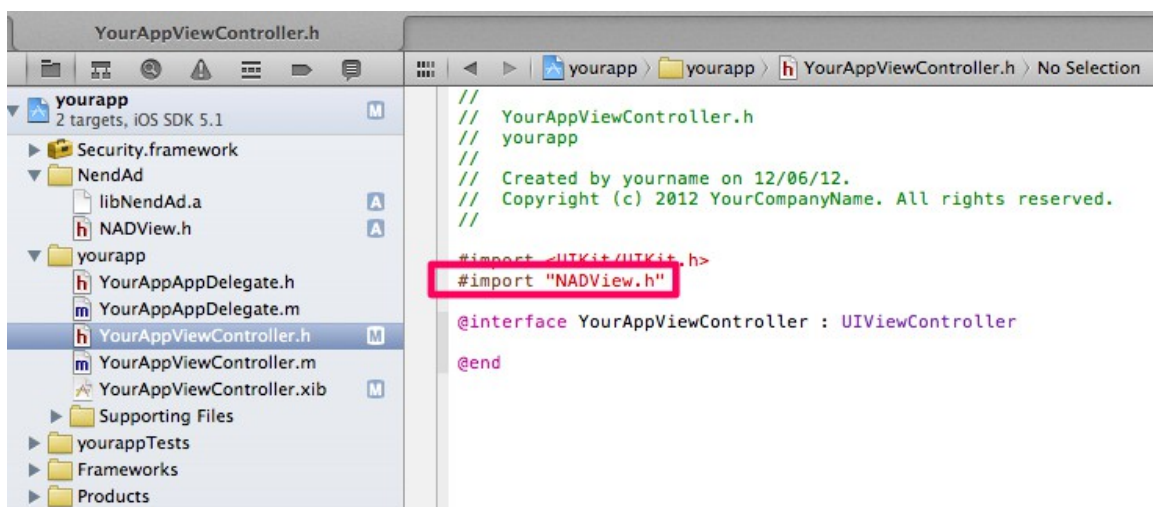
Link Binary With Libraries に2つのリンクが正常に追加されていることを確認します。

- Security.framework
- libNendAd.a



## ○ビルド

任意のヘッダファイルに「`#import "NADView.h"`」を記述し、一旦ビルドします。  
エラーが発生せずにビルドが正常に完了すれば、正常に SDK が追加できています。



※この時点でエラーが発生する場合は、Link Binary With Libraries の内容を再度確認してください。



## ◆NADView の内容

### ○メソッド

**- (void)setNendID:(NSString \*)apiKey spotID:(NSString \*)spotID;**

広告枠の apiKey と spotID をセットします。

**- (void)load;**

ロードを開始します。

**- (void)load:(NSDictionary \*)parameter;**

ロードを開始します。

接続エラーや広告設定受信エラーなどの場合にリトライする間隔を、NSDictionary で任意指定出来ます。ただし key は「retry」、value は 30 - 3600 の間で指定してください。範囲外指定された場合は標準で 60 秒が適用されます。標準で問題ない場合は parameter のない load; を利用してください。

**- (void) pause;**

広告の定期ロード中断を要求します

**- (void) resume;**

広告の定期ロード再開を要求します

### ○プロパティ

**@property (nonatomic, assign) id <NADViewDelegate> delegate;**

delegate オブジェクトの指定

**@property (nonatomic, assign) UIViewController \*rootViewController;**

モーダルビュー表示元のビューコントローラの指定(任意)

### ○Delegate

**- (void) nadViewDidFinishLoad:(NADView \*)adView;**

広告ロードが初めて成功した際に通知されます

**- (void) nadViewDidReceiveAd:(NADView \*)adView;**

広告受信が成功した際に通知されます。(任意)

**- (void) nadViewDidFailToReceiveAd:(NADView \*)adView;**

広告受信に失敗した際に通知されます。(任意)



## ◆実装

ここでは最もシンプルな構造を例にして NADView の実装方法について説明します。それ以外の詳しい実装についてはサンプルソースをご参照ください。

### ○ヘッダファイル

#### (1) delegate 準拠

広告ビューを保持するクラスのヘッダファイルで NADView.h をインポートして NADViewDelegate に準拠します。

例) YourAppViewController.h

```
#import <UIKit/UIKit.h>
#import "NADView.h"

@interface YourAppViewController : UIViewController <NADViewDelegate> {
    NADView *nadView_;
}
@end
```

注意: 複数画面で構成されるアプリケーションで、各 ViewController ごとにインスタンス生成するような実装方法の場合には、画面ごとに必ず後述の (8) 定期ロードの管理を行ってください。

