

アプリケーションの 配布に関するガイド

目次

アプリケーション配布作業の流れについて 11

はじめに 12

Apple Developer Programに加入してアプリケーションを配布する 12

サービスをアプリケーションに追加する 12

アプリケーションを配布できるよう準備する 13

アプリケーションを各種のデバイスやリリース上でテストする 13

ストアでのアプリケーションの登録と公開 13

ストアを通さずにアプリケーションを配布する 14

証明書、ID、プロファイルを管理する 14

この文書の使い方 14

関連項目 17

アプリケーションの間引き (iOS、tvOS、watchOS) 19

切り分け (iOS、tvOS) 19

ビットコード 21

オンデマンド型リソース (iOS、tvOS) 22

アカウントの管理 24

Apple Developer Programについて 25

個人として、あるいは組織として加入する 25

同じ人が複数のチームに加わることもできる 25

Xcodeで自分のApple IDアカウントを追加する 26

まとめ 29

アプリケーションを配布するためのXcodeプロジェクトの設定 30

Xcodeプロジェクトの作成時にプロパティを設定する 30

プロジェクトの設定を始める前に 32

IDおよびチームの設定をする 33

バンドルIDについて 34

バンドルIDを設定する 35

署名IDを選択する (Mac) 37

Xcodeプロジェクトをチームに対応づける 38

チームプロビジョニングプロファイルを生成する 39

アプリケーションカテゴリを設定する (Mac) 40

	版番号とビルド文字列を設定する	41
配備情報を設定する		43
	配備ターゲットを設定する	43
	ターゲットデバイスを設定する (iOS、watchOS)	44
アプリケーションアイコンや起動画面ファイルを追加する		45
	アートワークを用意する	46
	アプリケーションアイコンをアセットカタログに追加する	46
	起動画面ファイルを作成する	47
	起動画像を追加し、スクリーンショットを取得する	49
	アプリケーションアイコンと起動画像ファイルを個別に設定する	50
	画像をアセットカタログに移して管理する	50
	「Copyright」キーを設定する (Mac)	51
ビルド設定を検証する		52
	アーキテクチャを設定する (iOSアプリケーション)	53
	ベースSDKを設定する	54
	デバッグ情報の形式を設定する	55
まとめ		55
技術やサービスを組み込む		56
	エンタイトルメントについて	57
	事前の確認	57
	アプリケーションサンドボックスを設定する (Mac)	59
iCloudの操作機能を追加する		62
	iCloud を有効にする	62
	キー値ストレージを設定する	62
	文書ストレージを設定する	63
	CloudKitを使う	63
	独自のコンテナを指定する	64
プッシュ通知の設定をする (iOS、watchOS、Mac)		65
	プッシュ通知を有効にする	66
	プッシュ通知用のクライアントSSL証明書を作成する	66
	クライアントSSL署名IDをサーバにインストールする	69
Game Centerを有効にする (iOS、tvOS、Mac)		70
ゲームコントローラを有効にする (tvOS)		71
Walletを設定する (iOS、WatchKitエクステンション)		72
Apple Payを設定する (iOS、WatchKitエクステンション)		74
In-App Purchaseを有効にする (iOS、tvOS、Mac)		75
個人向けVPNを有効にする (iOS、Mac)		76
	「マップ(Maps)」を設定する (iOS、WatchKitエクステンション、Mac)	77

Xcodeで「Maps」を有効にする	77
経路制御アプリケーションを設定する (iOS、WatchKitエクステンション)	78
バックグラウンドモードを設定する (iOS、tvOS)	80
Inter-App Audioを有効にする (iOS)	81
キーチェーン共有の設定をする	82
Associated Domainsを有効にする (iOS、WatchKitエクステンション)	84
アプリケーショングループを設定する	84
HomeKitを追加する (iOS、WatchKitエクステンション)	86
データ保護を有効にする (iOS、WatchKitエクステンション、tvOS)	87
HealthKitを追加する (iOS、WatchKitエクステンション)	88
ワイヤレスアクセサリ設定の有効化 (iOS)	89
Newsstandを設定する (iOS、WatchKitエクステンション)	89
トラブルシューティング	90
まとめ	90
デバイス上でアプリケーションを起動する手順	91
Macアプリケーションを起動する	91
iOS/tvOS/watchOSアプリケーションを起動する	92
デバイスを「Scheme」メニューから削除する (iOS、tvOS、watchOS)	93
アプリケーションをデバイスから削除する (iOS、tvOS、watchOS)	94
ここまでの結果を確認する	95
トラブルシューティング	98
チームプロビジョニングプロファイルの詳細	99
まとめ	101
テスト用にアプリケーションをエクスポート (iOS、tvOS、watchOS)	102
アドホックプロビジョニングプロファイルについて	102
テスト用デバイスを登録する	103
アプリケーションをアーカイブする	103
ストアを通さず、テスト用にアプリケーションをエクスポートする	104
Xcodeを使ってテスト用デバイスにアプリケーションをインストールする	109
iTunesを使ってテスト用デバイスにアプリケーションをインストールする (iOS、watchOS)	110
Xcodeサーバを利用してアプリケーションを配布する	111
アプリケーションのサンドボックスデータをコピーする	111
テスト担当者にクラッシュレポートを送るよう依頼する	111
アドホックプロビジョニングプロファイルの詳細	113
まとめ	113
アプリケーションをテスト用にエクスポート (Mac)	114
テスト用Macコンピュータを登録する	114

- アプリケーションをアーカイブする 114
- チームプロビジョニングプロファイルを使ってアプリケーションをエクスポートする 115
- アプリケーションをテスト用Macコンピュータにインストールする 117
- Macインストーラパッケージをテストする 118
 - インストーラパッケージを作成する 118
 - インストーラパッケージをテストする 119

アプリケーションをiTunes Connectにアップロード 121

- ストアプロビジョニングプロファイルについて 121
- iTunes Connectでアプリケーションレコードを作成する 122
- 版番号やビルド文字列を更新する 122
- アプリケーションをアーカイブする 123
 - 「Archive」スキーム設定を確認する 123
 - アーカイブを作成する 124
- iTunes Connectの検証テストを実行する 125
- アプリケーションをアップロードする 126

TestFlightによるアプリケーションの配布 (iOS、tvOS、watchOS) 130

クラッシュレポートの分析 132

- クラッシュレポートサービスについて 132
- クラッシュレポートを閲覧する準備 134
 - 「Crashes」オーガナイザでクラッシュレポートを閲覧する 135
- クラッシュレポートを表示、検索する 136
- クラッシュレポートに関する情報を編集する 137
- ソースコードをデバッグナビゲータで開く 138
- クラッシュレポートに関する統計情報を表示する 141
- クラッシュデータをアプリケーション開発者に提供する 141
- クラッシュデータを「Devices」ウィンドウに表示し、インポートする 145
- Xcode上でクラッシュを再現する 147
- 表示可能なクラッシュレポート 148

アプリケーションの登録 149

- アプリケーションの登録に向けて準備する 149
 - ヒューマンインターフェイスおよびストアに関するガイドラインを確認する 149
 - 必要な追加情報をiTunes Connectに入力する 150
 - Xcodeプロジェクトを検証する 150
- 最終候補をアップロードする 151
- アプリケーションを登録する 151
- まとめ 151

iTunes Connectでのアプリケーション管理 152

- iTunes Connectユーザの役割と権限について 153
- iTunes Connectにアクセスする 153
- iTunes Connectユーザを追加し役割を確認する 154
- アプリケーションレコードを作成する 155
 - tvOSアプリケーションをiOSアプリケーションレコードに追加する 156
- アプリケーションの状態を確認する 156
- アプリケーションの公開日を変更する 157
- 解析結果を表示する 158
- 顧客レビューの表示 158
- アプリケーションの新版を作成する 159
- まとめ 159

メンバーセンターでのチームの管理 160

- Apple Developer Programにおけるチームの役割と権限について 160
 - チームの役割 160
 - チームの権限 161
 - チームエージェント 162
- チームメンバーを加えて役割を与える 162
 - チームメンバーを加える 162
 - チームの役割を変更する 164
- チームメンバーのデバイスを登録する 165
- チームエージェントの役割を委譲する 166
- チームメンバーを外す 167
- まとめ 167

署名IDや証明書の管理 168

- 署名IDと証明書について 168
- 署名IDやプロビジョニングプロファイルの内容を表示する 170
- 署名IDを作成する 172
 - ここまでの結果を確認する 174
 - トラブルシューティング 177
- 追加のDeveloper ID証明書を作成する 177
- 誤って削除した中間証明書を登録する 179
- 証明書やプロファイルをエクスポート/インポートする 180
 - 開発者プロファイルをエクスポートする 181
 - 特定の証明書のみをエクスポートする 182
 - 開発者プロファイルをインポートする 183
- 署名IDをキーチェーンから削除する 184
- 証明書を失効させる 185

権限を失効させる	186
Xcodeで証明書をリセットする	187
メンバーセンターで証明書を失効させる	188
期限切れになった証明書を更新する	189
証明書を再生成し、関係するプロビジョニングプロファイルを更新する	189
署名証明書の詳細	192
まとめ	194
ID、デバイス、プロファイルの管理	195
App IDを登録する	195
App IDを編集する	198
App IDを削除する	199
チームIDを調べる	201
メンバーセンターでデバイスを登録する	202
デバイスIDを調べる	202
個々のデバイスを登録する	205
デバイスを一括登録する	206
プレリリース版をiOS/Apple TVデバイスにインストールする	207
メンバーセンターでデバイスの有効/無効を切り替える	208
デバイス上のアプリケーションを管理する	209
からアプリケーションを削除する	209
デバイスでアプリケーションコンテナを表示、ダウンロード、置換する	210
メンバーセンターでプロビジョニングプロファイルを生成する	211
開発用プロビジョニングプロファイルを作成する	212
アドホックプロビジョニングプロファイルを作成する (iOS、tvOS、watchOS)	213
ストアプロビジョニングプロファイルを作成する	215
Xcode上でプロビジョニングプロファイルをダウンロードする	215
独自のプロビジョニングプロファイルを使用する	216
トラブルシューティング	219
Xcodeで管理されるプロビジョニングプロファイルを移行する	219
メンバーセンターでプロビジョニングプロファイルを編集する	220
期限切れになったプロビジョニングプロファイルを更新する	222
デバイス上でプロビジョニングプロファイルを確認して削除する	223
プロビジョニングプロファイルをメンバーセンターから削除する	224
プロビジョニングプロファイルをメンバーセンターからダウンロードする	225
アプリケーションバイナリのエンタイトルメントを検証する	226
まとめ	231
Apple Developer Enterprise Programアプリケーションの配布	232
Apple Developer Enterprise Programアプリケーションを開発する	232

チームを結成する (チームエージェント)	232
共有チームプロビジョニングプロファイルを作成する (チーム管理者)	233
開発を始める (チームメンバー)	233
Apple Developer Enterprise Programアプリケーションをテストする	234
別のエンタープライズ配布用証明書を作成する	234
証明書とプロビジョニングプロファイルの有効期限を管理する	236
社内でアプリケーションを配布する	236
アーカイブを作成する	236
iOS Appファイルを作成する	237
サーバツールを詳しく把握する	239
まとめ	240
Mac App Storeを通さずにアプリケーションを配布する手順	241
Developer IDを使って署名したアプリケーションやインストーラパッケージを作成する	241
署名IDとしてDeveloper IDを設定する	241
Developer ID証明書を作成する	242
アーカイブを作成する	243
Developer IDで署名したアプリケーションを検証する	243
Developer IDで署名したアプリケーションをエクスポートする	244
インストーラパッケージに署名する	245
ここまでの結果を確認する	246
Gatekeeperの有効/無効を切り替える	246
Gatekeeperの動作をテストする	248
まとめ	250
トラブルシューティング	251
証明書に関する問題	251
コード署名IDが見つからない	251
該当するプロビジョニングプロファイルが見つからない	251
Developer ID証明書に用いる秘密鍵が欠落している	252
中間証明書を紛失したため証明書が無効になった	252
証明書の信頼性に問題がある	252
証明書が期限切れになった	252
要求がタイムアウトになった	252
プロビジョニングプロファイルや署名IDがXcodeのメニューに現れない	253
プロビジョニングに関する問題	253
デバイスにインストールしたプロビジョニングプロファイルが無効である	253
メンバーセンターのプロビジョニングプロファイルが無効である	253
該当するプロビジョニングプロファイルが見つからない	254
ビルドおよびコード署名に関する問題	254

Xcodeがプロビジョニングプロファイルを見つけられない	254
Xcodeが証明書を信頼しない	255
ビルド設定「Code Signing Identity」がどの証明書とも合致しない	255
キーチェーンにコード署名IDが重複して登録されている	256
プロビジョニングプロファイルのアプリケーションIDが、アプリケーションのバンドルIDと一致しない	256
デバイスがリストに表示されないか、「Scheme」ツールバーに選択不可として表示される	257
デバッグ情報に関する問題	257
デバイスに接続したとき、Xcodeに「Unknown iOS Detected」というダイアログが現れる	257
アーカイブおよび登録の問題	257
サポートされるケイパビリティ	258
書類の改訂履歴	261
用語解説	263

表

iTunes Connectでのアプリケーション管理 152

表 12-1 iTunes Connectにおける役割と責任 153

メンバーセンターでのチームの管理 160

表 13-1 チームの役割 161

表 13-2 役割に応じた権限 161

署名IDや証明書の管理 168

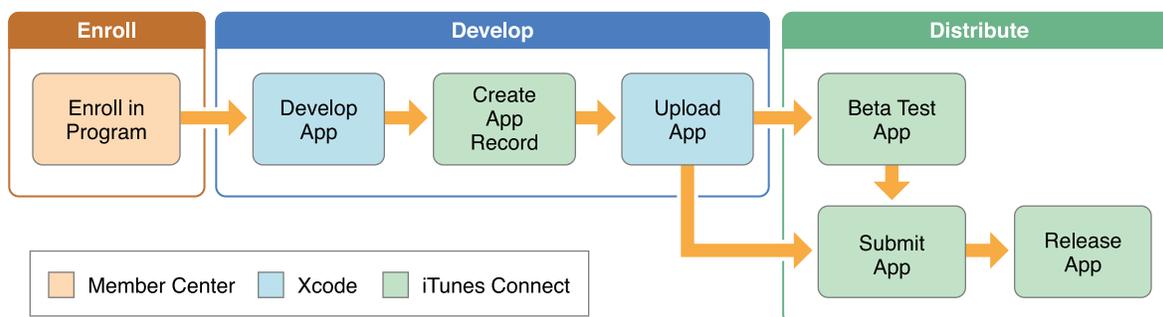
表 14-1 チーム証明書を失効させる権限 186

表 14-2 証明書の種類と名前 193

アプリケーション配布作業の流れについて

このガイドでは、App StoreやApple TV App Store、Mac App Storeを通してアプリケーションを配布するために知っておくべき、さまざまな事項を説明します。

- Apple Developer Programに加入し、アプリケーションを構築、テスト、登録する手順。
- ストアに登録したアプリケーションでしか利用できないサービスに関する設定。
- 各種のデバイスやシステム上でテストする手順、次にリリースするプレビュー版をテスト担当者に提供する手順。
- アプリケーションに関するメタデータをアップロードし、ストア上に表示して顧客に見てもらうための手順。
- アプリケーションがストアに登録可能な状態であることを確認し、ビルドをアップロードし、実際に登録する方法。
- アプリケーションの登録後、一般向けにリリースし、保守する方法。



こういった作業は、Xcodeの各種機能やウェブツールを使って行います。これはApple Developer Programの会員しか利用できません。iCloudやGame Centerなど、特定のアプリケーションサービスを利用するには、Apple Developer Programに加入する必要があります。ストアを通さずにアプリケーションを配布する場合であっても、顧客が配布元を確認できるようにするための手続きとして、Apple Developer Programに加入してください。

MacとiOS/tvOS/watchOSのどちらを対象とするアプリケーションも、開発にはXcodeを使います。Macアプリケーションは、OS Xが動作する開発用コンピュータ上で起動します。一方、iOS/tvOS/watchOSアプリケーションは、該当するデバイス、またはMac上で動作するシミュレータで実行しなければなりません。iOSアプリケーションが動作するデバイスは、iPad、iPhone、iPod touchです。tvOSアプリ

ケーションはApple TV上で動作します。watchOSアプリケーションの場合は、Apple WatchとペアリングしたiPhoneを使います。Xcodeには3種類のデバイスすべてに対応したシミュレータが付属していますが、十分にテストするためには、実デバイスをMacに接続する必要があります。

注意: Xcodeを使って、デバイス上でアプリケーションを実行したい、あるいはアプリケーションサービスを呼び出すコードを記述したいだけであれば、まず『*App Distribution Quick Start*』を読んでからこの資料に戻り、アプリケーションの存続期間中に実施すべき作業について目を通すとよいでしょう。

はじめに

このガイドは、アプリケーションを開発し、テストし、アップロードし、登録、公開する手順を説明するものです。各種ツールや配布手順を把握しておけば、開発した最新のアプリケーションやその更新版を、迅速に顧客に配布できるでしょう。

Apple Developer Programに加入してアプリケーションを配布する

Apple Developer Programへの加入は、[App Review](#)にアプリケーションを登録したり、社内向けiOSアプリケーションを配布したり、Mac App Storeを通さずに配布するアプリケーションにDeveloper IDで署名したりするための第一歩です。会員は、アプリケーションサービスを設定するのに必要な資料を入手し、新規アプリケーションやその更新版を登録することができます。

関連する章: [“アカウントの管理”](#) (24 ページ)

サービスをアプリケーションに追加する

Appleは、ある種の技術を利用するアプリケーションに対してはGame CenterやNewsstand、金銭のやり取りを伴うアプリケーションに対してはIn-App PurchaseやiAd Networkなど、高度な統合サービスを提供しています。このようなアプリケーションサービスを利用するためには、開発時とストアに登録する際に、若干の設定が必要です。その代表的な例として、Game CenterやiCloudの設定があります。このガイドでは、このようなキーとなる技術やサービスを、アプリケーションに組み込めるよう設定する方法を説明します。

関連する章: [“技術やサービスを組み込む”](#) (56 ページ)

アプリケーションを配布できるよう準備する

アプリケーションをテスト用に配布するか、承認を得るためにストアに登録するには、Xcodeプロジェクトに必要な設定を済ませておかなければなりません。開発が進んだXcodeプロジェクトには、アプリケーションアイコンや起動画像が保持されているとともに、動作するデバイスやオペレーティングシステムが指定されているはずです。

関連する章: [“アプリケーションを配布するためのXcodeプロジェクトの設定”](#) (30 ページ)

アプリケーションを各種のデバイスやリリース上でテストする

iOS/tvOS/watchOSアプリケーションは、シミュレータ上だけでなく、動作すると称するすべてのデバイスおよびそのリリース上でテストしなければなりません。さまざまな種類のデバイス上でテストし、想定どおりに動作することを確認します。アプリケーションのテストが終了したら、そのアプリケーションのベータ版をテスト担当者に配布します。ベータ版のアプリケーションを自分で配布するか、iTunes Connectを利用してベータテストを管理することができます。TestFlightやストアを通して配布するアプリケーションに関して、Appleは、クラッシュログを収集、集約するサービスを提供しています。開発者はこのログを、Xcodeでダウンロードし、分析できます。Macアプリケーションのクラッシュレポートも、Mac App Storeを通して入手可能です。

関連する章: [“テスト用にアプリケーションをエクスポート \(iOS、tvOS、watchOS\)”](#) (102 ページ)、[“クラッシュレポートの分析”](#) (132 ページ)

ストアでのアプリケーションの登録と公開

アプリケーションをストアに登録する手続きは、いくつかの手順を踏んで行います。まず、iTunes Connectにサインインし、アプリケーションレコードの作成と必要な情報の入力を行います。ストアを通して販売する場合は、iTunes Connect上で、代金の精算に必要な事項も入力してください。Xcodeで、アーカイブを作成し、配布用証明書を使って署名します。次に、XcodeまたはApplication Loaderを使ってアプリケーションをアップロードします。iTunes Connectを使って、アプリケーションをストアに登録します。承認が得られたら、iTunes Connectを使って公開開始日を設定してください。

関連する章: [“アプリケーションの登録”](#) (149 ページ)、[“iTunes Connectでのアプリケーション管理”](#) (152 ページ)、[“Apple Developer Enterprise Programアプリケーションの配布”](#) (232 ページ)、[“Mac App Storeを通さずにアプリケーションを配布する手順”](#) (241 ページ)

ストアを通さずにアプリケーションを配布する

Apple Developer Enterprise Programに加入すると、社内向けのアプリケーションを従業員に直接配布することもできます。Mac App Storeを通さずにMacアプリケーションを配布するには、そのアプリケーションを作成し、Developer ID証明書で署名します。なお、ストアを通さずに配布する場合の手順は若干異なります。iTunes Connectと一部のアプリケーションサービスを利用する権限がなければ、この手順を省略しても構いません。

関連する章: [“Apple Developer Enterprise Programアプリケーションの配布”](#) (232 ページ)、[“Mac App Storeを通さずにアプリケーションを配布する手順”](#) (241 ページ)

証明書、ID、プロファイルを管理する

Appleは、ユーザデータやアプリケーションが、知らない間に改竄、配布されてしまうことのないよう、基盤セキュリティモデルを構築しています。開発に当たっては、所定のアセットを作成して必要な情報を入力し、開発者自身や開発に用いるデバイス、あるいは開発されたアプリケーションを、Appleが識別できるようにしなければなりません。Xcodeは必要に応じて自動的に、さまざまな証明書やID、プロファイルを生成するようになっています。また、生成したApp IDやプロビジョニングプロファイルの管理も行いますが、他のアセットは対象外です。Apple Developer Programの会員である期間中、上記以外の証明書やID、プロファイルは、自分で管理しなければなりません。

関連する章: [“署名IDや証明書の管理”](#) (168 ページ)、[“ID、デバイス、プロファイルの管理”](#) (195 ページ)

この文書の使い方

この資料の読み進め方は、Apple Developer Programにおける自分の役割（チームエージェント、チーム管理者、メンバー）に応じて判断してください。Mac用のアプリケーションを開発する場合、他のプラットフォームの場合とは役割が異なるかも知れません。Macアプリケーションの場合は、Mac App Storeに登録して配布するか、Mac App Storeを通さずに配布するかによっても異なります。

まず、アカウントの種類（個人/組織）を選択します。必要ならばApple IDを生成し、Apple Developer Programに加入してください（“[アカウントの管理](#)”（24 ページ）を参照）。個人として加入した人は、1人だけから成るチームの「チームエージェント」になります。一方、組織として加入すれば、「チームエージェント」としてほかの人をチームに加えることができます（“[チームメンバーを加える](#)”（162 ページ）を参照）。さらに、それぞれの人をチーム管理者（チームエージェントの作業の大部分を代行可能）にするか、チームメンバー（メンバーセンターでアセットを生成する作業は不可）にするかを指定できます。チームにおける各人の役割について詳しくは、“[Apple Developer Programにおけるチームの役割と権限について](#)”（160 ページ）を参照してください。

加入したApple Developer Programおよびチームにおける役割に応じ、この節の該当する表で、資料の読み進め方を確認してください（この資料中に現れる用語の意味については「用語解説」を参照）。

チームエージェントまたはチーム管理者であり、アプリケーションをストアに登録する場合：

	知りたい事項	読むべき箇所
<input type="checkbox"/>	自分のApple IDをXcodeに追加する	“ Xcodeで自分のApple IDアカウントを追加する ”（26 ページ）
<input type="checkbox"/>	バンドルIDを設定し、プロジェクトをチームに対応づける	“ IDおよびチームの設定をする ”（33 ページ） “ アプリケーションを配布するためのXcodeプロジェクトの設定 ”（30 ページ）
<input type="checkbox"/>	アプリケーションサービスを設定する	“ 技術やサービスを組み込む ”（56 ページ）
<input type="checkbox"/>	デバイス上でアプリケーションを起動する	“ デバイス上でアプリケーションを起動する手順 ”（91 ページ）
<input type="checkbox"/>	配布に先立ち、最後の設定作業を行う	“ アプリケーションを配布するためのXcodeプロジェクトの設定 ”（30 ページ）
<input type="checkbox"/>	各種のデバイス上でiOS/tvOS/watchOSアプリケーションをテストする	“ テスト用にアプリケーションをエクスポート（iOS、tvOS、watchOS） ”（102 ページ）
<input type="checkbox"/>	テストで見つかった問題点を解消する	“ クラッシュレポートの分析 ”（132 ページ）
<input type="checkbox"/>	アプリケーションをiTunes Connectにアップロードして認定を求める	“ アプリケーションの登録 ”（149 ページ）
<input type="checkbox"/>	アプリケーションをストアに公開し、管理する	“ iTunes Connectでのアプリケーション管理 ”（152 ページ）

	知りたい事項	読むべき箇所
<input type="checkbox"/>	Apple Developer Programのアセットを管理する	“署名IDや証明書の管理” (168 ページ) “ID、デバイス、プロファイルの管理” (195 ページ)
<input type="checkbox"/>	コード署名アセットに関する問題に対処する	“トラブルシューティング” (251 ページ)

組織に属するチームエージェントまたはチーム管理者である場合：

	知りたい事項	読むべき箇所
<input type="checkbox"/>	組織としてのアカウントにチームメンバーを加え、役割を与える	“メンバーセンターでのチームの管理” (160 ページ)

組織のチームメンバーであり、ストアを通して配布するアプリケーションを開発する場合：

	知りたい事項	読むべき箇所
<input type="checkbox"/>	自分のApple IDをXcodeに追加する	“Xcodeで自分のApple IDアカウントを追加する” (26 ページ)
<input type="checkbox"/>	バンドルIDを設定し、プロジェクトをチームに対応づける	“IDおよびチームの設定をする” (33 ページ) “アプリケーションを配布するためのXcodeプロジェクトの設定” (30 ページ)
<input type="checkbox"/>	デバイス上でアプリケーションを起動する	“デバイス上でアプリケーションを起動する手順” (91 ページ)
<input type="checkbox"/>	コード署名アセットに関する問題に対処する	“トラブルシューティング” (251 ページ)

Apple Developer Enterprise Programでのチームエージェントまたはチーム管理者の場合：

	知りたい事項	読むべき箇所
<input type="checkbox"/>	証明書を管理し、アプリケーションを配布する	“Apple Developer Enterprise Programアプリケーションの配布” (232 ページ)

チームエージェントであって、Mac App Storeを通さずにMacアプリケーションを配布する場合：

	知りたい事項	読むべき箇所
<input type="checkbox"/>	配布に先立ち、最後の設定作業を行う	“アプリケーションを配布するためのXcodeプロジェクトの設定” (30 ページ)
<input type="checkbox"/>	Developer IDで署名したアプリケーションを作成する	“Mac App Storeを通さずにアプリケーションを配布する手順” (241 ページ)
<input type="checkbox"/>	追加のDeveloper ID証明書を作成する	“追加のDeveloper ID証明書を作成する” (177 ページ)

Macアプリケーションで、配布方法として「None」を選択した場合 ([“署名IDを選択する \(Mac\)”](#) (37 ページ) を参照)、このガイドを読む必要はありません。

関連項目

このガイドは、コードの記述に用いるソフトウェアやツールに読み手が習熟していると想定していません。そうでない場合は、プラットフォームに応じたチュートリアルをまず読んでください。その後、各種技術の概要を説明した資料、プラットフォームに応じたヒューマンインターフェイスガイドライン、そしてアプリケーションをストアに登録するためのガイドライン (これが最も重要) に目を通しておくといでしょう。

	iOS、tvOS、watchOS	Mac
手始めに...	Start Developing iOS Apps Today App Distribution Quick Start	Start Developing Mac Apps Today App Distribution Quick Start
各種技術についてより詳しく...	iOS Technology Overview iOS App Programming Guide App Programming Guide for tvOS WatchKit Programming Guide	Mac Technology Overview Mac App Programming Guide
ユーザインターフェイスガイドラインについて...	iOS Human Interface Guidelines App Store Review Guidelines for iOS Apps Apple TV Human Interface Guidelines Apple Watch Human Interface Guidelines	OS X Human Interface Guidelines App Store Review Guidelines for Mac Apps

	iOS、tvOS、watchOS	Mac
ツールについてより詳しく...	<i>Xcode Overview</i> <i>iTunes Connect Developer Guide</i> <i>iOS Simulator User Guide</i> <i>Using Application Loader</i>	<i>Xcode Overview</i> <i>iTunes Connect Developer Guide</i> <i>Using Application Loader</i>
大規模なチームで使用するツールについて	<i>Xcode Continuous Integration Guide</i> <i>Testing with Xcode</i> <i>Source Control Management Help</i>	<i>Xcode Continuous Integration Guide</i> <i>Testing with Xcode</i> <i>Source Control Management Help</i>

アプリケーション審査プロセスについて詳しくは、「[App Review](#)」で確認してください。

アプリケーションの間引き (iOS、tvOS、watchOS)

ストアおよびオペレーティングシステムには、インストール先に合わせてiOS/tvOS/watchOSアプリケーションを最適化するため、デバイスの能力に応じて取舍選択し、配布する容量を最小限に抑える働きがあります。この最適化を**アプリケーションの間引き (thinning)**と言います。デバイスの能力をできるだけ活かしつつ、占有ディスク容量を最小限に抑え、さらに将来、Appleが更新を適用する際に便宜を図ります。ダウンロード時間が短くなり、他のアプリケーションなどに割り当てるディスク容量が増えるので、ユーザにとってはよいことづくめです。

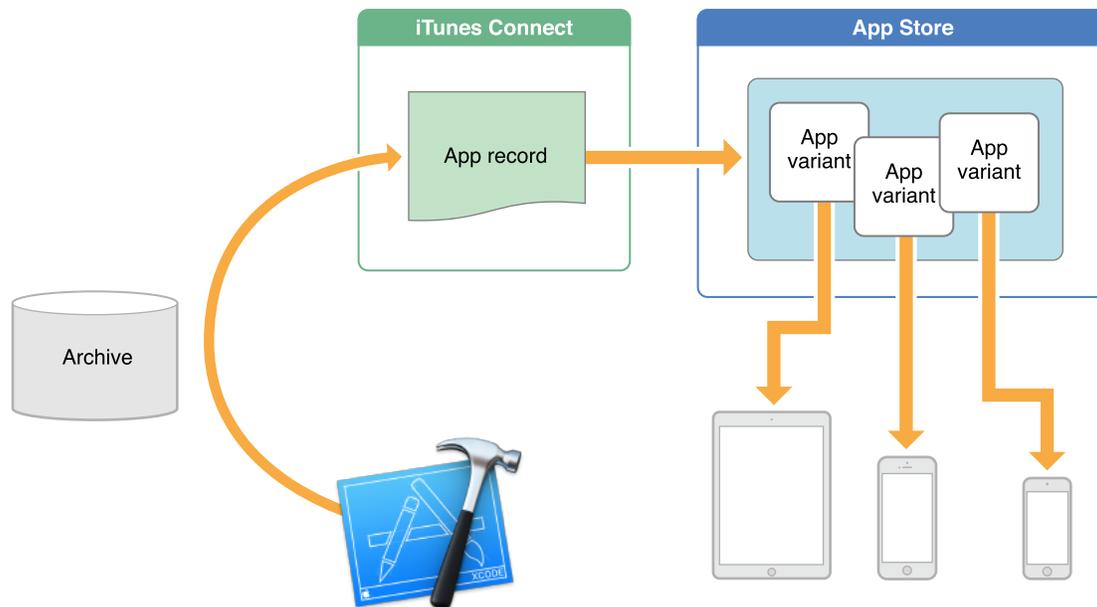
この章では「間引き」を実現する技術を3つ、すなわち、切り分け、ビットコード、オンデマンド型リソースを紹介します。

切り分け (iOS、tvOS)

切り分け (slicing)とは、個々のデバイスに合わせてアプリケーションバンドルの「変種」を生成し、配布することです。それぞれの**変種 (variant)**は、実行アーキテクチャやリソースのうち、対象デバイスに必要なものだけを収容しています。開発者は従来どおり、アプリケーションの完全版を開発し、iTunes Connectにアップロードします。ストアはサポート対象のデバイスごとに変種を生成し、配布します。画像リソースは解像度やデバイスファミリーごとに切り分けします。GPUリソースはデバイスのケイパビリティごとに切り分けします。tvOSアプリケーションの場合、ターゲット (iOSとtvOS) 間で共有するカタログ内のアセットを切り分けし、大きなアプリケーションアイコンを削除します。ユーザはインストールの際、そのデバイスに応じた変種をダウンロードすることになります。

Xcodeには切り分けのシミュレーション機能があるので、開発の際、変種をローカルに生成し、テストできます。あるデバイス上でビルド、実行する時点で、切り分けをおこなうようになっています。アーカイブを作成すれば、そこに収容されるのはアプリケーションの完全版ですが、ここからいつでも必要な変種をエクスポートできます。

注意: 切り分けが可能なのは、iOS 9.0以降が動作するデバイスのみです。それ以外の場合、ストアはユニバーサルアプリケーションを配布します。



切り分けは通常の開発および配布のワークフロー内で実施します。大まかな手順を以下に示します。

1. Xcode上で、対象デバイスを指定し、さまざまな解像度の画像をアセットカタログに収容してください。
リソースを切り分けするためには、アセットカタログが不可欠です。
2. アプリケーションをビルドし、シミュレータ上、あるいはデバイス上で実行してください。
Xcodeはデバイスの種類に応じて変種をビルドします。デバッグ用にビルドする時間を短縮するとともに、ローカルに変種をテストできるようにするのが目的です。
3. アプリケーションのアーカイブを生成し、対象デバイスに応じた変種を、ローカルにエクスポートしてください。
エクスポートした変種をすべて、該当するデバイス上でテストし、設定に問題があれば早いうちに見つけて対処します。
4. このアーカイブをiTunes Connectにアップロードしてください。
ストアはアーカイブから個々の変種を生成します。実際に生成される変種の個数は、Xcodeプロジェクトで指定したアーキテクチャリソースによって異なります。
5. iTunes Connectを通して、プレリリース版を所定のテスト担当者に配布してください。

テスト担当者はTestFlightを使って、サポート対象のデバイスにインストールします。TestFlightはデバイスに応じた変種をダウンロードするようになっています。

注意: ストアがビルドする変種を、ユーザーに配布する前にテストしたい場合は、内部のテスト担当者（チームに属するiTunes Connectユーザー）のみを募り、TestFlightを使って変種をダウンロードするよう指示してください。外部のテスト担当者（電子メールアドレスのみ指定するユーザ）を募る場合、Beta App Reviewによる承認が得られるまでは、変種をダウンロードできるようになりません。

6. iTunes Connectを通してアプリケーションを公開してください。

ユーザーがサポート対象のデバイスにインストールしようとする、ストアは当該デバイス用の変種がダウンロードされるよう制御します。

ビットコード

ビットコードとは、コンパイル済みプログラムの中間表現のことです。iTunes Connectにアップロードしたアプリケーションに、ビットコードが収容されていれば、ストアはこれをコンパイル、リンクして、バイナリを生成できます。ビットコードがあれば、Appleは今後いつでも、バイナリの最適化をやり直すことができます。開発者が新しい版を用意し、ストアに登録する必要はありません。

Xcodeは通常、ビルド時に生成したシンボルを消去してしまうので、Apple側でこれを読み取ることはできません。iTunes Connectにアップロードする際、明示的に指定した場合に限り、シンボル情報もAppleに送信されます。クラッシュレポートをAppleから受け取って役立てたい場合は、このように指定してください。

注意: iOSアプリケーションの場合、デフォルト設定のままであればビットコードが生成されますが、必須ではありません。watchOS/tvOSアプリケーションの場合、ビットコードは必須です。生成する場合は、アプリケーションバンドル内のアプリケーションやフレームワーク（プロジェクトのターゲットすべて）について生成する必要があります。

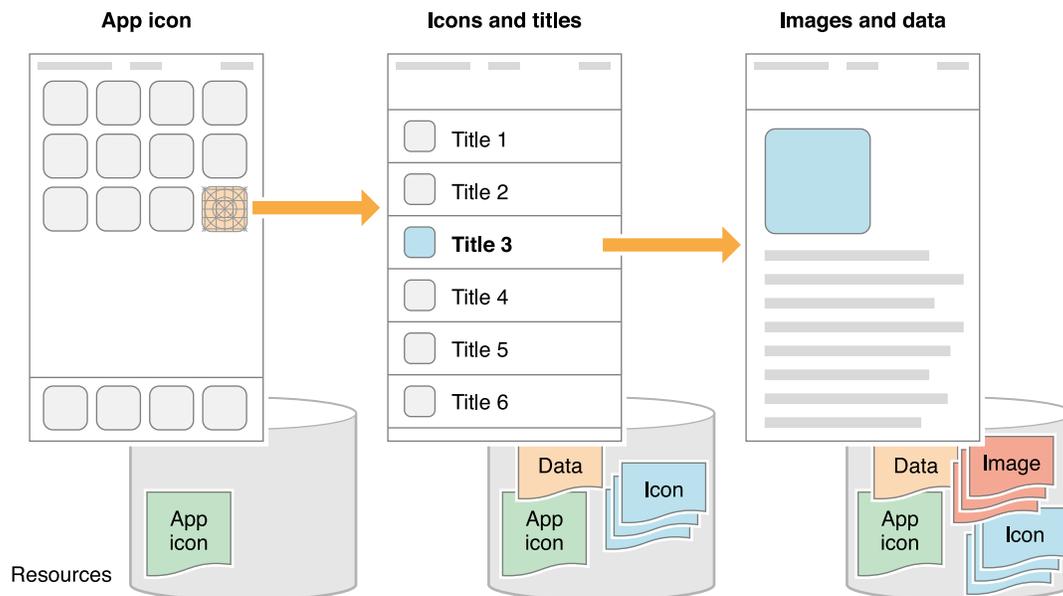
iTunes Connectを使ってアプリケーションを配布した後、ビルドに対応するdSYMファイルをダウンロードできます（[“クラッシュデータを「Devices」ウィンドウに表示し、インポートする”](#)（145 ページ））を参照）。

オンデマンド型リソース (iOS、tvOS)

オンデマンド型リソース (On-demand resources) とは、画像、音声などといったリソースのうち、キーワードでタグ付けしておき、グループ内でタグを使って要求できるようにしたものを言います。ストアは開発者に代わり、要求に応じてリソースを配布するAppleサーバを運営し、ダウンロード状況を管理します。アプリケーション本体のダウンロード時間や容量を減らす効果があり、ユーザにとって、初めて使う際の印象がよくなります。たとえばゲームで、レベルごとにリソースを分割する、という使い方が考えられます。ゲームが進行し、レベルが上がりそうになった時点で、該当するリソースを要求すればよいのです。同様に、In-App Purchaseのリソースを、ユーザが該当する商品を購入してから要求する、という使い方も考えられます。

オペレーティングシステムは、空きディスク容量が少なくなったとき、不要になったオンデマンド型リソースを消去します。テストのため、あるいはストアを通さずに配布するため、アプリケーションをエクスポートした場合は、オンデマンド型リソースを配布するサーバを、開発者自身が運営しなければなりません。なお、実行ファイルをオンデマンド型リソースとして扱うことはできません。ストアは、さらにユーザーの印象をよくするため、オンデマンド型リソースを切り分けることもあります（[“切り分け \(iOS、tvOS\)”](#)（19 ページ）を参照）。

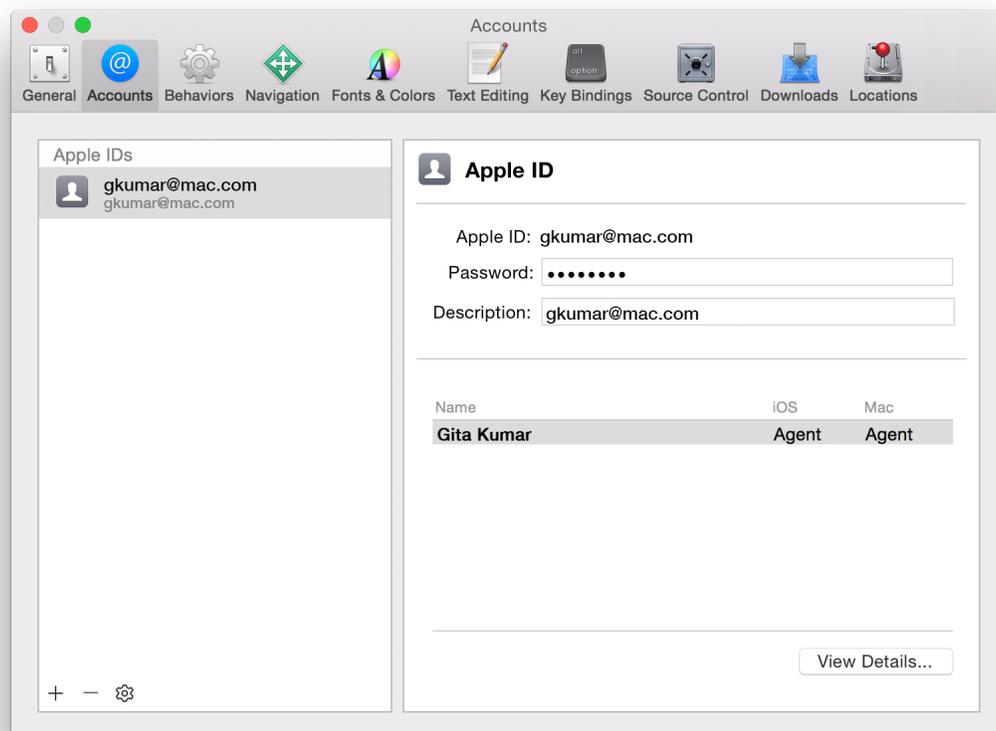
ユーザの側からは、オンデマンド型か否か区別できません。アプリケーションのさまざまな機能を渡り歩く過程で、必要なリソースがどこからか現れるように見えます。



オンデマンド型リソースをアプリケーションに取り入れる手順については、『*On Demand Resources Guide*』および『*NSBundleResourceRequest Class Reference*』を参照してください。

アカウントの管理

「Accounts」環境設定ペインでは、コード署名アセットの管理に用いるApple IDをはじめ、プロジェクトで用いるアカウントを集中管理できます。「Apple ID」アカウントを追加し、必要ならばApple Developer Programに加入し、プロジェクトをチームに対応づけることにより、証明書、ID、プロフィールをXcodeで管理できるようになります。Apple IDがない場合は、無料で新たに作成できます。ただし、Apple Developer Programに加入しなければ利用できないケイパビリティがいくつかあります（「サポートされるケイパビリティ」（258ページ））。この章では、Apple IDアカウントの管理方法を説明します。



Apple Developer Programについて

Apple Developer Programは、App StoreやApple TV App Store、Mac App Storeで配布するApp、およびMac App Store以外で配布するDeveloper IDで署名されたAppを開発する、個人/組織のお客様向けのプログラムです。Safari用拡張機能の配布手続きの支援もおこないます。**Apple Developer Enterprise Program**は、従業員を対象とした独自の社内向けAppを開発および配布する、組織のお客様向けのプログラムです。

技術資料や技術支援など、革新的なアプリケーション、Safari用拡張機能、各種デバイス用アクセサリの開発に必要な、あらゆるものが手に入ります。Apple Developer Programに加入すると、メンバーセンターおよびiTunes Connectのあらゆる機能が利用できるようになります。Apple Developer Enterprise Programに加入した場合、アプリケーションを従業員に配布することはできますが、ストアへの登録ができないため、メンバーセンターを利用することしかできません。

Apple Developer Programについて詳しくは、ウェブサイト「[Apple Developer Program](#)」を参照してください。Apple Developer Enterprise Programに加入するには、「[Apple Developer Enterprise Program](#)」にアクセスした後、「[Apple Developer Enterprise Programアプリケーションの配布](#)」（232 ページ）に目を通してください。

個人として、あるいは組織として加入する

加入手続きの際、個人または組織のどちらとして参加するか決める必要があります。**個人**として加入した場合、1人だけのチームと看做され、以下に説明する作業（複数のチームメンバーの管理を除く）をすべて自分で行うこととなります。

加入手続きの際、氏名、住所などの基本的な個人情報を訊ねられます。なお、組織として登録するのであれば、法人名やDUNS番号（Data Universal Numbering System、企業識別コード）などの情報も必要です。入力された情報の検証後、ライセンス契約を確認し、Apple Online Storeで代金を支払い、会員資格の詳しいアクティベーション手順を受け取ることとなります。

一方、**組織**として加入すれば、他の開発者をチームに加え、アカウント管理の権限を委ねることができます。各チームメンバーには、Appleが確認済みの、他とは異なる電子メールアドレスと結びついた、**Apple ID**が必要です。メンバーセンターやiTunes Connectには、このApple IDを使ってサインインします。また、メンバーはそれぞれ権限が異なるので、役割にもよりますが、この資料で説明している作業をすべて実行できるとは限りません。

個々の役割や権限については「[Apple Developer Programにおけるチームの役割と権限について](#)」（160 ページ）を参照してください。

同じ人が複数のチームに加わることもできる

Apple IDを使って複数のチームに参加することも可能ですが、いくつか制限があります。

同じApple IDに、複数のApple Developer Programアカウントを対応づけることができます。同じApple IDで、個人として加入する、あるいは他のチームに参加することも可能です。しかし、Apple IDに複数のiTunes Connectアカウントを対応づけることはできません。したがって、iTunes Connectで独立に管理する他のアカウントに対しては、別にApple IDを用意する必要があります。

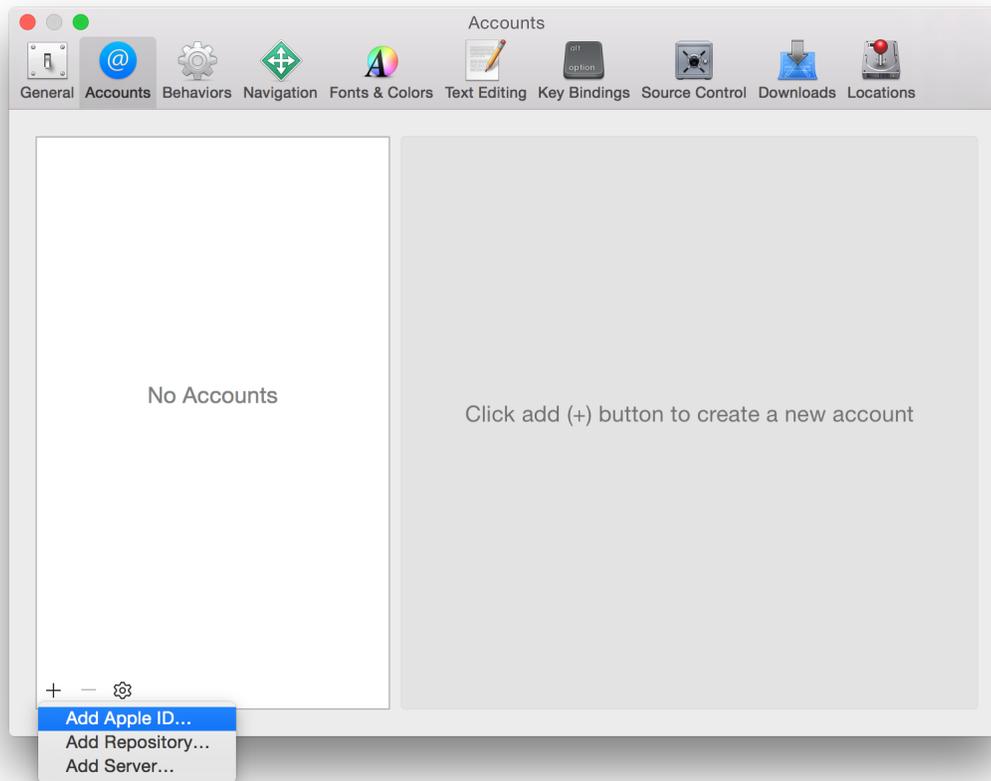
Xcodeで自分のApple IDアカウントを追加する

まず、Xcodeの「Accounts」環境設定ペインでApple IDを追加します。Apple Developer Programには、必要ならばXcodeから直接加入できます。Apple Developer Programに加入しなくても、デバイス上でアプリケーションを実行することは可能です。ただし、加入しなければ利用できない権限がいくつかあります。また、iTunes Connectユーザは、Apple IDを入力することにより、クラッシュレポートを参照できます（[“クラッシュレポートの分析”](#)（132ページ）を参照）。Apple IDアカウントはいくつでも追加できます。

Apple IDアカウントを追加するには

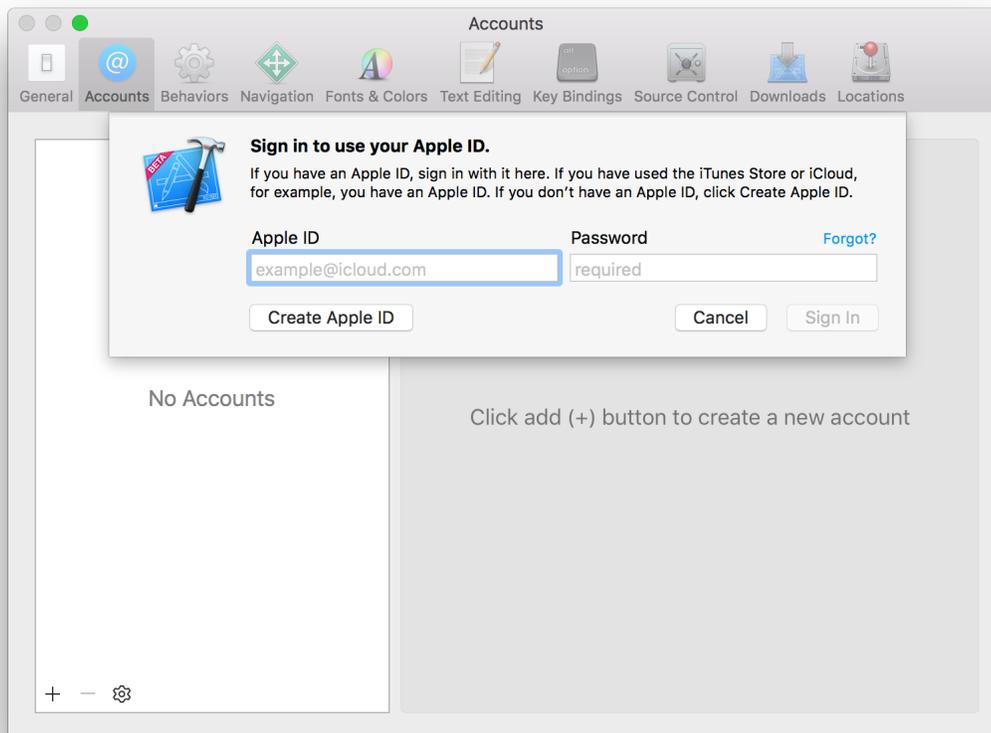
1. 「Xcode」 > 「Preferences」を実行して環境設定を起動します。
2. ウィンドウ上部の「Accounts」を押して画面を切り替えます。
3. 左下隅の「Add」ボタン（「+」の表示）を押してください。

4. ポップアップメニューから「Add Apple ID」コマンドを実行します。



5. すると現れるダイアログにApple IDとパスワードを入力し、「Sign in」を押してください。

OS X 10.11以降、Apple IDの2ステップ検証を有効にしていれば、さらに検証コードを入力しなければならないかも知れません（詳しくは「[Frequently asked questions about two-step verification for Apple ID](#)」を参照）。これ以前のOSであれば、アプリケーションに応じたパスワードを入力しなければならないことがあります（このパスワードの生成方法については「[Using app-specific passwords](#)」を参照）。



6. Apple IDがなければ、ダイアログの左下にある「Create Apple ID」を押してください。
Apple IDの作成後、改めて上記の手順を進めます。

Apple Developer Programの会員でなければ、チームテーブルには「個人」として現れ、プラットフォーム欄は「Free」となります。会員であれば属するチームがこのテーブルに現れ、その役割がプラットフォーム欄に表示されます。個人として加入した場合、役割は「エージェント」となります。

Apple IDアカウントを削除するには

1. 「Xcode」 > 「Preferences」を実行して環境設定を起動します。
2. ウィンドウ上部の「Accounts」を押して画面を切り替えます。
3. 削除するApple IDアカウントを左側の列から選択します。
4. 左下隅の「Delete」ボタン（「-」の表示）を押してください。

まとめ

この章では、Apple IDアカウントを追加する方法、Apple Developer Programに加入する方法を説明しました。続いて、Xcodeプロジェクトをいずれかのチームに対応づけます。Xcodeはこの情報に基づき、証明書、ID、プロファイルを生成します。

アプリケーションを配布するためのXcodeプロジェクトの設定

プロジェクト設定はいつでも編集できますが、その中には、開発時に必ず設定しなければならない事項と、ベータ版の配布時点では推奨にとどまり、ストアに登録する際に必須となる事項があります。

開発の際、アプリケーションにプロビジョニングやコード署名を施さないと、組み込んで実行することができないアプリケーションサービスがいくつかあります。バンドルIDやチームをプロジェクトに設定すると、Xcodeは自動的にメンバーセンターに接続し、必要な証明書、ID、プロファイルを生成できるようになります。この情報は、プロジェクトエディタで事前に入力しても、アプリケーションサービスを追加し、アプリケーションを起動する時点で、必要に応じて入力しても構いません。

アプリケーションをテスト用に配布し、あるいはストアに登録するためには、必要な情報をすべて入力しておかなければなりません。たとえば、アプリケーションのアイコンがないと、iTunes Connectの検証テストに合格することができません。

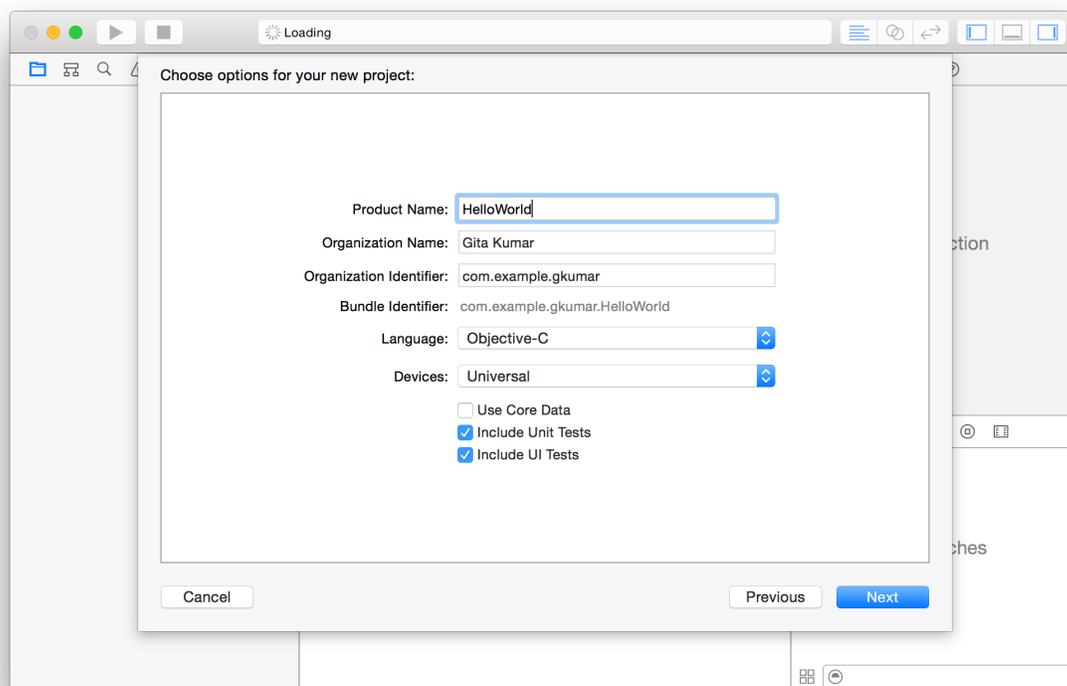
アプリケーションをiTunes Connectにアップロードする前に、ビルド設定を確認し、Macアプリケーションの場合は「Copyright」キーを設定してください。

Xcodeプロジェクトの作成時にプロパティを設定する

Xcodeプロジェクトの作成は、アシスタント機能により表示される指示に従って進めてください。まず、プロジェクトに適用するテンプレートを選択します。適切なテンプレートを使えば、開発作業が迅速に進みます。アシスタント機能の指示に従い、アプリケーションに関する情報を入力していきます。その能力を決めるとともに、顧客に配布するために必要な情報です。プロジェクト作成時に決まっていない事項は、後で設定しても構いません。Apple Developer Programの会員の場合、Xcodeプロジェクトで設定するデータの中には、iTunes Connectに入力するデータと似ているものもありますが、バンドルID以外は、一致していなくてもアプリケーションをiTunes Connectにアップロードできます。

注意: Xcodeプロジェクトがなければ、『*Start Developing iOS Apps Today*』または『*Start Developing Mac Apps Today*』のチュートリアルを読んで、作成しておいてください。この資料で説明する作業を行うためには、アプリケーションをビルド、実行するための場である、Xcodeプロジェクトをまず作成しなければなりません。

iOSアプリケーションの場合、テンプレートをもとにXcodeプロジェクトを作成しようとすると、次のようなダイアログが現れます。



「**Product Name**」はアプリケーション名で、顧客に対してはこの名前が表示されることとなります。後でiTunes Connectに入力するアプリケーション名とおおむね一致していなければなりません。最も重要なのは、ストアに表示されているアプリケーション名やアイコンが、デバイスにインストールすると現れるプロジェクト名やアイコンと対応していることを、ユーザーが即座に認識できることです。アプリケーションをインストールすると、Springboardにもこの名前が現れます。2文字以上255バイト以下で指定してください（アプリケーション名のつけ方に関するガイドラインが、『*iTunes Connect Developer Guide*』の“First Steps: Identifying Your App in iTunes Connect”に載っています）。

「**Organization Name**」はXcodeプロジェクトの属性で、プロジェクトフォルダの各所、たとえばソースファイルやヘッダファイルの著作権表示に現れます。後でiTunes Connectに入力する組織名と同じでなくても構いません。

ここに入力した「Product Name」と「**Organization Identifier**」を逆DNS（逆ドメイン名サービス）方式で連結したものが、デフォルトのバンドルIDになります。**バンドルID**にはアプリケーションを一意に特定する役割があるので、「Organization Identifier」も一意的な文字列にすることが大切です。

iOSアプリケーションであれば、動作するデバイスの種類を、「Devices」ポップアップメニューから選択します。Macアプリケーションの場合はこのメニューから「Mac App Store」というカテゴリを選択してください。

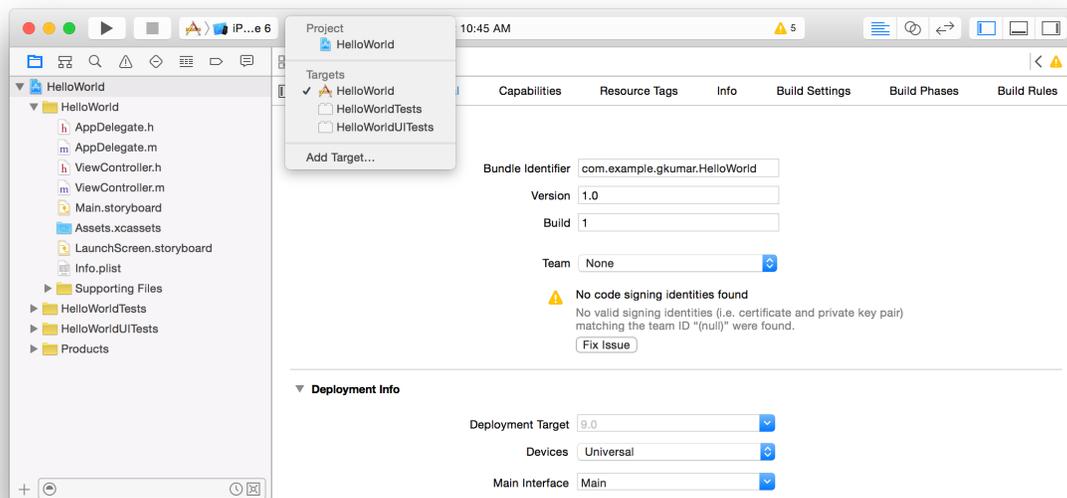
Xcodeテンプレートに設定されている他の値は、アプリケーションをビルドしてローカルに実行するだけであればそのまま構いませんが、特にバンドルIDのように、後で見直すことになるものもあります。さらに、ストアに関するプロパティは、アシスタント機能の対象外です。登録するまでに、基本的なストア設定を済ませておいてください。実際にはテスト用に配布する時点で済ませておくといでしょう。

アプリケーションを公開すると、こういったメタデータのいくつかは変更できなくなるので、慎重に設定してください。ある状態になったとき、iTunes Connectでロックされる（変更できなくなる）プロパティについては、『*iTunes Connect Developer Guide*』の“iTunes Connect App Properties”を参照してください。

プロジェクトの設定を始める前に

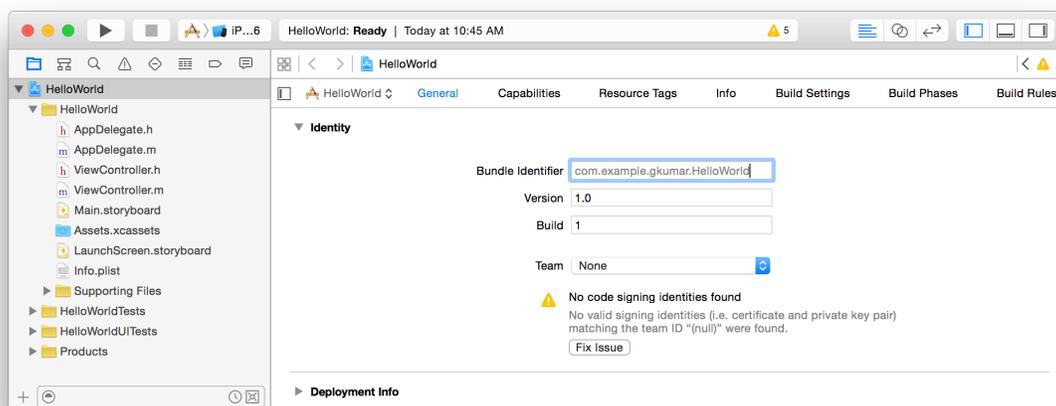
この章で説明する操作はすべて、ターゲットに対応するプロジェクトエディタの「General」ペインで行います。プロジェクトナビゲータを開きましょう。「View」>「Navigators」>「Show Project Navigator」コマンドを実行してください。次にターゲットを「Project/Targets」ポップアップメニューから選択します。2つめのサイドバーが現れていれば、その「Targets」セクションで選択しても構いません。「General」をクリックすると、この章で説明する設定事項が現れます。

iOSアプリケーション用の「General」ペインの画面例を示します。



IDおよびチームの設定をする

Xcodeでチームプロビジョニングプロファイルを生成するためには、アプリケーションのバンドルIDが一意的であり、かつ、プロジェクトにチームが設定されている必要があります。後で、アプリケーションの特定の版を識別するために、ほかにもいくつか情報を設定します。IDに関する設定事項は、ターゲットの「General」ペイン内、「Identity」セクションにあります。iOSアプリケーションの「Identity」セクションの様子を以下に示します。

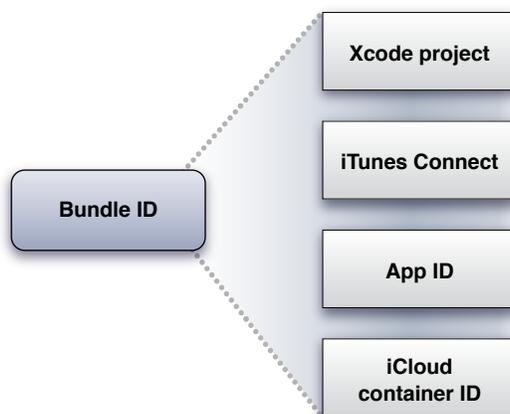


バンドルIDについて

バンドルIDは、特定のアプリケーションを明確に識別するために使います。開発期間中はデバイスのプロビジョニングのために必要であり、顧客に配布した後も、オペレーティングシステムが参照します。たとえば、該当するアプリケーションサービスを利用する際、Game CenterやIn-App PurchaseはバンドルIDでアプリケーションを特定します。また、環境設定システムはこの文字列を使って、設定をどのアプリケーションに適用するか識別します。同様にLaunch Servicesは、あるファイルを開くために使うアプリケーションの検索にバンドルIDを利用します。IDに合致する、最初に見つかったアプリケーションで開くのです。さらに、アプリケーションの署名を検証するためにも必要です。

バンドルIDの文字列は、英数字 (A-Z、a-z、0-9)、ハイフン (-)、ピリオド (.) のみで構成されるUTI (Uniform Type Identifier) でなければなりません。全体はいわゆる逆DNS形式の文字列です。たとえば、組織のドメインがAcme.comで、Helloという名前のアプリケーションを作成する場合、バンドルIDは「com.Acme.Hello」という文字列になります。

開発中はさまざまなところで、バンドルIDを使ってアプリケーションを識別します。



バンドルIDの具体的な保存場所と用途を以下に示します。

- Xcodeプロジェクトでは、バンドルIDを「情報プロパティリスト」ファイル (Info.plist) に保存します。このファイルは、ビルドの際、アプリケーションバンドルにコピーされます。
- メンバーセンターでは、アプリケーションのバンドルIDに合致するApp IDを生成します。これが特定App IDであれば、バンドルIDと完全に一致する文字列です。ただし、ドメイン名と違い、大文字と小文字の区別があります。App IDが小文字のみから成るならば、バンドルIDもそうでなければなりません。
- iCloudの場合、Xcodeプロジェクトで指定するコンテナIDは、アプリケーションのバンドルIDをもとに生成したものでなければなりません。

- iTunes Connectでは、バンドルIDを入力することによりアプリケーションを特定します。アプリケーションの最初のビルドをiTunes Connectにアップロードすると、バンドルIDの変更や関連する特定App IDの削除ができなくなります。

注意: Macアプリケーションが、iOS/tvOS用など他の種類のアプリケーションと同じバンドルIDを共有することはできません。

バンドルIDを設定する

XcodeプロジェクトにおけるデフォルトのバンドルIDは、たとえば「com.MyCompany.MyProductName」のような逆ドメイン形式の文字列です。XcodeはデフォルトのバンドルIDを、組織IDと製品名（テンプレートからプロジェクトを生成する際に入力したもの、「[Xcodeプロジェクトの作成時にプロパティを設定する](#)」（30 ページ）を参照）を連結して生成します（ただし製品名中の空白は別の文字に置換）。デフォルトのバンドルIDのうち、組織IDの部分（プレフィックス）のみ置換すれば通常は十分ですが、全体を置換しても構いません。たとえば、組織のドメイン名に合わせて組織ID部分を変更するか、特定App IDに合わせてバンドルID全体を置換します。

tvOSアプリケーションが、iOSアプリケーションのiTunes Connectアプリケーションレコードを共有する（「[tvOSアプリケーションをiOSアプリケーションレコードに追加する](#)」（156 ページ）を参照）場合、そのバンドルIDはiOSアプリケーションと同じでなければなりません。

watchOSアプリケーションの各ターゲット（埋め込みWatchKitアプリケーション、WatchKitエクステンション）は、バンドルIDのプレフィックスがiOSターゲットと同じでなければなりません。バンドルIDの全体は、WatchKitアプリケーションならば「[Bundle ID].watchkitapp」、WatchKitエクステンションならば「[Bundle ID].watchkitextension」という形になります。iOSアプリケーションのバンドルIDを変更する際には、WatchKitアプリケーションおよびWatchKitエクステンションの方もこれに合わせて変える必要があります。

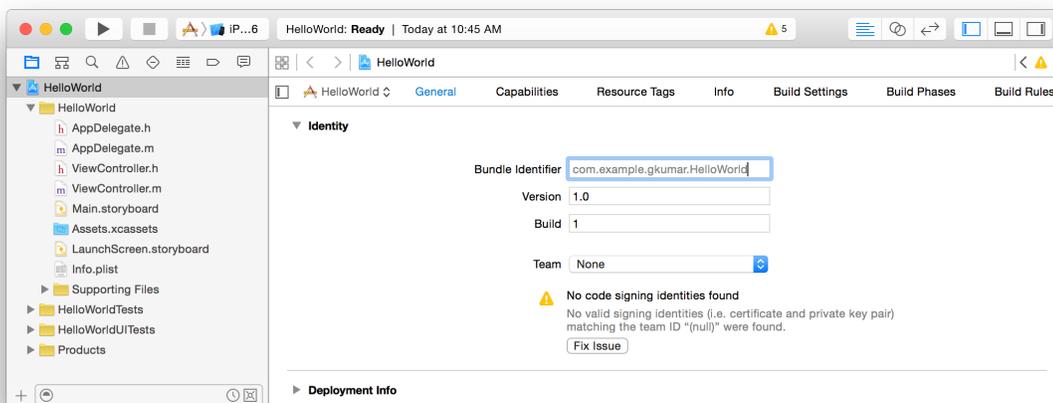
Macアプリケーションの場合、バンドルIDがすべて、アプリケーションバンドル内で一意であることを確認してください。たとえば、アプリケーションバンドルにヘルパアプリケーションを収容する場合、そのバンドルIDは、アプリケーション本体のそれと異なる文字列でなければなりません。

バンドルIDのプレフィックス部分は、プロジェクトエディタの「General」ペインで、以下の手順に従って変更できます。

バンドルIDプレフィックス部分を設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。

2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。

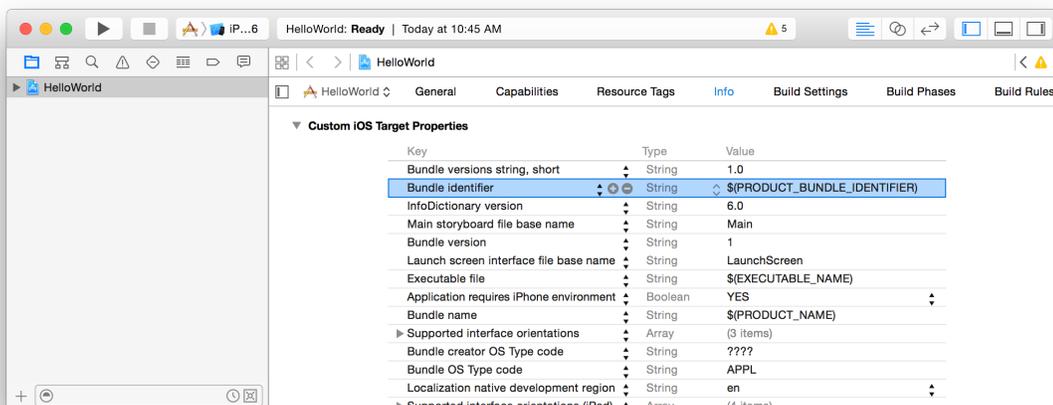


3. 「Bundle Identifier」フィールドに、バンドルIDのプリフィックスを入力します。

あるいは、プロジェクトエディタの「Info」ペインで、バンドルID全体を変更することも可能です。

バンドルIDを設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「Info」を押して画面を切り替えます。
3. バンドルIDを、「Bundle identifier」行の「Value」列に入力します。



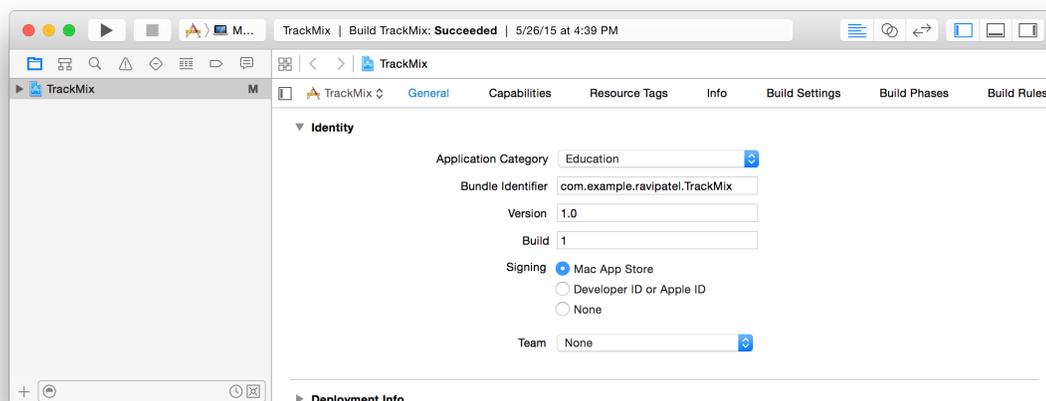
署名IDを選択する (Mac)

Macアプリケーションの配布方法には、ストアに登録する、Developer ID証明書を使って署名を施し、ストアを通さずに配布する、コード署名を施さない、という選択肢があります。「Mac App Store」を選択した場合は、Xcodeプロジェクトをチームに対応づけることになり、アプリケーションサービスを有効にすることができます（「[技術やサービスを組み込む](#)」（56 ページ）を参照）。「Developer ID」を選択した場合も、Xcodeプロジェクトをチームに対応づけることになりますが、利用できないケイパビリティがいくつかあります（Developer IDで署名したアプリケーションの生成方法については「[Mac App Storeを通さずにアプリケーションを配布する手順](#)」（241 ページ）を参照）。「None」を選択すると「Team」ポップアップメニューは開かなくなります。以下、このガイドを読む必要はありません。

署名IDを選択するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。
3. 使おうとする署名IDの種類を選択します。
 - アプリケーションをストアに登録する場合は、「Mac App Store」を選択します。
 - ストアを通さずにMacアプリケーションを配布し、あるいはApple Developer Programの会員ではないApple IDを使いたい場合は、「Developer ID or Apple ID」を選択してください。
 - アプリケーションに署名せず、アプリケーションサービスも利用しない場合は、「None」を選択します。

アプリケーションから利用できるケイパビリティは、ここで選択した署名IDによって決まります（「[サポートされるケイパビリティ](#)」（258 ページ）を参照）。

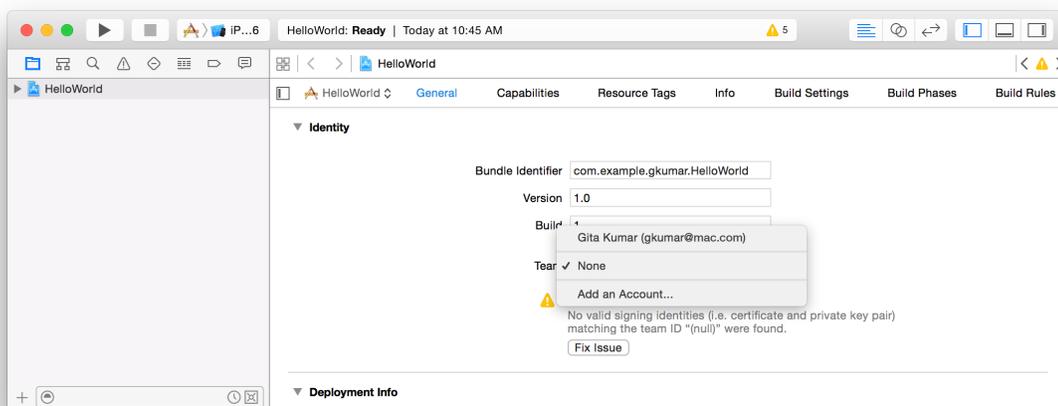


Xcodeプロジェクトをチームに対応づける

個々のXcodeプロジェクトには、ある1つのチームが対応します。個人として加入した人は、1人だけから成るチームとして扱います。チームアカウントには、アプリケーションのプロビジョニングに必要な、証明書、ID、プロファイルを格納します。すべてのアプリケーションにコード署名とプロビジョニングが必要です。これを施さないと、デバイス上で実行し、特定のアプリケーションサービスを利用することができません。Xcodeは必要に応じてこういったアセットを生成しますが、後で警告やダイアログが現れないよう、あらかじめチームを設定しておくといでしょう。

プロジェクトをチームに対応づけるには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。
3. 「Team」ポップアップメニューから該当するチームを選んでください。
 - 個人として加入している場合は、自分の名前を選択します。
 - 組織として加入している場合は、組織名を選択します。

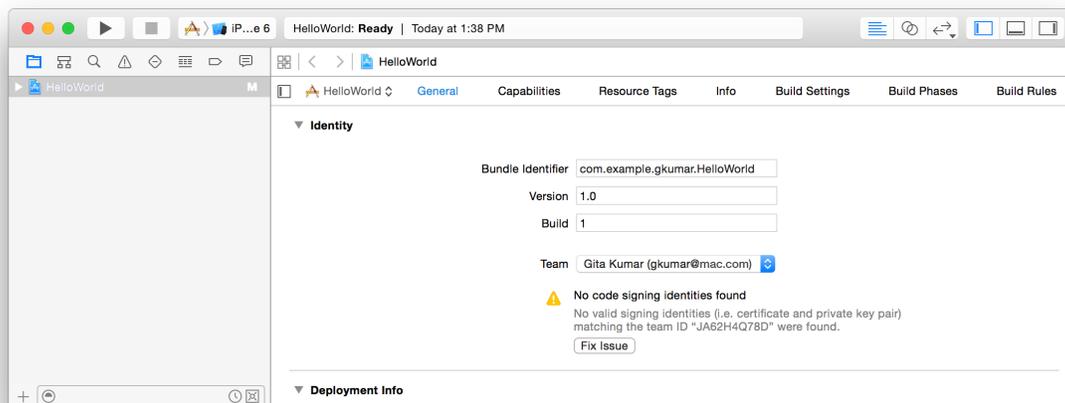


Xcodeはチームプロビジョニングプロファイルの生成を試みる場合があります。iOS/tvOS/watchOSアプリケーションでは、デバイスがMacに接続済みか、または登録済みの場合に限り、Xcodeがチームプロビジョニングプロファイルを生成します。Macアプリケーションの場合、Xcodeは開発用Macを登録します。これに関して問題が生じた場合は、[“チームプロビジョニングプロファイルを生成する”](#)（39 ページ）を参照してください。

4. 「Team」ポップアップメニューに該当するチームが現れない場合は、「Add an Account」コマンドを実行し、[“Xcodeで自分のApple IDアカウントを追加する”](#)（26 ページ）に従って操作してください。

チームプロビジョニングプロファイルを生成する

チームを選択した場合、Xcodeはチームプロビジョニングプロファイルという、特別な開発用プロビジョニングプロファイルを生成、管理します。これを使ってアプリケーションに署名を施すと、チームメンバーは誰でも、あらゆるデバイス上で実行できるようになります。生成に失敗した場合、「Team」ポップアップメニューの下に、警告と「Fix Issue」ボタンが現れます。



チームプロビジョニングプロファイルをあらかじめ生成しておけば、今後さまざまな問題を回避できます。多くの場合、警告メッセージの下に現れる「Fix Issue」を押すだけで、Xcodeが次の処理を実行します。

1. 開発用証明書を作成する

Xcodeは、「Team」ポップアップメニューで選択されたチームの、開発用証明書を作成します。

2. 「Scheme」ツールバーメニューで選択されたデバイスを登録するか、Macを登録する

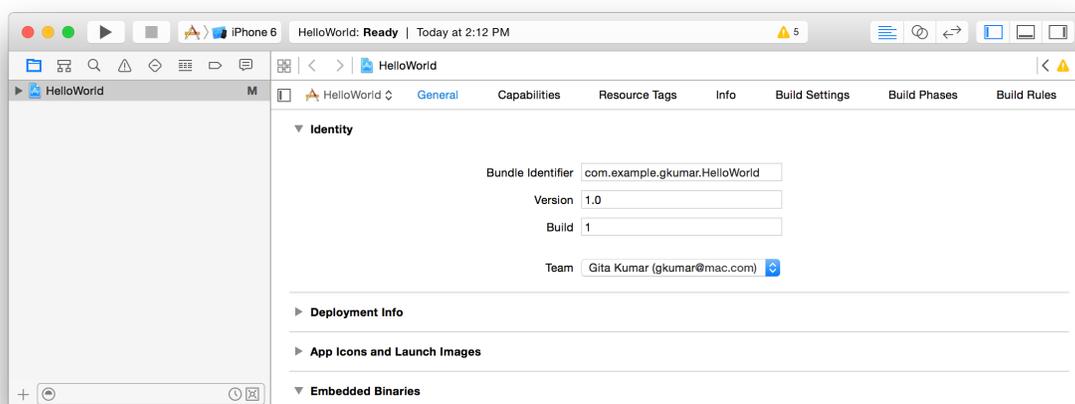
チームプロビジョニングプロファイルを生成するためには、デバイスがいくつか、アカウントに登録されている必要があります。Macアプリケーションであれば、Xcodeが動作しているMacが自動的に登録されます。iOS/tvOSアプリケーションの場合、肺初に使うデバイスを接続してください。watchOSアプリケーションについては、Apple WatchとペアリングしたiPhoneを接続します。デバイスが配備ターゲットと一致していれば、Xcodeがこのデバイスを登録します。デバイスが配備ターゲットと一致していないため、ターゲットとして適切でない場合は、デバイス上のOSをアップグレードするか、配備ターゲットを変更します（[「配備情報を設定する」](#)（43 ページ）を参照）。

