

『Xamarinネイティブによる モバイルアプリ開発』

(著者による追加情報：2018/07/11版)

◆0. 本稿の目的

Visual Studio 2017のVer.15.7で、Xamarinのプロジェクトテンプレートが大幅に変更されました。それに伴い、本書で使用していたテンプレートに該当するものがなくなってしまったため、追加情報として本稿でプロジェクトの作成方法を解説します。

【参考】「Visual Studio 2017 バージョン 15.7 リリース ノート」内、「iOS と Android の新しいプロジェクト テンプレート」

<https://docs.microsoft.com/ja-jp/visualstudio/releases/vs2017-relnotes#xamarin-templates>

◆1. Visual Studioは最新のものを利用する

本書は、Visual Studio 2017 Ver.15.3.1をもとに記述されています。もちろん古いバージョンのVisual Studio 2017には本書に記述されているテンプレートが含まれていますが、この使用はおすすめできません。

本書でも述べているように、Xamarinは、Android SDKやMac上のXcodeなどとの兼ね合いもあり、古いバージョンを使うと思わぬトラブルが発生しやすくなります。これを防ぐ近道は、Xamarin（やXamarinに関連するものすべて）を最新のバージョンにしておくことに他なりません。そのため、Visual Studioは最新のバージョンを利用することを強くおすすめします。

なお、少なくともXamarinに関係する部分については、ProfessionalでもCommunityでも同じテンプレートになっているようです。

◆2. Visual Studioのバージョンを確認する

Visual Studioのバージョンは [ヘルプ] → [Microsoft Visual Studioのバージョン情報] メニューで確認することができます (図1)。



図1：【バージョン情報】ウィンドウ

図1からわかるとおり、本稿は、Visual Studio Community 2017 Ver.15.7.3をもとに執筆しています。

◆3. 本書相当のテンプレートの作成

Ver.15.7では、本書で利用しているテンプレートはなくなってしまいましたが、以下の手順で同等のものを作成することができます。

※以下は「MyXamApp1」という名称にしていますが、本書の172ページと同じにする場合は「MyXamApp1」の部分で「XMemo」に置き換えてください。

●3.1. ソリューションの作成

まず、Visual Studioのメニューから【ファイル】→【新規作成】→【プロジェクト】を選択し、【新しいプロジェクト】ウィンドウを開きます。

そこで、【インストール済み】→【その他のプロジェクトの種類】→【Visual Studioソリューション】→【空のソリューション】で空のソリューションを作成します（図2）。

