

なりすまし判定技術 iOSアプリ

SDK組み込みマニュアル



株式会社スワローインキュベート

2022年08月23日

■はじめに

- ・なりすまし判定技術 iOSアプリは、株式会社スワローインキュベートが提供しています。
- ・本書は、予告なしに変更されることがあります。
- ・本書を無断で、複製、転用、公衆送信、貸与等を行わないようお願いします。
- ・本サンプルアプリをご利用いただくには、あらかじめ当社利用規約に同意いただく必要があります。
詳しくは営業担当までお問い合わせください。

■ご注意事項

- ・SDKをご購入いただいていない場合は、本アプリのソースコードはすべて当社に帰属します。
- ・当社は本プログラムに関する保証は一切行っておりません。お客様の責任にてご使用ください。
本プログラム使用による問題や損害が生じた場合にも当社は何ら責任を負いませんので、あらかじめ
ご了承ください。

お問い合わせ

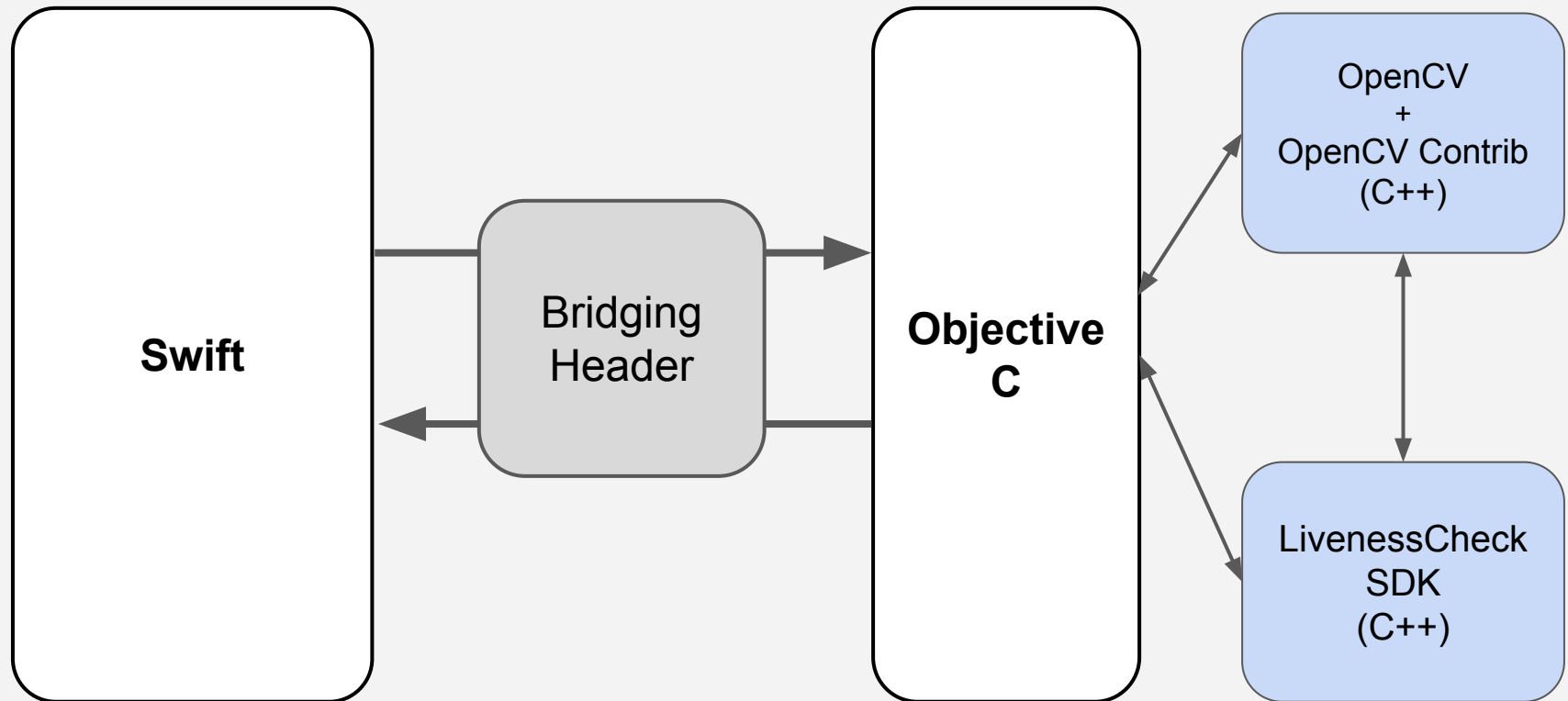
株式会社スワローインキュベート

なりすまし判定技術 テクニカルサポート窓口

TEL: 029-886-9912 MAIL: support@swallow-incubate.com

■なりすまし判定SDK組み込み時の構成

当社のFramework形式で提供されるなりすまし判定 SDKは、C++インターフェースとなっています。
SDKの各インターフェースを呼び出すためには、Swift言語のプログラムから Bridging-Headerを介して Objective-Cファイルとの連携を行った上で、Objective-C言語から呼び出していただく必要があります。
また本SDKはOpenCVにも依存しているため、当社より提供予定の OpenCV Frameworkも同様の形式で呼び出していただきます。詳しくはサンプルアプリのファイル構成及びソースコードを参照してください。



同梱内容

■同梱内容

以下のファイルが提供時の主要なファイルとなります。



Xcodeプロジェクトファイル
livenesscheck-ios-sample

なりすまし判定
iOSサンプルアプリ



Frameworkファイル
LivenessCheck.framework

なりすまし判定ライブラリ



Frameworkファイル
opencv2.framework

OpenCVライブラリ



期限付きトークン
(試用版)

アクティベーションツール



ご利用マニュアル
(本書)

事前準備

■ご利用環境

現在のバージョンでは、以下のご利用環境を推奨しています。

項目		内容
開発環境	OS環境	macOS 最新版を推奨
	プログラム言語	Swift / Objective-C / C++11
	開発環境	Xcode 最新版を推奨
	依存ライブラリ	LivenessCheck ライブラリ OpenCV ライブラリ 4.5.5 (一部OpenCV_Contribモジュール入り)
動作環境	OS	iOS 14以降推奨
	動作マシン	iPhone 8以降を推奨
	カメラ	フロントカメラを想定