3. アプリケーションのバンドルIDに合致するApp IDを生成し、アプリケーションサービスを有効にする

プロジェクト設定によって、XcodeがワイルドカードAppIDを生成する場合と、特定AppIDを生成する場合があります。Xcodeは特定AppIDを登録するためにバンドルIDを使うので、開発期間を通して、バンドルIDも一意的でなければなりません。AppIDを登録する際にエラーが生じないように、一意的なバンドルIDを入力してください（“[バンドルIDを設定する](#)”（35 ページ）を参照）。

4. 以上のアセットを収容するチームプロビジョニングプロファイルを生成する
5. これに従い、コード署名に関するプロジェクトのビルド設定を行う

Xcodeが以上の処理を実行した後、「Team」ポップアップメニューの下に出ている警告メッセージは消えます。



ダイアログや警告が表示された場合は、その内容に応じて問題を解消し、改めて「Fix Issue」ボタンを押します。「Your account already has a valid ... certificate」というダイアログが現れた場合、次のいずれかを実行してください。

- 署名IDがエクスポート済みであれば（“[証明書やプロファイルをエクスポート/インポートする](#)”（180 ページ）を参照）、「Import Developer Profile」を押し、指示に従って操作します。
- 署名IDのバックアップがない場合は「Reset」を押してください。

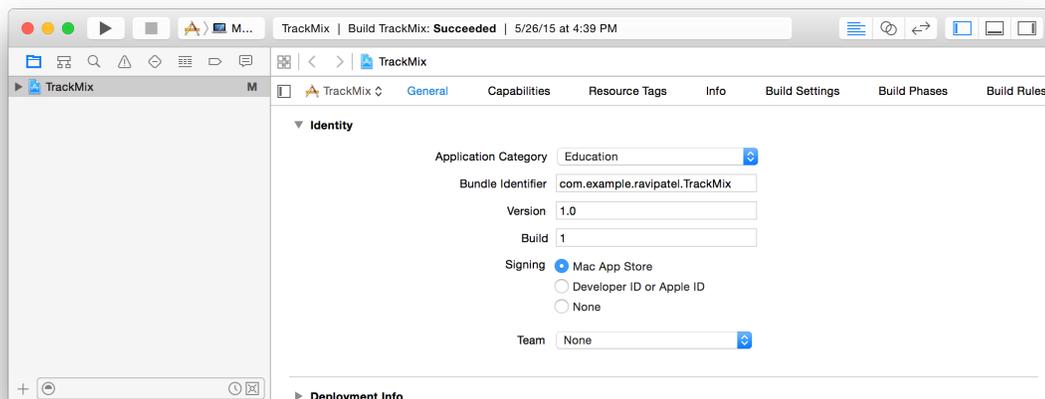
Xcodeは開発用証明書を失効させ、署名IDを作成し直します。

アプリケーションカテゴリを設定する (Mac)

MacアプリケーションをMac App Storeに表示する際のカテゴリを設定します。これは、後で[iTunes Connect](#)のアプリケーションレコードで選択するカテゴリと一致しなければなりません。

アプリケーションカテゴリを設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。
3. 「Application Category」ポップアップメニューからカテゴリを選んでください。



iOS/tvOS/watchOSアプリケーションのカテゴリはiTunes Connectでしか設定できません。アプリケーションカテゴリについて詳しくは、『*iTunes Connect Developer Guide*』の“First Steps: Identifying Your App in iTunes Connect”を参照してください。

版番号とビルド文字列を設定する

アプリケーションをiTunes Connectにアップロードする際には、版番号およびビルド文字列を設定/更新しなければなりません。ストアはこの番号を手がかりにして、個々のビルドを識別するからです。

版番号は、正の整数を3つ、ピリオド区切りで並べた文字列で、たとえば「4.5.2」のような形になります。それぞれ、メジャーリビジョン、マイナリビジョン、メンテナンスリリースを表す番号です。これはストアにも表示されます。また、後でiTunes Connectに入力する版番号と一致していなければなりません。取りうる値について詳しくは、『*Information Property List Key Reference*』の「CFBundleShortVersionString」 in *Information Property List Key Reference* を参照してください。

ビルド文字列はバンドルの状態（公開済みか否か）を表し、「1.2.3」のように正の整数値が2つのピリオドで区切られていることもあります。Macアプリケーションの場合、「About」ウインドウの版番号部分をクリックすると、版番号とビルド文字列の表示が交互に切り替わります。取りうる値について詳しくは、『*Information Property List Key Reference*』の「CFBundleVersion」 in *Information Property List Key Reference* を参照してください。

Important: iTunes Connectで新規にアプリケーションバージョンを作成したときは、版番号を更新してください。また、アプリケーションの新規ビルドをiTunes Connectにアップロードする際には、ビルド文字列を更新しなければなりません。ビルド文字列が表す番号を増やすのです。iOSアプリケーションの場合、iTunesはビルド文字列が変わったことを認識し、新しいアプリケーションを、適切にデバイスに組み込みます。版番号やビルド文字列は、TestFlightやストアを通して配布するため、あるいはクラッシュレポートや.dSYMファイルを識別するためにも使います。

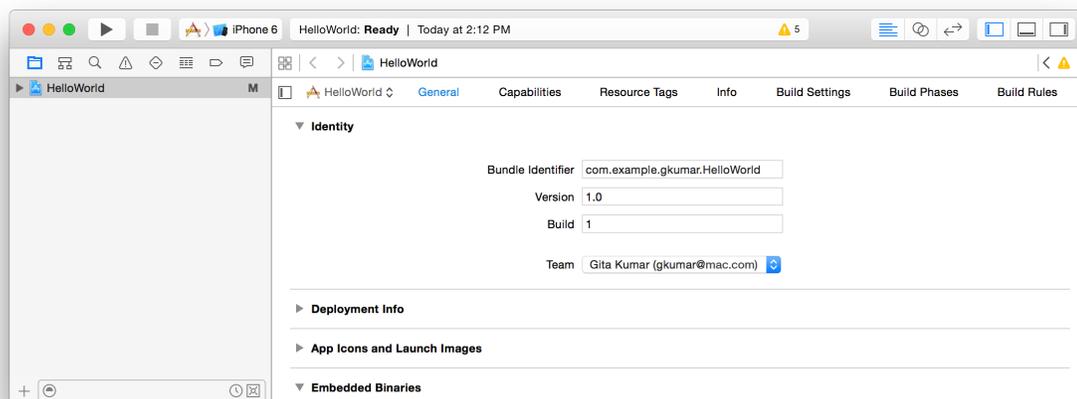
版番号やビルド文字列の設定は、プロジェクトエディタの「General」ペインで行います。

版番号やビルド文字列を設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。

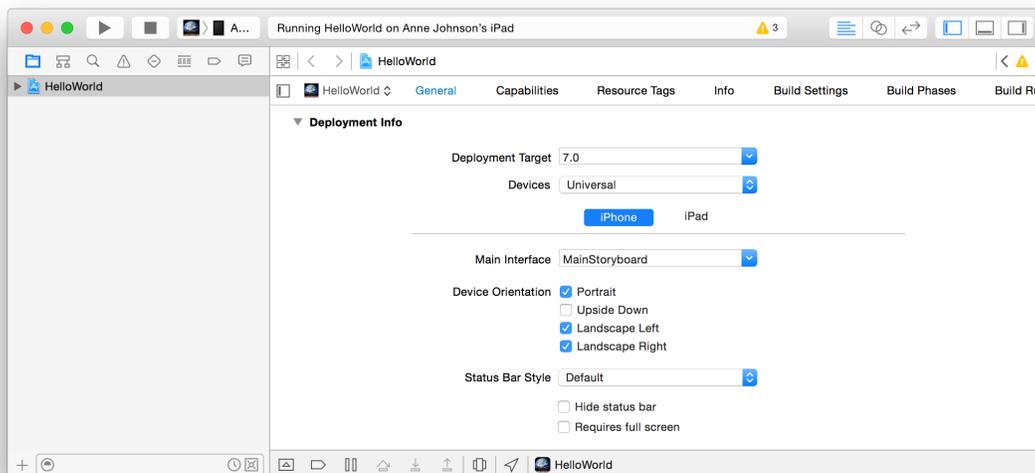
Important: watchOSアプリケーションの場合、埋め込みWatchKitアプリケーションおよびWatchKitエクステンションの版番号やビルド文字列は、対応するiOSアプリケーションと同じでなければなりません。

2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。
3. 「Version」欄に版番号、「Build」欄にビルド文字列を入力してください。



配備情報を設定する

開発中はデフォルトの配備設定でも構いませんが、配布が近づいたら設定内容を確認するとよいでしょう。設定事項のいくつかは、後でiTunes Connectで入力する値と一致していなければなりません。iOSアプリケーションの場合、配備設定は以下のように表示されます。



配備ターゲットを設定する

配備ターゲットの設定では、アプリケーションが動作できる最も古いOSの版を指定します。たとえばiPadアプリケーションの場合、iOS 4.3となります。

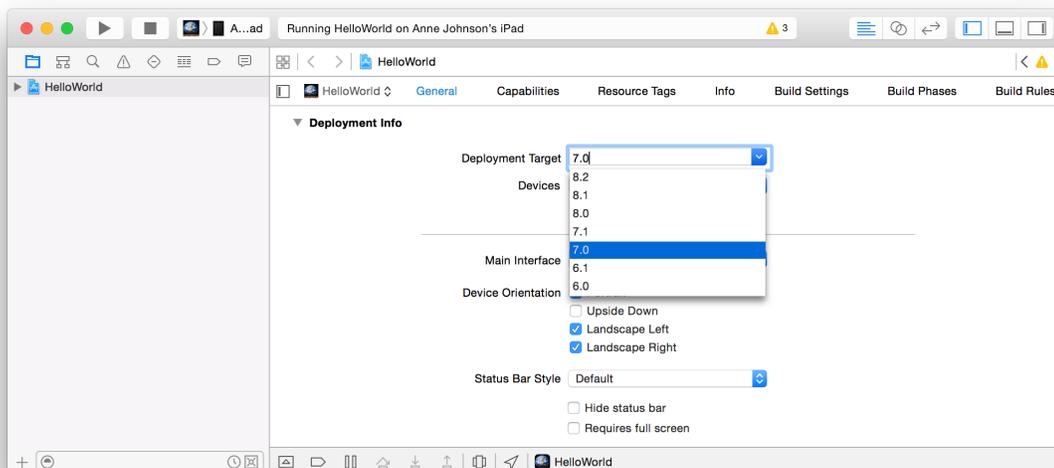
開発時に配備ターゲットを選択する方針はいくつか考えられます。OSの各版には、以前の版にはなかった機能が組み込まれています。新版が公開されれば早速アップグレードするユーザー、すぐにアップグレードせずそのまま様子を見るユーザーなどさまざまです。最新版をターゲットにすれば、新しい機能をすべて利用できますが、旧版を使っているユーザーは実行できません。逆に少し古い版までターゲットにすれば、実行できるユーザーは多くなりますが、最新の機能は利用できないこととなります。一方、古い版をターゲットにしつつ、新しい版の機能が使えるかどうか実行時に確認する、弱いリンクの仕組みを使う方法も考えられます。

弱いリンクについて詳しくは、『[“Weak Linking and Apple Frameworks” in SDK Compatibility Guide](#) 「Weak Linking and Apple Frameworks」 *SDK Compatibility Guide* を参照してください。

ターゲットの版を設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Deployment Info」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。

3. ターゲットとする版を「Deployment Target」ポップアップメニューから選択してください。



Xcodeはアプリケーションの情報プロパティリストに、「Minimum System Version」キーの値としてこの配備ターゲットを設定します。アプリケーションをストアに公開する際には、このプロパティを使って、アプリケーションが動作するシステムの版を示します。

注意: アプリケーションから利用できる機能は、配備ターゲットではなくSDKの版によって決まります。アプリケーションのビルドに使用したSDKが、配備ターゲットよりも新しければ、当該配備ターゲットでは利用できない機能をアプリケーションが使っている場合、Xcodeは警告を表示します。

使用するシンボルは、アプリケーションの実行環境で利用できるものでなければなりません。利用できるか否かを確認する方法については、『[SDKCompatibilityGuide](#)』を参照してください。

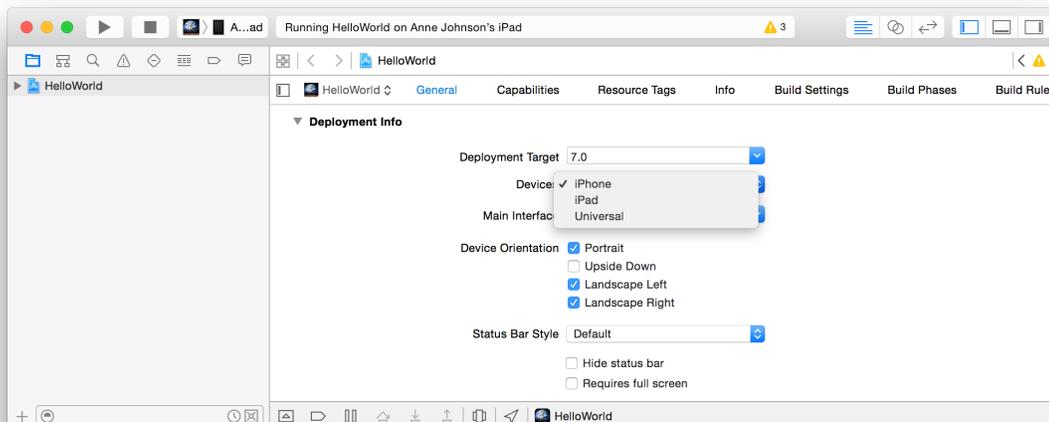
ターゲットデバイスを設定する (iOS、watchOS)

「Devices」設定では、アプリケーションが動作するデバイスの種類を指定します。デバイスタイプにはiPhoneとiPadの2種類があります。「iPhone」という種類は、iPhoneとiPod touchの両方を表します。一方「iPad」は、iPadやiPad miniを表します。

ターゲットデバイスを設定するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Deployment Info」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。

3. 「Devices」ポップアップメニューから、「iPhone」、「iPad」、「Universal」（どちらでも動作）のいずれかを選択してください。



動作対象デバイス（iPhone/iPad/両方）の設定について詳しくは、『*iOS App Programming Guide*』の“Advanced App Tricks”を参照してください。

アプリケーションアイコンや起動画面ファイルを追加する

アプリケーションアイコンや起動画面ファイルは、独立したアセットとしてiTunes Connectにアップロードするのではなく、アプリケーションバンドルに格納します。オペレーティングシステムでは、これらの画像と起動画面ファイル（.xibファイル）が該当するアプリケーションを表すものとしてデバイス上の各所で使用されます。一般に、オペレーティングシステムが表示するアートワークはバンドル内に置き、iTunesまたはストアが表示するアートワークはiTunes Connectにアップロードしてください。アプリケーションにはそれ自身を表すアイコンが必要であり、検証テストにも欠かせません。

アプリケーションアイコンを管理するアセットカタログを利用しても、個々の画像ファイルを自分で管理しても構いません。プロジェクトを新規に作成すると、アプリケーションアイコンを格納するのにアセットカタログがデフォルトで使用され、起動画面ファイルがスプラッシュ画面として使用されます。iOS 7の配備ターゲットには、起動画面ファイルと起動画像をどちらも設定できます。iOS 8では起動画面ファイルが使用され、iOS 7では起動画像が使用されます。旧式のプロジェクトで、個別にアイコン画像ファイルを管理していた場合でも、アセットカタログを使う方式に変更できます。旧式のXcodeプロジェクトに起動画像ファイルを追加することもできます。

アセットを個別の画像ファイルとして管理したい場合は、“[アプリケーションアイコンと起動画像ファイルを個別に設定する](#)”（50 ページ）を参照してください。アセットカタログに移したい場合は、“[画像をアセットカタログに移して管理する](#)”（50 ページ）を参照してください。アセットカタログを生成、管理する手順について詳しくは、『*Asset Catalog Help*』を参照してください。

アートワークを用意する

アートワークはいずれも、できるだけ容量を抑え、ユーザが購入する際に時間がかからないようにしなければなりません。

iOSアプリケーションに必要なアイコンや起動画像の大きさについては、『*iOS Human Interface Guidelines*』の“Icon and Image Sizes”を参照してください。

iOS/tvOSでは、アプリケーションアイコンをタップするとすぐに、デバイスの解像度に合った起動画像が表示されます。スクリーンショットをもとに作成するとよいでしょう。Retinaディスプレイを活かすためには、サポートする各デバイスに高解像度画像を提供する必要があります。起動画像を作成する方法については、“Launch Images”を参照してください。

Macアプリケーションに必要なアイコンについては、『*OS X Human Interface Guidelines*』の“Creating Great Icons for Any Resolution” in *OS X Human Interface Guidelines*を参照してください。この表には、Mac App Storeで使用できるアイコンのサイズが記載されています。アプリケーションアイコンファイルを作成する方法については、『*High Resolution Guidelines for OS X*』の“Provide High-Resolution Versions of All App Graphics Resources” in *High Resolution Guidelines for OS X*を参照してください。

後でiTunes Connectを使ってアップロードする、スクリーンショットその他のアートワークの仕様については、『*iTunes Connect Developer Guide*』の“iTunes Connect App Properties”を参照してください。

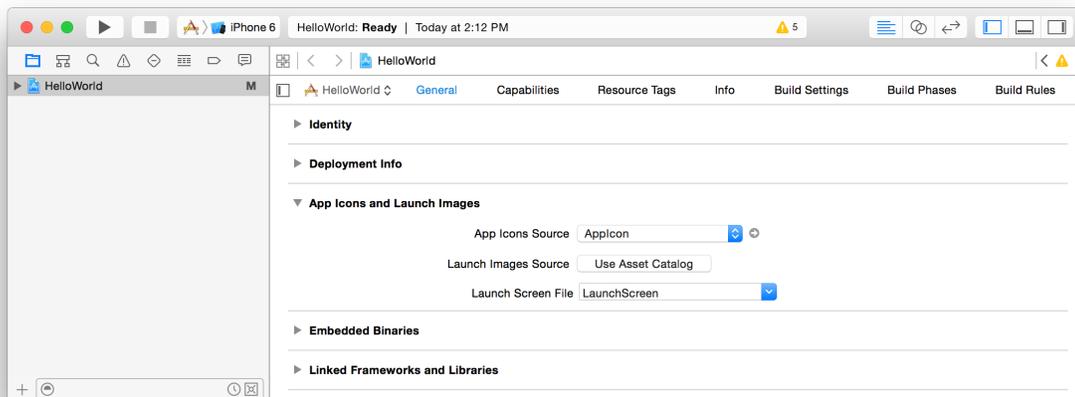
アプリケーションアイコンをアセットカタログに追加する

アセットカタログでは、各版のアプリケーションアイコンを画像セットとして整理します。Xcodeは自動的に、アプリケーションのターゲットデバイスごとに、画像セットを生成します。たとえば、iOSアプリケーションのターゲットが「ユニバーサル」であれば、iPhone用とiPad用の画像セットが現れます。

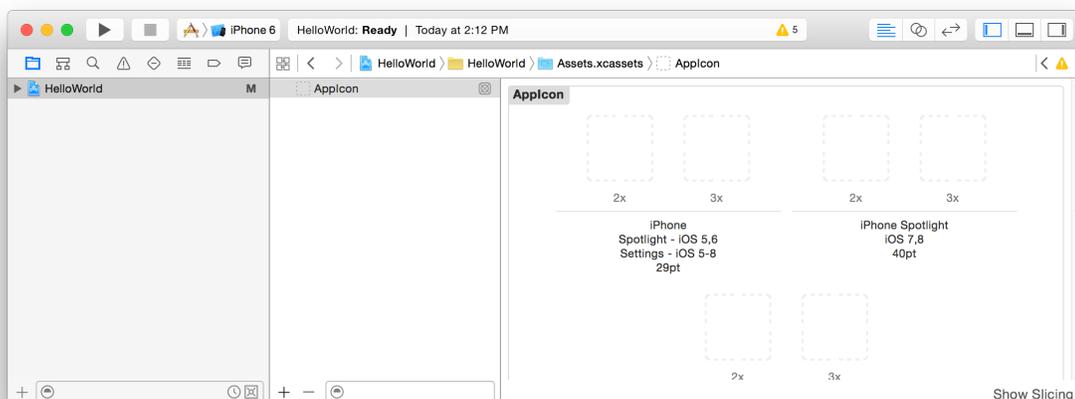
アプリケーションアイコンをアセットカタログに追加するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。

2. 「General」ペインの「App Icons and Launch Images」セクションで、「App Icons Source」ポップアップメニューの右にある矢印ボタンをクリックします。



3. Finder上でアプリケーションアイコンを掴み、プロジェクトナビゲータの、解像度が合致する画像ウェル上にドラッグしてください。



起動画面ファイルを作成する

起動画面ファイルは、アプリケーションの起動中にスプラッシュ画面として表示されます。単一のアトミック.xibファイルであり、サイズクラスを使ってさまざまなデバイスの解像度に対応しています。UIImageViewオブジェクトやUILabelオブジェクトなどの基本UIKitビューが組み込まれており、Auto Layout制約が適用されます。Xcodeにより、「LaunchScreen.xib」というデフォルトの起動画面ファイルがプロジェクトに追加されます。

起動画面ファイルを作成する際の注意点を以下に挙げます。

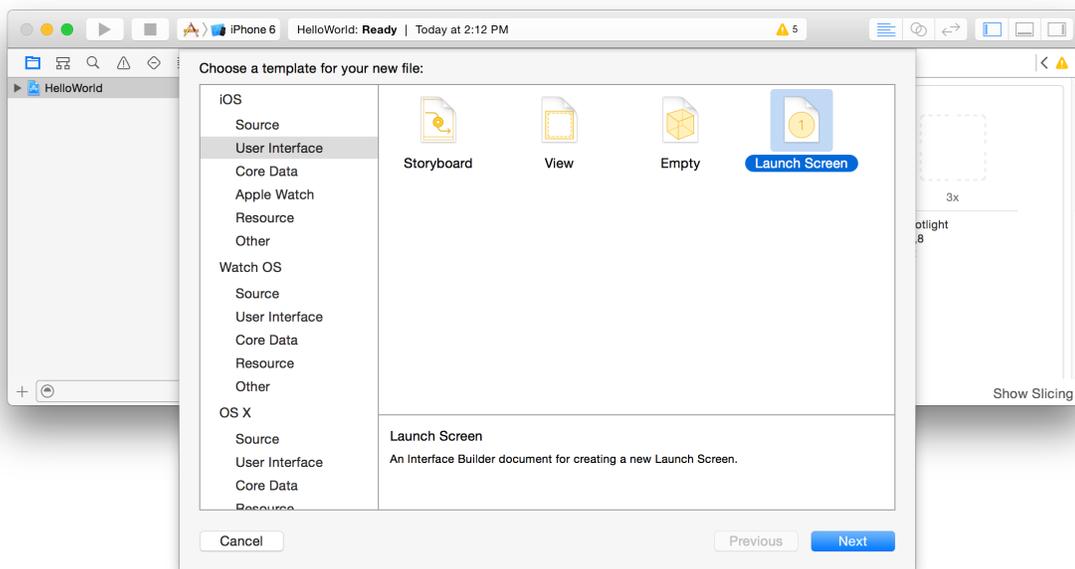
- UIKitクラスだけを使う。
- 1つのルートビュー（UIViewまたはUIViewControllerオブジェクト）を使う。

- コードへの接続を行わない（アクションもアウトレットも組み込まない）。
- UIWebViewオブジェクトを組み込まない。
- 独自のクラスを使わない。
- ランタイム属性を使わない。

旧式のXcodeプロジェクトに起動画像ファイルを追加できます。

既存のプロジェクト用の起動画面ファイルを作成するには

1. 「File」 > 「New」 > 「File」 の順に選択します。
2. 「iOS」 の下にある「User Interface」 を選択します。
3. 「Launch Screen」 をクリックしてから「Next」 をクリックします。



4. 「Save As」 テキストフィールドにファイル名を入力して「Create」 をクリックします。

起動画面ファイルを設定するには

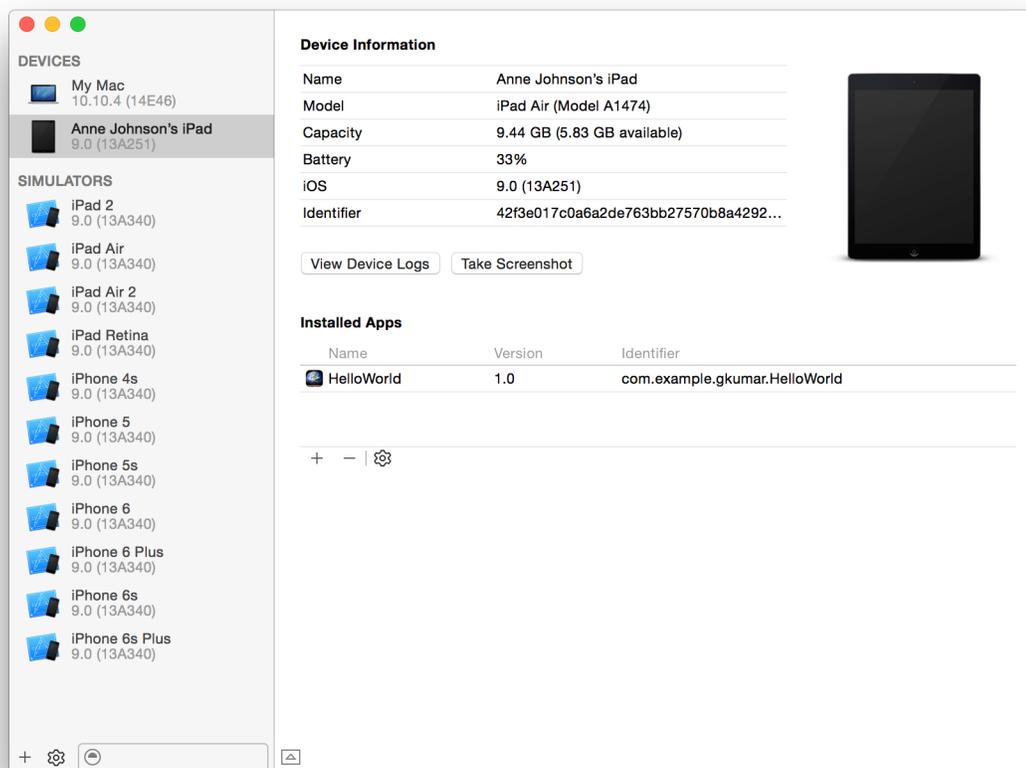
1. 必要な場合は、「General」 ペインの「App Icons and Launch Images」 セクションを開きます。
2. 「Launch Screen File」 ポップアップメニューから起動画面ファイルを選択します。

起動画像を追加し、スクリーンショットを取得する

デバイスをMacに接続しておき、以下の手順に従って、アプリケーションのスクリーンショットを取得してください。iOSアプリケーションのスクリーンショットをそのまま起動画面にすると、ステータスバーの部分も画面に入ってしまうますが、iOSでは、アプリケーションを起動する際、その時点のステータスバーに差し替えて表示するようになっています。

デバイス上で動作しているアプリケーションのスクリーンショットを取得するには

1. デバイスをMacに接続します。
2. 「Window」 > 「Devices」の順に選択した後、「Devices」から該当するデバイスを選択します。
3. 「Device Information」セクションで「Take Screenshot」をクリックします。



スクリーンショットがデスクトップ上に表示されます。

注意: iOS 7の配備ターゲットでは、起動画像のアセットカタログを追加します（[“画像をアセットカタログに移して管理する”](#)（50 ページ）を参照）。スクリーンショットを取得して起動画像を生成し、アプリケーションアイコンをアセットカタログに追加する場合と同様の方法で起動画像をアセットカタログに追加します（[“アプリケーションアイコンをアセットカタログに追加する”](#)（46 ページ）を参照）。

アプリケーションアイコンと起動画像ファイルを個別に設定する

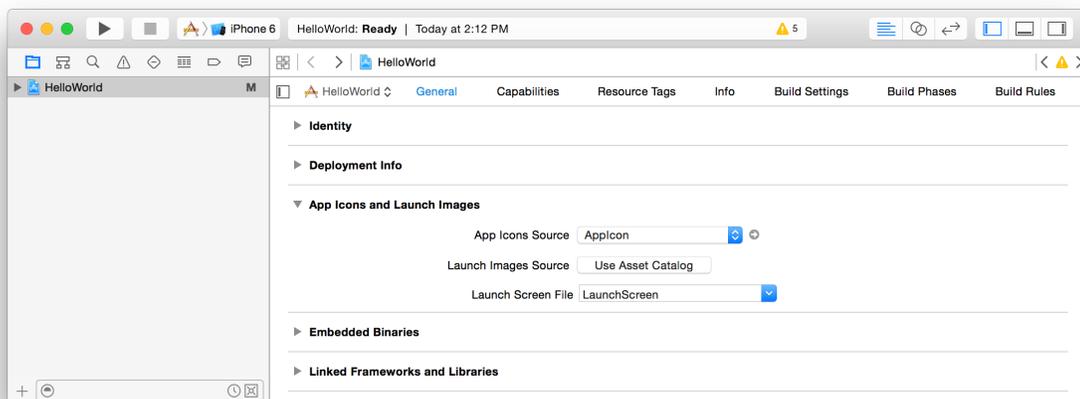
アセットカタログを使わない場合は、「General」ペインの「App Icons and Launch Images」セクションにある「App Icons Source」または「Launch Images Source」ポップアップメニューから「Don't use asset catalogs」を選択します。「Use Asset Catalog」ボタンが現れている場合、アセットカタログを使っていないこととなります。

画像をアセットカタログに移して管理する

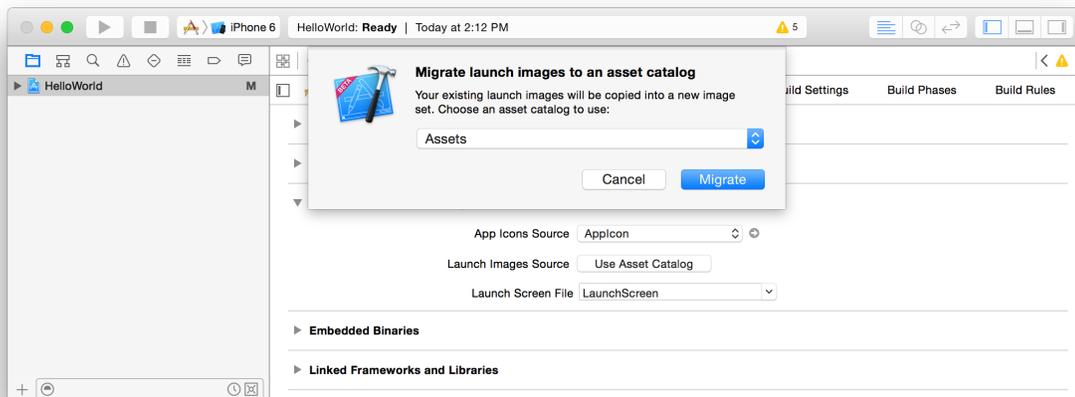
旧式のプロジェクトの場合、アプリケーションアイコンや起動画像などの画像ファイルは、アセットカタログに移して管理することができます。Xcodeは画像ファイルを、テーブルから新しいアセットカタログに移動するようになっています。アプリケーションアイコンと起動画像で別々のアセットカタログを作成できますが、手順は同じです。

アセットカタログに移行するには

1. プロジェクトナビゲータでプロジェクトとターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」をクリックして画面を切り替え、必要に応じて「App Icons and Launch Images」の横にある詳細表示用の三角印をクリックします。
3. 「App Icons Source」または「Launch Images Source」の横にある「Use Asset Catalog」ボタンをクリックします。



- 表示されるダイアログで、ポップアップメニューから「Assets」または「New Asset Catalog」を選択して「Migrate」をクリックします。



アセットカタログを表示するには、「App Icons Source」または「Launch Images Source」ポップアップメニューの右にある矢印ボタンをクリックします。

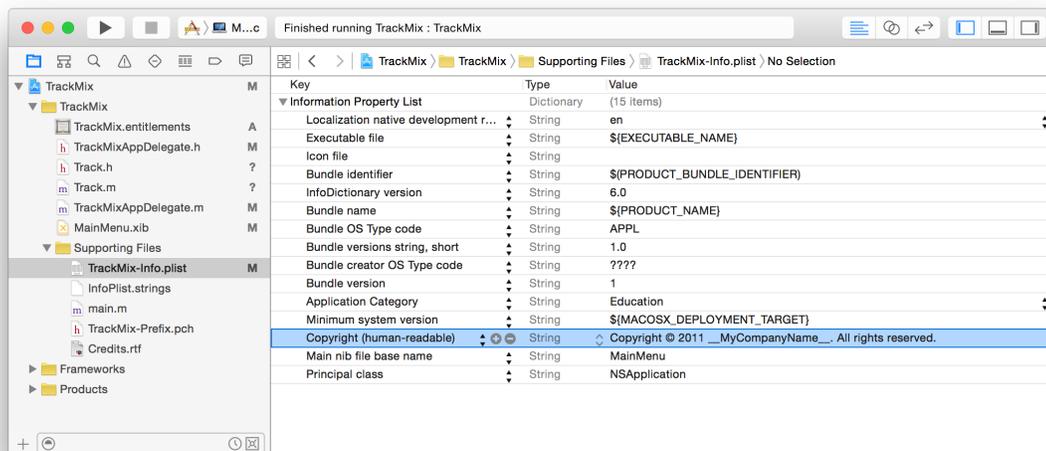
「Copyright」キーを設定する (Mac)

情報プロパティリストのファイルには、「Copyright」キーに対応する適切な値を記述しなければなりません。指定可能な値について詳しくは、『*Information Property List Key Reference*』の「NSHumanReadableCopyright」 in *Information Property List Key Reference* を参照してください。

情報プロパティリストの「Copyright」キーを編集するには

- Xcodeを起動し、プロジェクトナビゲータ上で該当するプロジェクトを選択します。
- `ProjectName` フォルダに添えられた展開表示用の三角印をクリックして、その内容を表示してください。
- さらに、「Supporting Files」サブフォルダの三角印もクリックして、その内容を表示します。
- 「`ProjectName-Info.plist`」という名前のファイルを選択してください。

情報プロパティリストが、プロパティリストエディタの右側に表示されます。



5. 「Copyright」キーに該当する行の「Value」列をダブルクリックします。
6. 新しい値を入力してください。

プロパティリストにある他のセルの編集方法については、『[Property List Editor Help](#)』を参照してください。

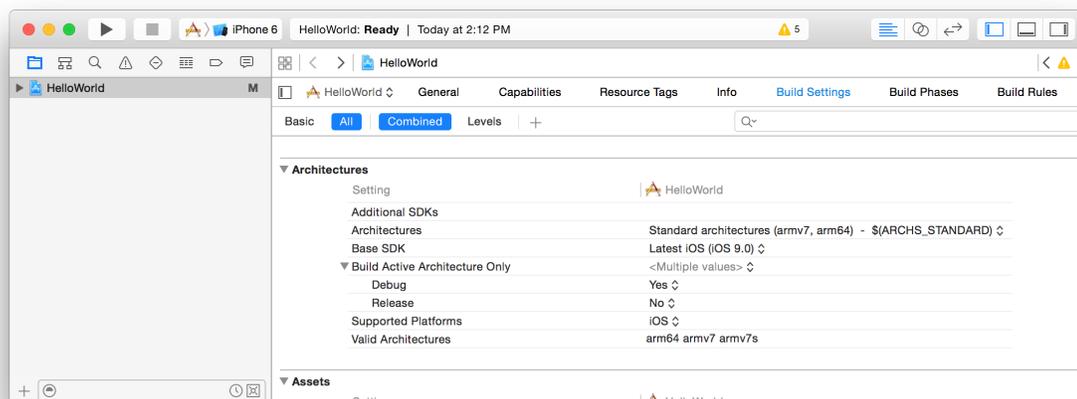
ビルド設定を検証する

デフォルトのビルド設定を変更した場合、アプリケーションをストアに登録する前に、設定内容を検証してください。これはプロジェクトエディタの「Build Settings」ペインで行います。

ビルド設定を編集するには

1. プロジェクトエディタで、ビルド設定を編集しようとするプロジェクトまたはターゲットを選択します。

- プロジェクトエディタの上部にある「Build Settings」をクリックします。



- 左側の列から該当するビルド設定を選択するか、右上隅にある検索欄にビルド設定名を入力してください。
- 必要なビルド設定項目が現れない場合は、「All」を押してみるとよいでしょう。
- 右欄にビルド設定の値を入力します。

アーキテクチャを設定する (iOSアプリケーション)

「アーキテクチャ(Architectures)」ビルド設定は、アプリケーションをビルドするためのアーキテクチャを指定します。iOSデバイスは、armv7、arm64など、各種のアーキテクチャに基づいて実装されています。次のいずれかを指定してください。

- Standard。** サポートされているすべてのデバイスと互換性のある、共通のアーキテクチャに基づいてアプリケーションバイナリを生成します。最低限の要件を満たすアプリケーションを生成できますが、あらゆるデバイス上で最高の処理速度を発揮できるような最適化は施しません。
- Other。** 指定したアーキテクチャ群に応じたバイナリを生成します。

「Other」を選択した場合、「Add」ボタン（「+」の表示）を押して、対応するアーキテクチャ名を入力してください。

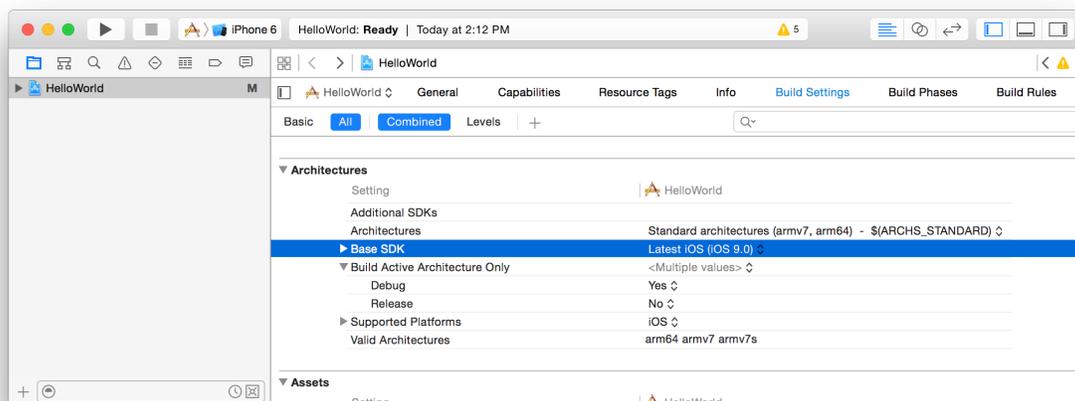
Important: armv7sのみに対応したビルドはストアが受け付けません。「Valid Architectures」リストに「armv7s」を入れた場合、「armv7」も入れる必要があります。

ベースSDKを設定する

ベースSDKの版番号は、開発デバイス上のソフトウェアの版番号以降にしてください。そうでなければ、当該デバイス上でデバッグセッションを開始できません。プロジェクトやターゲットのベースSDKは、最新のオペレーティングシステムに応じたものを指定してください（これがデフォルト値）。

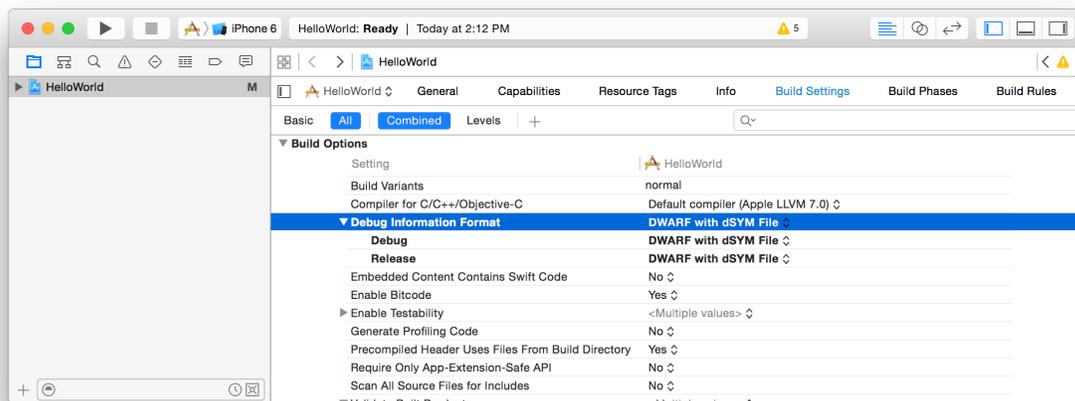
「Base SDK」プロパティは、「Build Settings」ペインの「Architectures」領域にあります。iOSアプリケーションの場合、「Base SDK」として「Latest iOS」を指定してください。Macアプリケーションであれば「Latest OS X」を指定します。それ以外を指定した場合は、デバイス上で動作するソフトウェアの版番号以降である、最新のSDKをダウンロード、インストールする必要があります。

「Architectures」領域に切り替えるため、該当するプロジェクトまたはターゲットを選択し、「Build Settings」を押してください。すると「Build Settings」ペインに、まず「Architectures」領域が現れます。



デバッグ情報の形式を設定する

「Debug Information Format」ビルド設定として「DWARF with dSYM File」を指定します。これはクラッシュレポート中のアドレスをシンボルに置き換えて表示するために必要です（[“クラッシュレポートの分析”](#)（132 ページ）を参照）。

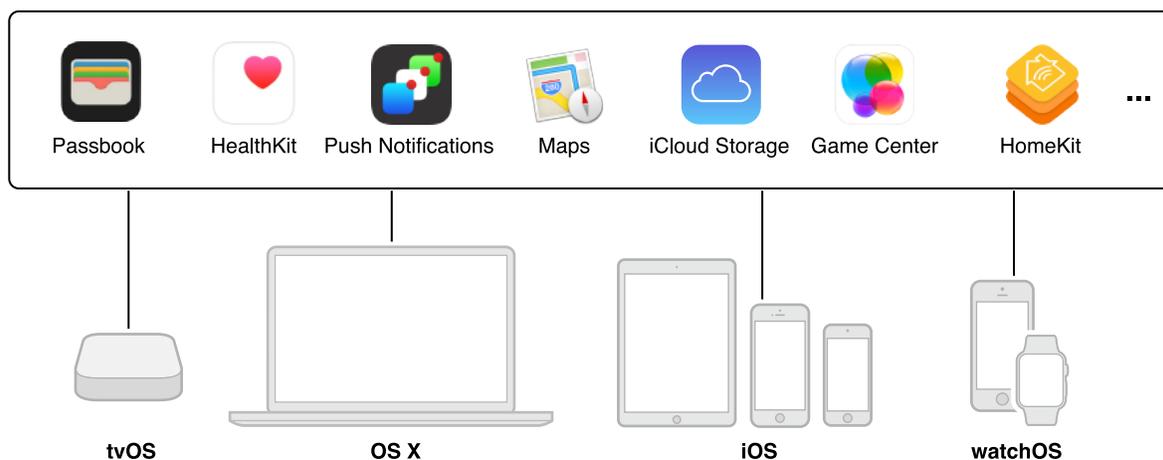


まとめ

この章では、テンプレートをもとにXcodeプロジェクトの設定を行い、アプリケーションのID情報を設定し、開発に用いるチームプロビジョニングプロファイルを生成する手順を説明しました。開発の際には、必要に応じてこの章を読み返してください。その後も、ストアに登録する際に必要な設定項目があるので、チェックリスト代わりに参照するとよいでしょう。

技術やサービスを組み込む

Game CenterやIn-App Purchaseなどの特定のアプリケーションサービスは、Apple Developer Programのメンバーが、ストアを通してアプリケーションを配布するためにしか使えません。これらのサービスを利用するには、Xcodeプロジェクト、メンバーセンター、場合によってはiTunes Connectで追加設定が必要となります。iTunes Connectでの設定を必要としないアプリケーションサービスは、Apple Developer Enterprise Programアプリケーションに組み込むこともできます。アプリケーションサービスの中には、ゲームやNewsstandなどの特定のアプリケーション向けのものだけでなく、新たな収入源に結びつくもの（In-App PurchaseやiAd Networkなど）もあります。



Appleは、ユーザデータやアプリケーションが、知らない間に改竄、配布されてしまうことのないよう、基盤セキュリティモデルを構築しています。そこで、アプリケーションにコード署名およびプロビジョニングを施して、指定したアプリケーションサービスだけを使うようにしなければなりません。これをXcode上で実行すれば、必要なプロジェクト設定はXcodeが自動的に行うようになっていきます。具体的には、エンタイトルメントや情報プロパティリストを自動的に編集し、必要に応じて、該当する技術に応じたフレームワークを追加するのです。また、エンタイトルメントを有効にするため、Xcodeはコード署名やプロビジョニングに必要なアセットを生成してチーム全体で利用できるようにするとともに、コード署名に関するビルド設定も行います。さらに、必要に応じてワイルドカードAppIDや特定AppIDを生成して、選択したアプリケーションサービスが使えるようにします。Game CenterやIn-App Purchaseのように、メンバーセンターとiTunes Connectで追加の設定が必要となるアプリケーションサービスもあります。プッシュ通知を組み込むためには、メンバーセンターで追加の設定作業が必要です。

この章では、アプリケーションからアプリケーションサービスを利用できるようにするための手順を解説します。利用可能なケイパビリティの一覧が、アプリケーションの種類ごとに、“サポートされるケイパビリティ”（258 ページ）に載っています。

エンタイトルメントについて

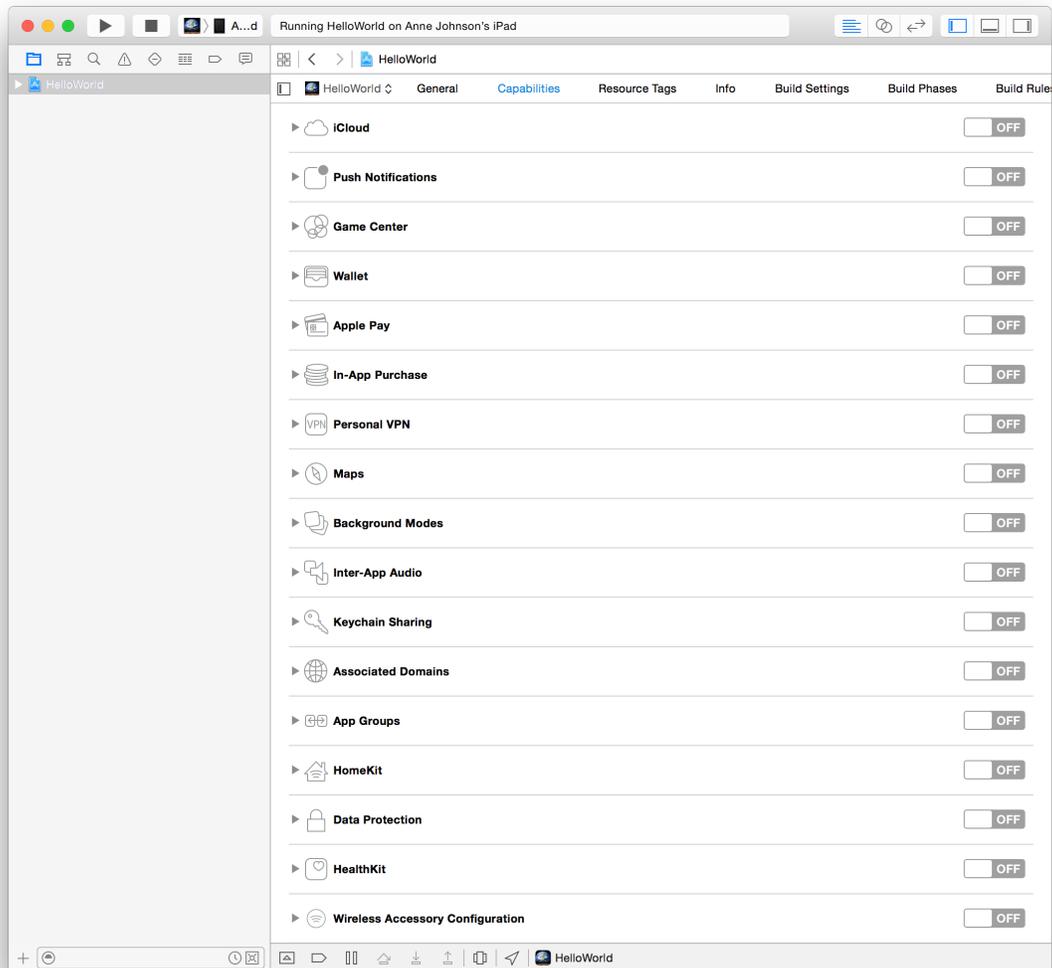
エンタイトルメントとは、ある特定のアプリケーションやツール、その他の実行形式ファイルに、追加する形で付与される権限のことです。エンタイトルメントはサンドボックスに関する文脈でよく使われるほか、AppIDについて言うこともあります。その場所は何通りかありますが、いずれもアプリケーションのコード署名に含まれる設定情報であって、アプリケーションがあるリソースにアクセスし、あるいはある処理を実行できるよう、システムに許可を求めます。実質的に、アプリケーションのサンドボックスや能力を拡張し、所定の処理を可能にする働きがあります。

Xcodeでアプリケーションサービスを有効にすることにより、エンタイトルメントをXcodeプロジェクトで設定する場合とメンバーセンターでAppIDに対して設定する場合があります。アプリケーションサービスをApp IDに対して有効にすると、1つまたは複数のアプリケーションで利用できるサービスのホワイトリストに登録されます。一部のアプリケーションサービスは、バンドルIDと完全に一致する特定App IDに対してデフォルトで有効になっています。Xcodeプロジェクト設定では、アプリケーションで実際にどのサービスを利用するかを指定します。

事前の確認

この章で説明するオプションはいずれも、ターゲットに対応する、プロジェクトエディタの「Capabilities」ペインにあります。

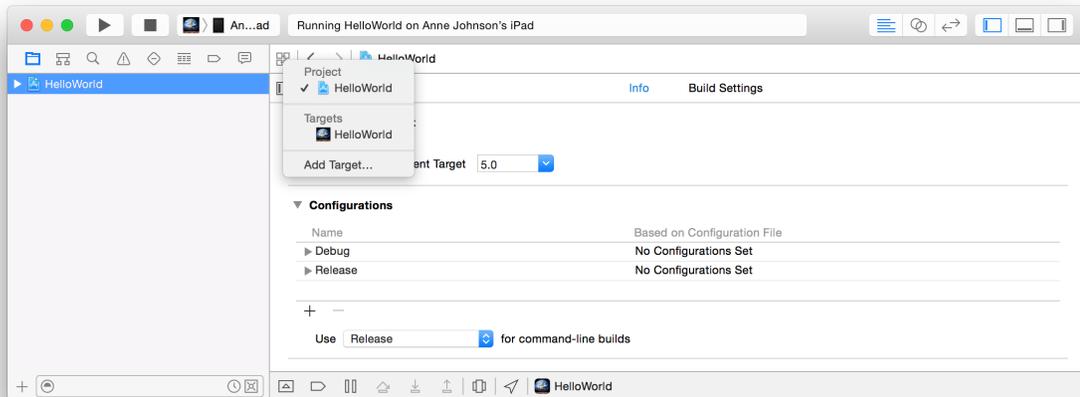
iOSアプリケーション用の「Capabilities」ペインの画面例を以下に示します。これらのアプリケーションサービスの一部はMacアプリケーションで利用できます。



「Capabilities」ペインを開くには

1. 「View」 > 「Navigators」 > 「Show Project Navigator」 コマンドを実行してください。

- 次にターゲットを「Project/Targets」ポップアップメニューから選択します。2つめのサイドバーが現れていれば、その「Targets」セクションで選択しても構いません。



- 「Capabilities」をクリックして、アプリケーションに組み込めるアプリケーションサービスを表示します。

Xcodeにより、必要に応じてコード署名とプロビジョニングに使用するアセットが生成されます。ただし、アセットの中には互いに依存するものもあるため、技術やサービスを有効にする際に、問題の修正を求めるダイアログが表示されることがあります。具体的な作業としては、チームをプロジェクトに対応づける、開発用証明書を生成する、iOS/tvOS/watchOSアプリケーションの場合に、デバイスを接続してXcodeでチームプロビジョニングプロファイルを生成できるようにする、などがあります。開発用プロビジョニングプロファイルは、技術やサービスを有効にする場合には必要ありませんが、これを利用するアプリケーションをビルドし、起動する場合に必要となります。ビルド、起動時にダイアログや警告が出ないように、コード署名IDやチームプロビジョニングプロファイルは、この時点で生成しておくといでしょう（[「チームプロビジョニングプロファイルを生成する」](#)（39 ページ）を参照）。あるいは、問題が生じた時点で、[「トラブルシューティング」](#)（90 ページ）に従って解決してください。

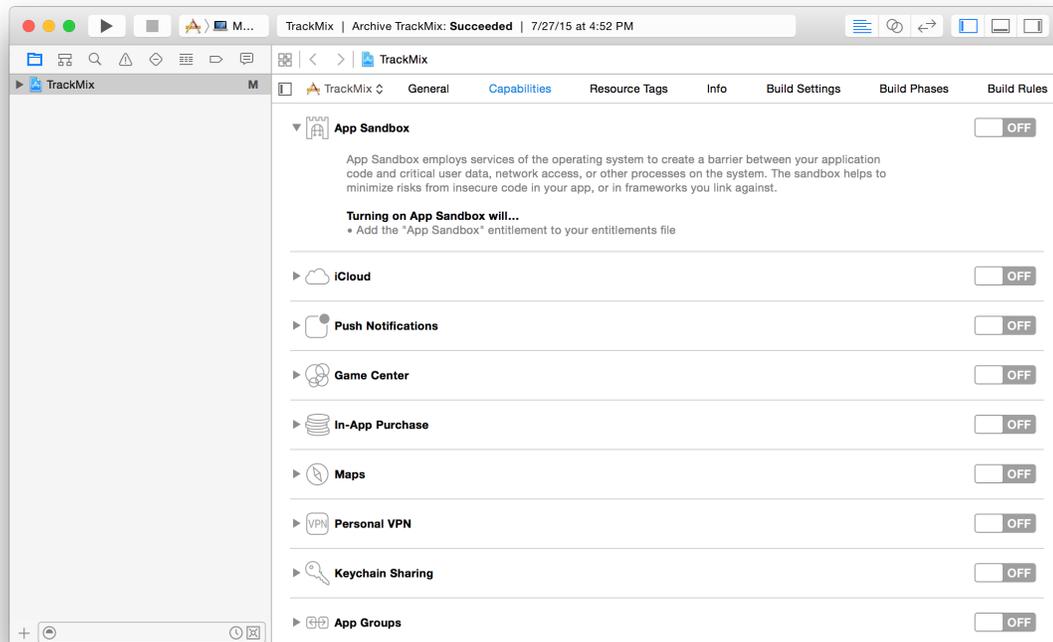
アプリケーションサンドボックスを設定する (Mac)

サンドボックスの機能は、悪質なコードでMacアプリケーションが不正に利用された場合に、ユーザーデータを盗難や破壊、削除から守るための、最後の砦です。アプリケーションやフレームワークのコードに誤りがあったとき、被害を最小限に抑える働きもあります。この機能を有効にするだけで、システムの他の部分と情報をやり取りする際、最大レベルの制約が課されることとなります。Mac App Storeから配布されるアプリケーションはいずれも、サンドボックスの機能を必ず利用しなければなりません。したがって、アプリケーションをiTunes Connectにアップロードする場合は、開発期間中にサンドボックス機能を有効にしておいてください。

サンドボックス機能の設定は、これを有効にした上で、必要に応じて特定機能の利用権限を付与する、という形で行います。

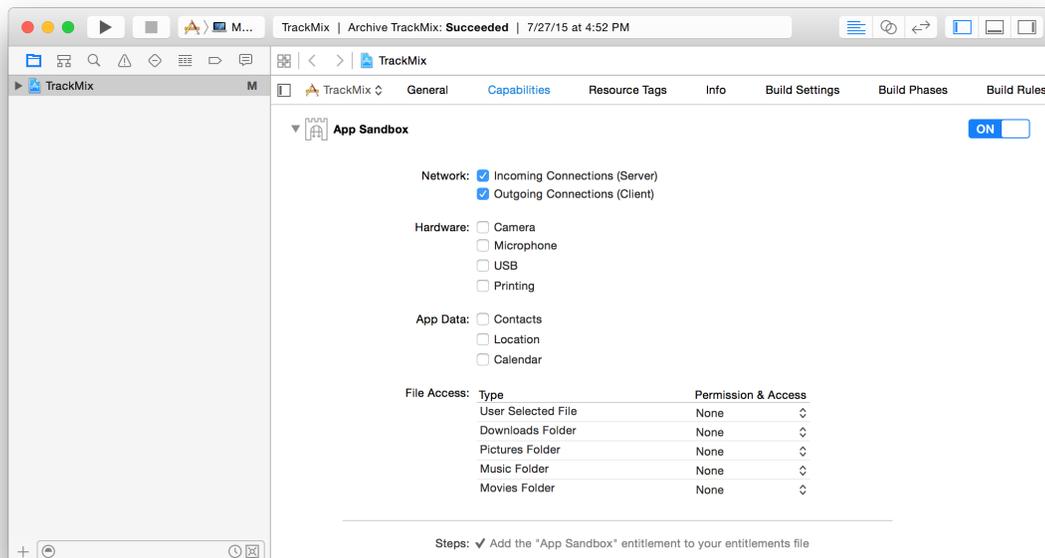
アプリケーションサンドボックスの設定をするには

1. 「Capabilities」ペインで「App Sandbox」が有効になっていない場合は、「App Sandbox」セクションのスイッチをオンにします。

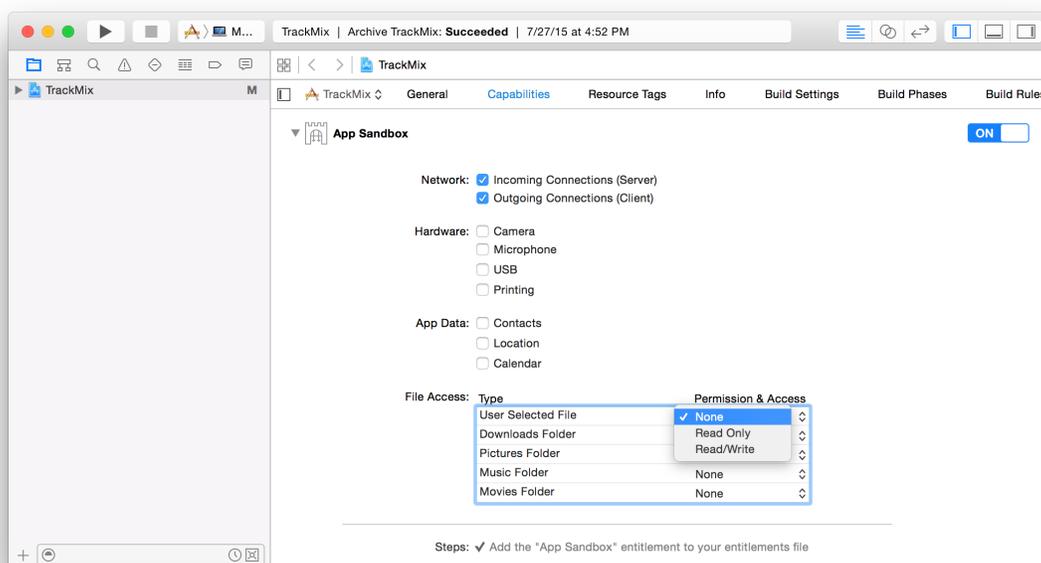


するとエンタイトルメントファイルがプロジェクトに追加され、いくつかのエンタイトルメントについては自動的にデフォルト値が入ります。また、「App Sandbox」エンタイトルメントも有効になります。

2. 「App Sandbox」以下に並んでいるチェックボックスで、ターゲットがその処理をするために必要な、最小限の能力を設定してください。



ファイルの種類に応じて操作権限を設定することも可能です。これは、ファイルの種類に応じた行にある、ポップアップメニューから選択することにより行います。



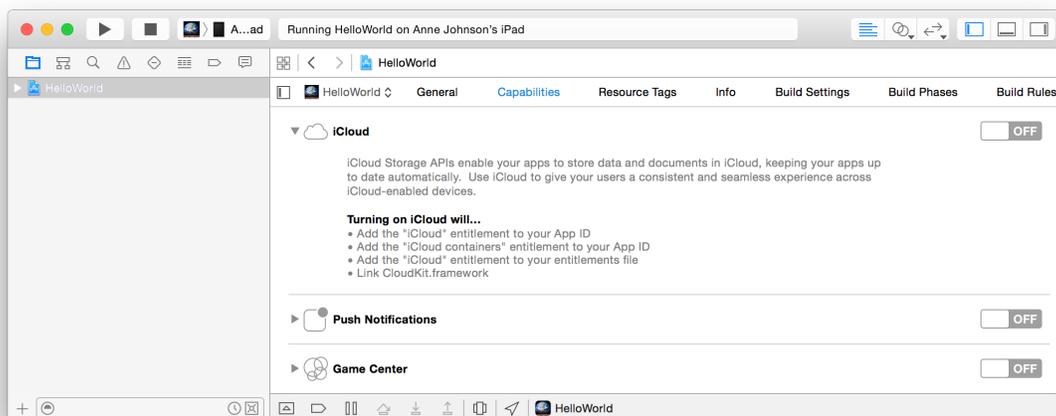
「App Sandbox」 エンタイトルメントについては、『*Entitlement Key Reference*』を参照してください。既存のアプリケーションに対してサンドボックス機能を有効にする場合は、新たにアクセスできるようになる場所について、『*App Sandbox Design Guide*』を参照してください。

iCloudの操作機能を追加する

iCloudストレージを使えば、デバイス上で動作するアプリケーションどうしで、ユーザデータまたはアプリケーションデータを共有できます。チームが開発したアプリケーションどうしでデータを共有することもできます。データの保存方法と取得方法に応じて、キー値ストレージ、文書ストレージ、CloudKitの中から、使用するiCloudサービスを選択します。文書ストレージまたはCloudKitを選択した場合は、アプリケーションで使用するコンテナを指定し、複数のアプリケーション間で共有する独自のコンテナを作成することができます。iCloudを利用するようにアプリケーションをプロビジョニングする必要があり、その一環として、特定App IDを作成し（まだ作成していない場合）、Xcodeプロジェクトでサービス固有のエンタイトルメントを設定します。

iCloud を有効にする

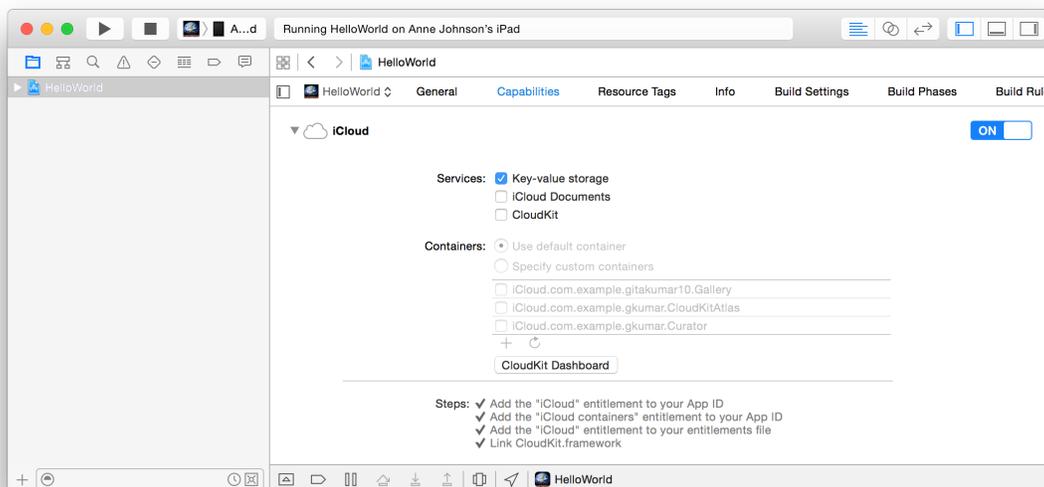
iCloudサービスを選択して設定する前に、XcodeでiCloudを有効にしておく必要があります。iCloudを有効にするには、「iCloud」セクションのスイッチをオンにします。Xcodeにより、アプリケーションでiCloudを利用できるようにプロビジョニングが施されます。



キー値ストレージを設定する

キー値ストレージには、他のデバイスで動作する同じアプリケーションと少量のデータを共有できるようにする働きがあります。キー値ストレージのコンテナIDは `iCloud.[$(TeamIdentifierPrefix)].[$(CFBundleIdentifier)]` であり、ここでチームIDはチー

ムに割り当てられた一意の文字列を示します。キー値ストレージを有効にするには、「Key-value storage」チェックボックスをオンにします。キー値ストレージの使い方について詳しくは、『*iCloud Design Guide*』を参照してください。



文書ストレージを設定する

文書ストレージには、ユーザ文書とアプリケーションデータがユーザのiCloudアカウントで格納されます。iCloud文書ストレージを有効にするには、「iCloud Documents」チェックボックスをオンにします。必要に応じて、Xcodeにより、文書ストレージ用のデフォルトのiCloudコンテナが作成されます。文書ストレージの使い方について詳しくは、『*iCloud Design Guide*』を参照してください。

Important: 文書ストレージを設定できるのは、チームエージェントおよび管理者に限ります。

CloudKitを使う

CloudKitを使うと、アプリケーションのデータをレコードとして取得および格納し、複数のデバイスからアプリケーションのデータにアクセスできます。また、データを公開領域に格納して、さまざまなユーザが実行しているアプリケーションのすべてのインスタンスからアクセスできるようにすることもできます。CloudKitを有効にして、CloudKitフレームワークをプロジェクトに追加するには、「CloudKit」チェックボックスをオンにします。CloudKitコンテナデータモデルとレコードを管理するには、「CloudKit Dashboard」ボタンをクリックします。CloudKitの使用を開始する場合は『*CloudKit Quick Start*』を参照し、CloudKitの詳細を確認する場合は『*CloudKit Framework Reference*』を参照してください。必要に応じて、Xcodeにより、CloudKit用のデフォルトのiCloudコンテナが作成されます。

Important: CloudKitを設定できるのは、チームエージェントおよび管理者に限ります。

独自のコンテナを指定する

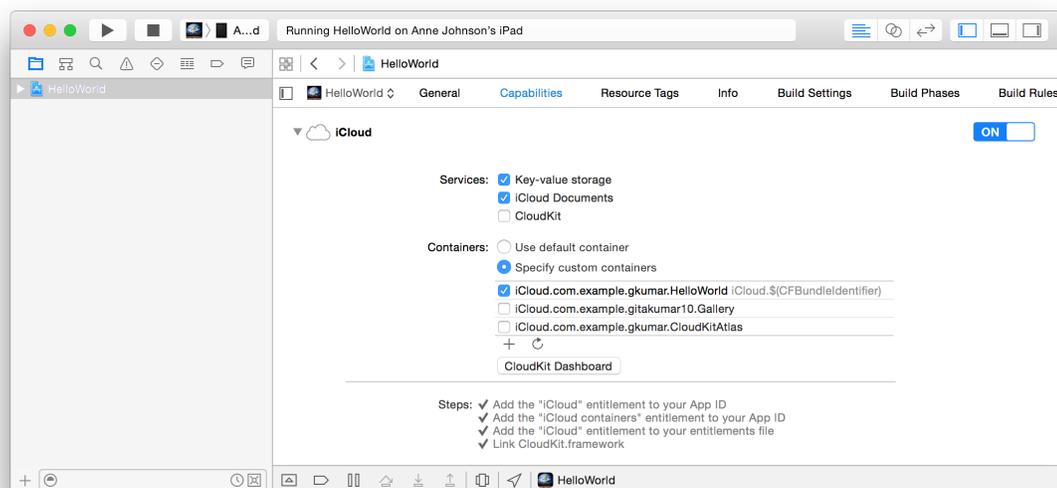
文書ストレージとCloudKitにデフォルトで設定されるコンテナIDは*iCloud.\$(CFBundleIdentifier)*であり、このIDは特定App IDと一致しています。必要に応じて、1つまたは複数の独自のコンテナを追加し、アプリケーション間で共有します。別のアプリケーションで使われている既存のコンテナIDを選択しても、新規のコンテナIDを作成しても構いません。

Important: 独自のコンテナを指定できるのは、チームエージェントおよび管理者に限ります。

コンテナIDを選択または選択解除するには

1. 「iCloud」設定で、「Specify custom containers」をオンにします。
2. 必要な場合は、テーブルの下にある「Refresh」ボタンをクリックして、別のアプリケーションで使われているコンテナIDをダウンロードします。
3. 該当するコンテナIDの左にあるチェックボックスをオン（そのコンテナを使用する場合）またはオフ（そのコンテナを使用しない場合）にします。

Xcodeにより、Xcodeプロジェクトのエンタイトルメントファイル内にあるコンテナIDのリストが更新されます。



既存のコンテナIDが十分でない場合は、アプリケーション用のコンテナIDを別途作成します。

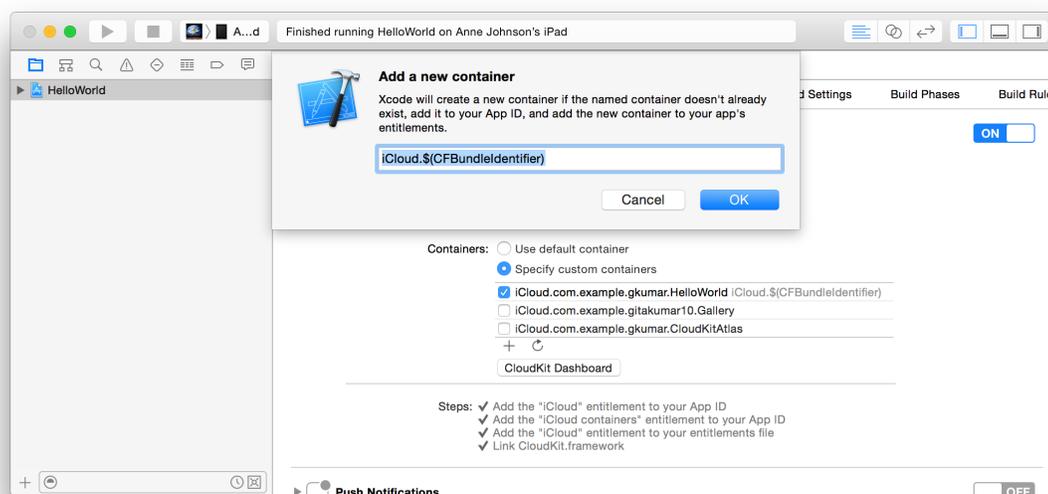
コンテナIDを追加するには

1. 必要な場合は、「Specify custom containers」をオンにします。

2. テーブルの下部にある「Add」ボタン（「+」の表示）を押します。
3. 表示されるダイアログで、追加するテナIDを入力します。

テナIDは、`iCloud.com.example.gkumar.shared`のように、「iCloud.」で始まり、その後逆DNS形式の文字列が続きます。

Warning: テナIDは削除できないので、文字列を選択する際には十分な注意が必要です。



4. 「OK」をクリックします。

Xcodeにより、Xcodeプロジェクトのエントリーメントファイルとメンバーセンターに新規のテナIDが追加されます。

iCloudテナの選び方については、『*iCloud Design Guide*』を参照してください。

プッシュ通知の設定をする (iOS、watchOS、Mac)

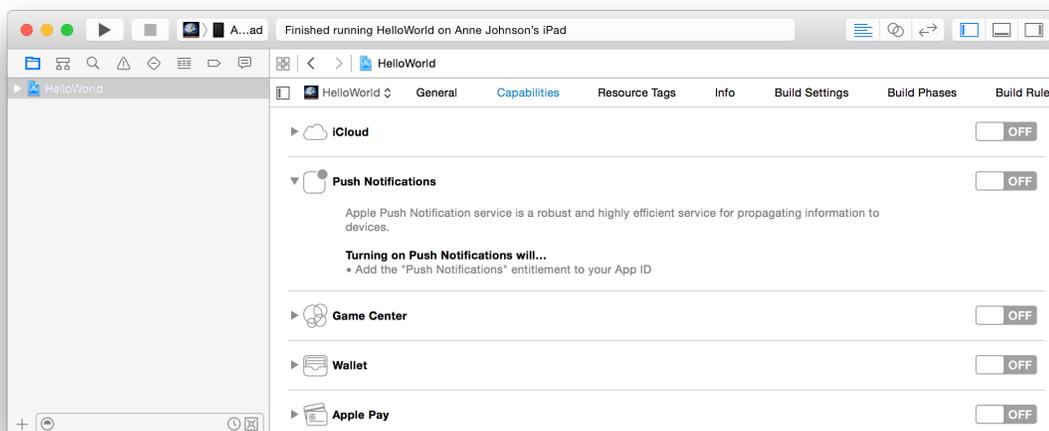
「Apple Push Notification service (APNs)」には、フォアグラウンドで動作していないアプリケーションが、何らかの情報をユーザに通知できるようにする働きがあります。プッシュ通知を有効にする設定は、Xcodeでおこないます。Xcodeは必要に応じて、APNsエントリーメントを有効にした特定App IDと、これを用いるチームプロビジョニングプロファイルを生成します。対応するクライアントSSL証明書を作成、ダウンロードし（プッシュ通知を完全に有効にする手順）、後でクライアントSSL証明書とキーをサーバにインストールすることになります。

プッシュ通知を有効にする

まず、Xcodeプロジェクトで、プッシュ通知を有効にします。

プッシュ通知を有効にするには

1. 「Capabilities」 ペインで「Push Notifications」 が有効になっていない場合は、「Push Notifications」 セクションのスイッチをオンにします。



クライアントSSL証明書を生成するまでは、メンバーセンターに、「Push Notifications」 サービスが（「Enabledではなく）「Configurable」である旨表示されます。

プッシュ通知用のクライアントSSL証明書を作成する

プッシュ通知用クライアントSSL証明書の作成はメンバーセンターで行います。**クライアントSSL証明書**には、通知サーバがAPNsに接続できるようにする働きがあります。この証明書はAppIDごとに必要です。さらに、署名証明書と同様、開発用と配布用で別々に用意しなければなりません。

Important: Apple Push NotificationサービスのSSL証明書を作成できるのは、チームエージェントおよび管理者に限ります。

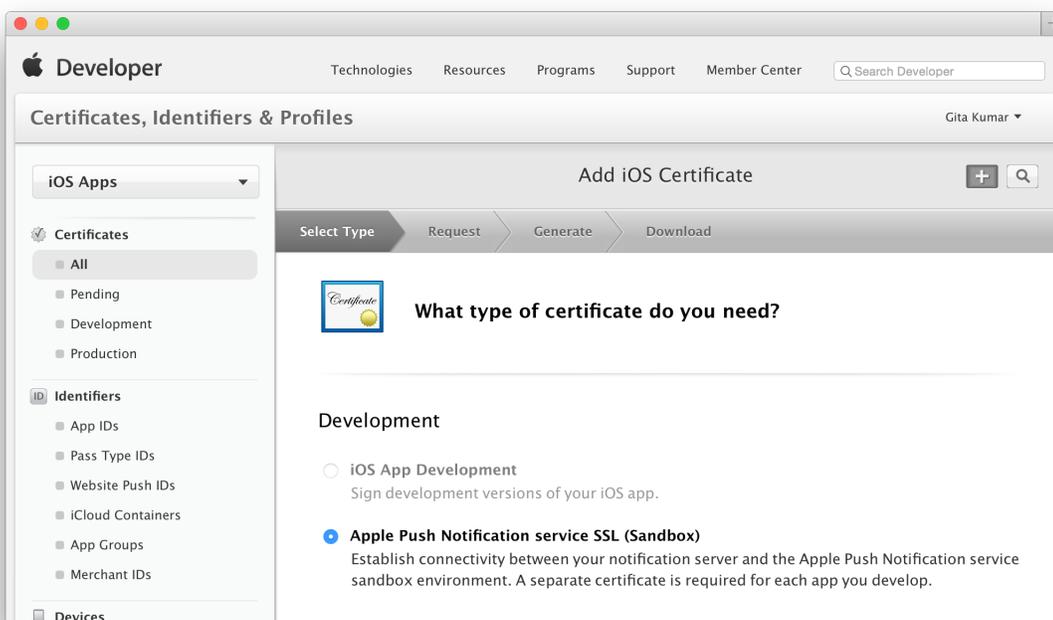
クライアントSSL証明書を生成するには

1. 「Certificates, Identifiers & Profiles」の画面で「Certificates」を選択します。

2. 右上隅の「Add」ボタン（「+」の表示）を押してください。

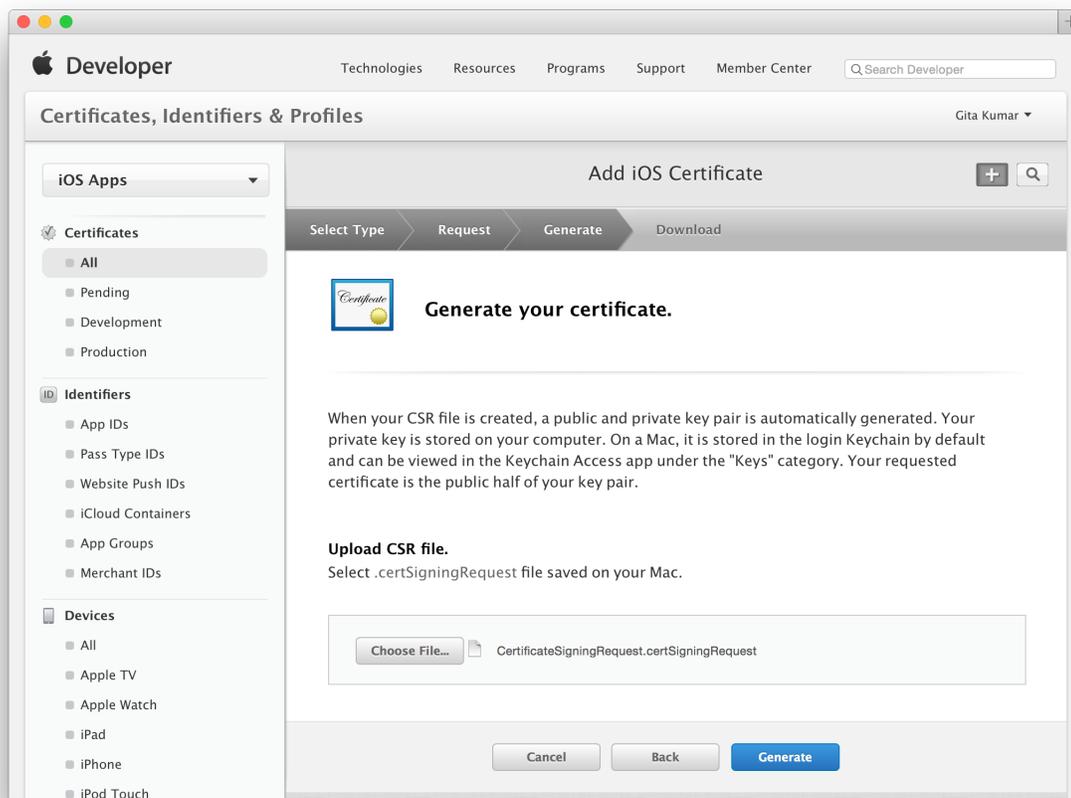


3. 「Development」または「Production」の下にある「Apple Push Notification service SSL」チェックボックスをオンにして「Continue」をクリックします。



4. 「App ID」ポップアップメニューから該当するApp IDを選択し、「Continue」を押します。バンドルIDに合致する特定App IDを選択してください。
5. 次のウェブページに表示される手順に従って、Mac上で証明書要求を作成し、「Continue」を押します。
6. 「Choose File」を押してください。
7. ダイアログが現れるので、証明書要求ファイル（拡張子「.certSigningRequest」）を選択し、「Choose」を押してください。