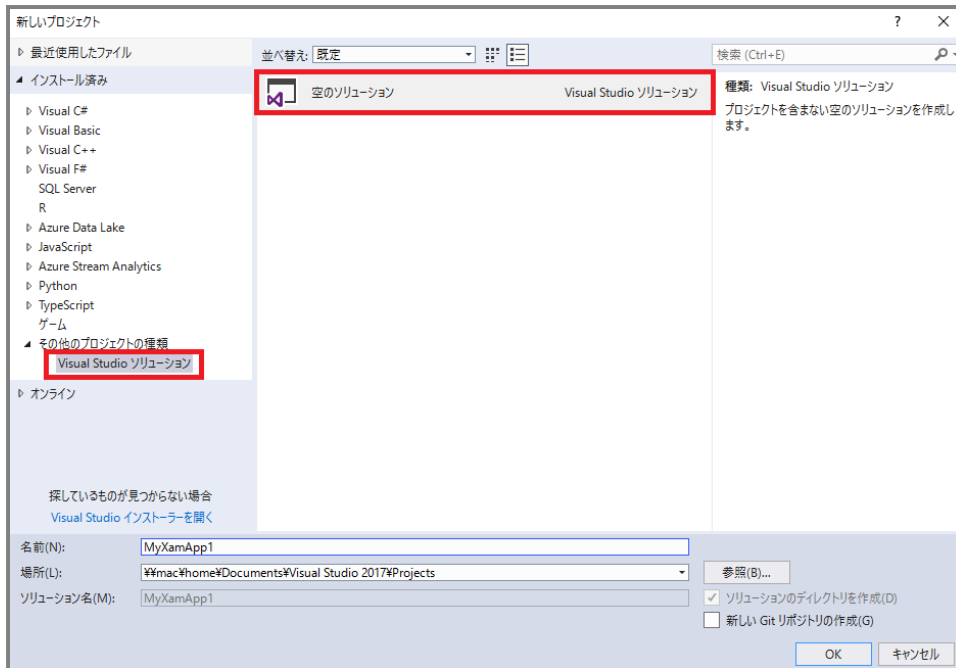


図2. 「新しいプロジェクト」 ウィンドウ

ここでは例としてソリューション名をMyXamApp1としています。

●3.2. iOSプロジェクトの作成

[ソリューションエクスプローラー] でソリューションを右クリックし、[追加] → [新しいプロジェクト] をクリックします (図3)。

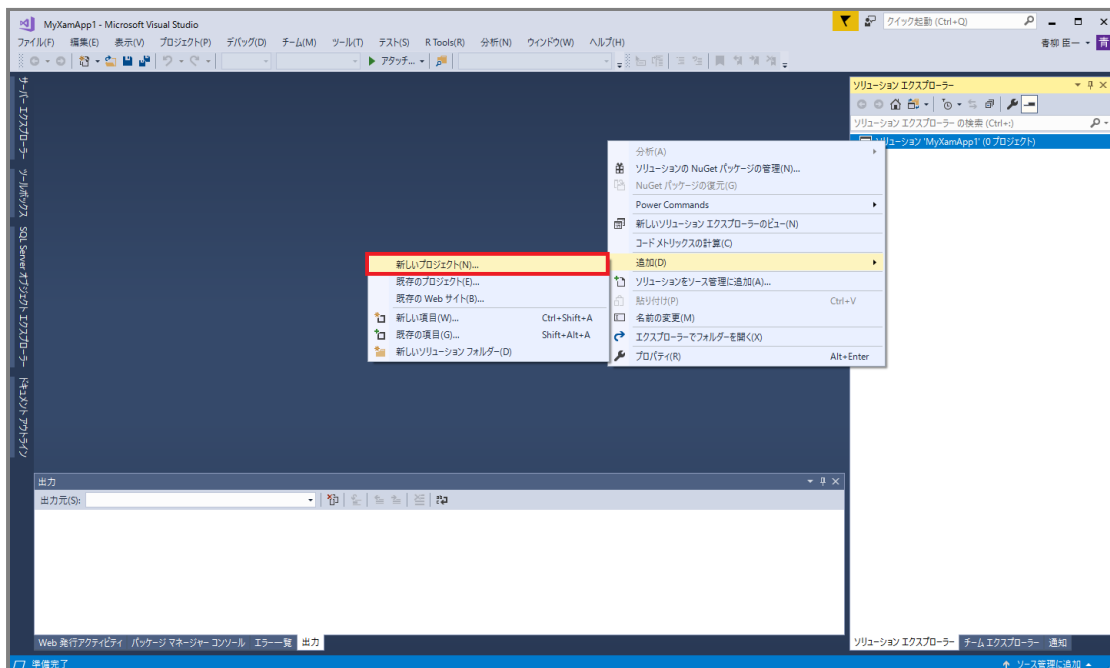


図3. [新しいプロジェクト] をクリック

[インストール済み] → [Visual C#] → [iPhone & iPad] の [iOSアプリ(Xamarin)] テンプレートでiOSプロジェクトを作成します。この時、ファイル名は「MyXamApp1.iOS」とします (図4)。