■事前準備

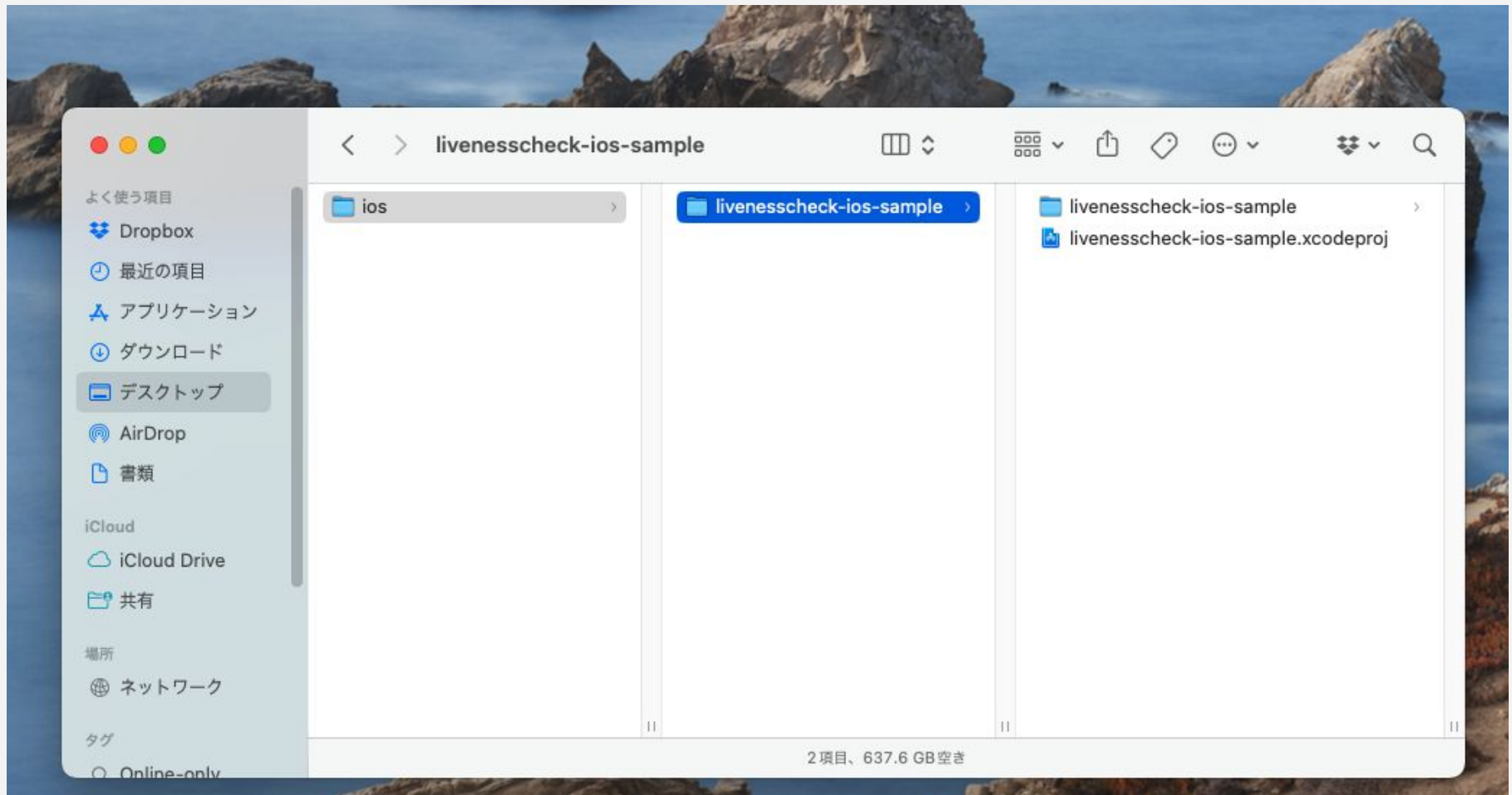
お使いのmacOSマシン上に、Xcodeアプリをあらかじめインストールしておいてください。すでにインストール済みの方は、最新版にアップデートしておくことを推奨します。