- 「Generate」を押します。



- 必要ならば「Download」を押してください。

- 「Done」を押してください。

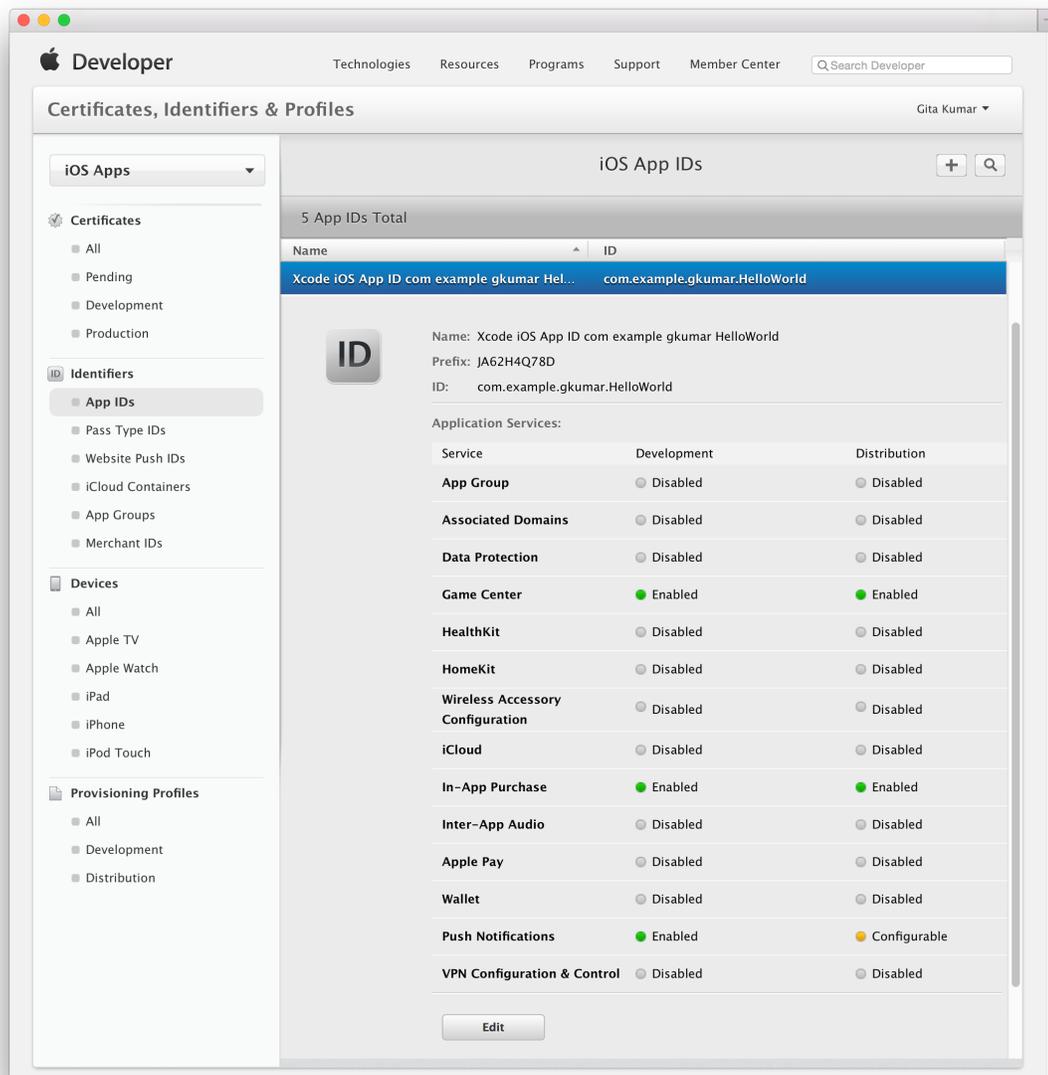
ここまでの結果を確認する

メンバーセンターで確認すると、指定した環境（開発用または配布用）の「Push Notifications」サービスの状態が、「Configurable」から「Enabled」に変わっています。

App IDの設定を確認するには

- 「Certificates, Identifiers & Profiles」画面で「Identifiers」を押し、「Identifiers」以下で該当するApp IDを選択します。
- バンドルIDに合致する特定App IDを選択してください。

緑色の丸の後に「Enabled」の表示が「Push Notifications」行と「Development」列または「Distribution」列（先に生成したクライアントSSL証明書の種類に応じて異なる）に現れているはずですが、黄色の丸の後に「Configurable」の表示が「Development」列または「Distribution」列に現れている場合、該当するクライアントSSL証明書はありません。



クライアントSSL署名IDをサーバにインストールする

以前に取得したSSL配布署名IDを、プロバイダコードを実行し、開発用または実稼働用の APN に関連付けられているサーバにインストールします。署名IDをその作成時に使用したMac上のキーチェーンからエクスポートし、サーバ上の適切な場所にコピーします。

クライアントSSL署名IDをエクスポートする

1. 「Keychain Access」を起動します。

2. 「Category」セクションの「My Certificates」を選択します。
3. インストールする証明書を検索し、中身を表示します。
すると証明書と秘密鍵の両方が表示されます。
4. 証明書と鍵の両方を選択して「File」>「Export Items」の順に選択します。
5. 「File Format」ポップアップメニューから、サーバで使用できるファイル形式を選択します。
6. 「Save As」フィールドにファイル名を入力して「Save」をクリックします。

Personal Information Exchange形式のファイルの拡張子は.p12、Privacy Enhanced Mail形式のファイルの拡張子は.pemになります。

プッシュ通知サーバに問題が生じた場合の対処方法は、『[Troubleshooting Push Notifications](#)』に載っています。

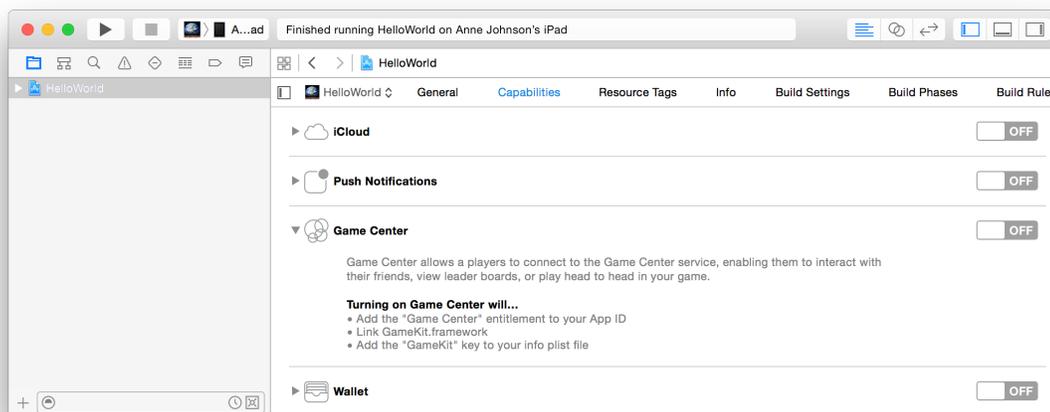
Game Centerを有効にする (iOS、tvOS、Mac)

Game Centerとは、Appleが提供するソーシャルゲームネットワークのことです。プレーヤーは自分のデバイスをGame Centerサービスに接続し、さまざまな情報を交換できます。

Game Centerを使うためには、まずXcodeでこれを有効にする必要があります。

Game Centerを有効にするには

1. 「Capabilities」ペインで「Game Center」が有効になっていない場合は、「Game Center」セクションのスイッチをオンにします。



Xcodeは自動的に、プロビジョニングを施してGame Centerを利用できるようにするとともに、Game Kitフレームワークをプロジェクトに追加します。

Macアプリケーションの場合、Xcodeは「Outgoing」ネットワークエンタイトルメントも設定します。これはXcodeの「Capabilities」ペイン内、「App Sandbox」セクションにあります。アプリケーションがネットワーク接続の監視も行うのであれば、内向きの接続を許可する必要があります。追加のネットワークエンタイトルメントを設定する手順については、「[アプリケーションサンドボックスを設定する \(Mac\)](#)」 (59 ページ) を参照してください。

GameKitを扱うコードの書き方については、『*Game Center Programming Guide*』に解説があります。iTunes Connectでアプリケーションの設定をする手順のうち、(特定App IDを登録した) アプリケーションレコードの作成については『*iTunes Connect Developer Guide*』の“Creating an iTunes Connect Record for an App”、その設定方法については『*Game Center Configuration Guide for iTunes Connect*』を参照してください。

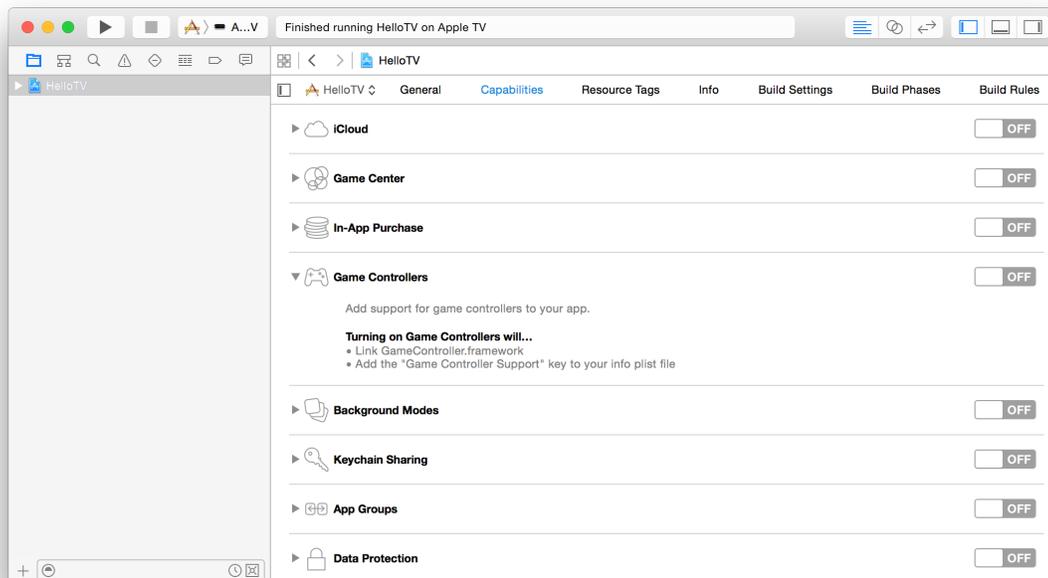
ゲームコントローラを有効にする (tvOS)

ゲームコントローラは、ゲームでアクションをトリガする物理コントロール要素を提供します。

Game Controllerフレームワークを使うためには、Xcodeでゲームコントローラを有効にする必要があります。

ゲームコントローラを有効にするには

1. 「Capabilities」ペインで「Game Controllers」が有効になっていない場合は、「Game Controllers」セクションのスイッチをオンにします。



ゲームコントローラを設定し、Game Controllerフレームワークに関するコードを記述する方法については、『*Game Controller Programming Guide*』を参照してください。

Walletを設定する (iOS、WatchKitエクステンション)

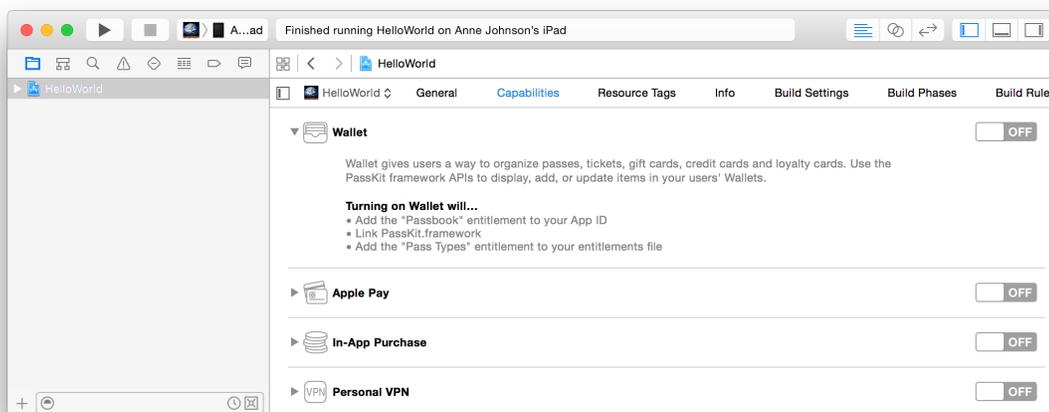
Walletは、実世界における商品やサービスと交換できる、クーポン、演劇の切符、搭乗券などといった情報を電子化して扱えるようにします。使い方はいくつか考えられます。

- パスを作成、配布、更新するためには、パス型識別子を登録し、パス署名証明書を要求する必要があります。アプリケーションやエンタイトルメントは必要ありません。詳しくは『*Passbook Programming Guide*』を参照してください。
- ユーザがアプリケーション上でパスをWalletに追加できるようにするためには、PassKitフレームワークを使います。Walletのエンタイトルメントは必要ありません。
- アプリケーションがユーザのパスにアクセスできるようにするためには、以下に説明する手続きが必要です。

まず、XcodeプロジェクトでWalletを有効にします。

Walletを有効にするには

1. 「Capabilities」ペインで「Wallet」が有効になっていない場合は、「Wallet」セクションのスイッチをオンにします。

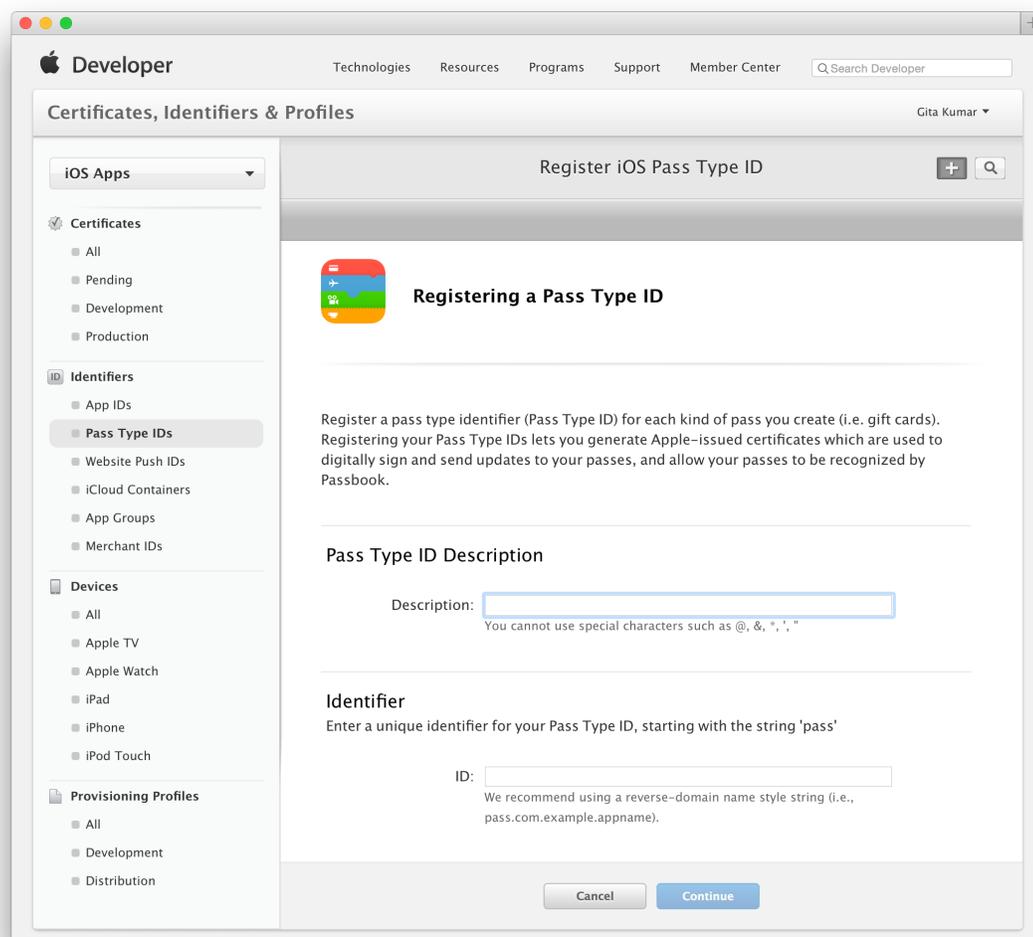


Xcodeは自動的に、プロビジョニングを施してWalletを利用できるようにするとともに、Pass Kitフレームワークをプロジェクトに追加します。

一部のパス型識別子しか扱えないよう制限することも可能です。これは特に、パスを扱うアプリケーションをいくつも開発する場合に有用です。パス型識別子がない場合は、あらかじめ生成してからこの機能を有効にしてください。

パス型識別子を生成するには

1. 「Certificates, Identifiers & Profiles」セクションで「Identifiers」を選択します。
2. 「Identifiers」以下の「Pass Type IDs」を選択します。
3. 右上隅にある「Add」ボタン（「+」の表示）を押します。
4. 説明文とIDを入力し、「Continue」を押してください。



5. 設定内容を確認し、「Register」を押します。
6. 「Done」を押してください。

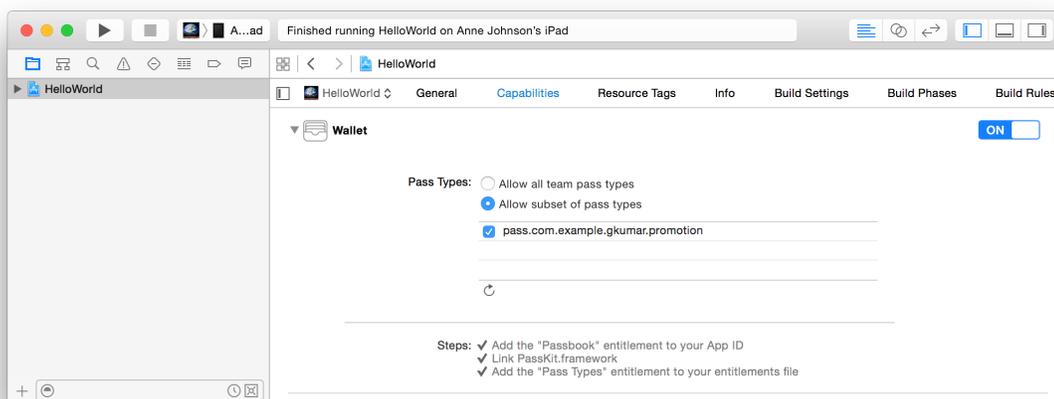
すると、Xcode上で、アプリケーションが扱えるパス型識別子を制限できるようになります。

一部のパス型識別子しか扱えないようアプリケーションを制限するには

1. 「Capabilities」 ペインで、必要に応じて「Wallet」の詳細表示用の三角印をクリックします。
2. 「Allow subset of pass types」をオンにしてください。

メンバーセンターにパス型識別子がなければ、ラジオボタンは「Allow all team pass types」がオンの状態に戻ります。

3. 必要ならば「Pass Types」リストの下にある「Refresh」ボタンを押して、パス型識別子を表示してください。



4. アプリケーションで扱えるようにするパス型識別子を選択します。

アプリケーションでパス型識別子を扱えるようにする手順については、『[Passbook Programming Guide](#)』の“Setting the Pass Type Identifier and Team ID”を参照してください。

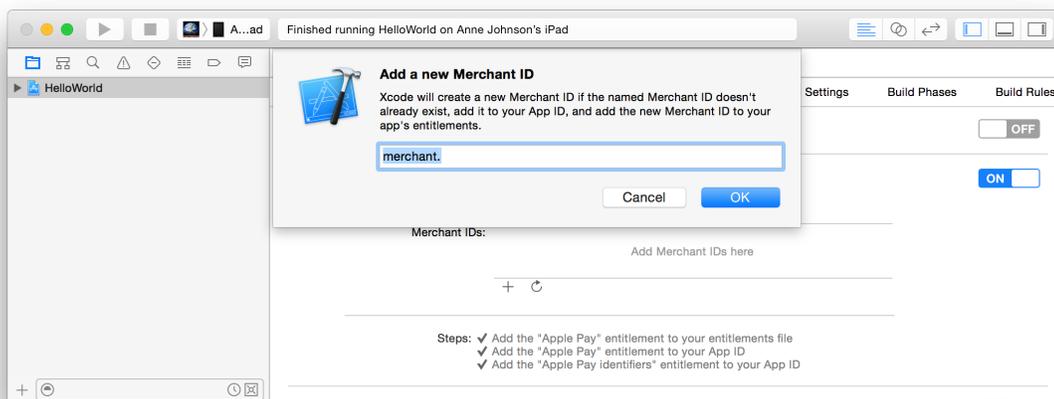
Apple Payを設定する (iOS、WatchKitエクステンション)

Apple Payを使うと、ユーザーはデバイスに格納された支払情報に従って商品とサービスの代金を安全に支払うことができます。

Apple Payを有効にして、マーチャントIDを作成するには

1. 「Capabilities」ペインで「Apple Pay」が有効になっていない場合は、「Apple Pay」セクションのスイッチをオンにします。
2. 「Apple Pay Identifiers」テーブルの下部にある「Add」ボタン（「+」の表示）を押します。

3. 表示されるダイアログで、ID名を入力して「OK」をクリックします。



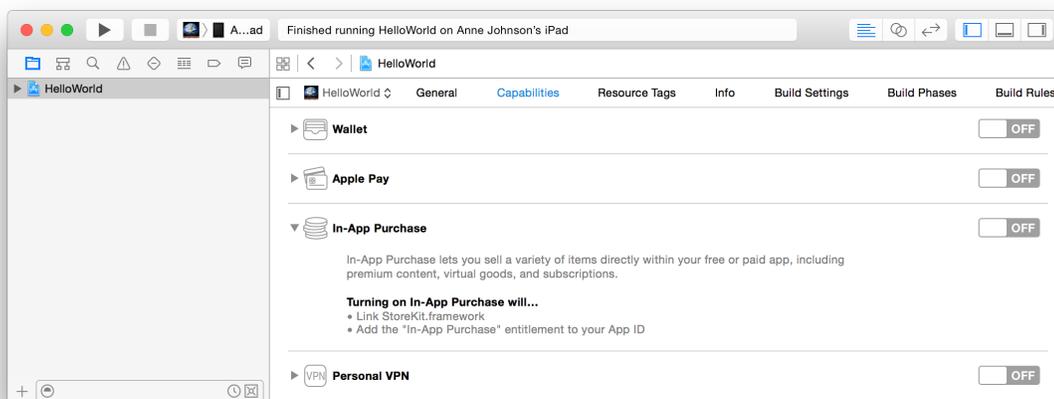
Apple Payについて詳しくは、『[Apple Pay Programming Guide](#)』および『[Pass Kit Framework Reference](#)』を参照してください。

In-App Purchaseを有効にする (iOS、tvOS、Mac)

In-App Purchaseには、ストアの機能をアプリケーションに直接埋め込んで、ユーザがストアに接続し、支払い処理を安全に進められるようにする働きがあります。ユーザに拡張機能や追加のコンテンツを販売し、その代金を集めることができるのです。Xcodeプロジェクトで設定した後、iTunesConnectでの設定も必要です。In-App Purchasesを作成するためにもiTunes Connectを使います。

In-App Purchaseを有効にするには

1. 「Capabilities」ペインで「In-App Purchase」が有効になっていない場合は、「In-App Purchase」セクションのスイッチをオンにします。



Xcodeは自動的に、プロビジョニングを施してIn-App Purchaseを利用できるようにするとともに、Store Kitフレームワークをプロジェクトに追加します。

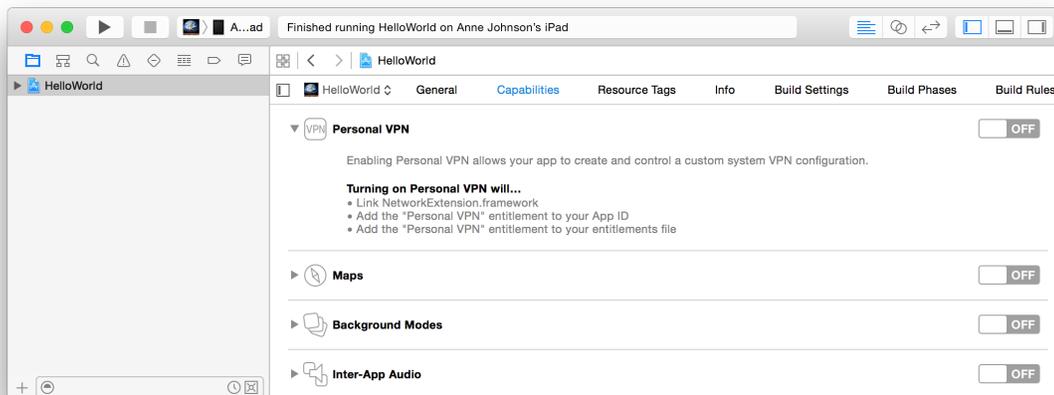
In-App Purchase周辺のコードの書き方については、『*In-App Purchase Programming Guide*』を参照してください。iTunes Connect上でアプリケーションレコードを作成し、特定App IDを入力する手順については、『*iTunes Connect Developer Guide*』の“Creating an iTunes Connect Record for an App”に解説があります。In-App Purchaseを作成してアップロードする方法は、『*In-App Purchase Configuration Guide for iTunes Connect*』を参照してください。

個人向けVPNを有効にする (iOS、Mac)

個人向けVPNを有効にすると、Network Extensionフレームワークを使ってアプリケーションで独自のシステムVPN構成を作成して制御できます。

個人向けVPNを有効にするには

1. 「Capabilities」 ペインで「Personal VPN」が有効になっていない場合は、「Personal VPN」セクションのスイッチをオンにします。



Xcodeは自動的に、プロビジョニングを施して個人向けVPNを利用できるようにするとともに、Network Extensionフレームワークをプロジェクトに追加します。

Network Extensionフレームワークについては、『*Network Extension Framework Reference*』を参照してください。

「マップ(Maps)」を設定する (iOS、WatchKitエクステンション、Mac)

「マップ(Maps)」サービスを利用して、道順情報を取得し、あるいは「マップ(Maps)」アプリケーションにその道順を表示するよう指示することができます。さらに、2地点間の経路情報を処理するiOSアプリケーションは、経路制御アプリケーションとして登録することにより、その情報を「マップ(Maps)」その他のアプリケーションに渡して、表示を委ねることができるようになります。「マップ(Maps)」サービスを有効にするためにはXcodeを使います。iOS用の経路制御アプリケーションの場合、iTunes Connectを使って対応地域カバレッジファイルをアップロードします。

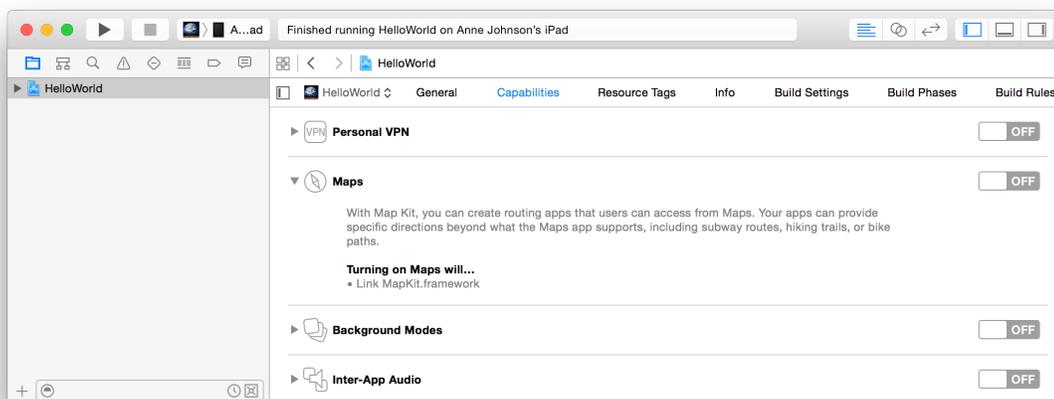
MapKitフレームワークを扱うコードの書き方については、『*Location and Maps Programming Guide*』を参照してください。

Xcodeで「Maps」を有効にする

XcodeプロジェクトでMapsを有効にし、iOS用の経路制御アプリケーションであれば、対応モードをいくつか選択します。

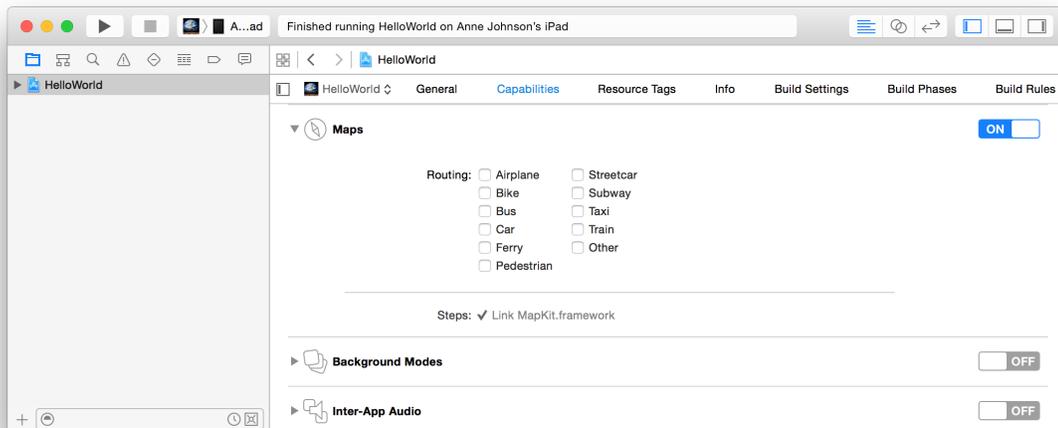
Mapsを有効にし、モードを選択するには

1. 「Capabilities」ペインで「Maps」が有効になっていない場合は、「Maps」セクションのスイッチをオンにします。



2. iOS用の経路制御アプリケーションであれば、その下のチェックボックスで、対応するモードを選択してください。

「Routing」モードは少なくとも1つ選ばなければなりません。



経路制御アプリケーションを設定する (iOS、WatchKitエクステンション)

2地点間の道順情報を他のアプリケーションに提供するアプリケーションには、追加の設定作業が必要になります。先に進む前に、実施すべき作業を確認しておきましょう。

	作業内容
<input checked="" type="checkbox"/>	Xcodeで「Maps」を有効にする。
<input checked="" type="checkbox"/>	Xcodeで対応するモードを選択する。
<input type="checkbox"/>	経路情報を処理するコードを記述する。
<input type="checkbox"/>	アプリケーションレコードを生成し、必要に応じて対応地域カバレッジファイルをアップロードする。
<input type="checkbox"/>	アプリケーションのビルドをストアにアップロードする。
<input type="checkbox"/>	必要ならば対応地域カバレッジファイルをアップロードする。

経路情報を処理する

経路制御アプリケーションの作成方法については、『*Location and Maps Programming Guide*』の“Registering as a Routing App” in *Location and Maps Programming Guide*を参照してください。

iTunes Connectでアプリケーションレコードを作成する

iTunes Connectでアプリケーションレコードを作成する手順については、『*iTunes Connect Developer Guide*』の“Creating an iTunes Connect Record for an App”を参照してください。経路制御アプリケーションには、アプリケーションが扱う地域を定義する、対応地域カバレッジファイルが必要です。対応地域カバレッジファイルをアップロードするのは、アプリケーションレコードの作成時でも、ビルドをアップロードした後も構いません（“対応地域カバレッジファイルをiTunes Connectにアップロードする”を参照）。

ビルドをストアに登録する

ビルドをiTunes Connectにアップロードする手順については、“[アプリケーションの登録](#)”（149 ページ）を参照してください。

対応地域カバレッジファイルをiTunes Connectにアップロードする

経路制御アプリケーションを登録しても、対応地域カバレッジファイルがアップロードされるまで、Appleは認定手続きを始めません。

アプリケーションのアップロード後に対応地域カバレッジファイルをアップロードするには

1. [iTunes Connect](#)にサインインしてください。
2. iTunes Connectホームページの「My Apps」をクリックします。
3. 編集するアプリケーションを見つけ、大きなアイコンまたはアプリケーション名をクリックしてください。
「Versions」ペインが表示されます。
4. 編集するアプリケーションの版をクリックします。
版情報が下に表示されます。
5. 「General App Information」セクションまでスクロールします。

- 「Routing App Coverage File」の下にある「Choose File」ボタンを押します。

The screenshot shows the 'Gita's HelloWorld Version 1' app page in App Store Connect. The 'Routing App Coverage File' section is highlighted with a blue border. It contains a 'Choose File' button and the text '(Optional)'. Other sections include 'App Icon', 'Copyright', 'Trade Representative Contact Information', 'Gita Kumar' (with fields for first name, last name, address, city, state, zip, phone number, and email), 'Apple ID' (877829200), 'Version' (1), 'Category' (Education), 'Rating' (Ages 4+), and 'License Agreement' (Apple's Standard License Agreement).

- ファイルを選択し、「Choose」をクリックします。

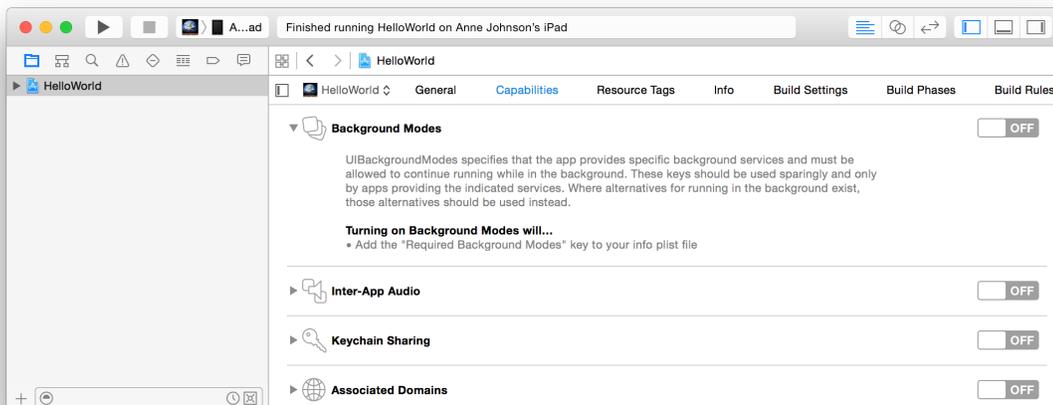
ファイルの書式が正しくないか、ファイル拡張子に誤りがある場合は、エラーメッセージが表示されます。

バックグラウンドモードを設定する (iOS、tvOS)

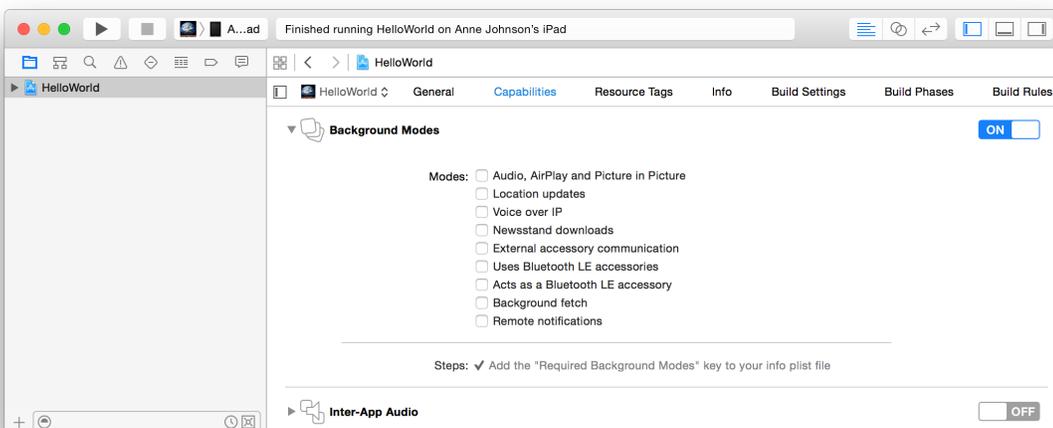
バックグラウンドモードを有効にすると、アプリケーションはバックグラウンドで実行を続けられるようになります。

バックグラウンドモードを有効にするには

1. 「Capabilities」 ペインで「Background Modes」 が有効になっていない場合は、「Background Modes」セクションのスイッチをオンにします。



2. 必要に応じ、その下のチェックボックスで、対応するモードを選択してください。



Xcodeはバックグラウンドモードを情報プロパティリストに追加します。

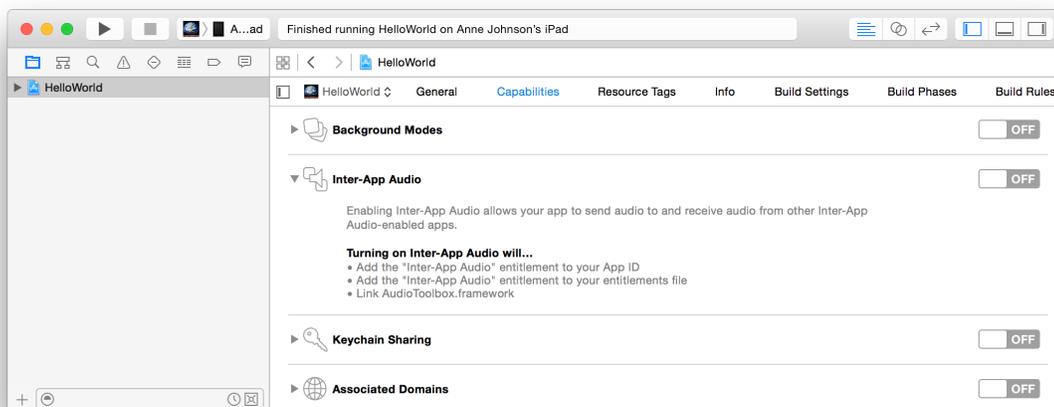
バックグラウンドモードの選択方法については、『*iOS App Programming Guide*』の“App States and Multitasking”を参照してください。

Inter-App Audioを有効にする (iOS)

Inter-App Audioには、オーディオをエクスポートして、他のアプリケーションも使えるようにする働きがあります。

Inter-App Audioを有効にするには

1. 「Capabilities」 ペインで「Inter-App Audio」が有効になっていない場合は、「Inter-App Audio」セクションのスイッチをオンにします。



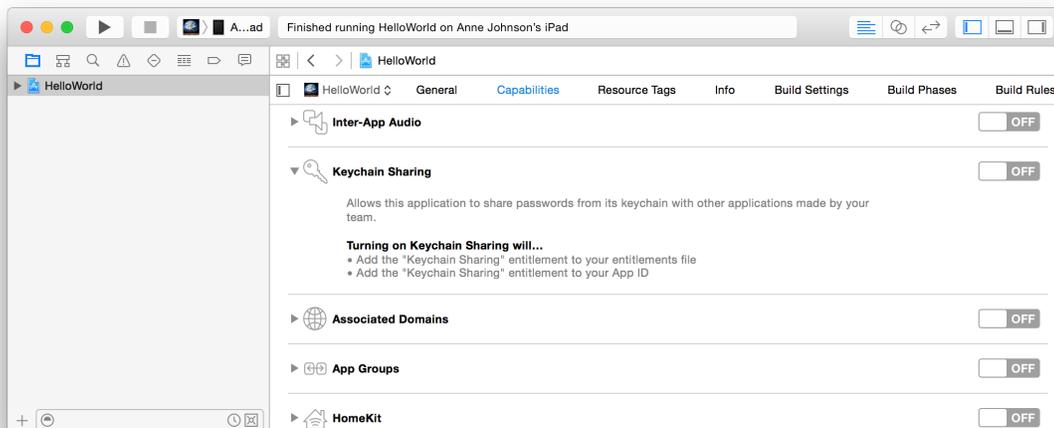
Xcodeは自動的に、プロビジョニングを施してInter-App Audioを利用できるようにするとともに、Audio Toolboxフレームワークをプロジェクトに追加します。Audio Toolboxフレームワークを扱うコードの書き方については、『[Audio Toolbox Framework Reference](#)』を参照してください。

キーチェーン共有の設定をする

キーチェーンの共有を有効にすると、ここに登録されたパスワードを、チーム内で開発した他のアプリケーションと共有できるようになります。

キーチェーンの共有を有効にするには

1. 「Capabilities」 ペインで「Keychain Sharing」が有効になっていない場合は、「Keychain Sharing」セクションのスイッチをオンにします。

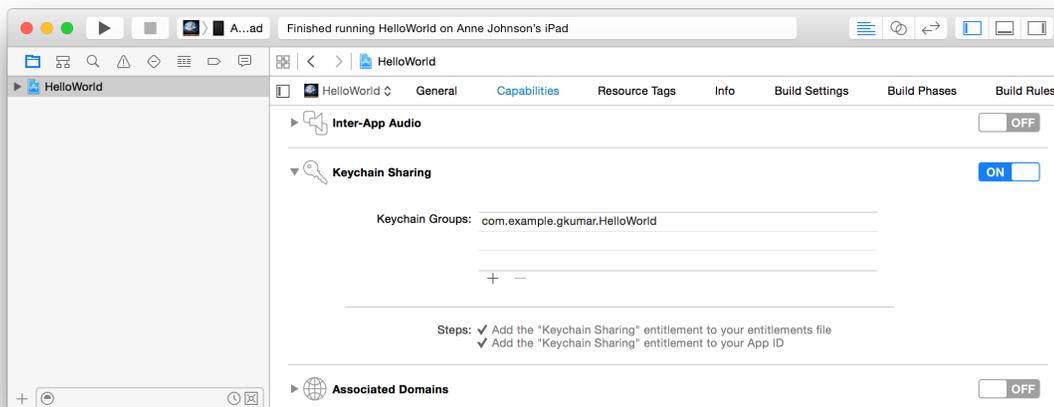


Xcodeは「keychain-access-groups」キーをエンタイトルメントファイルに追加します。

必要ならば、共有する範囲を、いくつかのキーチェーンアクセスグループに制限することも可能です。

共有する範囲をいくつかのキーチェーンアクセスグループに制限するには

1. 「Capabilities」 ペインで、必要に応じて「Keychain Sharing」の詳細表示用の三角印をクリックします。
2. 「Keychain Groups」領域の下部にある「Add」ボタン（「+」の表示）を押します。
3. テーブル内のプレースホルダテキストをダブルクリックして、追加するキーチェーンアクセスグループを入力します。



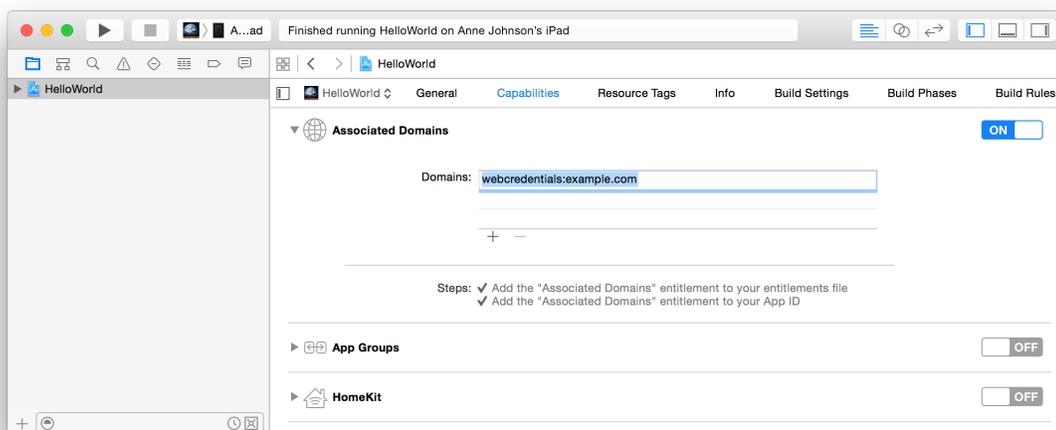
キーチェーンアクセスグループを削除する場合は、「Keychain Groups」領域から該当するものを選び、「Delete」ボタン（「-」の表示）を押してください。

Associated Domainsを有効にする (iOS、WatchKitエクステンション)

Associated Domainsを有効にするのは、アプリケーションをドメインに関連付けて、特定のサービス（Safariに保存されたパスワードやアクティビティの継続など）にアクセスする場合です。

Associated Domainsを有効にするには

1. 「Capabilities」ペインで「Associated Domains」が有効になっていない場合は、「Associated Domains」セクションのスイッチをオンにします。
2. 「Domains」テーブルの下部にある「Add」ボタン（「+」の表示）を押します。
3. テーブル内のプレースホルダテキストをダブルクリックして、追加するドメイン名を入力します。



アプリケーショングループを設定する

アプリケーショングループを使うと、複数のアプリケーションから共有コンテナにアクセスできるほか、アプリケーションどうしのプロセス間通信もできます。アプリケーショングループを有効にするには、「Capabilities」ペインで「App Groups」セクションのスイッチをオンにします。テーブルから既存のアプリケーショングループを選択しても、新規のアプリケーショングループを追加しても構いません。

アプリケーショングループを選択または選択解除するには

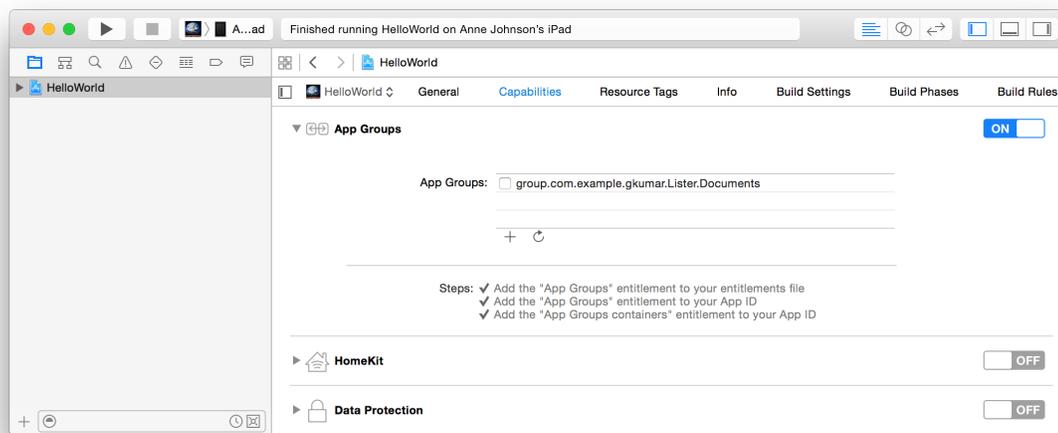
1. 必要な場合は、テーブルの下にある「Refresh」ボタンをクリックして、メンバーセンターからコンテナIDをダウンロードします。
2. 該当するコンテナIDの行にあるチェックボックスをオン（そのコンテナを使用する場合）またはオフ（そのコンテナを使用しない場合）にします。

Xcodeにより、Xcodeプロジェクトのエンタイトルメントファイル内にあるアプリケーショングループのリストが更新されます。

既存のアプリケーショングループが十分でない場合は、アプリケーショングループを別途作成します。

アプリケーショングループを作成するには

1. 「App Groups」テーブルの下部にある「Add」ボタン（「+」の表示）を押します。



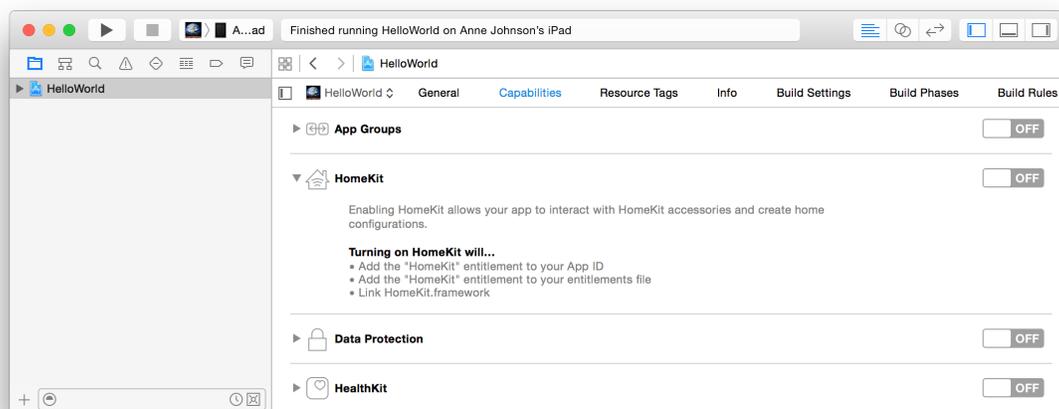
2. 表示されるダイアログで、テキストフィールドにコンテナIDを入力して「OK」をクリックします。

アプリケーショングループのコンテナIDは、「group.」で始まり、その後に逆DNS形式の文字列が続きます。

HomeKitを追加する (iOS、WatchKitエクステンション)

HomeKitを使うと、アプリケーションとユーザの自宅に接続されたアクセサリとの通信が可能になり、アプリケーションからアクセサリを制御できます。自宅に新しいアクセサリを導入することで、接続性がよくなるだけでなく、使い勝手も改善されます。HomeKitは、これらのアクセサリと通信し、自宅の構成を確立するための標準的な方法です。HomeKit Accessory Simulatorでは、HomeKitアプリケーションとシミュレートされたアクセサリとの通信のテストが実施されます。□

HomeKitエンタイトルメントとHomeKitフレームワークをプロジェクトに追加するには、「HomeKit」セクションのスイッチをオンにします。

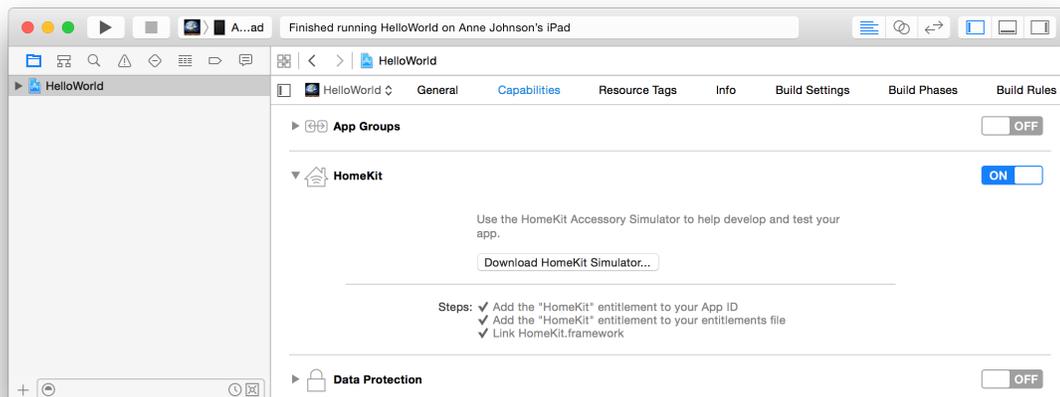


HomeKit Accessory SimulatorはXcodeで配布されません。

HomeKit Accessory Simulatorをダウンロードするには

1. 「Download HomeKit Accessory Simulator」をクリックします。

または、「Xcode」 > 「Open Developer Tool」 > 「More Developer Tools」の順に選択します。



2. ブラウザで「Hardware IO Tools for Xcode」DMGファイルを検索してダウンロードします。
3. Finderで、~/Downloads内のDMGファイルをダブルクリックします。
4. HomeKit Accessory Simulatorを/Applicationsフォルダまでドラッグします。

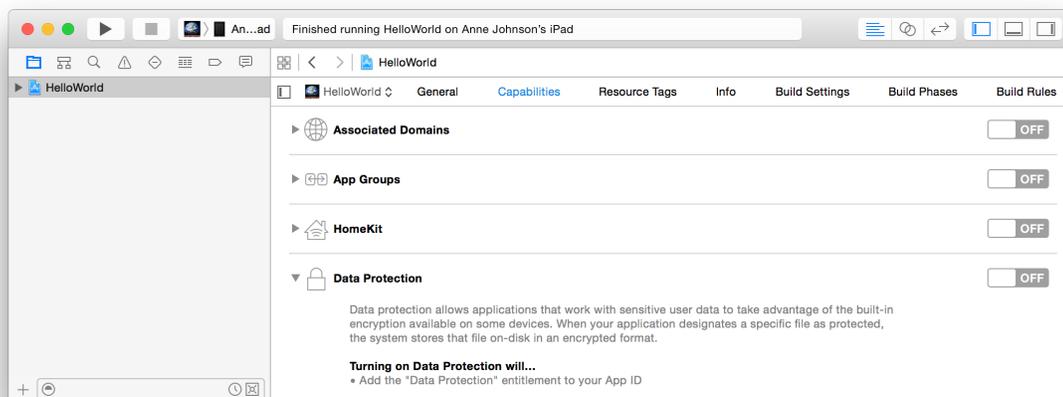
HomeKitフレームワークについては、『[HomeKit Framework Reference](#)』を参照してください。

データ保護を有効にする (iOS、WatchKitエクステンション、tvOS)

データ保護の機能には、アプリケーションがディスクに保存するファイルのセキュリティを強化する働きがあります。デバイスに暗号化ハードウェアが搭載されていれば、これを利用してディスク上のファイルを暗号化します。データ保護の機能をアプリケーションに組み込むためには、プロビジョニングが必要です。

データ保護を有効にするには

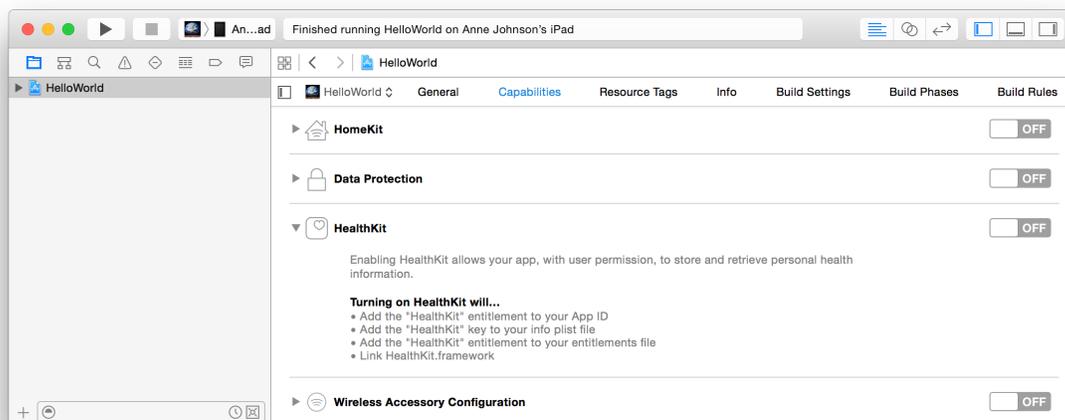
1. 「Capabilities」 ペインで「Data Protection」が有効になっていない場合は、「Data Protection」セクションのスイッチをオンにします。



デフォルトの保護レベルは「完全保護」で、ファイルを暗号化し、デバイスがロックされている間はアクセスをできないようにします。ファイルの保護レベルをプログラムで設定する方法については、『*iOS App Programming Guide*』の“Protecting Data Using On-Disk Encryption” in *iOS App Programming Guide* を参照してください。

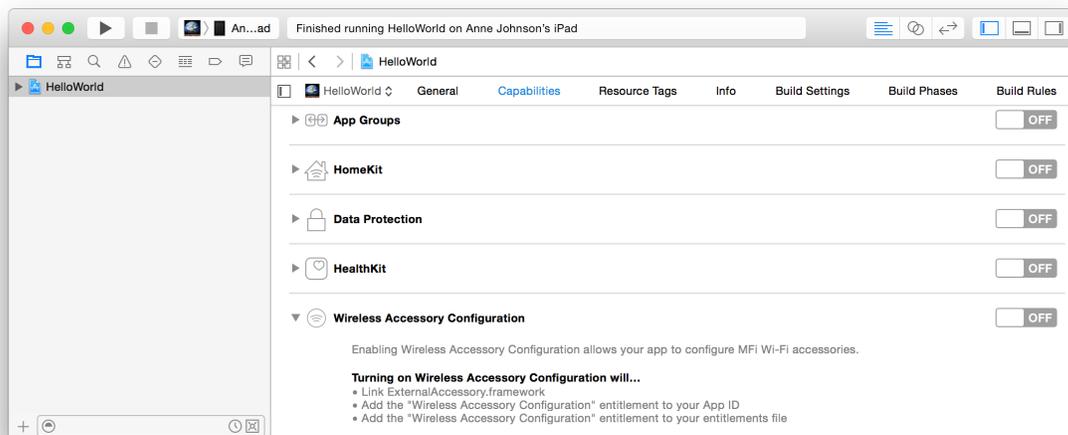
HealthKitを追加する (iOS、WatchKitエクステンション)

HealthKitを使うと、ユーザ権限が付与されたアプリケーションで個人の医療情報を取得および格納できます。このエンタイトルメントをApp IDに追加し、HealthKitフレームワークをプロジェクトに追加するには、「HealthKit」セクションのスイッチをオンにします。



ワイヤレスアクセサリ設定の有効化 (iOS)

ワイヤレスアクセサリ設定を有効にすると、External Accessoryフレームワークがプロジェクトに追加され、アプリケーションでMFi Wi-Fiアクセサリを設定できます。ワイヤレスアクセサリ設定を有効にするには、「Wireless Accessory Configuration」セクションのスイッチをオンにします。Xcodeから、エンタイトルメントファイルとApp IDの両方にエンタイトルメントが付与されます。



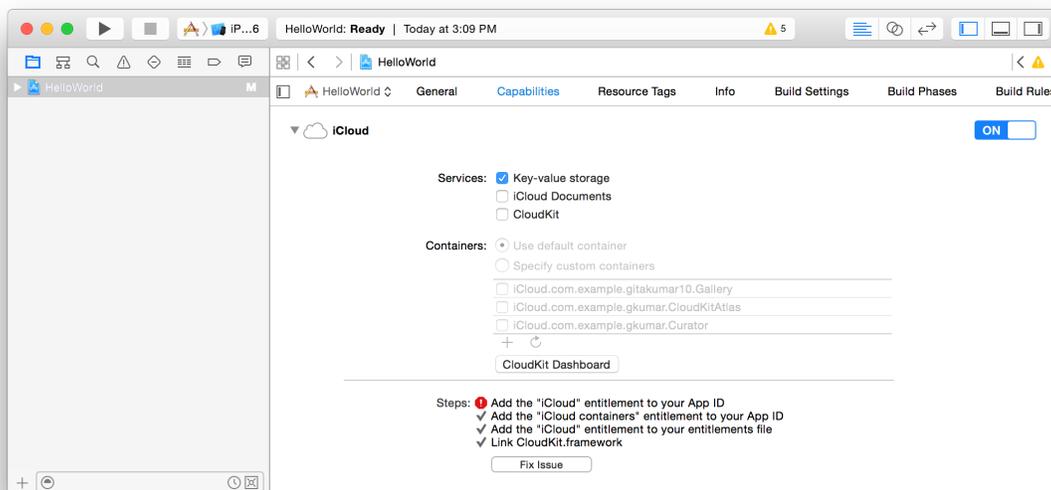
External Accessoryフレームワークについては、『[External Accessory Programming Topics](#)』を参照してください。

Newsstandを設定する (iOS、WatchKitエクステンション)

Newsstandには、ユーザーが購読している雑誌や新聞を、所定のフォルダに配送する働きがあります。Newsstandの機能は、情報プロパティリストにいくつかキーを追加し、Xcodeプロジェクトにアートワークを追加すると利用できるようになります。Newsstandアプリケーションの作成方法については、『[Newsstand for Developers](#)』を参照してください。また、Newsstandの表紙アイコンをXcodeプロジェクトに追加する方法は、『[iOS Human Interface Guidelines](#)』の「Newsstand Icons」 in *iOS Human Interface Guidelines* を参照してください。

トラブルシューティング

ここで説明したアプリケーションサービスを有効にする上で問題があると、プロジェクトエディタの「Steps」の下にエラーメッセージが表示されます。その内容を確認した上で、「Fix Issue」を押して対処してください。開発用証明書および「Scheme」ポップアップメニューから選択したiOSデバイス（iOS/tvOS/watchOSアプリケーションの場合）があれば、Xcodeはチームプロビジョニングプロファイルを生成できます。



まとめ

この章では、アプリケーションをXcode上で設定する手順を説明しました。メンバーセンターやiTunes Connectでの作業が発生する場合があります。

デバイス上でアプリケーションを起動する手順

iOS/tvOS/watchOSアプリケーションはすべて、デバイス上で起動できるようにするために、コード署名およびプロビジョニングを要します。Macアプリケーションも、所定のアプリケーションサービスを利用していれば、同様にコード署名が必要です。**プロビジョニング**とは、アプリケーションがデバイス上で起動し、特定のアプリケーションサービスを利用できるように、準備、設定する手続きのことです。Xcodeは与えられた情報に基づき、Xcodeプロジェクトをチームに対応づけた時点、あるいはキーとなる技術やサービスを初めてアプリケーションに組み込んだ時点で、チームプロビジョニングプロファイルを生成します。また、開発用証明書を自動的に生成する、接続されたデバイスまたはMacを登録する、といった処理も行います。Xcodeはこの情報を使って、プロビジョニングプロファイルを生成します。これは最終的に、デバイスにインストールされることとなります。iOS/tvOS/watchOSの場合、デバイス上でアプリケーションを実行するためには、当該デバイスがプロビジョニングプロファイルに登録されている必要があります。同様に、特定のアプリケーションサービスを組み込んだMacアプリケーションを起動するためには、Macがプロビジョニングプロファイルに登録されていなければなりません。

無料のApple IDアカウントでも、デバイス上でアプリケーションを起動できますが、利用可能なキーパビリティは、プラットフォームや、メンバーとして属するApple Developer Programによって異なります（[“サポートされるキーパビリティ”](#)（258 ページ）を参照）。



Tip: 後でアプリケーションを配布する際に問題が生じないように、デバッグナビゲータ（『[Debug Navigator Help](#)』を参照）、テストナビゲータ（『[Test Navigator Help](#)』を参照）、Instrumentsアプリケーション（『[Instruments User Guide](#)』を参照）を利用し、デバイス上でテストしてください。

Macアプリケーションを起動する

Macアプリケーションは、プロジェクトナビゲータの「Run」ボタンで起動します。署名IDが「None」であれば、Xcodeはそのままアプリケーションを起動します。そうでなければ、XcodeはMacを登録し、チームプロビジョニングプロファイルに追加した上で、アプリケーションを起動します。codesign コマンドが、キーチェーンにある鍵を使って署名できるようにするかどうか訊ねられた場合は、「Always Allow」を押してください。

iOS/tvOS/watchOSアプリケーションを起動する

コード署名IDおよびチームプロビジョニングプロファイルが生成済みであれば（“[チームプロビジョニングプロファイルを生成する](#)”（39 ページ）を参照）、ごく簡単な手順で、デバイス上でアプリケーションを起動できるようになります。生成済みでなければ、起動の過程でコード署名に関する問題を解決するため、いくつかダイアログや警告が現れることとなります。

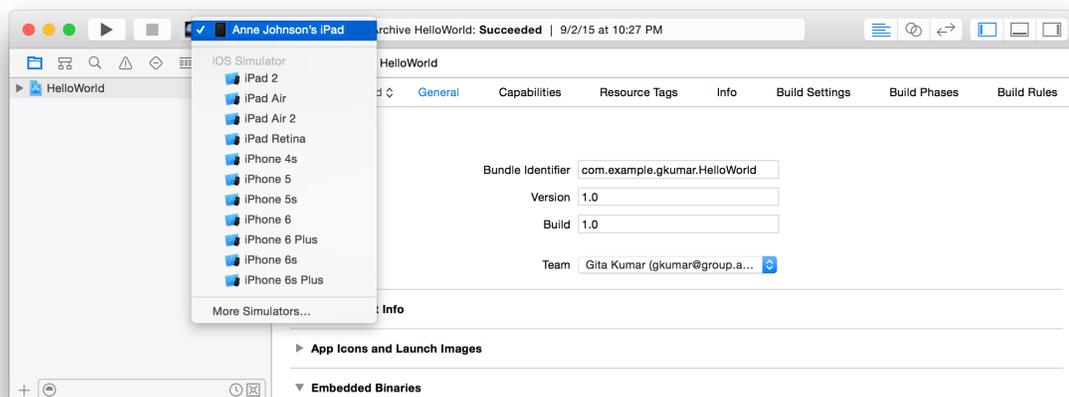
デバイス上でアプリケーションを起動するには

1. デバイスをMacに接続します。

watchOSに関する注意: Apple WatchとペアリングしたiPhoneをMacに接続してください。

2. プロジェクトナビゲータで、「Scheme」 ツールバーメニューから該当するデバイスを選択します。

Xcodeは、このデバイスを開発用に使うものと想定し、自動的に登録します。



デバイスが適切でないと判断され、「Scheme」 ツールバーメニューで無効になっている場合は、この問題を解消してから次に進んでください。「Ineligible Devices」以下の該当するデバイス上にマウスカーソルを置くと、適切でない理由が表示されます。たとえば、デバイスが配備ターゲットに合致しない場合、当該デバイスのOSの版をアップグレードするか、ターゲットにしたい版を「Deployment Target」ポップアップメニュー（「Deployment Info」セクション）から選択してください。次いで、「Scheme」 ツールバーメニューからデバイスを選択します。

3. 「Run」 ボタンを押してください。

Xcodeはアプリケーションをデバイスにインストールし、起動します。

4. `codesign` コマンドが、キーチェーンにある鍵を使って署名できるようにするかどうか訊ねられた場合は、「Always Allow」を押してください。

開発期間中に、アプリケーションが動作すると謳う、あらゆるデバイスとOSの版の組み合わせでテストしてください。シミュレータにはさまざまなinstrumentが付属しているので、アプリケーションを配布する前に、Instrumentsその他のツールを使い、シミュレータ上でもテストしなければなりません。デバイスシミュレータでアプリケーションをテストする手順については、『*iOS Simulator User Guide*』を参照してください。

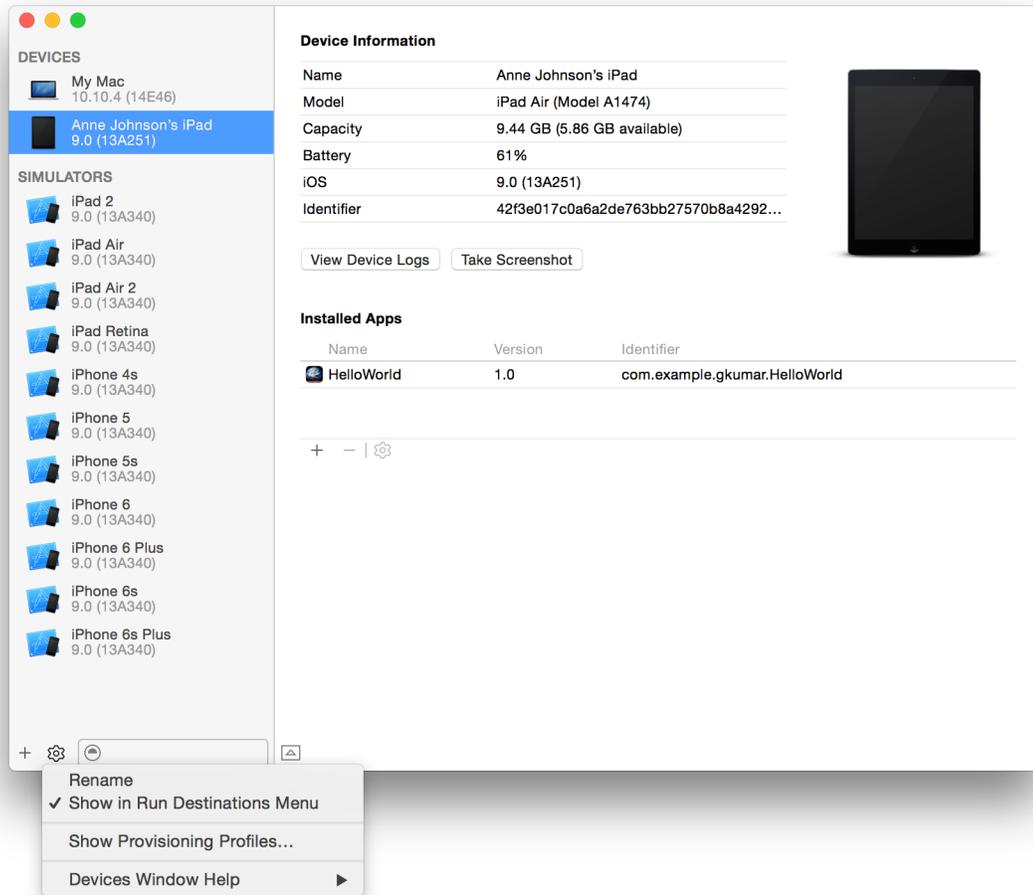
デバイスを「Scheme」メニューから削除する (iOS、tvOS、watchOS)

Xcodeで接続済みのデバイスが無視されるようにする (Xcodeでそのデバイスがチームプロビジョニングプロファイルに追加されないようにする) 場合は、「Scheme」メニューから該当するデバイスを削除します。

「Scheme」メニューからデバイスを削除するには

1. デバイスをMacに接続します。
2. Xcode上で、「Window」 > 「Devices」の順に選択した後、「Devices」から該当するデバイスを選択します。

3. 「Devices」 ウィンドウの左下隅にある「Action」 ボタン（「Add」 ボタンの右にある歯車アイコン）をクリックします。



4. ポップアップメニューから「Show in Run Destinations Menu」を選択解除します。
プロジェクトエディタの「Scheme」メニューから該当するデバイスが表示されなくなります。

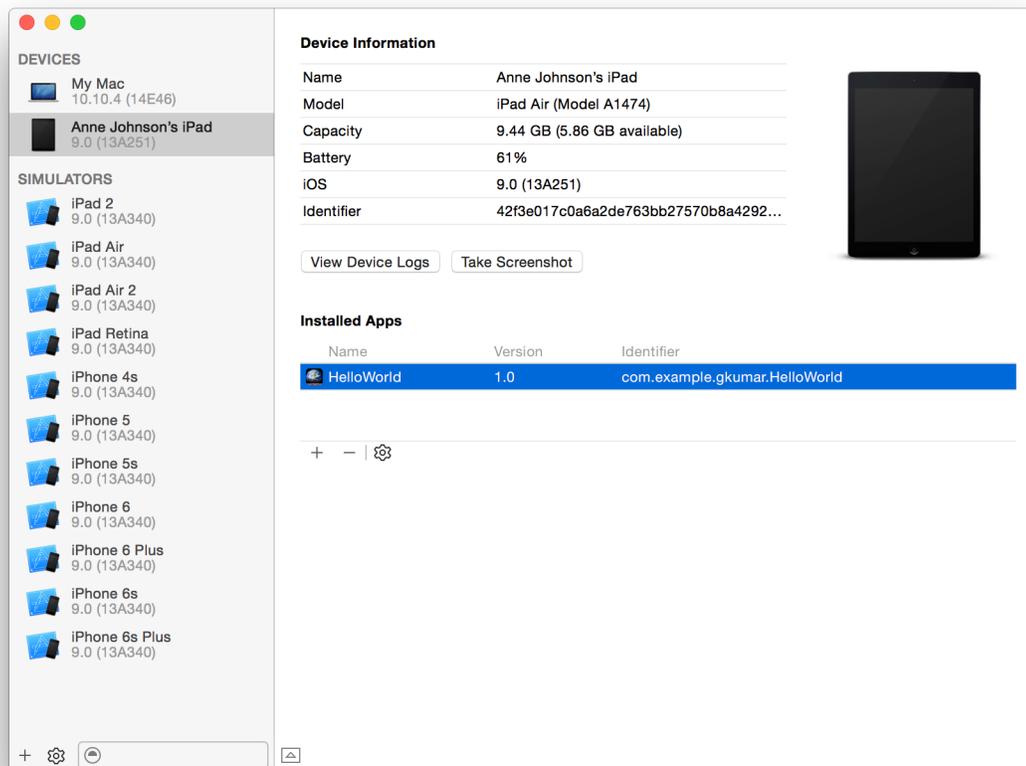
アプリケーションをデバイスから削除する (iOS、tvOS、watchOS)

デバイス上でアプリケーションを起動すると、Xcodeにより、そのアプリケーションがデバイスに組み込まれます。後で、Xcodeの「Devices」 ウィンドウを使ってそのアプリケーションをデバイスから削除できます。

アプリケーションをデバイスから削除するには

1. デバイスをMacに接続します。
2. 「Window」 > 「Devices」 の順に選択した後、「Devices」 から該当するデバイスを選択します。

3. 「Installed Apps」セクションで、テーブルから該当するアプリケーションを選択し、テーブルの左下隅にある「Delete」ボタン（「-」の表示）をクリックします。



4. 表示される確認ダイアログで「削除」をクリックします。

ここまでの結果を確認する

Xcodeがアプリケーションにプロビジョニングを施す方法は、[メンバーセンター](#)に登録されたチームプロビジョニングプロファイルで調べてください。デバイスまたはMacが登録済みであること、チームプロビジョニングプロファイルに追加されていることを確認できます。

デバイスが登録済みであることを確認するには

1. 「Certificates, Identifiers & Profiles」以下の「Devices」を選択します。
2. 「Devices」以下の「All」を選択してください。

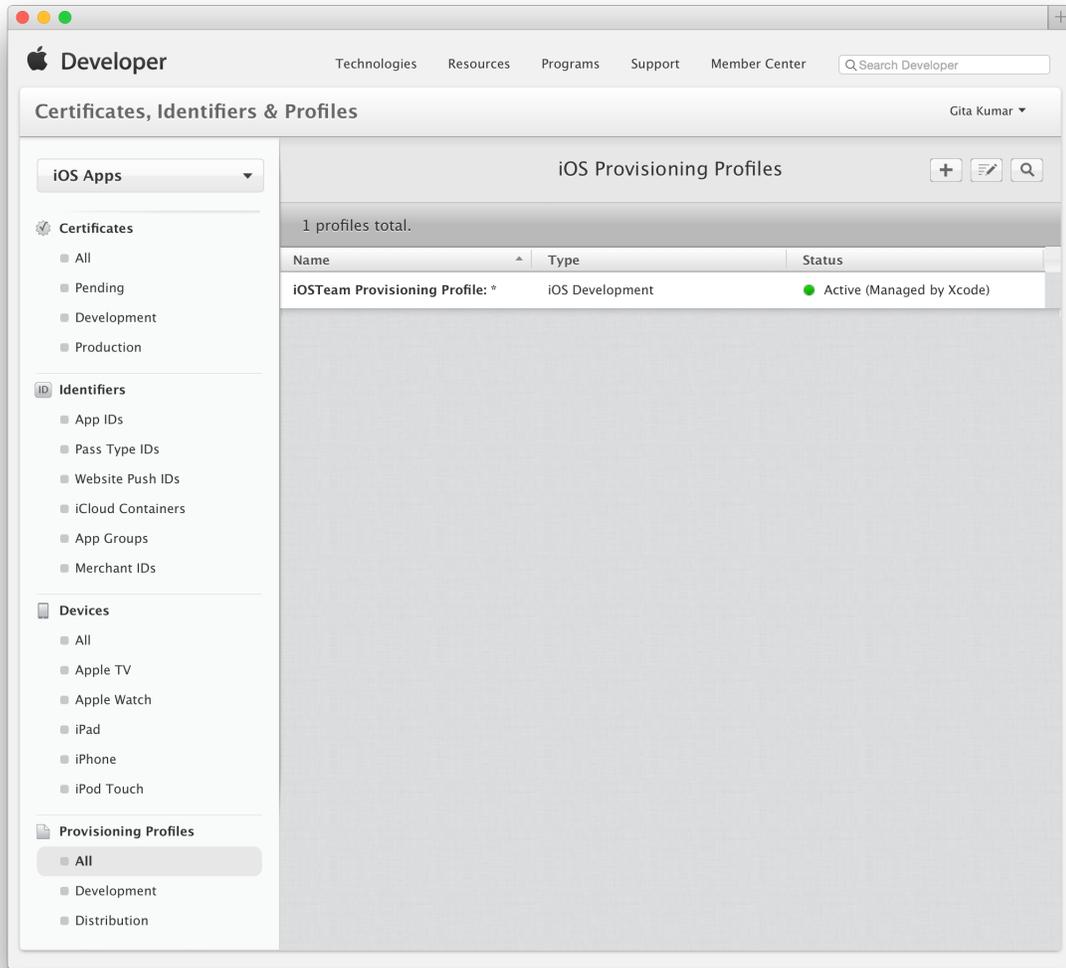
登録済みのデバイスが、有効な状態で列挙されるはずですが、有効になっているデバイスは黒、無効になっているデバイスは灰色の表示です。



デバイスがチームプロビジョニングプロファイルに登録されていることを確認するには

1. 「Certificates, Identifiers & Profiles」の画面で「Provisioning Profiles」を選択します。
2. 「Provisioning Profiles」セクションで「All」を選択してください。

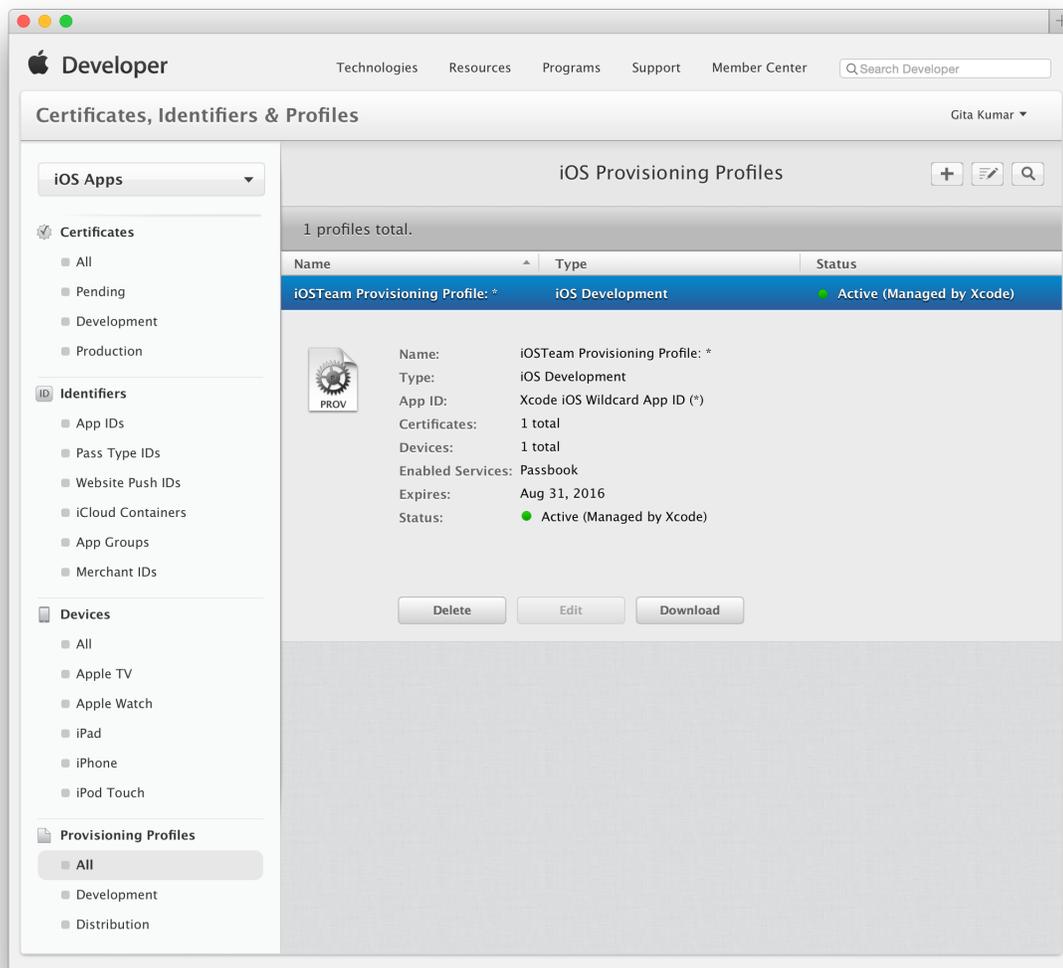
「iOS Team Provisioning Profile」または「Mac Team Provisioning Profile」で始まる名前のチームプロビジョニングプロファイルが、「iOS Provisioning Profiles」または「Mac Provisioning Profiles」以下に表示されます。



注意: 追加した技術やサービス、あるいはアプリケーションの数によっては、チームプロビジョニングプロファイルが複数あるかも知れません。

3. 該当するチームプロビジョニングプロファイルをクリックすると、その詳細表示になります。

その中にApp IDの表示があります。たとえば、iOSアプリケーションならば「Xcode iOS Wildcard App ID」、Macアプリケーションならば「Xcode Mac Wildcard App ID」です。「iOS Provisioning Profiles」の表示例を図に示します。



App IDの下に、プロビジョニングプロファイルに登録されている開発用証明書やデバイスの数が表示されています。これが、アカウントに登録されている開発用証明書の総数や、有効なデバイスの数と一致していなければなりません。個人開発者であれば、開発用証明書の数は1であるはずですが。

トラブルシューティング

デバイス上でアプリケーションが起動しない場合、その原因はいくつか考えられます。

- デバイスが「Scheme」ツールバーメニューで無効な状態になっている場合は、この問題を解消してから次に進んでください。

「Ineligible Devices」以下の該当するデバイス上にマウスカーソルを置くと、適切でない理由が表示されます。

- プロジェクトエディタの「General」ペイン、「Team」ポップアップメニューの下に警告やエラーメッセージが現れた場合は、「[チームプロビジョニングプロファイルを生成する](#)」（39 ページ）の手順に従い、問題点を解消してください。
- 「Capabilities」ペインに、プロビジョニングに関する同様のエラーメッセージが表示された場合は、「[技術やサービスを組み込む](#)」（56 ページ）を参照して、問題を解消する必要があります。

この章で説明した手順を進めてきた場合は、Xcodeのプロジェクト設定を確認するとよいでしょう。プロジェクトとデバイスの設定が合致していないかも知れません。

- さらに、配備ターゲットよりも、デバイスにインストールされたOSソフトウェアの版の方が新しいことを、「[配備ターゲットを設定する](#)」（43 ページ）に従って確認してください。

チームプロビジョニングプロファイルの詳細

チームプロビジョニングプロファイルとは、Xcodeが自動的に管理する、開発用プロビジョニングプロファイルのことです。開発用プロビジョニングプロファイルには、開発中、デバイス上でアプリケーションを起動するとともに、特定のアプリケーションサービスを利用できるようにする働きがあります。個人開発者の場合、チームプロビジョニングプロファイルがあれば、署名済みのアプリケーションがすべて、登録デバイス上で動作するようになります。組織の場合、チームが開発したあらゆるアプリケーションは、どのチームメンバーが署名しても、チームのあらゆるデバイス上で動作するようになります。

