



アセットのコンパイルについて

ojun

集まれSwift好き！Swift愛好会 vol.95

2025/09/08 @株式会社スマートバンク

自己紹介 (ojun)



- name: ojun
- 2023年新卒入社
- iOSエンジニア

X @ojun_9

Motivation

Motivation

ほとんどのプロジェクトでアセットが利用されている

アセットの管理にはアセットカタログ（`.xcassets`）を利用していることが多い

一方で、アセットがどのような形で、最終的にアプリで表示できているかというのはあまり知られていない

今日話すこと

今日話すこと

1. 前提: xcassets は ipa に含まれていない
2. actool とは
3. actool はプロジェクト外でも使える
4. Assets.car の中身を見る
5. xcassets がなくても Assets.car があれば画像を表示できる

前提: xcassets は ipa に含まれて
いない

前提: xcassets は ipa に含まれていない

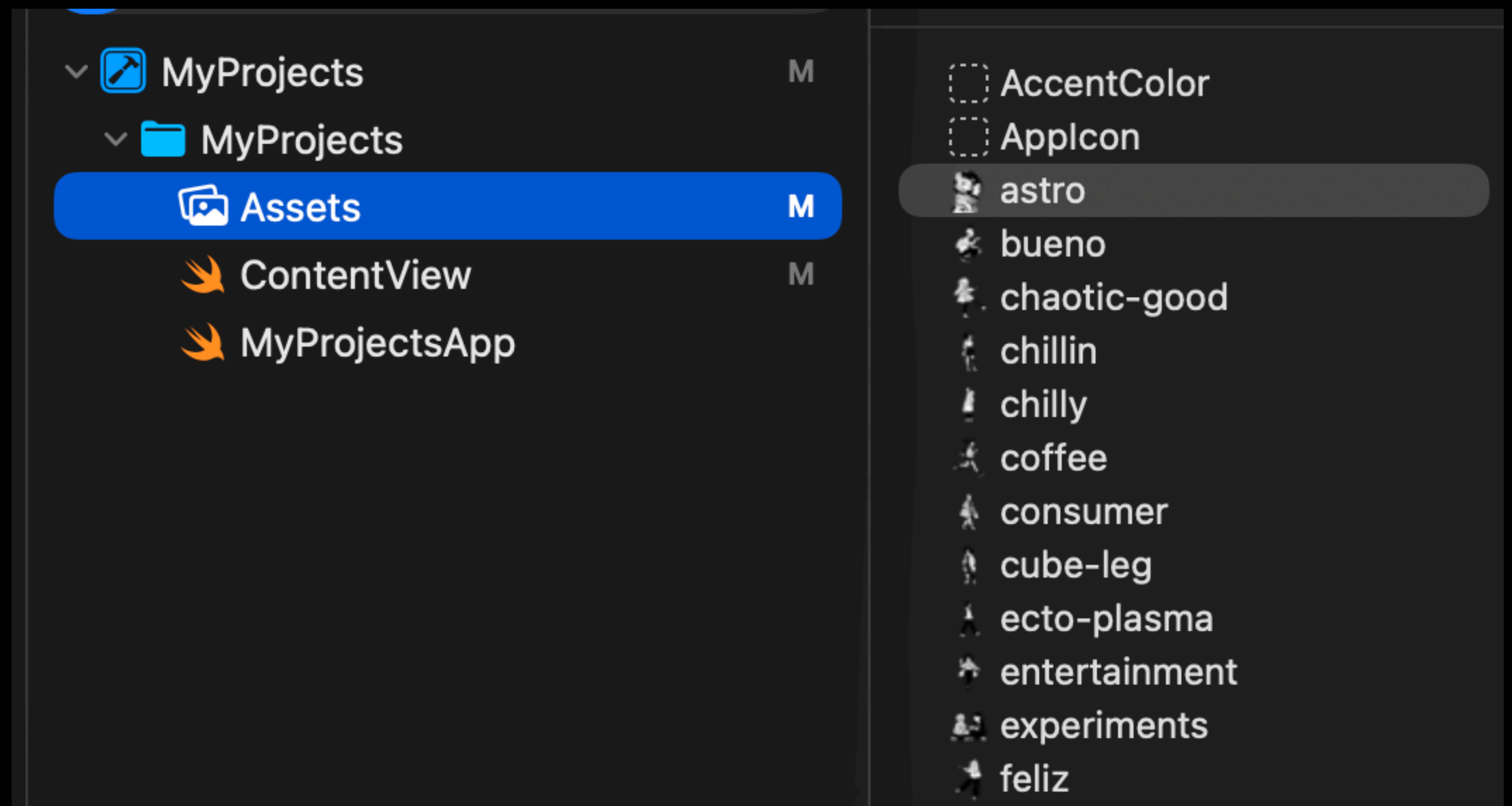
この前提を確認するために、最小の prj を作る

前提: xcassets は ipa に含まれていない

この前提を確認するために、最小の prj を作る

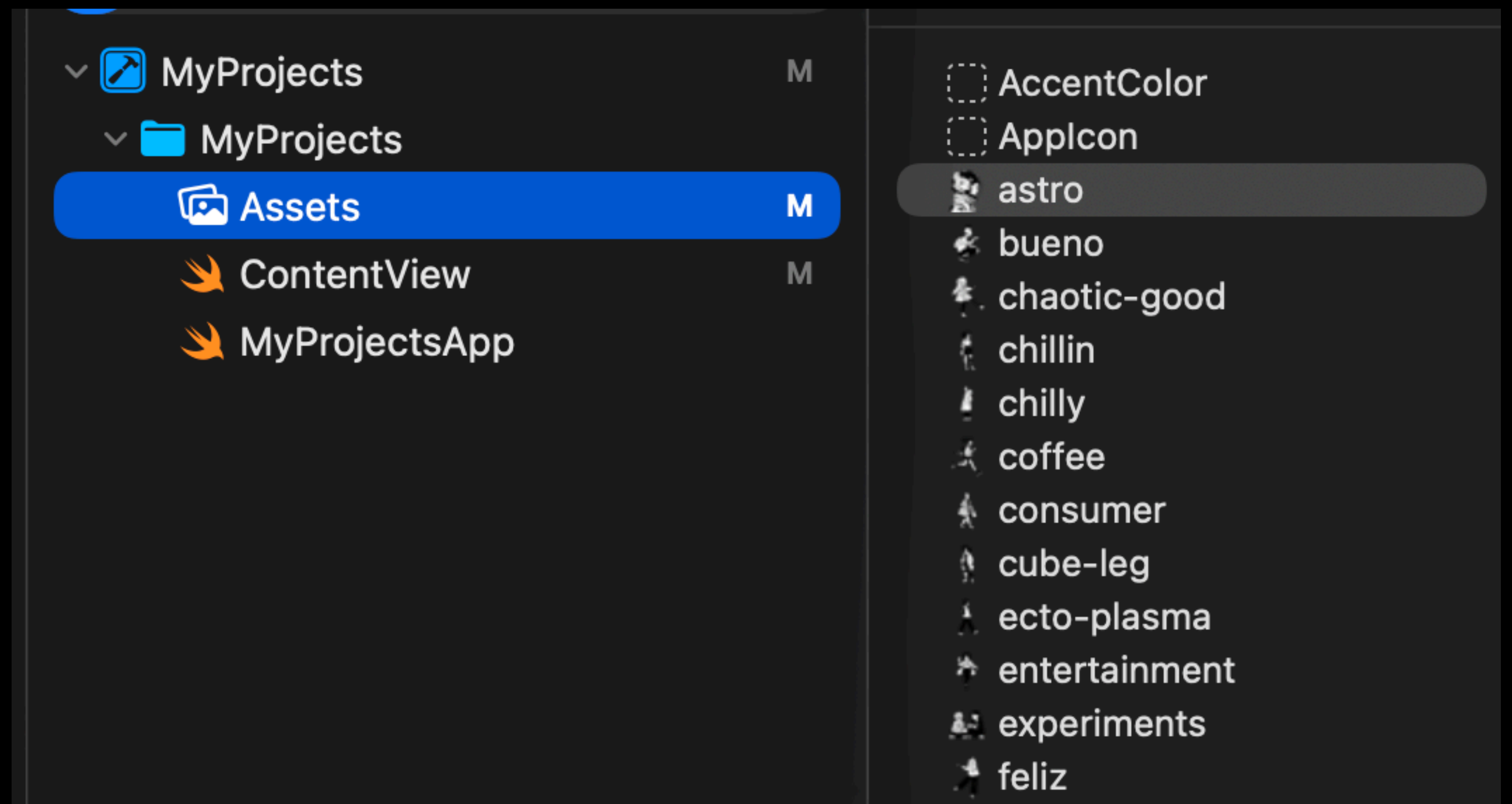
前提: xcassets は ipa に含まれていない

プロジェクトを作成し、アセットを複数追加した



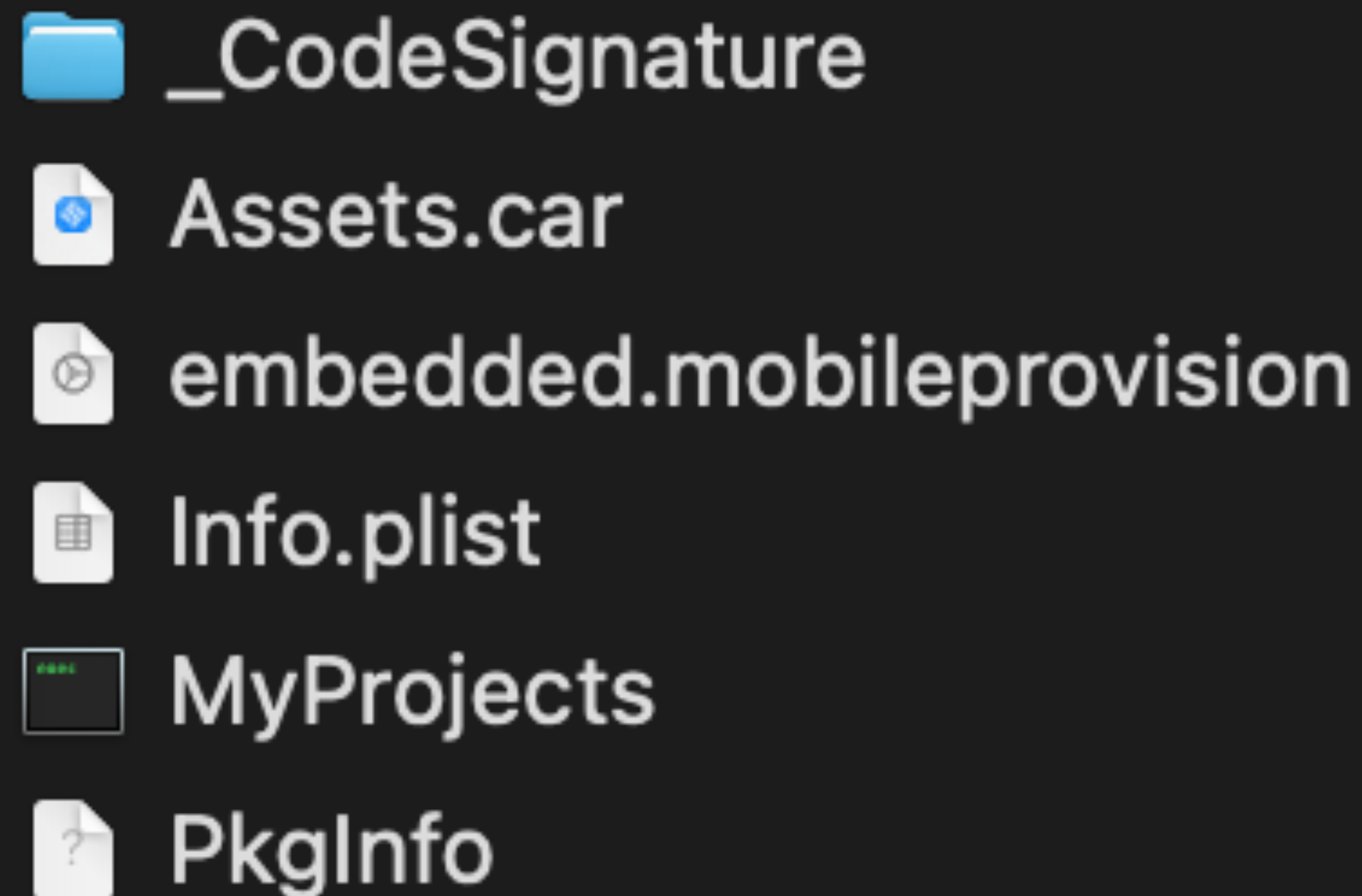
前提: xcassets は ipa に含まれていない