SDK組み込み手順

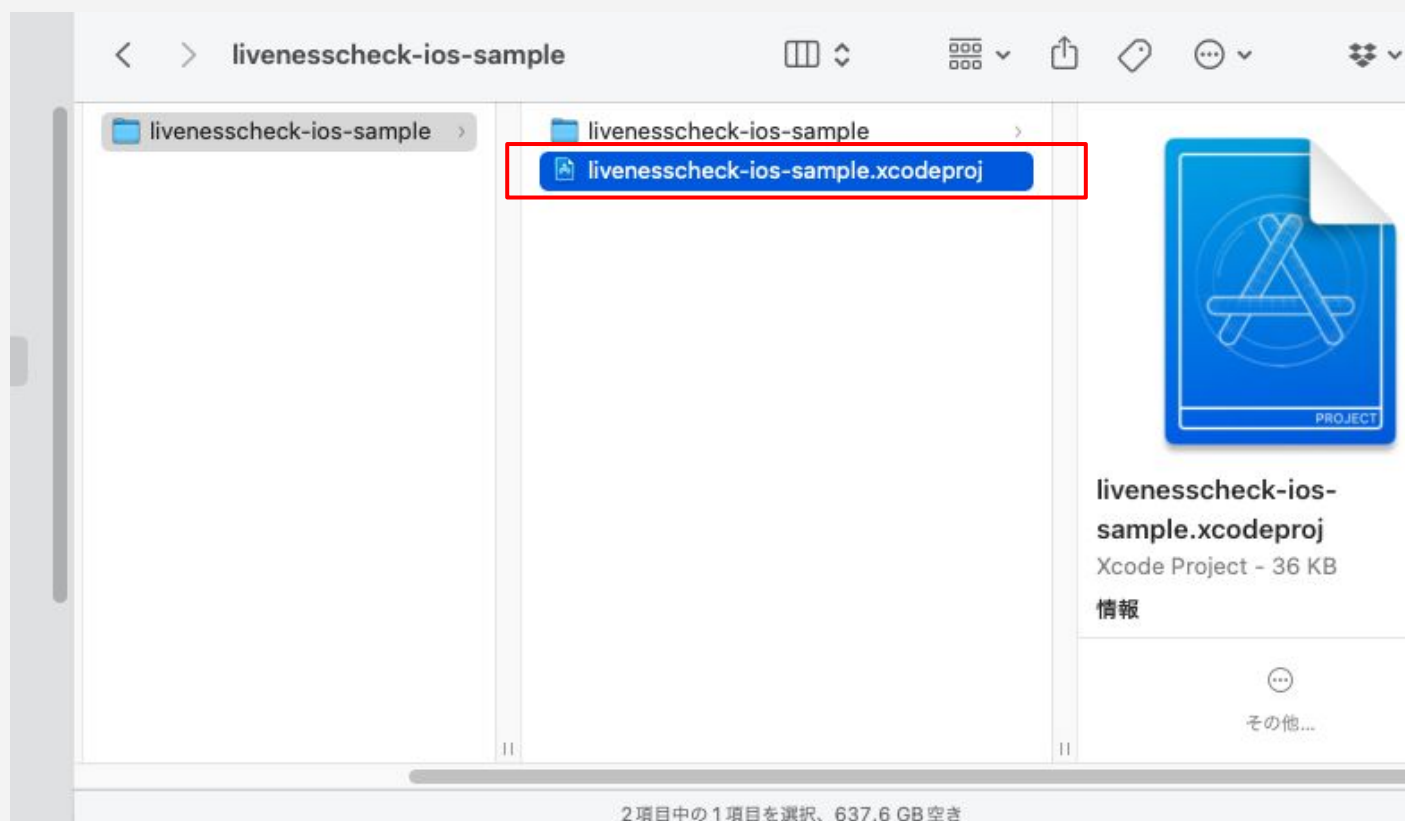
■ご利用手順①

ご提供したiOSサンプルアプリフォルダを任意の場所に設置してください。
本マニュアルでは、デスクトップのOSディレクトリに置いたものとして説明をします。



■ご利用手順②

なりすまし判定iOSサンプルアプリのディレクトリを開くと、Xcode用のプロジェクトファイルと、ソースコードやStoryboard、学習モデルファイル等の入った「livenesscheck-ios-sample」ディレクトリが同梱されています。お使いのマシンでXcodeが使える状態になっていることを確認の上、xcodprojファイルを開いてください。



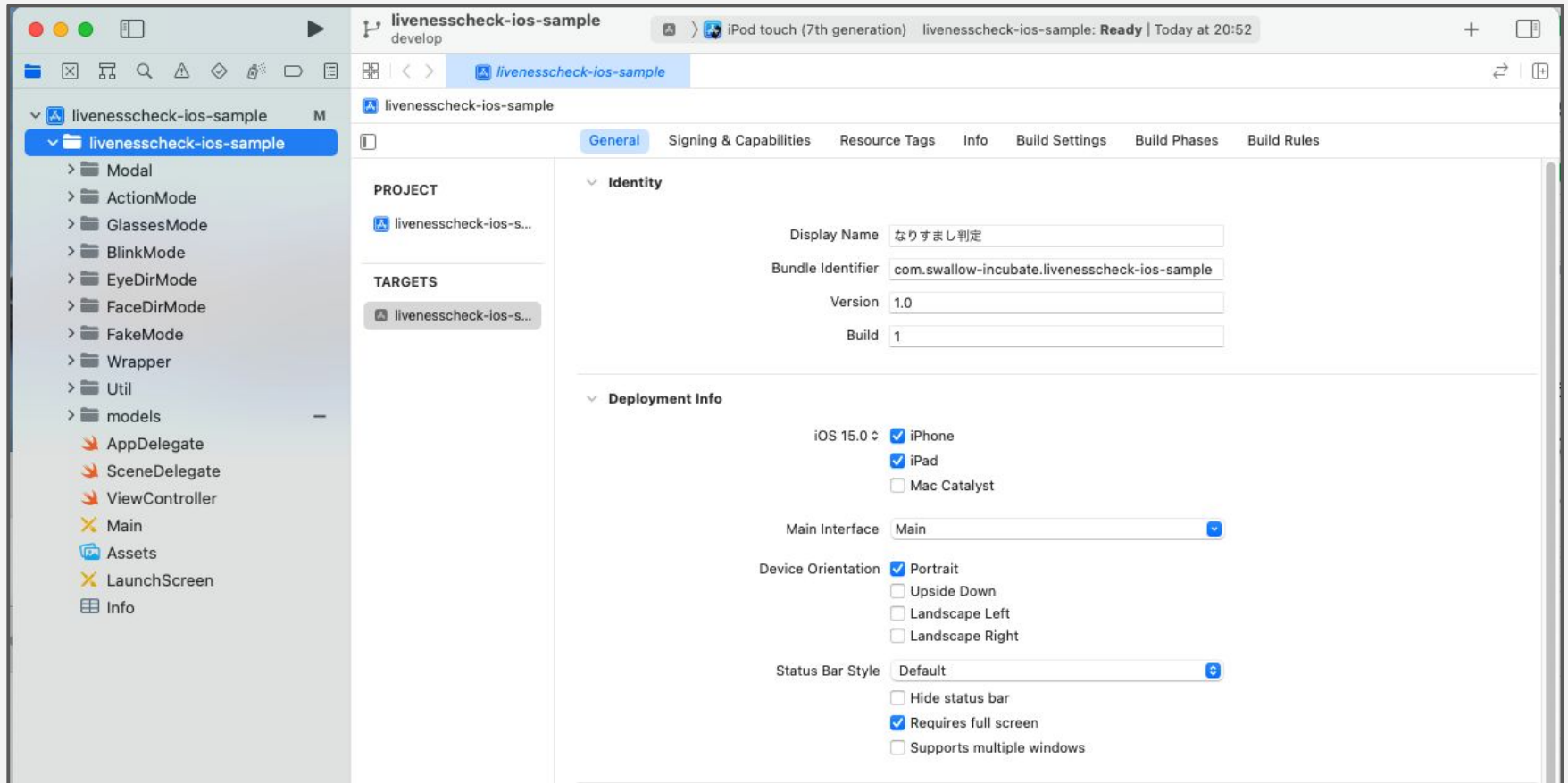
■ご利用手順②(補足)

サンプルアプリに含まれるディレクトリ構成およびファイルは以下の通りとなります。
(バージョンによって多少異なる場合があります)

第一階層	第二階層	説明
livenesscheck-ios-sample.xcodeproj		Xcodeで起動するプロジェクトファイルです。
livenesscheck-ios-sample/	AppDelegate.swift SceneDelegate.swift Assets.xcassets Info.plist	初期で付属しているファイルです。 特にこちらは初期状態から編集されていません。
	Base.lproj/	Main.storyboardとLaunchScreen.storyboardが格納されています
	ViewController.swift	Main.storyboardに対応するコントローラです
	ActionMode/	アクション判定機能に関するコントローラとStoryboardです
	BlinkMode/	まばたき判定機能に関するコントローラとStoryboardです
	EyeDirMode/	目向き判定機能に関するコントローラとStoryboardです
	FaceDirMode/	顔向き判定機能に関するコントローラとStoryboardです
	FakeMode/	フェイク判定機能に関するコントローラとStoryboardです
	GlassesMode/	メガネ装着判定機能に関するコントローラとStoryboardです
	Util/	カメラ起動及びSDK操作、トークン読み込みのためのSwiftファイルです
	Wrapper/	Bridging-HeaderおよびObjective-Cのラッパーファイルです
	models/	SDKに必要な学習モデルファイルが格納されています