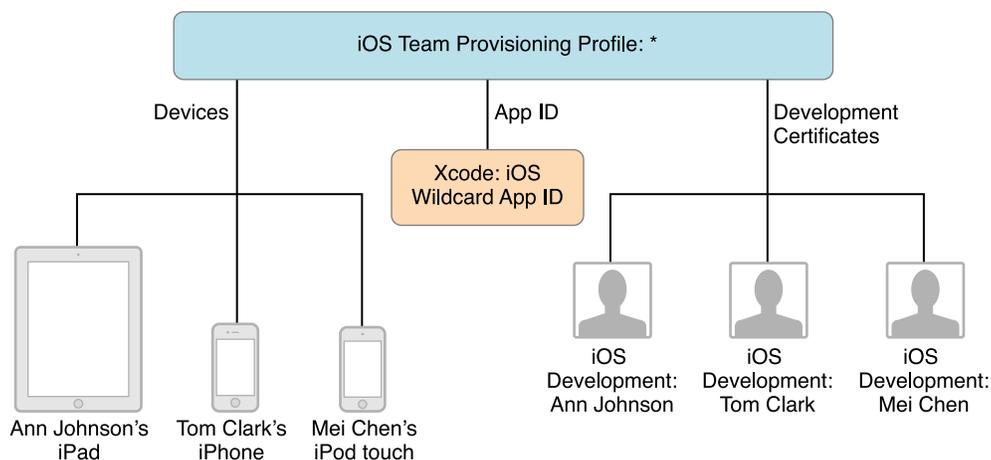
チームプロビジョニングプロファイルは次の要素から成ります。

- チームが開発するアプリケーションに合致するワイルドカードAppIDまたはある特定のアプリケーションに合致する特定App ID
- チームに関連づけて管理するすべてのデバイス
- チームに関連づけて管理するすべての開発用証明書

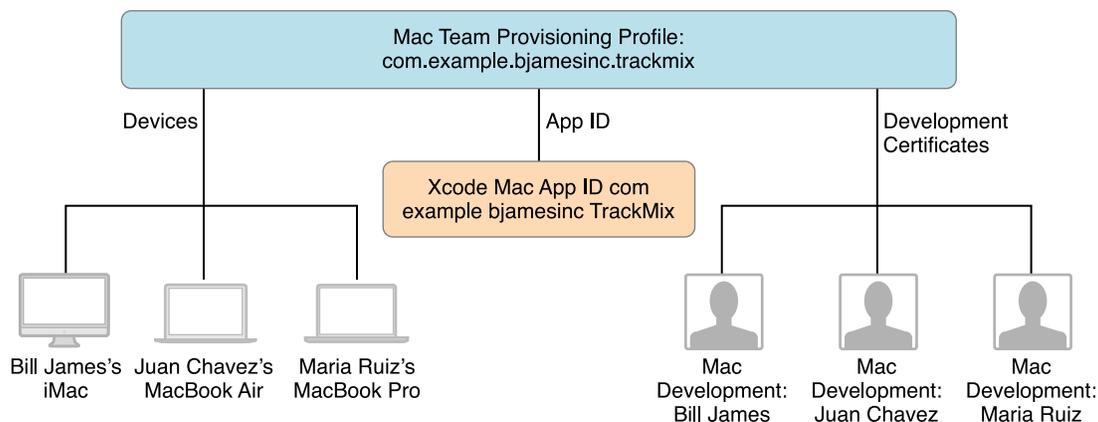
Xcodeは必要に応じて、アプリケーションの設定や、これに組み込んだ技術やサービスに基づき、App IDやチームプロビジョニングプロファイルを生成します。Xcodeは、チームメンバー全員のデバイスや開発用証明書を収集し、チームプロビジョニングプロファイルに追加するようになっています。以降、Xcode上で別のデバイスを登録し、開発用証明書を作成し、あるいはApp IDを変更すると、このチームプロビジョニングプロファイルが更新されます（ただし、メンバーセンターでチームアセットに変更を施しても、自動的にチームプロビジョニングプロファイルが更新されることはありません）。

開発中、アプリケーションがワイルドカードApp IDを使える場合、XcodeはワイルドカードApp IDを含むチームプロビジョニングプロファイルを生成します。たとえば、Xcodeは「iOS Team Provisioning Profile: *」または「Mac Team Provisioning Profile: *」という名前のチームプロビジョニングプロファイルを、やはりここで作成するワイルドカードApp IDを使って作成します。一部のアプリケーションサービスでワイルドカードApp IDを使用できます。ただし、特定App IDを要するケイパビリティ（iCloud、Game Center、In-App Purchaseなど）を組み込んだ場合、Xcodeは特定App IDとそれに対応するチームプロビジョニングプロファイルを生成します。たとえばXcodeは、「iOS Team Provisioning Profile:」または「Mac Team Provisioning Profile:」という文字列の後にバンドルIDを連結した名前で、チームプロビジョニングプロファイルを生成します。特定App IDはプロジェクトのバンドルIDと完全に一致するので、バンドルIDごとに1つしか登録できません。したがって、既にメンバーセンターに特定App IDがあれば、Xcodeは改めて生成することなく、そのままチームプロビジョニングプロファイルに追加して使います。

3人のチームメンバーから成る組織を例として、ワイルドカードApp IDを組み込んだ、「iOS Team Provisioning Profile」の構成を図に示します。



同様に次の図は、Macアプリケーションを開発する組織向けの、特定App IDを使った「Mac Team Provisioning Profile」の構成を表します。



まとめ

この章ではチームプロビジョニングプロファイルの使い方を説明しました。Xcodeが生成、管理するもので、デバイス上でアプリケーションを起動するために必要です。さらに、チームプロビジョニングプロファイルを生成するために必要な情報、デバイスが登録済みであるかどうか確認する方法、エラーや警告が現れた場合の対処方法についても説明しました。

テスト用にアプリケーションをエクスポート (iOS、tvOS、watchOS)

アプリケーションをiTunes Connectにアップロードする前に、これを配布し、あらかじめ登録したデバイス上でテストすることも可能です。この目的には、アドホックプロビジョニングプロファイル、またはチームプロビジョニングプロファイルを使います。この方法で、Xcodeでローカルにビルドした、アプリケーションのさまざまな変種をテストできます。テスト担当者は、チームメンバーである必要も、アプリケーションを実行するiTunes Connectユーザである必要もありませんが、自分のデバイスをメンバーセンターに登録しておく必要があります。自分のチームが開発やテストに用いるデバイスを、製品ファミリーごとに、年間100台まで登録可能です。該当するデバイスのいくつかをテスト用に確保できる、あるいはテスト担当者が用意しているデバイスのIDが分かるのであれば、こういった手段も検討してください。

注意: Apple Developer Enterprise Programの会員は、iTunes Connectを利用できないので、アプリケーションのベータテストを実施する際には、この方法を用います。

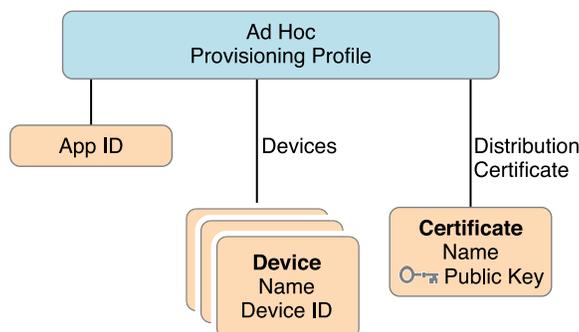
テスト用にアプリケーションをエクスポートする手順を以下に示します。

1. テスト用デバイスをすべて登録する。
2. アプリケーションをアーカイブする。
3. アーカイブをエクスポートする（コード署名には、アドホックプロビジョニングプロファイル、またはチームプロビジョニングプロファイルを使用）。
4. テスト用デバイスにインストールする。
5. テスト担当者に、クラッシュした場合は報告するよう依頼する。

アドホックプロビジョニングプロファイルについて

アドホックプロビジョニングプロファイルは、配布用プロビジョニングプロファイルの一種であり、所定のデバイス上にアプリケーションをインストールし、Xcodeがなくてもアプリケーションサービスを利用できるようにする働きがあります。アプリケーションに対して作成できる2種類の配布用プロビジョニングプロファイルのうちの一つです（もう一つは後でストアに登録する際に使います）。知らない間にテスト版がコピー、配布されないよう保護することができます。

テスト担当者に配布する準備が整い次第、配布するアプリケーション（複数可）に合致するApp ID、テストに用いるデバイスのリスト、配布用証明書（1件）を指定して、アドホックプロビジョニングプロファイルを作成します。



テストに用いるデバイスはそれぞれ、**一意的なデバイスID (UDID)** で識別されます。プロビジョニングプロファイルに登録したデバイスに関する情報は、メンバーセンターに保存されます。開発/テスト用のデバイスは、製品ファミリーごとに、1年当たり100台まで登録できます。デバイスの種類ごとに、1年当たり100台までです。したがってiOSアプリケーションの場合、iPadを100台、iPhoneを100台、iPod Touchを100台まで登録可能です。

テスト用デバイスを登録する

アドホック（またはチーム）プロビジョニングプロファイルを作成する前に、1つまたは複数のテスト用デバイスを登録しておきます。テスト用デバイスを登録するためには、テスト担当者が持っているデバイスのIDを収集し、メンバーセンターに追加する必要があります。

テスト担当者はXcodeをインストールしなくてもデバイスIDを調べることができます。iOSアプリケーションであればiTunesを使って調べます。その手順（[「iTunesでiOSデバイスIDを調べる」](#)（203ページ）を参照）をテスト担当者に伝え、デバイスIDを返送するよう依頼してください。自分の手元にあるデバイスについても、同じ手順でIDを調べます。Apple TVデバイスの場合、[「System Information」でデバイスIDを調べる](#)（204ページ）に示した手順をテスト担当者に伝えてください。調べる際には、Apple TVをMacに接続する必要があります。

この後、メンバーセンターに1つまたは複数のデバイスを登録します（[「メンバーセンターでデバイスを登録する」](#)（202ページ）を参照）。

アプリケーションをアーカイブする

次にアプリケーションのアーカイブを作成します。Xcodeはこのアーカイブを「Archives」オーガナイザに保存するようになっています。

アーカイブを作成するには

1. Xcodeプロジェクトエディタで、総称デバイス（「Generic iOS Device」、「Generic tvOS Device」、「Generic iOS Device+watchOS Device」のいずれか）、または具体的なデバイス名を、ツールバーの「Scheme」メニューから選択してください。

シミュレータ用に構築したアプリケーションのアーカイブは作成できません。デバイスがMacに接続されている場合は、その名前がツールバーの「Scheme」メニューに現れます。Macから切り離すと、メニュー項目が総称デバイス名に変わります。

2. 「Product」 > 「Archive」コマンドを実行します。

「Archives」オーガナイザが開き、新しいアーカイブが表示されます。

Xcodeはアーカイブに対して予備検証テストを実行します。その結果、プロジェクトエディタに警告が表示されることがあります。たとえば、アセットカタログを使っておらず、必要なアプリケーションアイコンもない場合（[“アプリケーションアイコンと起動画像ファイルを個別に設定する”](#)（50ページ）を参照）、「Info.plist」に関する警告メッセージが表示されます。この場合は、問題を解消した後、アーカイブをもう一度作成してください。

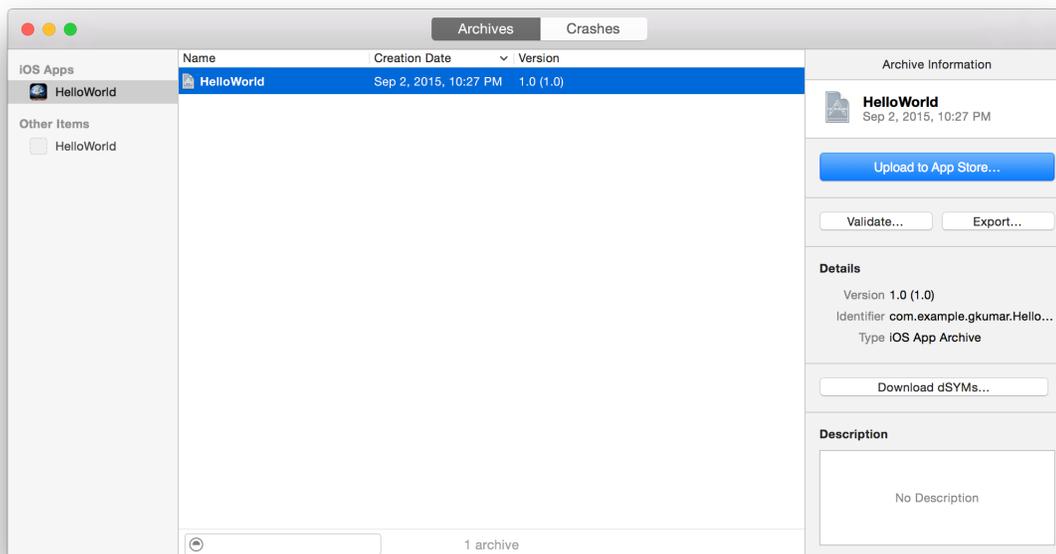
ストアを通さず、テスト用にアプリケーションをエクスポートする

テスト担当者はXcode上でアプリケーションを実行することができないので、**iOS App**ファイル（拡張子は「.ipa」）を作成して、（Xcodeなしで）デバイス上にアプリケーションをインストールできるようにしなければなりません。この方法で、ユニバーサルアプリケーション（サポート対象のあらゆるデバイス上で動作）も、デバイスごとの変種（実際にストアが各ユーザーに配布することになるもの）もテスト可能です。

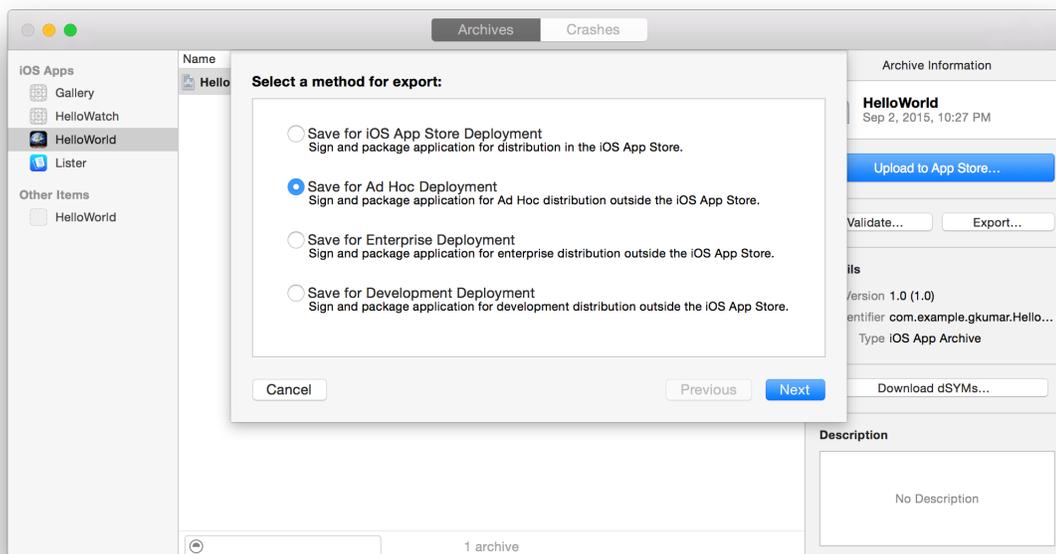
注意: tvOSバイナリは、iOS Appファイル（拡張子は「.ipa」）でもあります。

テスト用のiOS Appファイルを作成するには

1. 「Archives」オーガナイザで、該当するアーカイブを選択します。

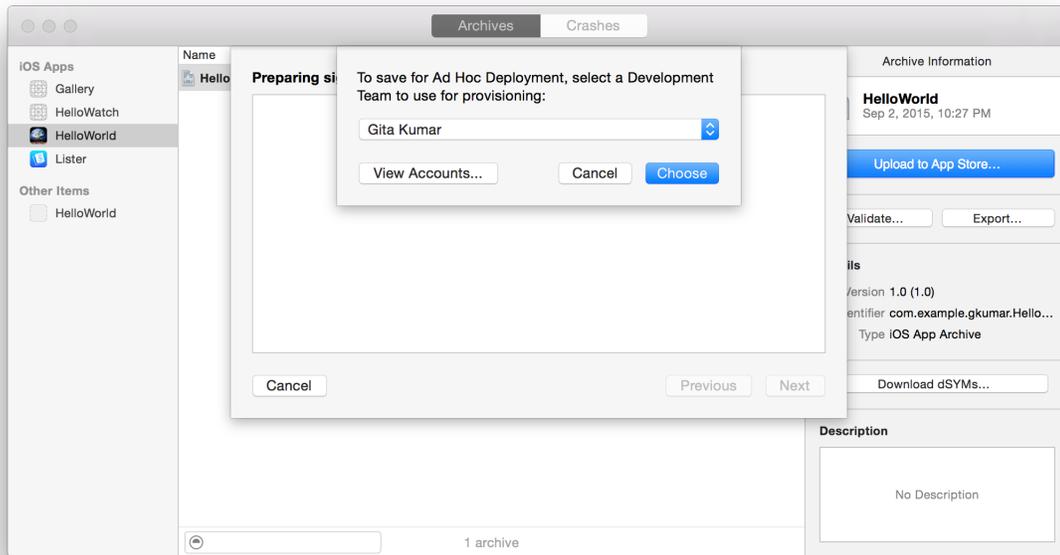


2. 「Export」ボタンを押し、エクスポートオプションを選択した上で、「Next」を押してください。
所定のデバイスを使うユーザに配布する場合は、「Save for Ad Hoc Deployment」を選択します。
アプリケーションは配布用証明書でコード署名されます。
内部テスト用に配布する場合は、「Save for Development Deployment」を選択します。アプリケーションは開発用証明書でコード署名されます。



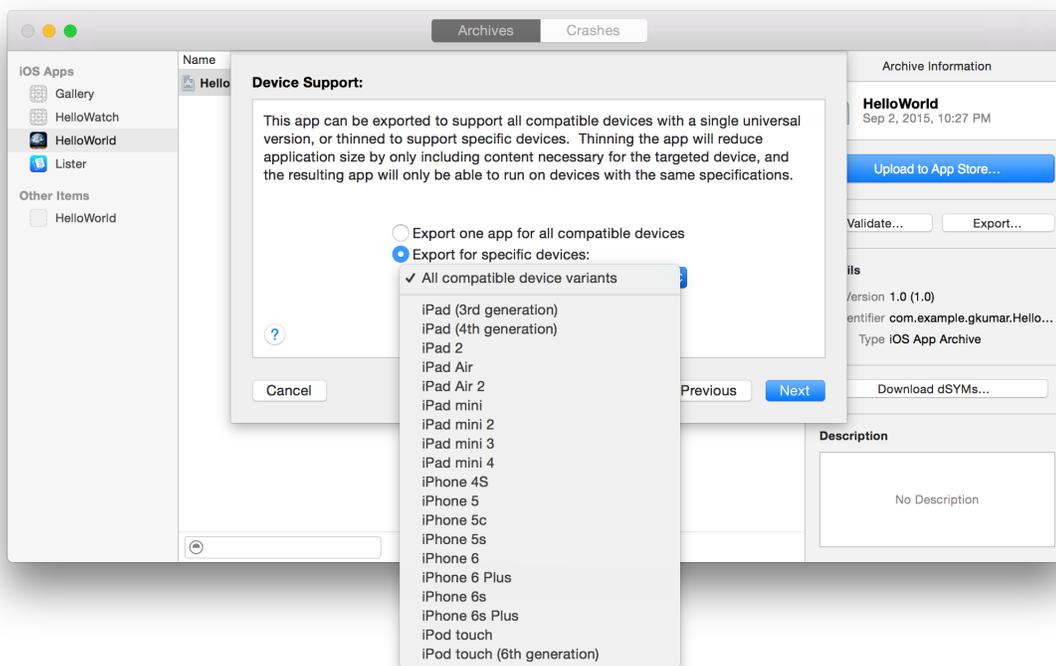
3. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。

Xcodeは必要に応じ、署名IDやプロビジョニングプロファイルを生成します。



4. 「DeviceSupport」ダイアログで、エクスポート対象（ユニバーサルアプリケーション、または特定デバイス用の変種）を選択し、「Next」を押してください。
 - サポート対象のあらゆるデバイス上で動作するようにしたければ、「Export one app for all compatible devices」を選択します。
 - あらゆるデバイスの変種でテストしたければ、「Export for specific devices」を選択し、次いでポップアップメニューから「All compatible device variants」を選択してください。

- 特定デバイス向けの変種をテストしたい場合は「Export a thinned app for a specific device」を選択し、次に該当するデバイスファミリーをポップアップメニューから選択します。



5. 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認します。

アドホックプロファイルプロファイルは、「XC Ad Hoc:」で始まるテキストです。同様にチームプロビジョニングプロファイルは、先頭が「iOS Team Provisioning Profile:」となっているはずで

6. ビルドオプションを確認し、「Next」を押してください。
 - オンデマンド型リソースを利用する場合は「Include manifest for over-the-air installation」をオンにします。

ストアを通さずにアプリケーションを配布する場合、要求に応じて当該リソースを配布できるように、自分で管理しなければなりません。マニフェストファイルはXML形式のplistで、デバイスがウェブサーバからアプリケーションを検索、ダウンロードし、インストールするために使います。

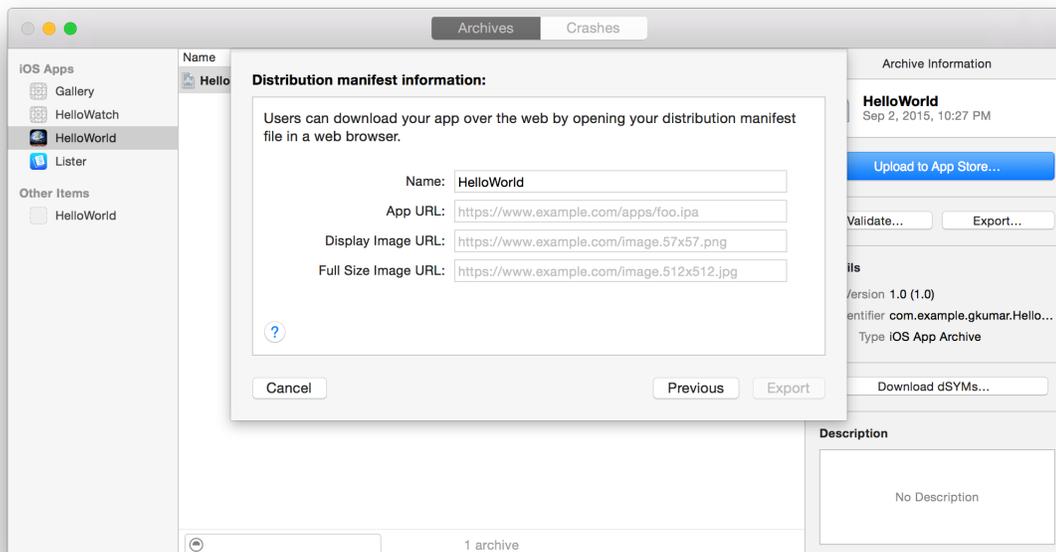
- ビットコードから生成したビルドをテストしたい場合は、「Export from bitcode」をオンにしてください。

ビットコードとは、コンパイル済みアプリケーションの中間表現のことです。ストアはこれをもとにバイナリをビルドします（「[ビットコード](#)」（21 ページ）を参照）。ビットコードから生成したバイナリはXcode上でテストできますが、TestFlightを使って、ストアが生成したバイナリをテストすることも必要です（「[TestFlightによるアプリケーションの配布 \(iOS、tvOS、watchOS\)](#)」（130 ページ）を参照）。

7. マニフェストファイルを要求する場合、「Distribution manifest information」ダイアログが現れるので、ウェブサーバの詳細を入力してください。

下記の情報を入力します。

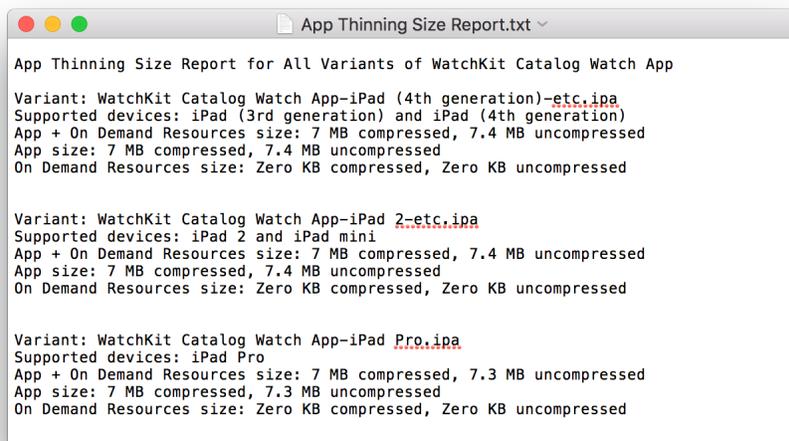
- **Name**。ダウンロードとインストール中に表示されるアプリケーション名。
- **App URL**。iOS Appファイルの完全修飾URL (HTTPSスキーム)。
- **Display Image URL**。ダウンロード/インストール中に現れるアプリケーションアイコンの完全修飾URL (HTTPSスキーム)。画像ファイルは57×57ピクセルのPNG形式でなければなりません。
- **Full Size Image URL**。iTunesに表示される大きな画像の完全修飾URL (HTTPSスキーム)。画像ファイルは512×512ピクセルのPNG形式でなければなりません。



8. 「Export」を押してください。

Finder上にエクスポートしたファイルが現れます。

Xcodeは指定されたiOS Appファイルを収容するフォルダを生成します。マニフェストファイルを要求した場合、「manifest.plist」というファイルが、iOS Appファイルと同じ場所にできています。変種を要求した場合、フォルダには、間引きによって容量を削減できた度合いも収容されます。「App Thinning Size Report」ファイルには各変種のサイズの要約、「app-thinning.plist」ファイルにはその詳細が記載されています。



```
App Thinning Size Report for All Variants of WatchKit Catalog Watch App

Variant: WatchKit Catalog Watch App-iPad (4th generation)-etc.ipa
Supported devices: iPad (3rd generation) and iPad (4th generation)
App + On Demand Resources size: 7 MB compressed, 7.4 MB uncompressed
App size: 7 MB compressed, 7.4 MB uncompressed
On Demand Resources size: Zero KB compressed, Zero KB uncompressed

Variant: WatchKit Catalog Watch App-iPad 2-etc.ipa
Supported devices: iPad 2 and iPad mini
App + On Demand Resources size: 7 MB compressed, 7.4 MB uncompressed
App size: 7 MB compressed, 7.4 MB uncompressed
On Demand Resources size: Zero KB compressed, Zero KB uncompressed

Variant: WatchKit Catalog Watch App-iPad Pro.ipa
Supported devices: iPad Pro
App + On Demand Resources size: 7 MB compressed, 7.3 MB uncompressed
App size: 7 MB compressed, 7.3 MB uncompressed
On Demand Resources size: Zero KB compressed, Zero KB uncompressed
```

Xcodeを使ってテスト用デバイスにアプリケーションをインストールする

Xcodeを使ってiOS Appファイルをデバイスにインストールできます。tvOSアプリケーションの場合、この方法でしか、iOS AppファイルをApple TVにインストールすることはできません

アプリケーションをデバイスにインストールするには

1. デバイスをMacに接続します。
2. Xcode上で、「Window」 > 「Devices」の順に選択した後、「Devices」から該当するデバイスを選択します。
3. 「Installed Apps」テーブルの下にある「Add」ボタン（「+」の表示）を押してください。
4. するとダイアログが現れるので、iOS Appファイルを選択し、「Open」を押します。

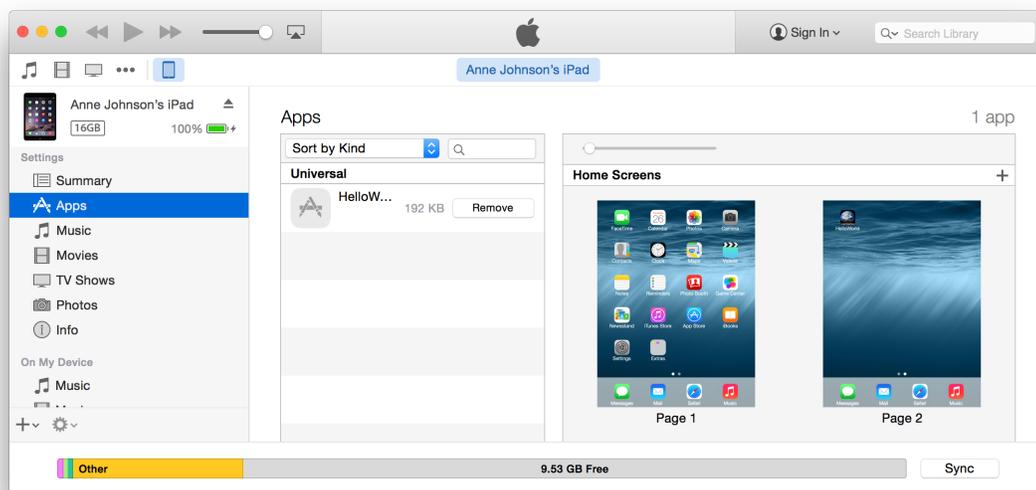
「Devices」ウインドウではほかにもさまざまな作業が可能です（[“デバイス上のアプリケーションを管理する”](#)（209 ページ）を参照）。

iTunesを使ってテスト用デバイスにアプリケーションをインストールする (iOS、watchOS)

テスト担当者にアプリケーションを配布する前に、アプリケーションをデバイスにインストールし、実行する手順を手元で一度試してみてください。開発用ではないデバイスに、iTunesを使ってインストールします。オペレーティングシステムは、自動的にプロビジョニングプロファイルをアプリケーションから取り出し、デバイスに組み込むようになっています。その上で、実際にデバイス上でテストしてみましょう。

アプリケーションをiOSデバイスにインストールするには

1. テストに用いるデバイスを、iTunesが稼働しているMacに接続します。
できるだけ、開発用のMac以外を使ってください。watchOSアプリケーションについては、Apple WatchとペアリングしたiPhoneを接続します。
2. 生成済みのiOS Appファイルをダブルクリックしてください。
3. iTunesで、ウィンドウの左上隅にあるデバイスを押します。
4. 「Apps」 ボタンをクリックします。
iTunesのアプリケーションリストに表示されます。



5. 「Apps」 以下のポップアップメニューから、「Sort by Name」または「Sort by Kind」を選択してください。
「Install」 ボタンまたは「Remove」 ボタンが、アプリケーション名の隣に現れます。
6. 「Install」 ボタンが現れた場合はこれを押してください。
ボタンの表示が「Will Install」に変わります。
7. 「Apply」 ボタンまたは右下隅にある「Sync」 ボタンをクリックして、デバイスを同期します。

アプリケーションがデバイスにアップロードされ、テストを開始できる状態になります。

最後に、iOS Appファイルをテスト担当者に送信します。アプリケーションのインストール手順やクラッシュレポートの提出手順を説明した文書も添えてください（“テスト担当者にクラッシュレポートを送るよう依頼する”を参照）。

Xcodeサーバを利用してアプリケーションを配布する

この章で説明する手順でベータテスト用にアプリケーションをエクスポートした後、必要に応じて、アプリケーションを配布するようにXcodeサービスを設定することも可能です。Xcodeサーバは、アプリケーションのビルド、解析、テスト、アーカイブ作成という、一連の処理を自動化します。ウェブサイトを利用して、実稼働版およびそのアーカイブを、テスト担当者や他のチームメンバーに配布することも可能です。Xcodeサーバを利用して、各版のアプリケーションをテスト担当者に配布する手順については、『*Xcode Continuous Integration Guide*』を参照してください。

アプリケーションのサンドボックスデータをコピーする

アプリケーションの問題点をさらに詳しく調べるために、デバイス上のアプリケーションコンテナの内容を表示、ダウンロード、置換できます（“デバイスでアプリケーションコンテナを表示、ダウンロード、置換する”（210 ページ）を参照）。

テスト担当者にクラッシュレポートを送るよう依頼する

デバイス上で実行したアプリケーションがクラッシュすると、オペレーティングシステムはそのイベントのレポートを出力します。その後、デバイスを再びiTunesに接続すると、クラッシュログとも呼ばれるこのレコードがMacにダウンロードされます。テスト担当者は、バグレポートに添えて、このクラッシュログも開発者に送らなければなりません。さらに、アプリケーション公開後に届いたクラッシュレポート（実際の顧客環境で発生したもの）も、iTunes Connectから取得できます。

注意: TestFlightを使ってプレリリース版アプリケーションを配布した場合（“[TestFlightによるアプリケーションの配布 \(iOS、tvOS、watchOS\)](#)”（130 ページ）を参照）、Appleはクラッシュログを収集、集約し、クラッシュレポートとして提供します。

テスト担当者に、クラッシュレポートをデバイスから取得し、開発者に送信する手順を説明してください。

Macからクラッシュレポートを送信するには

1. テストに用いるデバイスを、iTunesが稼働しているMacに接続します。
クラッシュレポートがMacにダウンロードされます。
2. ファインダ上で、「移動」>「フォルダへ移動」を実行してください。
3. ~/Library/Logs/CrashReporter/MobileDeviceと入力します。
4. デバイス名で決まるフォルダを開きます。
5. テスト中のアプリケーションから始まる名前を持つクラッシュログを選択します。
6. 「Finder」>「サービス」>「Mail」>「ファイルを送信」を選びます。
7. 「新規メッセージ」ウィンドウで、「宛先」フィールドに開発者の電子メールアドレス、「件名」フィールドに適切なテキストを入力します。
8. 「メッセージ」>「送信」を選びます。
9. 重複送信を避けるため、送信済みのクラッシュレポートは削除してください。

Windowsからクラッシュレポートを送信するには

1. Windowsの検索フィールドに、使っているオペレーティングシステムのクラッシュログディレクトリを入力します。ここで、「<user_name>」はWindowsユーザ名に置換されます。
 - Windowsの場合、クラッシュログの場所として次のように入力：

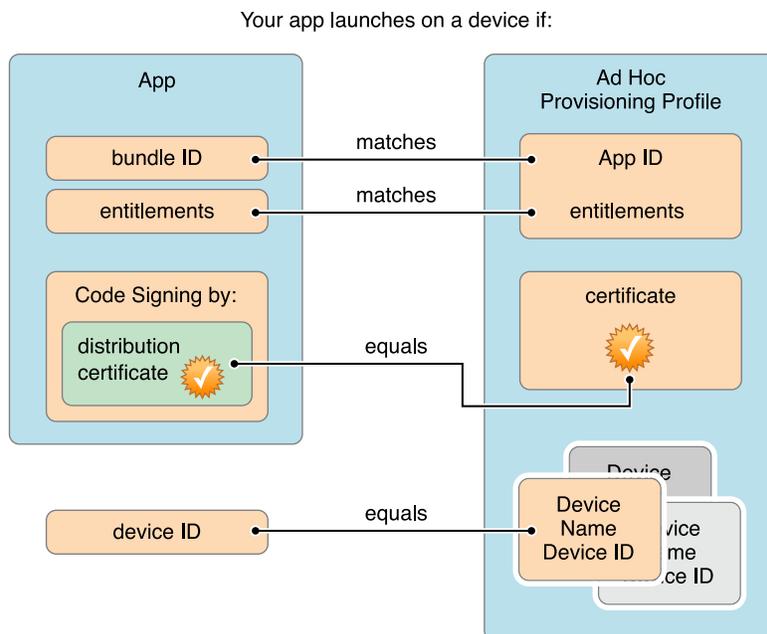
```
C:\Users\<user_name>\AppData\Roaming\Apple  
computer\Logs\CrashReporter\MobileDevice
```
 - Windows XPの場合、クラッシュログの場所として次のように入力：

```
C:\Documents and Settings\<user_name>\Application Data\Apple  
computer\Logs\CrashReporter
```
2. デバイス名で始まる名前のフォルダを開き、テスト中のアプリケーションのクラッシュログをアプリケーションの開発者に電子メールで送信します。その際、件名を「<app_name> crash logs from <your_name>」という形式にしてください（たとえば、「MyTestApp crash logs from Anna Haro」）。

テスト担当者から届いたレポートの読み方については、「[クラッシュデータを「Devices」ウィンドウに表示し、インポートする](#)」（145 ページ）を参照してください。

アドホックプロビジョニングプロファイルの詳細

iOS Appファイルに、アドホックプロビジョニングプロファイルで指定されている配布用証明書を使って署名しました。アドホックプロビジョニングプロファイルは、アプリケーションをビルドし、アーカイブ化する際、アプリケーションバンドルに収容されます。次に、アプリケーションをテスト用デバイスにインストールしました。アプリケーションを正常に起動するためには、バンドルIDがApp IDと合致していること、署名が配布用証明書と合致していること、デバイスがアドホックプロビジョニングプロファイルに登録されていることが必要です。



まとめ

この章では、アドホックプロビジョニングプロファイルを使ってアプリケーションを所定のテスト用デバイスに配布する方法を習得しました。さらに、ベータ版のインストール手順やクラッシュレポートの送信方法をアドホックテスト担当者に指示することについても理解しました。

アプリケーションをテスト用にエクスポート (Mac)

チームプロビジョニングプロファイルを使って、所定のMacコンピュータ上で外部のXcodeを起動するようにMacアプリケーションを設定できます。チームプロビジョニングプロファイルでは、開発用に登録しているMacコンピュータと同じプールが使われるため、開発用とテスト用のMacコンピュータは合わせて年間100台に限定されています。

チームプロビジョニングプロファイルを使ってベータテスト担当者にアプリケーションを配布する手順は以下の通りです。

1. テスト用のMacコンピュータをすべて登録する。
2. アプリケーションをアーカイブする。
3. チームプロビジョニングプロファイルを使ってアプリケーションをエクスポートする。
4. テスト用Macコンピュータにアプリケーションをインストールする。

最終候補をアップロードし、ストアに登録する準備が整ったら、インストーラパッケージの動作をテストしてください（[“Macインストーラパッケージをテストする”](#)（118 ページ）を参照）。

テスト用Macコンピュータを登録する

Xcodeを使ってアプリケーションをエクスポートする前に、テスト用に使用するMacコンピュータをすべてメンバーセンターに登録しておきます。これにより、Xcodeはチームプロビジョニングプロファイルを生成する前に、開発用Macを自動的に登録します。テスト用Macコンピュータを登録するには、テスト担当者が持っているデバイスのIDを収集し、メンバーセンターに追加する必要があります。

チームプロビジョニングプロファイルに複数のテスト用Macコンピュータを追加する方法は、[“Registering Devices Using Member Center”](#)（202 ページ）を参照してください。

アプリケーションをアーカイブする

次にアプリケーションのアーカイブを作成します。Xcodeはこのアーカイブを「Archives」オーガナイザに保存するようになっています。

アーカイブを作成するには

1. 「Scheme」 ツールバーメニューからターゲットを選択して「Run」をクリックします。
必ずアプリケーションのテストを行ってから、アプリケーションをエクスポートしてください。
2. 「Product」 > 「Archive」 コマンドを実行します。
「Archives」 オーガナイザが開き、新しいアーカイブが表示されます。

チームプロビジョニングプロファイルを使ってアプリケーションをエクスポートする

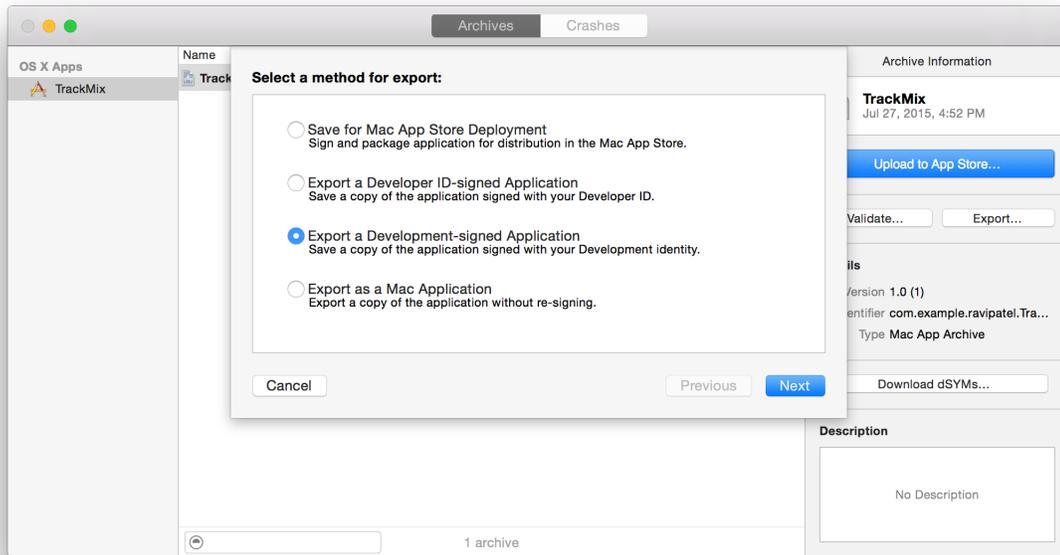
XcodeからMacアプリケーションをエクスポートすると、チームプロビジョニングプロファイルがバンドルに埋め込まれ、アプリケーションが開発用証明書でコード署名されます。これで、エクスポートしたアプリケーションをすべての登録済みMacコンピュータ上で起動できるようになります。

Important: 独自のプロビジョニングプロファイルを使うようにXcodeプロジェクトを設定している場合は、この手順を実行する前に、該当するチームプロビジョニングプロファイルを使うようにプロジェクトを設定しておく必要があります（“[Xcodeで管理されるプロビジョニングプロファイルを移行する](#)”（219ページ）を参照）。このように設定しておかないと、テスト用Macコンピュータ上でアプリケーションが起動しない場合があります。

チームプロビジョニングプロファイルを使って開発用証明書で署名済みのアプリケーションをエクスポートするには

1. 「Archives」 オーガナイザで、該当するアーカイブを選択して「Export」をクリックします。

2. 「Export a Development-signed Application」をオンにして「Next」をクリックします。

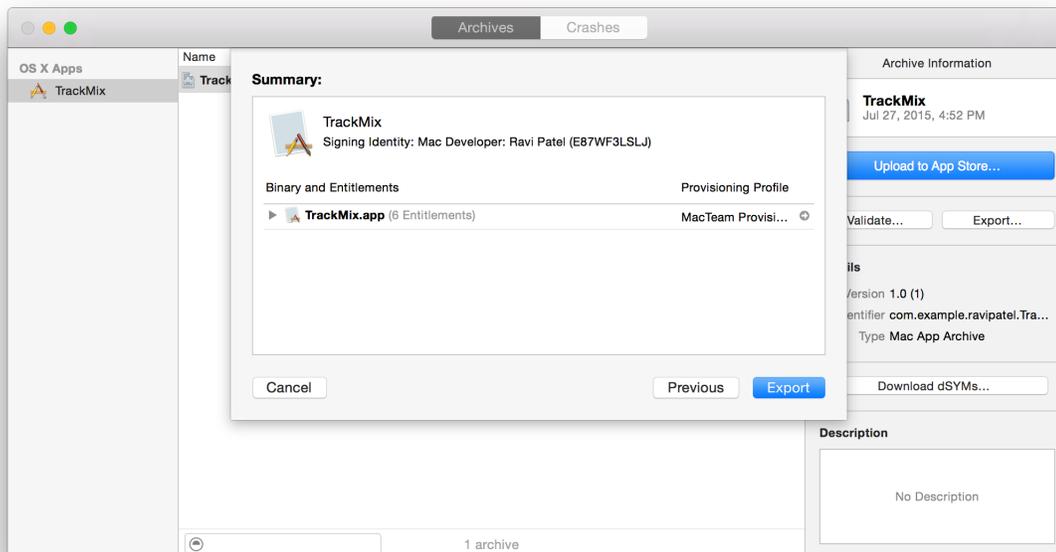


3. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。

Xcodeは必要に応じ、署名IDとプロビジョニングプロファイルを生成します。

4. 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認して「Export」をクリックします。

同様にチームプロビジョニングプロファイルは、先頭が「Mac Team Provisioning Profile:」となっているはずですが。



Finder上にエクスポートしたファイルが現れます。

アプリケーションをテスト用Macコンピュータにインストールする

テスト担当者にアプリケーションを配布する前に、開発用として使用しない別の登録済みMac上でアプリケーションを起動します。これで、このアプリケーションはチームプロビジョニングプロファイルに指定されたMacコンピュータ上でしか起動しなくなります。アプリケーションの開発者の身許が分からないため、そのアプリケーションを所定のMac上で起動できない場合は、OS Xのセキュリティ設定をバイパスして起動するようにします。

開発者の身許が分からないアプリケーションを開くには

1. Finderで、Controlキーを押しながら該当するアプリケーションのアイコンをクリックして、ショートカットメニューから「Open」を選択します。
2. 表示されるGatekeeperダイアログで「Open」をクリックします。

Macインストーラパッケージをテストする

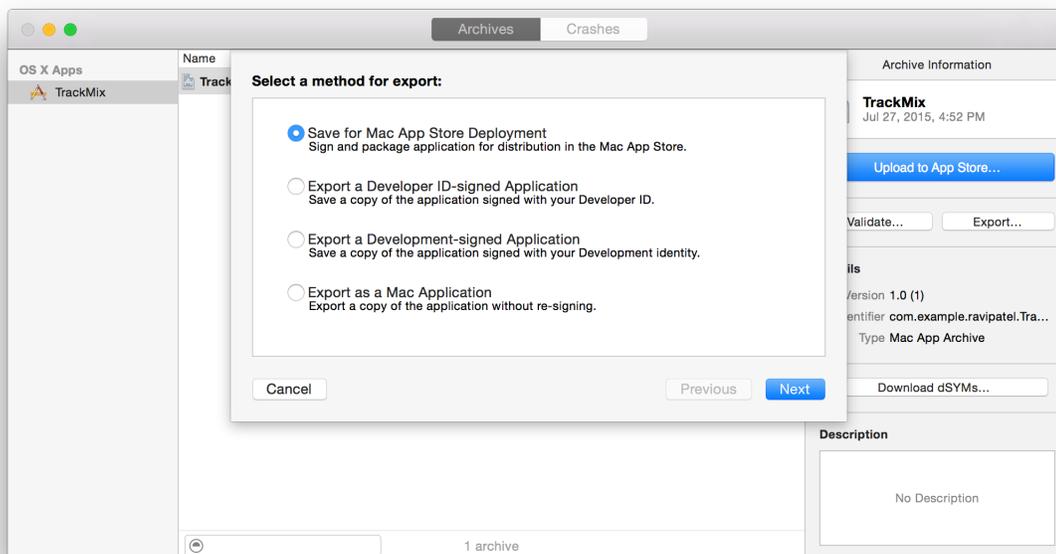
アプリケーションをMac App Storeに登録する前に、適切にインストールできるか確認するため、ひと通り実行してみてください。ディスク上にインストーラパッケージを保存し、「installer」コマンドを実行する、という方法で行います。

インストーラパッケージを作成する

インストーラパッケージをディスクに保存する手順は、Macアプリケーションを検証し、配布する場合と同様です。

インストーラパッケージを作成するには

1. 「Archives」オーガナイザで、該当するアーカイブを選択します。
2. 「エクスポート」ボタンをクリックします。
3. 「Save for Mac App Store Deployment」をオンにし、「Next」を押してください。



4. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。

必要に応じて、Xcodeで配布用証明書と配布用プロビジョニングプロファイルが作成されます。

5. 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認します。

配布用プロビジョニングプロファイルの名前は「xc:」で始まります。

6. 「Export」を押してください。

Finder上にエクスポートしたファイルが現れます。インストーラファイルは拡張子が「.pkg」となっています。

インストーラパッケージをテストする

「Installer」アプリケーションでパッケージを開く、という方法でインストール処理をテストすることはできません。Mac App Storeから購入したとき、適切にインストールできるかどうか確認するためには、「installer」コマンドを使うしかありません。

実際にインストーラパッケージをテストしてみましょう。「Terminal」のウィンドウに、次のコマンドを入力してください。

```
sudo installer -store -pkg path-to-package -target /
```

「installer」コマンドの出力は次のようになります。

```
rpatel$ sudo installer -store -pkg ../Documents/TrackMix.pkg -target /

WARNING: Improper use of the sudo command could lead to data loss
or the deletion of important system files. Please double-check your
typing when using sudo. Type "man sudo" for more information.

To proceed, enter your password, or type Ctrl-C to abort.

Password:
installer: Note: running installer as an admin user (instead of root) gives better
Mac App Store fidelity
installer: TrackMix.pkg has valid signature for submission: 3rd Party Mac Developer
Installer: Ravi Patel (7U3X8B3P5Z)
installer: Installation Check: Passed
installer: Volume Check: Passed
installer: Bundle com.example.rpatel.TrackMix will be installed to
/Applications/TrackMix.app
installer: Starting install
installer: Install 0.0% complete
installer: Install 90.8% complete
installer: Install 100.0% complete
installer: Finished install
```

```
rpatel$
```

同じバンドルIDのアプリケーションがインストール済みであった場合は、これをアップグレードすることになります。既存のアプリケーションを別の場所に移動しておいても、これがアップグレードされてしまいます。（たとえば製品ビルド用のディレクトリ以下に）インストール済みであった場合、あらかじめこれを削除して、`/Applications`以下にインストールされるようにするとよいかも知れません。あるいは、既存の版を残しておくため、ZIPファイルの形でアーカイブ保存する、他のボリュームに移動して当該ボリュームをマウント解除する、という方法も考えられます。

アプリケーションをiTunes Connectにアップロード

アプリケーションの版やビルドをiTunes Connectにアップロードして、テスト用に配布したり、ストアに登録したりすることができます。アップロードに先立ち、iTunes Connect検証テストを実行することも可能です。これ以降、iTunes Connectを使って、アップロードした各版やビルドを管理することになります（『*iTunes Connect Developer Guide*』の「[Viewing Builds](#)」を参照）。

iTunes Connectは、必要な情報をアプリケーションバンドルから抽出し、アップロードされたアプリケーションの版と対応づけるようになっているので、事前に以下の作業を実行してください。

1. iTunes Connectのアプリケーションレコードを作成する。
2. 版番号とビルド文字列を更新する。
3. アプリケーションのアーカイブを作成する。
4. アーカイブを検証する。
5. アプリケーションをiTunes Connectにアップロードする。

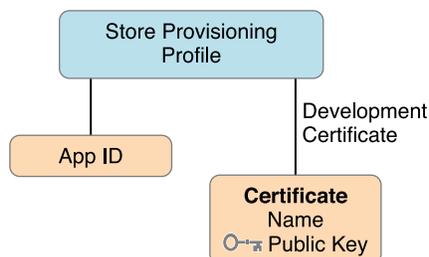
アプリケーションをiTunes Connectにアップロードする際、シンボル情報を添えておけば、各ユーザのデバイス上でクラッシュが発生したとき、そのデータをAppleが収集、集約します。したがって開発者は、いつでも「Crashes」オーガナイザでクラッシュレポートを閲覧できることとなります（「[クラッシュレポートの分析](#)」（132 ページ）を参照）。

テスト用に配布した最新のアーカイブをストアに送信すれば、最終候補をアップロードする前に、Appleガイドラインに沿っているか、Xcodeプロジェクトの設定が適切か、確認できます（「[アプリケーションの登録に向けて準備する](#)」（149 ページ）を参照）。

ストアプロビジョニングプロファイルについて

ストアプロビジョニングプロファイルは配布用プロビジョニングプロファイルの一種であり、特定のアプリケーションサービスを利用する権限をアプリケーションに付与し、アップロードしているのが開発者本人であることを保証する働きがあります。1つまたは複数のアプリケーションに対応する単一のApp IDと、配布用証明書が収容されています。特定のアプリケーションサービスを利用するために、Xcodeで間接的にApp IDを設定できます。アプリケーションサービスを有効にして設定するためには、エンタイトルメントその他の設定が必要です。App ID（チームが開発するアプリケーションに

共通) に対して有効にするエンタイトルメントとXcodeプロジェクトで設定するエンタイトルメントがあります。アプリケーションをストアにアップロードすると、Xcodeにより、ストアプロビジョニングプロファイルから参照される配布用証明書でアプリケーションバンドルへの署名が行われます。



iTunes Connectでアプリケーションレコードを作成する

アプリケーションを検証してiTunes Connectにアップロードするためには、iTunes Connectのアプリケーションレコードが必要です。アプリケーションレコードを作成する場合は、“[アプリケーションレコードを作成する](#)” (155 ページ) に目を通してから、次に進んでください。iTunes Connectで入力するバンドルIDは、Xcodeプロジェクトのそれと合致していなければなりません。初回ビルドをアップロードすると、iTunes Connect側に設定したバンドルIDは変更できなくなります。XcodeプロジェクトでバンドルIDを変更する手順については、“[バンドルIDを設定する](#)” (35 ページ) を参照してください。

版番号やビルド文字列を更新する

版番号やビルド文字列を更新する (“[版番号とビルド文字列を設定する](#)” (41 ページ) を参照) のは、iTunes Connectがこれを手がかりに、アプリケーションの版やビルドを特定するからです。

Important: iTunes Connectで新規にアプリケーションバージョンを作成したときは、版番号を更新してください。また、アプリケーションの新規ビルドをiTunes Connectにアップロードする際には、ビルド文字列を更新しなければなりません。ビルド文字列が表す番号を増やすのです。iOSアプリケーションの場合、iTunesはビルド文字列が変わったことを認識し、新しいアプリケーションを、適切にデバイスに組み込みます。版番号やビルド文字列は、TestFlightやストアを通して配布するため、あるいはクラッシュレポートや.dSYMファイルを識別するためにも使います。

アプリケーションをアーカイブする

アプリケーションを配布する際は、その方法にかかわらず、アーカイブを作成する必要があります。Xcodeのアーカイブ機能を利用すれば、アプリケーションをビルドし、重要なデバッグ情報とともに、Xcodeが管理するバンドルに格納することができます。デバッグ情報をアーカイブに入れておけば、たとえばテスト担当者に配布し、クラッシュレポートが送られてきたとき、その分析が容易になります。

Important: 配布するアーカイブおよびビルド結果は、すべての版を保存しておいてください。アプリケーションをiTunes Connectにアップロードする際、アーカイブおよび対応するプロジェクトが必要になります。Crashesログ内のスタックフレームから、デバッグナビゲータに開いた、ソースコード上の該当箇所を表示するために使います（[「クラッシュレポートの分析」](#)（132ページ）を参照）。デバッグ情報はさらに、ほかから受け取ったクラッシュログのシンボル変換をおこなうためにも必要です。

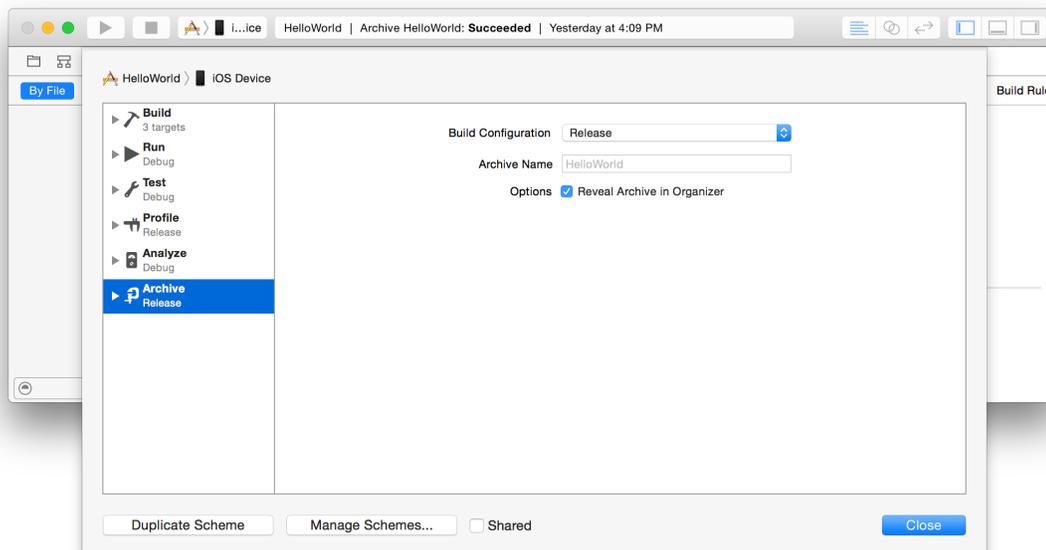
「Archive」スキーム設定を確認する

初めてアーカイブを作成する際には、誤ってデバッグ版をアーカイブ化してしまわないよう、「Archive」スキーム設定を確認してください。

「Archive」スキーム設定を確認するには

1. Xcodeプロジェクトエディタの「Product」 > 「Scheme」 > 「Edit Scheme」コマンドで、スキームエディタを開きます。

2. 左側の列から「Archive」を選択してください。



3. 「Build Configuration」ポップアップメニューから「Release」を選択して「Close」をクリックします。

アーカイブを作成する

次にアプリケーションのアーカイブを作成します。Xcodeはこのアーカイブを「Archives」オーガナイザに保存するようになっています。

アーカイブを作成するには

1. iOS/tvOS/watchOSアプリケーションの場合、総称デバイスまたは具体的なデバイス名を、プロジェクトエディタのツールバーにある「Scheme」メニューから選択してください。

シミュレータ用に構築したアプリケーションのアーカイブは作成できません。デバイスがMacに接続されていれば、その名前がツールバーの「Scheme」メニューに現れます。Macから切り離すと、メニュー項目が総称デバイス名に変わります。

2. 「Product」 > 「Archive」コマンドを実行します。

「Archives」オーガナイザが開き、新しいアーカイブが表示されます。

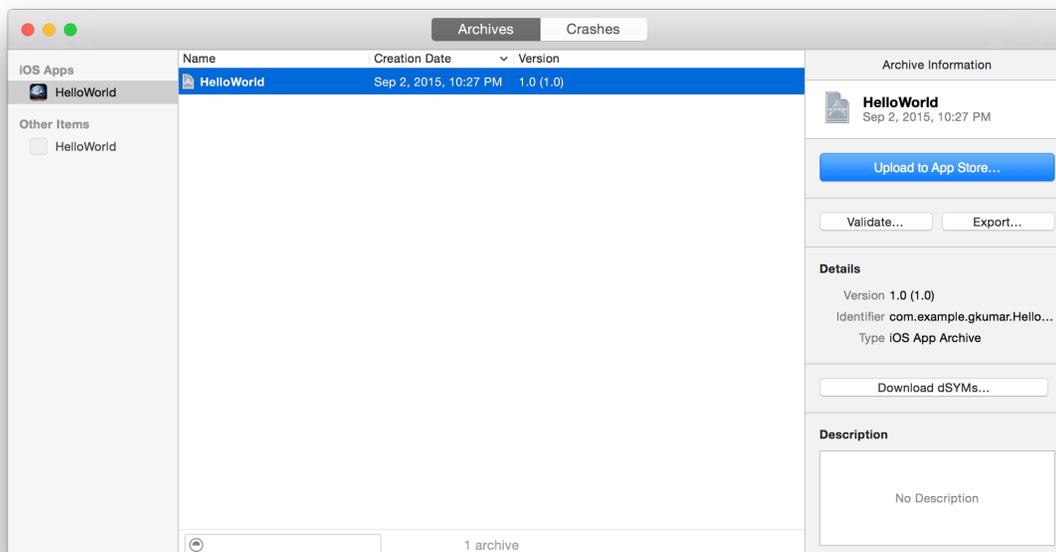
Xcodeはアーカイブに対して予備検証テストを実行します。その結果、プロジェクトエディタに警告が表示されることがあります。この場合は、問題を解消した後、アーカイブをもう一度作成してください。

iTunes Connectの検証テストを実行する

問題を早いうちに見つけて対処するため、iTunes Connectへのアップロードに先立ち、アーカイブを検証してください。検証テストはiTunes Connectを使って行います。これは、ストアに登録できるよう、Xcodeプロジェクトが適切に設定されていることを確認する手続きです。たとえば、必要なアプリケーションアイコンが未設定であれば、問題として報告されます（“[アプリケーションアイコンと起動画像ファイルを個別に設定する](#)”（50 ページ）を参照）。アップロードの際には、同じ検証テストを改めて実行することになります。もっとも、iTunes Connectはアップロード後、より徹底的な検証テストを実施します。

アーカイブを検証するには

1. 「Archives」オーガナイザで、配布するアーカイブを選択し、「Distribute」を押してください。

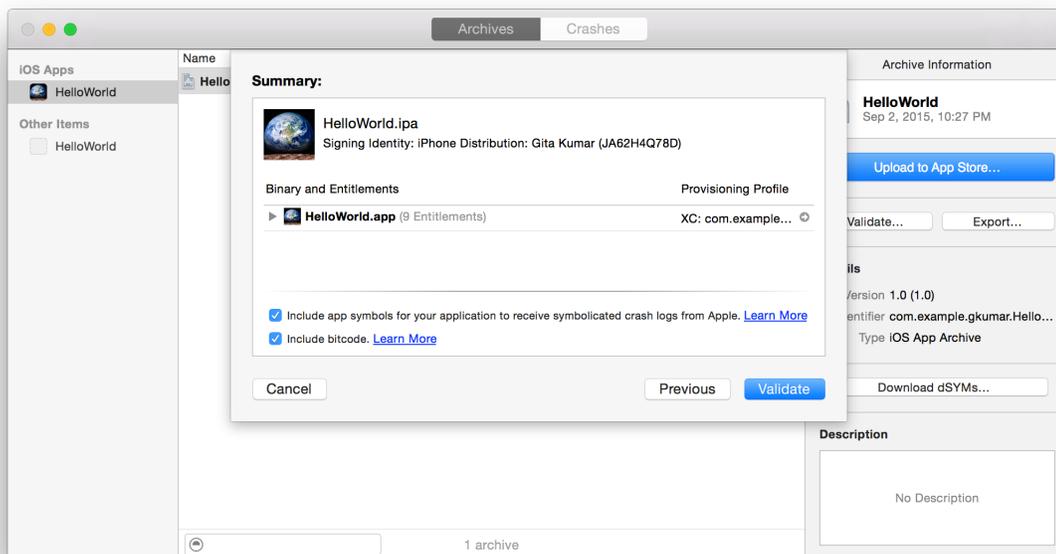


2. Macアプリケーションの場合、検証方法として「Validate for the Mac App Store」を選択して「Next」をクリックします。
3. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。

必要に応じて、Xcodeで配布用証明書と配布用プロビジョニングプロファイルが作成されます。配布用プロビジョニングプロファイルの名前は「XC:」で始まります。

4. 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認して「Validate」をクリックします。

Xcodeにより、アーカイブがiTunes Connectにアップロードされ、iTunes Connectで検証テストが実行されます。アプリケーションレコードが見つからない旨のダイアログが表示された場合は、「Done」をクリックし、iTunes Connect上でアプリケーションレコードを作成してから次に進んでください。



5. 見つかった問題点を確認した上で「Done」をクリックします。

問題点をすべて修正した上で改めてアーカイブを作成し、検証してください。ただし、エラーが残っていても、アップロードし、TestFlightで配布することは可能です。

アプリケーションをアップロードする

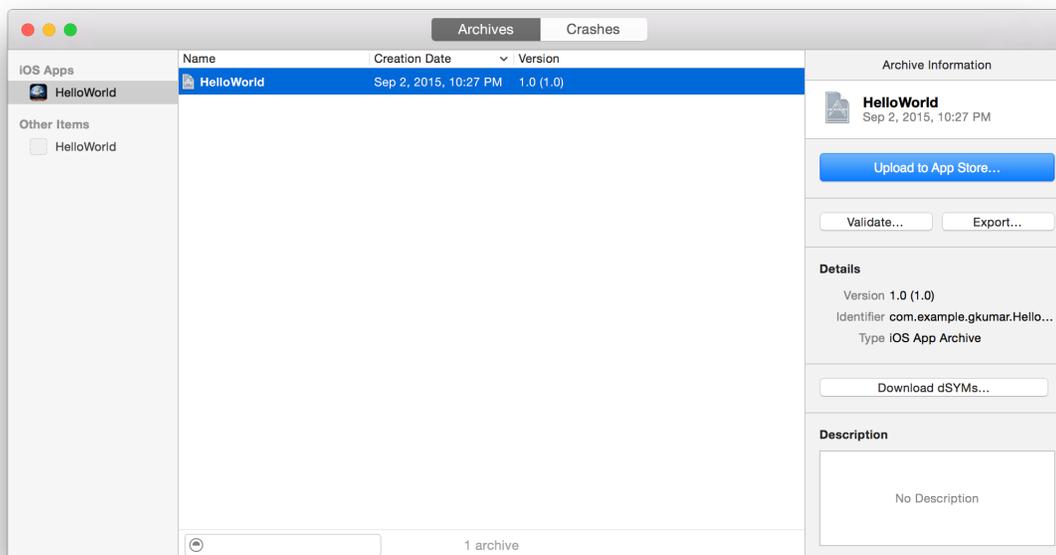
コード署名およびプロビジョニングを施したアプリケーションを、iTunes Connectにアップロードします。確かに開発者本人がアーカイブを作成したこと、特定のアプリケーションサービスを利用する権限を本人がアプリケーションに付与したことを保証するために必要となる重要な手続きです。アプリケーションやインストーラパッケージにコード署名すれば、何者かが改竄してアップロードすることを防止できます。iTunes Connectにアップロードするためには、配布用証明書に対応する秘密鍵が必要であるからです。

テスト用に配布したビルドと同じものをアップロードし、後からストアに登録することができます。重要なのは、ストアに登録するものと、完全に同じビルドをテストすることです。ビットコードを収容している場合（[“ビットコード”](#)（21 ページ）を参照）、ストアがビルドする変種をTestFlightで取得し、テストしてください（[“TestFlightによるアプリケーションの配布（iOS、tvOS、”](#)

[watchOS](#) ” (130 ページ) を参照)。Macアプリケーションの場合、iTunes Connectにアップロードするのと同じアーカイブから、テスト用アプリケーションをエクスポートします (“[アプリケーションをテスト用にエクスポート \(Mac\)](#) ” (114 ページ) を参照)。

アプリケーションをiTunes Connectにアップロードするには

1. 「Archives」 オーガナイザでアップロードするアーカイブを選択し、「Upload to App Store」を押してください。



2. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。

必要に応じて、Xcodeで配布用証明書と配布用プロビジョニングプロファイルが作成されます。配布用プロビジョニングプロファイルの名前は「xc:」で始まります。

3. 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認します。

4. Appleからクラッシュレポートを受け取れるようにしたければ、「Include app symbols for your application...」をオンにしてください。

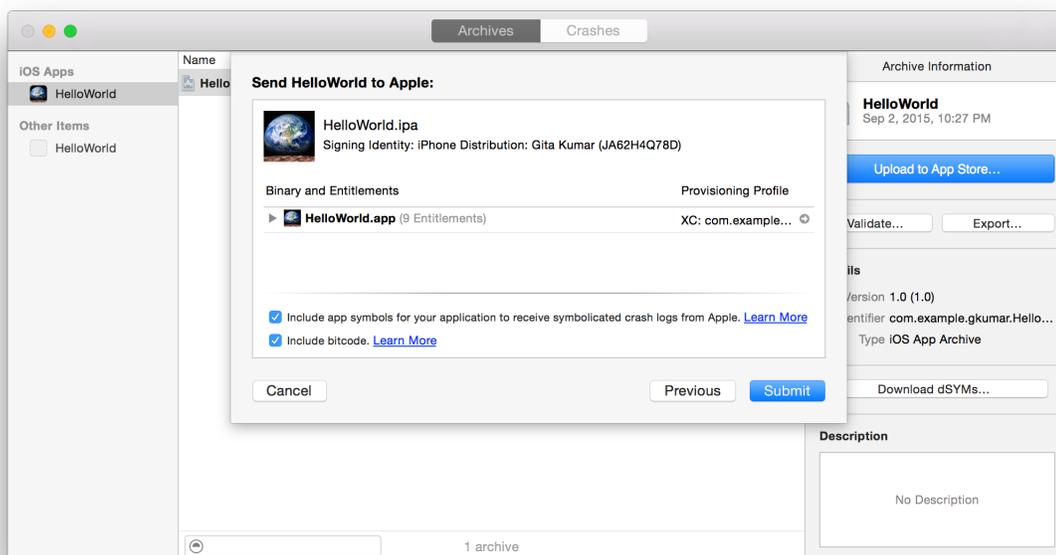
こうしておけば、シンボル変換済みのクラッシュログを、「Crashes」オーガナイザで閲覧できるようになります (“[クラッシュレポートの分析](#)” (132 ページ) を参照)。

5. ビットコードを収容する場合は、「Include bitcode...」がオンになっていることを確認してください (デフォルト設定のままであればビットコードは有効)。

iTunes Connectにアップロードしたアプリケーションに、ビットコードが収容されていれば、ストアはこれをコンパイル、リンクして、バイナリを生成できます。ビットコードがあれば、ストアは今後いつでも更新を適用できます。開発者が新しい版を用意し、[App Review](#)に登録する必要はありません。

- iOSアプリケーションの場合、ビットコードは必須ではありません。
- watchOSアプリケーションであれば、ビットコードは必須です。このチェックボックスをオフにしても、ビットコードが削除されるのはiOSアプリケーションの方だけです。
- tvOSアプリケーションの場合、ビットコードは必須なので、このチェックボックスは現れません。
- Macアプリケーションはビットコードに未対応であり、このチェックボックスは現れません。

ビットコードを収容する場合、同じアプリケーションバンドルに属するアプリケーションやフレームワークすべてを対象とする必要があります。



Important: アプリケーション配布後、iTunes Connectを使ってdSYMsファイルをダウンロードする場合は、アーカイブを選択してから「Download dSYMs」を押します。詳しくは「クラッシュデータを「Devices」ウィンドウに表示し、インポートする」（145 ページ）を参照してください。

6. 「Submit」を押してください。

Xcodeにより、アーカイブがiTunes Connectにアップロードされ、iTunes Connectで検証テストが実行されます。アプリケーションレコードが見つからない旨のダイアログが表示された場合は、「Done」をクリックし、iTunes Connect上でアプリケーションレコードを作成してから次に進んでください。

7. 問題が見つかった場合は、「Done」をクリックし、その問題を解消してから次に進んでください。

配布前に解消しておくべき問題すべての一覧が『*App Store Review Guidelines for iOS Apps*』に記載しています。アプリケーション審査手続きについては、「[App Review](#)」で確認してください。

8. 問題がなければ、「Submit」をクリックして、アプリケーションをアップロードします。

XcodeはアーカイブをAppleに送ります。Appleでは、ガイドラインに則っているかどうかを審査します。ビルドが却下された場合は、指摘された問題点を修正し、新しいビルドをアップロードします。正常にアップロードできたら、版番号とビルド番号をiTunes Connectで確認してください（“[Viewing Binary Details](#)”を参照）。iTunes Connectを使って配布オプションを選択します。たとえば、内部や外部のテスト担当者に配布する、という選択肢があります（「[TestFlight Beta Testing \(Optional\)](#)」を参照）。アプリケーションを[App Review](#)に登録する準備が整ったら、“[アプリケーションの登録](#)”（149ページ）を参照してください。

TestFlightによるアプリケーションの配布 (iOS、tvOS、watchOS)

ここまでの章で、Xcode上でアプリケーションを開発し、必要に応じてアプリケーションサービスを組み込みました。さらに、デバイスシミュレータや手元にあるデバイスでアプリケーションをテストしました。いよいよ、多くの人たちにアプリケーションを配布し、「現実の環境」におけるテストを実施します。次の版になる候補をプレビュー版として配布することもあります。

プレリリース版の配布はAppleがおこないます (TestFlightを使用)。iTunes Connectにアップロードした後 (“[アプリケーションをiTunes Connectにアップロード](#)” (121 ページ) を参照)、iTunes Connectを使ってテスト担当者を募ってください。ひとつのアプリケーションにつき、内部25人 (iTunes Connect ユーザ)、外部1000人 (電子メールアドレスのみ指定するユーザ) までのテスト担当者に配布することも可能です。テスト担当者はTestFlight (ストアから入手可能) を使って、デバイスに応じたアプリケーションの変種をダウンロード、インストールします。Appleはクラッシュレポート作成サービスを提供します (“[クラッシュレポートの分析](#)” (132 ページ) を参照)。TestFlightを使って配布したアプリケーションについて、Xcode上で直接、クラッシュレポートを閲覧可能です。

内部からのみ募った場合、Beta App Reviewには登録されません。

Important: アプリケーションのテストは、各種のデバイスを使い、さまざまな版のオペレーティングシステム上で、厳格に実施してください。デバイスの種類やの版によって能力が異なるので、開発用にプロビジョニングしたデバイスやシミュレータ上でアプリケーションをテストするだけでは不十分です。実デバイス上で稼働するスレッドがすべて、シミュレータでも動いているわけではありません。また、Xcodeを使ってデバイス上でアプリケーションを起動すると、ウォッチドッグタイマーの一部が無効になってしまいます。少なくとも、動作すると思われる入手可能なあらゆるデバイス上でテストしなければなりません。さらに、互換性テストのため、旧リリースをインストールしたデバイスも確保しておくといでしょう。特定のデバイスやリリースに対応できない (動作を保証しない) 場合、Xcodeのプロジェクトターゲット設定にその旨を明記してください (“[配備情報を設定する](#)” (43 ページ) を参照)。

アプリケーションをiTunes Connectにアップロードすると (“[アプリケーションをiTunes Connectにアップロード](#)” (121 ページ) を参照)、次のような作業ができるようになります。

1. ビルドの詳細を参照する。
2. プレリリース版のアプリケーションにメタデータを追加する。
3. 内部および外部のテスト担当者を募る。
4. 外部のテスト担当者を募った場合、アプリケーションをBeta App Reviewに登録する。

詳しい手順は『*iTunes Connect Developer Guide*』の「TestFlight Beta Testing (Optional)」を参照してください。

クラッシュレポートの分析

TestFlightを使ってテスト用にアプリケーションを配布し、あるいはストアを通して配布した後は、定期的にクラッシュレポートをダウンロード、分析してください。アプリケーションがクラッシュすると、システムはクラッシュログを生成します。アプリケーションが停止した状況を記録したもので、多くの場合、実行スレッドそれぞれの完全なスタックトレースが入っています。Appleは、クラッシュログを収集、集約し、クラッシュレポートとして開発者に提供します。クラッシュレポートの対象は、アプリケーション、WatchKitエクステンション、および他のアプリケーションのエクステンションです。

デバイスに直接クラッシュログを表示し、あるいはクラッシュレポートをインポートすることも可能です（[「クラッシュデータを「Devices」ウィンドウに表示し、インポートする」](#)（145 ページ）を参照）。たとえばストアを通さずにアプリケーションを配布する場合、クラッシュレポートを直接送るようユーザに依頼し、Xcodeにインポートして内容を調べることになります。

クラッシュレポートサービスについて

Appleは開発者向けに、クラッシュレポートのダウンロードサービスを提供しています。ストアを通して配布している、最新版のアプリケーションが対象です。クラッシュログを収集し、似た内容のものを集約して、クラッシュレポートとしてまとめます。アプリケーションのビルドごとに、複数のクラッシュレポートができます。各クラッシュレポートは、多くのクラッシュログを集約した内容です。ただし、クラッシュログがAppleに送られるのは、クラッシュデータを開発者に渡すことに、ユーザが同意した場合に限ります。TestFlightのユーザは、自動的にこれに同意したことになります。Appleがクラッシュレポートを生成する手順は次の通りです。

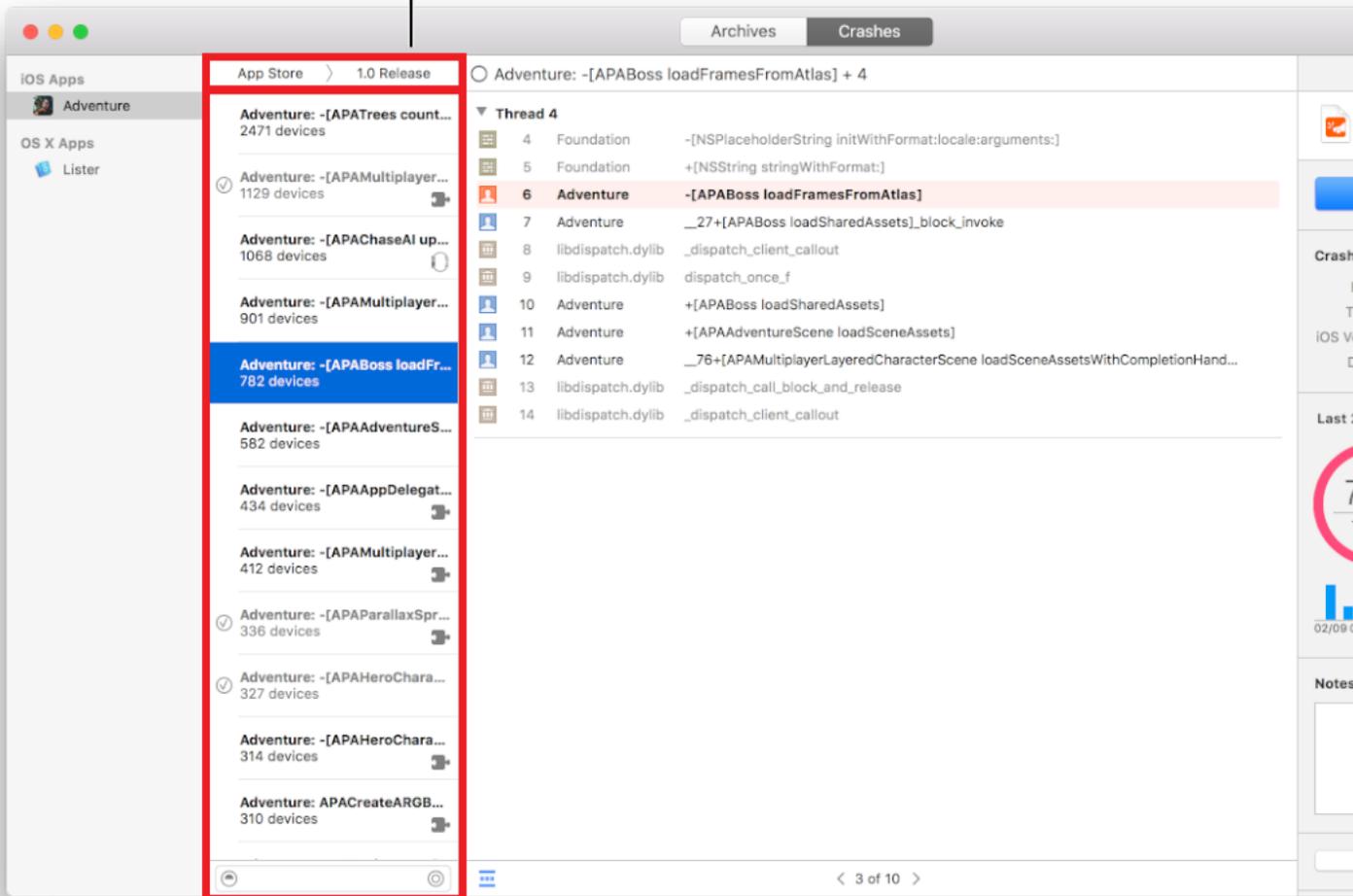
- ユーザデバイス上で動作するアプリケーションやエクステンションから、クラッシュログを収集する。
- クラッシュログをシンボル変換する（メモリアドレスを人が読んで理解しやすいシンボルに置換する）。
- スタックトレースを比較して同類のクラッシュログを特定し、集約してクラッシュレポートにまとめる。
- クラッシュログ中の個人情報をすべて削除する。
- クラッシュが発生したデバイスの異なり総数を求める。
- 各クラッシュレポートに、該当するクラッシュログの例を組み入れる。

- クラッシュレポートを日次で更新する。

クラッシュレポートに関する情報はXcodeの「Crashes」オーガナイザに表示されます。

App and archive list

Version and build menu



Crash report list

Crash report detail area

クラッシュレポートを閲覧する準備

クラッシュレポートサービスを十分に活用するためには、以下に示す準備が必要です。これまでにアプリケーションを（シンボル情報とともに）iTunes Connectにアップロードし、TestFlightやストアを通して配布したことがあれば、「Crashes」オーガナイザでクラッシュレポートを目にしたことがあるでしょう。まだ見たことがなければ、以下の作業を適切に実施したかどうか確認してください。