この状態でプロジェクトをアーカイブし ipa ファイルを作成する



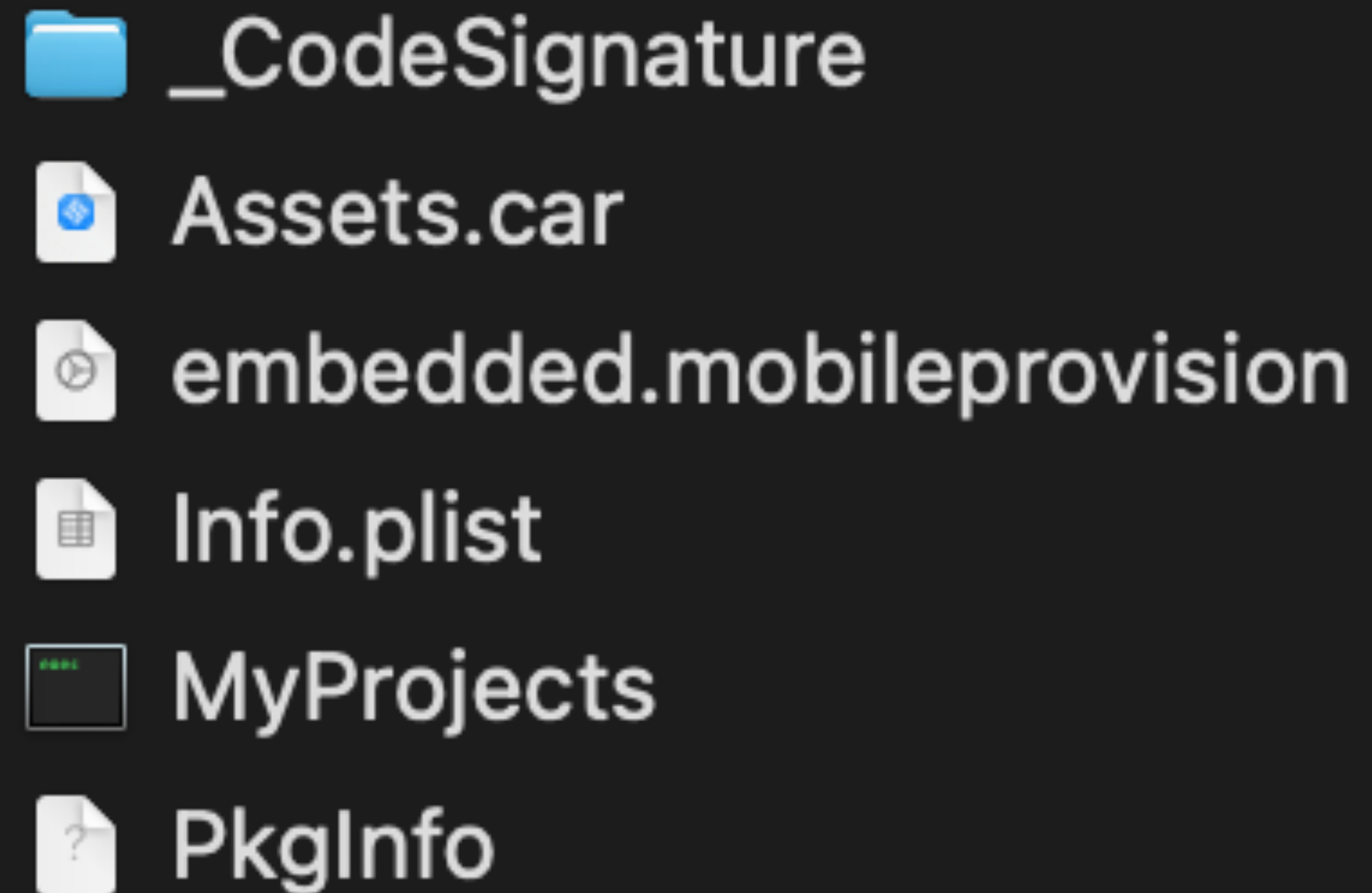
前提: xcassets は ipa に含まれていない

ipa ファイルの中身は右の通り

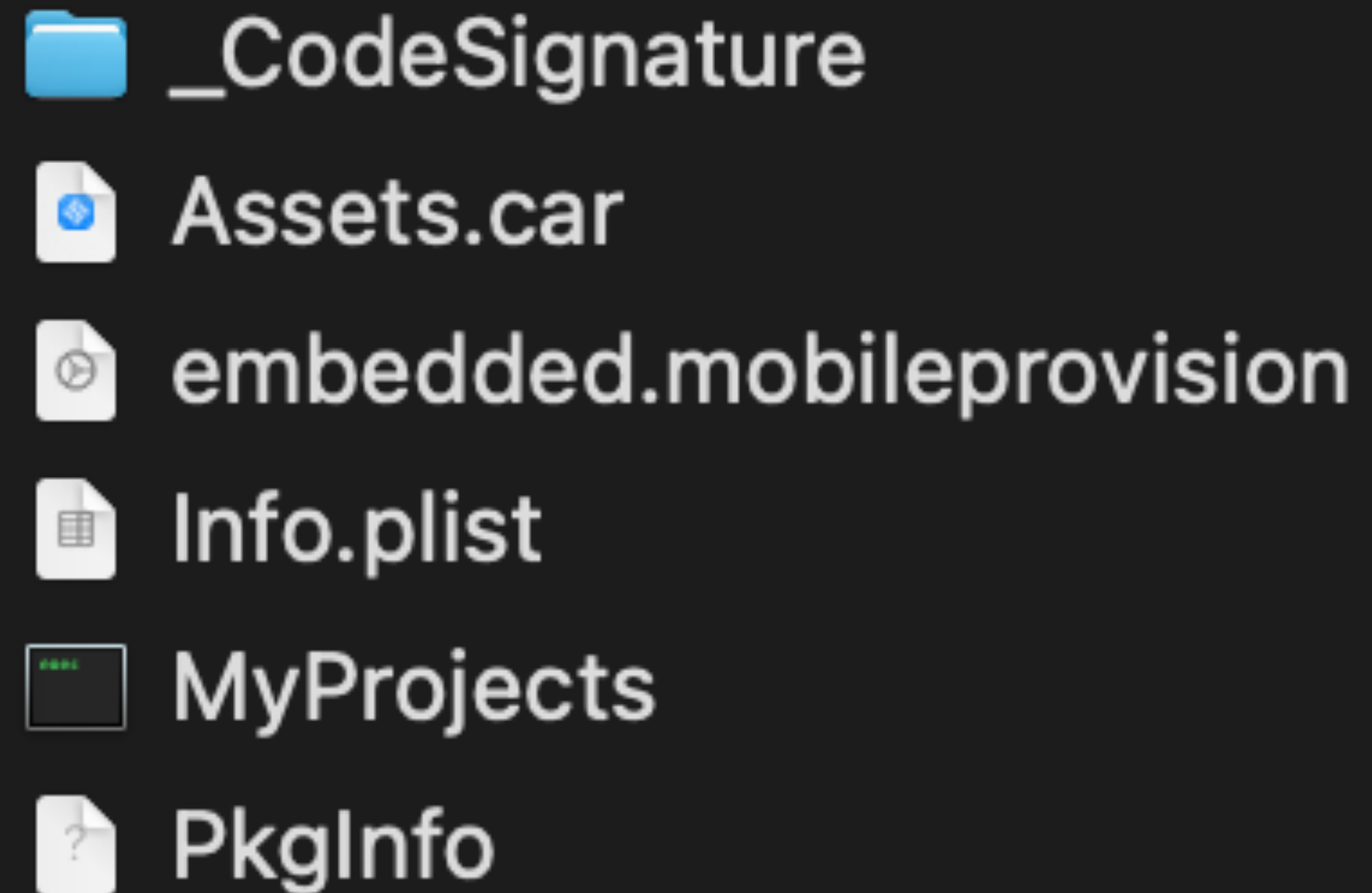
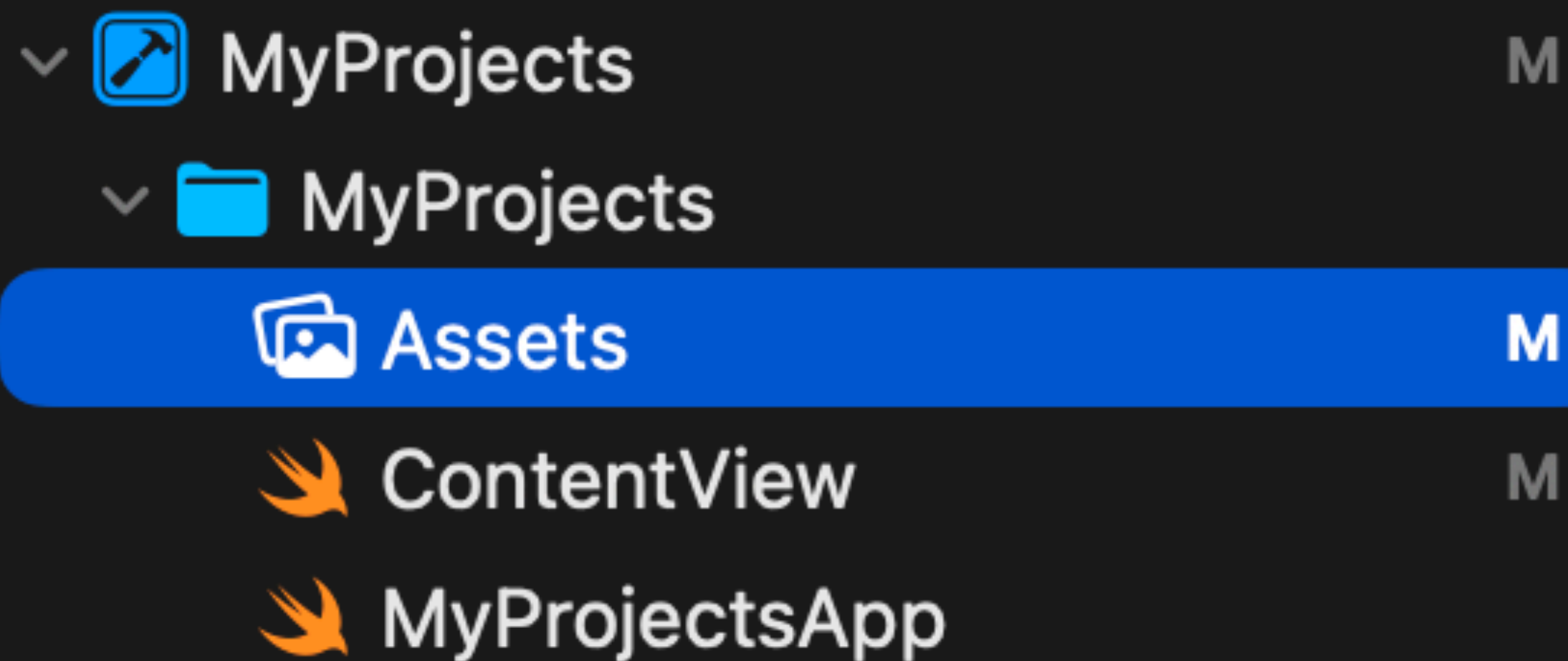


前提: xcassets は ipa に含まれていない

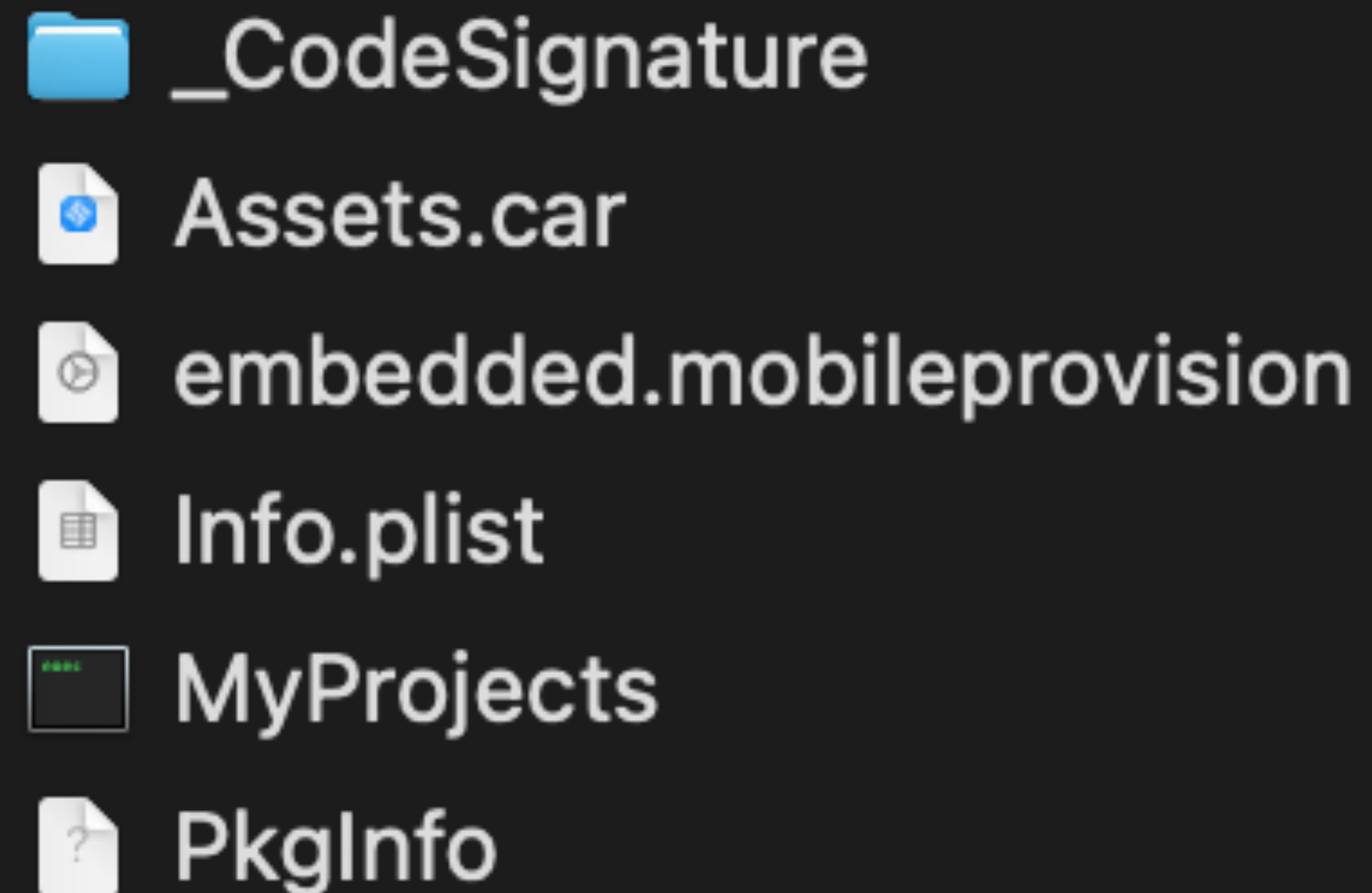
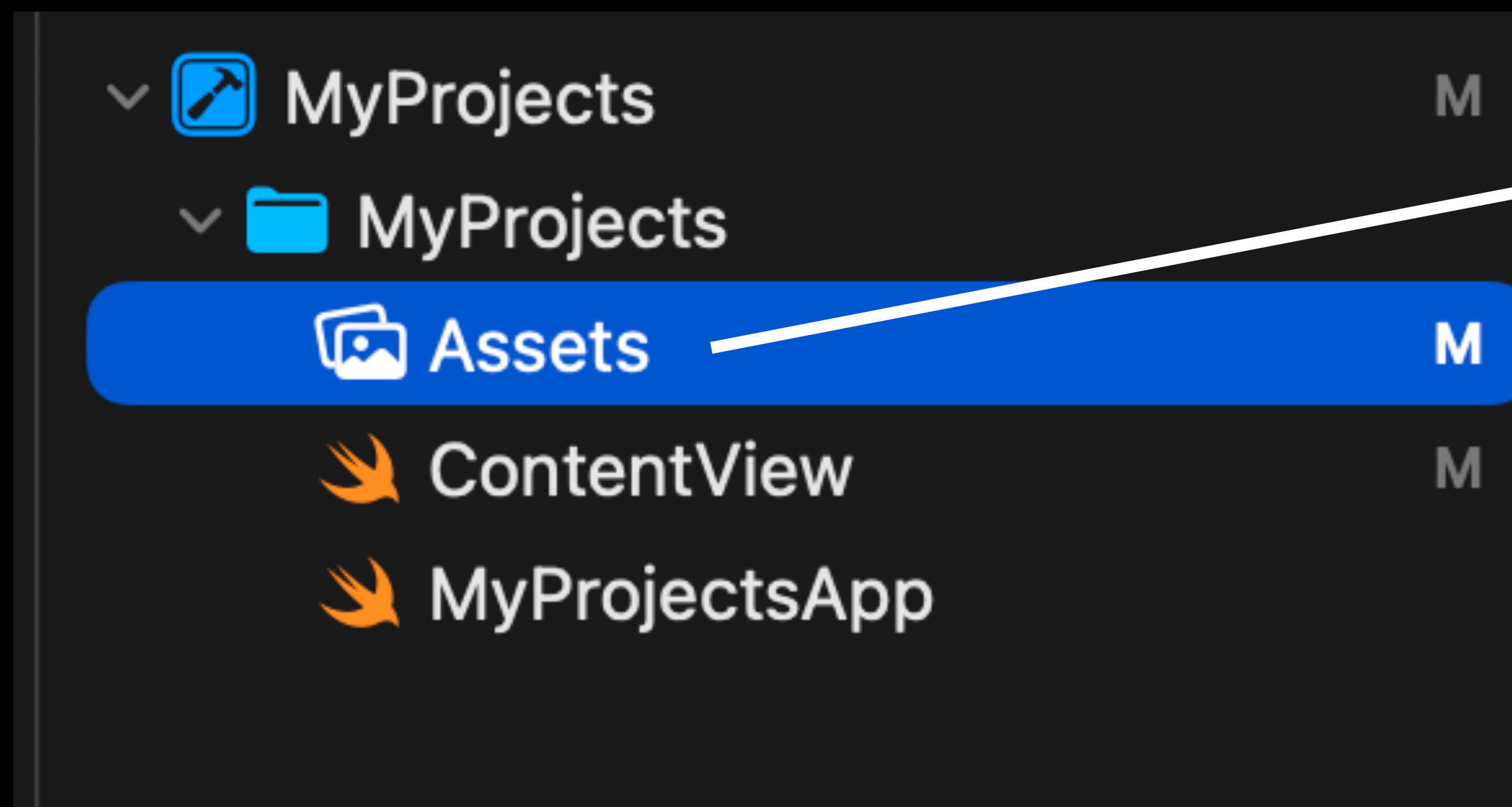
ipa ファイルの中身は右の通り
色々ファイル/フォルダが存在
しているが、xcassets は存在して
いないことがわかる