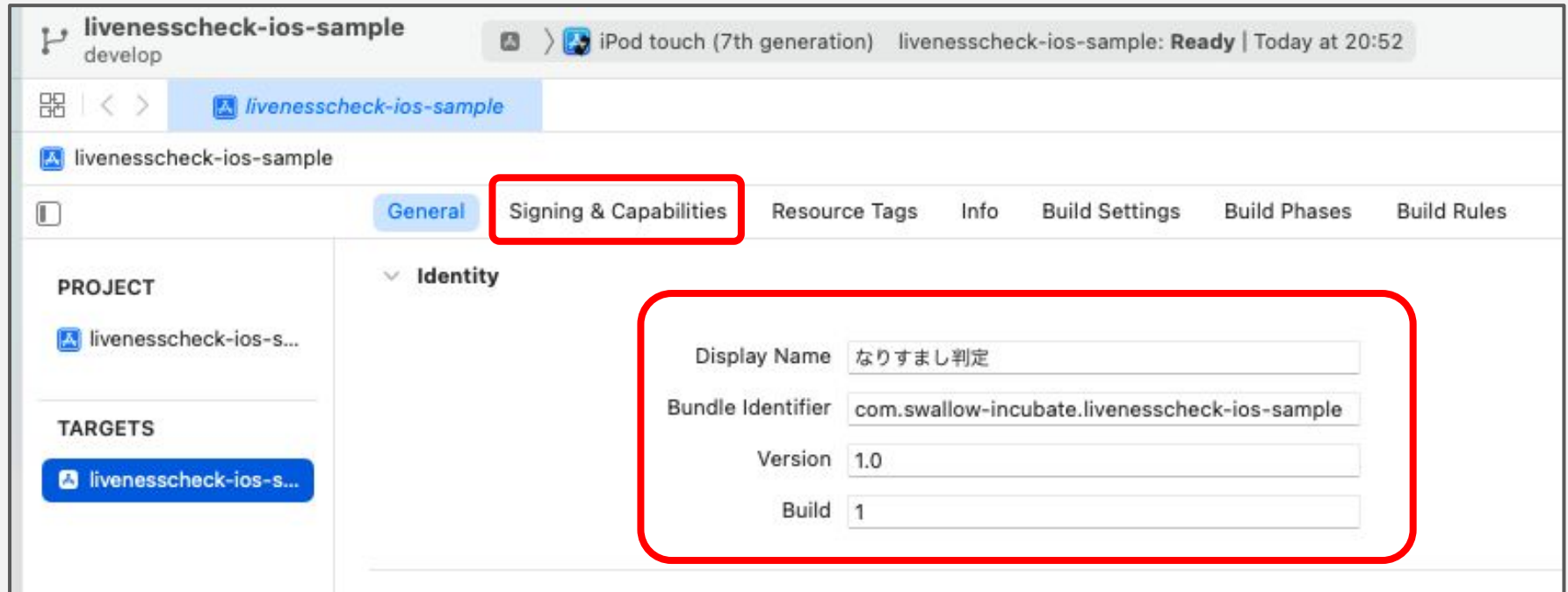
■ご利用手順③

Xcodeでなりすまし判定iOSサンプルアプリを開くと、以下のようなウィンドウが立ち上がります。



■ご利用手順③(補足)

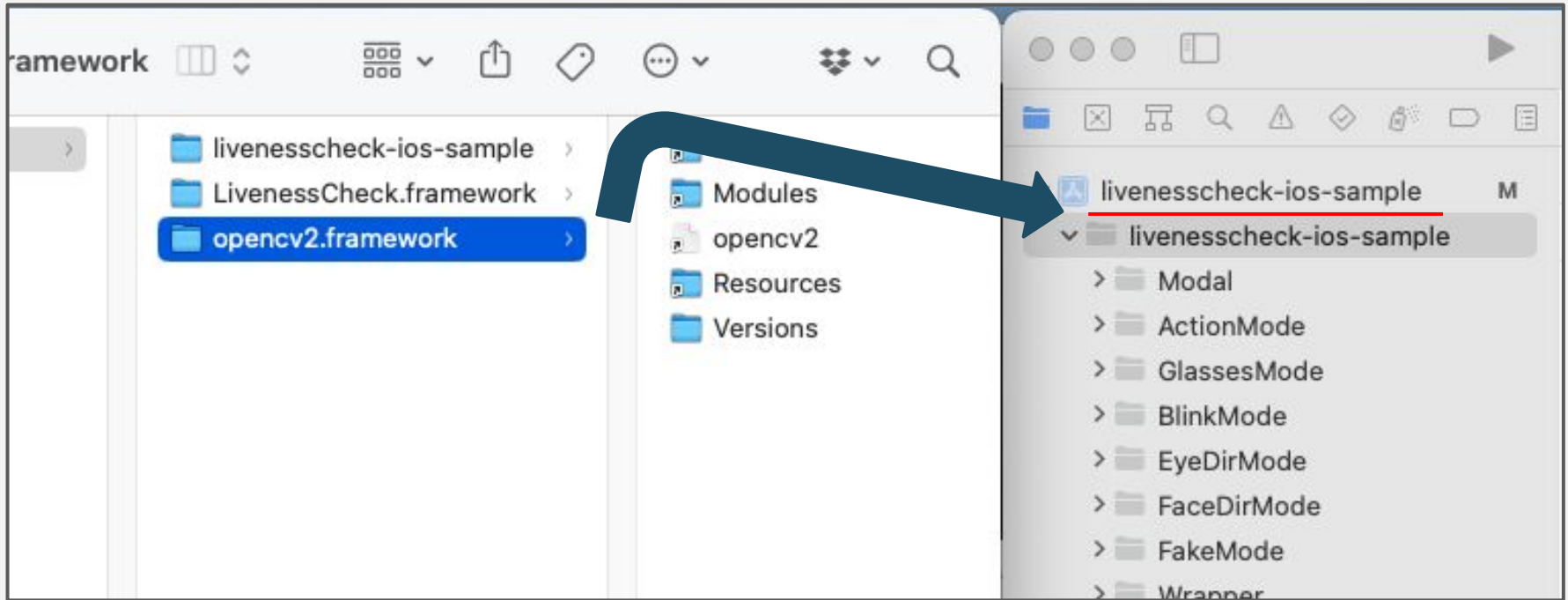
Display NameやBundle Identifierなどは初期値が入っていますが、適宜修正していただいても構いません。Signingについても同様にご自身のものに設定しておいてください。