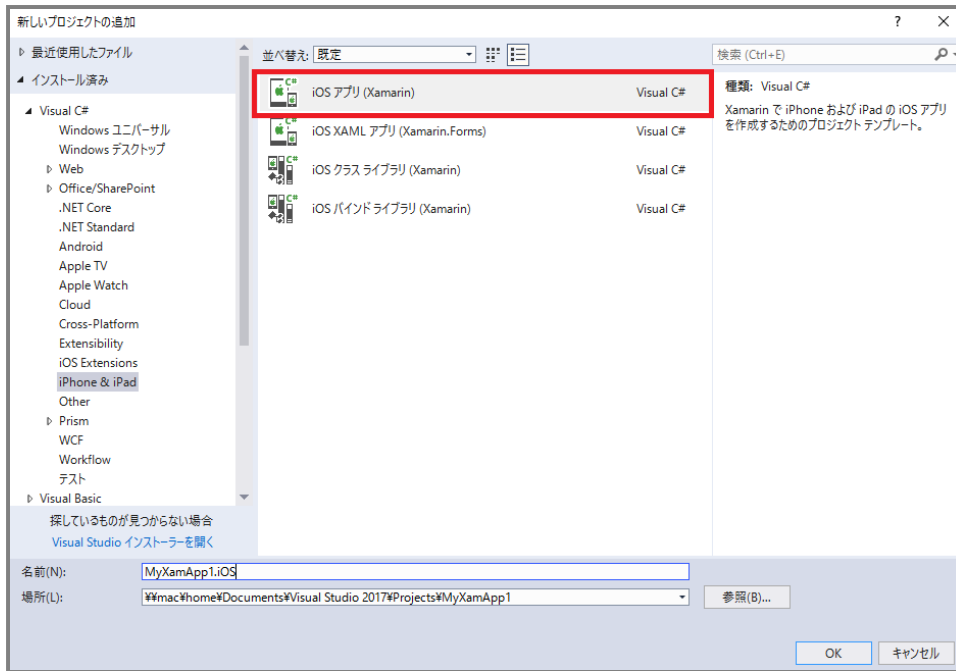


図4. [iOSアプリ(Xamarin)] テンプレートを選択する

iOSプロジェクトでは、図5のように各種の設定を選択することができますが、すべてデフォルトのままでも問題ありません。

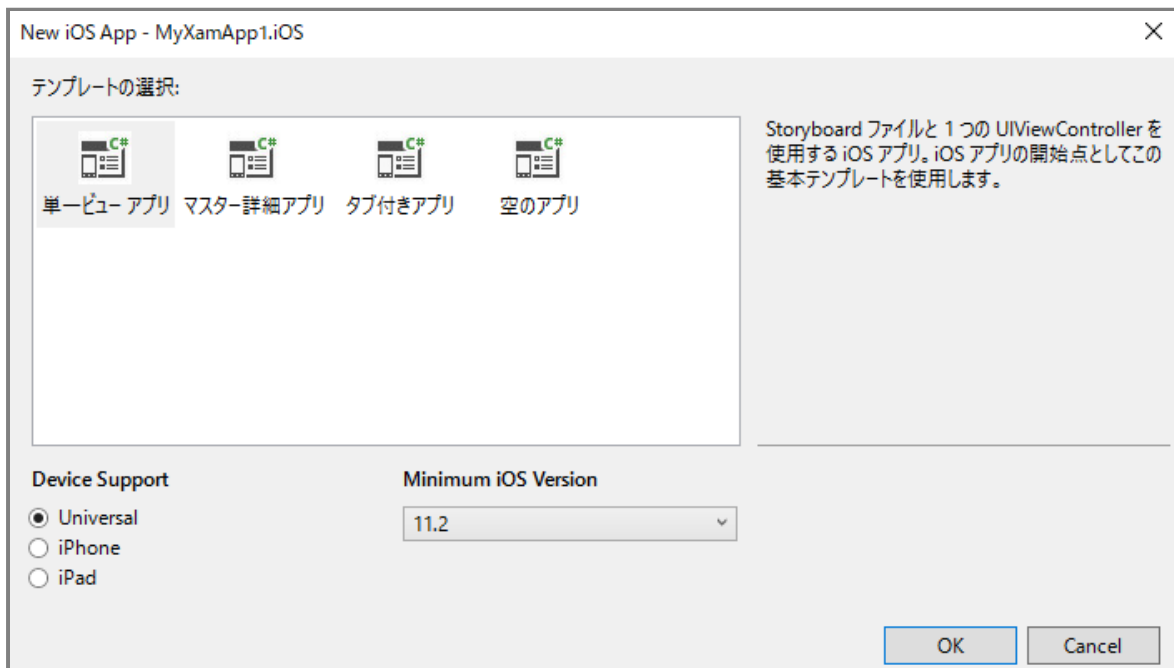


図5. iOSプロジェクトの設定

●3.3. Androidプロジェクトの作成

3.2と同様に、ソリューションを右クリックしてAndroidプロジェクトを追加します。

Androidプロジェクトは [インストール済み] → [Visual C#] → [Android] → [Androidアプリ (Xamarin)] テンプレートです。名前は「MyXamApp1.Android」とします (図6)。