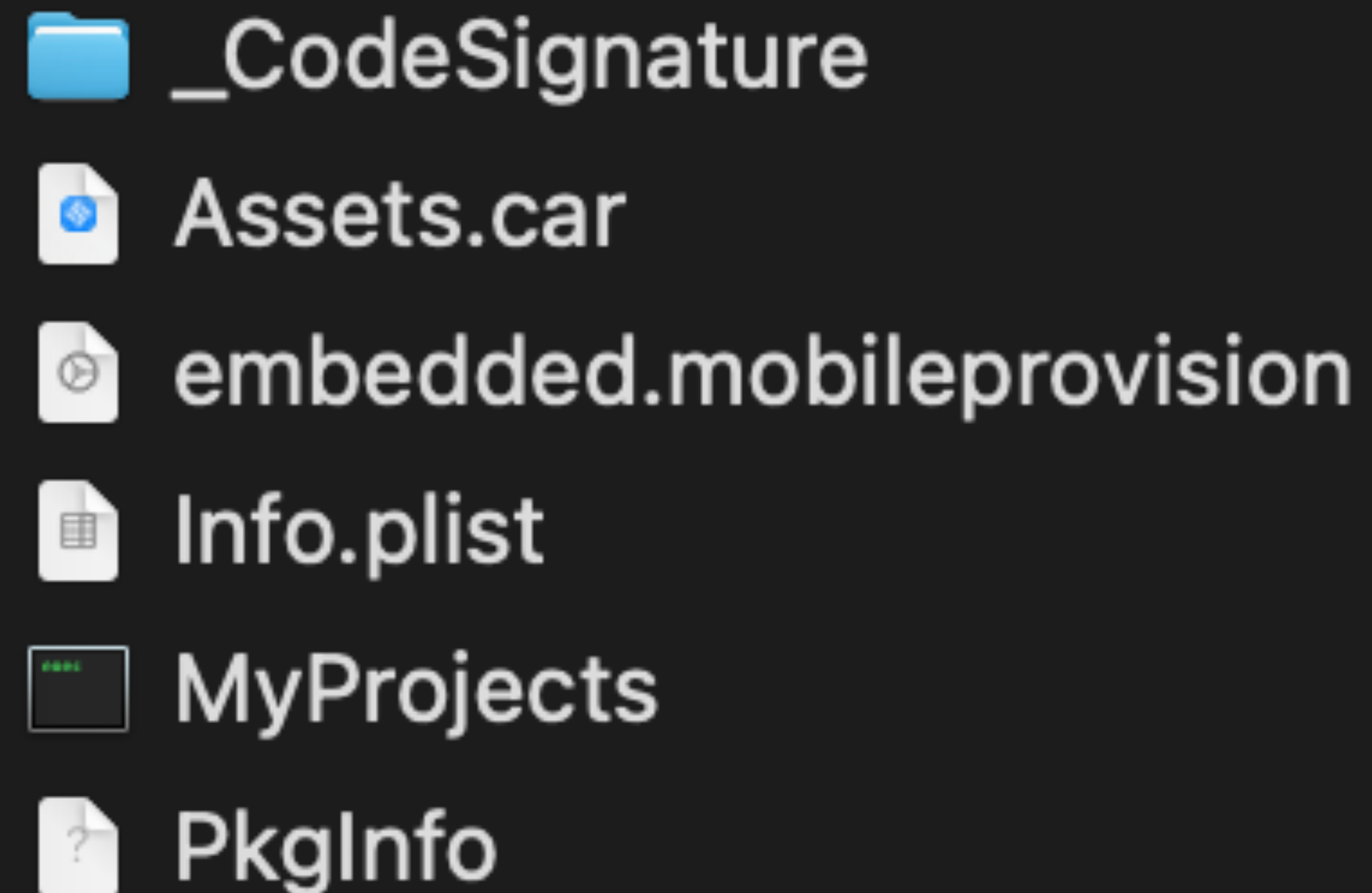
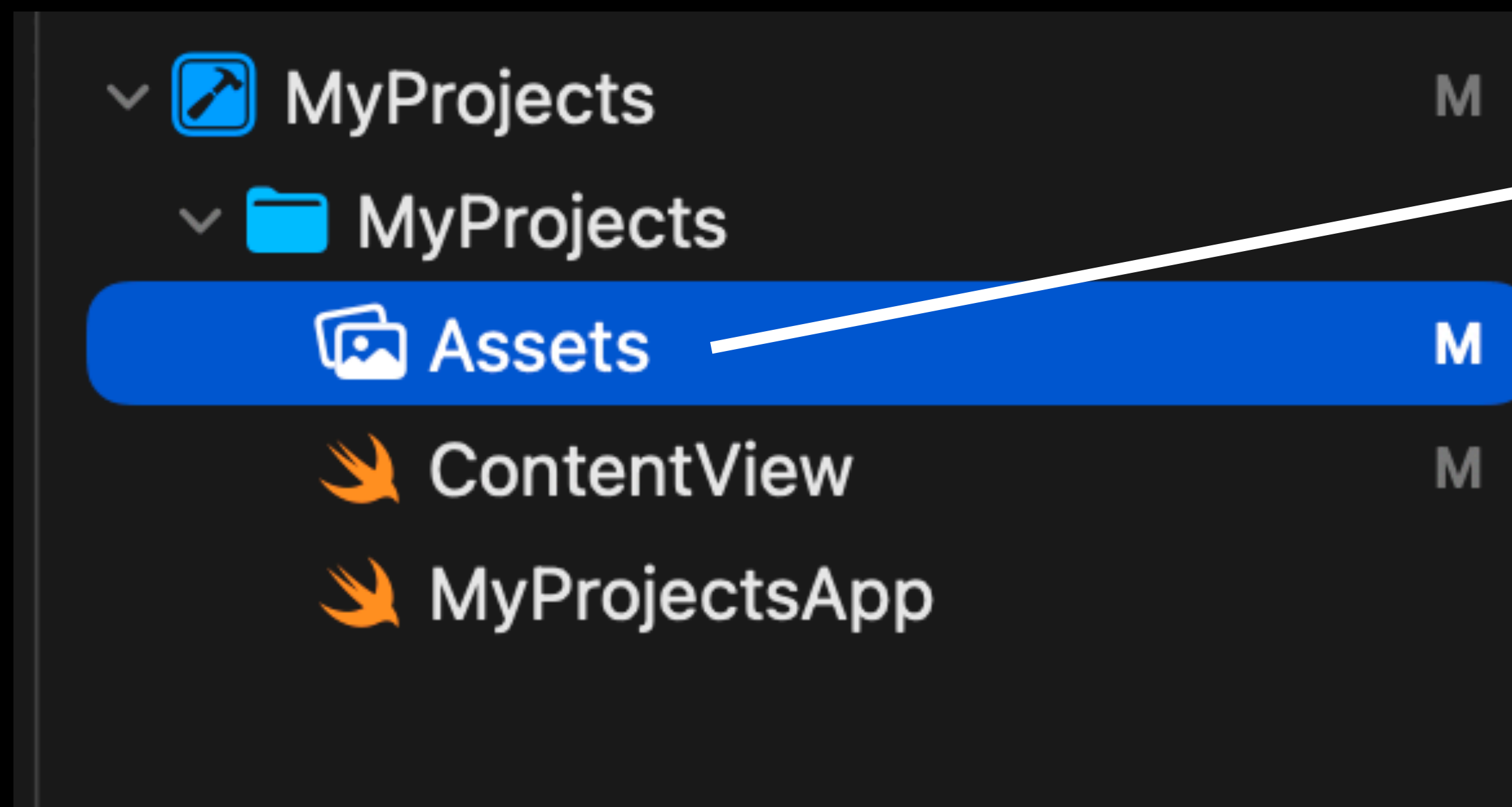
前提: xcassets は ipa に含まれていない