OpenCVライブラリの導入

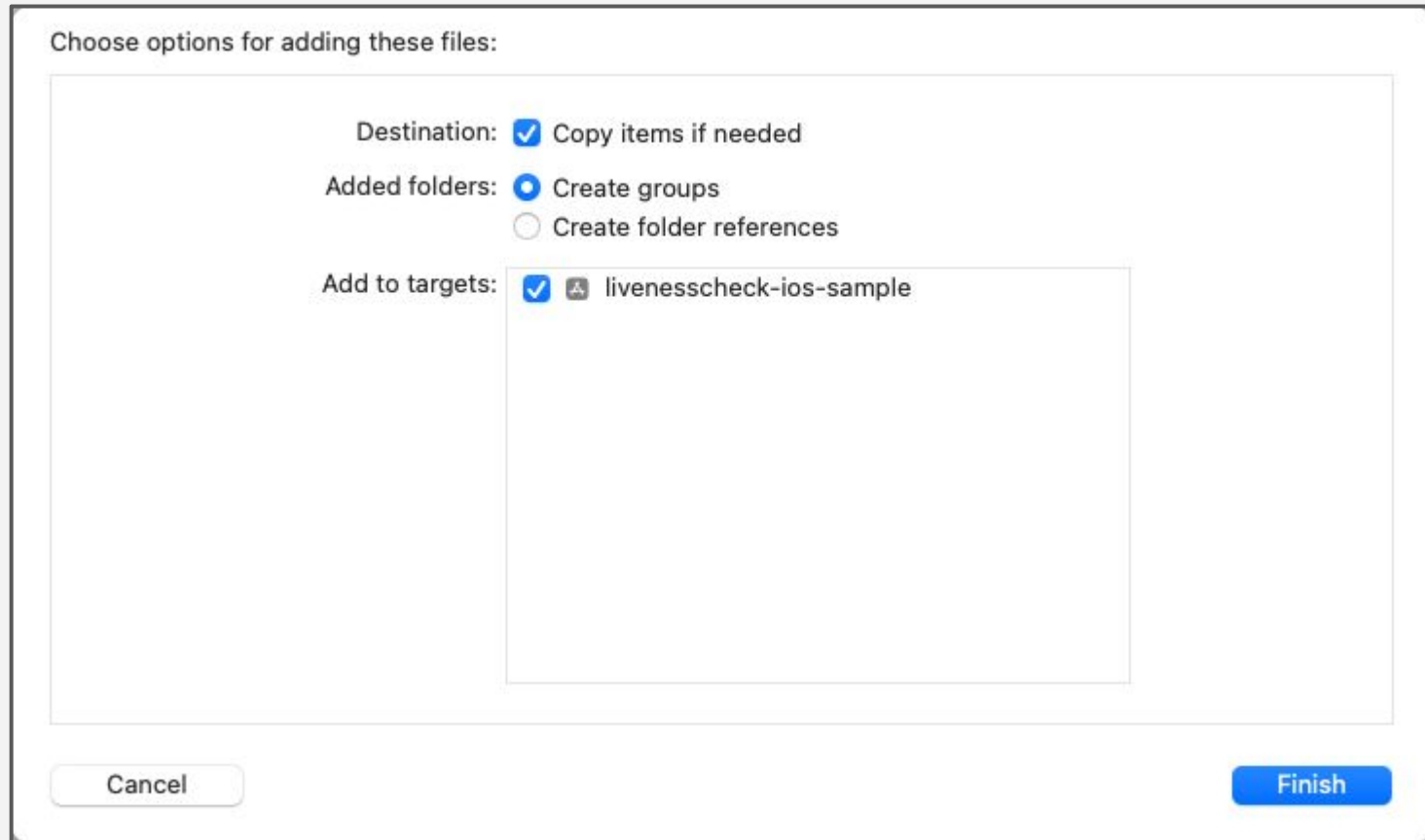
■OpenCVライブラリの導入①

まずはOpenCVライブラリをプロジェクトに導入していきます。当社より提供されたframework形式のopencv2.frameworkを用意し、ディレクトリごとlivenesscheck-ios-sampleプロジェクト直下(赤線の部分)にドラッグ&ドロップしてください。



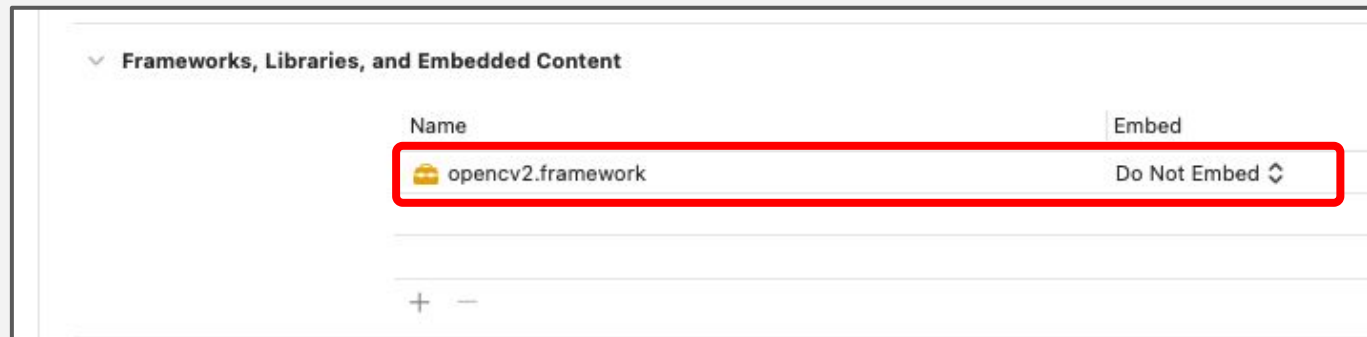
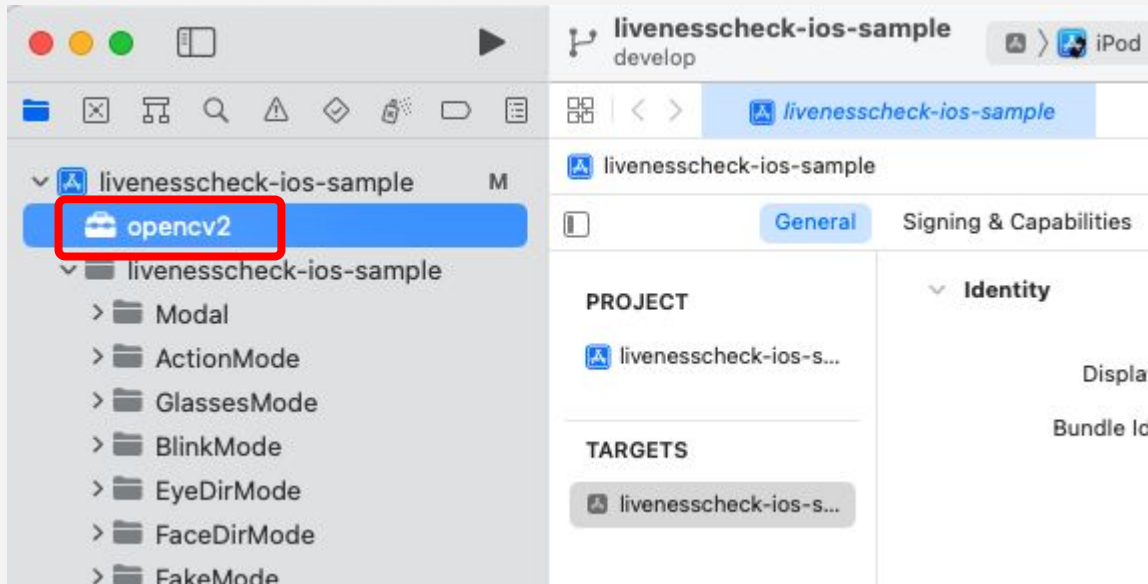
■OpenCVライブラリの導入②