## ○実装ファイル

例) YourAppViewController.m (一部) - NADView 生成からロードまで

```
- (void)viewDidLoad {
    [super viewDidLoad];

    // (2) NADView の作成
    nadView_ = [[NADView alloc] initWithFrame:CGRectMake(0,0,
                                                         NAD_ADVIEW_SIZE_320x50.width,
                                                         NAD_ADVIEW_SIZE_320x50.height)];

    // (3) set apiKey, spotID.
    [nadView_ setNendID:@"a6eca9dd074372c898dd1df549301f277c53f2b9"
                 spotID:@"3172"];
    [nadView_ setDelegate:self]; //(4)
    [nadView_ setRootViewController:self]; //(5)
    [nadView_ load]; //(6)

    [self.view addSubview:nadView_]; // 最初から表示する場合
}
```

### (2)NADView 生成

任意の位置、320x50 サイズの CGRect を引数にインスタンスを生成します。

```
nadView_ = [[NADView alloc] initWithFrame:
             CGRectMake(0,0, NAD_ADVIEW_SIZE_320x50.width, NAD_ADVIEW_SIZE_320x50.height)];
```

### (3)apiKey,spotID のセット

nend 管理画面で発行した、該当アプリの広告枠の apiKey、spotID をセットします。

```
[nadView_ setNendID:@"a6eca9dd074372c898dd1df549301f277c53f2b9" spotID:@"3172"];
```

※この時点で広告設定情報へのアクセスを開始します。

広告枠ステータスが「承認中」の場合、広告のロードをしても受信エラーになります。nend 管理画面で該当アプリの広告枠ステータスが「アクティブ」であることを確認してください。

広告枠の申請は、承認(アクティブ状態に)されると承認メールが登録メールアドレス宛に届きます。  
メール到着の約1時間後には広告配信ができる状態になります。

単に実装方法の確認を行う場合は一時的に表示テスト用 ID (→◆検証) を利用するなどしてください。

#### (4) デリゲートオブジェクトの指定

NADView が広告を受信開始した場合に指定されたデリゲートの `nadViewDidLoad` メソッドを呼んで通知を行います。指定するデリゲートは「`nadViewDidLoad`」メソッドを実装する「NADViewDelegate」プロトコルに準拠させたクラスを指定します。

```
[nadView_ setDelegate:self];
```

#### (5) RootViewController の指定

アプリ内ブラウザを生成する際の親のビューを保持するビューコントローラを指定します。

```
[nadView_ setRootViewController:self];
```

※広告クリック時にアプリ内ブラウザを利用するかどうかは配信される広告の種類によります。  
指定されていない場合は、標準のブラウザアプリケーションが起動します。

#### (6) 広告のロード

NADView の `load` メソッドで広告のロードを開始します。

```
[nadView_ load];
```

リトライ間隔を標準値以外で指定したい場合には、以下のメソッドを利用してロードを開始します。

// 例) 問い合わせエラー時には 60 分間隔で再問い合わせする

```
[nadView_ load:[NSDictionary dictionaryWithObjectsAndKeys:@"3600", @"retry", nil]];
```

## (7) Delegate 通知

### ロード完了

広告のロードが完了すると(4)で指定したデリゲートの `nadViewDidLoad` メソッドに通知されます。  
ロードが完了してから `NADView` を表示したい場合はここで行うことができます。

```
-(void)nadViewDidLoad:(NADView *)adView {  
    NSLog(@"delegate nadViewDidLoad:");  
}
```

### 【任意】 広告受信通知

広告の受信に成功した場合通知されます。  
広告を受信するたびに任意の処理を行いたい場合に利用します。

```
-(void)nadViewDidReceiveAd:(NADView *)adView {  
    NSLog(@"delegate nadViewDidReceiveAd:");  
}
```

### 【任意】 広告受信エラー通知

広告の受信に失敗した場合に通知されます。  
通信エラー、広告在庫がなくなった場合など、何らかの理由で広告を表示できない場合に通知します。  
エラー時に広告を表示しないようにするなどの処理が必要な場合に利用します。

```
-(void)nadViewDidFailToReceiveAd:(NADView *)adView {  
    NSLog(@"delegate nadViewDidFailToLoad:");  
}
```

## (8) 定期ロードの管理

広告のロードを開始した後に画面内に表示しないケースがある場合には、必ず以下の処理を行ってください。

これには、

- ・ 複数画面で構成された画面遷移が発生するアプリケーションにおいて各画面で個別に広告のインスタンスを生成している場合
  - ・ 複数広告の切り替え処理（広告スイッチング SDK の利用含む）を行う場合
- などが該当します。