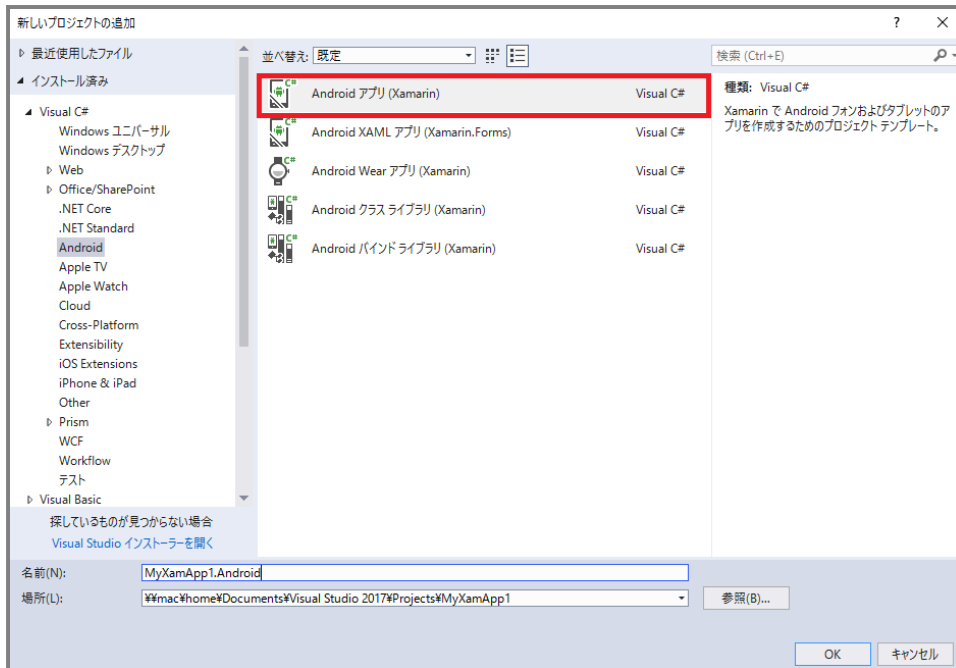


図6. [Androidアプリ(Xamarin)] テンプレートを選択する

Androidプロジェクトの設定も、すべてデフォルトのまま問題ありません (図7)。

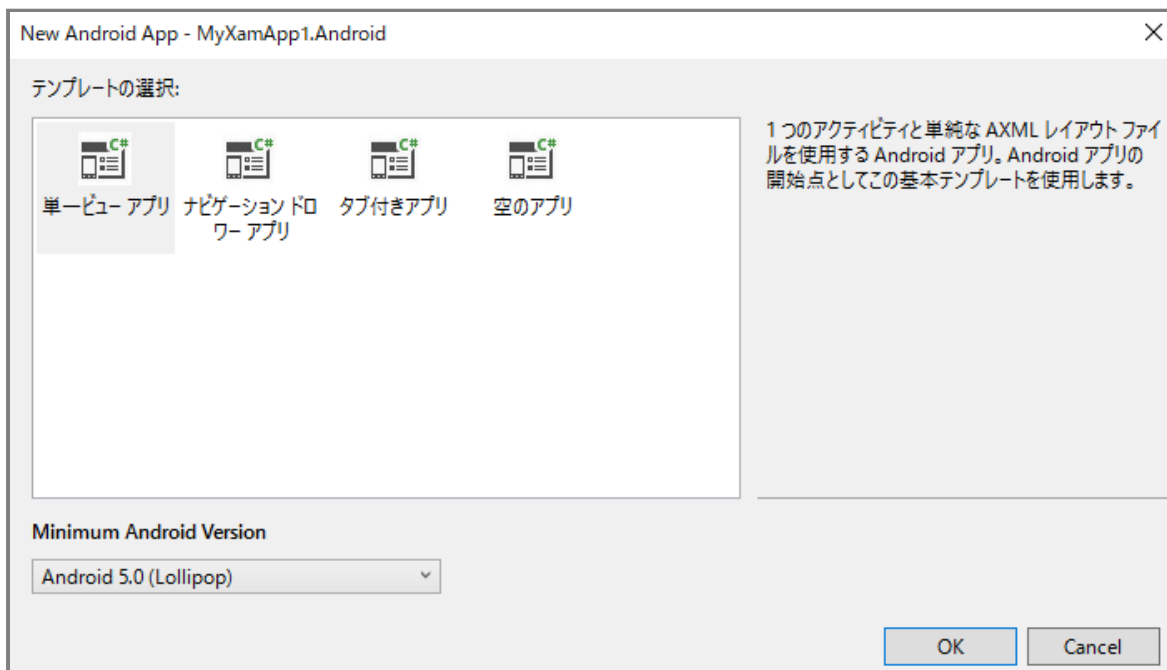


図7. Androidプロジェクトの設定

●3.4. PCLプロジェクトの作成

3.2と同様に、ソリューションを右クリックして移植可能 (PCL) プロジェクトを追加します。

移植可能プロジェクトは [インストール済み] → [Visual C#] → [クラスライブラリ(レガシ ポータブル)] テンプレートです。名前は「MyXamApp1」とします (図8)。