ドラッグ&ドロップをすると以下のような画面が表示されますので、targetsにチェックを入れ、その他の項目も適宜選択したあと「Finish」を押します。



■OpenCVライブラリの導入③

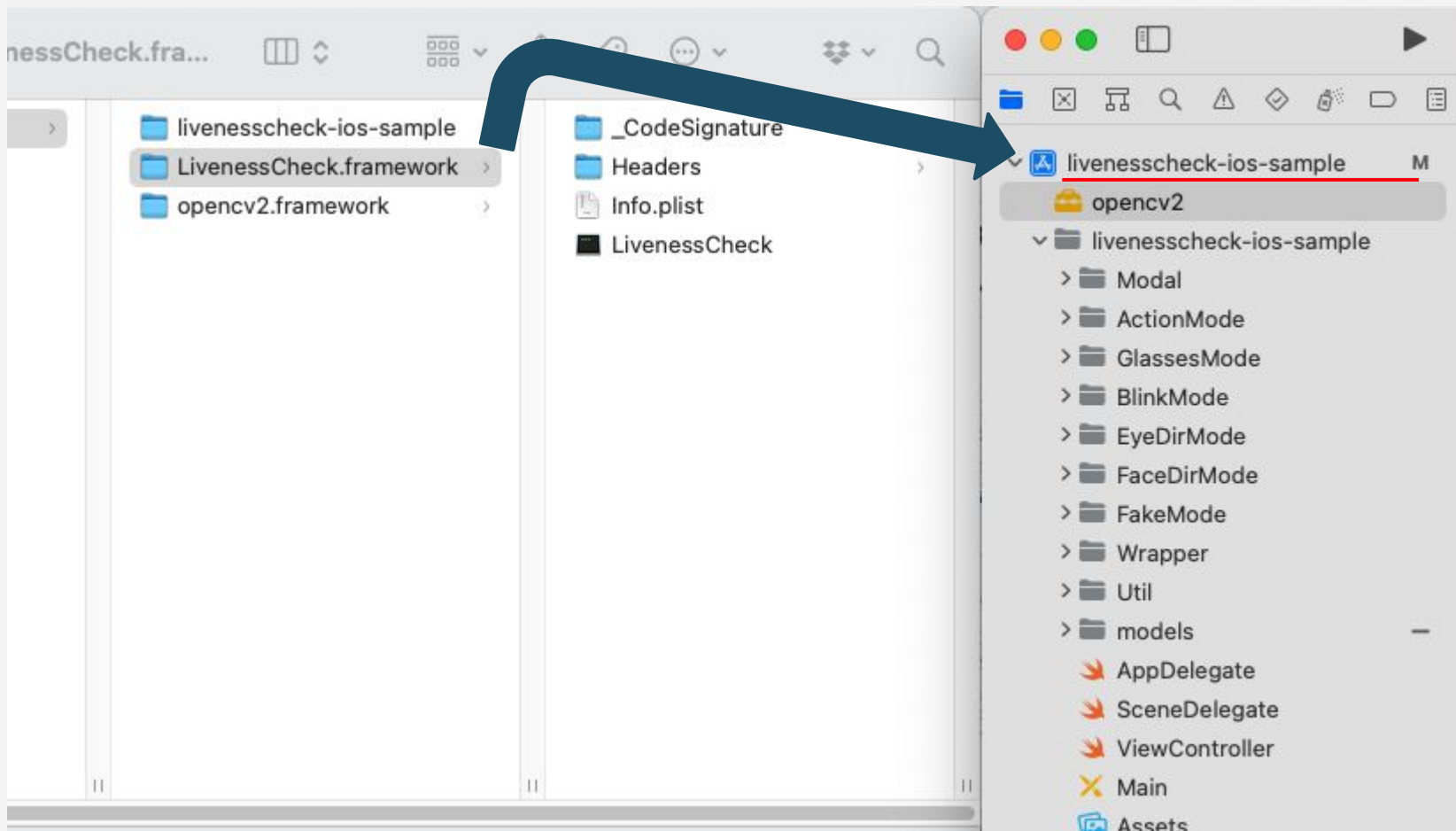
以下の通りに、livenesscheck-ios-sampleプロジェクト直下に「opencv2」が表示されていればOpenCVライブラリの導入は完了です。TARGETSのGeneralタブで表示される「Frameworks, Libraries, and Embedded Content」欄にも表示されていることを確認してください。