前提: xcassets は ipa に含まれていない

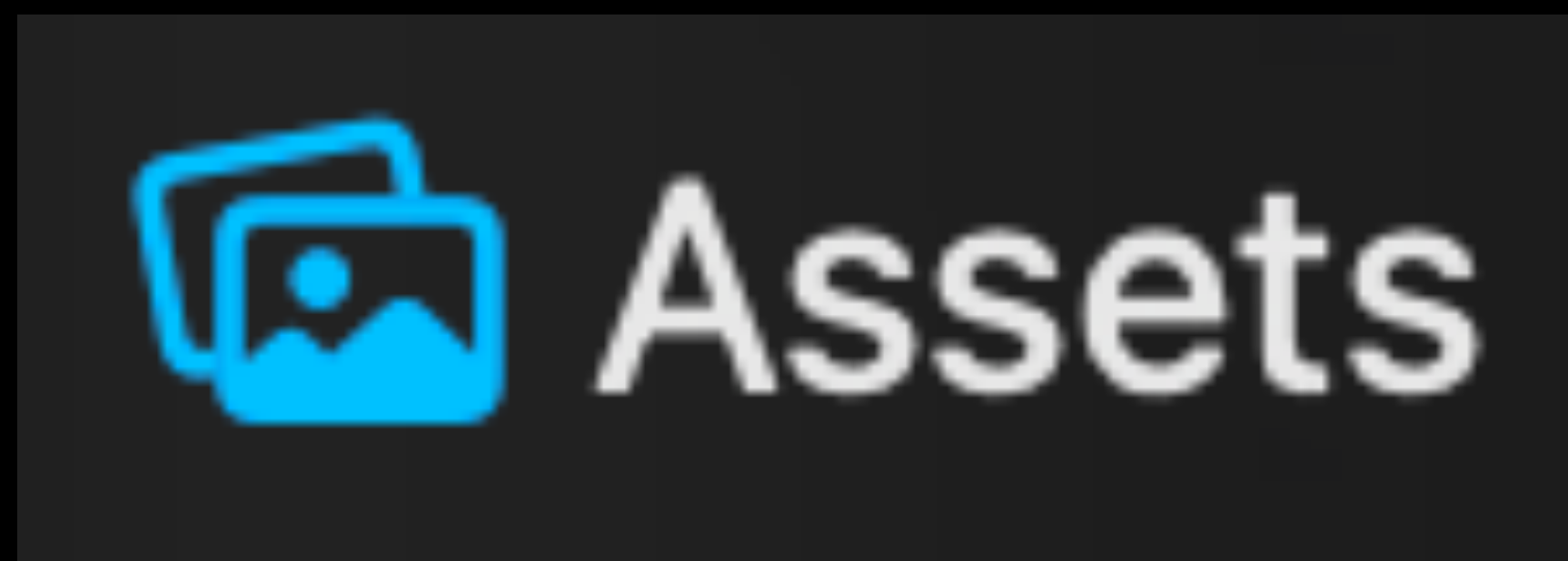


前提: xcassets は ipa に含まれていない



Assets.car になっている

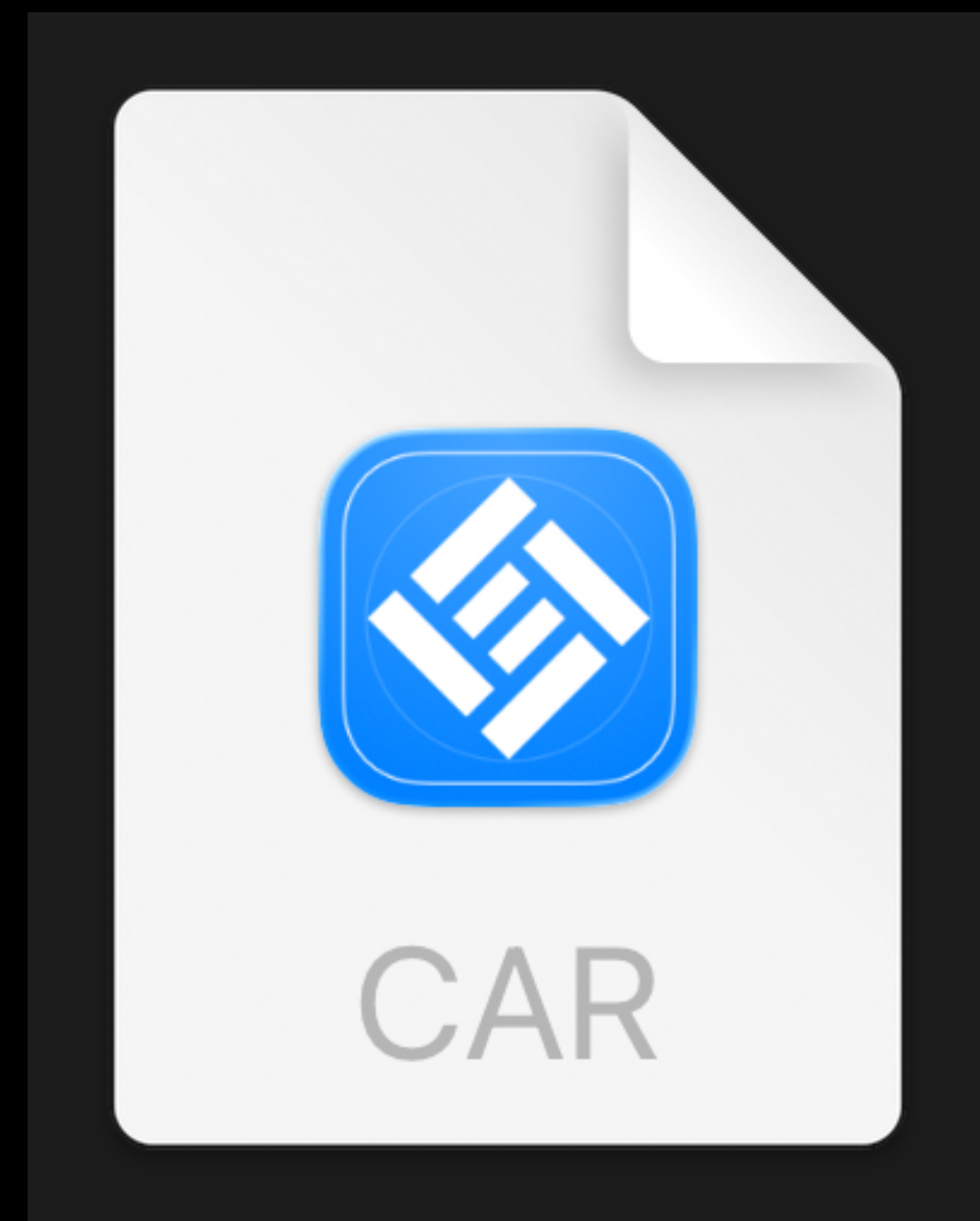
前提: xcassets は ipa に含まれていない



Assets.xcassets



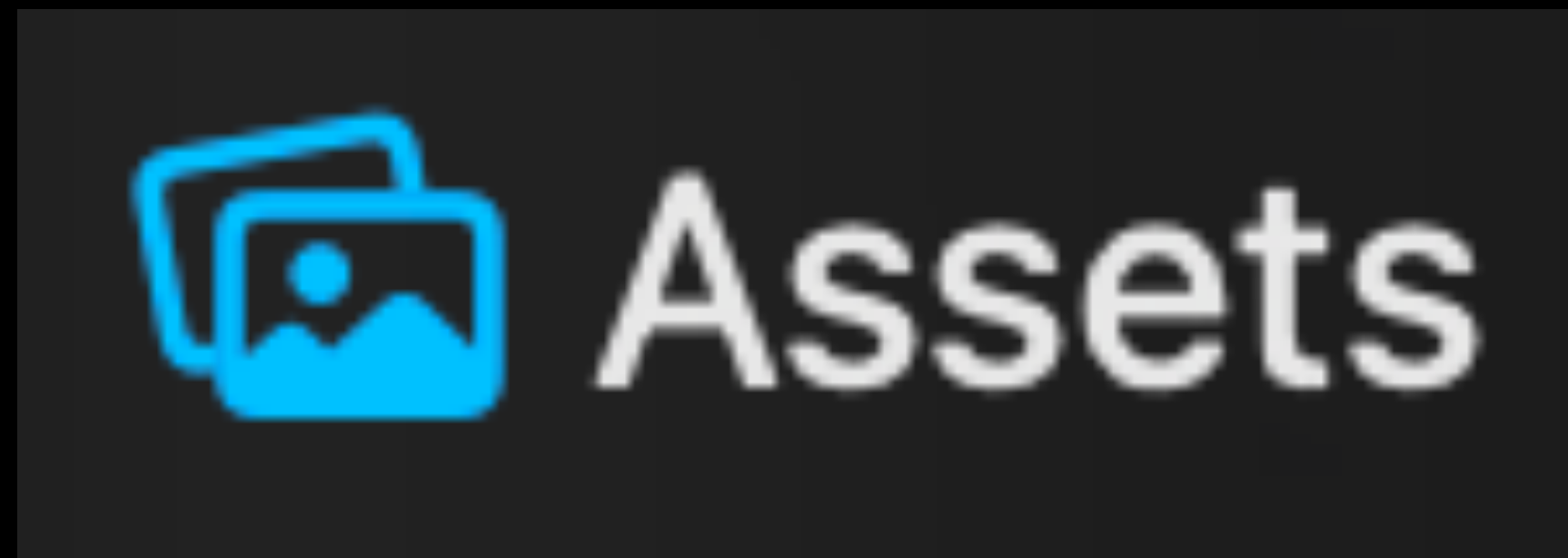
ビルド



Assets.car

前提: xcassets は ipa に含まれていない

ビルドの過程を経て Assets.car に変換されている



Assets.xcassets



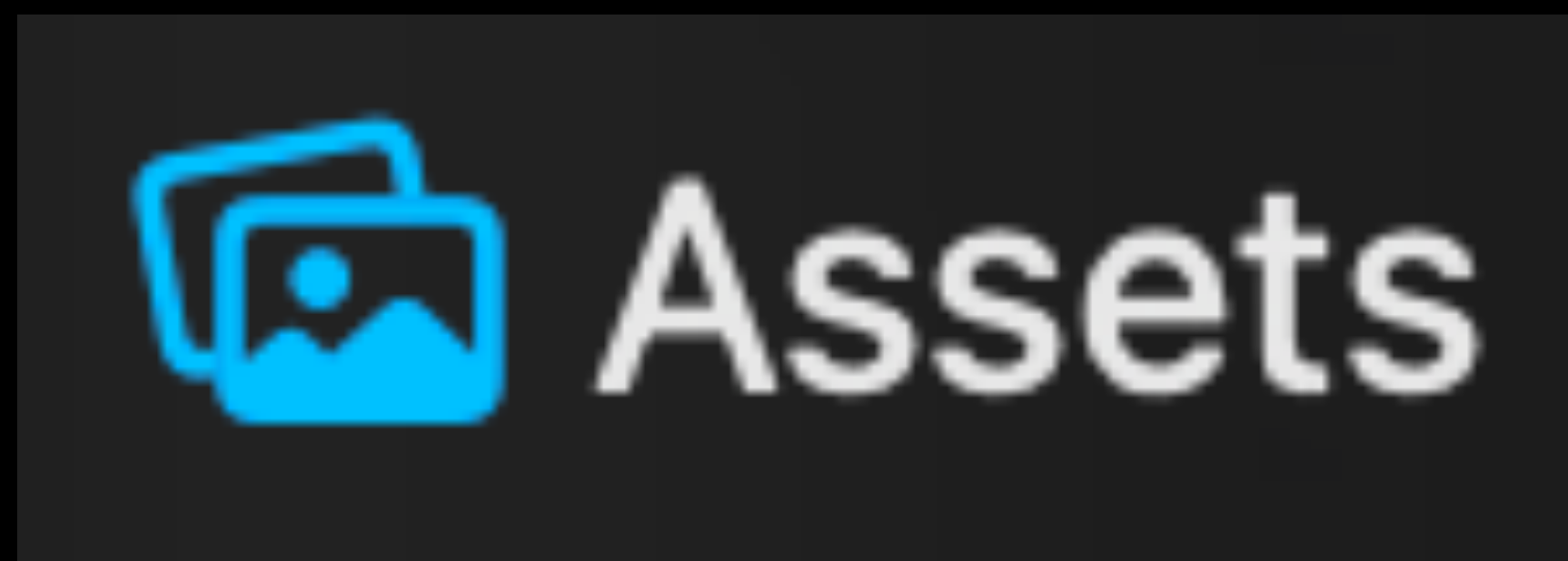
ビルド



Assets.car

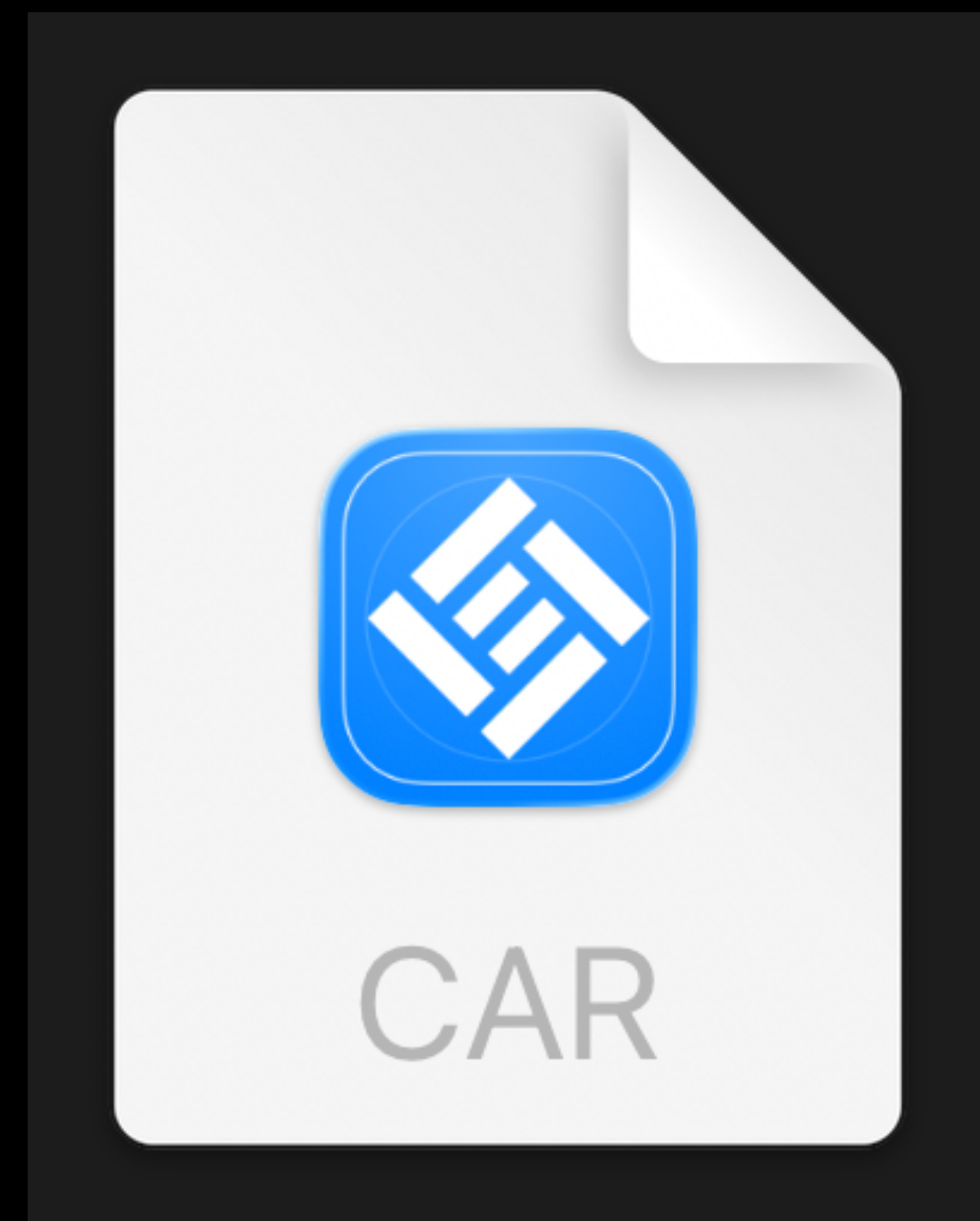
前提: xcassets は ipa に含まれていない

ビルドの過程を経て Assets.car に変換されている



Assets.xcassets

—————→
actool を利用してビルド



Assets.car

actool とは

actool とは

xcassets をコンパイルするツール

Xcode に内包されている

Xcode.app/Contents/Developer/usr/bin/ の中に入っている

ビルド時に自動で呼ばれアセットがコンパイルされる

actool とは

\$ man actool によりコマンドの使い方を見ることができる

```
actool(1)                                General Commands Manual                                actool(1)

NAME
    actool - compiles, prints, updates, and verifies asset catalogs.

SYNOPSIS
    actool [options] document

DESCRIPTION
    actool verifies, updates, and prints the contents of an asset catalog, generating its
    output in standard plist format. The tool follows a "read", "modify", "write", "print"
    order of operations.

OPTIONS
    Specifying Output:
        --output-format format
            By default, actool provides output in the form of an XML property list. Specifying
            binary1 will instruct actool to output a binary property list. Similarly, xml1
            specifies an XML property list, and human-readable-text specifies human readable
            text.

    Compiling:
        --compile path
```