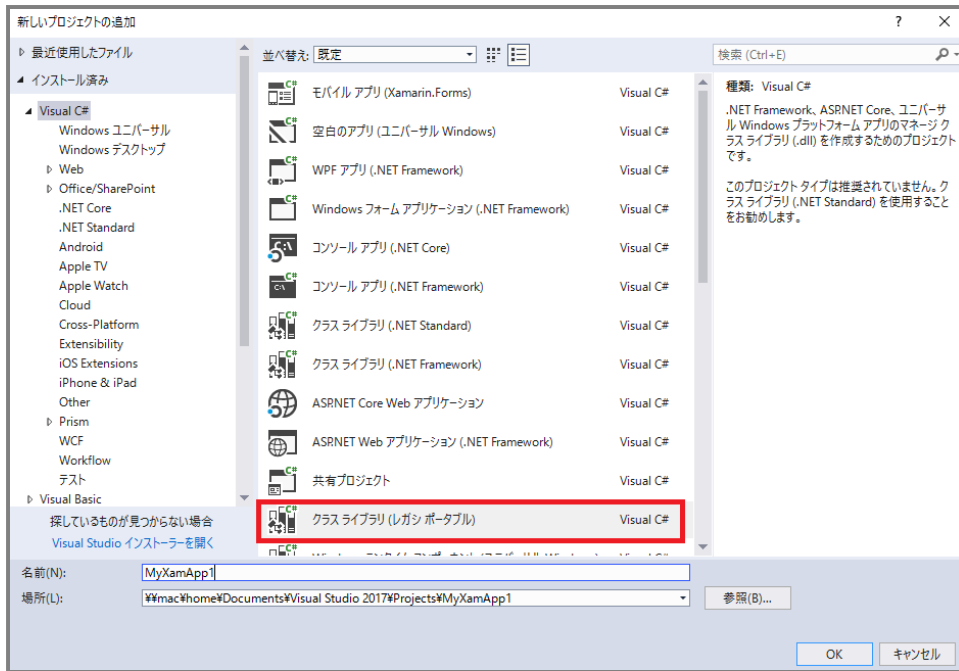


図8. [クラスライブラリ(レガシ ポータブル)] テンプレートを選択する

移植可能プロジェクトの設定は、図9を参考にして下さい。

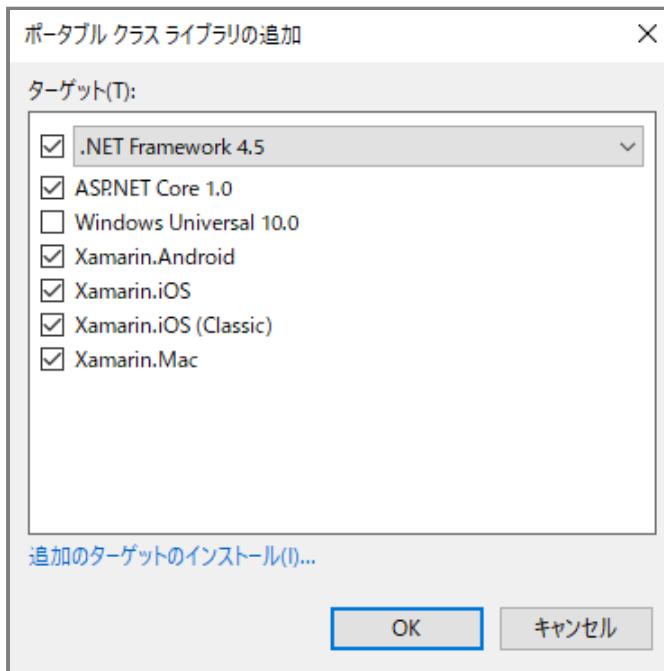


図9. 移植可能プロジェクトの設定

●3.5. プロジェクト間の参照を追加する

本書175ページに「XMemo.iOSとXMemo.Androidは移植可能プロジェクトを参照している」とありますが、本稿の手順で作成したソリューションの場合はプロジェクト参照が追加されていません。