なりすまし判定ライブラリの導入

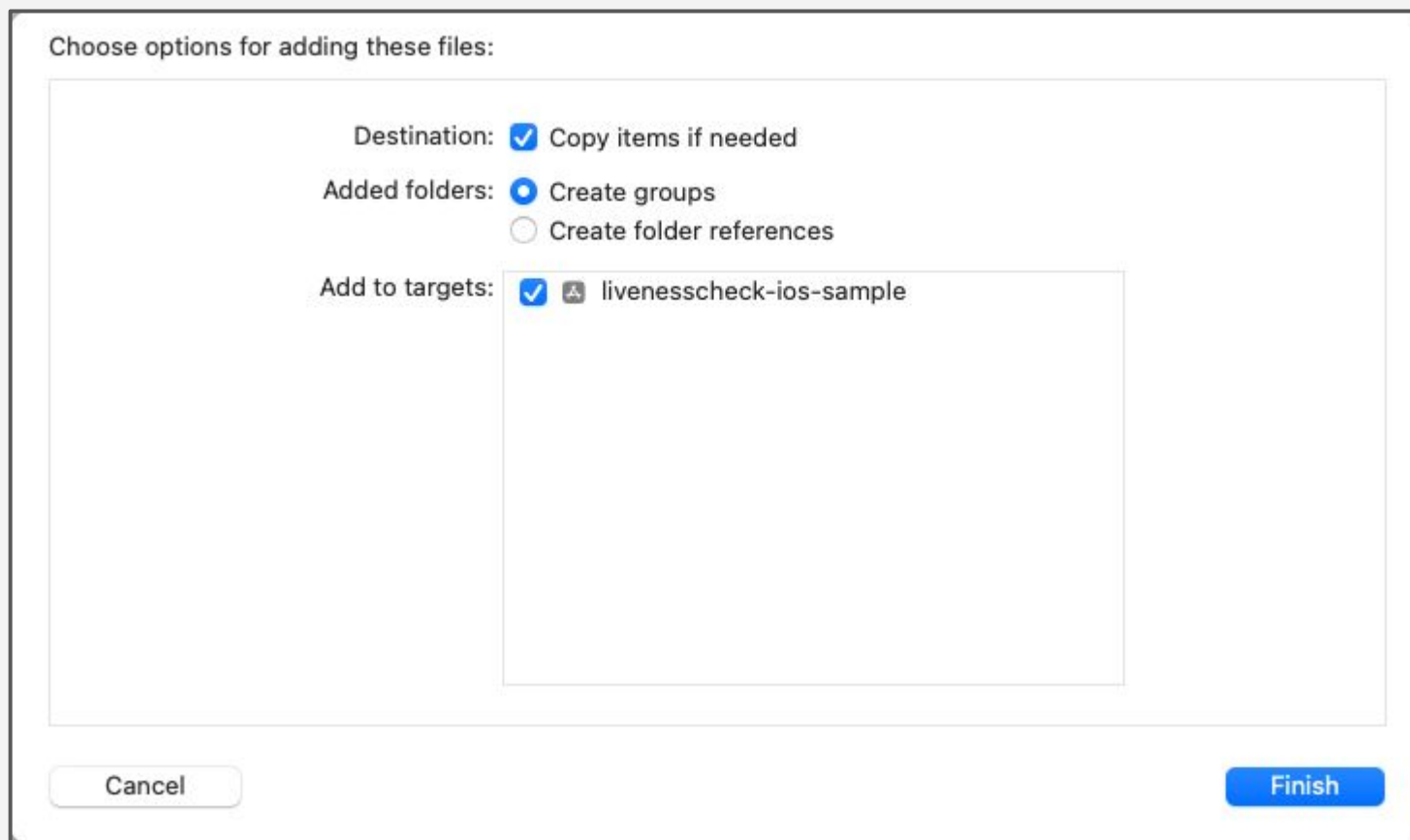
■なりすまし判定ライブラリの導入①

次になりすまし判定ライブラ리를プロジェクトに導入していきます。当社より提供されたframework形式のLivenessCheck.frameworkを用意し、ディレクトリごとlivenesscheck-ios-sampleプロジェクト直下（赤線の部分）にドラッグ&ドロップしてください。



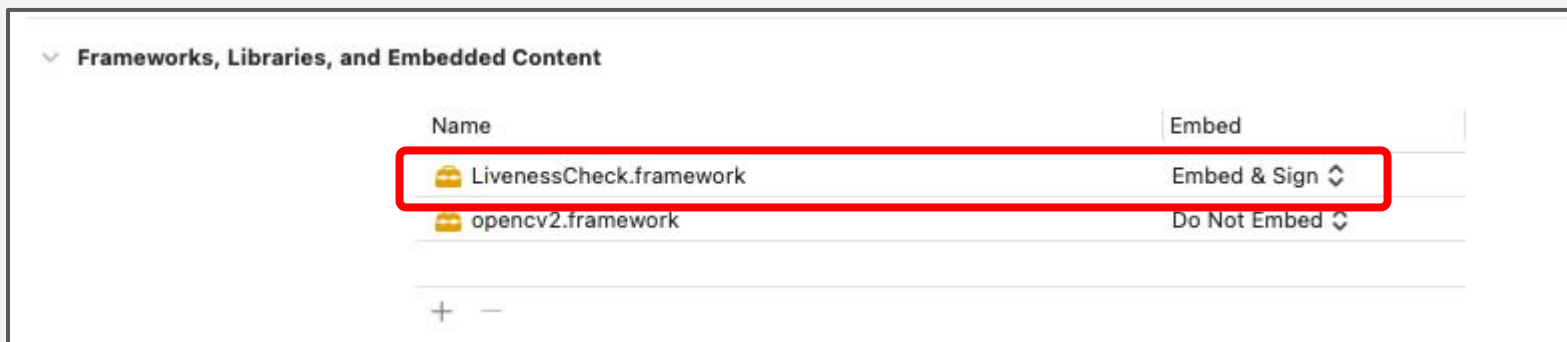
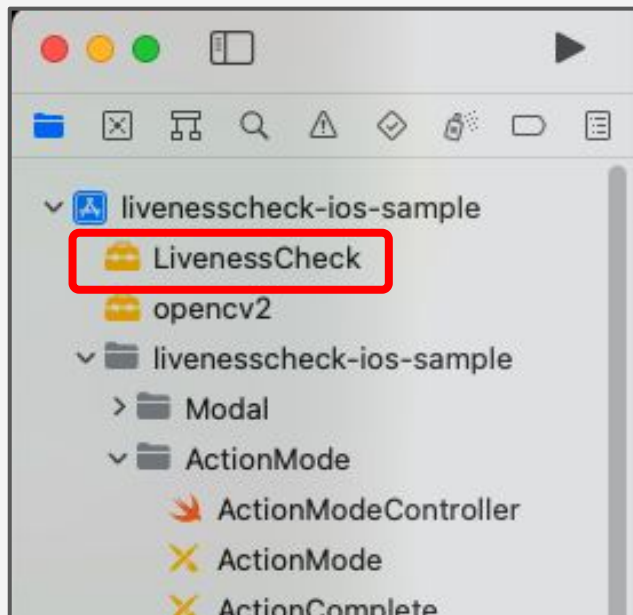
■なりすまし判定ライブラリの導入②

ドラッグ&ドロップをすると以下のような画面が表示されますので、targetsにチェックを入れ、その他の項目も適宜選択したあと「Finish」を押します。



■なりすまし判定ライブラリの導入③

以下の通りに、livenesscheck-ios-sampleプロジェクト直下に「LivenessCheck」が表示されていればなりすまし判定ライブラリの導入は完了です。TARGETSのGeneralタブから表示される「Frameworks, Libraries, and Embedded Content」欄にも表示されていることを確認してください。なお、なりすまし判定ライブラリは「Embed」欄を「Embed & Sign」に変更しておきます。