	ステップ
<input checked="" type="checkbox"/>	Xcodeで、Apple ID（Apple Developer Programに参加している旨の証し）を「Accounts」環境設定に入力する（「 Xcodeで自分のApple IDアカウントを追加する 」（26 ページ）を参照）。
<input checked="" type="checkbox"/>	版番号とビルド番号を設定する（「 版番号とビルド文字列を設定する 」（41 ページ）を参照）。別のビルドを配布する際には、ビルド番号を増やす。
<input checked="" type="checkbox"/>	単一のアプリケーションアーカイブを作成し、シンボル情報とともにiTunes Connectにアップロードする（「 アプリケーションをiTunes Connectにアップロード 」（121 ページ）を参照）。このとき、「Include app symbols for your application...」をオンにしてから「Submit」ボタンを押すこと）。 iTunes Connectにアップロードしたアーカイブは削除しない。
<input checked="" type="checkbox"/>	iTunes Connectで、TestFlightを使ってアプリケーションをテスト担当者に配布する（「 TestFlight Beta Testing (Optional) 」を参照）。または、審査用にアプリケーションを登録する（「 Submitting the App 」を参照）。
<input checked="" type="checkbox"/>	アプリケーションをストアからダウンロードしたユーザーは、開発者にクラッシュデータを提供することに同意する必要がある（「 クラッシュデータをアプリケーション開発者に提供する 」（141 ページ）を参照、TestFlightのユーザーは、自動的にこれに同意したことになる）。
<input checked="" type="checkbox"/>	Xcode上にクラッシュレポートを表示するためには、チームエージェントまたは管理者としての、またはAdmin/TechnicalロールのiTunes Connectユーザとしての、Apple IDを入力する必要がある（「 Xcodeで自分のApple IDアカウントを追加する 」（26 ページ）を参照）。
<input checked="" type="checkbox"/>	Xcodeの「Crashes」オーガナイザに表示されたスタックフレームから、デバッグナビゲータに開いたソースコードの該当箇所に移動できるよう、アーカイブを作成しiTunes Connectにアップロードする際に用いたXcodeプロジェクトを開く。

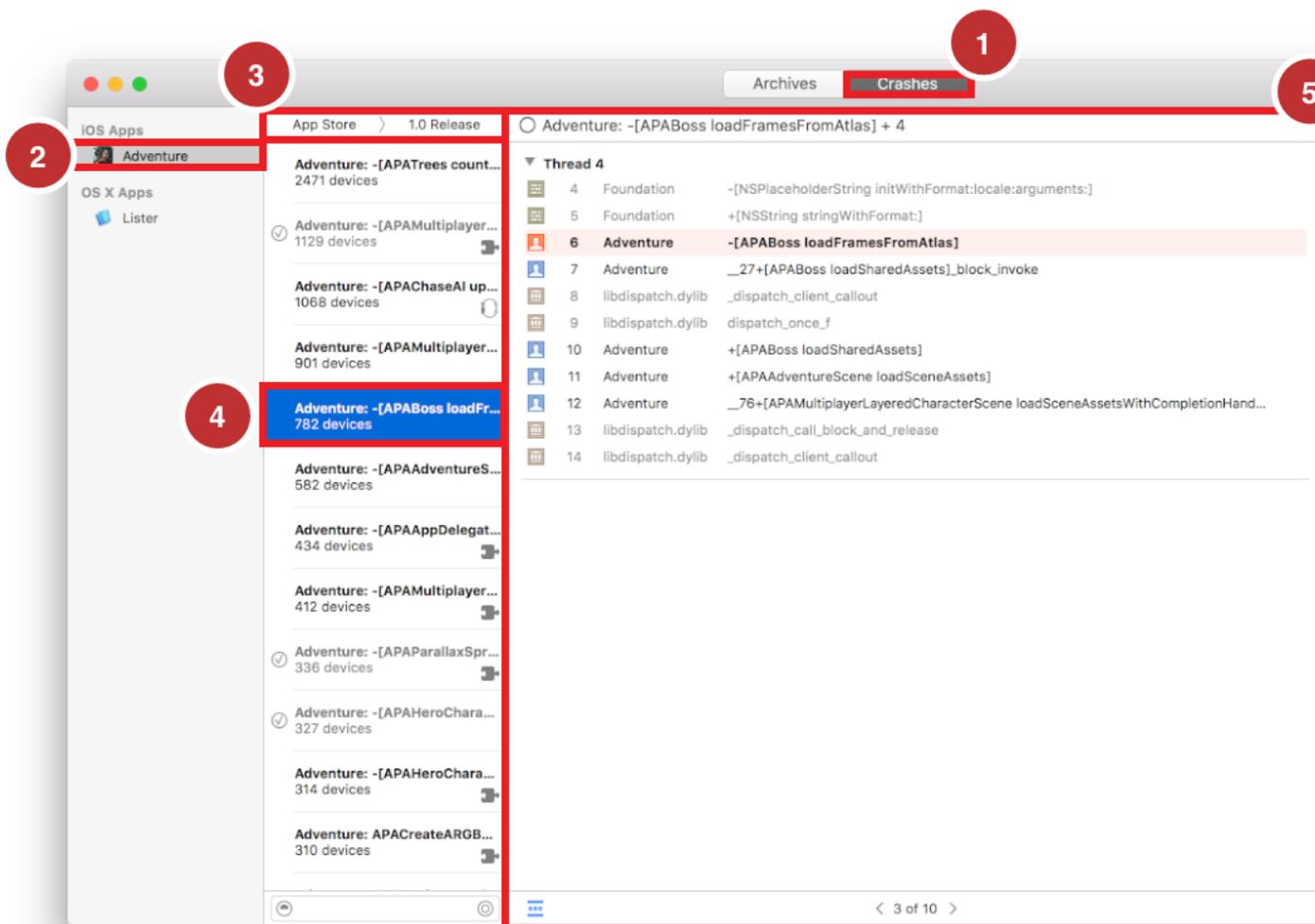
なお、アプリケーションの配布を始めてから、クラッシュレポートがXcodeに表示されるようになるまで、3日程度を要します。プライバシーに配慮して、クラッシュデータは、ユーザが同意した場合に限ってAppleに送信されるようになっています。

「Crashes」オーガナイザでクラッシュレポートを閲覧する

「Crashes」オーガナイザには、チームが開発したアプリケーションすべてについて、クラッシュレポートが表示されます。

クラッシュレポートを表示するには

1. オーガナイザの「Crashes」を押してください。



2. 左側の列に並んでいる中から、アプリケーションまたはローカルアーカイブを選択します。
このリストはiTunesConnectから取得したもので、アップロードしたすべての版やビルドに関する情報があります。
3. 第2列の一番上にあるポップアップメニューで、版とビルド番号を選択してください。

Xcodeは、該当する版とビルド番号のクラッシュレポートを、最新のものに更新します。ダウンロードするのは、同一デバイス上で発生したクラッシュ回数が多い順に、25位まで（過去2週間分が対象）のクラッシュレポートです。

4. 第2列からクラッシュレポートを選択してください。

見出しは通常、クラッシュしたと思われる箇所のスタックフレームを表すようになっています。その下の数字は、クラッシュしたデバイスの異なり数です。WatchKitエクステンションに関するクラッシュレポートは「時計」アイコン（）、それ以外は「パズル」アイコン（）で表します。

5. 詳細領域でクラッシュログを閲覧します。

ここにはスレッドごとのスタックトレースが表示されます。クラッシュした箇所のスタックフレームは、橙色の強調表示になります。

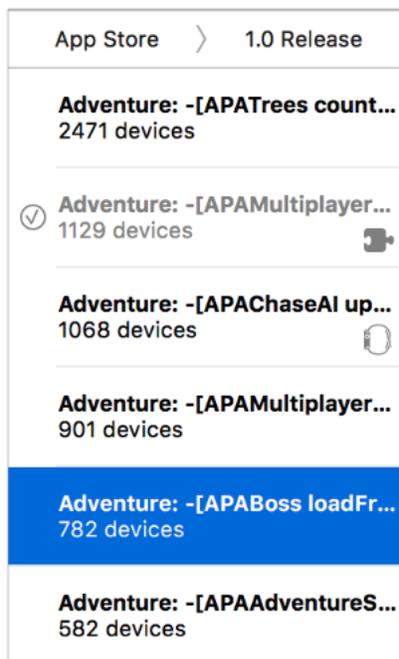
6. インスペクタ上で、情報を追加し、クラッシュレポートに関する統計情報を見ることができます。

クラッシュレポートを表示、検索する

クラッシュレポートのリストに施せる操作：

- 一番下の検索欄に文字列を入力し、クラッシュレポートを検索できます。

- クラッシュレポートに収録されているクラッシュログを、Finder上に表示できます。Controlキーを押しながら当該クラッシュレポートをクリックし、「Show in Finder」を実行してください。



詳細領域で実行できる操作：

- クラッシュレポートに収録されている、前/次のクラッシュログに切り替えて表示できます。フッタにある、ページを表す矢印（「<」または「>」）を押してください。また、ページ番号の表示（「11 of 20」など）をクリックし、ポップアップメニューからページ番号を選択することにより、当該ページに移動できます。
- クラッシュログの折りたたみ表示/展開表示を切り替えることができます。フッタの左下隅にあるフィルタアイコン（）を押してください。折りたたんだ状態では、アプリケーション側のスタックフレームしか表示されません。展開表示にすると、フレームワークやシステムのスタックフレームも現れます。

クラッシュレポートに関する情報を編集する

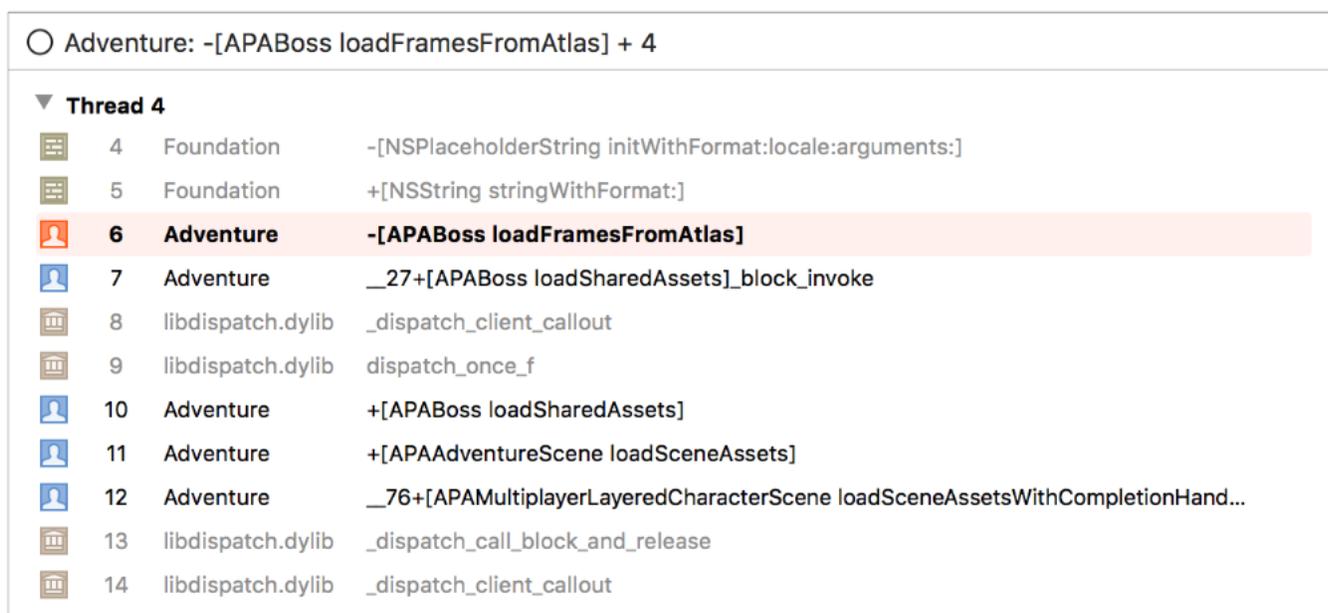
問題を解消する作業をしながら、クラッシュレポートに関する情報をローカルに保存しておくことができます。

クラッシュレポートのリストに施せる操作：

- 問題が解消した旨の印をつけることができます。クラッシュレポートの隣、またはインスペクタの下部にある円を選択し、「Mark as Resolved」を押してください。
- 解決済みのクラッシュレポートは、フッタの、検索欄の隣にある円 (⊙) で、表示/非表示を切り替えることができます。

詳細領域で実行できる操作：

- クラッシュレポートの見出しを変更できます。ヘッダにカーソルを移動し、テキストを編集してください。



インスペクタで実行できる操作：

- クラッシュレポートに関するメモを残せます。「Notes」欄に入力してください。

ソースコードをデバッグナビゲータで開く

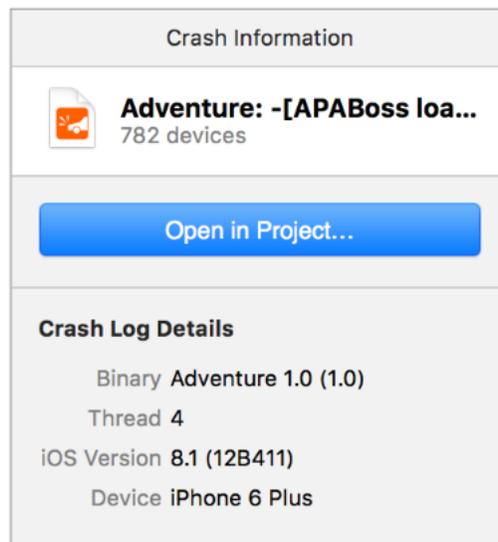
詳細領域で実行できる操作：

- ソースコードの、スタックフレームに対応する箇所を参照できます。スタックフレーム上にマウスカーソルを持っていくと右側に矢印が現れるので、これをクリックしてください。

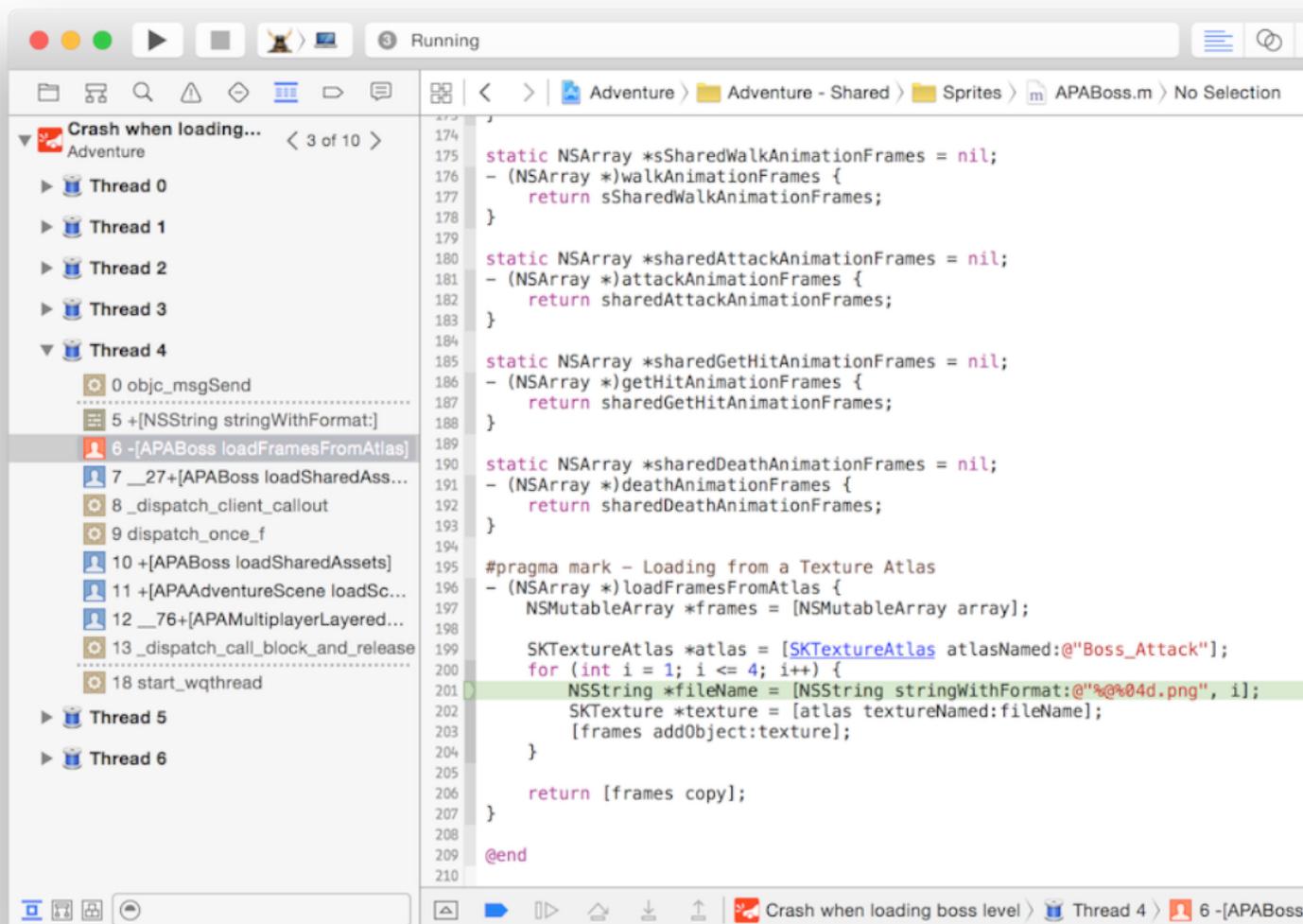
- クラッシュした箇所に対応するソースコードを参照できます。強調表示されているスタックフレーム上にマウスカーソルを持っていくと右側に矢印が現れるので、これをクリックしてください。

インスペクタで実行できる操作：

- ソースコードの、スタックフレームに対応する箇所を参照できます。「Open in Project」を押してください。

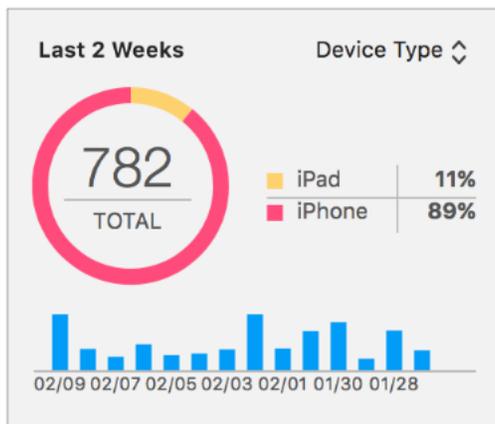


Xcodeは該当するプロジェクトを開き、デバッグナビゲータに、該当するコードの行を表示します。



クラッシュレポートに関する統計情報を表示する

過去2週間分のクラッシュデータを、デバイスの種類別、オペレーティングシステム別、日付別に表示できます。デバイスの種類は、ポップアップメニューの「Device Type」で変更します。オペレーティングシステムも同様に、ポップアップメニューの「Operating System」で変更します。



クラッシュデータをアプリケーション開発者に提供する

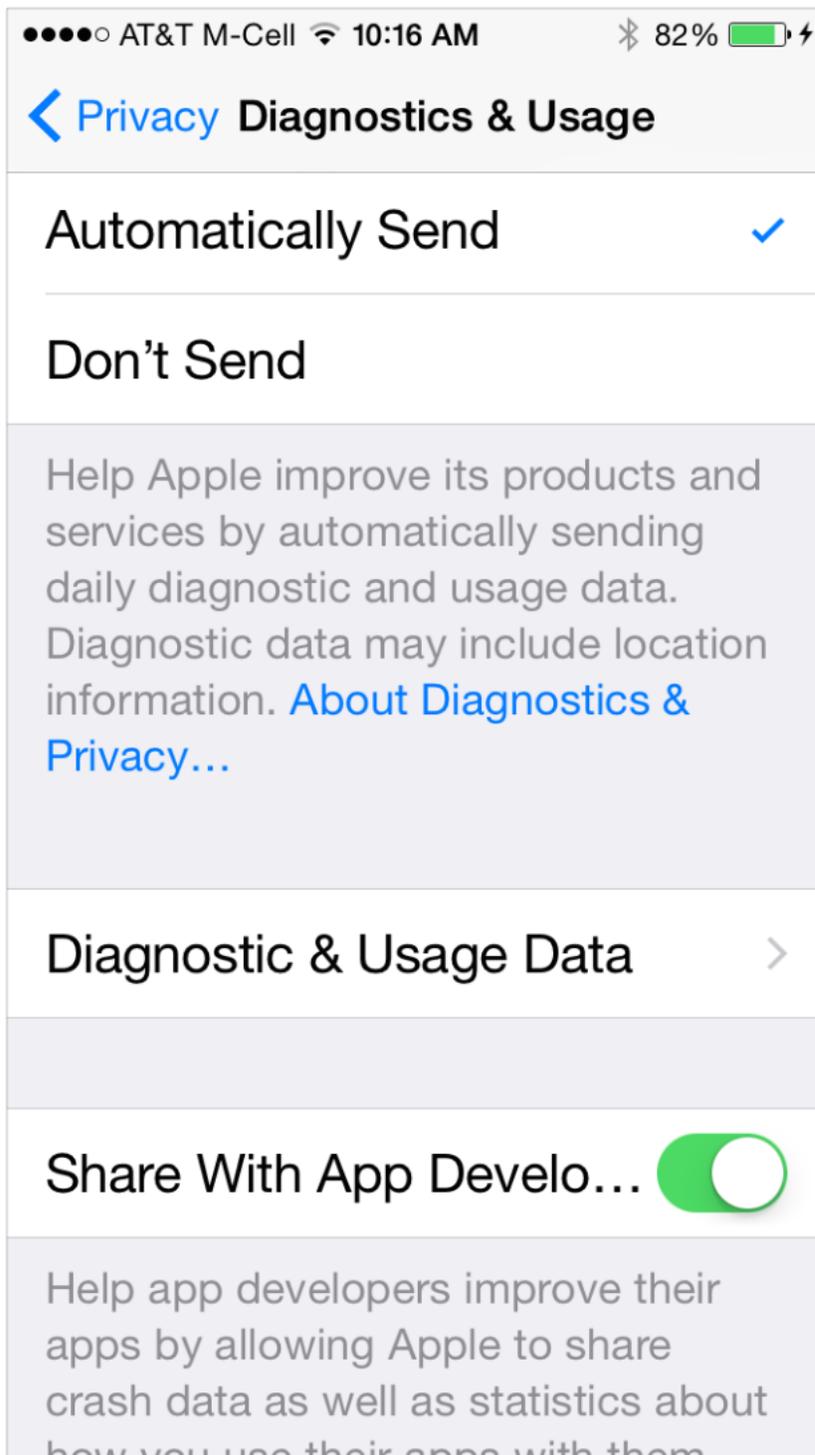
ユーザからクラッシュデータが提供されなければ、Appleがこれを開発者に提供することもできません。クラッシュした旨の報告だけで、対応するクラッシュレポートがない場合は、当該ユーザに、クラッシュデータを提供するよう依頼してください。TestFlightで配布した場合、クラッシュデータは自動的にAppleに送信されますが、ユーザーがストアからダウンロードした場合はそうなりません。

以下にクラッシュデータを提供するための手順を示すので、ユーザーに伝えてください。プラットフォームにかかわらず、クラッシュデータに関するオプションは、「Privacy」設定の「Diagnostics」セクションにあります。

クラッシュデータをiOS開発者に提供するには

1. 「Settings」を起動し、「Privacy」をタップします。

- 下にスクロールし、「Diagnostics & Usage」をタップしてください。



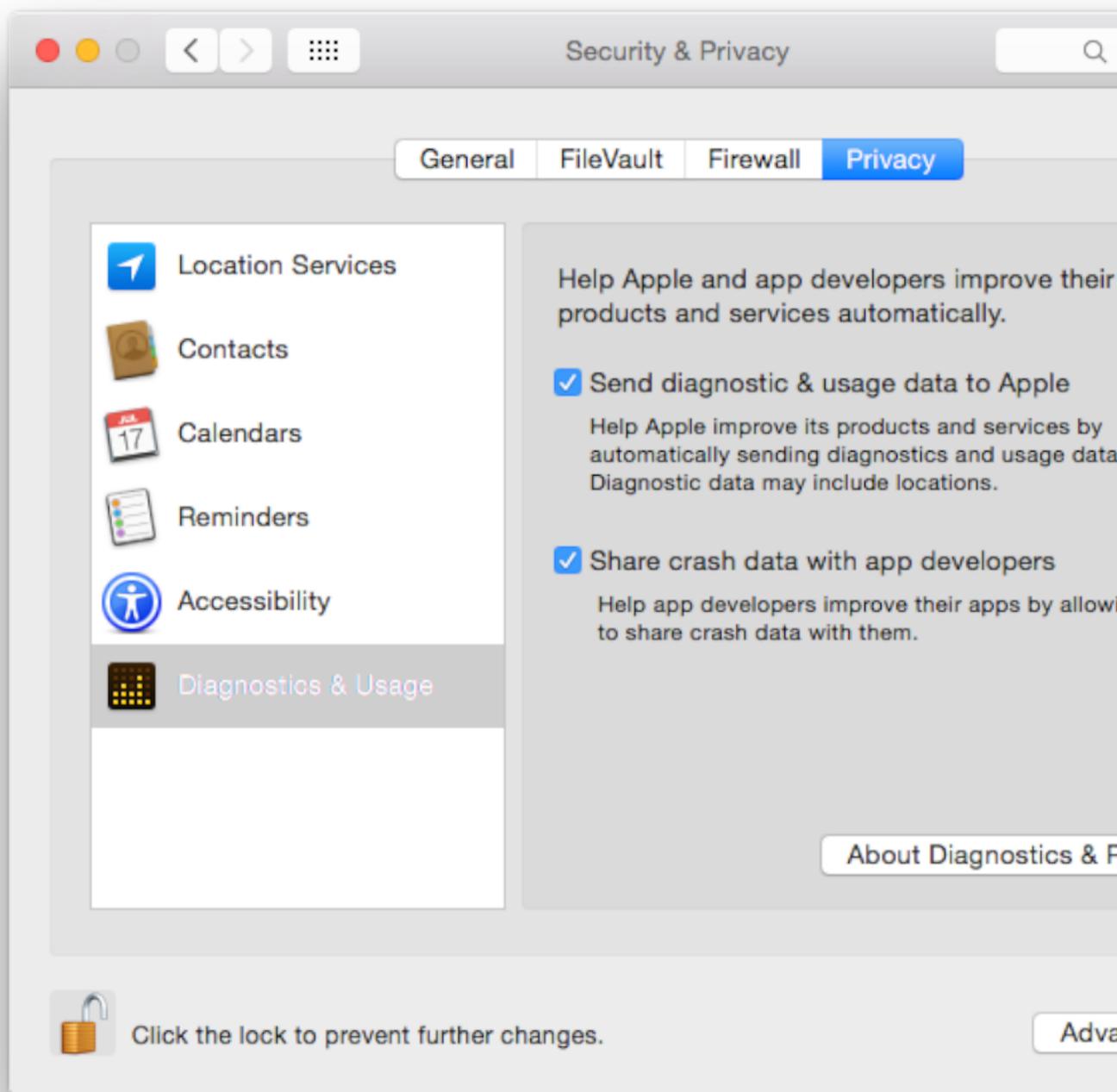
- 必要ならば「Share With App Developer」スイッチをオンにします。

Macユーザにも同様に、クラッシュデータをアプリケーション開発者に提供するように依頼してください。

クラッシュデータをMac開発者に提供するには

1. システム環境設定で、「セキュリティとプライバシー」をクリックします。

2. 「Privacy」に切り替え、左側の列にある「Diagnostics & Usage」をクリックしてください。



3. 「Share crash data with app developers」をオンにします。

「Privacy」オプションを有効にするには、まず鍵のアイコンをクリックしてください。するとダイアログが現れるので、管理者権限があるシステムアカウントの証明書を入力し、「Unlock」を押します。

Apple TVユーザーは自分のクラッシュデータも共有できます。

クラッシュデータをtvOS開発者に提供するには

1. 「Settings」画面の「General」、次いで「Privacy」を押してください。
2. 「Diagnostic Data」以下の「Share with App Developers」をクリックします。
3. するとダイアログが現れるので、「Send to Apple」を押してください。

クラッシュデータを「Devices」ウインドウに表示し、インポートする

「Devices」ウインドウでは、クラッシュログの詳細をデバイス上に直接表示し、あるいは他の情報源から得たクラッシュレポートをインポートすることができます。該当するビルドおよび.dSYMファイルを収容したアーカイブがあれば、Xcodeは自動的にクラッシュレポートを**シンボル変換**します。これは、メモリアドレスを、人が読んで理解しやすいシンボルに置換する処理です。その結果、各実行スレッドのスタックトレースを、「Devices」ウインドウに表示できるようになります。

そのためにも、アップロード、エクスポートし、ユーザーに配布したアーカイブは、削除しないようにしてください。たとえば、アドホックプロビジョニングプロファイルまたはチームプロビジョニングプロファイルを使ってアプリケーションを配布する（[“テスト用にアプリケーションをエクスポート \(iOS、tvOS、watchOS\)”](#)（102 ページ）および[“アプリケーションをテスト用にエクスポート \(Mac\)”](#)（114 ページ）を参照）場合、テスト担当者から届いた、シンボル変換していないクラッシュレポートをインポートできるよう、アーカイブを保存しておいてください。

クラッシュレポートはTestFlightやストアでも表示できます。アプリケーションと同時にビットコードもアップロードすると、ストアは.dSYMファイルを保存します。これをダウンロードしない限り、「Devices」ウインドウにクラッシュレポートを表示することはできません。ビットコードを収容しなかった、あるいはストアを経由せずに配布した場合、アーカイブには既に.dSYMファイルが入っています。

注意: シンボル情報つきでアプリケーションをアップロードした場合、代わりに「Crashes」オーガナイザにクラッシュレポートを表示できます（「[「Crashes」オーガナイザでクラッシュレポートを閲覧する](#)」（135 ページ）を参照）。

アップロードの際にビットコードも含めていた場合、対応する.dSYMファイルをまずダウンロードしてください。

dSYMファイルをダウンロードするには

1. 「Archives」オーガナイザで、該当するアーカイブを選択します。
2. 「Download dSYMs」ボタンを押してください。

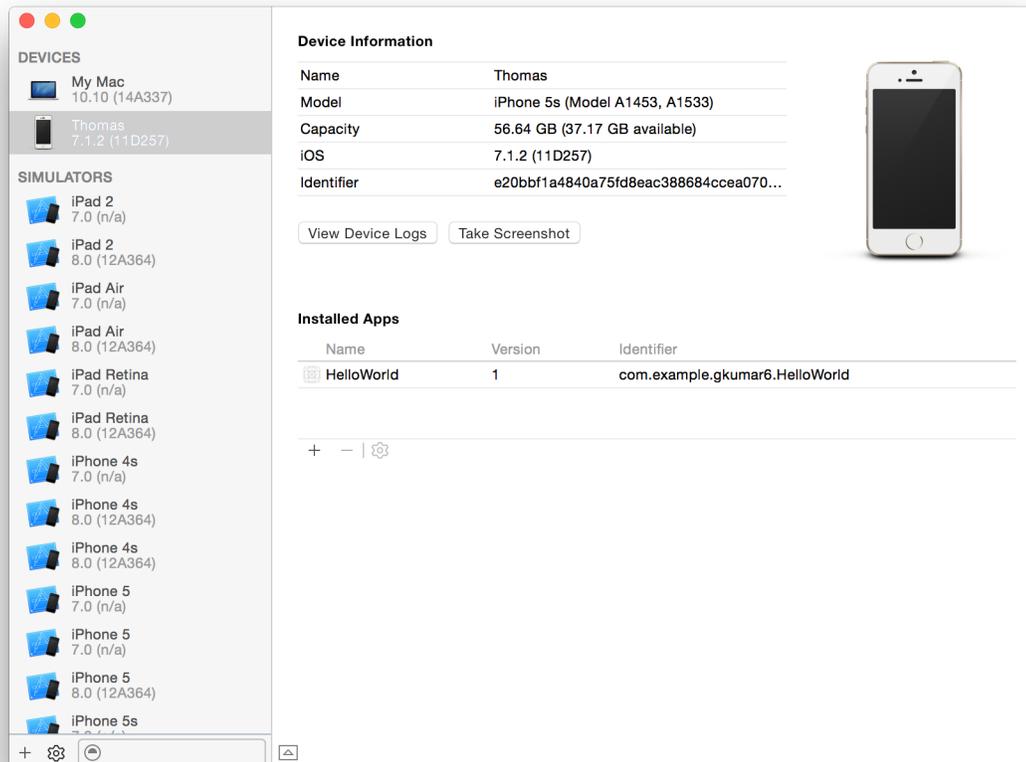
Xcodeは.dSYMファイルを、選択されたアーカイブに挿入します。

Important: Xcodeはアーカイブの版番号およびビルド文字列を調べて、必要な.dSYMファイルを要求します。したがって、アップロードの際には必ず、事前にビルド番号を増やすようにしてください（[「版番号とビルド文字列を設定する](#)」（41 ページ）を参照）。

デバイスにクラッシュログを表示し、あるいはクラッシュレポートをインポートするには

1. 「Window」 > 「Devices」 コマンドを実行してください。
2. 必要な場合は、デバイスをMacに接続します。

3. 「Devices」 から該当するデバイスを選択して「View Device Logs」 ボタンをクリックします。



そのデバイスのクラッシュログを含むシートが表示されます。

4. クラッシュログを表示するには、左側の列で該当するレポートを選択します。
そのクラッシュログが右側の詳細ビューに表示されます。
5. クラッシュレポートをインポートするには、該当するレポートをFinderからシートの左側の列までドラッグします。
6. 「Done」を押してください。

Xcode上でクラッシュを再現する

テストでは、クラッシュしたビルドとまったく同じものを使わなければなりません。テスト用に配布し、あるいはiTunes Connectにアップロードしたアーカイブは、すべて保存しておいてください。また、Xcode側のアーカイブとクラッシュレポートについて、ビルド番号および版番号が合致しているか確認してください。ストアに登録したアプリケーションバイナリと同じかどうか確認する必要がありますが、その手順も同様です。

Macアプリケーションの場合、クラッシュレポートに対応する版のOS Xを新規にインストールし、クラッシュを再現してください。開発者や管理者のアカウントを使ってはなりません。問題が再現しないかも知れないからです。

表示可能なクラッシュレポート

クラッシュログを収集する際に用いたオペレーティングシステムの版と、これを表示できるXcodeの版との対応を示します。

クラッシュレポートの対象	収集に用いたOS	表示可能なXcode
iOSアプリケーション	iOS 8.2以降	Xcode 6.3以降
iOSアプリケーション	iOS 8.3以降	Xcode 7.0以降
Macアプリケーション	OS X 10.10以降	Xcode 7.0以降
Macアプリケーションエクステンション	OS X 10.11以降	Xcode 7.1以降
WatchKitエクステンション	watchOS 2アプリケーション (iOS 9.1アプリケーションとペアリングしたもの)	Xcode 7.1以降

アプリケーションの登録

配布の手続きについて説明していますが、続いての手順は、アプリケーションをストアに登録する作業です。最終テストの際に配布した、最新のアーカイブを登録するよう推奨します。テストした最終ビルドは、製品として十分な品質であり、iTunes Connectの検証テストに合格するものでなければなりません。

アプリケーションの登録に向けて準備する

最終候補を登録する前に、ヒューマンインターフェイスおよびストアに関するガイドラインを確認し、必要な情報をiTunes Connectに入力し、Xcodeプロジェクトの設定を確認してください。

ヒューマンインターフェイスおよびストアに関するガイドラインを確認する

認定手続きを円滑に進めるため、以下のガイドラインを確認し、問題がある場合はこの時点で対処し、アプリケーションを再テストしてください。

- ユーザインターフェイスガイドラインについて：

iOS Human Interface Guidelines

Apple TV Human Interface Guidelines

Apple Watch Human Interface Guidelines

OS X Human Interface Guidelines

- ストアに関するガイドラインについて：

App Store Review Guidelines for iOS Apps

App Store Review Guidelines for Mac Apps

WatchKitアプリケーションについては、「[Preparing Your App Submission for Apple Watch](#)」を参照してください。アプリケーション審査プロセスについて詳しくは、「[App Review](#)」で確認してください。

必要な追加情報をiTunes Connectに入力する

iTunes Connectはウェブツールで、ストアを通して販売するアプリケーションについて、必要な情報を入力するために使います。iTunes Connectは、アプリケーションに関するメタデータ（Xcodeを使ってアップロードする版番号やビルド番号など）を、すべて保存しています。登録に先立ち、アプリケーションの種類に応じて、必要な情報をすべて入力してください（“[Viewing and Changing Your App’s Metadata](#)”を参照）。メタデータについては“[iTunes Connect App Properties](#)”に解説があります。

Xcodeプロジェクトを検証する

Xcodeプロジェクトが適切に設定されているか、以下の項目を検証してください。

- Xcodeプロジェクトの設定を確認します。“[アプリケーションを配布するためのXcodeプロジェクトの設定](#)”（30 ページ）を参照してください。
- Xcodeプロジェクトで設定したバンドルID（“[バンドルIDを設定する](#)”（35 ページ）を参照）が、iTunes Connectに入力したものと合致しなければなりません。

注意: iTunes ConnectにアップロードしたビルドのバンドルIDは変更できません。

- 開発に用いたのと同じApp IDを使って、アーカイブにコード署名します。
特定App IDを要するアプリケーションサービスを利用しないのであれば、XcodeワイルドカードApp IDでも構いません。新規にApp IDを作成するのであれば、“[App IDを登録する](#)”（195 ページ）を参照してください。ただし、App IDを変更した場合は、ビルドをiTunes Connectにアップロードする前に、新しいApp IDを使ってもう一度テストしなければなりません。
- 版およびビルドの設定を確認します（“[版番号とビルド文字列を設定する](#)”（41 ページ）を参照）。iTunes Connectは公開前版の版番号とビルド番号をアーカイブから抽出します。
- 主要なアプリケーションサービスをアプリケーションで利用できるように、App IDの設定を確認します。“[技術やサービスを組み込む](#)”（56 ページ）を参照。

Macアプリケーションの場合は以下の作業も必要です。

- 開発中の署名方法として「Mac App Store」を選択します（“[署名IDを選択する \(Mac\)](#)”（37 ページ）を参照）。
- アプリケーションはすべて、App Sandboxを有効にしなければなりません（“[アプリケーションサンドボックスを設定する \(Mac\)](#)”（59 ページ）を参照）。
- アプリケーションやインストーラパッケージをMac App Storeに登録するためには、すべてコード署名が必要です。補助アプリケーションを使っている場合は、その設定方法について『[Daemons and Services Programming Guide](#)』を参照してください。

最終候補をアップロードする

アプリケーションをiTunes Connectにアップロードするのが初めてであれば、アーカイブの作成、検証、アップロード手順について、“[アプリケーションをiTunes Connectにアップロード](#)”（121ページ）を参照してください。TestFlightを使ってアプリケーションをテスト担当者に配布した場合（“[テスト用にアプリケーションをエクスポート（iOS、tvOS、watchOS）](#)”（102ページ）を参照）、実際に配布した最終ビルドを、App Reviewに直接登録できます。

アプリケーションを登録する

アプリケーションのビルドをAppReviewに登録する手順については、『[iTunesConnectDeveloperGuide](#)』の“Submitting the App”を参照してください。

まとめ

この章では、アプリケーションをAppReviewに登録するための準備をし、実際に登録する手順を解説しました。登録後の状況を確認する方法については、“[アプリケーションの状態を確認する](#)”（156ページ）を参照してください。

iTunes Connectでのアプリケーション管理

iTunes Connectを使えば、ストアからアプリケーションを配布する期間中、その各版をリリースするほか、さまざまな管理作業をおこなえます。iTunes Connectはマーケティング/ビジネス用のウェブツールでもあり、チームメンバー向けに、契約書に署名する、税や銀行に関する情報をセットアップする、販売/財務レポートを取得する、などの機能があります。開発期間中は、アプリケーションに関するメタデータ、アプリケーションで使用されるアプリケーションサービス、版情報などをiTunes Connectに入力します。この章では、開発中およびストアを通して配布した後に、iTunes Connectで実施する追加作業を説明します。

当初iTunes Connectにアクセスできるのは、Apple Developer Programに加入した者だけです。iTunes Connectは主として、ビジネス面におけるアプリケーション管理を目的としており、開発者が関与することはあまりないので、メンバーセンターのアカウントとは切り離して、iTunes Connectへのアクセス制御を厳格に行うことができます。たとえば、開発者以外のiTunes Connectユーザを追加し、役割や権限を設定して、メタデータへのアクセスを制御する、といったことが可能です。

iTunes Connectのユーザのうち、管理や技術の役割を担う者が、開発チームの支援を受けながら、アプリケーションをストアに登録する処理に関連して、この章で説明するさまざまな作業を行います。

1. iTunes Connectのユーザを追加して、他のチームメンバーがiTunes Connectにアクセスできるようにする。
2. アプリケーションレコードを作成して、主要なアプリケーションサービスを設定し、アプリケーションをアップロードできるようにする。
3. 登録の準備が整い、あるいは認定を待っている段階で、アプリケーションの状態を確認する。
4. アプリケーションの各版の公開日を変更する。
5. 公開後、解析結果や顧客の評価を閲覧する。
6. アプリケーションの新版を作成する。

さらに詳しくは、『*iTunes Connect Developer Guide*』を参照してください。

iTunes Connectユーザの役割と権限について

開発者プログラムに加入した者（チームエージェント）は、iTunes Connectへのアクセス権限を管理します。たとえば、アプリケーションの販売価格を変更する権限は、ごく少数に制限するべきでしょう。「iTunes Connect」ツールのアクセス権限は、チームメンバーに対して設定するアクセス権限とは独立に、しかもきめ細かく設定できるようになっています。ここでもやはり、各ユーザにいくつかの役割を割り当てることができ、役割ごとにそれぞれ異なる権限が対応しているのです。表 12-1 に、それぞれの役割を大まかに説明します（それぞれの役割については「[iTunes Connectユーザを追加し役割を確認する](#)」（154 ページ）を参照）。

表 12-1 iTunes Connectにおける役割と責任

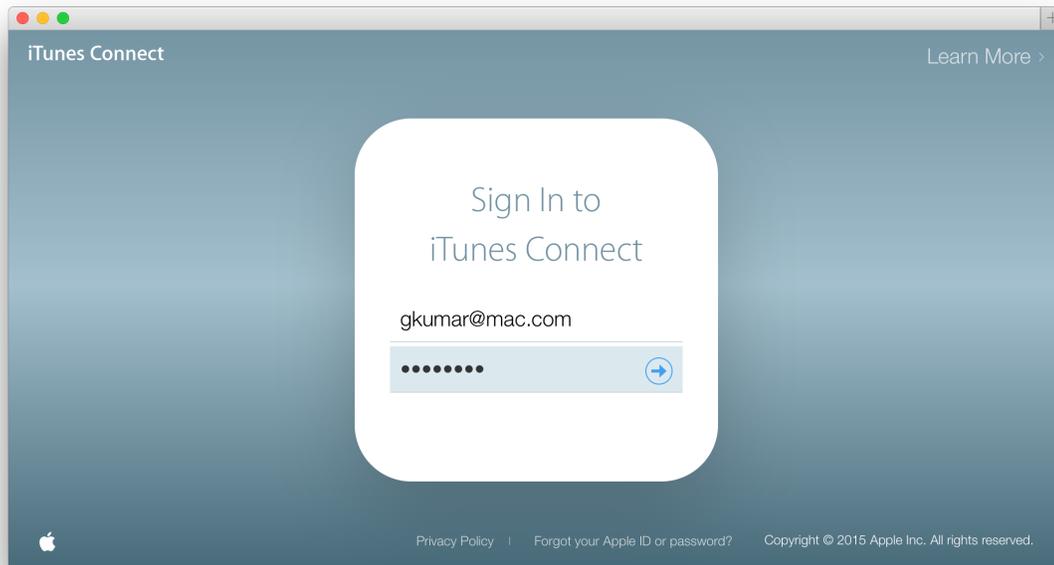
役割	責任
Admin	契約を除き、「iTunes Connect」におけるあらゆる作業が可能です。チームエージェントには必ず割り当てられる役割であり、チームエージェントを交代させない限り、この役割を奪うことはできません。「iTunes Connect」における役割を、各員に割り当てることができます。
Legal	チームエージェントに自動的に割り当てられる役割です。法的な契約書その他に署名する権限です。
Technical	iTunes Connectに格納されたアプリケーション情報を編集し、In-App Purchase など、特定のアプリケーションサービスを利用するためのテストユーザを管理できます。また、ビルドを iTunes Connect にアップロードし、アプリケーションを App Review に登録する権限を付与することもできます。
Finance	財務報告書、販売情報、アプリケーション解析にアクセスできます。また、契約、税務、銀行取引などの情報を閲覧できます。
Sales	販売データおよびアプリケーション解析にのみアクセスできます。
マーケティング	マーケティングデータにのみアクセスできます。

iTunes Connectにアクセスする

iTunes Connect は、ストアに関係するアセット（アプリケーションのビルドを含む）のいわば「倉庫」です。アプリケーションを販売/配布する、契約の状況を確認する、税や銀行に関する事項を設定する、販売/財務レポートを取得する、アプリケーションに関するメタデータを管理する、などの機能があります。同じ iTunes Connect アカウントに、複数のユーザがアクセスすることもできます。iTunes Connect にはメンバーセンターを経由してアクセスするほか、[iTunes Connect](#) のウェブサイトも直接開くことも可能です。

メンバーセンター経由でiTunes Connectにアクセスするには

1. [メンバーセンター](#)にサインインします。
2. iTunes Connectのアイコンまたはテキストをクリックしてください。
3. Apple IDおよびパスワードを入力し、矢印ボタンを押します。



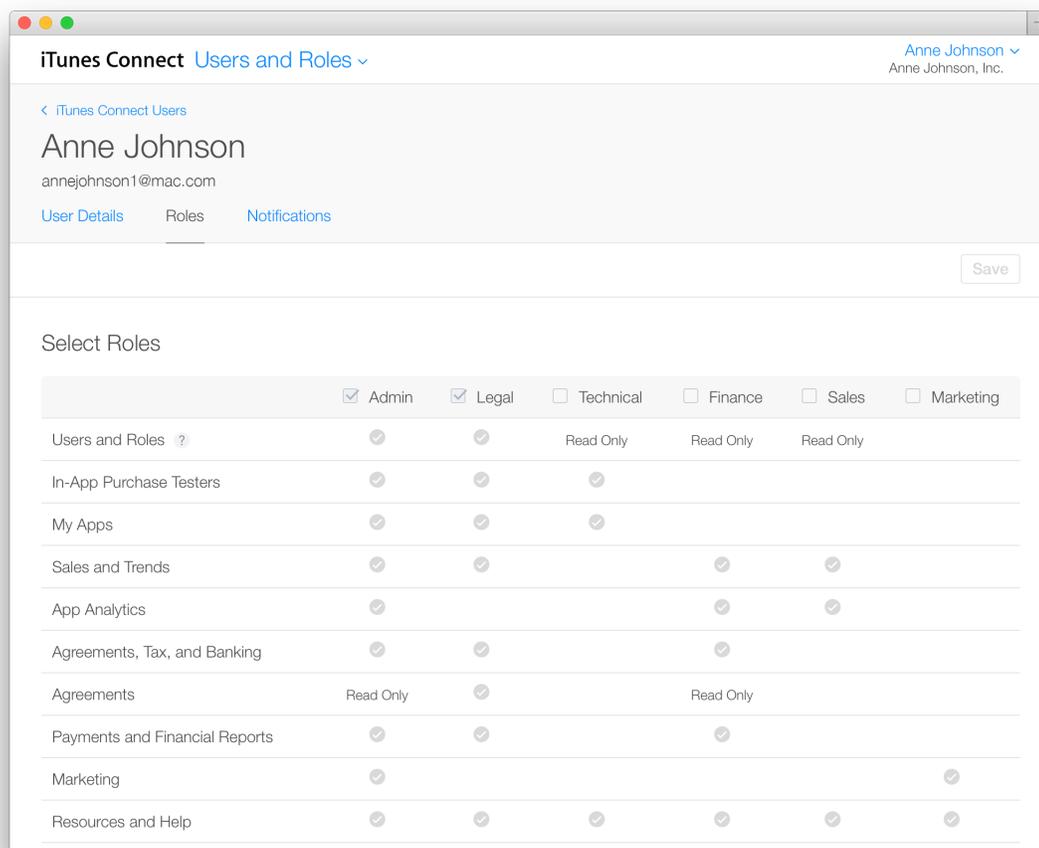
iTunes Connectユーザを追加し役割を確認する

チームエージェントは、iTunes Connectユーザを追加し、その役割を変更できます。ユーザの追加に先立ち、それぞれの役割に与えられた権限をiTunes Connectで確認してください。

それぞれの役割に与えられた権限を表示するには

1. [iTunes Connect](#)にサインインします。
2. 「Users and Roles」をクリックしてください。
3. ユーザリストから、該当するユーザのApple IDを選択します。
4. 「Roles」を押してください。

それぞれの役割に課された責任が、表の形で現れます。



iTunes Connect ユーザを追加し、その役割を変更する手順については、“Setting Up User Accounts” in *iTunes Connect Developer Guide* を参照してください。

アプリケーションレコードを作成する

特定のアプリケーションサービスを利用するには、開発の際にアプリケーションレコードを作成し、iTunes Connect を使ってバンドルIDを入力する必要があります。アプリケーションレコードは、後で配布用にアップロードする際にも必要です。アプリケーションレコードを生成できるようになったら、“Creating an iTunes Connect Record for an App” を参照してください。iTunes Connect に入力する他の情報や、アップロードしたアプリケーションの版番号とビルド番号を確認する手順については、“Viewing and Changing Your App’s Metadata” を参照してください。

tvOSアプリケーションをiOSアプリケーションレコードに追加する

watchOS/tvOSアプリケーションは、iOSアプリケーションと同じアプリケーションレコードを共有できます。watchOSアプリケーションは、iOSアプリケーションと同じXcodeプロジェクトに属します。tvOSアプリケーションはiOSアプリケーションと同じバンドルIDを共有しますが、独立したXcodeプロジェクトの形でアップロードしなければなりません。

tvOSアプリケーションをiOSアプリケーションレコードに追加するには

1. Xcodeで、tvOSアプリケーション用の独立したプロジェクトを作成してください。
2. tvOSターゲットを選択し、iOSアプリケーションと同じバンドルIDを設定します。
3. 版番号とビルド文字列を設定してください。
これはiOSアプリケーションと違っていても構いません。
4. [iTunes Connect](#) ホームページの「My Apps」をクリックします。
5. 編集するiOSアプリケーションをクリックします。
6. 左欄の「Version or Platform」ポップアップメニューから「tvOS」を選択してください。
tvOSアプリケーションのバージョンレコードが、「tvOS App」の下に現れます。
7. 版情報を入力するため、tvOSのバージョンレコードをクリックしてください。
tvOSアプリケーションに必要な情報を右側に入力します。
8. Xcodeで、tvOSアプリケーションをiTunes Connectにアップロードしてください。
iTunes ConnectはビルドをtvOSバージョンに追加します。

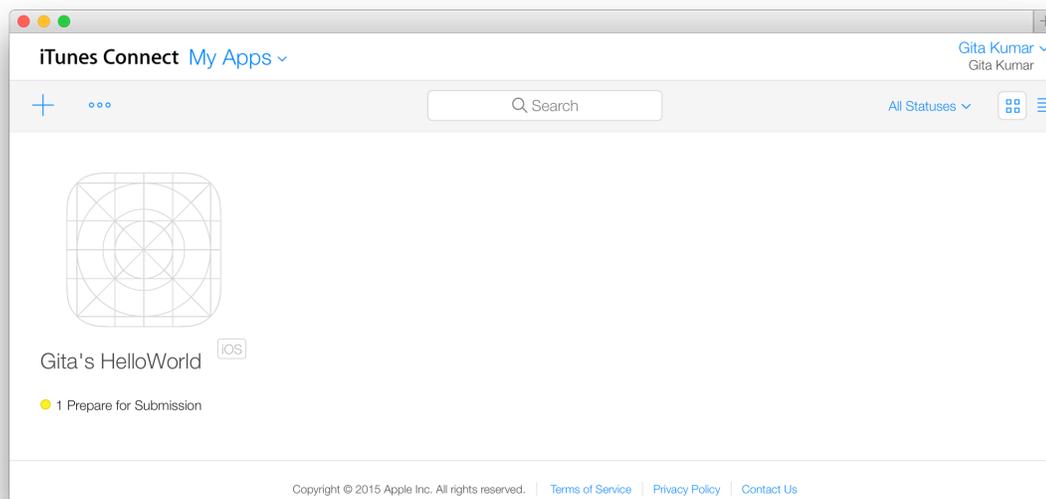
アプリケーションの状態を確認する

状態はiTunes Connectで確認できます。

アプリケーションの状態を確認するには

1. [iTunes Connect](#) にサインインしてください。
2. iTunes Connect ホームページの「My Apps」をクリックします。

3. 編集するアプリケーションを選択します。



各版のアプリケーションの状態がアイコンとタイトルの下に表示されます。

4. 大きいアイコンまたはアプリケーション名をクリックして、詳細を表示します。

それぞれの状態の詳細と状態の履歴を表示する方法については、“Viewing and Changing Your App’s Status and Availability”を参照してください。

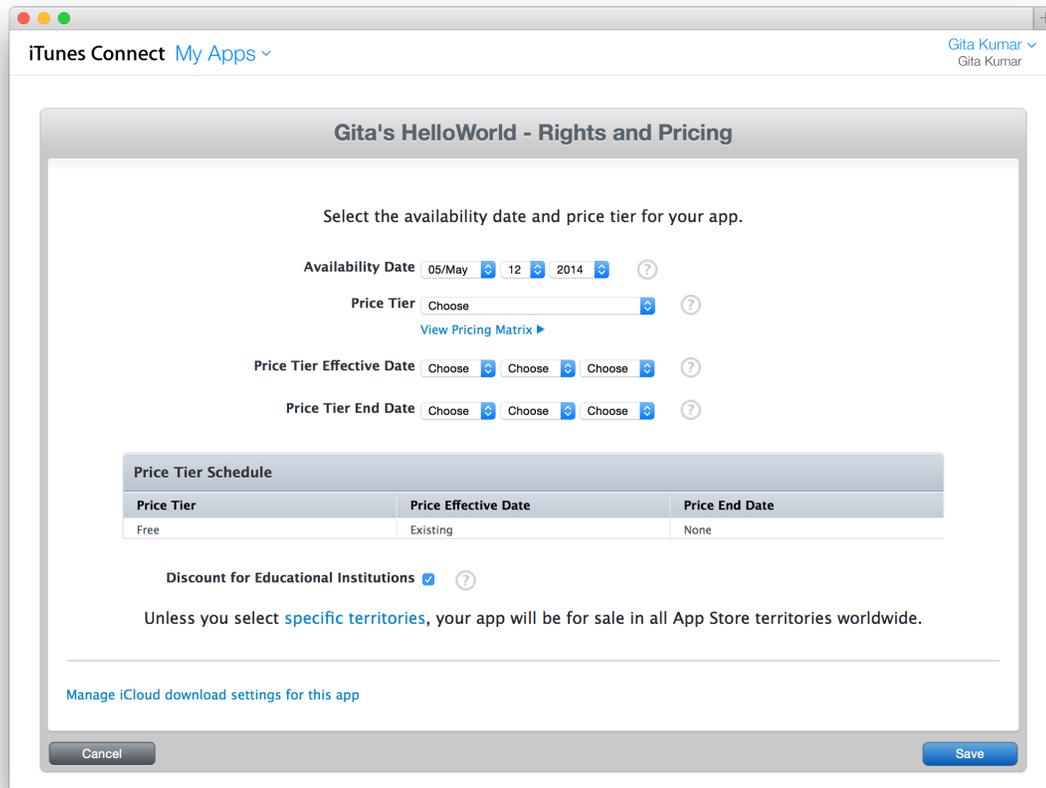
アプリケーションの公開日を変更する

iTunes Connect上で、アプリケーションをストアに公開する日を設定できます。認定後直ちに公開しても、将来の日付を設定しても構いません。マーケティング活動など、公開に向けて準備が必要であれば、それに応じた日付を設定するとよいでしょう。

公開日を設定するには

1. iTunes Connectにサインインしてください。
2. iTunes Connectホームページの「My Apps」をクリックします。
3. 該当するアプリケーションの場所を確認し、大きいアイコンまたはアプリケーション名をクリックします。
4. 「Pricing」をクリックします。

5. 「Availability Date」ポップアップメニューで日付を選択します。



6. 必要に応じ、そのほかのフィールドも編集できます。
7. 「Save」を押してください。

解析結果を表示する

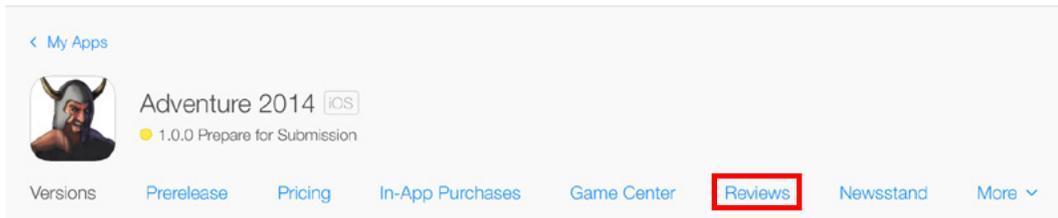
Admin、Finance、Salesのいずれかの役割が付与されていれば、アプリケーション解析を閲覧して、取得/使用/売上の状況を追跡できます。「iTunes Connect」ホームページ上で「App Analytics」をクリックすると、各アプリケーションについて収集したデータが閲覧可能です。解析結果の見方については、「App Analytics User Guide」を参照してください。

顧客レビューの表示

ストアで入手可能なアプリケーションに関する顧客からの最新のフィードバックを閲覧できます。

顧客の反応を表示するには

1. [iTunes Connect](#)にサインインしてください。
2. iTunes Connectホームページの「My Apps」をクリックします。
3. 該当するアプリケーションの場所を確認し、大きいアイコンまたはアプリケーション名をクリックします。
4. 「レビュー」をクリックします。



顧客による評価が一覧表示されます。

5. 表示する評価を選択します。
6. 「Done」をクリックします。

アプリケーションの新版を作成する

アプリケーションの新版を作成する手順については、“Replacing Your App with a New Version” in *iTunes Connect Developer Guide*を参照してください。

まとめ

この章では、iTunes Connectへのアクセス権限を付与する手順、開発に際して技術チームメンバーがよく実施する作業の手順を説明しました。

メンバーセンターでのチームの管理

組織としてApple Developer Programに参加する場合、他の人をチームに加えて役割を付与し、チームアセットに対するアクセスを許可することができます。チームメンバーには開発および配布のプロセスに応じた役割と権限が与えられます。「役割」が規定するのは、あるアプリケーションサービスを有効化できるのは誰か、誰がIDを生成できるか、誰が配布アセットを作成できるか、などといった事項です。チームメンバーを追加すれば、その人のために行う作業が生じるでしょう。たとえば、あるApp IDに対するケイパビリティを有効にする、APNs、Pass Type ID、iCloudコンテナを利用するためのクライアントSSL証明書を作成する、などといった作業です。個人としてApple Developer Programに加入している場合、この章で解説するチーム管理作業は必要ありません。

注意: チームメンバーとiTunes Connectのユーザは、同じとは限りません。当初はApple Developer Programに加入した者だけがiTunes Connectにアクセスできます。iTunes Connectのユーザを追加する手順については、「[iTunes Connectでのアプリケーション管理](#)」（152 ページ）を参照してください。

Apple Developer Programにおけるチームの役割と権限について

チーム各員の「役割」には、チームのアセットに対するアクセス権限のレベルや、メンバーセンターやXcodeを使って実行できる作業の種類を規定する働きがあります。この権限レベルによって、チームを代表して実行できる作業の種類が決まります。このように、チームの役割を制御できるようにしているのは、チームのセキュリティを管理しやすくするのが目的です。

注意: たとえば、ビルドをiTunes Connectにアップロードする権限やApp Reviewに登録する権限など一部の権限は、iTunes Connectユーザの役割で制御されます（「[iTunes Connectユーザの役割と権限について](#)」（153 ページ）を参照）。

チームの役割

表 13-1 に、チームの各員に割り当てることができる「役割」と、その簡単な説明を示します。いずれも、それより下位の役割に割り当てられた権限は行使可能です。

表 13-1 チームの役割

役割	説明
チームエージェント	チームエージェント は法律的にチームを代表し、Appleとの第一義的な連絡窓口になります。また、チームメンバーを加え、それぞれの権限レベルを変更できます。チームエージェントは1人に限ります。
チーム管理者	チーム管理者 は、チームエージェントを除く他のチームメンバーの権限レベルを設定できます。アプリケーションに署名するためのアセットを管理します。開発段階か、配布できる状態になった段階かを問いません。また、App IDを編集してアプリケーションサービスを有効にし、システム全体で使用する、ある技術に特化したIDを生成できます。アプリケーションに署名を施し、非開発用デバイスに配布することも可能です。
チームメンバー	チームメンバー はXcode上で、開発用証明書を作成し、自分が使うデバイスを登録し、チームプロビジョニングプロファイルを作成できます。

チームの権限

チームの各「役割」は、一連の権限、すなわち、各員が実行できる作業を規定します。表 13-2に、チームの各員に付与する権限を示します。チーム各員が失効させることができる証明書の種類については、[表 14-1](#)（186 ページ）を参照してください。

表 13-2 役割に応じた権限

権限	チームエージェント	チーム管理者	チームメンバー
法的な契約の締結	✓	✗	✗
メンバーシップの更新	✓	✗	✗
Developer ID証明書の作成	✓	✗	✗
メンバーの招待と役割の割り当て	✓	✓	✗
配布用証明書の作成	✓	✓	✗
App IDの登録、設定、削除	✓	✓	✗
配布用プロビジョニングプロファイルの作成	✓	✓	✗
他のアプリケーションサービス用IDの作成	✓	✓	✗

権限	チームエージェント	チーム管理者	チームメンバー
開発用デバイスの登録	✓	✓	✗
開発用プロビジョニングプロファイルの作成	✓	✓	✗
開発用証明書の作成	✓	✓	✓
ワイルドカードAppIDおよびチームプロビジョニングプロファイルの作成 (Xcode上)	✓	✓	✓

チームエージェント

最初はある1人が、Apple Developer Programに加入し、チームエージェントになります。この過程でチームエージェントは、Apple開発者になるための法的な契約書に署名し、ストアからの売上金決済に必要な財務書類を作成します。

チームエージェントは権限に何の制限もなく、「開発チーム」を対象とするあらゆる作業ができる一方、法的な責任も負います。当初はチーム編成作業の大部分も行います。チームにほかの人を加えれば、必要に応じてこの権限を委任し、作業を代行させることも可能です。

アプリケーションに特定のサービスを組み込む場合など、ライセンス契約を更新したり、新たに締結したりするために、チームエージェントの署名を求められることがあります。たとえばiAdサービスを利用する場合、別途契約書に署名しなければなりません。

チームメンバーを加えて役割を与える

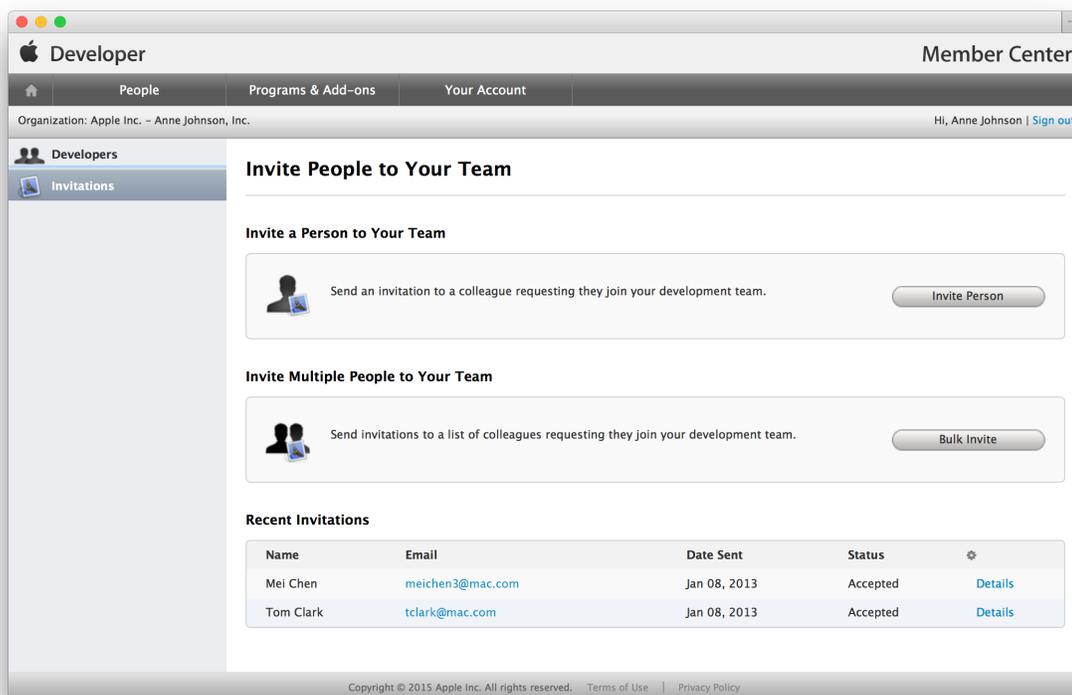
組織として加入すれば、チームエージェントになり、他の開発者をチームメンバーとしてアカウントに追加する権限を得ます。一般のチームメンバーには、メンバーセンターで管理されている情報を表示、ダウンロードする権限がありますが、大部分のアセットに関して、書き込みはできません。しかし、管理者としての役割のあるチームメンバーに付与し、チームエージェントの権限の一部を委ねることも可能です。チーム管理者は、たとえばアプリケーションサービス用のIDを作成できますが、契約に署名することはできません。このように、役割を付与することによって、責任の一部を委ねることができるようになっています。

チームメンバーを加える

チームメンバーを加えるためには、新しいメンバーに関する情報を入力し、役割を付与する必要があります。

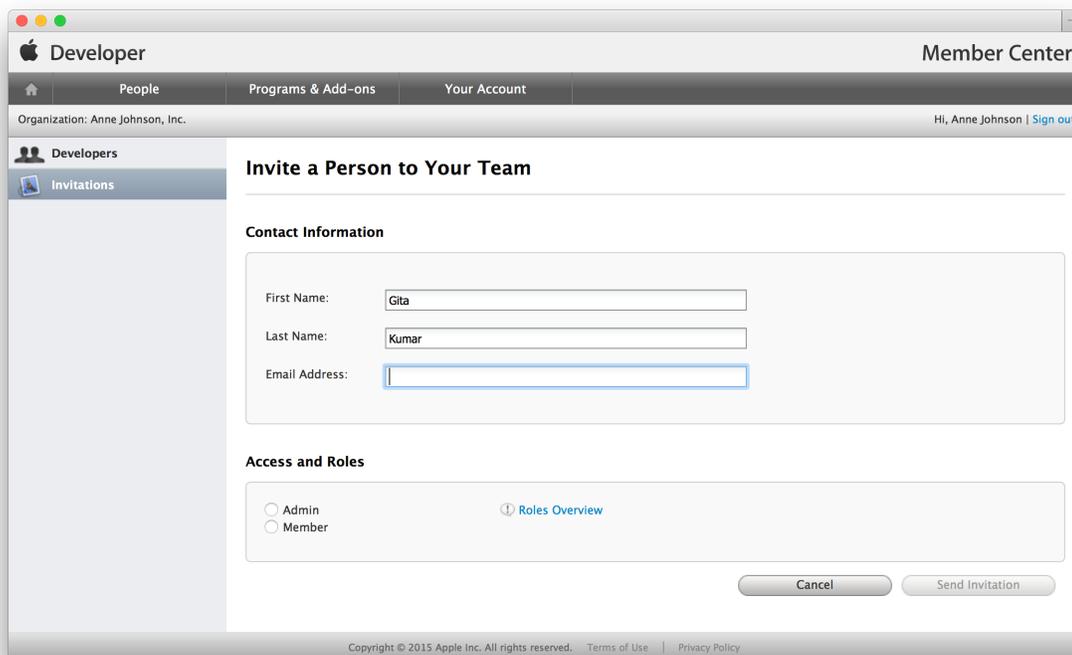
チームメンバーを加えるには

1. [メンバーセンター](#)のウェブページ上部にある「People」をクリックします。
2. サイドバーの「Invitations」を押します。
3. 「Invite Person」を押してください。



4. 追加する人の姓、名前、電子メールアドレスを入力します。
5. 各プラットフォームにおけるアクセス権限と役割を指定します。

それぞれの役割についての詳細は、「Roles Overview」を押すと表示されます。



6. 「Send Invitation」を押してください。

すると参加を勧誘する電子メールが本人宛に送信されます。受け取った側では、応じる旨の確認のため、招待コードをクリックします。ただし、Apple IDを持っていない場合は、あらかじめ作成してから招待に応じなければなりません。

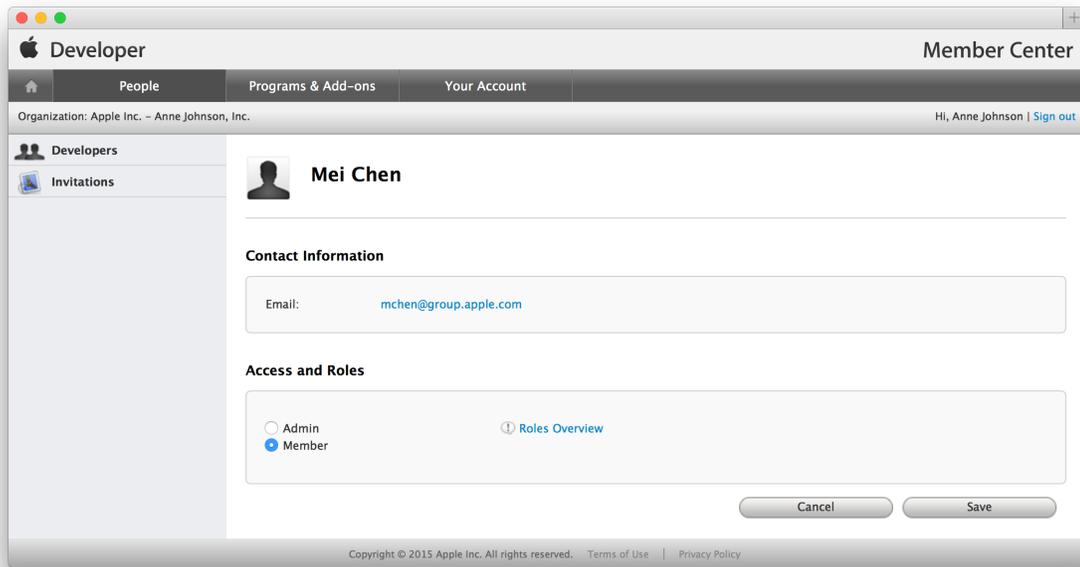
チームの役割を変更する

チームメンバーが招待に応じると、チームエージェントに確認の電子メールが届き、チームメンバーはメンバーセンターにアクセスできるようになります。チームエージェントはいつでもチームメンバーの役割を変更できます。

チームメンバーの役割を変更するには

1. [メンバーセンター](#)のウェブページ上部にある「People」をクリックします。
2. サイドバーの「Developers」を押します。
3. 該当する人を表す行の右端にある「Details」を押してください。
4. 役割を指定して「Save」を押します。

それぞれの役割についての詳細は、「Roles Overview」を押すと表示されます。



チームメンバーのデバイスを登録する

チームメンバーが自分のデバイス上でアプリケーションを起動するためには、当該デバイスを登録し、チームプロビジョニングプロファイルに追加する必要があります。Xcodeは必要に応じて自動的にデバイスを登録します（[“デバイス上でアプリケーションを起動する手順”](#)（91ページ）を参照）。ただし、チームメンバーのデバイスの登録は、チームエージェントまたはチーム管理者が、メンバーセンター上で代行することも可能です。

チームメンバーは、デバイス名とデバイスIDをチーム管理者に通知します。チームメンバーはXcodeの「Devices」ウインドウで該当するデバイスを選択して、デバイスIDを確認できます（[“XcodeでデバイスIDを調べる”](#)（202ページ）を参照）。開発者であれば、「System Information」アプリケーションで調べることも可能です（[“「System Information」でデバイスIDを調べる”](#)（204ページ）を参照）。

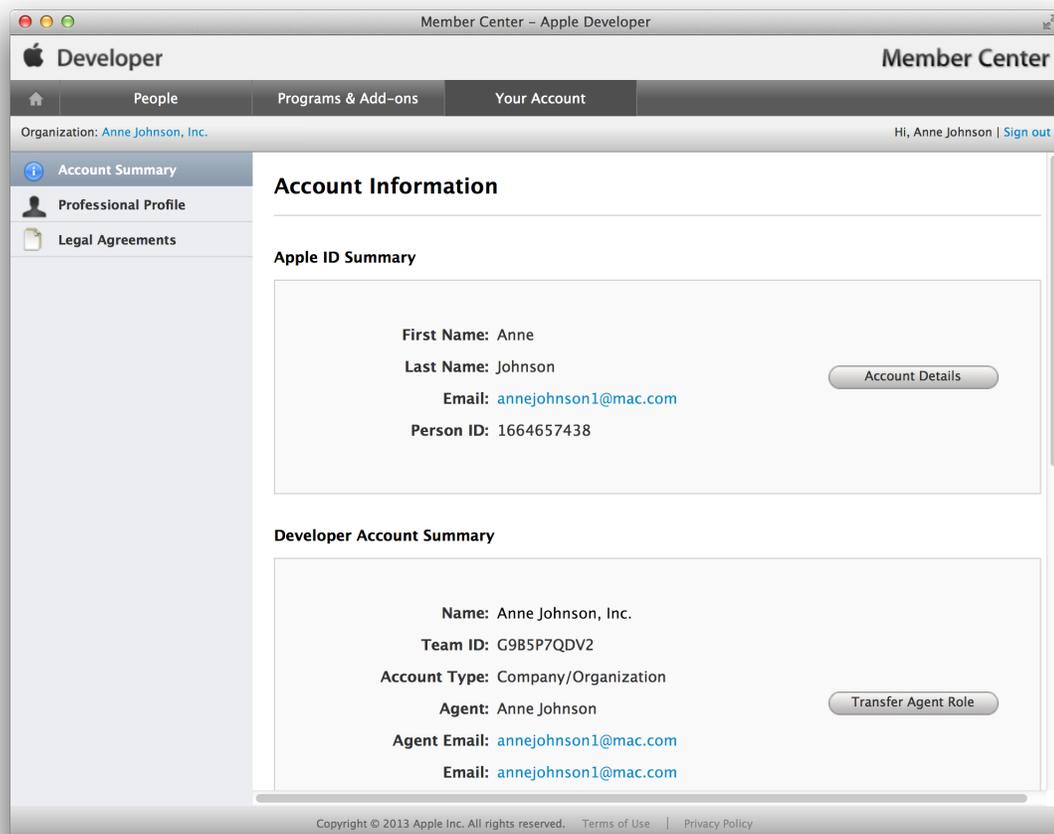
チーム管理者はメンバーセンターで、デバイスを個別に（[“個々のデバイスを登録する”](#)（205ページ）を参照）、あるいは一括して（[“デバイスを一括登録する”](#)（206ページ）を参照）登録することができます。

チームエージェントの役割を委譲する

チームエージェントは法的にチームを代表する人なので、他のチームメンバーが、降格させたり、権限を制限したりすることはできません。一方、チームエージェントは、メンバーセンター上で他のチームメンバーに自分の役割を委譲することができます。

チームエージェントの役割を委譲するには

1. [メンバーセンター](#)にチームエージェントとしてサインインし、ウェブページの上にある「Your Account」をクリックしてください。



2. 「Developer Account Summary」セクションで、自分の名前と同じ行にある「Transfer Agent Role」ボタンを押します。
3. するといくつかダイアログが現れるので、指示に従って操作してください。
新しいチームエージェントを選ぶこと、エージェント委譲の取り決めに署名することなどを求められます。

チームメンバーを外す

チームエージェントやチーム管理者は、プロジェクトにおける作業がなくなったチームメンバーを、チームから外すことができます。

チームメンバーを外すには

1. [メンバーセンター](#)のウェブページ上部にある「People」をクリックします。
2. サイドバーの「Developers」を押します。
3. チームから外すチームメンバーの行にあるチェックボックスをオンにします。
4. 左上隅にある「Remove Access」を押してください。



5. ダイアログが表示されるので、「OK」を押します。

まとめ

この章では、権限がないチームメンバーに代わって、開発用証明書を作成し、デバイスを登録する手順を説明しました。

署名IDや証明書の管理

アプリケーションにコード署名を施すことにより、その開発者をAppleが承認していること、何者かに改竄されていないことを、ユーザが信頼できるようになります。アプリケーションはすべて、コード署名およびプロビジョニングを施さなければ、デバイス上で起動し、特定のアプリケーションサービスを利用し、テスト用に配布し、iTunes Connectにアップロードすることができません。コード署名とは、暗号技術を活用して、アプリケーションやインストーラパッケージにデジタル署名を施すことです。アプリケーションに署名およびプロビジョニングを施すためには、署名IDを作成してキーチェーンに保存するとともに、証明書を作成してメンバーセンターに保存する必要があります。これらのアセットには、開発者や開発チームを一意に識別する働きがあるので、安全に保持することが重要です。この章では、プロジェクトが継続する期間中、署名IDや証明書を安全に管理するために行う主な作業について説明します。

開発、テスト、配布に用いる証明書の種類については、[“署名証明書の詳細”](#)（192 ページ）を参照してください。

署名IDと証明書について

アプリケーションにコード署名を施しておけば、オペレーティングシステムは、誰が署名したかを認識し、署名時以降に改竄されていないことを検証できます。アプリケーションバンドル内の実行コードが改竄されれば署名は無効になるので、結果的に実行コードが保護されることになるのです。なお、画像やnibファイルは署名の対象でないことに注意してください。これを変更しても署名は無効になりません。

コード署名は、AppID、プロビジョニングプロファイル、エンタイトルメントと組み合わせて、次の事項を確保するために使います。

- 自分自身または信頼できるチームメンバーが、アプリケーションをビルドし、署名したこと
- 自分自身またはチームが署名したアプリケーションが、所定の開発用デバイス上でしか動作しないこと
- アプリケーションが、所定のテスト用デバイス上でしか動作しないこと
- 自分が関知していないアプリケーションサービスをアプリケーションが利用していないこと
- アプリケーションの各ビルドをiTunes Connectにアップロードできるのは自分自身だけであること
- ストアを通さずに配布する場合（Macのみ）、自分以外がアプリケーションを修正、配布できないこと

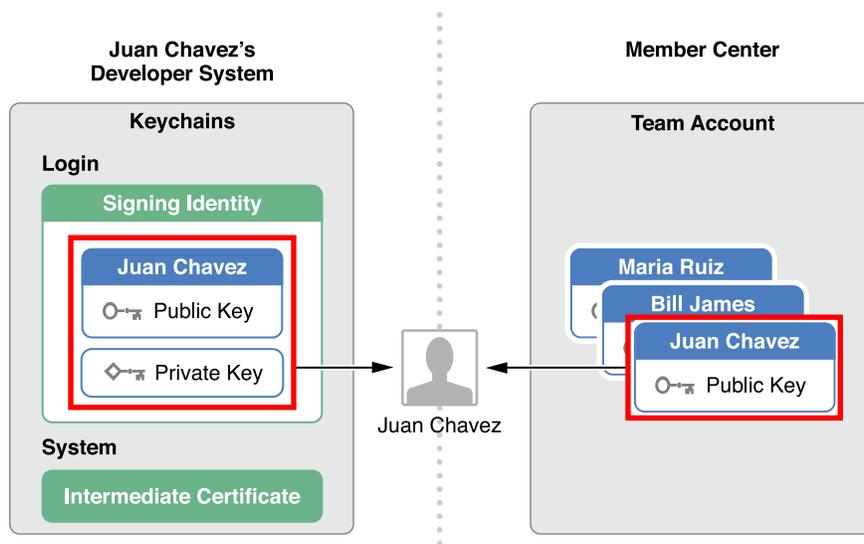
コード署名は、信頼できる者が除去したうえで、再署名することも可能です。たとえば、アプリケーションに署名してiTunes Connectにアップロードした後、Appleが署名し直して顧客に配布する、ということがあるのです。あるいは、完全にテスト済みの開発用ビルドに署名し直した上で、ストアに登録することもできます。

Xcodeはビルド処理の際、署名IDを使って署名を施します。この**署名ID**の実体は公開鍵/秘密鍵の組で、Appleが発行します。公開鍵/秘密鍵の組はキーチェーンに保存され、暗号プログラムで署名を生成するのに使われます。開発者アカウントに保存される証明書には、公開鍵しか含まれていません。**中間証明書**もキーチェーンに登録して、上記の証明書の発行元が**認証局**である旨を証明するために使います。

署名を施すためには、署名IDと中間証明書の両方をキーチェーンに登録しておく必要があります。Appleの中間証明書は、Xcodeをインストールする際、自動的にキーチェーンに追加されるので、署名IDのみ別途作成すればよいこととなります。署名IDがキーチェーンに追加され、対応する証明書がメンバーセンターのアカウントに追加されます。

