

— ChatGPT ▾ 無料オファー ×

↑ ⌂ ...

mac用のライブラリやサンプルプログラム、フレームワークを作った。

これをインストールできるようにインストーラーを作りたい。

おそらく `pkgbuild` と `productbuild` を使うと思うのだけどどうやってやるの？

またこれを自動化できる？

いいテーマですね。

macOS の 公式なインストーラ (`.pkg`) を作るなら、考えている通り **`pkgbuild + productbuild`** が王道です。

以下では **基本構造 → 実践例 → 自動化（スクリプト / CI）** の順で説明します。

全体像（`pkgbuild` / `productbuild` の役割）

css

□ コードをコピーする

[インストールされるファイル群]

↓

`pkgbuild` → `component.pkg` (中身入りのpkg)

↓

`productbuild` → `installer.pkg` (配布用)

コマンド

役割

`pkgbuild`

ファイルを配置する「部品pkg」を作る

`productbuild`

複数pkg・Distributionをまとめて最終pkgを作る

① まずはディレクトリ構成を決める

例：

ライブラリやフレームワークを `/Library/MySDK` に入れたい場合

makefile

□ コードをコピーする

```

project/
|   └── payload/
|       |   └── Library/
|       |       |   └── MySDK/
|       |           |       └── libmylib.a
|       |           |       └── include/
|       |           └── Samples/
|
|   └── scripts/
|       |   └── postinstall
|       └── preinstall (必要なら)

```

⚠ payload以下の構造がそのまま / に展開される のが重要。

② pkgbuild で component.pkg を作る

sh

コードをコピーする

```

pkgbuild \
--root payload \
--scripts scripts \
--identifier com.example.mysdk \
--version 1.0.0 \
--install-location / \
MySDK.pkg

```

オプション解説

オプション	意味
--root	インストール対象ファイル
--scripts	preinstall / postinstall
--identifier	pkgのID (逆DNS推奨)
--install-location	展開先 (通常 /)

postinstall の例

sh

コードをコピーする

```
#!/bin/sh
echo "MySDK installed"
chmod -R 755 /Library/MySDK
exit 0
```

sh

□ コードをコピーする

```
chmod +x scripts/postinstall
```

③ productbuild で配布用 pkg を作る

単純なケース：

sh

□ コードをコピーする

```
productbuild \
--package MySDK.pkg \
MySDK-Installer.pkg
```

これで ダブルクリックでインストールできるpkg が完成。

④ Distribution を使う (UI制御したい場合)

- ✓ ライセンス表示
- ✓ macOS バージョン制限
- ✓ 複数pkg
- ✓ インストール説明文

Distribution.xml 作成

sh

□ コードをコピーする

```
productbuild --synthesize --package MySDK.pkg Distribution.xml
```

編集例：

xml

□ コードをコピーする

```
<installer-gui-script minSpecVersion="1">
<title>MySDK Installer</title>
```

```

<options customize="never"/>

<installation-check>
  <os-version min="11.0"/>
</installation-check>

<choices-outline>
  <line choice="default">
    <line choice="mysdk"/>
  </line>
</choices-outline>

<choice id="mysdk" visible="false">
  <pkg-ref id="com.example.mysdk"/>
</choice>

<pkg-ref id="com.example.mysdk" version="1.0.0">MySDK.pkg</pkg-ref>
</installer-gui-script>

```

Distribution 付き productbuild

sh

□ コードをコピーする

```

productbuild \
--distribution Distribution.xml \
--package-path . \
MySDK-Installer.pkg

```

⑤ 署名 (macOSではほぼ必須)

Installer証明書で署名

sh

□ コードをコピーする

```

productbuild \
--sign "Developer ID Installer: Your Name (TEAMID)" \
--package MySDK.pkg \
MySDK-Installer.pkg

```

確認：

```
sh  
pkgutil --check-signature MySDK-Installer.pkg
```

⑥ 自動化できる？ → 完全に可能

1 シェルスクリプト化（最小）

```
sh  
 コードをコピーする  
  
#!/bin/sh  
set -e  
  
VERSION=1.0.0  
IDENTIFIER=com.example.mysdk  
  
pkgbuild \  
  --root payload \  
  --identifier $IDENTIFIER \  
  --version $VERSION \  
  MySDK.pkg  
  
productbuild \  
  --package MySDK.pkg \  
  MySDK-$VERSION.pkg
```

2 Makefile

```
makefile  
 コードをコピーする  
  
VERSION=1.0.0  
  
all: build  
  
build:  
    pkgbuild --root payload \  
        --identifier com.example.mysdk \  
        --version $(VERSION) MySDK.pkg  
    productbuild --package MySDK.pkg MySDK-$(VERSION).pkg
```

clean:

```
rm -f *.pkg
```

3 CI (GitHub Actions例)

yaml

 コードをコピーする

```
- name: Build Installer
  run: |
    pkgbuild --root payload \
      --identifier com.example.mysdk \
      --version $VERSION MySDK.pkg
    productbuild --package MySDK.pkg MySDK.pkg
```

※ 署名する場合は

- Developer ID証明書
 - Keychainの設定
が必要。
-

⑦ フレームワークの場合の注意点

- /Library/Frameworks
- /Applications/MyApp.app/Contents/Frameworks
- rpath / install_name_tool

などを postinstall で調整するケースが多い。

まとめ

- ✓ pkgbuild → 中身
 - ✓ productbuild → 配布用
 - ✓ Distribution.xml でUI制御
 - ✓ shell / Makefile / CIで自動化可能
 - ✓ macOS配布なら署名はほぼ必須
-

次のステップとしておすすめなのは：

- **Homebrew Formula化**
- **uninstall用pkg**
- **notarization対応**

どこまでやりたいですか？



Is this conversation helpful so far?