actool とは

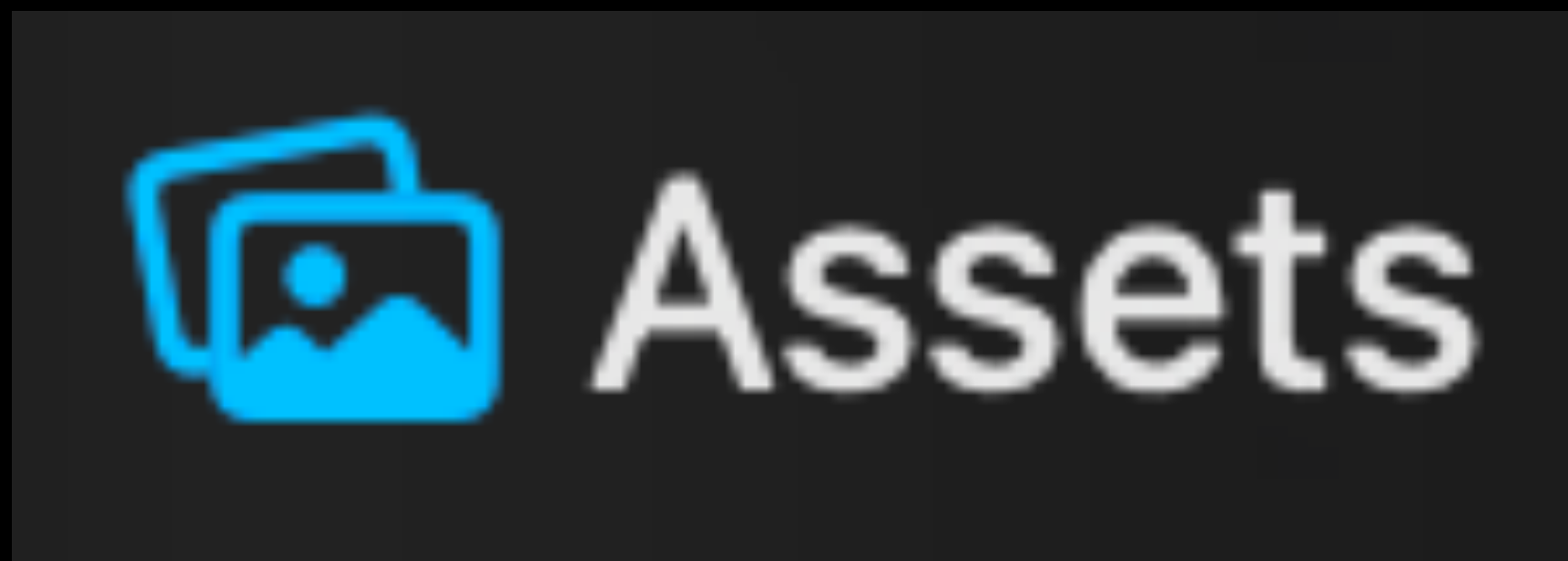
✓ Generate asset symbols 0.1 seconds

✓ Compile asset catalogs 14.3 seconds

```
CompileAssetCatalogVariant thinned /Users/jun/Library/Developer/Xcode/Diagnoses/Diagnostics/Logs/MyProjects/Assets.xcassets (in target 'MyProjects')
  cd /Users/jun/xcode/MyProjects
  /Applications/Xcode-26.0.0-beta.7.app/Contents/Developer/Tools/actool
MyProjects/Assets.xcassets --compile /Users/jun/Library/Developer/Xcode/Diagnoses/Diagnostics/Logs/MyProjects/Assets.xcassets --output
dgfulvjvucxqcmbpzzyauuzswpnbn/Build/Intermediates.noobj/MyProjects.build/assetcatalog_output/thinned --output
--export-dependency-info /Users/jun/Library/Developer/Xcode/Diagnoses/Diagnostics/Logs/MyProjects/Assets.xcassets
```

```
/Applications/Xcode-26.0.0-beta.7.app/Contents/Developer/usr/bin/actool \  
/Users/jun/xcode/MyProjects/MyProjects/Assets.xcassets \  
--compile \  
    /Users/jun/Library/Developer/Xcode/DerivedData/MyProjects-dgfulvjvucxqcmbpzzyaunuzswpn/Build/Intermediat  
    es.noindex/MyProjects.build/Debug-iphonesimulator/MyProjects.build/assetcatalog_output/thinned \  
--output-format human-readable-text \  
--notices \  
--warnings \  
--export-dependency-info \  
    /Users/jun/Library/Developer/Xcode/DerivedData/MyProjects-dgfulvjvucxqcmbpzzyaunuzswpn/Build/Intermedia  
    tes.noindex/MyProjects.build/Debug-iphonesimulator/MyProjects.build/assetcatalog_dependencies_thinned \  
--output-partial-info-plist \  
    /Users/jun/Library/Developer/Xcode/DerivedData/MyProjects-dgfulvjvucxqcmbpzzyaunuzswpn/Build/Intermedi  
    ates.noindex/MyProjects.build/Debug-iphonesimulator/MyProjects.build/assetcatalog_generated_info  
    .plist_thinned \  
--app-icon AppIcon \  
--accent-color AccentColor \  
--compress-pngs \  
--enable-on-demand-resources YES \  
--filter-for-thinning-device-configuration iPhone17,1 \  
--filter-for-device-os-version 26.0 \  
--development-region en \  
--target-device iphone --target-device ipad \  
--minimum-deployment-target 26.0 \  
--platform iphoneos
```

```
1
2 /Applications/Xcode-26.0.0-beta.7.app/Contents/Developer/usr/bin/actool \
3     /path/to/xcassets \
4     --compile path/to/output \
5     --output-format human-readable-text \
6     --notices \
7     --warnings \
8     --export-dependency-info /path \
9     --output-partial-info-plist /path \
10    --app-icon AppIcon \
11    --accent-color AccentColor \
12    --compress-pngs \
13    --enable-on-demand-resources YES \
14    --filter-for-thinning-device-configuration iPhone17,1 \
15    --filter-for-device-os-version 26.0 \
16    --development-region en \
17    --target-device iphone --target-device ipad \
18    --minimum-deployment-target 26.0 \
19    --platform iphoneos
```



Assets.xcassets

—————→
actool を利用してビルド



Assets.car

actool はプロジェクト外でも使える

actool はプロジェクト外でも使える

コマンドとして公開されているので、プロジェクト外でも使える

actool はプロジェクト外でも使える

コマンドとして公開されているので、プロジェクト外でも使える



xcassets ファイルがあればどこでも Assets.car を作成可能

actool はプロジェクト外でも使える

Assets.xcassets を作成する

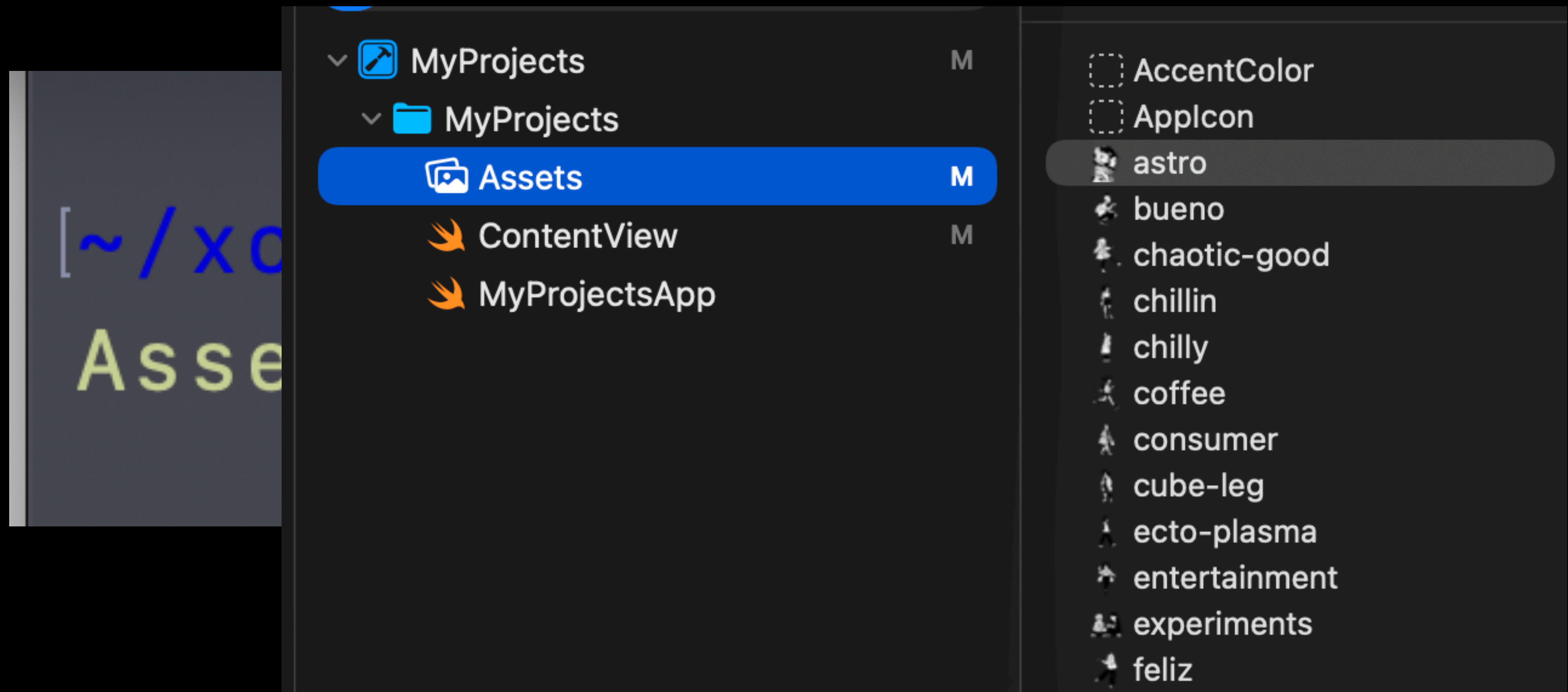
actool はプロジェクト外でも使える

Assets.xcassets を作成する

```
[~/xcode/tmp $ ls  
Assets.xcassets/
```

actool はプロジェクト外でも使える

先ほどの pjr ファイルの Assets をそのまま持ってきている



actool はプロジェクト外でも使える

actool のコマンドを実行する

actool はプロジェクト外でも使える

actool のコマンドを実行する

```
~/xcode/tmp $  
/Applications/Xcode-26.0.0-beta.7.app/Contents/Developer/usr/bin/actool \  
Assets.xcassets \  
--compile . \  
--output-format human-readable-text \  
--notices \  
--warnings \  
--output-partial-info-plist ./hoge.plist \  
--app-icon AppIcon \  
--accent-color AccentColor \  
--compress-pngs \  
--enable-on-demand-resources YES \  
--filter-for-thinning-device-configuration iPhone17,1 \  
--filter-for-device-os-version 26.0 \  
--development-region en \  
--target-device iphone --target-device ipad \  
--minimum-deployment-target 26.0 \  
--platform iphoneos
```

actool はプロジェクト外でも使える

結果として Assets.car を生成できる🎉

```
[~/xcode/tmp $ ls  
Assets.car          Assets.xcassets/  hoge.plist
```