署名IDは、アプリケーションやインストーラパッケージに署名するために使います。**開発用証明書**には、ある開発者をチームメンバーとして識別する働きがあり、開発用プロビジョニングプロファイルに収容して使います。これがあれば、アプリケーションに署名し、デバイス上で起動できるようになります。一方、**配布用証明書**には、開発チーム/組織を識別する働きがあり、配布用プロビジョニングプロファイルに収容して使います。これがあれば、アプリケーションをストアに登録できます。配布用証明書を作成できるのは、チームエージェントまたはチーム管理者に限ります。iOS、tvOS、watchOSの各アプリケーションに、同じ開発用/配布用証明書を使っても構いません。ただしMacアプリケーション用には別に用意します。各種の証明書については、[“署名証明書の詳細”](#)（192 ページ）を参照してください。

組織の場合、他のチームメンバーも固有の署名IDをMacコンピュータに組み込んでいます。メンバーセンターには各種のチームアセットを保存するリポジトリがありますが、秘密鍵は保存していません。



秘密鍵は個々のMacにローカルに保存してあるので、アカウントのパスワードと同様、紛失や盗難がないよう保全しなければなりません。公開鍵と秘密鍵の組を、安全な場所にバックアップしておいてください。秘密鍵を紛失すると、新たに生成し直さない限り、コードに署名することはできなくなります。さらに、秘密鍵を盗まれると、成りすましの恐れがあります。不正なコードを仕込んだアプリケーションを配布しようとするかも知れないのです。その結果、アプリケーションの登録が却下されるだけでなく、Appleが開発者証明書を失効させることもありえます。秘密鍵はキーチェーンのみに格納し、紛失しても他人に盗用されないようにしてください。

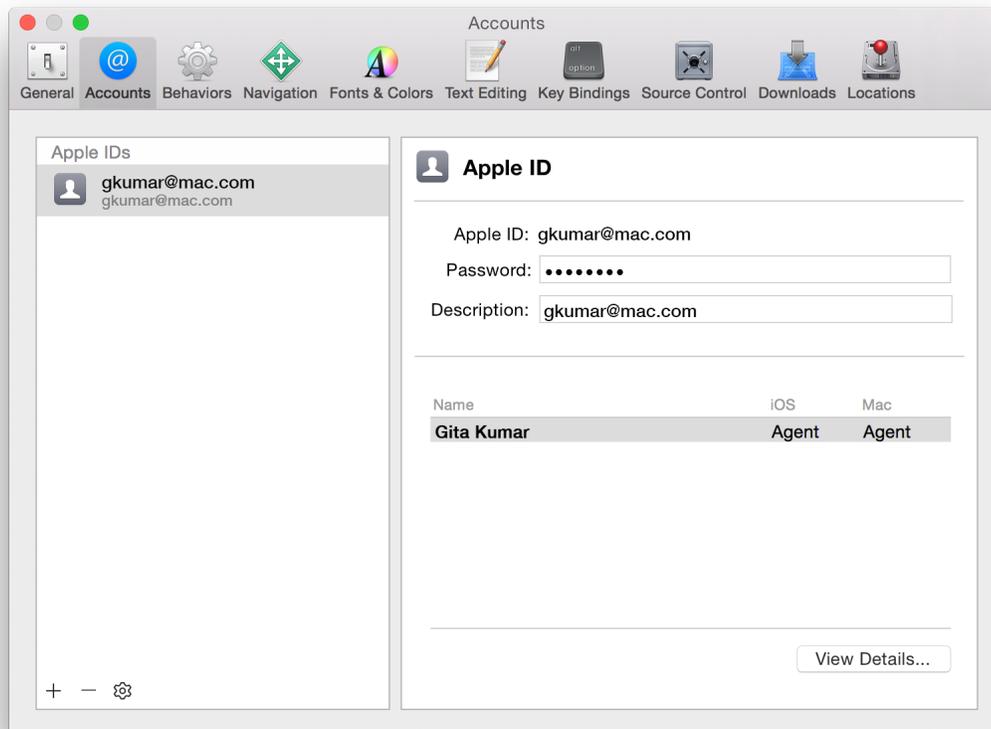
他のMac上でコード署名をする場合、証明書作成に用いたMac上の開発者プロファイルをエクスポートし、コード署名をするMac側にインポートしなければなりません。この機能を利用して、配布用証明書を複数のチームエージェント間で共有することも可能です。（チームメンバーは開発用証明書を共有できません。）

署名IDやプロビジョニングプロファイルの内容を表示する

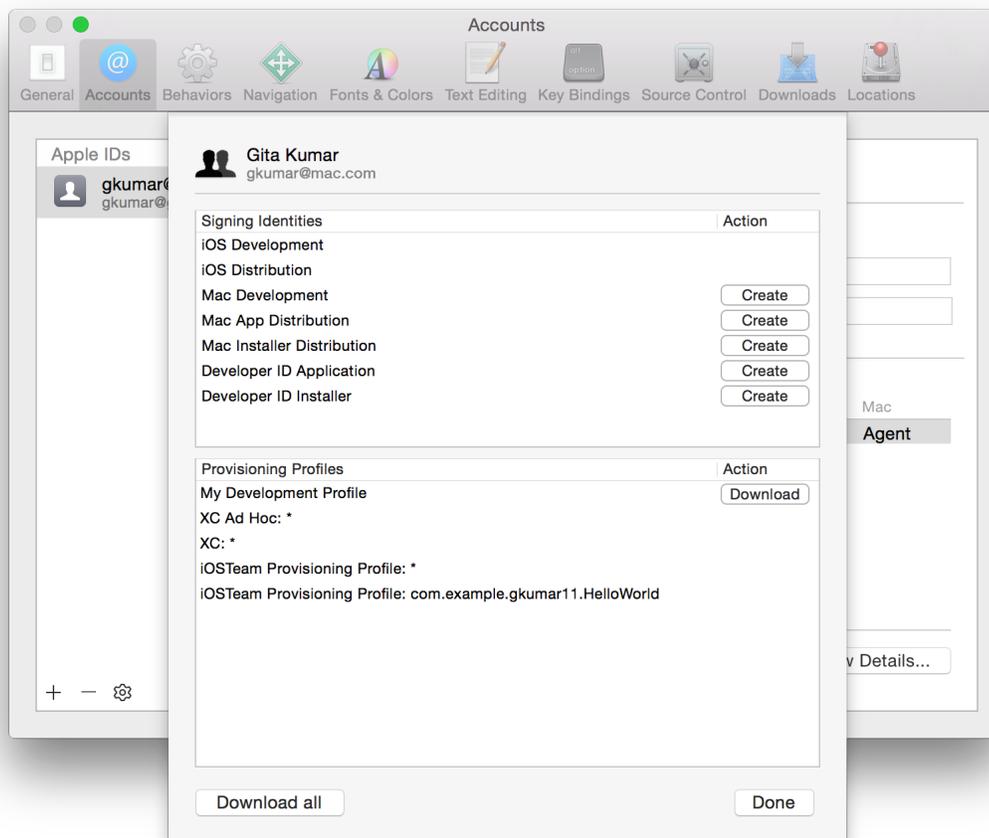
証明書やプロファイルは、その内容を検証し、あるいは障害に対処するため、Xcodeの「Accounts」環境設定画面に表示することができます。通常の管理はXcodeに任せておいて構いませんが、場合によっては、ある特定の証明書を作成し、失効させ、あるいはプロビジョニングプロファイルをダウンロードする必要が生じます。

アカウント詳細を表示するには

1. 「Xcode」 > 「Preferences」 を実行して環境設定を起動します。
2. ウィンドウ上部の「Accounts」 を押して画面を切り替えます。
3. 該当するチームを選択し、「View Details」 を押してください。



開いたダイアログで、署名IDとプロビジョニングプロファイルを表示します。証明書の隣に「Create」ボタンが現れている場合、まだ生成されていません。プロビジョニングプロファイルの隣に「Download」ボタンが現れている場合、これは自分自身のMacではありません。



4. 「Done」をクリックします。

署名IDを作成する

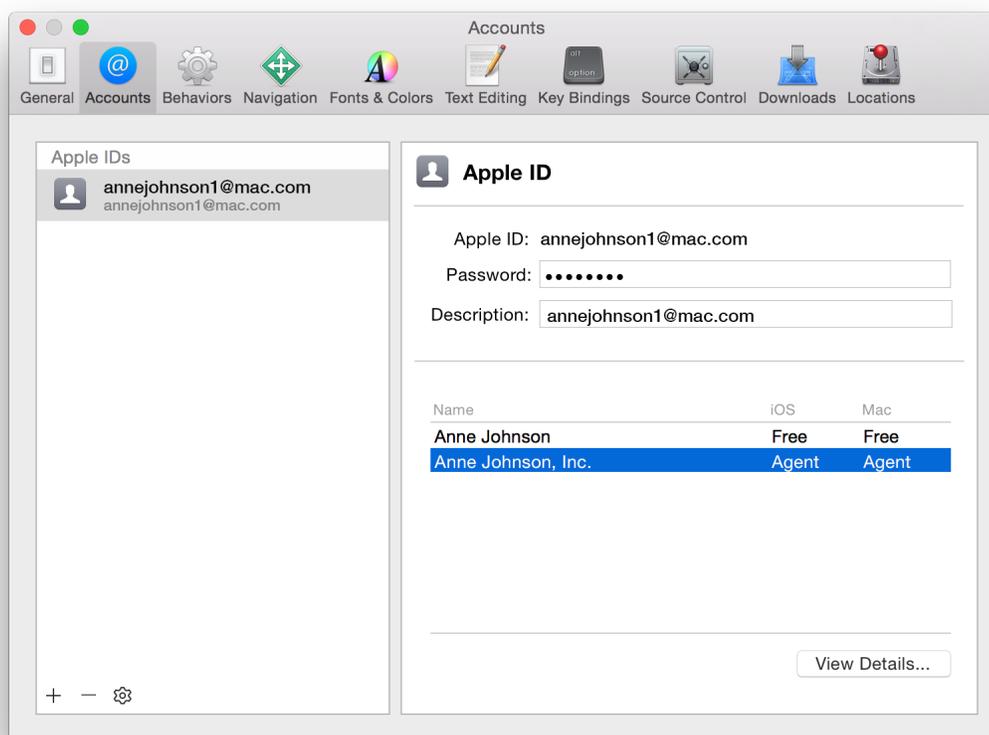
アプリケーションにコード署名を施すためには、開発用証明書を作成しなければなりません。また、アプリケーションをiTunes Connectにアップロードする際には、配布用証明書が必要になります。必要な証明書や署名IDは、いずれもXcode上で作成できます。署名IDを作成し、ダウンロードし、インストールする処理は、すべてXcodeが自動的に行います。

チームメンバーは誰でも、自身の開発用証明書を作成できます。配布用証明書を作成できるのは、チームエージェントおよびチーム管理者に限ります。Developer ID証明書を作成できるのはチームエージェントだけです。組織として開発する場合、チームの役割や、チームエージェントがメンバーに代わって行う作業については、「[メンバーセンターでのチームの管理](#)」（160 ページ）を参照してください。

Xcodeは必要に応じ、開発用証明書を作成します。たとえば、プロジェクトをチームに対応づけ、あるいはチームプロビジョニングプロファイルを作成しようとする（[“IDおよびチームの設定をす](#)
[る”](#)（33 ページ）を参照）、開発用証明書を作成し、チームプロビジョニングプロファイルに追加するのです。したがって通常、配布用証明書の作成は、Xcodeの「Preferences」ウインドウで行います。

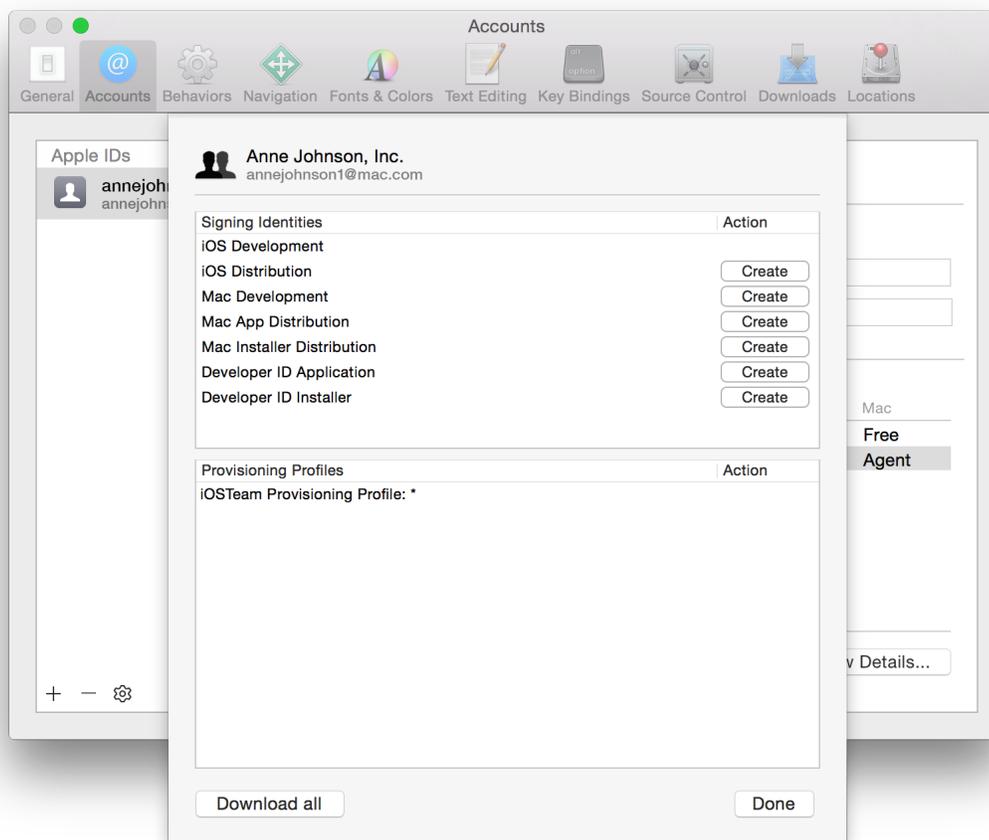
署名IDを作成するには

1. Xcodeの「Preferences」ウインドウで、「Accounts」を押して表示を切り替えます。
2. 該当するチームを選択し、「View Details」を押してください。



3. ダイアログが現れるので、作成する証明書の種類に応じた行の「Create」ボタンを押してください。

押せない状態になっている場合、該当する種類の署名IDは作成できません。証明書の種類については、表 14-2（193 ページ）を参照してください。



署名IDを作成すると「Create」ボタンは消えます。

4. 「Done」を押すと「Accounts」環境設定画面に戻ります。

作成した署名IDは、エクスポートしてバックアップ保存することができます（“[開発者プロファイル](#)をエクスポートする”（181 ページ）を参照）。

ここまでの結果を確認する

証明書が適切で、すぐに使える状態であることを確認しましょうアプリケーションに（Macの場合はインストーラパッケージにも）署名するためには、証明書が有効でなければなりません。

初めて確認する際は、Xcode、Keychain Access、メンバーセンターのそれぞれについて、証明書がどこにあり、ツールにどのように表示されるか調べてください。Keychain Accessおよびメンバーセンターには、署名IDや証明書の有効期限が表示されます。後で障害が発生した場合は、Keychain Accessを使って対処することになります。

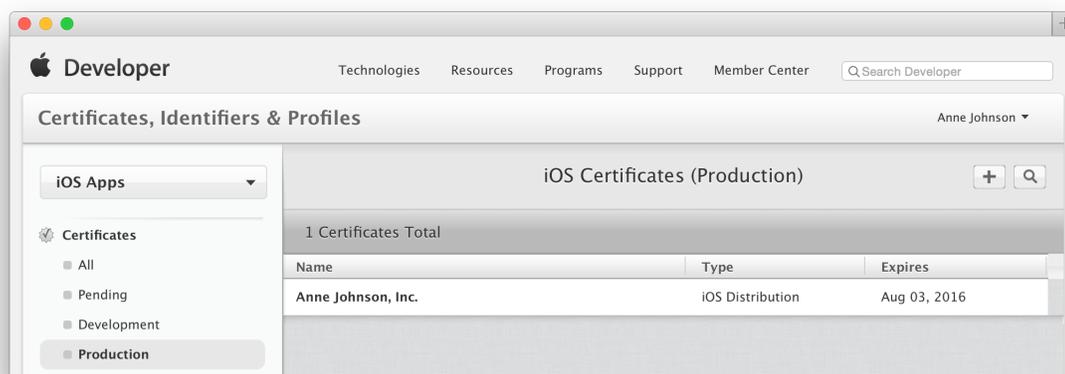
メンバーセンター上で確認する

メンバーセンターには公開鍵が保存されているので、XcodeやKeychain Accessに表示されたのと同じ証明書が現れるはずですが。

メンバーセンターで署名証明書を確認するには

1. 「Certificates, Identifiers & Profiles」の画面で「Certificates」を選択します。
2. 「Certificates」セクションで、確認する証明書の種類（「Development」または「Production」）を選択してください。

証明書の種類や有効期限が、Keychain Accessに表示されるものと一致していなければなりません。



注意: キーチェーン中の証明書の名前は、Xcodeやメンバーセンターに現れる、証明書の種類を表す名前とは違っていています。両者の対応関係については表 14-2（193 ページ）を参照してください。

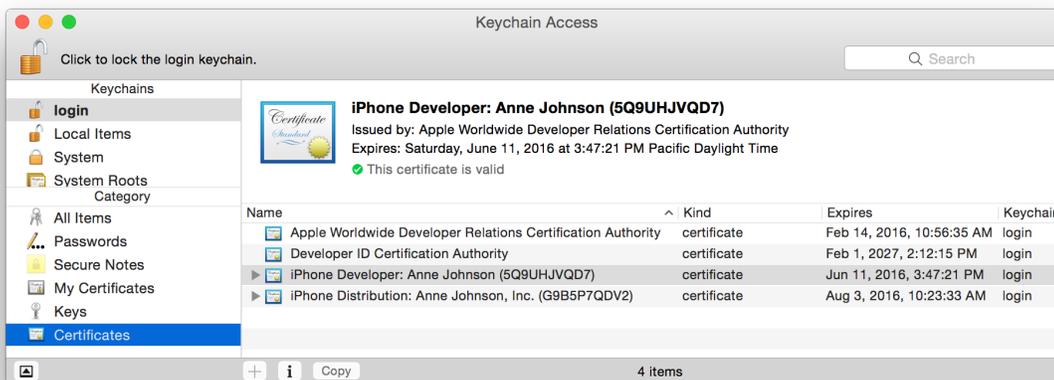
Keychain Accessを使って確認する

Keychain Accessには、署名IDの秘密鍵/公開鍵が表示されます。

Keychain Accessで署名IDの内容を確認するには

1. Keychain Access（/Applications/Utilities以下）を起動してください。
Xcodeで開発用/配布用証明書を作成した場合、証明書は自動的に、ログインキーチェーンに登録されます。
2. 左側ペインで、「Keychains」セクションの「login」、および「Category」セクションの「Certificates」を選択してください。

開発用/配布用証明書が、Keychain Accessの「Certificates」カテゴリに表示されます。その名前は、iOS/tvOS/watchOSアプリケーションならば先頭が「iPhone Developer」、Macアプリケーションならば先頭が「Mac Developer」で、その後に開発者の名前が続く形です（開発用証明書は開発者ごとに作成されることに注意）。

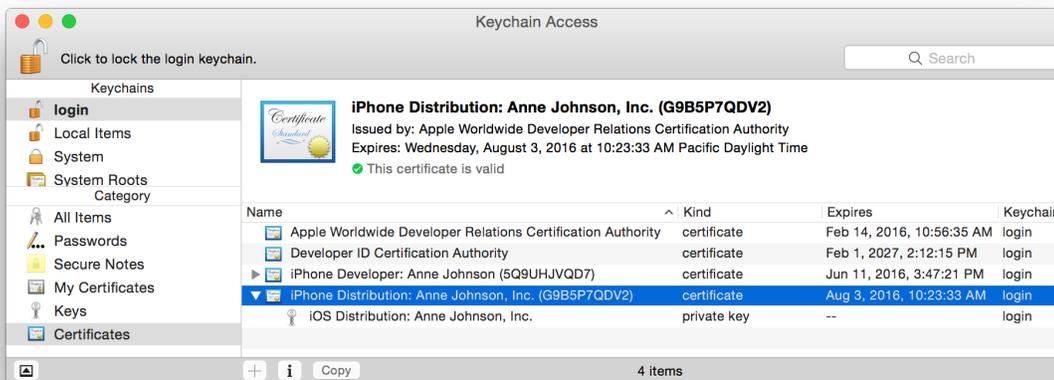


なお、ここには開発用以外の証明書も現れます。

Important: 個人に属する証明書その他は、ログインキーチェーンに登録してください。ログインキーチェーンに証明書がない場合、原因として、これがデフォルトのキーチェーンでないことが考えられます。デフォルトのキーチェーンは、Keychain Accessの「Keychains」列に太字で表示されます。ログインキーチェーンがデフォルトでない場合は、「Keychains」列で「login」を選択し、「File」>「Make Keychain "login" Default」を実行してください。

3. 証明書の左側に、詳細表示用の三角印があるはずですが。

これをクリックすると秘密鍵が表示されます。一方、この三角印が表示されない場合、秘密鍵がないこととなります（対処方法については[「コード署名IDが見つからない」](#)（251ページ）を参照）。



4. 証明書が有効であることを確認します。

Keychain Access上で証明書を選択すると、証明書リストの上側に、緑の円で囲んだチェックマークと「This certificate is valid.」という文言が現れるはずですが、

トラブルシューティング

XcodeやKeychain Accessに表示された証明書がメンバーセンターのそれと一致していない場合は、「[証明書に関する問題](#)」（251 ページ）に従って対処してください。

追加のDeveloper ID証明書を作成する

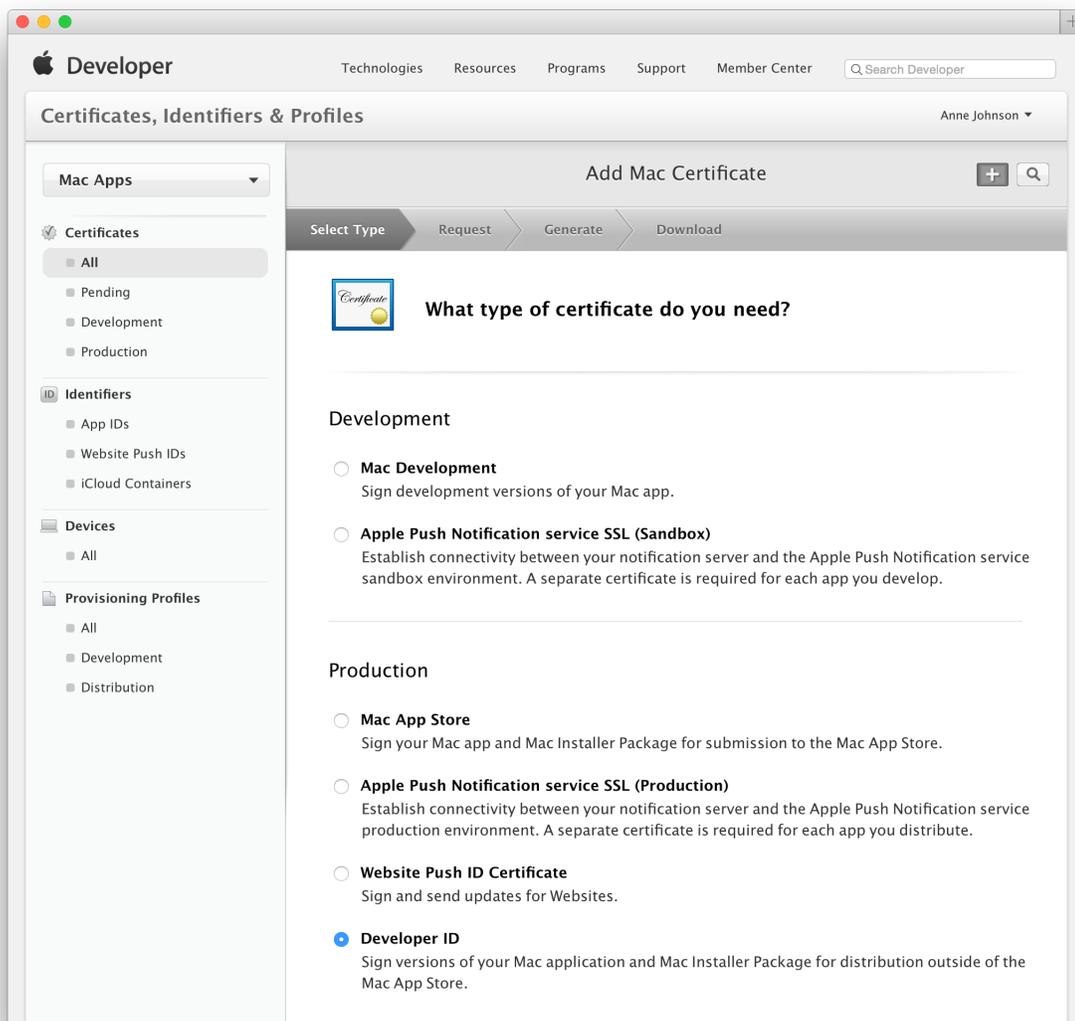
Developer ID証明書は、Mac App Storeを通さずにアプリケーションを配布する際に必要です。他の種類の証明書と同様、Xcode上で作成できます（「[署名IDを作成する](#)」（172 ページ）を参照）。複数必要であれば、メンバーセンターで種類ごとに5件まで作成可能です。

Important: チームエージェントだけが追加のDeveloper ID証明書を作成できます。

Developer ID証明書を作成するには

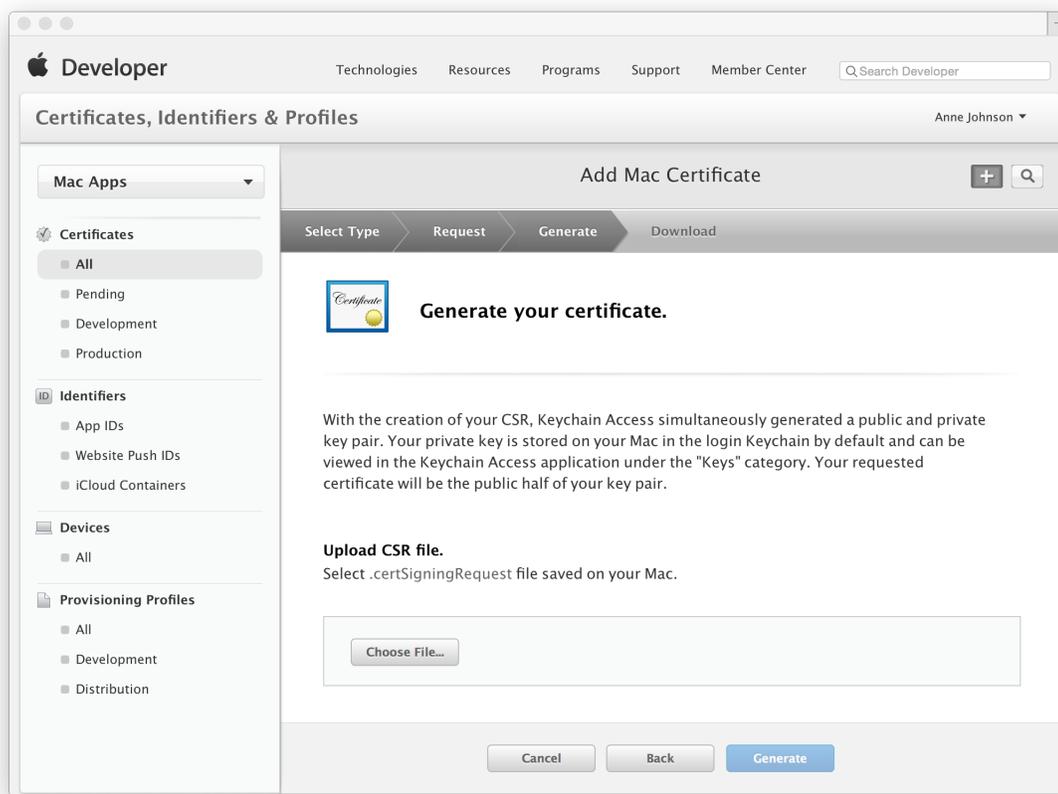
1. 「[Certificates, Identifiers & Profiles](#)」の「Mac Apps」セクションで、「Certificates」を選択してください。
2. 「Certificates」以下の「All」を選択します。
3. 右上隅にある「Add」ボタン（「+」の表示）を押します。

4. 「Production」以下の「Developer ID」を選択し、「Continue」を押してください。



5. 証明書の種類として「Developer ID Application」または「Developer ID Installer」を選択し、「Continue」を押します。
6. 表示される手順に従い、Keychain Accessを使って証明書署名要求（CSR、Certificate Signing Request）を作成し、「Continue」を押します。

7. 「Choose File」 ボタンを押します。



8. CSRファイル（拡張子「.certSigningRequest」）を選択し、「Choose」を押します。
9. 「Generate」を押します。
10. 「Download」を押してください。

証明書ファイルがDownloadsフォルダにダウンロードされます。

Developer ID証明書は、そのファイル（拡張子「.cer」）をダブルクリックするだけでキーチェーンに登録できます。Keychain Accessの「My Certificates」カテゴリに、この証明書が現れます。

誤って削除した中間証明書を登録する

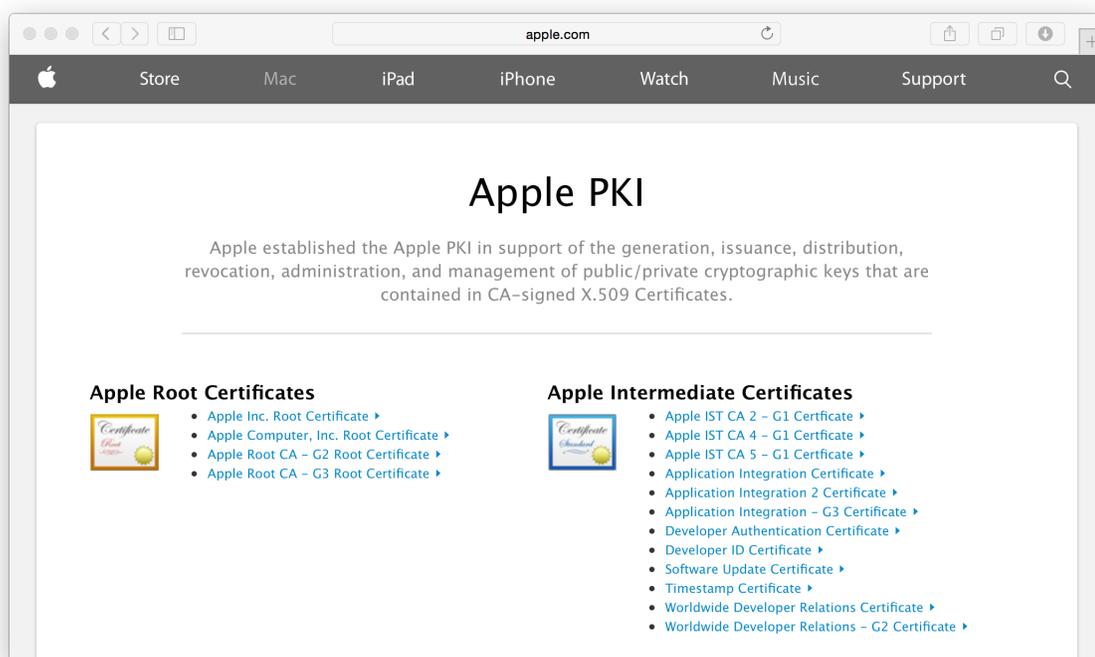
証明書を使用するためには、正しい中間証明書がキーチェーンに登録されていなければなりません。中間証明書には、信頼できる機関が証明書を発行した旨を保証する働きがあります。これには「Apple Worldwide Developer Relations Certification Authority」という名前がついており、Xcodeのインストール時に、システムキーチェーンに登録されます。Developer ID証明書に用いる中間証明書は、「Developer ID Certification Authority」という名前です。誤って削除してしまった場合、登録し直さなければなりません。

まず、Xcode上でプロビジョニングプロファイルをダウンロードして（“[Xcode上でプロビジョニングプロファイルをダウンロードする](#)”（215ページ）を参照）、欠落した中間証明書がインストールされないか試してください。うまくいかなければ、改めてダウンロードし、インストールします。

誤って削除した中間証明書を登録するには

1. <http://www.apple.com/certificateauthority>にアクセスします。
2. 「Apple Intermediate Certificates」以下にある、該当する中間証明書のリンクをクリックしてください。

これは「.cer」という拡張子のファイルで、Downloadsフォルダにダウンロードされます。



3. 証明書ファイルをダブルクリックすると、システムキーチェーンに登録されます。

証明書やプロファイルをエクスポート/インポートする

Xcodeが生成した証明書やプロファイルをエクスポートして、アセットすべてのバックアップを作成できます。たとえば、開発用に使っている別のMacにアセットを転送しておけば、秘密鍵を紛失しても証明書を復元できることとなります。Xcode上でプロファイルをダウンロードしても、欠落した秘密鍵が復元されることはないので、バックアップからインポートしなければなりません。

エクスポートしたファイル（**開発者プロファイル**）には、次のようなチームアセットが収容されています。

- 開発用証明書
- 配布用証明書
- プロビジョニングプロファイル

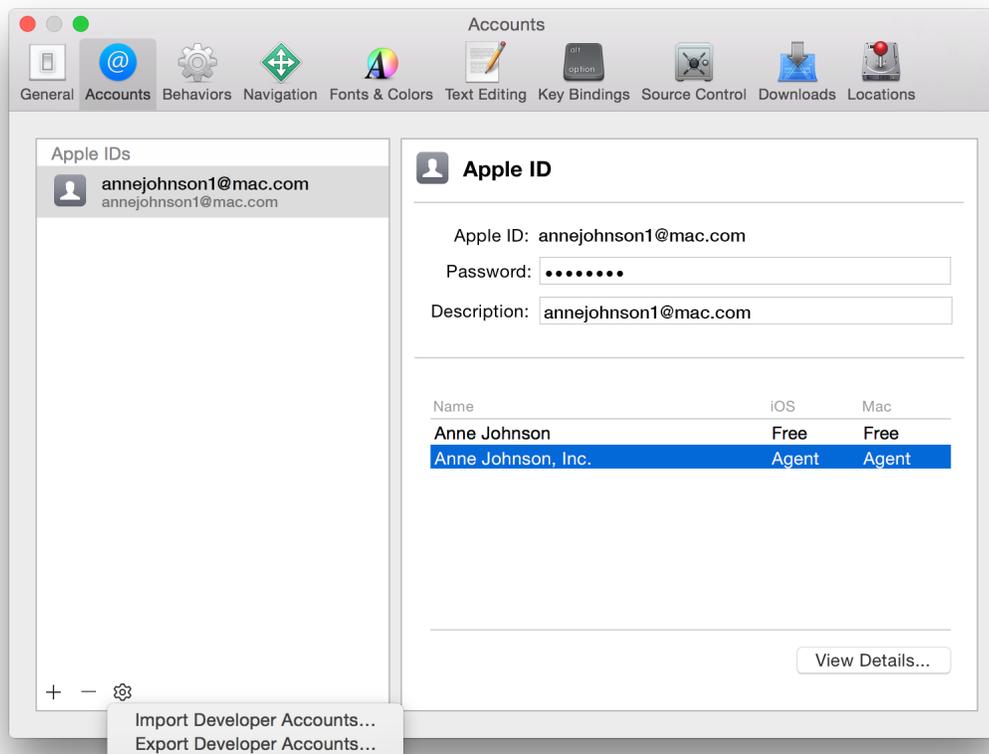
特定の証明書をエクスポートし、他のチームメンバーと共有することも可能です。この場合、エクスポートファイルには、明示的に選択した証明書のみ収容されることになります。

開発者プロファイルをエクスポートする

開発者プロファイルは、アプリケーションに署名してストアに登録する際の身許証明になるので、エクスポートの際には、暗号化を施し、パスワードで保護するようになっています。

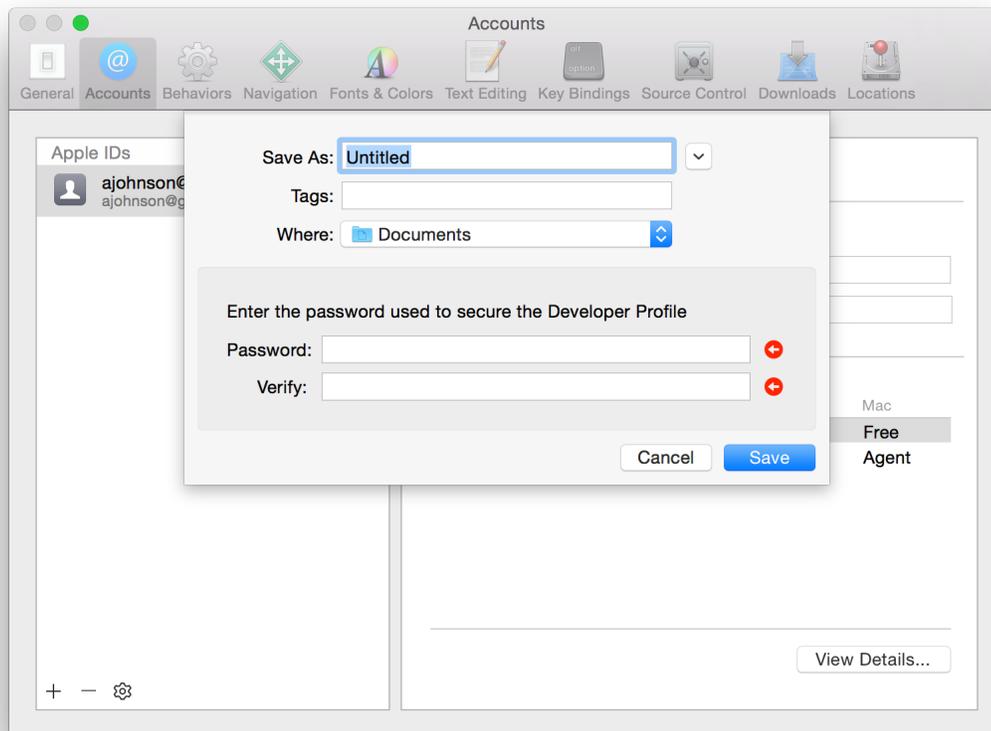
開発者アカウントのアセットをエクスポートするには

1. 「Xcode」 > 「Preferences」 を実行して環境設定を起動します。
2. ウィンドウ上部の「Accounts」 を押して画面を切り替えます。
3. 左下隅にある「Action」 ボタン（「Delete」 ボタンの右にある歯車アイコン） をクリックして、ポップアップメニューから「Export Developer Accounts」 を選択します。



4. 「Save As」 フィールドにファイル名を入力し、「Password」 フィールドと 「Verify」 フィールドの両方にパスワードを入力します。

ファイルは、暗号化を施し、パスワードで保護するようになっています。



5. 「Save」を押してください。

指定した場所に、「.developerprofile」という拡張子で保存されます。

6. ダイアログが表示されるので、「OK」を押します。

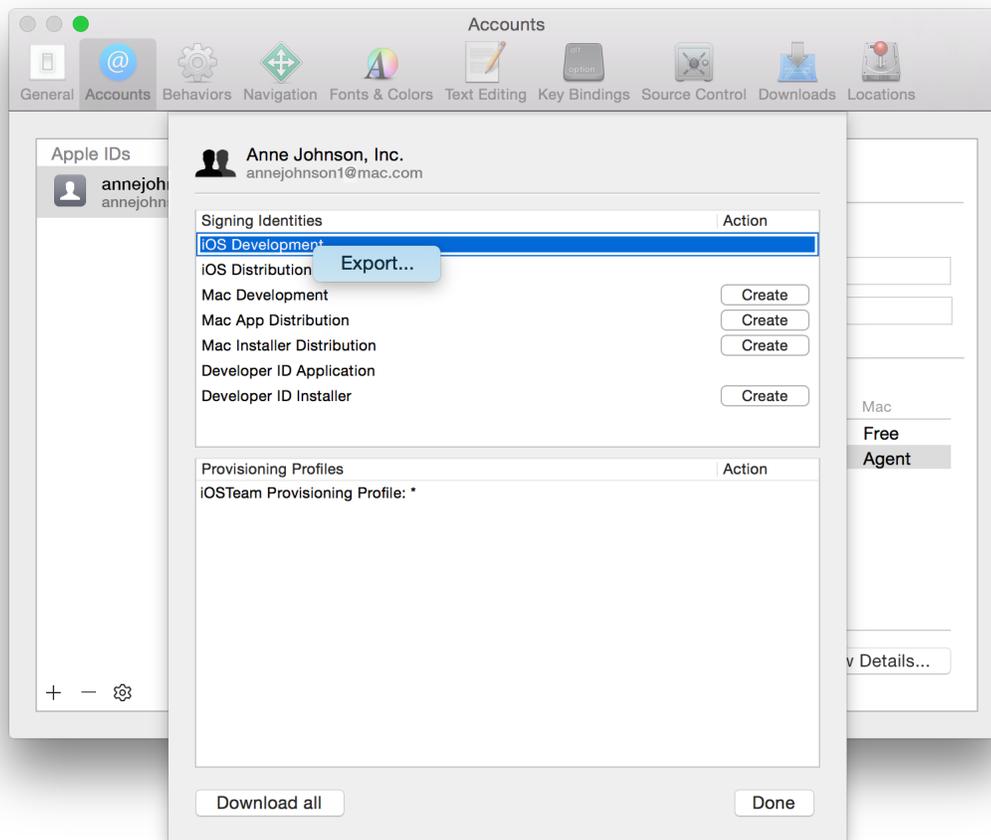
特定の証明書のみをエクスポートする

特定の証明書のみをエクスポート（プロファイルは除外）する場合は、詳細ダイアログで該当する証明書を選択します。

特定の証明書のみをエクスポートするには

1. 「Xcode」 > 「Preferences」を実行して環境設定を起動します。
2. ウィンドウ上部の「Accounts」を押して画面を切り替えます。
3. 該当するチームを選択し、「View Details」を押してください。

- 「Signing Identities」テーブルで、Controlキーを押しながらエクスポートしようとする証明書をクリックし、ポップアップメニューの「Export」を実行します。



- 「Save As」フィールドにファイル名を入力し、「Password」フィールドと「Verify」フィールドの両方にパスワードを入力します。
ファイルは、暗号化を施し、パスワードで保護するようになっています。
- 「Save」を押してください。
指定した場所に、「.p12」という拡張子で保存されます。

あるいは、コマンドラインユーティリティ「security」で証明書をエクスポートしても構いません。

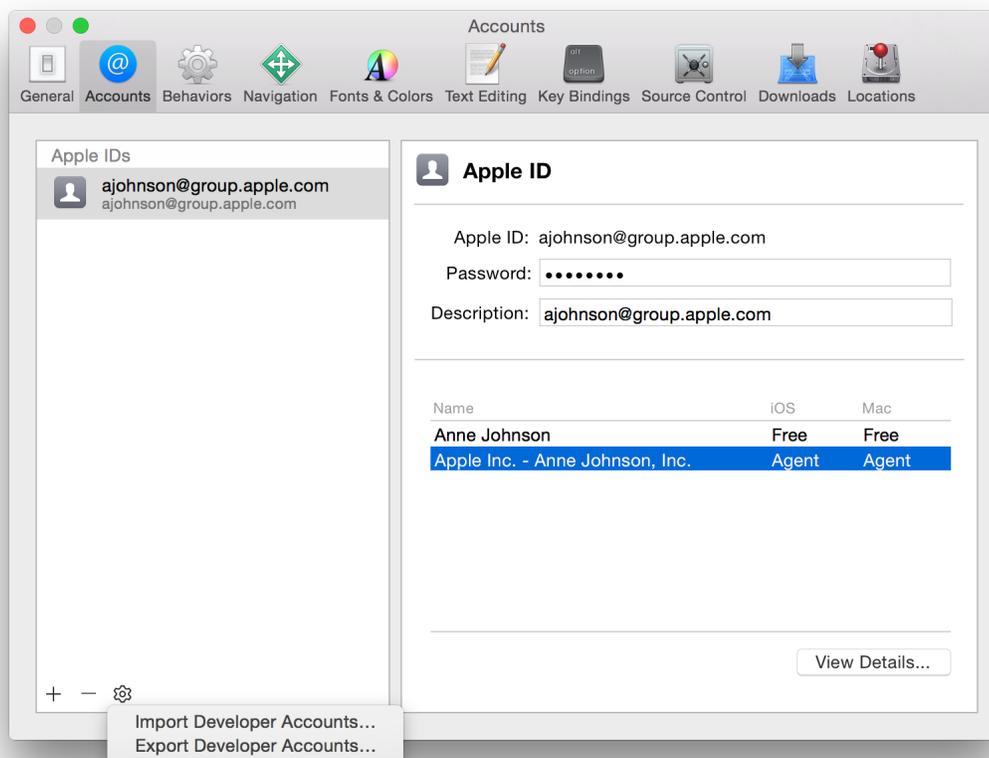
開発者プロファイルをインポートする

秘密鍵を紛失したとき、あるいは他のMacに移行する際には、開発者プロファイルをインポートする必要があります。

開発者アカウントのアセットをインポートするには

- 「Xcode」 > 「Preferences」を実行して環境設定を起動します。
- ウィンドウ上部の「Accounts」を押して画面を切り替えます。

3. 左下隅にある「Action」ボタン（歯車アイコン）をクリックして、ポップアップメニューから「Import Developer Accounts」を選択します。



4. 開発者プロファイルが収容されているファイルを選択し、「Open」を押してください。拡張子が「.developerprofile」のファイルです。
5. 暗号化の際に用いたパスワードを入力し、「OK」を押してください。
6. ダイアログが表示されるので、「OK」を押します。

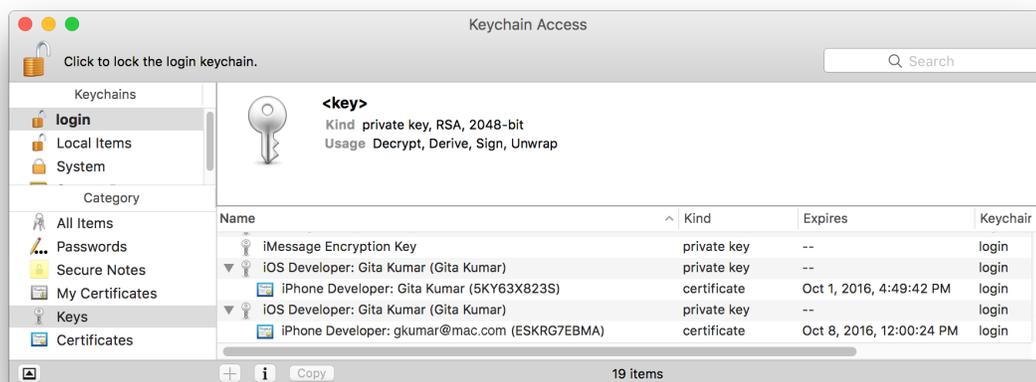
署名IDをキーチェーンから削除する

署名IDが無効になった、使わなくなった（チームを移ったためなど）、秘密鍵を紛失したため使えなくなった、などの場合、キーチェーンから署名IDを削除しなければなりません。秘密鍵を紛失した場合でも、署名IDのバックアップが残っていれば、キーチェーンからいったん署名IDを削除し、すぐに開発者プロファイルをインポートすることにより対処できます（[“開発者プロファイルをインポートする”](#)（183 ページ）を参照）。それ以外の理由で署名IDを削除する際には、後でコード署名に関する問題が起こらないよう、[“証明書を再生成し、関係するプロビジョニングプロファイルを更新する”](#)（189 ページ）に示した手順をよく読んでから進めてください。

Warning: 秘密鍵をキーチェーンから削除すると、開発者プロファイルからインポートしない限り、再生成はできません。秘密鍵がなければ、この証明書を使って署名することはできません。

署名IDをキーチェーンから削除するには

1. Keychain Access (~/Applications/Utilities) を起動してください。
2. 「Category」セクションの「Keys」を選択します。
3. 秘密鍵に添えられた展開表示用の三角印をすべてクリックして、対応する証明書を表示します。



4. 削除しようとする証明書に対応する秘密鍵をすべて選択します。
証明書の種類を、「Keychain Access」に表示される名前から判断する方法については、[表 14-2](#)（193 ページ）を参照してください。
5. 秘密鍵それぞれに対応する公開鍵を選択します。
6. Deleteキーを押し、次に現れるダイアログの「Delete」ボタンを押してください。
7. 「Category」セクションの「Certificates」を選択します。
8. 削除する証明書をすべて選択してください。
この証明書には、対応する秘密鍵がありません。
9. Deleteキーを押し、次に現れるダイアログの「Delete」ボタンを押してください。

証明書を失効させる

証明書が不要になったときや、コード署名に関する問題のため再生成しようとする場合（発生しうる問題については[「証明書に関する問題」](#)（251 ページ）を参照）、その証明書を失効させる必要があります。盗用などの疑いがある場合も失効させなければなりません。さらに、組織のチーム管理者は、

プロジェクトから抜けたチームメンバーの開発用証明書を失効させることになります。証明書を失効させると、プロビジョニングプロファイルが無効になることがあるので、後でコード署名に関する問題が起らないよう、“[証明書を再生成し、関係するプロビジョニングプロファイルを更新する](#)”（189 ページ）に示した手順をよく読んでから進めてください。

Important: 開発用/配布用証明書を失効させても、ストアに登録済みのアプリケーションには影響がなく、その更新も引き続き可能です。

権限を失効させる

表 14-1 に、チームの役割ごとに、失効させることができる証明書の種類を示します。個人開発者は、1人だけのチームのエージェントでもあるので、あらゆる開発用/配布用証明書を失効させる権限があります（表に示すものを除く）。組織の場合、チームメンバーは自分自身の開発用証明書を失効させることができますが、配布用証明書を失効させるのは、チームエージェントまたは管理者に限ります。

表 14-1 チーム証明書を失効させる権限

証明書の種類	チームエージェント	チーム管理者	チームメンバー
自分自身の開発用証明書： iOS Development Mac Development	✓	✓	✓
他のチーム管理者やメンバーの証明書： iOS Development Mac Development	✓	✓	✗
チームエージェントの証明書： iOS Development Mac Development	✓	✗	✗
ストア配布用証明書： iOS Distribution Mac App Distribution Mac Installer Distribution	✓	✓	✗

証明書の種類	チームエージェント	チーム管理者	チームメンバー
Developer ID証明書： Developer ID Application Developer ID Installer	✗	✗	✗
プッシュ通知証明書： APNs Development iOS APNs Production iOS APNs Development Mac APNs Production Mac	✓	✓	✗
パス証明書： パス型識別子	✗	✗	✗

Developer IDやPass Type ID証明書を、メンバーセンターで失効させることはできません。Apple (product-security@apple.com) 宛てに失効要求を送信してください。

AppleがDeveloper ID証明書を失効させると、その証明書で署名したアプリケーションは、誰もインストールできなくなります。失効させる代わりに、メンバーセンターで追加のDeveloper ID証明書を作成することも可能です（“追加のDeveloper ID証明書を作成する”（177 ページ）を参照）。

Xcodeで証明書をリセットする

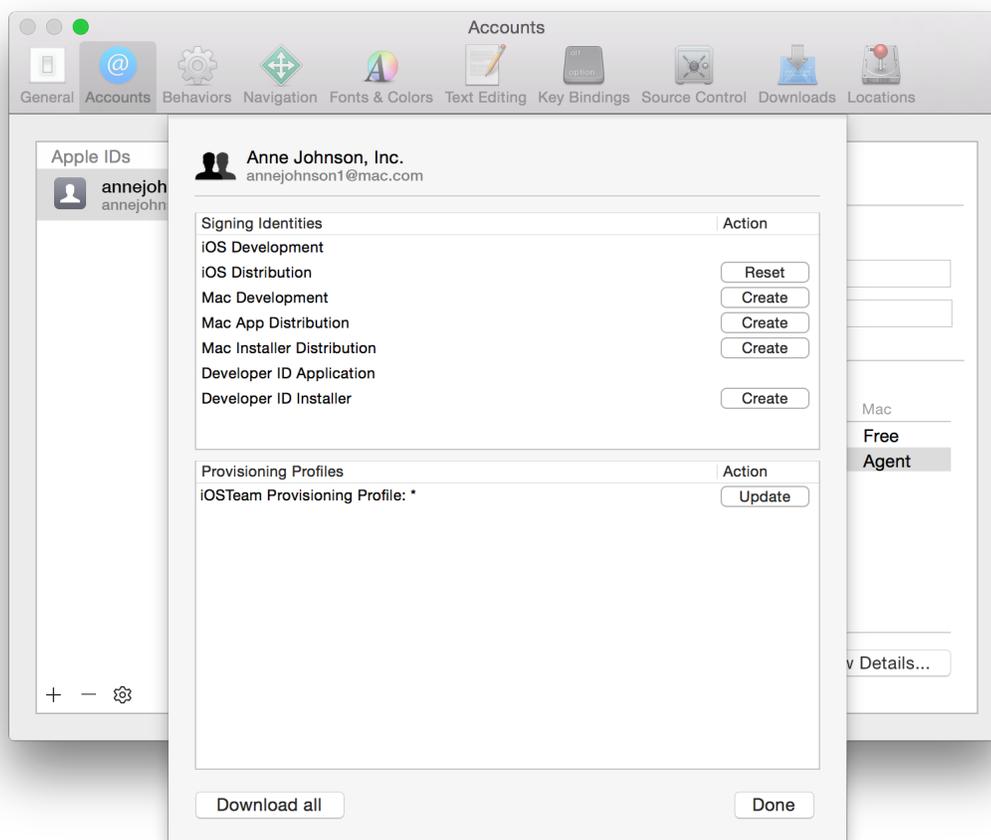
署名IDに問題があると、Xcodeの「Accounts」環境設定画面に、状況に応じたアクションボタンが現れます。「Create」ボタンが現れた場合、メンバーセンターにもMac上にも署名IDがありません。

「Reset」ボタンが現れた場合、秘密鍵がないなどの理由で、当該Mac上では署名IDが使用できません。このボタンを押すと、Xcodeは対応する証明書をいったん失効させ、改めて作成するようになっています。

メンバーセンターで証明書を失効させることも可能です（“メンバーセンターで証明書を失効させる”を参照）。チーム管理者であれば、メンバーセンター上で、チームメンバーの開発用証明書を失効させることも可能です。

Xcode上で証明書を失効させるには

1. 「Xcode」 > 「Preferences」の順に選択して環境設定を起動し、ウィンドウの上部にある「Accounts」をクリックして画面を切り替えます。
2. チームを選択して「View Details」を押します。
3. 表示されるダイアログで、失効させ改めて作成しようとする署名IDに対応する、「Action」列の「Reset」ボタンを押します。



4. するとダイアログが現れるので、「Done」を押してください。

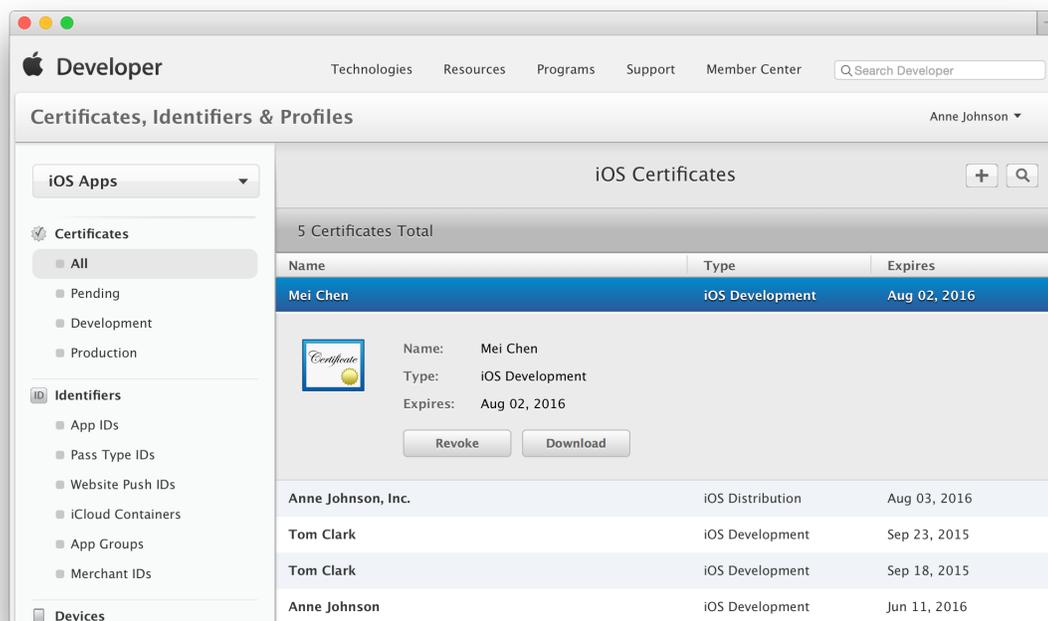
メンバーセンターで証明書を失効させる

メンバーセンターでは、チームに属するあらゆる証明書を失効させることができます。他のチームメンバーが使っている開発用証明書と、もう使っていない配布用証明書や再生成する予定がない配布用証明書も、失効させる対象になります。

メンバーセンターで証明書を失効させるには

1. 「Certificates, Identifiers & Profiles」の画面で「Certificates」を選択します。
2. 「Certificates」以下の「All」を選択します。

- 失効させようとする証明書を選択し、「Revoke」を押してください。



- するとダイアログが現れるので、「Revoke」を押します。

期限切れになった証明書を更新する

開発用/配布用証明書が期限切れになった場合、削除した上で、Xcodeで新しい証明書を作成する必要があります。証明書を再生成する手順は、「[証明書を再生成し、関係するプロビジョニングプロファイルを更新する](#)」（189ページ）に示す手順と同様です。Keychain Accessまたはメンバーセンターで、署名IDや証明書の有効期限を確認してください（「[メンバーセンター上で確認する](#)」（175ページ）、「[Keychain Accessを使って確認する](#)」（175ページ）を参照）。

証明書を再生成し、関係するプロビジョニングプロファイルを更新する

証明書を再生成し、関係するプロビジョニングプロファイルを更新する作業は、そう簡単ではありません。この2つは相互に関連し合っており、しかもMac上とメンバーセンターの両方に置かれているからです。ある証明書を失効させると、これを収容しているプロビジョニングプロファイルはすべて無効になります。チームプロビジョニングプロファイルはXcodeが必要に応じて再生成しますが、それ以外のプロビジョニングプロファイルは自分で管理しなければなりません。この節では、コード署名アセットを完全に復元するための手順を解説します。

証明書を再生成し、関係するプロビジョニングプロファイルを更新する状況は、たとえば次のようにいくつか考えられます。

- 誤ってキーチェーンから1つまたは複数の秘密鍵を削除してしまい、復元に必要なバックアップもないという状況。
- 失効または期限切れになった証明書とそれに関係するプロビジョニングプロファイルを削除したいという状況。
- チームを移るので、元のチームの証明書とプロビジョニングプロファイルをすべて削除したいという状況。
- 開発に複数台のMacコンピュータを使っているか、複数のチームに属していて、新規プロジェクト用にMacをセットアップしたいという状況。

もっとも、証明書やプロビジョニング、あるいはビルドに関する問題が発生した場合は、まず[“証明書に関する問題”](#)（251ページ）に示す事項を確認してください。証明書を削除してしまうと、元に戻すことはできません。

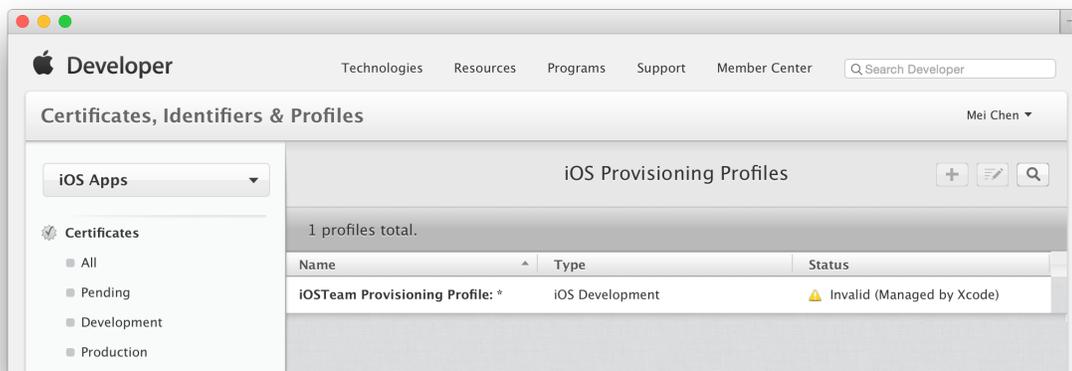
再生成する証明書を選択します。たとえば、デバイス上で実行しようとする問題が生じる場合、再生成が必要なのは開発用証明書だけです。配布用証明書を再生成しても、開発用証明書や開発用プロビジョニングプロファイルには影響が及びません。同様に、開発用証明書を再生成しても、配布用証明書や配布用プロビジョニングプロファイルには何の関係もないのです。

Important: 開発用/配布用証明書を再生成しても、ストアに登録済みのアプリケーションには影響がなく、その更新も引き続き可能です。

証明書を再生成し、関係するプロビジョニングプロファイルを更新するには

1. メンバーセンターで証明書を失効させます（[“証明書を失効させる”](#)（185ページ）を参照）。

証明書が失効すれば、これを収容しているプロビジョニングプロファイルも無効になります。たとえば、チームメンバーの開発用証明書を失効させると、これを収容する開発用プロビジョニングプロファイルもすべて無効になってしまうのです。

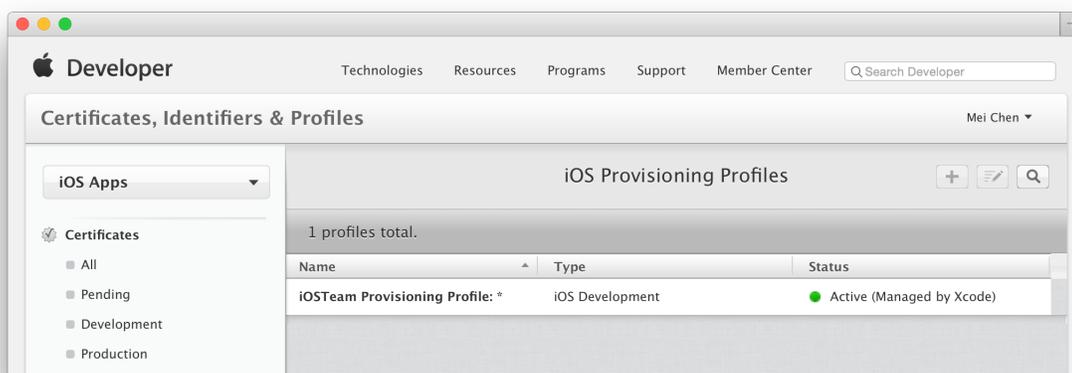


2. 必要ならば当該証明書の署名IDをキーチェーンから削除してください（[“署名IDをキーチェーンから削除する”](#)（184 ページ）を参照）。

失効させたのが自分の開発用/配布用証明書であれば、対応する署名IDをキーチェーンから削除します。そうでなければ、証明書の所有者が、Macから署名IDを削除しなければなりません。

3. 必要ならばXcodeで新しい証明書を作成してください（[“署名IDを作成する”](#)（172 ページ）を参照）。

開発用証明書を失効させると、Xcodeの「General」ペイン上、「Team」ポップアップメニューの下に警告メッセージが現れます。「Fix Issue」ボタンを押すと、開発用証明書が作成され、さらにチームプロビジョニングプロファイルが再生成されるので、メンバーセンターに「invalid」と表示されることはなくなります。



4. 失効した証明書を収容している他のプロビジョニングプロファイルを、削除または再生成してください（[“メンバーセンターでプロビジョニングプロファイルを編集する”](#)（220 ページ）を参照）。

メンバーセンターで作成した、配布用プロビジョニングプロファイルや（独自作成の）開発用プロビジョニングプロファイルを、Xcodeが自動的に再生成することはありません。

5. メンバーセンターでプロビジョニングプロファイルに変更を施した場合は、Xcode上でこれをダウンロードしてください（[“プロビジョニングプロファイルをメンバーセンターからダウンロードする”](#)（225 ページ）を参照）。
6. 必要ならば、修正済みのプロビジョニングプロファイルをデバイスに組み込みます（[“デバイス上でプロビジョニングプロファイルを確認して削除する”](#)（223 ページ）を参照）。
7. 主たるMac上で証明書を修復した後、開発者プロファイルをエクスポートしてください（[“開発者プロファイルをエクスポートする”](#)（181 ページ）を参照）。

複数のMacコンピュータを修理している場合は、別のMacコンピュータで以下の追加手順を実行します。

1. 署名IDをキーチェーンから削除します（[“署名IDをキーチェーンから削除する”](#)（184 ページ）を参照）。
2. 主たるMac上で生成した開発者プロファイルをインポートしてください（[“開発者プロファイルをインポートする”](#)（183 ページ）を参照）。

署名証明書の詳細

キーチェーンに登録されているコード署名IDは、Apple Developer Programの開発用/配布用証明書を表しています。証明書の名前はメニューに表示され、その種類はリストに表示されるので、十分に確認し、誤ってキーチェーンやメンバーセンターから消去してしまわないようにしてください。

目的に応じて各種の署名証明書があります。**開発用証明書**は、チームメンバーごとに存在し、デバイス上でアプリケーションを実行するために使います。開発/テストの際には、デバイス上で実行するすべてのiOS/tvOS/watchOSアプリケーションと、iCloudやGame Centerなどのアプリケーションサービスを組み込んだMacアプリケーションに、署名を施す必要があります。

配布用証明書はチームにひとつ存在し、ストアにアプリケーションを登録する、あるいはストアを通さずに配布する（Macアプリケーションの場合）際に使います。組織の場合、配布用証明書は、アプリケーションの登録権限を持つチームメンバーが共有します。配布方法によっていくつか種類があります。コード署名IDも、Macアプリケーション用とは異なります。

署名証明書はAppleが発行、認証します。この証明書を使うためには、Appleが提供する中間証明書をシステムキーチェーンに登録しておかなければなりません。Appleが提供し、Xcodeが登録する中間証明書には、次の2種類があります。

- **Apple Worldwide Developer Relations Certification Authority**。開発用証明書、ストア証明書の検証に使用します。
- **Developer ID Certification Authority**。Mac App Storeを通さずに配布するとき、Developer ID証明書を検証するために使用します。

表 14-2に、証明書の種類と、Keychain Accessに表示される名前、およびその目的を示します。

メンバーセンターには、証明書ごとに、チーム名（または個人名）と種類が表示されます。Xcodeの「Accounts」環境設定画面を開くと、「Signing Identities」列に証明書の種類が表示されています。一方、Keychain Accessや、Xcodeのビルド設定に用いる「Code Signing Identity」ポップアップメニューには、証明書の名前が表示されます。

MacまたはiOSの開発用証明書は、チームメンバーごとにそれぞれ存在します。したがってその名前には、個人名が含まれています。それ以外の証明書はチーム単位（複数のチームメンバーが共有）なので、名前にはチーム名が含まれています。個人開発者は1人のチームになるので、個人名とチーム名は同じです。

表 14-2 証明書の種類と名前

証明書の種類	証明書の名前	説明
iOS Development	iPhone Developer: チームメンバー名	開発中、デバイス上でiOS/tvOS/watchOSアプリケーションを実行するとともに、特定のアプリケーションサービスを利用する。
iOS Distribution	iPhone Distribution: チーム名	テスト用として指定したデバイスにiOS/tvOS/watchOSアプリケーションを配布し、あるいはストアに登録する。
Mac Development	Mac Developer: チームメンバー名	開発/テストの際に、特定のアプリケーションサービスを有効にする。
Mac App Distribution	3rd Party Mac Developer Application: チーム名	Mac App Storeに登録する際、Macアプリケーションに署名する。
Mac Installer Distribution	3rd Party Mac Developer Installer: チーム名	Macインストーラパッケージ（署名済みのアプリケーションを収容）に署名を施し、Mac App Storeに登録する。
Developer ID Application	Developer ID Application: チーム名	Mac App Storeを通さずに配布する際、Macアプリケーションに署名する。
Developer ID Installer	Developer ID Installer: チーム名	Macインストーラパッケージ（署名済みのアプリケーションを収容）に署名を施し、Mac App Storeを通さずに配布する。

この資料では、メンバーセンターに現れる他の種類の証明書については説明しません。

- Pass Type ID証明書について詳しくは、『*Passbook Programming Guide*』の“Interacting with Passes in Your App”を参照してください。
- Website Push ID証明書については『*Notification Programming Guide for Websites*』に解説がありません。
- VoIP Services証明書については、『*Energy Efficiency Guide for iOS Apps*』の「Voice Over IP (VoIP) Best Practices」で説明しています。
- Apple Pay証明書については、『*Apple Pay Programming Guide*』の「Configuring Your Environment」を参照してください。

まとめ

この章では、アプリケーションが存続する期間中、継続的に使うことになる開発用と配布用の署名IDを管理する方法を説明しました。また、Xcode、Keychain Access、メンバーセンターで、各種の証明書を識別する方法も説明しました。

ID、デバイス、プロファイルの管理

メンバーセンターでは、アプリケーションのコード署名に用いるID、デバイス、プロファイル进行管理し、アプリケーションをデバイス上で起動できるようにし、特定のアプリケーションサービスを利用可能にすることができます。Xcodeは開発期間を通して、このようなアセットを生成、管理します。テスト用に配布できる状態になって以降、一部のアセットはメンバーセンターで管理することになります。たとえば、独自の開発用プロビジョニングプロファイルを作成し、テストに用いるデバイスを一括登録することも可能です。この章では、ID、デバイス、プロファイル进行管理するための、さまざまな作業について説明します。

Website Push IDの使い方については『*Notification Programming Guide for Websites*』を参照してください。

App IDを登録する

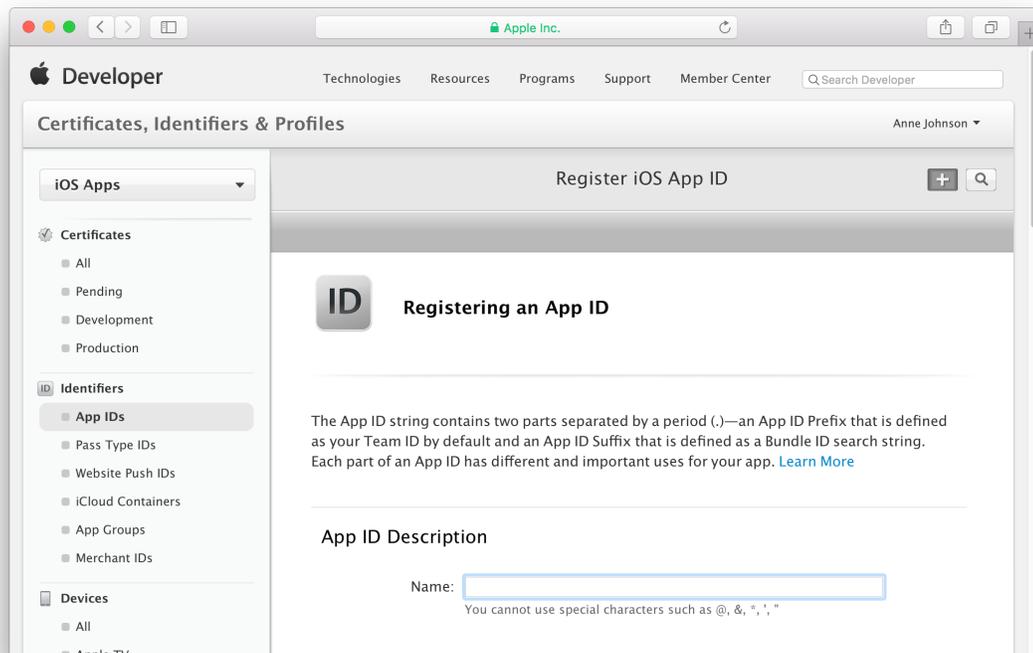
App IDには、複数のアプリケーションに合致するワイルドカード型と、あるバンドルIDに正確に合致する特定型の2種類があります。アプリケーションサービスをApp IDに対して有効にすると、1つまたは複数のアプリケーションで利用できるサービスのホワイトリストに登録されます。アプリケーションが実際に利用するサービスは、Xcodeプロジェクトで設定します。アプリケーションサービスを有効にする手続きは、App IDを生成する時点で行うほか、後でこの設定を変更しても構いません。特定App IDは、最初からGame CenterおよびIn-App Purchaseが有効になっています。

多くの場合、何らかの技術やサービスをアプリケーションに組み込む時点でXcodeが生成する（“[技術やサービスを組み込む](#)”（56 ページ）を参照）、ワイルドカード型または特定型のApp IDがあれば、開発、テスト、配布には充分です。特定App IDはアプリケーションごとに1つしかないので、既に生成済みであれば、Xcodeはこれを使ってチームプロビジョニングプロファイルを生成します。

App IDを登録するには

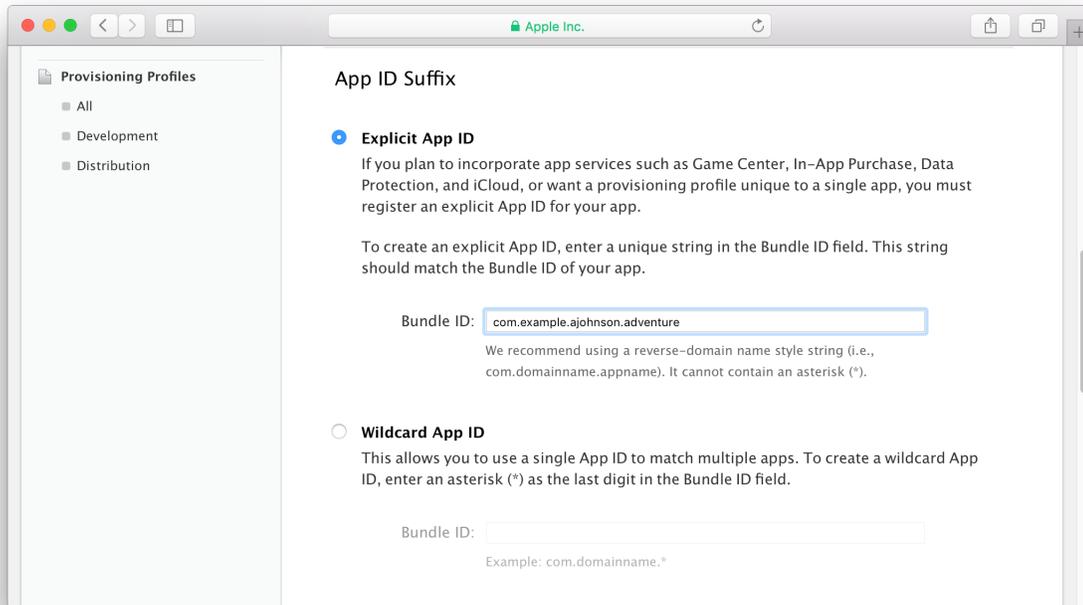
1. メンバーセンターで「[Certificates, Identifiers & Profiles](#)」を選択してください。
2. 「Identifiers」以下の「App IDs」を選択します。
3. 右上隅にある「Add」ボタン（「+」の表示）を押します。

4. App IDの名前または説明を「Description」フィールドに入力します。



5. 特定App IDを生成するのであれば、「Explicit App ID」をオンにし、「Bundle ID」欄にバンドルIDを入力してください。

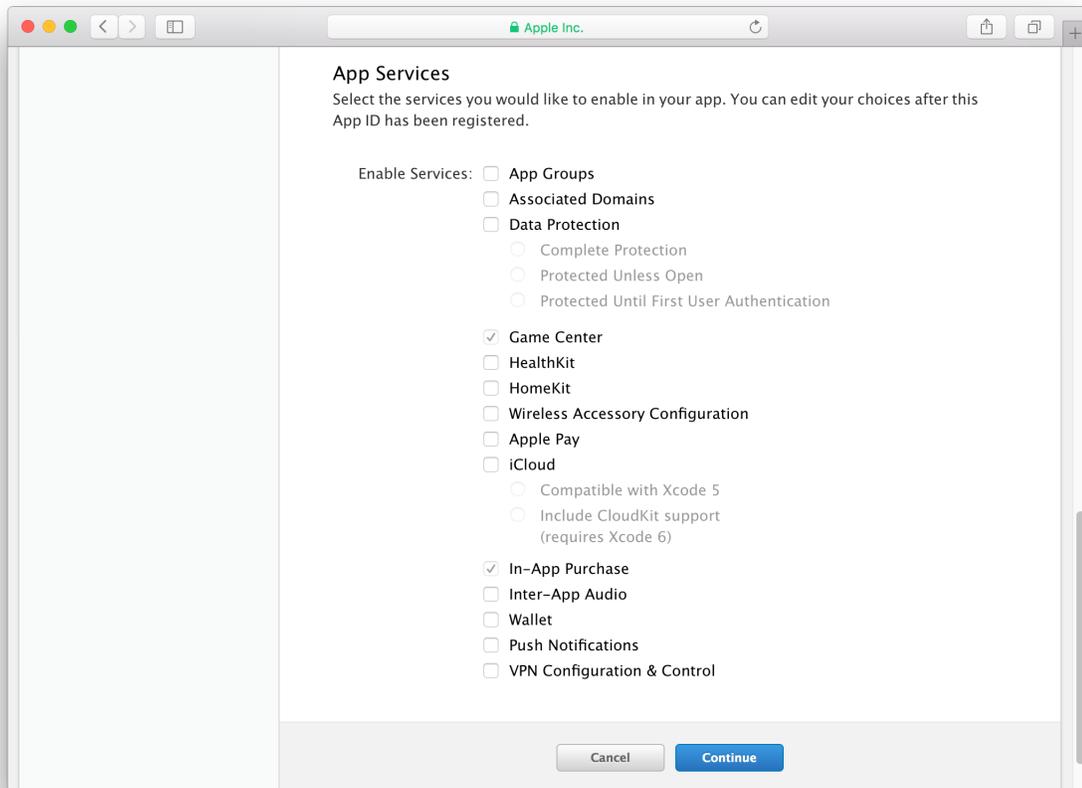
これはビルドするアプリケーションのバンドルID（たとえば「com.example.ajohnson.adventure」）と完全に一致するものです。アスタリスク（*）を含めることはできません。



Xcodeで、ターゲットの「Summary」ペインにバンドルIDを入力した場合、ここで入力するApp IDは、バンドルIDと合致したものでなければなりません。

6. ワイルドカードApp IDを生成する場合は、「Wildcard App ID」をオンにし、「Bundle ID」欄にバンドルIDのサフィックスを入力してください。
7. 有効にしたいアプリケーションサービスに対応するチェックボックスをオンにします。

ワイルドカードApp IDを選択した場合、特定App IDを要する技術に対応するチェックボックスはオフになっています。それ以外はいずれも、デフォルト値はオンです。



8. 「Continue」をクリックします。
9. 登録情報を確認し、「Submit」を押します。
10. 「Done」を押してください。

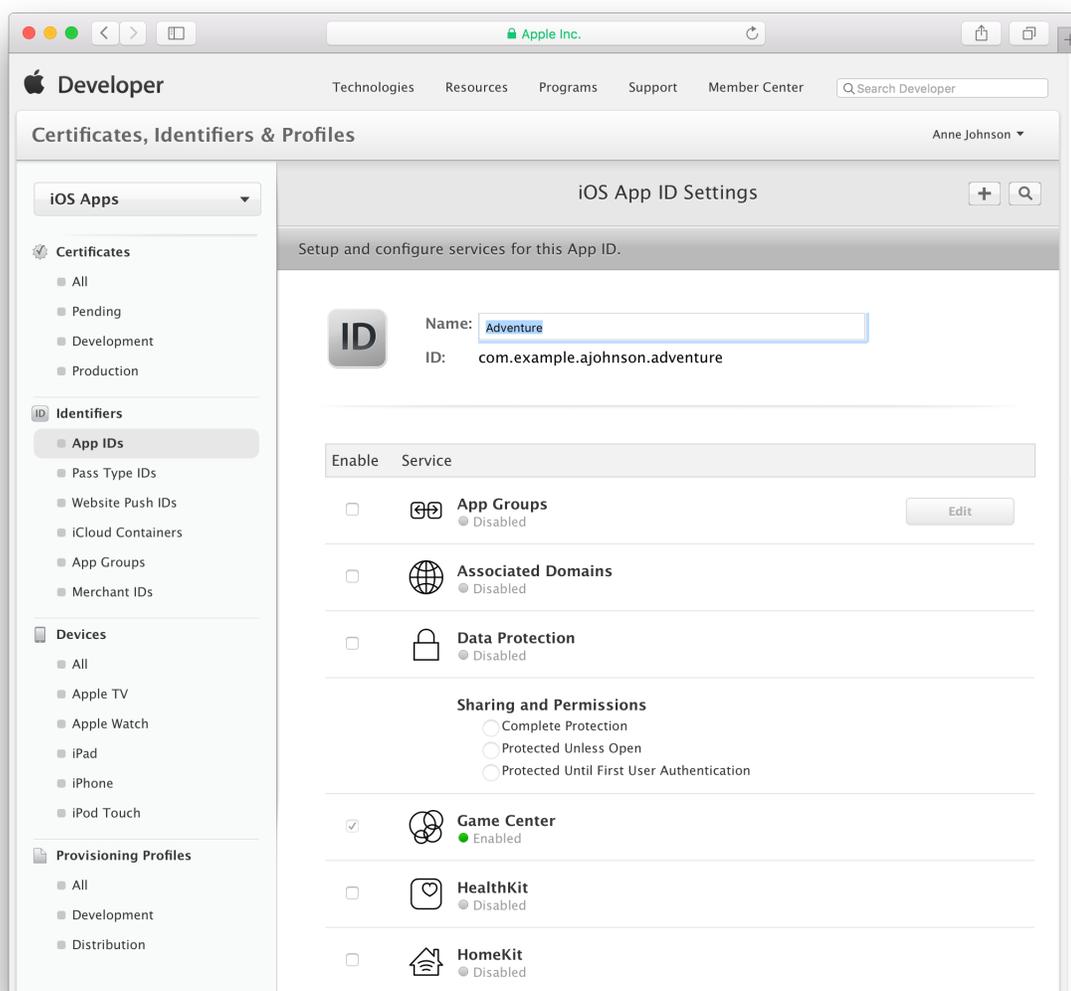
App IDを編集する

App IDは、メンバーセンターで直接編集できます。

既存のApp IDに対してアプリケーションサービスを有効にするには

1. メンバーセンターで「[Certificates, Identifiers & Profiles](#)」を選択してください。
2. 「Identifiers」以下の「App IDs」を選択します。
3. 設定を変更しようとするApp IDを選択し、「Edit」を押してください。