### 定期ロードの中断

広告を画面内に表示しない、もしくは画面遷移等で View 自体が表示されない場合には pause メッセージを送信、広告の定期ロードを中断します。

例) 画面が隠れたら定期ロードを中断

```
- (void)viewWillDisappear:(BOOL)animated {  
    [nadView_ pause];  
}
```

### 定期ロードの再開

広告を再び画面内に表示、または画面遷移等で View 自体を表示する場合には resume メッセージを送信、広告の定期ロードを再開します。

例) 画面が表示されたら定期ロードを再開

```
- (void)viewWillAppear:(BOOL)animated {  
    [nadView_ resume];  
}
```

## (9) リリース

dealloc 時には、必ず **delegate** プロパティに **nil** をセットしてからリリースを行うようにしてください。

### ARC の場合

```
- (void) dealloc {  
    [nadView_ setDelegate:nil]; // delegate に nil をセット  
    nadView_ = nil;  
}
```

### 非 ARC の場合

```
- (void) dealloc {  
    [nadView_ setDelegate:nil]; // delegate に nil をセット  
    [nadView_ release];  
    nadView_ = nil;  
}
```

### 【重要】

**delegate** プロパティに **nil** をセットしない場合、メモリの解放が適切に行われず、稀に予期しないクラッシュを引き起こす場合があります。必ず **delegate** に **nil** をセットしてから **release** を行うようにしてください。

また、**delegate** に **nil** がセットされた際、自動的に広告受信ローテーションを中断(pause)します。  
これにより **release** 前の **pause**メッセージ送信が不要になります。

以上で設定完了となります。

## ◆検証

iOS アプリ向け表示テスト用の apiKey と spotID を設定していただくことで対象アプリケーション用の広告枠のステータスが「未承認」である場合でも広告表示の確認ができます。

### iOS アプリ向け表示テスト用 ID

apiKey: a6eca9dd074372c898dd1df549301f277c53f2b9

spotID: 3172

※確認後は必ず各アプリケーション用広告枠の apiKey と spotID に書き換えてください。

※テスト用 ID のままアプリケーションをストアへ申請、配布された場合、広告効果は正しく集計されませんのでご注意ください。また、テスト用 ID のまま申請してしまった場合、nend 側での保障等はできかねますのでご了承ください。

広告受信エラーの最も簡単な再現方法はオフラインにすることです。現状、nendSDK 内でオンライン状況のチェックは行っておりませんので、オフライン時に何らかの処理を行う場合は、アプリケーション側で実装する必要があります。



## ◆よくある質問

### ■nendSDK の導入について

Q: 広告が表示されません (受信エラーだけが通知される)

A: 広告受信エラーの際に NSLog を出力していますのでエラー内容を確認してください。

#### 主なエラー内容

[nendSDK Error] IE410: Invalid parameter.

無効な apiKey と spotID を渡している可能性があります。

nend 管理画面で発行された apiKey と spotID を正しくセットしているかを確認してください。

[nendSDK Error] IE400: Failed to get ad-spot.

[nendSDK Error] IE420: Unapproved, 'apiKey' or 'spotID' is wrong.

未承認、もしくは apiKey、spotID が正しくない。特に、広告枠のステータスが「承認中」の場合は広告を表示できません。承認された際にはご登録メールアドレス宛に広告枠承認のお知らせが届きますが、「アクティブ」に変わってから広告配信が反映されるまでに 1 時間ほどかかる場合があります。

[nendSDK Error] IE430: Ad is out of stock.

広告枠のステータスもアクティブですが、広告枠の設定に対して配信可能な広告がない可能性があります。

### ■nendSDK の機能について

Q: 広告が配信されなかったことをプログラム内で検出する方法はありますか？

A: 接続エラーや広告の在庫切れなど、次の広告を表示できない場合、広告受信エラーを通知します。このエラーは広告配信されないたび、もしくはリトライが失敗するたびに通知されます。

→詳しくは「◆実装 (7)delegate」をご覧ください。

Q: iPad アプリでも SDK を利用できますか？

A: 動作保証環境内であれば利用はできますが、表示サイズは iPhone 同様 320\*50 になります。

## お問合せ先

nend - お問合せフォーム

<https://www.nend.net/inquiries/form/>

または

株式会社ファンコミュニケーションズ ADN 事業部  
[media@nend.net](mailto:media@nend.net)

※お問い合わせ時には、メディア登録名、apikey/spotID、SDK バージョン番号  
必要に応じてご利用中の開発環境詳細などを情報としてお寄せ頂きますとよりスムーズです。