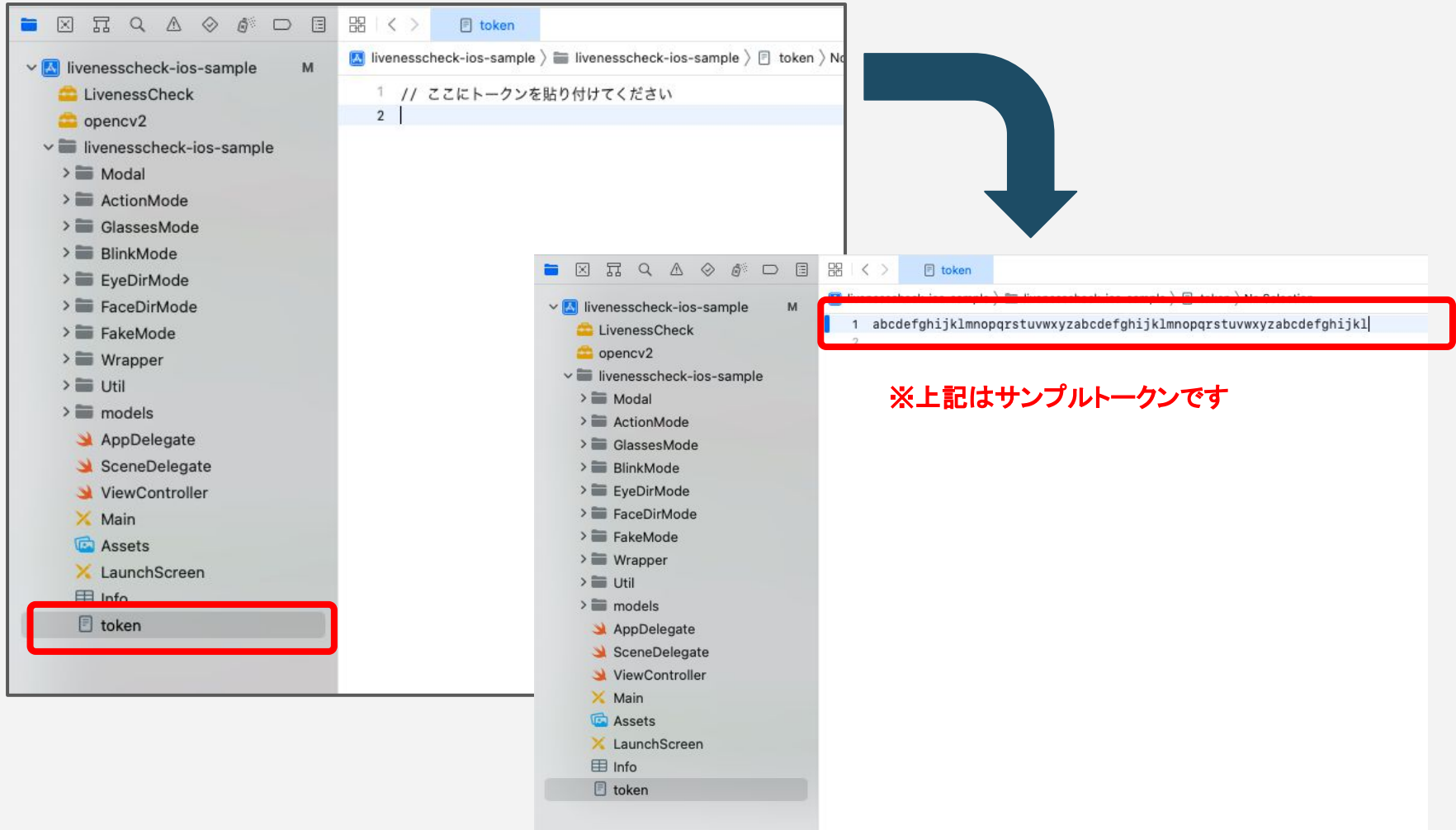


アクティベーショントークンの導入

■アクティベーショントークンの導入

なりすまし判定ライブラリを動作させるにはアクティベーショントークンが必要となります。

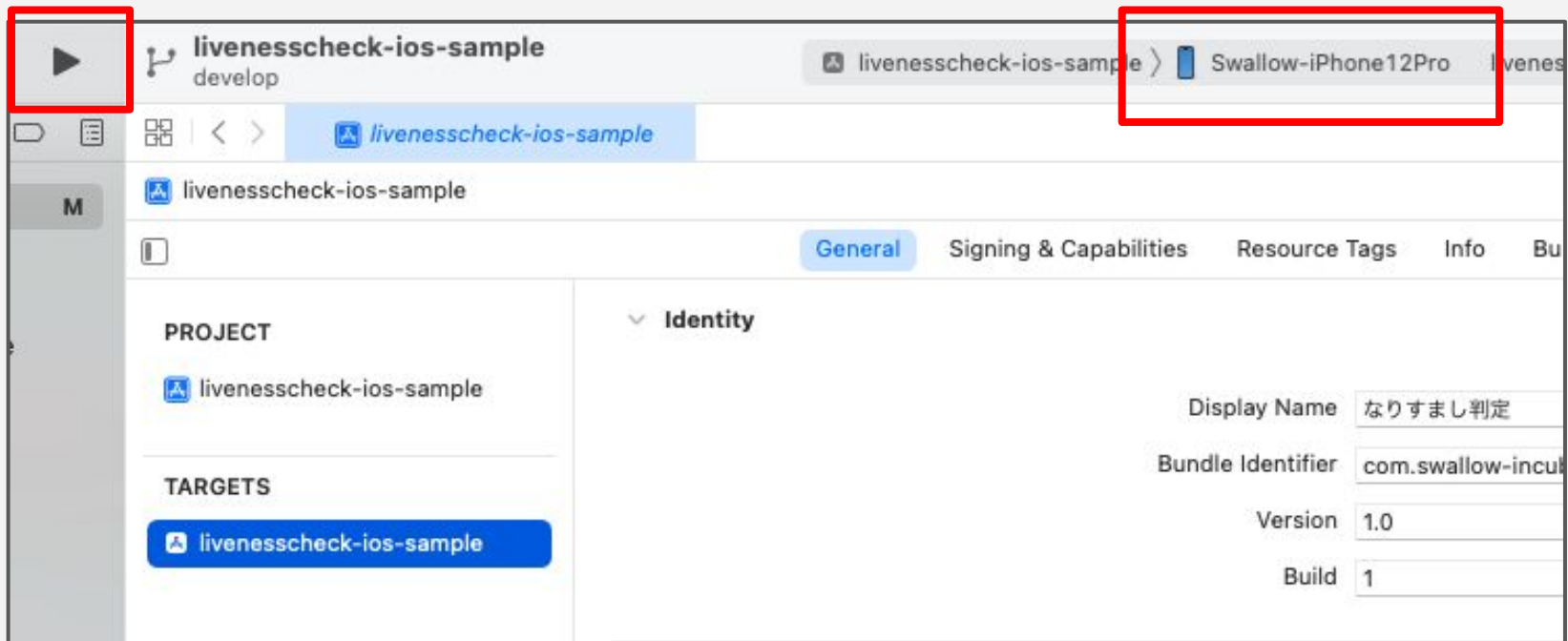
LivenessCheck.framework内には予めトークン情報が埋め込まれており、任意のトークン文字列とマッチさせることで所定の有効期限まで動作します。当社より提供されるtoken.txtの中身をサンプルアプリにプリセットされているtoken.txtファイルにコピーします。



アプリのビルド

■アプリのビルド

ここまでの作業を完了したら、現在サンプルアプリを開いているmacOSマシンに、アプリを設置するiOS端末を接続し、Xcode上で認識されていることを確認して、選択しておきます。
これでアプリをビルドする準備が整いましたので「▶」マークを押してビルドを行います。



■アプリのビルド

アプリをビルドしたiOS端末に、アイコン画像なしの「なりすまし判定サンプル」と表示されたアプリアイコンが表示されていればビルドは完了です。アプリアイコンをタップしてアプリを起動すると、以下のようなアプリ画面が立ち上がります。動作を確認しながら、サンプルアプリのソースコードを参照してください。