- 有効にしたいアプリケーションサービスに対応するチェックボックスをオンにします。



- 警告ダイアログが現れた場合は「OK」を押してください。
後でこのApp IDを使うプロビジョニングプロファイルを生成し直すことになります。
- 「Done」を押してください。

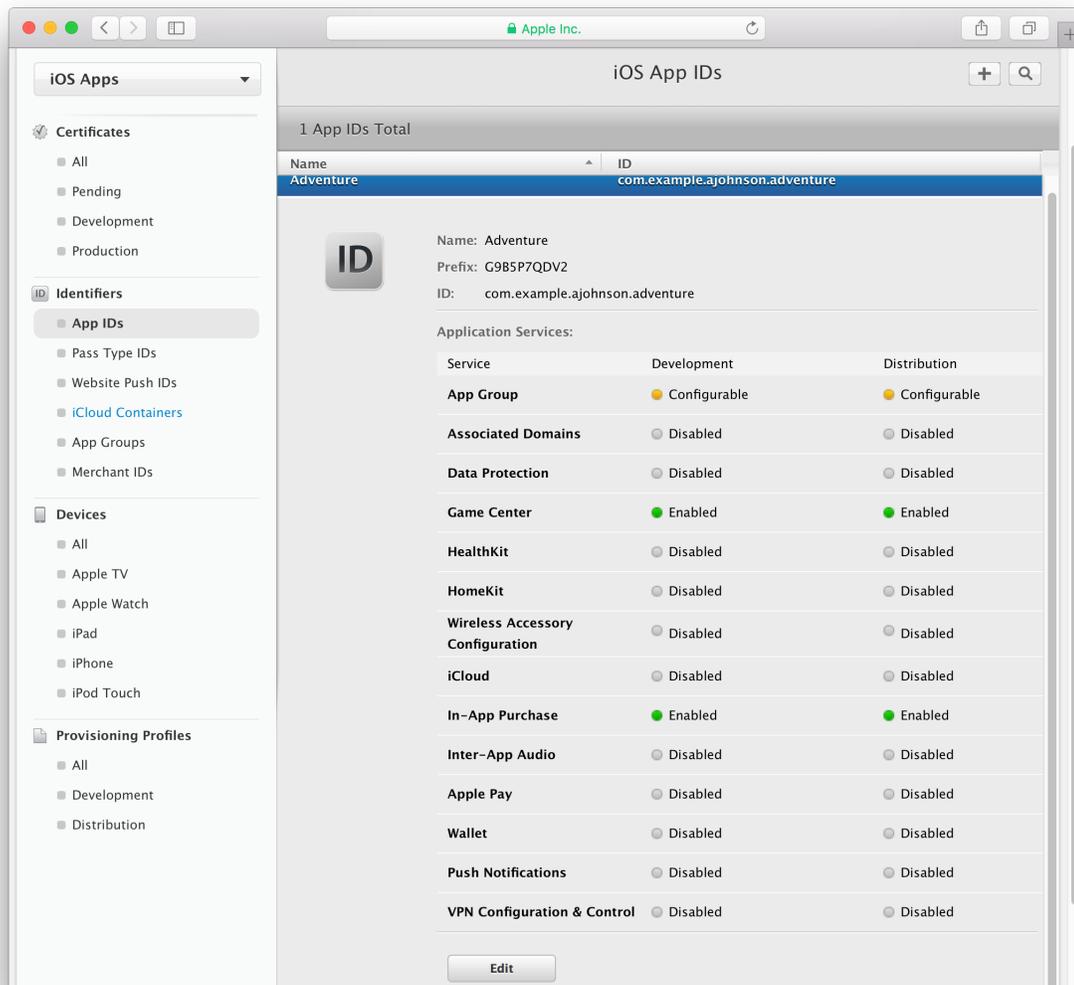
メンバーセンターに「設定可能」と表示されているアプリケーションサービスを完全に有効にする手順については、「[技術やサービスを組み込む](#)」（56 ページ）を参照してください。

App IDを削除する

不要になったApp IDは削除して構いません。ただし、iTunes Connectにアップロードするアプリケーションの特定App IDを削除することはできません。

App IDを削除するには

1. メンバーセンターで「[Certificates, Identifiers & Profiles](#)」を選択してください。
2. 「Identifiers」以下の「App IDs」を選択します。
3. 削除するApp IDを選択して「Edit」をクリックします。



4. ページの一番下までスクロールして「Delete」をクリックします。
5. 表示されるダイアログの内容に目を通した上で「Delete」をクリックします。

このApp IDが設定されたプロビジョニングプロファイルは無効になります。プロビジョニングプロファイルに設定されたApp IDを変更する（[“メンバーセンターでプロビジョニングプロファイルを編集する”](#)（220 ページ）を参照）か、または削除してください。

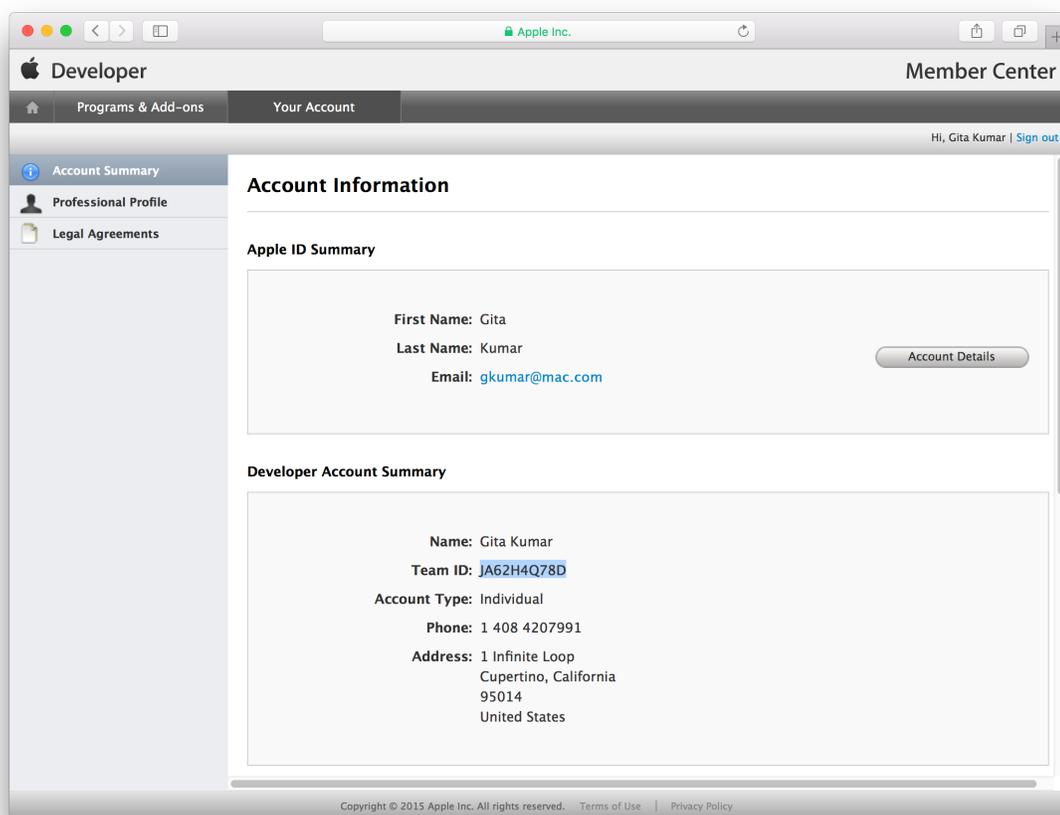
チームIDを調べる

チームIDを調べる必要が生じることもあります。たとえば、他の開発者からアプリケーションの所有権を送ってもらう場合、iTunes Connectでの送信先を示すチームIDを知らせる必要があります。チームIDはAppleが生成する10字の一意的な文字列で、チームに対して割り当てられます。どのような文字列かは、メンバーセンターで調べることができます。

チームIDを調べるには

1. [メンバーセンター](#)を開き、「Your Account」画面に切り替えてください。

するとチームIDが、「Developer Account Summary」セクション内、チーム名の行の下に表示されます。



メンバーセンターでデバイスを登録する

メンバーセンターでは、必要に応じて個別にデバイスを登録するほか、各デバイスの情報を記述したファイルをアップロードすることにより、一括して登録することも可能です。登録可能なデバイスの数は、1年ごとに製品ファミリーにつき100台までとなっています。何台か登録を解除しても（[“メンバーセンターでデバイスの有効/無効を切り替える”](#)（208 ページ）を参照）、その年内は登録済み数が減りません。

デバイスIDを調べる

デバイスを登録するためには、その名前とデバイスIDが必要です。Xcodeがインストール済みか否かに応じて、デバイスIDの調べ方は何通りかあります。

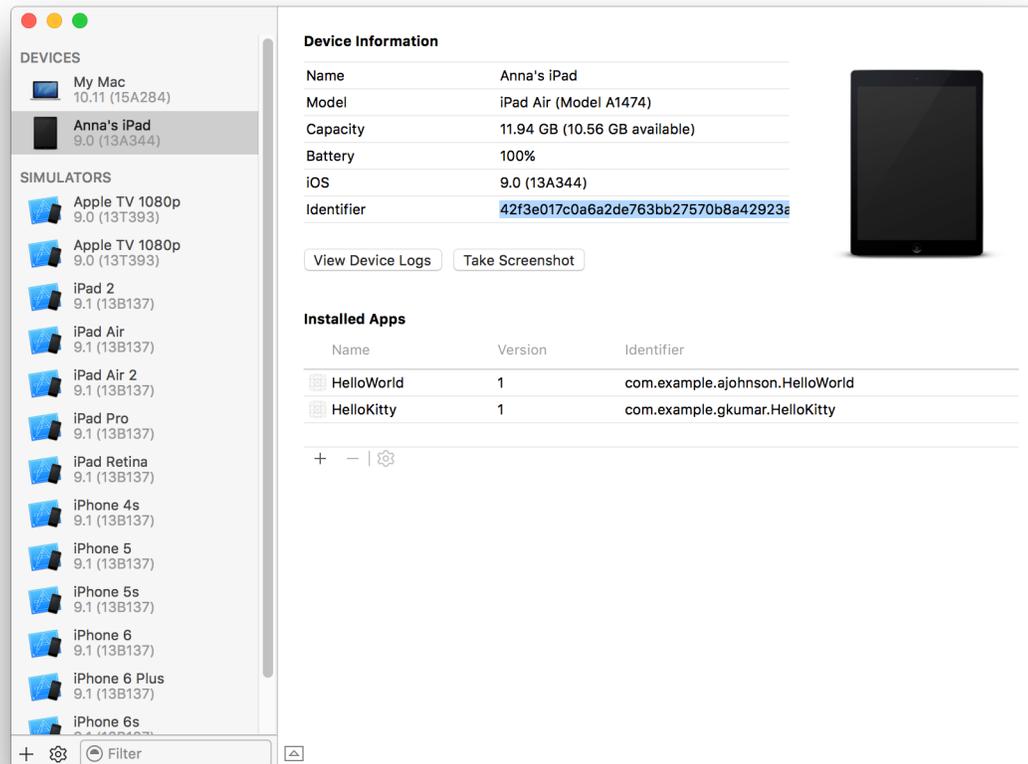
XcodeでデバイスIDを調べる

チームメンバーは、Xcodeの「Devices」ウィンドウで該当するデバイスを選択し、デバイスIDを調べることができます。

XcodeでデバイスIDを調べるには

1. 「Window」 > 「Devices」 コマンドを実行してください。
2. iOS/tvOSアプリケーションの場合、デバイスをMacに接続します。watchOSアプリケーションについては、Apple WatchとペアリングしたiPhoneを接続します。
3. 「Devices」 ウィンドウの「Devices」 から該当するデバイスを選択します（自分のMacもデバイスと見なします）。

デバイスIDが「Identifier」フィールドに表示されます。



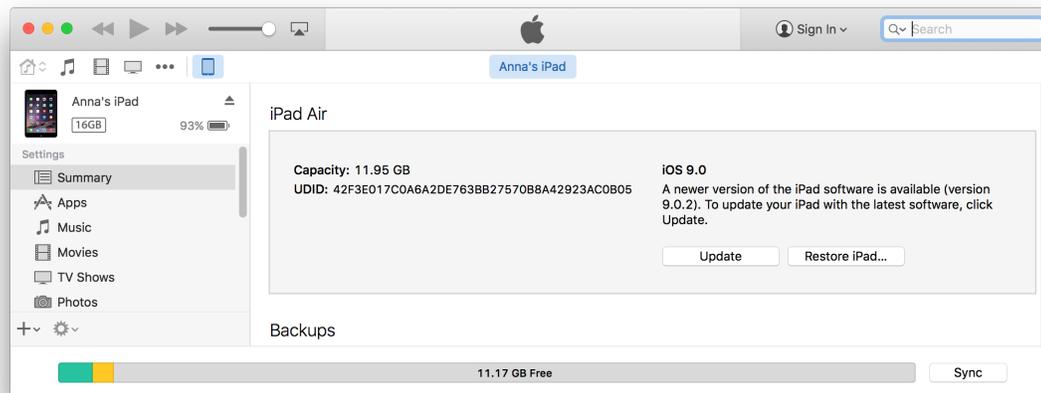
iTunesでiOSデバイスIDを調べる

iOSデバイスについては、iTunesでデバイスIDを調べることも可能です。たとえばテスト担当者は、Xcodeがない環境でも、以下の手順により、iTunesでデバイスIDを調べることになります。

iTunes上でiOSデバイスIDを調べるには

1. Mac上でiTunesを起動します。
2. デバイスをMacに接続します。
3. 右上隅で該当するデバイスを選択します。
4. 「Summary」ペインで、「Capacity」または「Phone Number」の下にある、「Serial Number」というラベルをクリックしてください。

ラベルが「Serial Number」から「UDID」に変わり、デバイスIDが表示されます。



5. Controlキーを押しながら該当するIDをクリックし、ショートカットメニューから「Copy」を選択して、デバイスIDをコピーします。
6. その後、文書や電子メールの本文に、デバイスIDの文字列を貼り付けてください。

「System Information」でデバイスIDを調べる

「System Information」アプリケーションでデバイスIDを調べることができます。たとえば開発に使っていないMacをテスト用に登録するときには、この方法で調べるとよいでしょう。

「System Information」でMacのデバイスIDを調べるには

1. 「System Information」アプリケーション（/Applications/Utilitiesフォルダ以下）を起動します。
2. 左の列から「Hardware」を選択してください。
デバイスID（あるいはハードウェアUUID）は、「Hardware Overview」ペインの下の方に表示されます。「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX」という形式の文字列です。

この方法は、Xcodeを使わずにiOS/Apple TVデバイスのデバイスIDを収集するためにも有用です。

「System Information」で接続されたデバイスのIDを調べるには

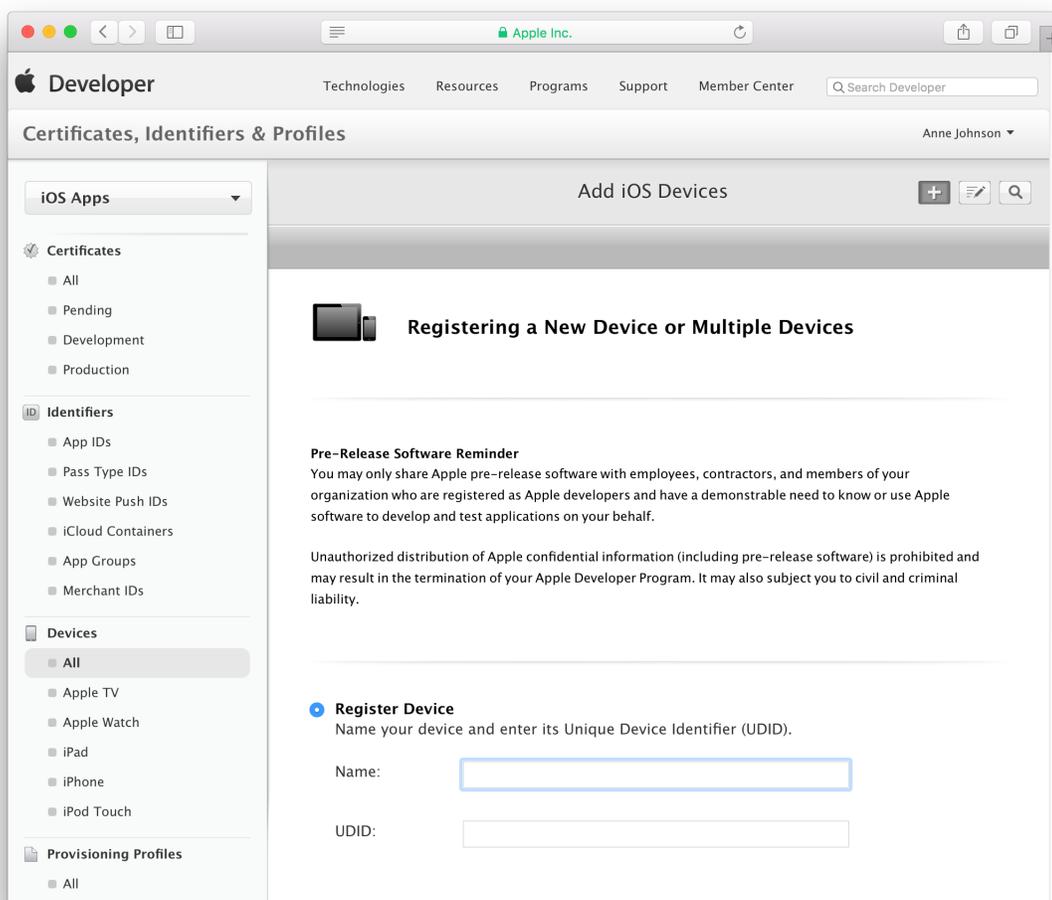
1. デバイスをMacに接続します。
2. 「System Information」アプリケーション（/Applications/Utilitiesフォルダ以下）を起動します。
3. 左列の「Hardware」以下にある「USB」を選択してください。
4. 右側の「USB Device Tree」以下で、接続されたデバイスを選択してください。
デバイスIDまたは「シリアル番号」が下に表示されます。

個々のデバイスを登録する

メンバーセンターでデバイスを登録するためには、あらかじめデバイス名とデバイスIDを調べておく必要があります。

個別にデバイスを登録するには

1. 「Certificates, Identifiers & Profiles」以下の「Devices」を選択します。
2. 「Devices」以下の「All」を選択してください。
3. 右上隅にある「Add」ボタン（「+」の表示）を押します。
4. 「Register Device」を押してください。
5. デバイス名とデバイスID（UUID）を入力します。



6. ページの下までスクロールし、「Continue」を押します。
7. 登録情報を確認し、「Register」を押します。

デバイスを一括登録する

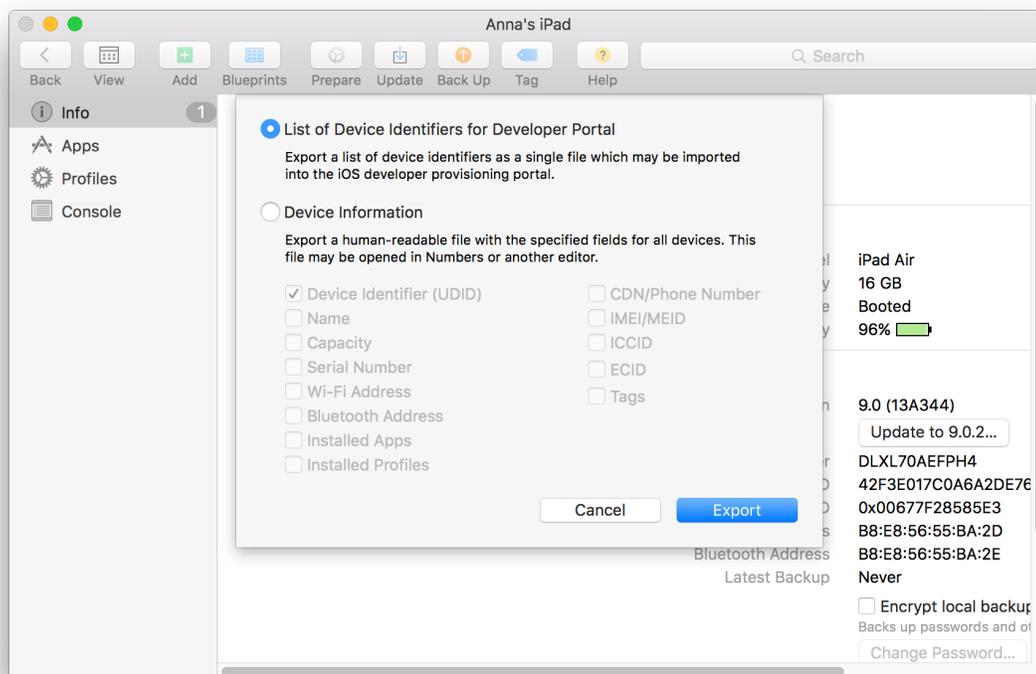
多数のテストデバイスを登録する場合、デバイス名とデバイスIDを列挙したファイルを用意し、メンバーセンターにアップロードする、という方法もあります。記述方法には、拡張子「.deviceids」のプロパティリスト形式と、プレーンテキスト形式の2通りがあります。登録するデバイスにアクセスできるか否か、に応じて選択してください。

プロパティリスト形式でデバイス登録ファイルを作成するApple Configuratorを利用

テスト用デバイスにアクセスできるのであれば、Apple Configuratorを使って、デバイスの名前とIDを記述した、プロパティリスト形式のファイルを生成できます。Mac App StoreからApple Configuratorをダウンロードします。

プロパティリスト形式のデバイスファイルを作成するには

1. Apple Configuratorを起動し、「Actions」>「Export」>「Info」を実行してください。
2. 「List of Device Identifiers for Developer Portal」を選択し、「Export」を押します。



3. ファイル名と保存先の場所を指定し、「Save」を押してください。
ファイルの拡張子は「.deviceids」となります。

プレーンテキスト形式でデバイス登録ファイルを作成する

テスト用デバイスにアクセスできない場合、デバイス名とデバイスIDを別の手段で収集し、.txtファイルを作成してください。各行に、デバイスIDとデバイス名を、タブで区切って記述します。なお、1行めは読み飛ばされるので、見出し行として使って構いません。

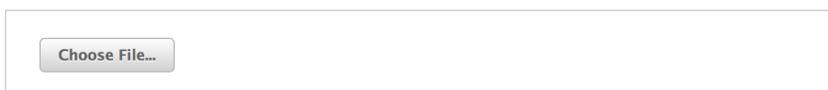
デバイス登録ファイルをアップロードする

どちらの形式のデバイス登録ファイルも、[メンバーセンター](#)で、同じ手順でアップロードできます。

デバイスを一括登録するには

1. 「[Certificates, Identifiers & Profiles](#)」以下の「[Devices](#)」を選択します。
2. 「[Devices](#)」以下の「[All](#)」を選択してください。
3. 右上隅にある「[Add](#)」ボタン（「+」の表示）を押します。
4. 「[Register Multiple Devices](#)」をオンにしてください。
5. 「[Choose File](#)」ボタンを押します。

- **Register Multiple Devices**
Upload a file containing the devices you wish to register. Please note that a maximum of 100 devices can be included in your file and it may take a few minutes to process.
[Download sample files](#)



6. アップロードするデバイス登録ファイルを選択し、「[Choose](#)」を押してください。先に用意した、拡張子が「.deviceids」または「.txt」のファイルを指定します。
7. 「[Continue](#)」を押してください。
8. 登録情報を確認し、「[Submit](#)」を押します。

プレリリース版をiOS/Apple TVデバイスにインストールする

事前にインストールしようとするプレリリース版を[Apple Developer](#)からダウンロードしておいてください。

プレリリース版を開発用デバイスにインストールするには

1. デバイスをMacに接続します。
2. iTunesの左上隅から該当するデバイスを選択します。
Apple TVデバイスの場合、iTunesにデバイスが現れなければ、復旧モードにしてください。

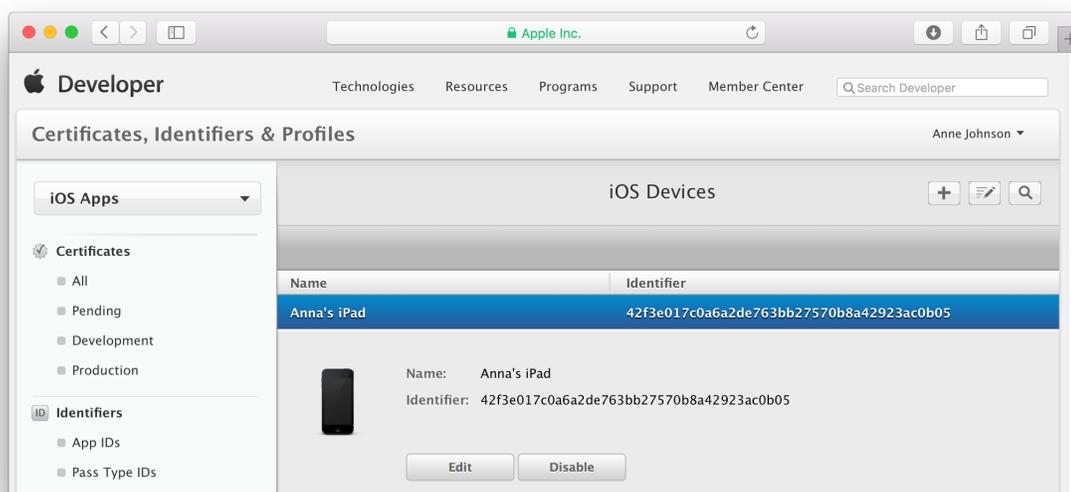
3. 「Summary」ペインで、Optionキーを押しながら「Restore iPhone/iPad/iPod」ボタンまたは「Restore Apple TV」ボタンを押してください。
4. プレリリース版ソフトウェアのリストアイメージを選択し、「Open」を押してインストールを開始します。
5. インストール終了後、デバイスをアクティブ化し、その内容をiTuneから復元してください。

メンバーセンターでデバイスの有効/無効を切り替える

メンバーセンターで、デバイスの有効/無効を切り替えることはできますが、削除はできません。今後、開発やテストに使わないデバイスは、無効にすることができます。しかしこれは、そのデバイスの設定を収容しているプロビジョニングプロファイルは無効にするだけであり、年度内に使えるデバイス総数が増えるわけではありません。また、いったん無効にしたデバイスを有効にすることも可能です。

デバイスを無効に/有効にするには

1. 「Certificates, Identifiers & Profiles」以下の「Devices」を選択します。
2. 「Devices」以下の「All」を選択してください。
3. 無効に/有効にするデバイスを選択します。
4. 「Enable」または「Disable」を押してください。



5. するとダイアログが現れるので、再び「Enable」または「Disable」を押します。

デバイスを無効にした後、無効になったチームプロビジョニングプロファイルを再生成する手順については、「[Xcode上でプロビジョニングプロファイルをダウンロードする](#)」（215 ページ）を参照してください。無効になったデバイスを、独自に管理している他のプロビジョニングプロファイルから削除する手順は、「[メンバーセンターでプロビジョニングプロファイルを編集する](#)」（220 ページ）を参照してください。

Xcode上でデバイスを有効にすることも可能です。iOSアプリケーションの場合は、プロジェクトエディタの「Scheme」ツールバーメニューで選択されている接続済みデバイスがXcodeで自動的に登録されます。Macアプリケーションであれば、Xcodeが動作しているMacが自動的に登録されます。デバイスやMacが既に登録済みであれば、無効であってもXcodeが有効にします。

デバイスを無効にしても、古い版のプロビジョニングプロファイルがインストールされていれば、アプリケーションが動作することがあります。古いプロビジョニングプロファイルを削除する手順については、「[デバイス上でプロビジョニングプロファイルを確認して削除する](#)」（223 ページ）を参照してください。

デバイス上のアプリケーションを管理する

「Devices」ウィンドウで、iOS/tvOSデバイスにインストールされている各アプリケーションを管理できます。まず、デバイスをMacに接続し、「Window」>「Devices」の順に選択した後、「Devices」から該当するデバイスを選択します。iOS/tvOSアプリケーションの場合、デバイスをMacに接続します。watchOSアプリケーションについては、Apple WatchとペアリングしたiPhoneを接続します。

からアプリケーションを削除する

デバイス上のアプリケーションに関連したデータファイルをすべて削除するには、該当するデバイスからアプリケーションを削除します。この操作は、アプリケーションの初回起動時のみ実行されるコードをテストする目的で行っても構いません。Xcode上で削除できるのは、自分が作成したアプリケーションのみです。システムに組み込まれたアプリケーションや他社製のアプリケーションをXcodeで削除することはできません。

アプリケーションをデバイスから削除するには

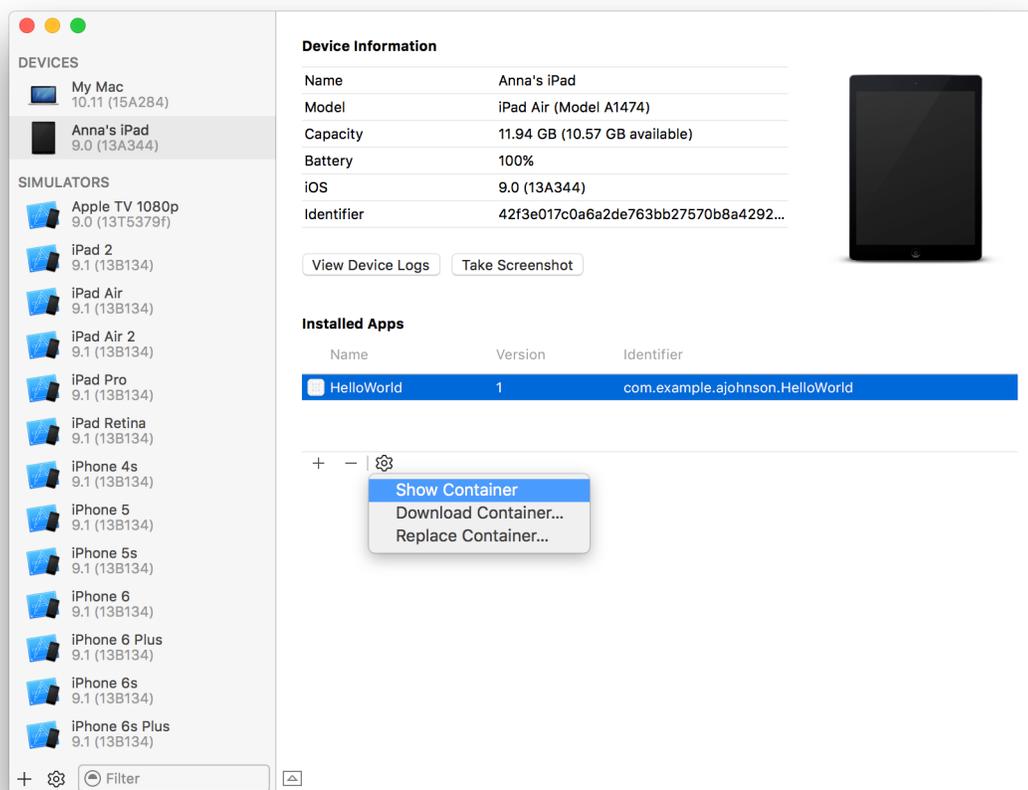
1. 「Installed Apps」テーブルで、アプリケーションを選択して、下部にある「Delete」ボタン（「-」の表示）をクリックします。
2. 表示されるダイアログ、「Delete」ボタンをクリックします。

デバイスでアプリケーションコンテナを表示、ダウンロード、置換する

iOSデバイス上のアプリケーションコンテナのファイル構造はXcodeで直接表示できますが、ファイルの内容は表示できません。

アプリケーションコンテナの内容を表示するには

1. 「Installed Apps」にあるリストから該当するアプリケーションを選択します。
2. 「Action」メニュー（「Delete」ボタンの右にある歯車アイコン）から「Show Container」を選択します。



3. 「Done」をクリックします。

アプリケーションコンテナに含まれるファイルを確認または変更するには、該当するファイルをファイルシステムにダウンロードします。アプリケーションコンテナの内容を分析すると、問題を診断する場合や細部を調整する場合に役立つ重要な情報が得られることがあります。

アプリケーションコンテナの内容をダウンロードするには

1. 「Installed Apps」にあるリストから該当するアプリケーションを選択します。
2. 「Action」ポップアップメニューから「Download Container」を選択します。

3. 「Save As」テキストフィールドにファイル名を入力して「Save」をクリックします。

特定の状態にあるアプリケーションをテストする場合は、デバイス上のアプリケーションコンテナを置き換えます。

デバイス上のアプリケーションコンテナを置き換えるには

1. 「Installed Apps」にあるリストから該当するアプリケーションを選択します。
2. 「Action」メニューから「Replace Container」を選択します。
3. ファイルピッカーからコンテナを選択して「Open」をクリックします。

メンバーセンターでプロビジョニングプロファイルを生成する

メンバーセンターでは、開発用と配布用のどちらのプロビジョニングプロファイルも生成できます。

チームプロビジョニングプロファイルはXcodeが生成、管理するので、開発用プロビジョニングプロファイルを独自に生成しなければならないのは、特定のチームメンバーやデバイスに、アプリケーション開発を制限したい場合のみです。開発用プロビジョニングプロファイルを独自に作成する場合は、「[開発用プロビジョニングプロファイルを作成する](#)」（212 ページ）の手順に従ってください。

アプリケーションのアーカイブをエクスポートして、アドホック配備オプションを選択すると、XcodeでiOSアドホックプロビジョニングプロファイルが作成されます（「[ストアを通さず、テスト用にアプリケーションをエクスポートする](#)」（104 ページ）を参照）。アドホックプロビジョニングプロファイルをメンバーセンターで直接作成する場合は、「[アドホックプロビジョニングプロファイルを作成する \(iOS、tvOS、watchOS\)](#)」（213 ページ）を参照してください。

同様に、アプリケーションをストアに登録すると、Xcodeでストアプロビジョニングプロファイルが作成されます。ストアプロビジョニングプロファイルをメンバーセンターで直接作成する場合は、「[ストアプロビジョニングプロファイルを作成する](#)」（215 ページ）を参照してください。

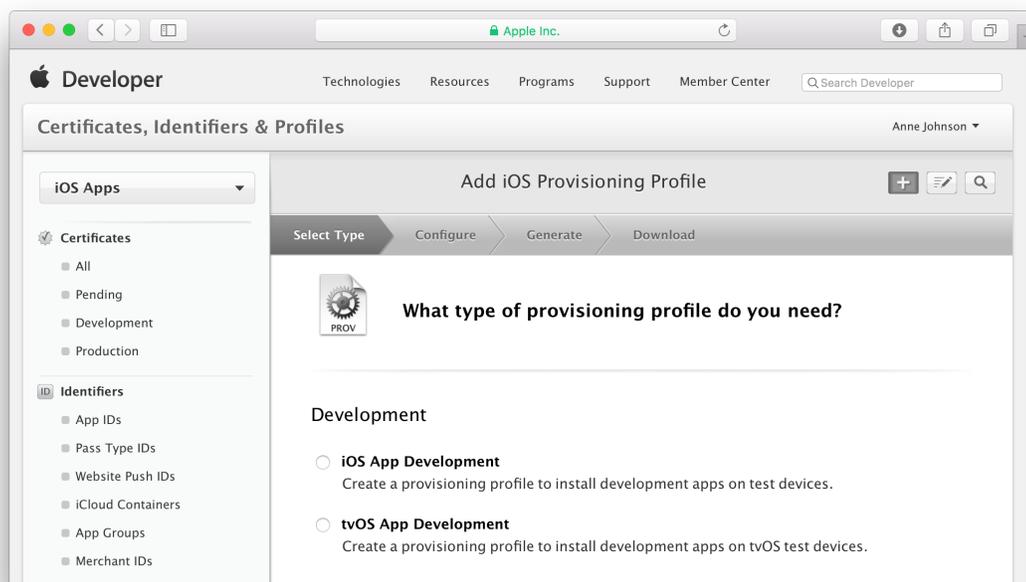
プロビジョニングプロファイルを作成したら、Xcode上でダウンロードしてください（「[Xcode上でプロビジョニングプロファイルダウンロードする](#)」（215 ページ）を参照）。該当するプロビジョニングプロファイルが「Accounts」環境設定画面の詳細表示ダイアログ内の「Provisioning Profiles」テーブルに表示されるはずですが。

開発用プロビジョニングプロファイルを作成する

開発用プロビジョニングプロファイルの生成に先立ち、AppIDと、必要な数の開発用証明書およびデバイスを用意してください。独自に用意したApp IDを登録する手順については、「[App IDを登録する](#)」（195 ページ）を参照してください（Xcodeが管理するApp IDを使っても構いません）。開発用証明書を生成する必要がある場合は、「[署名IDを作成する](#)」（172 ページ）を参照してください。デバイスの登録が必要な場合は、「[メンバーセンターでデバイスを登録する](#)」（202 ページ）を参照してください。

開発用プロビジョニングプロファイルを作成するには

1. 「[Certificates, Identifiers & Profiles](#)」の画面の「Provisioning Profiles」以下で「All」を選択します。
2. 右上隅にある「Add」ボタン（「+」の表示）を押します。
3. 「Development」以下で、作成しようとするプロビジョニングプロファイルの種類を選択し、「Continue」を押してください。
 - iOS/watchOSアプリケーションの場合は「iOS App Development」を選択します。
 - tvOSアプリケーションならば「tvOS App Development」です。
 - Macアプリケーションであれば「Mac App Development」を選んでください。



4. 開発に使うApp IDを選択し、「Continue」を押します。
5. 開発用証明書（複数可）を選択し、「Continue」を押します。
6. デバイス（複数可）を選択し、「Continue」を押します。
7. プロファイル名を入力し、「Generate」を押します。
8. 「Done」を押してください。

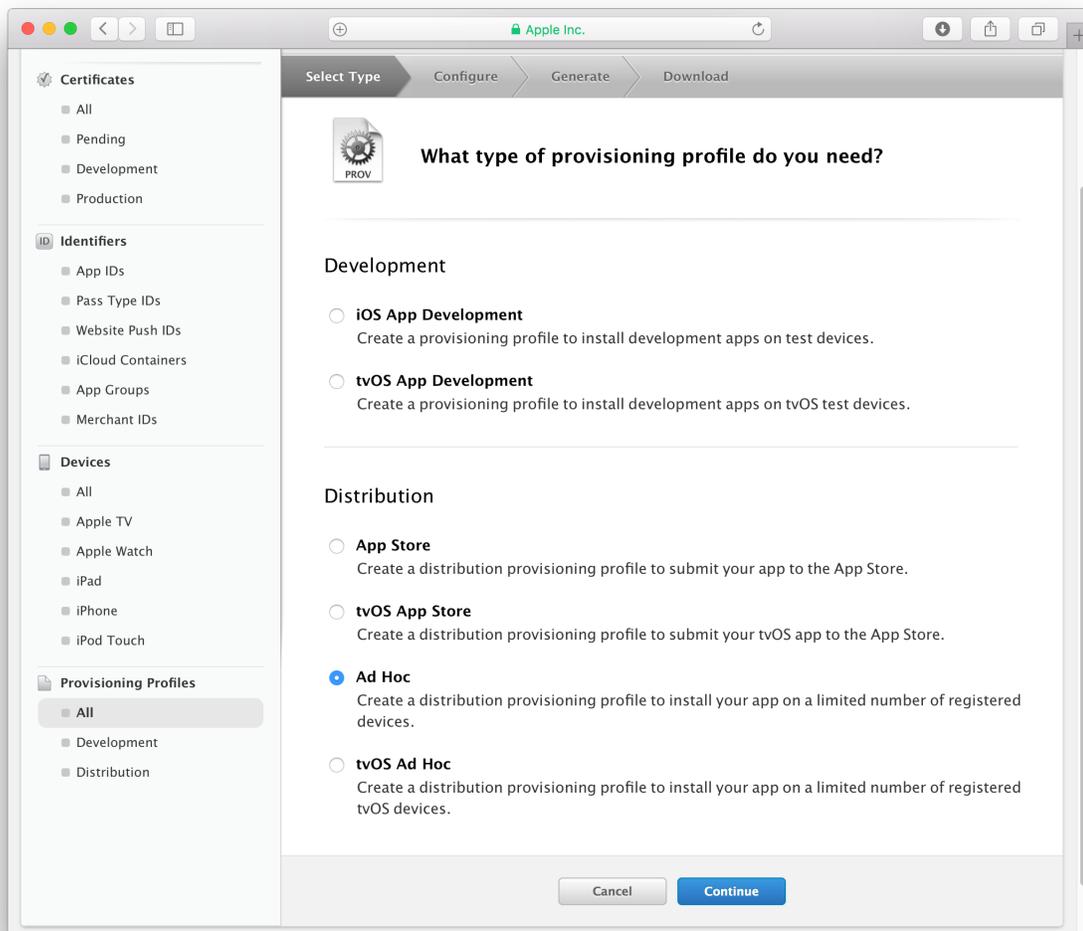
アドホックプロビジョニングプロファイルを作成する (iOS、tvOS、watchOS)

アドホックプロビジョニングプロファイルを作成すると、テスト担当者はXcodeを使わずにデバイス上でアプリケーションを実行できるようになります。作成の際には、AppIDと配布用証明書、そして実際にテストに使うデバイス群を指定します。

Important: 配布用証明書がない場合は、Xcodeを使って作成してから（「署名IDを作成する」（172ページ）を参照）、次に進んでください。

アドホックプロビジョニングプロファイルを作成するには

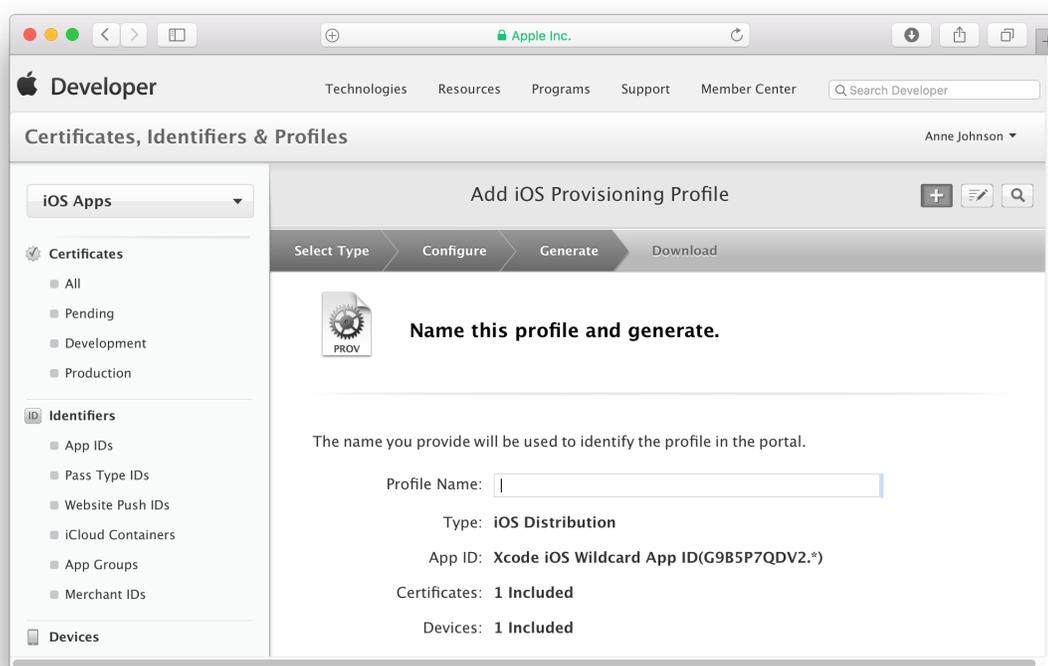
1. 「[Certificates, Identifiers & Profiles](#)」の画面で「Provisioning Profiles」を選択します。
2. 右上隅にある「Add」ボタン（「+」の表示）を押します。
3. 配布方法として「Ad Hoc」または「tvOS Ad Hoc」を選択し、「Continue」を押してください。



4. 開発に使っていたApp ID（バンドルIDと合致するApp ID）を「App ID」ポップアップメニューから選択して「Continue」をクリックします。

開発中、チームプロビジョニングプロファイルを使っていて、メニューに「Xcode iOS Wildcard AppID」しか現れていない場合は、これを選択します。Xcodeが管理する別の特定AppID（「Xcode」で始まり、バンドルIDが含まれる名前）があれば、これになります。独自にApp IDを生成した場合は、これを選択してください。

5. 実際に使う配布用証明書を選択し、「Continue」を押してください。
6. テストに使うデバイス群を選択し、「Continue」を押します。
7. プロファイル名を入力し、「Generate」を押します。



メンバーセンターにプロビジョニングプロファイルが生成されるまで、しばらく待ってください。

8. ページの一番下にある「Done」をクリックします。

Xcode上で、アドホックプロビジョニングプロファイルをダウンロードしてください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215ページ）を参照）。該当するアドホックプロビジョニングプロファイルが「Accounts」環境設定画面の詳細表示ダイアログ内の「Provisioning Profiles」テーブルに表示されるはずですが。

ストアプロビジョニングプロファイルを作成する

アプリケーションをストアにアップロードする前に、ストアプロビジョニングプロファイルでプロビジョニングを施します。（Macアプリケーションでどのアプリケーションサービスも有効にしない場合は、配布用証明書だけを使ってアプリケーションにコード署名しても構いません。）ストアプロビジョニングプロファイルを作成する際には、デバイスを選択する必要もありません。

ストアプロビジョニングプロファイルを作成するには

1. 「**Certificates, Identifiers & Profiles**」の画面で「**Provisioning Profiles**」を選択します。
2. 右上隅にある「Add」ボタン（「+」の表示）を押します。
3. 配布方法を選択し、「Continue」を押してください。
 - iOS/watchOSアプリケーションの場合は「App Store」を選択します。
 - tvOSアプリケーションならば「tvOS App Store」です。
 - Macアプリケーションであれば「Mac App Store」を選んでください。

4. 開発に使っていたApp ID（バンドルIDと合致するApp ID）を「App ID」ポップアップメニューから選択して「Continue」をクリックします。

開発中、チームプロビジョニングプロファイルを使っていて、メニューに「Xcode Wildcard App ID」しか現れていない場合は、これを選択します。Xcodeが管理する特定App ID（「Xcode」で始まり、バンドルIDが含まれる名前）があれば、これになります。独自にApp IDを生成した場合は、これを選択してください。

5. 配布用証明書を選択し、「Continue」を押してください。

ストアプロビジョニングプロファイルには、配布用証明書が1つだけ入ります。

6. プロファイル名を入力し、「Generate」を押します。

メンバーセンターにプロビジョニングプロファイルが生成されるまで、しばらく待ってください。

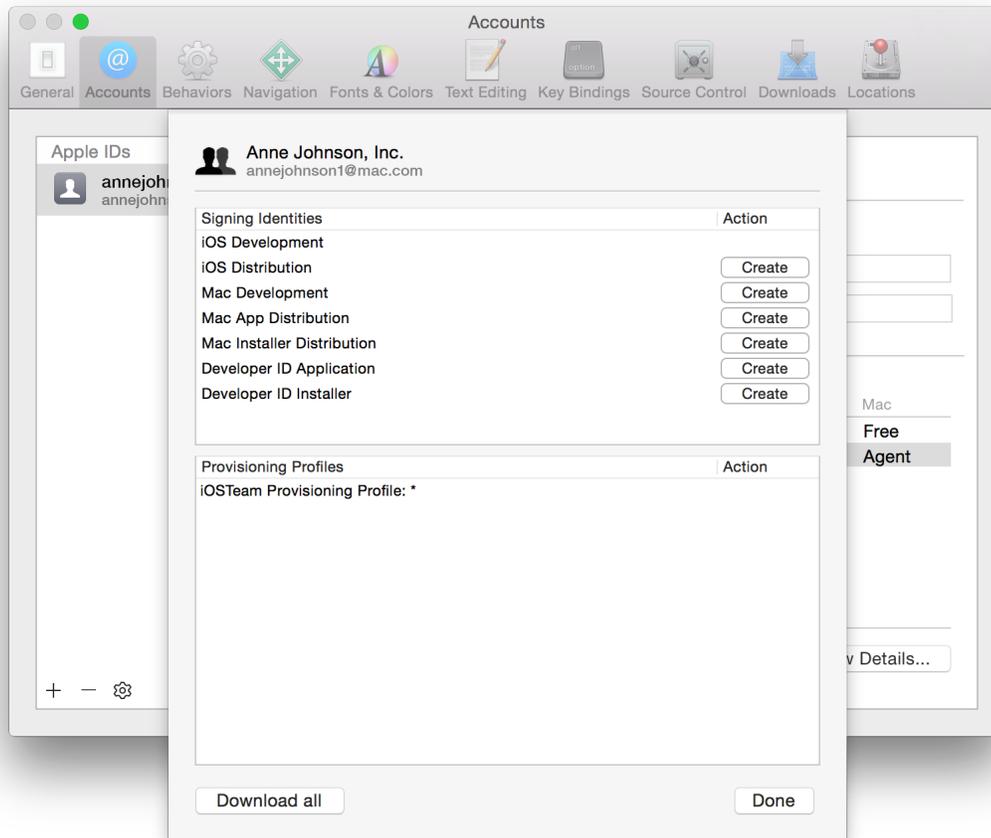
7. ページの一番下にある「Done」をクリックします。

Xcode上でプロビジョニングプロファイルをダウンロードする

メンバーセンター上で何らかの変更を施しても、自動的にXcodeに反映されるわけではありません。たとえば、メンバーセンターでプロビジョニングプロファイルを生成あるいは編集した場合、Xcode側でもプロビジョニングプロファイルをダウンロードして、変更後の内容を取得する必要があります。同様に、開発用証明書を失効させた場合も、プロビジョニングプロファイルをダウンロードすることにより、Xcodeが管理するチームプロビジョニングプロファイルを再生成しなければなりません。

Xcodeでプロビジョニングプロファイルをダウンロードするには

1. Xcodeの「Preferences」ウィンドウで、「Accounts」を押して表示を切り替えます。
2. チームを選択し、「View Details」ボタンを押してください。



3. するとダイアログが現れるので、「Provisioning Profiles」テーブルの下、左下隅にある「Download all」ボタンを押してください。

「Provisioning Profiles」テーブル内のプロファイル一覧が更新されます。

独自のプロビジョニングプロファイルを使用する

場合によっては、Xcodeが管理するチームプロビジョニングプロファイルの代わりに、開発用プロビジョニングプロファイルを独自に用意して使いたいことがあるかも知れません。たとえば、開発作業を一部の開発者のみに限定する、通常とは異なるアプリケーション設定でテストする、などといった状況です。この場合、開発用プロビジョニングプロファイルを生成し（[“メンバーセンターでプロビジョニングプロファイルを生成する”](#)（211ページ）を参照）、これを使ってコード署名IDに関するビルド設定を行います。

アプリケーションを構築する際、該当するプロビジョニングプロファイルに収容された証明書に合致する署名IDを使って、コード署名を施す必要があります。「Code Signing Identity」ポップアップメニューには次の選択肢があります。

- **Don't Code Sign**。アプリケーションに署名しません。サンドボックス化などのエンタイトルメントも無効になります。
- **Automatic Profile Selector**。開発者/配布用証明書の名前に一致するIDを選択します。
- **Identities without Provisioning Profiles**。プロビジョニングプロファイルに含まれている以外のコード署名IDを選択します。
- **Other**。コード署名IDを別途選択します。デフォルトのキーチェーンに登録されているコード署名IDが、名前順に列挙されるので、その中から選んでください。期限切れまたはその他の理由で無効になったIDは、グレイ表示されており選択できません。

ビルド設定の「Code Signing Identity」ポップアップメニューには、開発用証明書が収容されたプロビジョニングプロファイルそれぞれに対応する項目が並んでいます。既定値は、「Automatic Profile Selector」メニュー項目に現れる、プラットフォームに対応した開発用証明書で、これは開発用証明書に合致します。メニューに現れる各種の証明書については表 14-2（193 ページ）を参照してください。

あらかじめ、「Code Signing Identity」ビルド設定を、プロジェクトごと、ターゲットごとのどちらで行うか決めておきます。ターゲットがひとつしかなければ、一貫している限りどちらでも構いません。どのターゲットにも同じコード署名IDを用いる場合は、プロジェクトごとに設定します。一方、ターゲットによってコード署名IDを使い分ける場合は、ターゲットごとに設定してください。プロジェクトごとの設定にすると、たとえば同じプロジェクトで補助アプリケーションをいくつか開発する場合、いずれも主たるアプリケーションと同じ方法でコード署名を施すことになります。

「Provisioning Profile」ビルド設定は開発用プロファイル、「Code Signing Identity」ビルド設定は開発用証明書に施します。

コード署名IDとして開発用証明書を設定するには

1. Xcodeのプロジェクトエディタ上でターゲットを選択します。

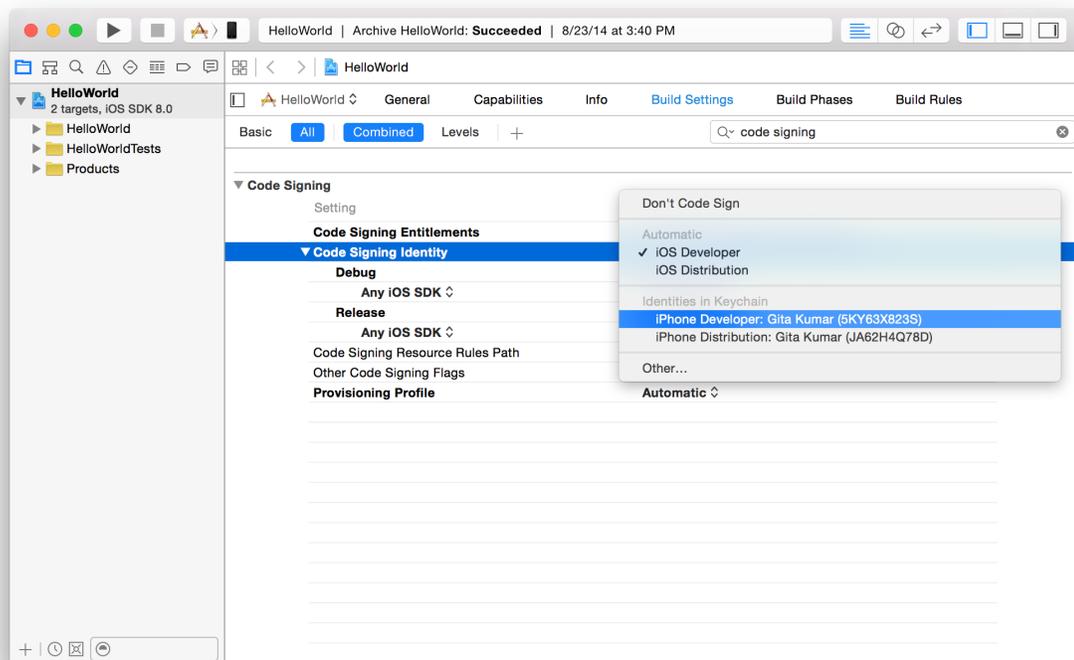
Important: どのターゲットも同じコード署名IDを使って署名する場合は、個々のターゲットではなくプロジェクトを選択してください。

2. 「Build Settings」ボタンを押します。
3. 「Build Settings」ペインで、「All」を選択し、検索欄に「code signing」と入力してください。
4. 「Provisioning Profile」ポップアップメニューから、該当する開発用プロビジョニングプロファイルを選択してください。

Xcodeにより、自動的に「Code Signing Identity」ビルド設定が、iOSアプリケーションの場合は「iOS Developer」、Macアプリケーションの場合は「Mac Developer」に設定されます。

5. 必要ならば「Code Signing Identity」ポップアップメニューから開発用証明書を選択してください。

iOSアプリケーションの場合、「Provisioning Profile」メニューから、「iPhone Developer:」の後に自分の名前が続く形の証明書を選択します。Macアプリケーションであれば、「Provisioning Profile」メニューから、「Mac Developer:」の後に自分の名前が続く形の証明書になります。



次にアプリケーションをビルドする際に、コード署名が施されます。Macアプリケーションは、「Run」ボタンを押すだけでビルド、実行できます。iOSアプリケーションの場合は、「[iOS/tvOS/watchOSアプリケーションを起動する](#)」（92 ページ）に示す手順に従って署名を施し、デバイス上で実行してください。

後でまたチームプロビジョニングプロファイルを使うときには、「Provisioning Profile」ビルド設定を「None」に変更します。Appleのコード署名技術について詳しくは、『[Code Signing Guide](#)』を参照してください。

トラブルシューティング

開発用プロビジョニングプロファイルが「Provisioning Profile」ポップアップメニューに表示されない場合は、プロビジョニングプロファイルをダウンロードしてください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215ページ）を参照）。その後、「Provisioning Profile」ビルド設定を再試行します。

アプリケーションのビルド時にコード署名のエラーが発生する場合は、ビルド設定「Code Signing Identity」が適切かどうか確認してください。これがプロジェクトごとの設定か、ターゲットごとの設定かも確認します（ターゲットごとの設定が優先）。ビルド設定「Code Signing Identity」に関する障害が発生したときは、[“ビルドおよびコード署名に関する問題”](#)（254ページ）も参照してください。

Xcodeで管理されるプロビジョニングプロファイルを移行する

Xcodeにより、コード署名IDおよびプロビジョニングプロファイルの作成と更新が自動的に行われます。ただし、Xcode 4以前で作成したプロジェクトの場合は、「Provisioning Profile」ビルド設定を「Automatic」に設定して、この機能を使用できるようにする必要があります。

Xcodeで管理されるプロビジョニングプロファイルを使用するには

1. Xcodeのプロジェクトエディタ上でターゲットを選択します。
2. 「Build Settings」ボタンを押します。
3. 「Build Settings」ペインで、「All」を選択し、検索欄に「Code Signing」と入力してください。
4. 「Provisioning Profile」ポップアップメニューから「Automatic」を選択します。

Xcodeにより、最上位の「Code Signing Identity」ビルド設定が「Don't Code Sign」に設定され、「Any ... SDK」の「Code Signing Identity」ビルド設定が、iOSアプリケーションの場合は「iOS Developer」、Macアプリケーションの場合は「Mac Developer」に設定されます。

必要に応じて、プロジェクトレベルの「Provisioning Profile」ビルド設定を「Automatic」に設定します。プロビジョニングプロファイルのトラブルシューティングを実行する場合は、[“プロビジョニングに関する問題”](#)（253ページ）を参照してください。

メンバーセンターでプロビジョニングプロファイルを編集する

プロビジョニングプロファイルの内容を変更したい場合でも、再生成する必要はありません。プロビジョニングプロファイルの種類に応じ、名前その他のプロパティを変更できるのです。種類にかかわらず、App IDは変更可能です。また、iOS/tvOS/watchOSアプリケーションであれば、アドホックプロビジョニングプロファイルにデバイスを追加できます。変更したときは、デバイスに組み込んだプロビジョニングプロファイルも忘れずに置き換えてください。

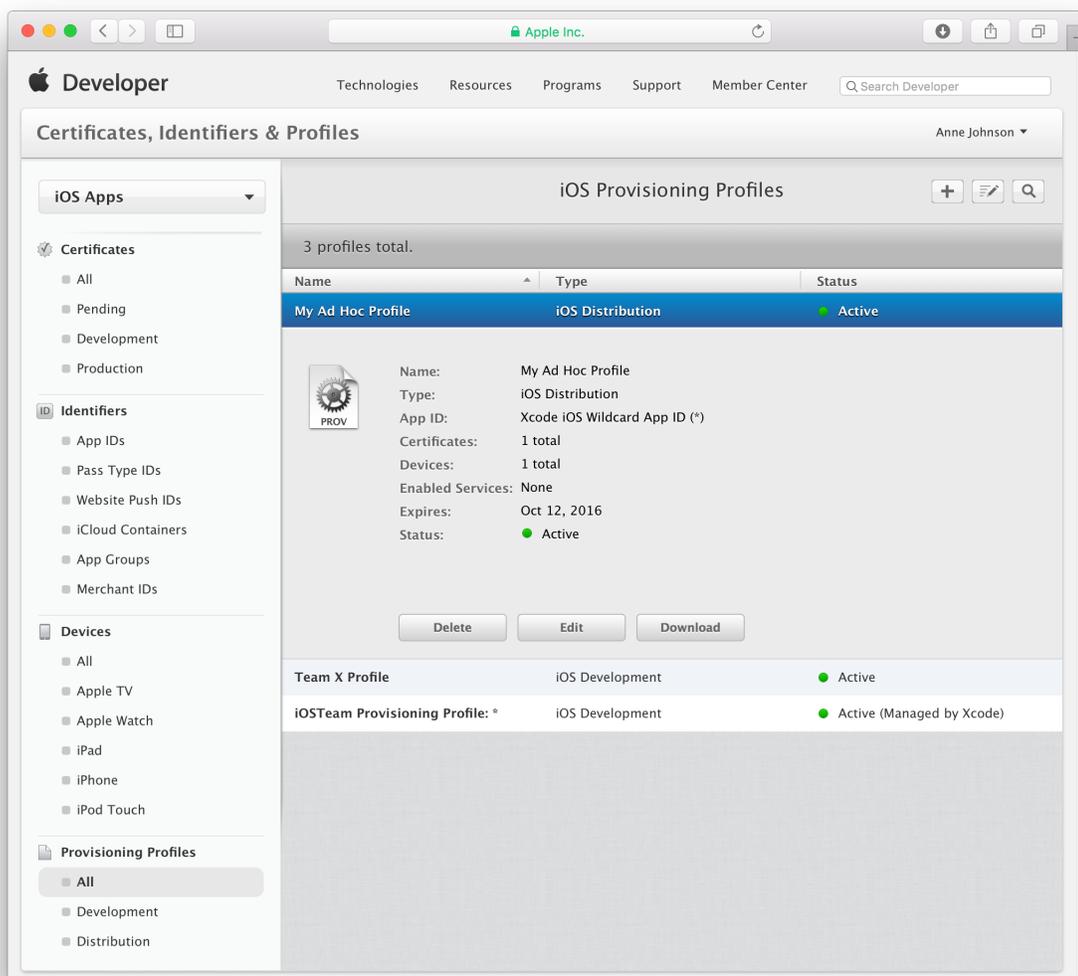
証明書を再生成したため、プロビジョニングプロファイルを修復しなければならない場合は、[“証明書を再生成し、関係するプロビジョニングプロファイルを更新する”](#)（189ページ）に示す手順で操作してください。

注意: Xcodeが管理するチームプロビジョニングプロファイルを変更することはできません。

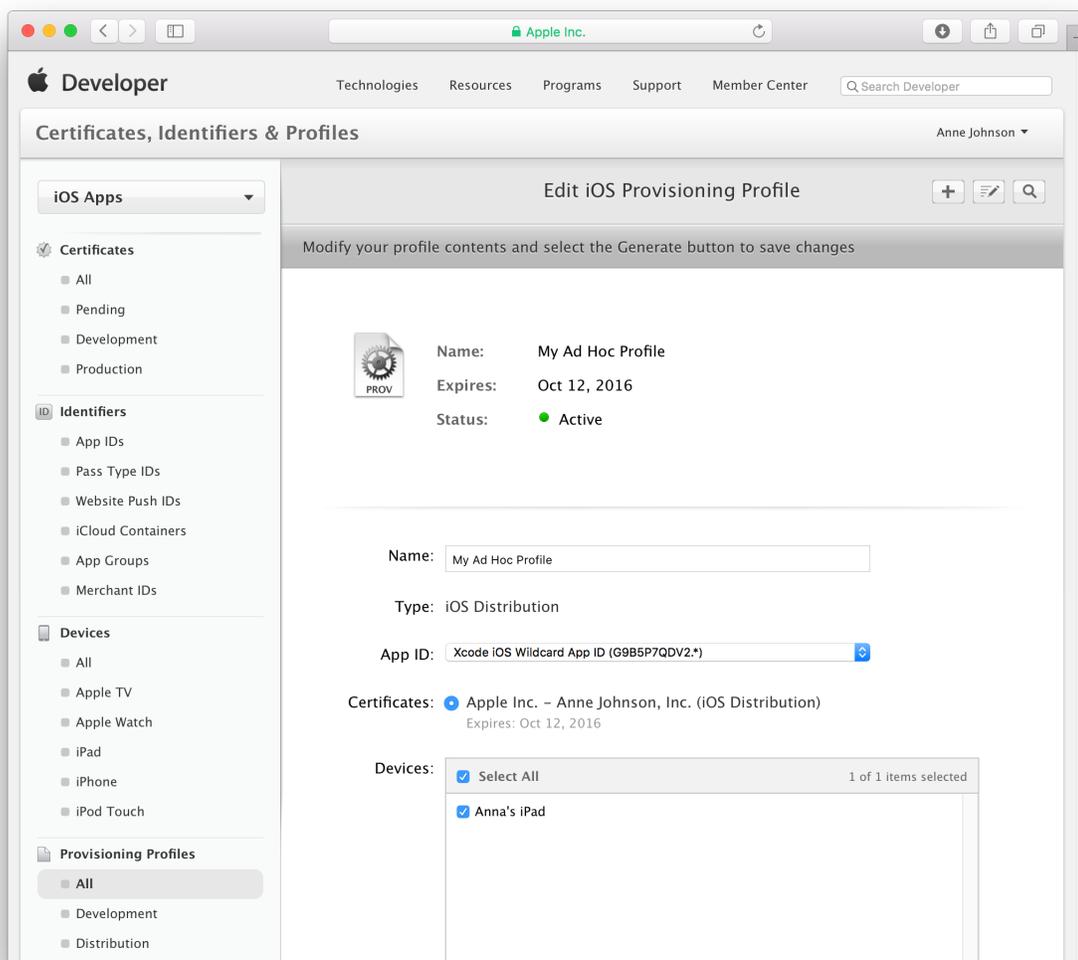
プロビジョニングプロファイルを編集するには

1. 「[Certificates, Identifiers & Profiles](#)」の画面で「Provisioning Profiles」を選択します。
2. 「Provisioning Profiles」で「All」、「Development」、「Distribution」のいずれかを選択します。

3. 変更を施すプロビジョニングプロファイルを選択し、「Edit」を押してください。



- 名前を修正する、証明書を追加する、別のデバイス群を登録するなど、必要な変更を施します。



- 「Generate」を押します。

以上が終了したら、Xcode上でプロビジョニングプロファイルをダウンロードしてください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215 ページ）を参照）。

期限切れになったプロビジョニングプロファイルを更新する

プロビジョニングプロファイルが期限切れになると、メンバーセンターには状態が「Expired」と表示されます。これは、[“メンバーセンターでプロビジョニングプロファイルを編集する”](#)（220 ページ）の手順で編集、再生成することにより、期限を延ばすことができます。プロビジョニングプロファイル自体に変更を施す必要はありません。「Edit iOS Provisioning Profile」ページまたは「Edit Mac Provisioning Profile」ページの末尾までスクロールして「Generate」をクリックするだけです。

期限切れになったプロビジョニングプロファイルがデバイスにインストールされている場合は、削除してください（“[デバイス上でプロビジョニングプロファイルを確認して削除する](#)”（223 ページ）を参照）。アドホックプロビジョニングプロファイルに置き換えた場合は、これを使ってアプリケーションに署名し直し、再配布してください（“[テスト用にアプリケーションをエクスポート（iOS、tvOS、watchOS）](#)”（102 ページ）を参照）。

デバイス上でプロビジョニングプロファイルを確認して削除する

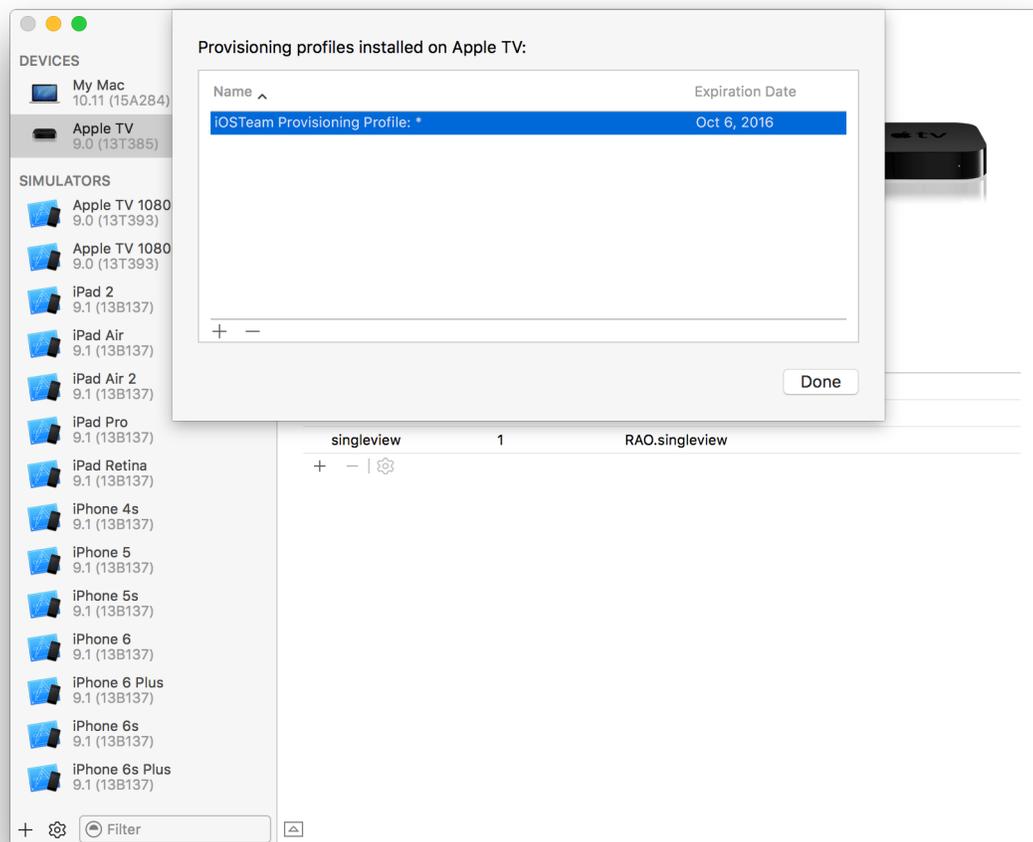
Xcodeを使って、プロビジョニングプロファイルがデバイスにインストールされているかどうかを確認するか、期限切れになっているプロビジョニングプロファイルをデバイスから削除します。

注意: プロビジョニングプロファイルを自分でインストールすることはほとんどありません。デバイス上でアプリケーションを起動する際、iOSやOS Xが自動的に、バンドルに収容されているプロビジョニングプロファイルをインストールするからです。

プロビジョニングプロファイルを確認または削除するには

1. iOS/tvOSアプリケーションの場合、デバイスをMacに接続します。watchOSアプリケーションについては、Apple WatchとペアリングしたiPhoneを接続します。
2. 「Window」 > 「Devices」の順に選択した後、「Devices」から該当するデバイスを選択します。
3. 左側の列の一番下にある「Action」ボタン（「Add」ボタンの右にある歯車アイコン）をクリックします。
4. ポップアップメニューから「Show Provisioning Profiles」を選択します。
プロビジョニングプロファイルシートが表示されます。

5. プロビジョニングプロファイルを削除するには、リストから該当するプロビジョニングプロファイルを選択して、左下隅にある「Delete」ボタン（「-」の表示）をクリックします。



6. 「Done」をクリックします。

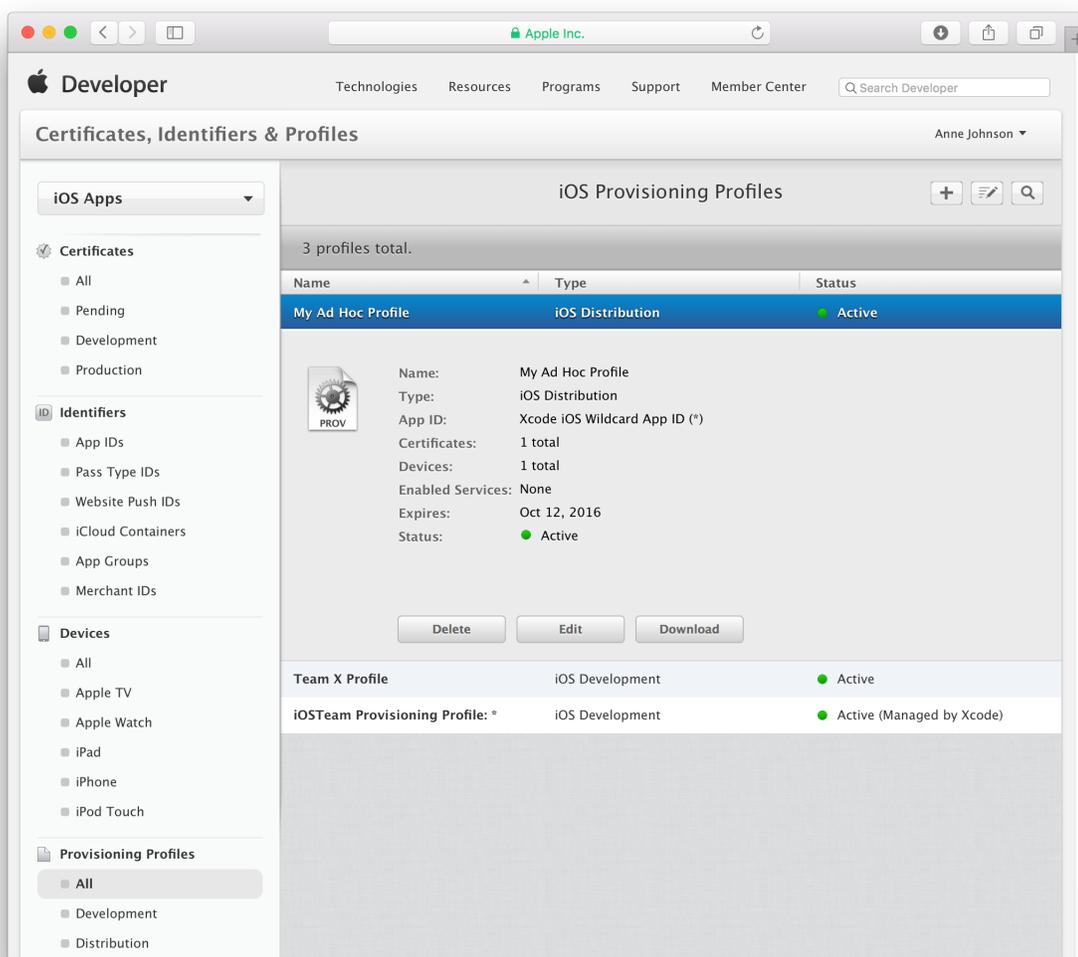
プロビジョニングプロファイルをメンバーセンターから削除する

プロビジョニングプロファイルをチームから削除する必要があることもあります。

プロビジョニングプロファイルをチームから削除するには

1. 「Certificates, Identifiers & Profiles」の画面で「Provisioning Profiles」を選択します。
2. 「Provisioning Profiles」以下の「All」を選択します。
3. 削除しようとするプロビジョニングプロファイルを選択します。

- 「Delete」を押してください。



- 表示される確認ダイアログで「削除」をクリックします。

以上が終了したら、Xcode上でプロビジョニングプロファイルをダウンロードし（“[Xcode上でプロビジョニングプロファイルをダウンロードする](#)”（215ページ）を参照）、さらにデバイスから削除してください（“[デバイス上でプロビジョニングプロファイルを確認して削除する](#)”（223ページ）を参照）。

プロビジョニングプロファイルをメンバーセンターからダウンロードする

必要ならば、メンバーセンターから直接、プロビジョニングプロファイルを指定してダウンロードできます。

プロビジョニングプロファイルをダウンロードするには

1. 「Certificates, Identifiers & Profiles」の画面で「Provisioning Profiles」を選択します。
2. 「Provisioning Profiles」以下の「All」を選択します。
3. 該当するプロビジョニングプロファイルを選択します。
4. 「Download」を押してください。

プロビジョニングプロファイルの拡張子を「.mobileprovision」に変更した名前のファイルが、「Downloads」フォルダに現れます。

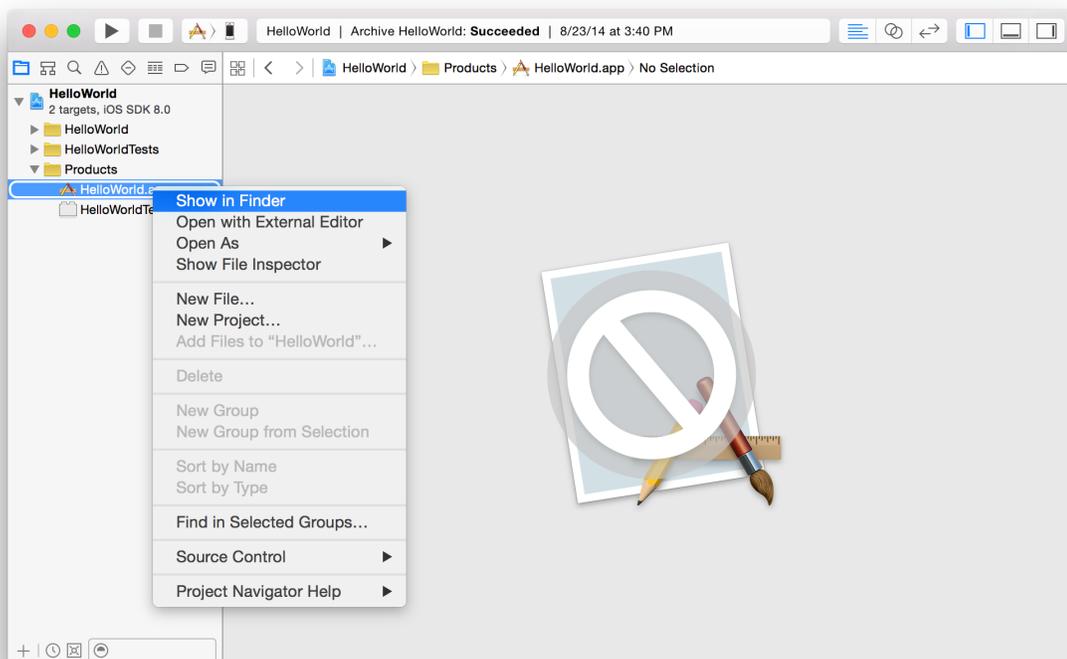
アプリケーションバイナリのエンタイトルメントを検証する

エンタイトルメントには、Xcodeプロジェクトで設定するもの（「App Sandbox」エンタイトルメントなど）と、プロビジョニングプロファイルで設定するものがあります。署名済みアプリケーションのエンタイトルメントが正しいことは、その署名を調べて確認できます。問題がある場合は、バイナリに埋め込まれているプロビジョニングプロファイルを調べてください。

署名済みアプリケーションのエンタイトルメントを確認するには

1. Xcodeのプロジェクトナビゲータで、プロジェクトを選択します。
2. プロジェクト名に添えられた詳細表示用の三角印をクリックして、その内容を表示します。
3. 「Products」の隣にある詳細表示用の三角印をクリックして、アプリケーションバイナリを表示します。

- Controlキーを押しながらアプリケーションのバイナリファイルをクリックし、ショートカットメニューから「Show in Finder」を選択して、Finder内のXcodeのビルドの場所に移動します。



- (/Applications/Utilitiesにある)「Terminal」を起動し、次のテキストとその後に空白を入力してください（まだReturnキーは押しません）。

```
codesign -d --entitlements -
```