Assets.car の中身を見る

Assets.car の中身を見る

assetutil コマンドを利用することで確認可能

Assets.car の中身を見る

assetutil コマンドを利用することで確認可能

```
[~/xcode/tmp $ assetutil --info Assets.car]
[
  {
    "Appearances" : {
      "UIAppearanceAny" : 0
    },
    "AssetStorageVersion" : "Xcode 26.0 (17A5305k) via AssetCatalogSimulatorAgent",
    "Authoring Tool" : "@(#)PROGRAM:CoreThemeDefinition PROJECT:CoreThemeDefinition-652 [IIIO-2773.0.1.2]",
    "CoreUIVersion" : 970,
    "DumpToolVersion" : 969,
    "Key Format" : [
      "kCRThemeAppearanceName",
      "kCRThemeLocalizationName",
      "kCRThemeScaleName",
      "kCRThemeIdiomName",
      "kCRThemeSubtypeName",
      "kCRThemeIdentifierName",
      "kCRThemeElementName",
      "kCRThemePartName"
    ],
    "MainVersion" : "@(#)PROGRAM:CoreUI PROJECT:CoreUI-970.1",
```

```
,  
{  
  "AssetType" : "Image",  
  "BitsPerComponent" : 8,  
  "ColorModel" : "Monochrome",  
  "Colorspace" : "gray gamma 22",  
  "Compression" : "deepmap2",  
  "Encoding" : "Gray",  
  "Idiom" : "universal",  
  "Name" : "rogue",  
  "NameIdentifier" : 49792,  
  "Opaque" : false,  
  "PixelHeight" : 3000,  
  "PixelWidth" : 3000,  
  "RenditionName" : "rogue.png",  
  "Scale" : 1,  
  "SHA1Digest" : "50C40D418D009087F56B077A8FDF9BC9AA442958D3BD81FBAE055B6B99820E57",  
  "SizeOnDisk" : 121069,  
  "Template Mode" : "automatic"  
},  
-
```

Assets.car の中身を見る

Asset Catalog Tinkererを使うと.carファイルの中身をビジュアルで見れる

<https://github.com/insidegui/AssetCatalogTinkerer>



Assets.car の中身を見る

Asset Catalog Tinkererを使うと.carファイルの中身をビジュアルで見れる

<https://github.com/insidegui/AssetCatalogTinkerer>

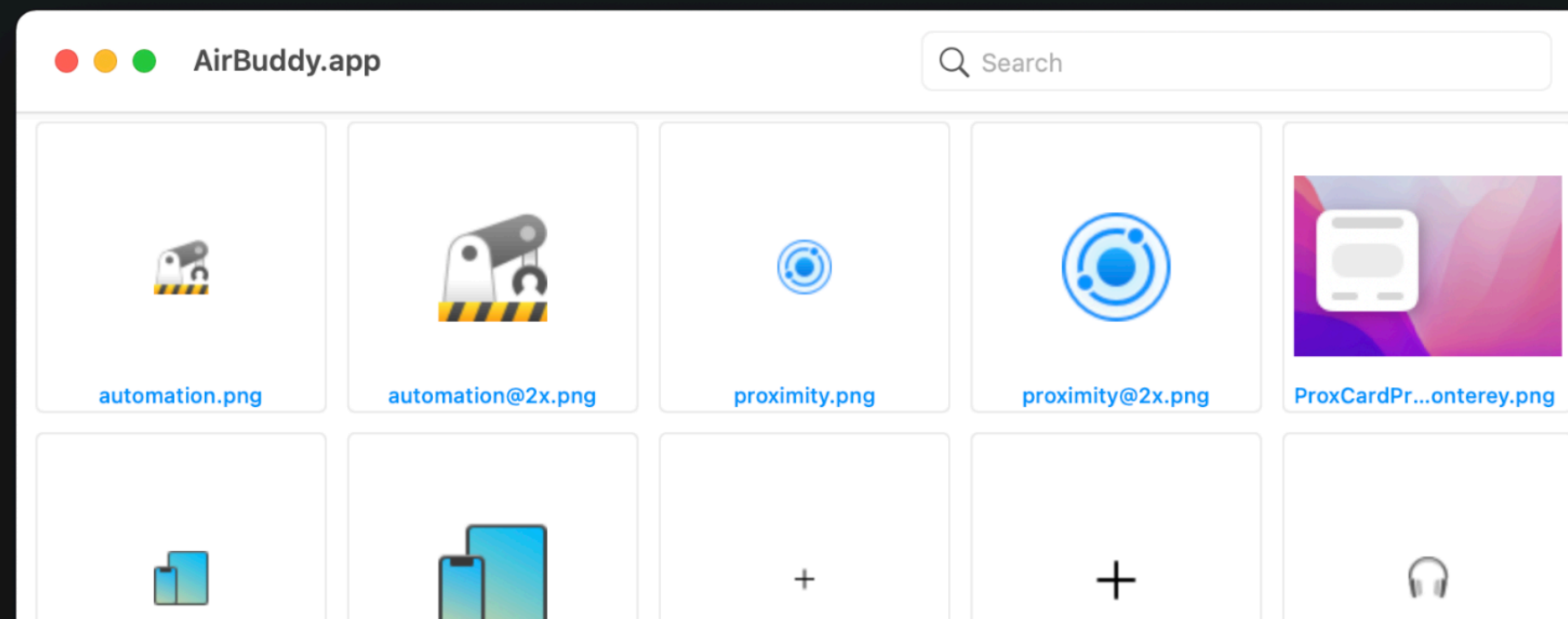
Asset Catalog Tinkerer

Asset Catalog Tinkerer lets you open asset catalog files (`.car`) and view the images that they contain. You can also copy and export individual images out or export all images from an asset catalog.

↓ [Download and contribute with any amount on Gumroad](#)

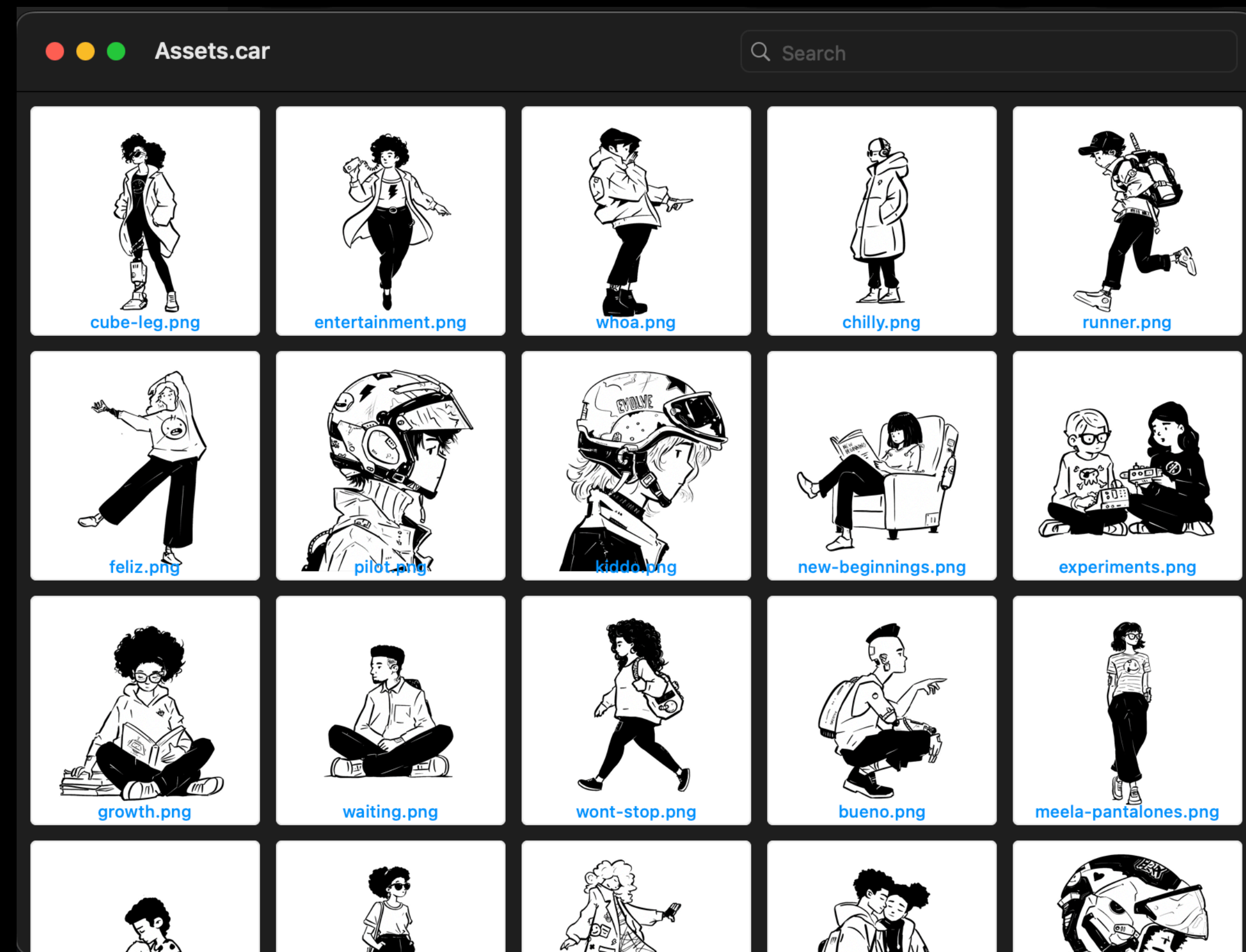
↓ [Download Latest Release](#)

You can also install it with [Homebrew](#) by running `brew install asset-catalog-tinkerer` !



Assets.car の中身を見る

Asset Catalog Tinkererを使うと.carファイルの中身をビジュアルで見れる



**xcassets がなくとも Assets.car
があれば画像を表示できる**

今日話すこと

1. 前提: xcassets は ipa に含まれていない
2. actool とは
3. actool はプロジェクト外でも使える
4. Assets.car の中身を見る
5. xcassets がなくても Assets.car があれば画像を表示できる
→ 事実上コンパイルをスキップすることが可能

**xcassets がなくとも Assets.car
があれば画像を表示できる**

デモ

まとめ

1. xcassets は actool が利用され Assets.car に変換される
2. actool は Xcode に同梱されており、コマンドとして利用可能
3. xcassets がなくても Assets.car があれば画像を表示できる

ありがとうございました！