- Finder上でアプリケーションバイナリを掴み、「Terminal」上にドラッグします。
- Returnキーを押してください。

たとえばiCloudキー値ストレージを有効にしたiOSアプリケーションの場合、その出力には「com.apple.developer.ubiquity-kvstore-identifier」というエンタイトルメントのキーが含まれています。同様に、App Sandboxを有効にしたMacアプリケーションの出力には、「com.apple.security.app-sandbox」というキーが入っています。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>application-identifier</key>
```

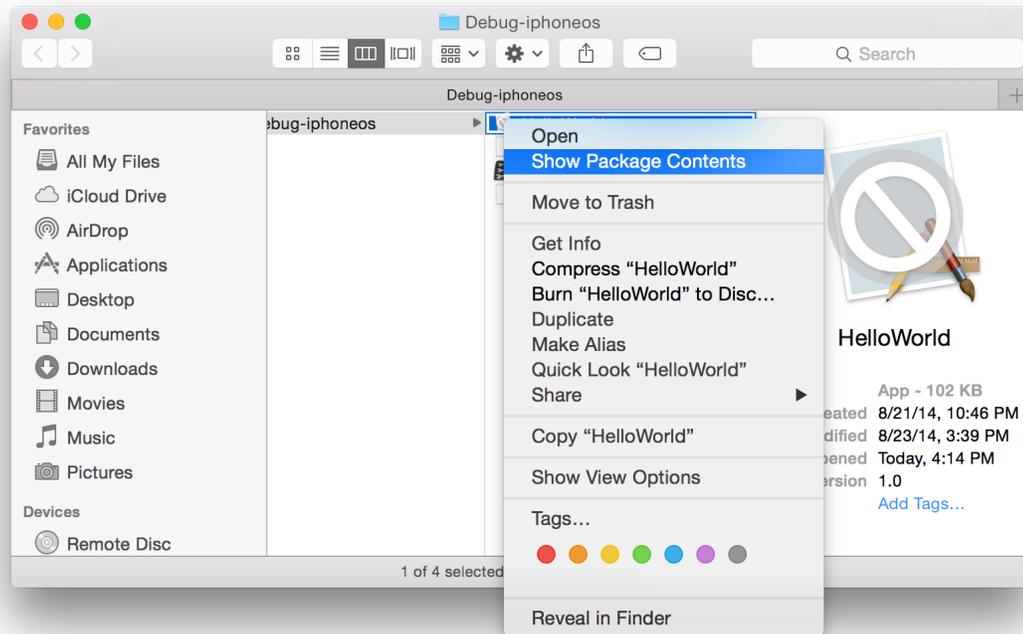
```
<string>G9B5P7QDV2.edu.self.HelloWorld</string>
<key>com.apple.developer.pass-type-identifiers</key>
<array>
  <string>G9B5P7QDV2.*</string>
</array>
<key>com.apple.developer.ubiquity-container-identifiers</key>
<array>
  <string>G9B5P7QDV2.edu.self.HelloWorld</string>
</array>
<key>com.apple.developer.ubiquity-kvstore-identifier</key>
<string>G9B5P7QDV2.edu.self.HelloWorld</string>
<key>get-task-allow</key>
<true/>
</dict>
</plist>
```

アプリケーションのエンタイトルメントが、自分が設定したはずのものと違っている場合は、正しいプロビジョニングプロファイルを埋め込んだかどうか確認してください。まず、埋め込んだプロビジョニングプロファイルを見つける必要があります。

アプリケーションバイナリに埋め込んだプロビジョニングプロファイルを見つけるには

1. Xcodeのプロジェクトナビゲータで、プロジェクトを選択します。
2. プロジェクト名に添えられた詳細表示用の三角印をクリックして、その内容を表示します。
3. 「Products」の隣にある詳細表示用の三角印をクリックして、アプリケーションバイナリを表示します。
4. Controlキーを押しながらアプリケーションのバイナリファイルをクリックし、ショートカットメニューから「Show in Finder」を選択して、Finder内のXcodeのビルドの場所に移動します。
5. Finder上で、Controlキーを押しながらアプリケーションのバイナリファイルをクリックし、ショートカットメニューから「Show Package Contents」を選択します。

iOSアプリケーションであれば、「embedded.mobileprovision」というプロビジョニングプロファイルがFinderのウィンドウに現れます。Macアプリケーションの場合は「embedded.provisionprofile」という名前です。

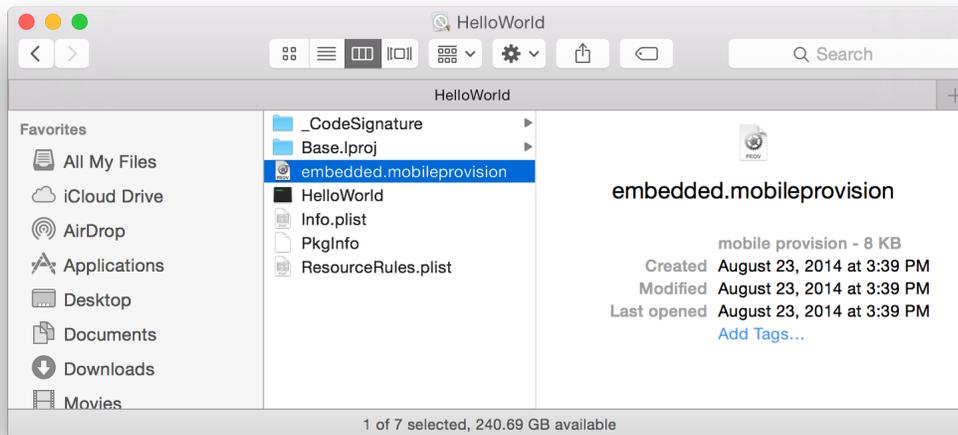


埋め込まれているプロビジョニングプロファイルのエンタイトルメントを確認するには

1. (/Applications/Utilitiesにある) 「Terminal」を起動し、次のテキストを入力してください (まだReturnキーは押しません)。

```
security cms -D -i
```

2. Finder上でアプリケーションバイナリ内のプロビジョニングプロファイルを掴み、「Terminal」にドラッグします。



3. Returnキーを押してください。

このコマンドの出力は、XML形式のプロパティリストです。

4. Entitlementsキーを見つけ、application-identifierキーの値が正しいエンタイトルメントであることを確認してください。

たとえば次に示すのは、データ保護、Wallet、iCloudを有効にしたiOSアプリケーションのもので、iCloudのエンタイトルメントは、先頭が「com.apple.developer.ubiquity」という文字列になっています。

```
<key>Entitlements</key>
  <dict>
    <key>application-identifier</key>
    <string>G9B5P7QDV2.*</string>
    <key>com.apple.developer.default-data-protection</key>
    <string>NSFileProtectionComplete</string>
    <key>com.apple.developer.pass-type-identifiers</key>
    <array>
      <string>G9B5P7QDV2.*</string>
    </array>
    <key>com.apple.developer.ubiquity-container-identifiers</key>
    <array>
      <string>G9B5P7QDV2.*</string>
    </array>
  </dict>
</key>
```

```
</array>
<key>com.apple.developer.ubiquity-kvstore-identifier</key>
<string>G9B5P7QDV2.*</string>
<key>get-task-allow</key>
<true/>
<key>inter-app-audio</key>
<true/>
<key>keychain-access-groups</key>
<array>
  <string>G9B5P7QDV2.*</string>
</array>
</dict>
```

上記コマンドの他の使い方については、`codesign`および`security`を参照してください。

まとめ

この章では、証明書やプロビジョニングプロファイルを有効な状態に維持する方法、不要になったアセットを削除する手順を説明しました。証明書やプロビジョニングプロファイルに関して問題が生じた場合は、“[トラブルシューティング](#)”（251 ページ）を参照してください。

Apple Developer Enterprise Program アプリケーションの配布

Apple Developer Enterprise Program を利用すると、独自の社内向け iOS/watchOS アプリケーションを開発して、従業員に、ストアを通すことなく配布できます。

Apple Developer Enterprise Program に加入していない場合は、[Apple Developer Enterprise Program](#) にアクセスして、加入手続きができます。

注意: Apple Developer Enterprise Program の会員は、Developer ID 証明書を生成して Mac アプリケーションを配布することもできます ([“Mac App Store を通さずにアプリケーションを配布する手順”](#) (241 ページ) を参照)。

Apple Developer Enterprise Program アプリケーションを開発する

Apple Developer Enterprise Program アプリケーションを開発するワークフローは、大規模な組織がストア向けに数多くのアプリケーションを開発する際に使っているワークフローとよく似ています。開発の期間中、アセットの管理は Xcode に任せ、必要な場合にだけメンバーセンターを利用します。Xcode により、App ID が生成され、有効にするアプリケーションサービスを確実に使用するようにプロジェクトが設定されます。

チームを結成する (チームエージェント)

チームエージェント (Apple Developer Enterprise Program の加入者) の場合は、まずチームメンバーを招待し、それぞれの役割を決めて、チームを結成します。チームの管理に携わる人にチーム管理者の役割を与えます。アプリケーションの開発には携わるが、アプリケーションを配布する権限のない人には、チームメンバーの役割を与えます。チームエージェントとチーム管理者には、ストアを通さずに配布できるよう、アプリケーションをエクスポートする責任があります。チームエージェントが実行できる作業の大部分は、チーム管理者が代行できます。最初の段階では、チームエージェントがただ1人のチームメンバーです。他のメンバーを招待する方法については、[“チームメンバーを加えて役割を与える”](#) (162 ページ) を参照してください。

チームを管理する方法については、[“メンバーセンターでのチームの管理”](#) (160 ページ) を参照してください。

共有チームプロビジョニングプロファイルを作成する (チーム管理者)

チーム管理者は、以下の手順に従って、開発チームを有効にする必要があります。

1. チームプロビジョニングプロファイルを作成して、技術やサービスを有効にします。
チームプロビジョニングプロファイルを作成して、技術やサービスを有効にするための手順は、個人の開発者の場合と同じです (“[アプリケーションを配布するためのXcodeプロジェクトの設定](#)” (30 ページ) および “[技術やサービスを組み込む](#)” (56 ページ) を参照)。Xcodeにより、適切なApp ID とプロビジョニングプロファイルが自動的に作成されます。APNを使用する場合は、“[プッシュ通知の設定をする \(iOS、watchOS、Mac\)](#)” (65 ページ) で追加手順を確認してください。エンタープライズアプリケーションで利用できるケイパビリティについては、“[サポートされるケイパビリティ](#)” (258 ページ) を参照してください。
2. ここでテストデバイスを登録することも可能です。
Xcodeは自動的に、Macに接続されたデバイスや、チームメンバーがツールバーの「Scheme」メニューから選択したデバイスを登録します。

Important: チームメンバーにはそれぞれ自分専用のApple IDと開発用証明書が必要です。チームメンバー間で署名IDを共有してはなりません。

チーム管理者は、必要に応じてチームメンバーをチームから外すこともできます。アプリケーションの開発を請負業者に委託している場合は、チームに招き入れてチームメンバーの役割を与えます。請負契約が満了になった時点で、チームから外します (“[チームメンバーを外す](#)” (167 ページ) を参照)。

開発を始める (チームメンバー)

チームメンバーは、以下の手順に従って、開発を始めます。

1. Xcodeの「Accounts」環境設定画面に自分のApple IDを追加します (“[Xcodeで自分のApple IDアカウントを追加する](#)” (26 ページ) を参照)。
2. 開発用証明書を作成し、デバイスを登録する手順については、“[アプリケーションを配布するためのXcodeプロジェクトの設定](#)” (30 ページ) を参照してください。

Apple Developer Enterprise Programの会員であるチームを、「Team」ポップアップメニューから選択してください。Xcodeは必要に応じて、チームプロビジョニングプロファイルを再生成し、ダウンロードします。

Apple Developer Enterprise Program アプリケーションをテストする

ベータテストを実施するためにXcodeからApple Developer Enterprise Program アプリケーションをエクスポートする場合は、アドホックプロビジョニングプロファイルを使用します（“[アプリケーションをアーカイブする](#)”（103ページ）を参照）。チームエージェントとチーム管理者だけがアドホックプロビジョニングプロファイルを配布用に作成できます。アプリケーションのエクスポートが終了したら、Xcodeサーバを利用して、そのアプリケーションをテスト担当者とはほかのチームメンバーに配布することを検討してください。Xcodeサービスの使用方法について詳しくは、『*Xcode Continuous Integration Guide*』を参照してください。

Important: Apple Developer Enterprise Programの会員には、iTunes Connectアカウントも、iTunes Connectを使ってベータテスト用にアプリケーションを配布する権限もありません。

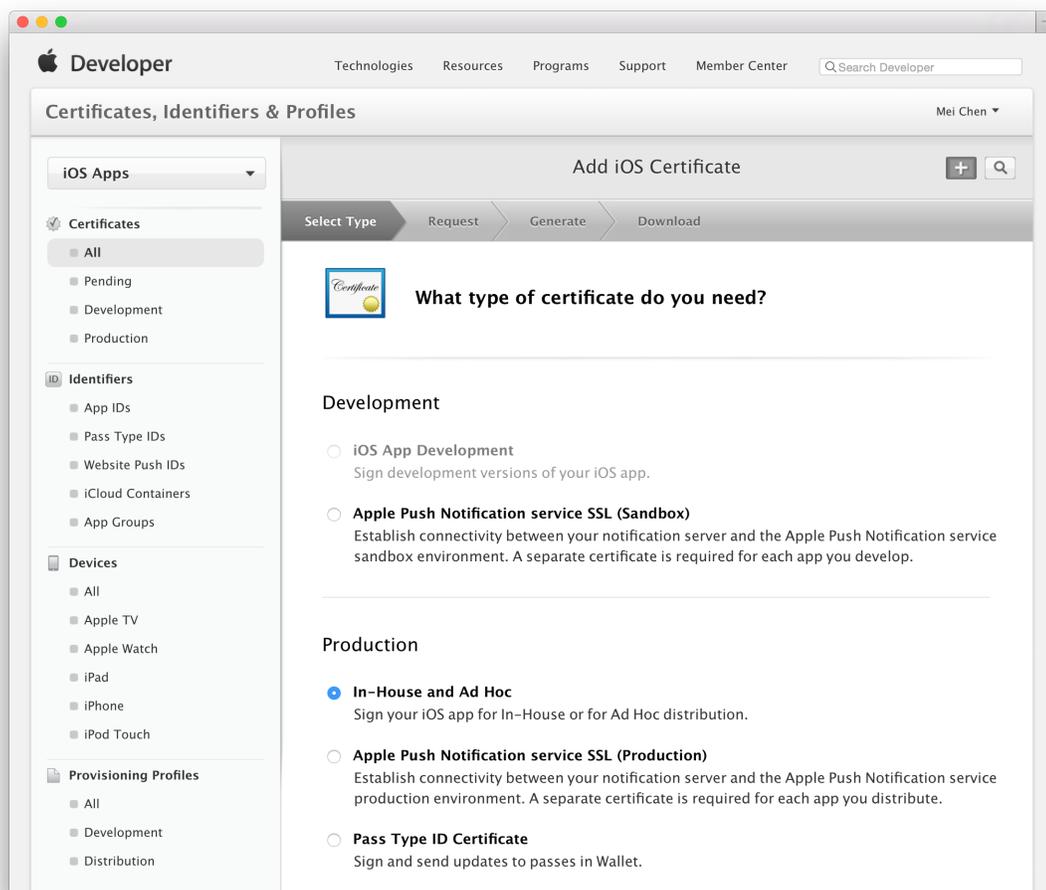
別のエンタープライズ配布用証明書を作成する

Apple Developer Enterprise Programの会員は、複数の配布用証明書を作成することが許可されています。

別の配布用証明書を作成するには

1. 「[Certificates, Identifiers & Profiles](#)」の画面で「Certificates」を選択します。
2. 右上隅にある「Add」ボタン（「+」の表示）を押します。

3. 「Production」で「In-House and Ad Hoc」を選択して「Continue」をクリックします。



4. 表示される手順に従い、Keychain Accessを使って証明書署名要求（CSR、Certificate Signing Request）を作成し、「Continue」を押します。
5. 「Choose File」を押してください。
6. CSRファイル（拡張子「.certSigningRequest」）を選択し、「Choose」を押します。
7. 「Generate」を押します。
8. 「Download」を押してください。

証明書ファイルがDownloadsフォルダに表示されます。

証明書は、そのファイル（拡張子「.cer」）をダブルクリックするだけでキーチェーンに登録できます。この配布用証明書は、Keychain Accessの「My Certificates」カテゴリに表示されます。

証明書とプロビジョニングプロファイルの有効期限を管理する

チームの証明書とプロビジョニングプロファイルを管理する責任があります。Apple Developer Enterprise Programの証明書は3年後に期限切れになり、プロビジョニングプロファイルは1年後に期限切れになります。

配布用証明書が期限切れになる前に、別の配布用証明書を作成します（[“別のエンタープライズ配布用証明書を作成する”](#)（234 ページ）を参照）。期限切れになった証明書を更新することはできません。その代わりに、期限切れになった証明書を新しい証明書に置き換えることができます（[“期限切れになった証明書を更新する”](#)（189 ページ）を参照）。

配布用プロビジョニングプロファイルが期限切れになった場合は、有効な配布用証明書があることを確認した上で、プロビジョニングプロファイルを更新します（[“期限切れになったプロビジョニングプロファイルを更新する”](#)（222 ページ）を参照）。

Xcodeで開発用証明書とチームプロビジョニングプロファイルが管理されます。

社内でアプリケーションを配布する

ストアを通さずに従業員への配布を目的としてアプリケーションをエクスポートするには、以下の手順を実行してください。

1. アプリケーションをアーカイブする。
2. アーカイブをiOS Appファイル（拡張子は「.ipa」）としてエクスポートします。
3. MDMを使ってiOS Appファイルを配布します。

アーカイブを作成する

アプリケーションを配布する際は、その方法にかかわらず、アーカイブを作成する必要があります。Xcodeのアーカイブ機能を利用すれば、アプリケーションをビルドし、重要なデバッグ情報とともに、Xcodeが管理するバンドルに格納することができます。

Important: 配布するアーカイブは、すべての版を保存しておいてください。後でクラッシュレポートを分析する際、アーカイブ内のデバッグ情報が必要になります。

アーカイブを作成するには

1. Xcodeプロジェクトエディタのツールバーにある「Scheme」メニューから、総称デバイス名またはデバイス名を選択します。

シミュレータ用に構築したアプリケーションのアーカイブは作成できません。デバイスがMacに接続されていれば、その名前がツールバーの「Scheme」メニューに現れます。それ以外の場合は総称デバイスを選択してください。

2. 「Product」 > 「Archive」 コマンドを実行します。

「Archives」 オーガナイザが開き、新しいアーカイブが表示されます。

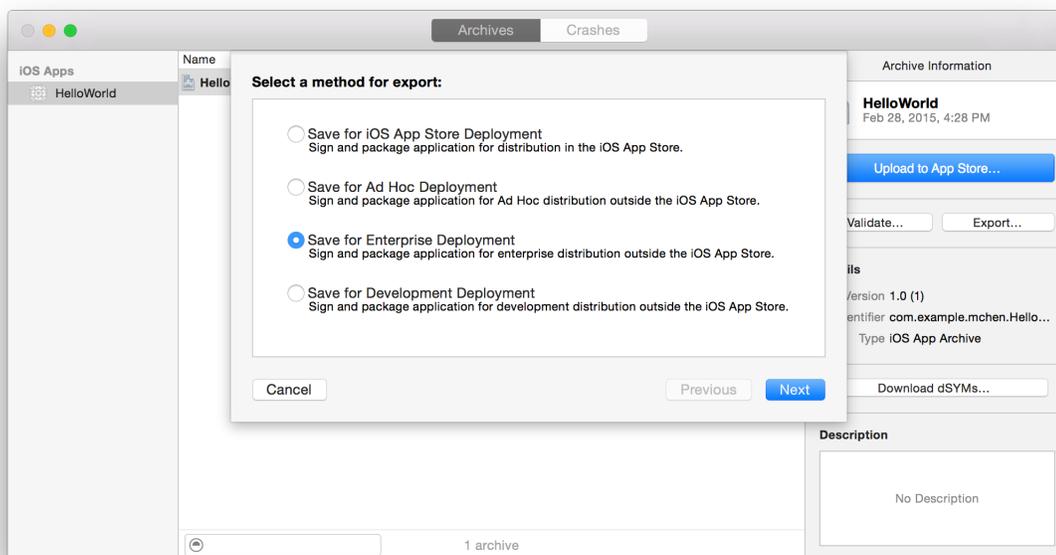
Xcodeはアーカイブに対して予備検証テストを実行します。その結果、プロジェクトエディタに警告が表示されることがあります。たとえば、必要なアプリケーションアイコンを設定していないと（[“アプリケーションアイコンと起動画像ファイルを個別に設定する”](#)（50 ページ）を参照）、「Info.plist」に関する警告メッセージが表示されます。この場合は、問題を解消した後、アーカイブをもう一度作成してください。

iOS App ファイルを作成する

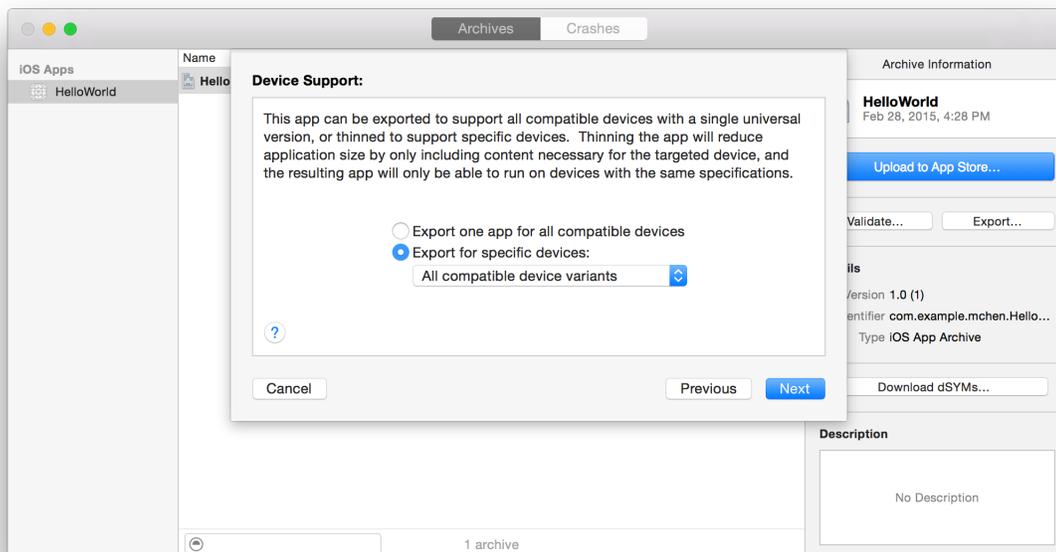
iOS App ファイルを作成するのは、ユーザが自分のデバイスにアプリケーションをインストールできるようにするためです。iOS App ファイル（拡張子は「.ipa」）はアーカイブから生成します。アプリケーションをエクスポートすると、Xcodeにより、必要な配布用証明書とプロビジョニングプロファイルが自動的に作成されます。

配布用のiOS App ファイルを作成するには

1. 「Archives」 オーガナイザで、該当するアーカイブを選択します。
2. 会員でない場合は、「Export」 ボタンをクリックし、「Save for Enterprise Deployment」を選択して「Next」をクリックします。



- 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。
必要に応じて、Xcodeで配布用証明書、プロビジョニングプロファイル、特定App IDが作成されます。
- 「Device Support」ダイアログで、エクスポート対象（ユニバーサルアプリケーション、または特定デバイス用の変種）を選択し、「Next」を押してください。
 - サポート対象のあらゆるデバイス上で動作するようにしたければ、「Export one app for all compatible devices」を選択します。
 - あらゆるデバイスの変種でテストしたければ、「Export for specific devices」を選択し、次いでポップアップメニューから「All compatible device variants」を選択してください。
 - 特定デバイス向けの変種をテストしたい場合は「Export a thinned app for a specific device」を選択し、次に該当するデバイスファミリーをポップアップメニューから選択します。



- 表示されるダイアログで、アプリケーション、エンタイトルメント、プロビジョニングプロファイルを確認します。
ワイルドカードAppIDを使ってエンタープライズアプリケーションを配布することはできません。配布用プロビジョニングプロファイルの名前は、「XC:」で始まり、その後にAppIDが続きます。ワイルドカードAppIDを使っている場合、配布用プロビジョニングプロファイルの名前は「XC:*」になります。
Finder上に、エクスポートしたファイル（拡張子「.ipa」）が現れます。
- ビルドオプションを確認し、「Next」を押してください。

オンデマンド型リソースを利用する場合は「Include manifest for over-the-air installation」をオンにします。マニフェストファイルはXML形式のplistで、デバイスがウェブサーバからアプリケーションを検索、ダウンロードし、インストールするために使います。

- マニフェストファイルを要求する場合、「Distribution manifest information」ダイアログが現れるので、ウェブサーバの詳細を入力してください。

下記の情報を入力します。

- **Name**。ダウンロードとインストール中に表示されるアプリケーション名。
- **App URL**。iOS Appファイルの完全修飾URL（HTTPSスキーム）。
- **Display Image URL**。ダウンロード/インストール中に現れるアプリケーションアイコンの完全修飾URL（HTTPSスキーム）。画像ファイルは57x57ピクセルのPNG形式でなければなりません。
- **Full Size Image URL**。iTunesに表示される大きな画像の完全修飾URL（HTTPSスキーム）。画像ファイルは512x512ピクセルのPNG形式でなければなりません。

- 「Export」を押してください。

Finder上にエクスポートしたファイルが現れます。

サーバツールを詳しく把握する

多くのソフトウェア開発チームを支援するXcodeサーバツールを有効に活用します。

詳しく把握する内容	参照先
継続的な統合 OSX Server上で実行されるXcodeサービスを利用して、アプリケーションのビルド、分析、テスト、アーカイブを自動化します。	<i>Xcode Continuous Integration Guide</i>
自動テスト Xcodeサービスで実行できるテストを準備します。	<i>Testing with Xcode</i>
サーバ上のホストソース制御リポジトリ Xcodeサービスを利用してリモートリポジトリに接続します。	<i>Xcode Continuous Integration Guide</i> <i>Source Control Management Help</i>
モバイルデバイス管理 MDMを使って業務用/教育用アプリケーションを配備します。	http://www.apple.com/support/iphone/enterprise/ にアクセスし、左側列の「Mobile Device Management」をクリックしてください。

まとめ

この章では、Apple Developer Enterprise Program の会員が実行できるさまざまな開発手順と配布手順を説明しました。また、チームを結成する方法、共有チームアセットを作成する方法、テストの目的でアプリケーションをエクスポートする方法、ストアを通さずに配布する目的でアプリケーションをエクスポートする方法も説明しました。

Mac App Storeを通さずにアプリケーションを配布する手順

Mac App Storeを通さずにアプリケーションを配布したいこともあります。その場合、Developer ID証明書を使って、Appleが開発者を認識している旨、ユーザが確認できるようにしてください。

Macユーザはセキュリティ保全のため、Gatekeeperを活用していることがあります。Mac App Storeを通して配布され、または身許が明らかな開発者によるソフトウェアしか、インストールできないようにしているのです。Appleが発行したDeveloper ID証明書を使った署名のないアプリケーションは、Gatekeeperが有効になっているMac上では起動できません。このような状況を回避するため、Developer ID証明書を使って、作成したアプリケーションや、プラグイン、インストールパッケージに署名してください。さらに、Gatekeeperが有効になっているMacでの動作を、十分にテストした上で配布する必要があります。

この章では、Mac App Storeを通さずに配布する場合に、Developer IDで署名したアプリケーションをXcode上で開発およびテストする手順を説明します。

Developer IDを使って署名したアプリケーションやインストーラパッケージを作成する

Developer IDを使って署名したアプリケーションやインストーラパッケージを作成する手続きは、いくつかの手順に分かれます。まず、Mac App Storeを通さずに配布する旨をXcodeに通知した上で、Developer ID証明書を作成します。これには2種類あり、「Developer ID Application」はアプリケーション、「Developer ID Installer」はインストーラパッケージに署名するために使います。Xcode上で、「Developer ID Application」証明書を使って、アプリケーションのアーカイブをエクスポートし、署名を施します。コマンドラインユーティリティで、「Developer ID Installer」証明書を使ってインストーラパッケージに署名することも可能です。

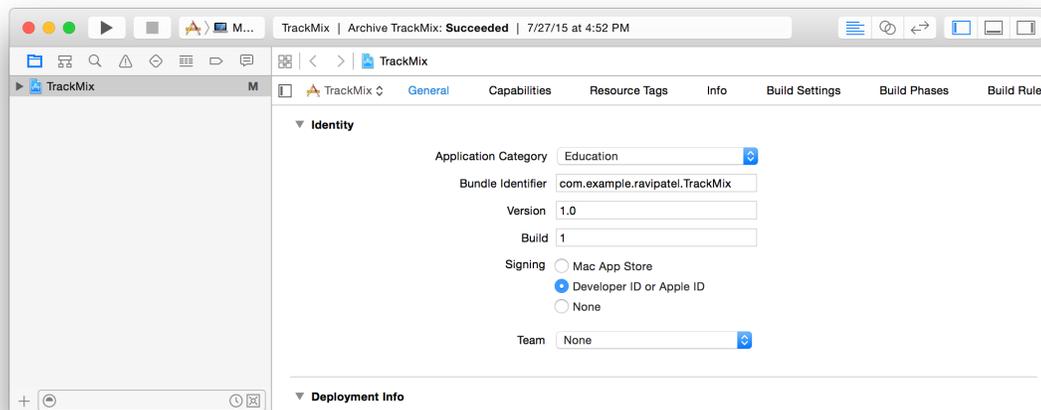
Important: 事前にApple Developer Programに加入しておいてください（“[Xcodeで自分のApple IDアカウントを追加する](#)”（26 ページ）を参照）。Developer ID証明書を作成し、これを使ってアプリケーションやインストーラパッケージに署名できるのは、Apple Developer Programに加入しているチームエージェントに限ります。

署名IDとしてDeveloper IDを設定する

最初に、「General」ペインで、署名IDとしてDeveloper IDを設定します。

署名IDとしてDeveloper IDを設定するには

1. プロジェクトナビゲータでターゲットを選択し、プロジェクトエディタを表示します。
2. 「General」を押して画面を切り替え、必要ならば「Identity」に添えられた詳細表示用の三角印をクリックして、設定事項を表示します。
3. バンドルIDが一意的であることを確認してください。
4. 「Signing」以下で、署名IDとして「Developer ID or Apple ID」を選択します。



5. 必要ならば、「Team」ポップアップメニューからチームを選択するか、「Add an Account」を実行してください。

Mac App Storeを通さずに配布する場合、一部のアプリケーションサービスは使えません。「Capabilities」ペインでいずれかを有効にすると（[“技術やサービスを組み込む”](#)（56ページ）を参照）、「Signing」ラジオボタンの設定は「Mac App Store」に戻ってしまいます。Apple IDをXcodeに追加し、Apple Developer Programに加入する場合は、[“アカウントの管理”](#)（24ページ）を参照してください。

Developer ID証明書を作成する

Mac App Storeを通さずにアプリケーションを配布する場合、名前が「Developer ID」で始まる署名証明書を使います。「Accounts」環境設定画面で、Developer ID証明書を特定して作成することも可能です（[“署名IDを作成する”](#)（172ページ）を参照）。「Accounts」環境設定で、署名ID「Developer ID Application」または「Developer ID Installer」に添えられた「Create」ボタンを押してください。Developer ID証明書があるのに秘密鍵がない場合も、別にDeveloper ID証明書を作成します（[“追加のDeveloper ID証明書を作成する”](#)（177ページ）を参照）。この証明書を使うためには、「Developer ID Certification Authority」という中間証明書も必要です（Xcodeが自動的にキーチェーンに登録）。この中間証明書がない場合は、[“誤って削除した中間証明書を登録する”](#)（179ページ）に従って復元してください。

Developer ID署名IDを生成したら、すぐにそのバックアップを取っておくことをお勧めします（“[開発者プロフィールをエクスポートする](#)”（181 ページ）を参照）。

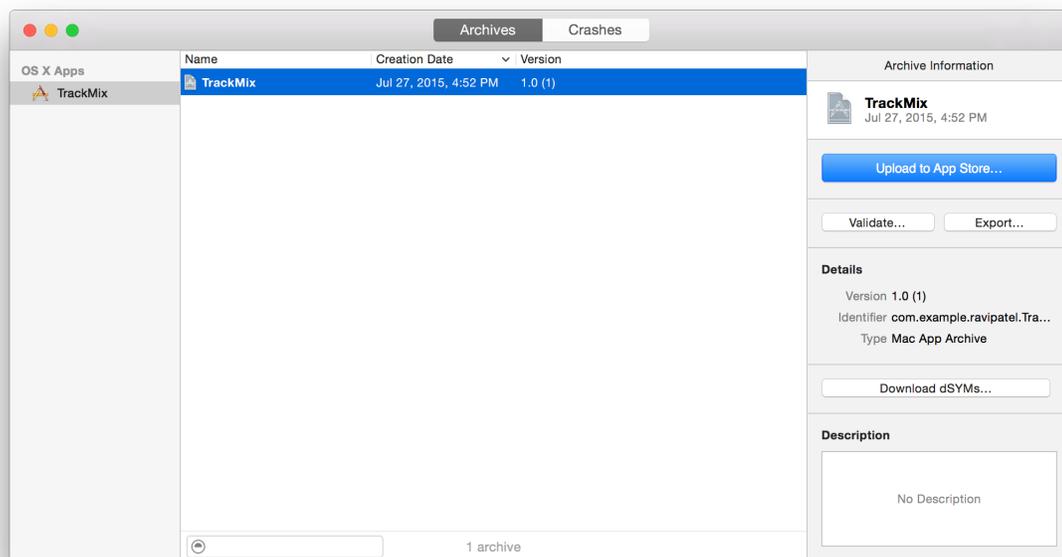
Important: Developer ID証明書を作成できるのはチームエージェントに限ります。個人開発者であれば、チームエージェントでもあるので、自分でこの証明書を作成できます。Developer ID証明書を失効させたいときは、product-security@apple.comに連絡してください。

アーカイブを作成する

しかしその前に、アプリケーションを改めてビルド、実行し、確かに配布しようとする版であることを確認してください。

アーカイブを作成するには

1. Xcodeプロジェクトエディタで、該当するプロジェクトを選択してください。
2. 「Product」 > 「Archive」 コマンドを実行します。
「Archives」 オーガナイザが開き、新しいアーカイブが表示されます。



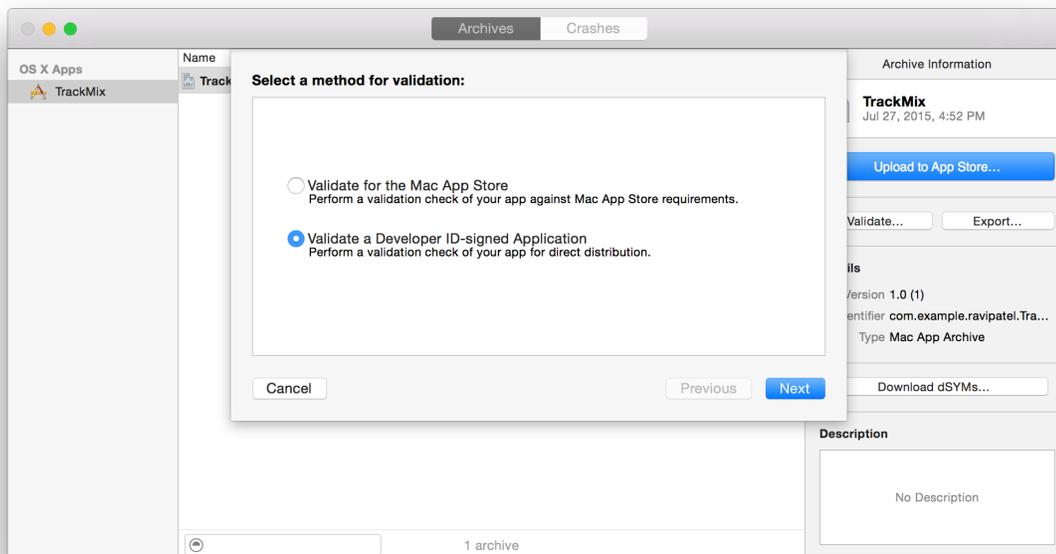
Developer IDで署名したアプリケーションを検証する

アーカイブを作成したらすぐに検証し、問題があった場合は解消しておきます。

Developer IDで署名したアーカイブを検証するには

1. 「Archives」 オーガナイザで、該当するアーカイブを選択して「Validate」 ボタンをクリックします。

- 表示されるダイアログで、「Validate a Developer ID-signed Application」を検証方法として選択し、「Next」をクリックします。



- 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。
- 署名IDとエンタイトルメントを確認して「Validate」をクリックします。
- 問題点が見つかった場合は確認した上で、「Done」をクリックします。

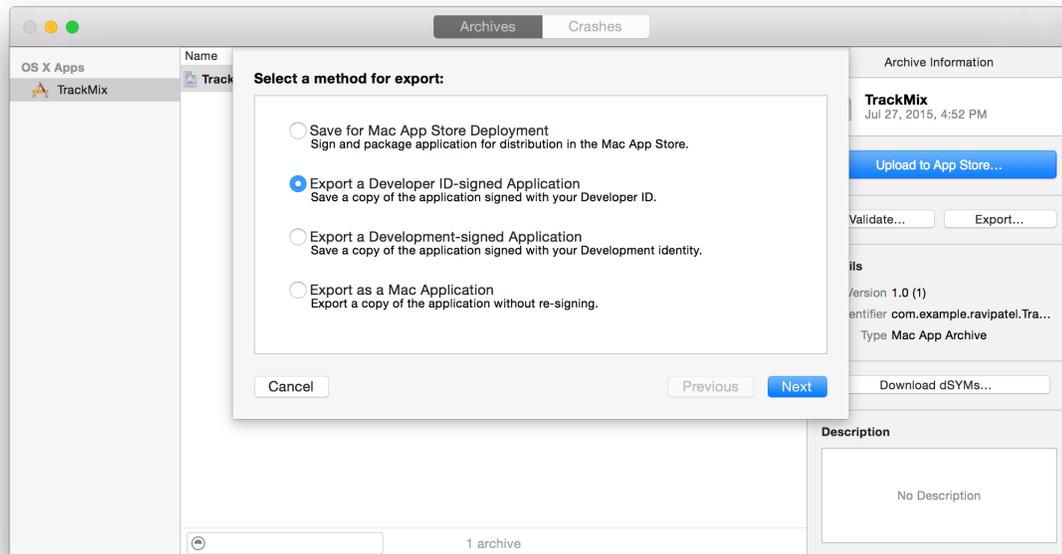
Developer IDで署名したアプリケーションをエクスポートする

Mac App Storeを通さずに配布できるようにアプリケーションをエクスポートするには、「Archives」オーガナイザを使います。

Developer IDで署名したアプリケーションを作成するには

- 「Archives」オーガナイザで、該当するアーカイブを選択して「Export」をクリックします。

2. 配布方法を選択できるダイアログが表示されるので、「Export a Developer ID-signed Application」を選択して「Next」をクリックします。



3. 表示されるダイアログで、ポップアップメニューからチームを選択して「Choose」をクリックします。
個人開発者であれば、メニューには本人の名前が現れます。そうでなければ組織名です。
4. 表示されるダイアログで、署名IDとエンタイトルメントを確認して「Export」をクリックします。
Finder上にエクスポートしたファイルが現れます。

インストーラパッケージに署名する

Mac App Storeを通さずに、インストーラパッケージの形でアプリケーションを配布したい場合、通常と同じようにパッケージを作成します。「Developer ID Installer」証明書を使い、「productsign」コマンドラインユーティリティでパッケージにコード署名を施します。「spctl」コマンドラインユーティリティで、インストーラパッケージをテストできます（「MyPackageName.pkg」の部分は実際のパッケージファイル名に置き換え）。

```
spctl -a -v --type install MyPackageName.pkg
```

 **Warning:** インストーラパッケージに署名する際は、「Developer ID Installer」証明書を使ってください。コマンドラインユーティリティ「productsign」を使えば、「Developer ID Application」証明書で署名することもできてしまいます。一見うまくいっているように見えますが、実際にMac上にインストールしようとしても失敗します。

開発の過程でコマンドラインからコード署名を施す場合は、『Code Signing Guide』を参照してください。

ここまでの結果を確認する

実際にアプリケーションを配布する前に、Gatekeeperを有効にした環境、無効にした環境で起動したときの動作をテストしてください。Gatekeeperの有効/無効の切り替えは「システム環境設定」で行います。Gatekeeperの検証やテストにも、コマンドラインユーティリティ「spctl」を使います。エンドユーザにとってどのように見えるかシミュレートするためには、アプリケーションを「隔離」し、Gatekeeperを有効にして再度テストしなければなりません。

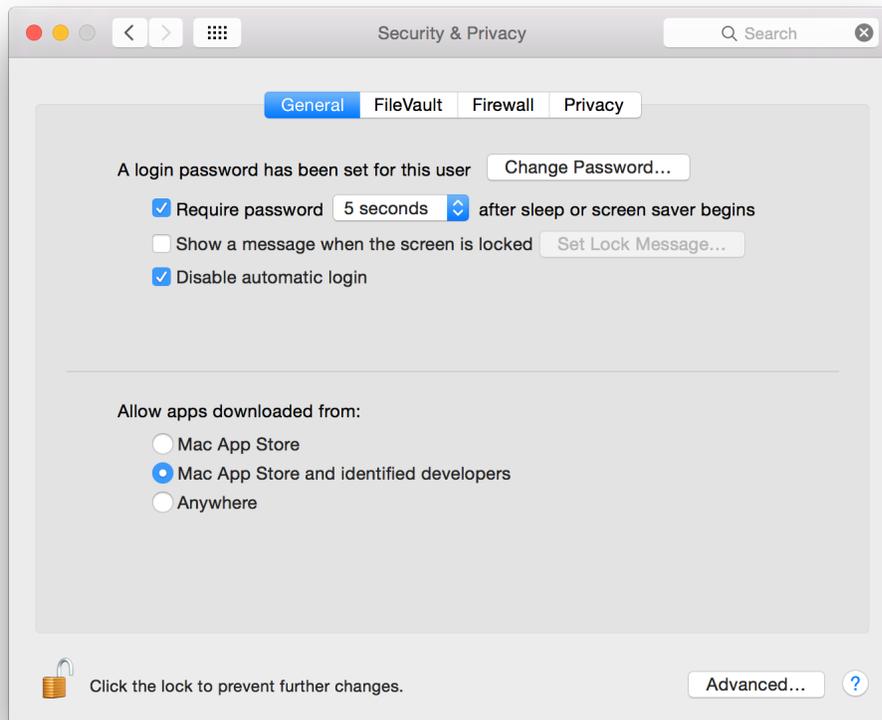
Gatekeeperの有効/無効を切り替える

Gatekeeperの有効/無効の切り替えは、「システム環境設定」の「Security & Privacy」画面で行います。コマンドラインユーティリティ「spctl」で、Gatekeeperをオフにし、状態を確認できます。

「Security & Privacy」環境設定画面でGatekeeperの有効/無効を切り替えるには

1. Finder上で「システム環境設定」を起動し、「Security & Privacy」を選択します。
2. ロックされている場合は「鍵」ボタンを押し、管理者パスワードを入力します。

3. Gatekeeperを有効にする場合は「Mac App Store and identified developers」を選択します。



4. Gatekeeperを無効にする場合は、「Mac App Store」または「Anywhere」を選択し、表示されるダイアログで、選択内容を確認します。

「spctl」コマンドでGatekeeperを無効にするには

1. 「Terminal」上で次のコマンドを入力します。

```
$ sudo spctl --master-disable
```

2. Returnキーを押してください。
3. パスワードを訊ねられるので、管理者パスワードを入力してください。

Gatekeeperが有効になったことを「spctl」コマンドで確認するには

1. 「Terminal」上で次のコマンドを入力します。

```
$ spctl --status
```

2. Returnキーを押してください。

Gatekeeperが有効であれば、コマンド出力は次のようになります。

```
assessments enabled
```

無効であれば次のようになります。

```
assessments disabled
```

Gatekeeperの動作をテストする

Developer ID証明書を使ってアプリケーションに署名した後、想定どおりに署名されているかどうかテストし、Gatekeeperが有効な環境でアプリケーションを起動したときの動作をシミュレートします。Gatekeeperが有効なMac上で、「隔離」したアプリケーションを起動するためには、Developer IDで署名されていなければなりません（隔離については[Knowledge Baseの記事](#)を参照）。さらに、Developer IDで署名していないアプリケーションを起動したときの、Gatekeeperの動作もテストできます。

Developer IDで署名したアプリケーションをテストする

「spctl」コマンドラインユーティリティを使って、アプリケーションが想定どおり、Developer ID証明書を使って署名されているかどうか確認できます。

Developer IDで署名したアプリケーションをテストするには

1. テスト用MacでGatekeeperを有効にします。「システム環境設定」の「Security & Privacy」画面で「Mac App Store and identified developers」を選択してください。
2. 「Terminal」上で次のコマンドを実行します。ただし「TrackMix.app」の部分は実際のアプリケーションパスで置き換えてください。

```
$ spctl -a -v TrackMix.app
```

3. Returnキーを押してください。

正しく署名されていれば、コマンド出力は次のようになります。

```
./TrackMix.app: accepted  
source=Developer ID
```

起動時の動作をテストする

Developer IDで署名したアプリケーションを包括的にテストするため、開発用ではないMac上で起動してみます。

Gatekeeperの動作をテストするための準備をするには

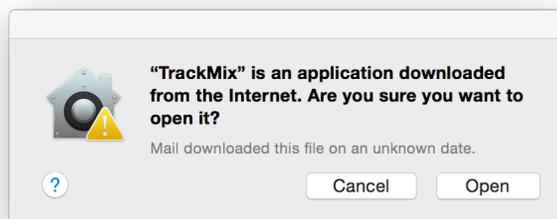
1. テスト用MacのGatekeeperを有効にします（“[Gatekeeperの有効/無効を切り替える](#)”（246 ページ）を参照）。
2. Developer IDで署名したアプリケーションを「隔離」します。これには次の2通りの方法があります。
 - Developer IDで署名したアプリケーションを、自分自身宛ての電子メールで送信し、受け取って使う。
 - Developer IDで署名したアプリケーションをローカル/遠隔サーバに置き、Safariを使ってダウンロードする。

これでGatekeeperの動作をテストする準備が整いました。

Developer IDで署名したアプリケーションを起動したときの、Gatekeeperの動作をテストするには

1. Finder上で、Developer IDで署名した「隔離」済みのアプリケーションのアイコンをダブルクリックしてください。

Mac上に、このアプリケーションを起動してよいか訊ねる警告が現れます。



これは、Gatekeeperが有効な環境で、「隔離」したアプリケーションを起動できることを表すので、Developer IDアプリケーションを適切にビルドできたことになります。

Important: この時点で警告が現れない場合は、「隔離」済みでないアプリケーションを起動しようとしている可能性があります。

Developer IDで署名していないアプリケーションを起動したときの、Gatekeeperの動作をテストするには

1. テスト用MacのGatekeeperを有効にします（“[Gatekeeperの有効/無効を切り替える](#)”（246 ページ）を参照）。

2. Developer IDで署名していない、「隔離」済みのアプリケーションを用意してください。

先に説明したのと同様、「隔離」する方法は2通りあります。

- Developer IDで署名したアプリケーションを、自分自身宛ての電子メールで送信し、受け取って使う。
- Developer IDで署名したアプリケーションをローカル/遠隔サーバに置き、Safariを使ってダウンロードする。

3. Finder上で、Developer IDで署名していない「隔離」済みのアプリケーションのアイコンをダブルクリックしてください。

Mac上にはアプリケーションの起動を遮断した旨の警告が現れます。このように、開発者の身許が分からないアプリケーションを起動しようとする、Gatekeeperが遮断し、Macを保護するのです。なお、一度起動すれば「隔離」されていない状態になるので、Gatekeeperが起動を遮断することはありません。

まとめ

この章では、Mac App Storeを通さずにMacアプリケーションを配布し、起動を遮断されないようにする手順を説明しました。

トラブルシューティング

アプリケーションを開発、テスト、登録、公開する工程では、トラブルシューティングの検討が欠かせません。起こりうる問題を、証明書、プロビジョニング、ビルドおよびコード署名、デバッグの4つに分類し、それぞれの対処法をまとめました。実際に問題が生じたときに、該当する項を参照してください。

証明書に関する問題

証明書が無効または使用不可になっても、再生成する前に、次のような問題がないか検討してください。

コード署名IDが見つからない

署名IDが欠落していれば、Xcodeがその旨を検出します。これは通常、別のMacに移行したときに起こります。[“チームプロビジョニングプロファイルを生成する”](#) (39 ページ) の手順に従って、署名IDを作成し、チームプロビジョニングプロファイルに追加してください。他のMacから署名IDをインポートする、リセットする、という選択肢もあります。独自にプロビジョニングプロファイルを作成し、自分で管理している場合、開発用証明書を失効させればこれも無効になります。[“メンバーセンターでプロビジョニングプロファイルを編集する”](#) (220 ページ) の手順に従い、生成し直してください。

この問題を回避するため、移行元のMac上で、開発者プロファイルのファイルとして証明書をエクスポートし、これを移行先のMacにインポートしてください ([“証明書やプロファイルをエクスポート/インポートする”](#) (180 ページ) を参照)。

該当するプロビジョニングプロファイルが見つからない

開発用プロビジョニングプロファイルが欠落している場合、Xcodeはこれを検出し、「General」ペインと「Capabilities」ペインの両方に警告メッセージを表示します。「No matching provisioning profiles found」というメッセージが、「General」ペインの「Team」ポップアップメニューの下に現れた場合、その下の「Fix Issue」ボタンを押してください。「No Devices Registered」ダイアログが現れた場合は、適切なデバイスをMacに接続 ([“チームプロビジョニングプロファイルを生成する”](#) (39 ページ) を参照) した上で、このボタンを押します。Xcodeが問題を解消できない場合は、[“IDおよびチームの設定をする”](#) (33 ページ) の手順に従い、プロジェクトの設定を確認してください。

Developer ID証明書に用いる秘密鍵が欠落している

Developer IDを失効させる必要がある場合は、Apple (product-security@apple.com) に連絡してください。あるいは、追加のDeveloper ID証明書を作成（[“追加のDeveloper ID証明書を作成する”](#)（177 ページ）を参照）して、開発や配布を続ける、という方法もあります。

中間証明書を紛失したため証明書が無効になった

証明書が無効になった原因として、その認証に用いる中間証明書を紛失したことも考えられます。Keychain Accessで証明書を検証（[“Keychain Accessを使って確認する”](#)（175 ページ）を参照）したとき、緑の円で囲んだチェックマークではなく、赤の丸で囲んだ「X」印と「This certificate was signed by an unknown authority」という文言が現れた場合、中間証明書がなくなっています。システムキーチェーンに「Apple Worldwide Developer Relations Certification Authority」という証明書がない場合は、[“誤って削除した中間証明書を登録する”](#)（179 ページ）に従って再登録してください。Developer ID証明書に用いる中間証明書は「Developer ID Certification Authority」という名前です。

証明書の信頼性に問題がある

「Keychain Access」で証明書を表示したとき、詳細領域に、緑の円で囲まれたチェックマークではなく、青い円で囲まれた白い「+」が現れる場合、信頼性の問題があります。対処方法については、[“Xcodeが証明書を信頼しない”](#)（255 ページ）を参照してください。

証明書が期限切れになった

期限切れになった証明書の期限を延ばすことはできません。削除して新たに作成する手順については、[“期限切れになった証明書を更新する”](#)（189 ページ）を参照してください。

Macに関する注: Developer ID証明書が期限切れになっても、ユーザはなお、この証明書で署名したMacアプリケーションをダウンロード、インストール、実行できます。しかし、更新版に署名し、あるいは新規にアプリケーションを開発するためには、新しいDeveloper ID証明書が必要です。

要求がタイムアウトになった

要求がタイムアウトになった旨のダイアログが現れた場合、メンバーセンターの「Certificates, Identifiers & Profiles」セクションが、保守のため停止している可能性があります。後でもう一度やり直してください。

プロビジョニングプロファイルや署名IDがXcodeのメニューに現れない

「Devices」ウィンドウでアプリケーションを配布する場合やプロジェクトエディタで「Code Signing Identity」ビルド設定を行う場合に、プロビジョニングプロファイルまたは署名IDが「Provisioning Profile」にもポップアップメニューにも表示されないことがあります。この場合、Xcode上でプロビジョニングプロファイルをダウンロードしてください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215 ページ）を参照）。

プロビジョニングに関する問題

プロビジョニングに関する問題の多くは、プロビジョニングプロファイルがアプリケーションと対応していないか、無効あるいは期限切れになっているために起こります。

デバイスにインストールしたプロビジョニングプロファイルが無効である

開発用デバイスのプロビジョニングプロファイルが期限切れ、または無効である場合、[“デバイス上でプロビジョニングプロファイルを確認して削除する”](#)（223 ページ）の手順に従い削除してください。

メンバーセンターのプロビジョニングプロファイルが無効である

失効した証明書や期限切れの証明書が含まれている場合、そのプロビジョニングプロファイルは無効です。また、App IDのエンタイトルメントが変更された場合やApp ID自体が削除された場合にも、プロビジョニングプロファイルが無効になります。無効化されたデバイスが含まれている場合は、開発用プロビジョニングプロファイルまたはアドホックプロビジョニングプロファイルも無効になります。

メンバーセンターのプロビジョニングプロファイルが無効の状態で見られる場合、Xcode上でダウンロードしてください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215 ページ）を参照）。たとえば、開発用証明書を失効させる場合、Xcodeを使って関連するチームプロビジョニングプロファイルを再生成します。

自分で管理しているプロビジョニングプロファイルがメンバーセンターに無効な状態で表示される場合は、削除するか（[“プロビジョニングプロファイルをメンバーセンターから削除する”](#)（224 ページ）を参照）、編集してください（[“メンバーセンターでプロビジョニングプロファイルを編集する”](#)（220 ページ）を参照）。

該当するプロビジョニングプロファイルが見つからない

「Your build settings specify a provisioning profile with [...] however, no such provisioning profile was found.」という警告が、「General」ペインにある「Team」ポップアップメニューの下に表示された場合、旧版のXcodeから引き継いだ「Provisioning Profile」および「Code Signing Identity」ビルド設定が適切でないかも知れません。

この場合、警告メッセージの下にある「Fix Issue」ボタンを押してください。このボタンをもう一度押す必要があるかも知れません。「Provisioning Profile」ビルド設定は「None」、「Code Signing Identity」ビルド設定は、iOS/tvOS/watchOSアプリケーションの場合は「Automatic」および「iOS Developer」、Macアプリケーションの場合は「Automatic」および「Mac Developer」になるはずですが。旧プロジェクトでXcodeが管理していたプロビジョニングプロファイルを使いたい場合は、「[Xcodeで管理されるプロビジョニングプロファイルを移行する](#)」（219 ページ）を参照してください。

独自の開発用プロビジョニングプロファイルを使っている場合は、「[独自のプロビジョニングプロファイルを使用する](#)」（216 ページ）を参照して、プロジェクトの設定方法を確認してください。

ビルドおよびコード署名に関する問題

開発中、Xcodeが管理するチームプロビジョニングプロファイルを使っている（「[チームプロビジョニングプロファイルの詳細](#)」（99 ページ）を参照）場合、コード署名やプロビジョニングに関する問題があれば、Xcodeはアプリケーションをビルドする前に修復しようと試みます。この場合、「Code Signing Identity」ビルド設定を独自に設定してはいないはずですが。一方、独自の開発用プロビジョニングプロファイルを使い、このビルド設定を施している場合（「[独自のプロビジョニングプロファイルを使用する](#)」（216 ページ）を参照）、ここに示されたビルドの問題が起きることがあります。ビルドに失敗する原因として多いのは、コード署名IDが不適切であることです。

Xcodeがプロビジョニングプロファイルを見つけられない

次のエラーメッセージは、AppIDが変わったなどのため、プロビジョニングプロファイルを置き換えたときに現れることがあります。

```
Code Sign error: Provisioning Profile 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx' can't be found
```

この場合、「Code Signing Identity」ビルド設定の値が、正しいプロビジョニングプロファイルとコード署名IDであることを確認してください。また、プロジェクトとターゲットの「Code Signing Identity」ビルド設定が、一致していないことも考えられます。

Xcodeが証明書を信頼しない

開発用証明書や配布用証明書の正当性をXcodeが確認できない場合に、このエラーメッセージが現れます。

```
Code Sign error: CSSMERR_TP_NOT_TRUSTED
```

信頼設定が「Use System Defaults」でなければ、アプリケーションのビルド、実行時に、コマンドラインユーティリティ「codesign」が「CSSMERR_TP_TRUSTEDというエラーメッセージを出します。証明書の信頼設定を、デフォルト値である「Use System Defaults」以外に変更しないでください。以下の手順で、開発用、配布用、および中間証明書の信頼設定を修復します。

To :証明書の信頼レベルをシステムデフォルトに設定するには

1. 「Keychain Access」を起動します。
2. 「Category」セクションの「My Certificates」を選択します。
3. 証明書をダブルクリックしてください。
4. 証明書ウインドウで、展開表示用の三角印をクリックし、「Trust」セクションを表示します。
5. 「When using this certificate」として「Use System Defaults」を選択します。
6. 証明書ウインドウを閉じてください。
7. 証明書情報を調べ、有効であることを確認します。

ビルド設定「Code Signing Identity」がどの証明書とも合致しない

次のエラーメッセージは、証明書が期限切れになったなど、無効である場合に現れます。

```
Code Signing Identity 'iPhone Developer' doesn't match any valid, non-expired, certificate/private key pair in your keychain.
```

同じコード署名IDで複数のターゲットに署名する場合、このビルド設定は、ターゲットごとではなく、プロジェクトレベルで設定しなければなりません。両方の設定があれば、ターゲットに対する設定が優先します。ターゲットに対して「Code Signing Identity」が設定されている場合、これを選択し、「Edit」>「Delete」コマンドで削除してください。その上で、プロジェクトに対する「Code Signing Identity」として、適切な証明書を設定します。

開発用証明書やチームプロビジョニングプロファイルが「Code Signing Identity」ポップアップメニューに表示されない場合は、プロビジョニングプロファイルのダウンロードを試してください（[“Xcode上でプロビジョニングプロファイルをダウンロードする”](#)（215ページ）を参照）。Xcodeで管理される

プロビジョニングプロファイルを使っている場合は、「[Xcodeで管理されるプロビジョニングプロファイルを移行する](#)」（219ページ）を参照してください。独自のプロビジョニングプロファイルを使っている場合は、「[独自のプロビジョニングプロファイルを使用する](#)」（216ページ）を参照してください。

キーチェーンにコード署名IDが重複して登録されている

キーチェーンにコード署名IDが重複していると、開発用IDが重複している、配布用IDが重複している、などのエラーメッセージが現れます（キーチェーンにはそれぞれのIDを1つずつしか登録できません）。

```
Build error "iPhone Developer: <your_name> (XYZ123ABC): ambiguous (matches "iPhone Developer: <your_name> (XYZ123ABC)" in /Library/Keychains/System.keychain and "iPhone Developer: <your_name> (XYZ123ABC)" in /Users/../../Library/Keychains/login.keychain)"
```

```
[BEROR]CodeSign error: Certificate identity 'iPhone Distribution: <your_name>' appears more than once in the keychain. The codesign tool requires there only be one.
```

この場合はまず、重複しているコード署名IDを、キーチェーンから削除してみてください（「[署名IDをキーチェーンから削除する](#)」（184ページ）を参照）。

プロビジョニングプロファイルのアプリケーションIDが、アプリケーションのバンドルIDと一致しない

ビルド設定「Code Signing Identity」に指定されているプロビジョニングプロファイルのApp IDと、アプリケーションのバンドルIDが合致しない場合、次のようなエラーメッセージが現れます。

```
Code Sign error: Provisioning profile 'MyApp Profile' specifies the Application Identifier 'com.mycompany.MyApp.*' which doesn't match the current setting 'com.mycompany.MyApp'
```

この場合、XcodeプロジェクトでバンドルIDが正しく設定されていること、ビルド設定「Code Signing Identity」で指定された証明書やプロビジョニングプロファイルが適切であること、プロビジョニングプロファイルに正しいApp IDが使われていることを確認してください。

デバイスがリストに表示されないか、「Scheme」ツールバーに選択不可として表示される

プロジェクトやワークスペースを開いた際、接続済みのデバイスが、ツールバーの「Scheme」メニューの「Ineligible Devices」以下に表示されない場合、マウスカーソルを当該デバイス上に持っていけば、その理由を確認できます。以下を確認してください。

1. アプリケーションが動作するiOSの版が、デバイスにインストールされたiOSの版以降であること。
詳しくは[“ターゲットデバイスを設定する \(iOS、watchOS\)”](#) (44ページ) を参照してください。
2. プロジェクトで使っているiOS SDKの版番号が、当該デバイスで動作しているiOSの版以降であること。
たとえば、Xcodeには「iOS SDK 4.3」と表示されるが、デバイスにはiOS 5.0がインストールされている場合、iOS SDK 5.0が付属する版のXcodeをMacにインストールしなければなりません。

デバッグ情報に関する問題

デバッグに関してよく起こるのは、Xcodeがまだデバイスから必要な情報を収集していない、という問題です。

デバイスに接続したとき、Xcodeに「Unknown iOS Detected」というダイアログが現れる

このメッセージが現れるのは、デバイスで動作している版のiOSをXcodeがまだ認識しておらず、そのデバッグシンボルを（デバイスから）ダウンロードする必要がある場合です。デバイス上で動かしてデバッグするためには、ダウンロードが終わるまで、接続したまま待たなければなりません。

アーカイブおよび登録の問題

「Archives」ウィンドウでアーカイブを選択した際に、「Validate」ボタンと「Submit」ボタンが無効になっている場合は、そのアーカイブに1つの最上位アプリケーションが含まれているかどうかを確認してください。

サポートされるケイパビリティ

アプリケーションから利用できるケイパビリティは、プラットフォーム、アプリケーションの種類、署名IDによって異なります。署名IDは、ターゲットのチーム設定（「General」ペインで設定）によって異なります。Macアプリケーションのケイパビリティは、「Team」ポップアップメニューで選択した署名ID（Mac App Store、Developer ID、None）によっても異なります（[「IDおよびチームの設定をす](#)
[る」](#)（33 ページ）を参照）。

ケイパビリティ	iOS (ADP)	iOS (エンタープライズ)	iOS (Apple ID)	watchOS 2 WatchKit エクステンション	watchOS 2 WatchKit アプリケーション	tvOS (ADP)	Mac (ADP)	Mac (Developer ID)
Appグループ	✓	✓	✓	✓	✓	✓	✓	✓
Apple Pay	✓	✗	✗	✓	✗	✗	✗	✗
アプリケーションのサンドボックス	✗	✗	✗	✗	✗	✗	✓	✓
Associated Domains	✓	✓	✗	✓	✗	✗	✗	✗
バックグラウンドモード	✓	✓	✓	✗	✗	✓	✗	✗
データ保護	✓	✓	✓	✓	✗	✓	✗	✗
Game Center	✓	✗	✗	✗	✗	✓	✓	✗
ゲームコントロール	✗	✗	✗	✗	✗	✓	✗	✗

キイパビリティ	iOS (ADP)	iOS (エンタープライズ)	iOS (Apple ID)	watchOS 2 WatchKit エクステンション	watchOS 2 WatchKit アプリケーション	tvOS (ADP)	Mac (ADP)	Mac (Developer ID)
HealthKit	✓	✓	✓	✓	✗	✗	✗	✗
HomeKit	✓	✓	✓	✓	✗	✗	✗	✗
iCloud	✓	✓	✗	✓	✗	✓	✓	✗
In-App Purchase	✓	✗	✗	✗	✗	✓	✓	✗
Inter-App Audio	✓	✓	✓	✗	✗	✗	✗	✗
キーチェーンの共有	✓	✓	✓	✓	✗	✓	✓	✗
マップ	✓	✓	✓	✓	✗	✗	✓	✗
個人向けVPN	✓	✓	✗	✗	✗	✗	✓	✗
Push Notification	✓	✓	✗	✓	✗	✗	✓	✗
Wallet	✓	✓	✗	✓	✗	✗	✗	✗
ワイヤレスアクセサリの設定	✓	✓	✓	✗	✗	✗	✗	✗

iOS (ADP) 。チームがApple Developer Programの会員であるiOSアプリケーションターゲット。

iOS (エンタープライズ) 。チームがDeveloper Enterprise Programの会員であるiOSアプリケーションターゲット。MapKitフレームワークは使えますが、経路処理アプリケーションは開発できません（ジオロケーションファイルをiTunes Connectにアップロードできないため）。

iOS (Apple ID) 。チームがApple Developer Programの会員ではないiOSアプリケーションターゲット。

watchOS 2 WatchKitエクステンション。チームがApple Developer Programの会員であるwatchOS 2 WatchKitエクステンションターゲット。

watchOS 2 WatchKitアプリケーション。チームがApple Developer Programの会員であるwatchOS 2 WatchKitアプリケーションターゲット。

tvOS (ADP)。チームがApple Developer Programの会員であるtvOSアプリケーションターゲット。

Mac (ADP)。チームがApple Developer Programの会員であり、署名IDがMac App StoreであるMacアプリケーションターゲット。

Mac (Developer ID)。チームがApple Developer Programの会員であり、署名IDがDeveloper IDであるMacアプリケーションターゲット。Game CenterやIn-App Purchaseを利用するためには、アプリケーションレコードを作成し、iTunesConnect上でサービスを設定する必要があります。ただしアプリケーションをアップロードする必要はありません。

Mac (Apple ID)。チームがApple Developer Programの会員ではないMacアプリケーションターゲット。

書類の改訂履歴

この表は「アプリケーションの配布に関するガイド」の改訂履歴です。

日付	メモ
2015-10-21	Xcode 7.1に合わせて改訂しました。
2015-09-16	Xcode 7.0に合わせて改訂しました。
2015-06-16	Apple Developer Programの変更に伴う改訂を施しました。
2015-04-08	Xcode 6.3に合わせて改訂しました。「Archives」オーガナイザのワークフローを改訂し、「Crashes」オーガナイザの操作手順を「クラッシュレポートの分析」の章に追加しました。
2014-12-18	全体を通して細かな変更を行いました。
2014-10-20	細かな変更を加えました。
2014-10-16	Xcode 6.1とiTunes Connectの変更に反映しました。
2014-09-17	Xcode 6に合わせて改訂しました。
2014-02-11	全体を通して細かな変更を行いました。
2013-12-12	「ID、デバイス、プロファイルの管理」の章に「App IDを削除する」を追加し、iTunes Connectの資料を参照するリンクを更新しました。

日付	メモ
2013-10-22	アセットカタログを管理する、Xcode上で証明書を失効させる、iOS 開発者向けのプレビュー版をインストールする、期限切れになったプロビジョニングプロファイルの期限を延ばす、アプリケーション エンタイトルメントを検証する、さまざまな問題に対処する、などの手順を追加しました。
2013-09-18	Xcode 5に合わせて改訂しました。
2013-04-23	全体を通して細かな変更を行いました。
2013-04-05	新規文書。アプリケーションを開発、テスト、配布する作業の流れを説明しています。

用語解説

アドホックプロビジョニングプロファイル (ad hoc provisioning profile) 配布用プロビジョニングプロファイルの一種。iOS/tvOS/watchOSアプリケーションをテスト用に配布するときに使います。

AppID ある特定のチームが開発したアプリケーション (群) を識別するための文字列。バンドルID照合文字列 (bundle ID search string) とチームID (Team ID) (Appleが生成する10文字の文字列で、チームを一意に識別可能) を連結した形です。

Apple Developer Program Apple開発者向けの購読サービス。ストア向けのiOS/watchOS/tvOS/Macアプリケーション開発に必要な資料や技術サポートを提供します。

Apple ID Appleが発行する開発者向けアカウント。名前とパスワードを付与します。開発者はこれを使って、Apple Developer Programが提供するツールにサインインします。同じ開発者 (あるいはApple ID) が複数のチームに参加することも可能です。

Apple Push Notificationサービス (Apple Push Notification service (APNs)) プッシュ通知の機能をアプリケーションに組み込めるよう、Appleが提供しているサービス (サーバその他の基盤機能)。このサービスが送信するメッセージをプッシュ通知 (push notification) と言います。

Apple TV App Store tvOSアプリケーションを購入、ダウンロードするためのサービス。Apple TV App StoreはtvOSデバイスから利用できます。

Apple Worldwide Developer Relations Certification Authority ストアに登録されたアプリケーションの、開発用/配布用証明書を検証する認証局。

アプリケーションのバイナリ (app binary) コンピュータ上で実行可能なマシンコードを収容するファイル。

App Store iOS/watchOSアプリケーションを購入、ダウンロードするためのサービス。iOSデバイスからだけでなく、Mac/Windowsコンピュータ上でも、iTunes Storeを通して利用できます。

アプリケーションの間引き (app thinning) メモリやディスクを効率よく使うよう、アプリケーションをダウンロード、インストールするための手段。切り分け (slicing)、ビットコード (bitcode)、オンデマンド型リソース (on-demand resources) の総称。

ビルド (build) Xcodeプロジェクトからビルドした製品、あるいはiTunes Connectにアップロードするアーカイブの内容。ストアやオペレーティングシステムは、Info.plistに記述された版番号とビルド文字列にもとづきビルドを識別します。

バンドルID (bundle ID) 特定のアプリケーションを他から厳密に識別する、逆DNS形式の文字列。

ビットコード (bitcode) コンパイル済みプログラムの中間表現。

バンドルID照合文字列 (bundle ID search string) App IDの後半部分。開発者が設定する文字列で、ある一群のバンドルID (個々のバン

ドルIDは特定のアプリケーションを識別)に合致します。たとえば、バンドルID照合文字列が「com.mycompany.MyApp」、あるいはワイルドカードが含まれる「com.mycompany.*」の場合、「com.mycompany.MyApp」というバンドルIDに合致します。

認証局 (certificate authority) 証明書を認証する機関。

証明書、識別子、プロフィール メンバーセンター内の領域のひとつ。Apple Developer Programの会員は、iOS/tvOS/watchOS/MacアプリケーションやSafari拡張機能を開発するために必要な資料その他を、ここから入手できます。

CSR (証明書署名要求、certificate signing request) 署名証明書を生成するために使う、個人情報や記述したファイル。さらに、証明書に含まれる公開鍵や、識別情報も記述します。

クライアントSSL証明書 (client SSL certificate) 開発者のサーバからAppleサービスに接続するための証明書。たとえばApple Push Notificationサービスと通信する場合に必要です。

コード署名証明書 (code signing certificate) アプリケーションやインストーラに署名するために用いる署名証明書。

コンテナID (container ID) アプリケーションのiCloudコンテナを表す一意的な識別子。

クラッシュレポート (crash report) アプリケーションがクラッシュしたときにオペレーティングシステムが生成するレポート。

データ保護 (data protection) アプリケーションがディスクに保存するファイルに対し、セキュリティレベルを上げる働きをするデジタル的な関門。

Developer ID Mac App Storeを通さずに、コード署名を施したアプリケーションを配布するために、開発者が利用する仕組みの名称。

開発者プロフィール (developer profile) 開発者の開発用証明書、配布用証明書、プロビジョニングプロフィールを収容するファイル。

開発用証明書 (development certificate) 署名証明書的一种。開発中、チームのある開発者を識別するために使います。Xcodeを介して、デバイス上でアプリケーションを起動できるようにする働きがあります。

開発用プロビジョニングプロフィール (development provisioning profile) プロビジョニングプロフィールの一種。アプリケーションから特定のサービスを利用するとともに、開発期間中、所定のデバイス上で実行できるようにする働きがあります。開発者の名前と、いくつかの開発用証明書、開発に用いるデバイス群、App IDから成ります。

デバイス (device) アプリケーションが動作する各種デバイスの総称として、機種を区別する必要がないときに使います。iOSアプリケーションの場合、iPad、iPhone、iPod touchを指します。tvOSアプリケーションならばApple TVのことです。watchOSアプリケーションの場合はApple WatchとペアリングしたiPhoneのことです。MacアプリケーションであればMacコンピュータを指します。

デバイスID (device ID) Appleデバイス (Macコンピュータを含む) を一意に識別する手段。

配布用証明書 (distribution certificate) 署名証明書的一种。アプリケーションを配布し、Xcodeがなくてもデバイス上で起動できるようにするために使います。個々のチームメンバーではなく、開発チームを特定するようになっています。

配布用プロビジョニングプロファイル

(distribution provisioning profile) プロビジョニングプロファイルの一種。Xcodeがなくてもデバイス上でアプリケーションを起動し、特定のサービスを利用できるようにする働きがあります。アプリケーションをストアに登録する、エクスポートしてストアを通さずに配布する、などのために使います。

エンタイトルメント (entitlement) 特定のアプリケーションやツールその他の実行形式ファイルに対して、通常ならば与えられない、追加の権限を付与するもの。

特定App ID (explicit App ID) ある特定のバンドルIDに合致するApp ID。複数のバンドルIDに合致する、ワイルドカードApp IDとは対照的な働きをします。

Game Center Appleが提供するソーシャルゲームネットワーク。プレイヤはこのサービスに接続し、他のプレイヤと情報を交換できます。

Gatekeeper OS Xの機能。Appleが認識している開発者がコード署名したアプリケーション以外は、起動できないようにする働きがあります。

iCloud ストレージの一種。iOSデバイスやOS Xデバイスで動作する複数のアプリケーション間で、ユーザのデータを共有できます。

In-App Purchase 埋め込み機能を購入し、直接アプリケーションに組み込めるようにする仕組み。開発者はストアに接続し、ユーザーからの決済処理を安全に行えます。

個人 (individual) Apple Developer Programに参加するアカウントの種類のひとつ。1人だけの開発者から成ります。

中間証明書 (intermediate certificate) 開発者のキーチェーンに登録して、署名証明書の発行元が信頼できると証明するために使う証明書。

iOS Appファイル OS Xのファイルの一種。iOS/tvOSデバイスにインストールできます。

Macインストーラパッケージ (Mac Installer Package) OS Xのファイルの一種。ダブルクリックするとインストーラが起動し、Macアプリケーションをインストールできます。

Newsstand iOSアプリケーションのひとつ。新聞や雑誌を購読し、フォルダに配送してもらう、という機能があります。

オンデマンド型リソース (on-demand resources) 実行ファイル以外のリソースで、あらかじめタグ付けしておき、アプリケーションバンドルとは別にダウンロードできるようにしたもの。

組織 (organization) Apple Developer Programに参加するアカウントの種類のひとつ。一般に複数のチームメンバーから成ります。

Wallet iOSアプリケーションのひとつ。パス、チケット、クーポンなどをiOS上で利用できます。

パス (passes) クーポン、催し物のチケット、搭乗券などの情報をデジタル的に実装したものの。現実世界における製品やサービスと交換可能です。

プロビジョニング (provisioning) アプリケーションをデバイス上で起動し、ある種のサービスが利用できるよう、準備、設定する手続き。

プロビジョニングプロファイル (provisioning profile) システムプロファイルの一種。アプリケーション（複数可）のプロビジョニングをする働きがあります。

プッシュ通知 (push notification) フォアグラウンドで動作していないアプリケーションが、ユーザに対して送信するメッセージ。Apple Push Notificationサービス (Apple Push Notification service (APNs)) を利用して通知します。

隔離 (quarantine) ファイルやアプリケーションの状態のひとつ。ユーザが初めて開こうとすると、Gatekeeperが働き出す状態です。OS X上で、ウェブ経由でダウンロードし、あるいは電子メールで受け取ったファイルは、この状態になります。

経路制御アプリケーション (routing app) 経路情報を処理するアプリケーション。進路が変わるごとにナビゲーションを行うサービスなどがあります。経路制御アプリケーションとして登録すると、その情報を「Maps」その他のアプリケーションに渡して、表示を委ねることができます。

署名証明書 (signing certificate) インストーラパッケージ、電子メールメッセージなどに、署名するために用いる証明書。

署名ID (signing identity) コード署名 (アーカイブ署名を含む) に用いるデジタルID。キーチェーンに格納された、証明書とその秘密/公開鍵を含みます。

切り分け (slicing) ターゲットデバイスに合わせてアプリケーションバンドルの「変種」を生成する処理。

ストア (store) App Store、Mac App Store、Apple TV App Storeの総称。それぞれの違いが問題にならない文脈でよく使います。

シンボル解決 (symbolicate) クラッシュレポート中のメモリアドレスを、人が見て分かりやすい関数名や行番号に置き換える処理。

チーム管理者 (team admin) 開発チームに属し、チームエージェントの権限のいくつか (契約書に署名する権限を除く) を持っている人。チームエージェントの責任の一部を代行します。参照：チームエージェント (team agent)、チームメンバー (team member)。

チームエージェント (team agent) 開発チームに属し、チームに対してあらゆる作業ができる一方、法的な責任を負う人。参照：チーム管理者 (team admin)、チームメンバー (team member)。

チームID (Team ID) Appleが生成する10文字の文字列。チームを一意に特定できます。App IDの前半部分としても使われます。

チームメンバー (team member) 開発チームに属し、最小限の権限を持つ人。参照：チームエージェント (team agent)、チーム管理者 (team admin)。

チームプロビジョニングプロファイル (team provisioning profile) Xcodeが生成、管理する、開発用プロビジョニングプロファイルの一種。チームの開発用証明書、チームに登録されたデバイス、ワイルドカードAppID (これもXcodeが生成するもの) を収容しています。

tvOS Apple TVデバイス上で動作するオペレーティングシステム。

ユニバーサルアプリケーション (universal app) サポート対象であるどのオペレーティングシステムやターゲットデバイスでも動作するビルド。「変種 (variant)」も参照。

変種 (variant) 特定のターゲットデバイスに必要なアーキテクチャやリソースのみから成るアプリケーション。

watchOS Apple Watchデバイス上で動作するオペレーティングシステム。

ワイルドカードApp ID (wildcard App ID) App IDのうち、開発チームが用いる複数のバンドルIDに合致するもの。参照：[特定App ID \(explicit App ID\)](#)。

Xcode iOSワイルドカードApp ID (Xcode iOS Wildcard App ID) [ワイルドカードApp ID \(wildcard App ID\)](#) のうち、iOS開発者に代わってXcodeが管理するもの。

Xcode MacワイルドカードApp ID (Xcode Mac Wildcard App ID) [ワイルドカードApp ID \(wildcard App ID\)](#) のうち、Mac開発者に代わってXcodeが管理するもの。



Apple Inc.
Copyright © 2015 Apple Inc.
All rights reserved.

の法的権利を与え、地域によってはその他の権利がお客様に与えられる場合もあります。

本書の一部あるいは全部を Apple Inc. から書面による事前の許諾を得ることなく複写複製（コピー）することを禁じます。また、製品に付属のソフトウェアは同梱のソフトウェア使用許諾契約書に記載の条件のもとでお使いください。書類を個人で使用する場合に限り1台のコンピュータに保管すること、またその書類にアップルの著作権表示が含まれる限り、個人的な利用を目的に書類を複製することを認めます。

Apple ロゴは、米国その他の国で登録された Apple Inc. の商標です。

キーボードから入力可能な Apple ロゴについても、これを Apple Inc. からの書面による事前の許諾なしに商業的な目的で使用すると、連邦および州の商標法および不正競争防止法違反となる場合があります。

本書に記載されているテクノロジーに関しては、明示または黙示を問わず、使用を許諾しません。本書に記載されているテクノロジーに関するすべての知的財産権は、Apple Inc. が保有しています。本書は、Apple ブランドのコンピュータ用のアプリケーション開発に使用を限定します。

本書には正確な情報を記載するように努めました。ただし、誤植や制作上の誤記がないことを保証するものではありません。

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
U.S.A.

Apple Japan
〒106-6140 東京都港区六本木 6
丁目10番1号 六本木ヒルズ
<http://www.apple.com/jp>

Offline copy. Trademarks go here.

Apple Inc. は本書の内容を確認しておりますが、本書に関して、明示的であるか黙示的であるかを問わず、その品質、正確さ、市場性、または特定の目的に対する適合性に関して何らかの保証または表明を行うものではありません。その結果、本書は「現状有姿のまま」提供され、本書の品質または正確さに関連して発生するすべての損害は、購入者であるお客様が負うものとなります。

いかなる場合も、Apple Inc. は、本書の内容に含まれる瑕疵または不正確さによって生じる直接的、間接的、特殊的、偶発的、または結果的損害に対する賠償請求には一切応じません。そのような損害の可能性があらかじめ指摘されている場合においても同様です。

上記の損害に対する保証および救済は、口頭や書面によるか、または明示的や黙示的であるかを問わず、唯一のものであり、その他一切の保証にかわるものです。Apple Inc. の販売店、代理店、または従業員には、この保証に関する規定に何らかの変更、拡張、または追加を加える権限は与えられていません。

一部の国や地域では、黙示あるいは偶発的または結果的損害に対する賠償の免責または制限が認められていないため、上記の制限や免責がお客様に適用されない場合があります。この保証はお客様に特定