そこで、参照を手動で追加します。

ソリューションエクスプローラーのMyXamApp1.iOSの [参照] を右クリックし、 [参照の追加] をクリックします (図10)。

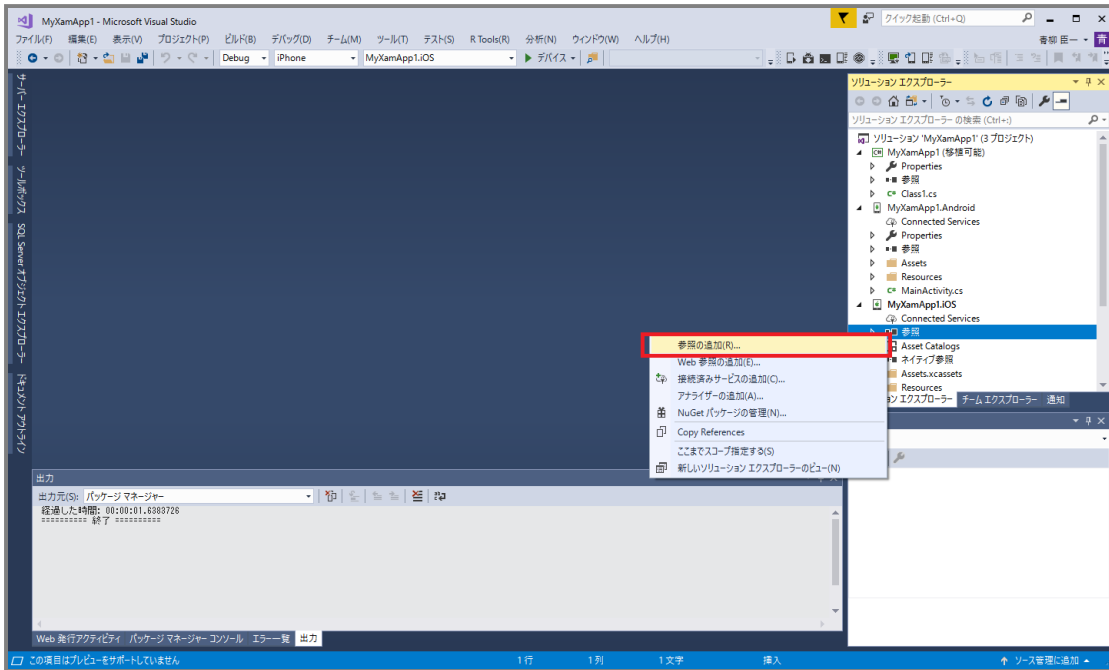


図10. [参照の追加] をクリック

表示された「参照マネージャー」で [プロジェクト] を選択し、 [MyXamApp1] のチェックをオンにして、 [OK] ボタンをクリックします (図11)。

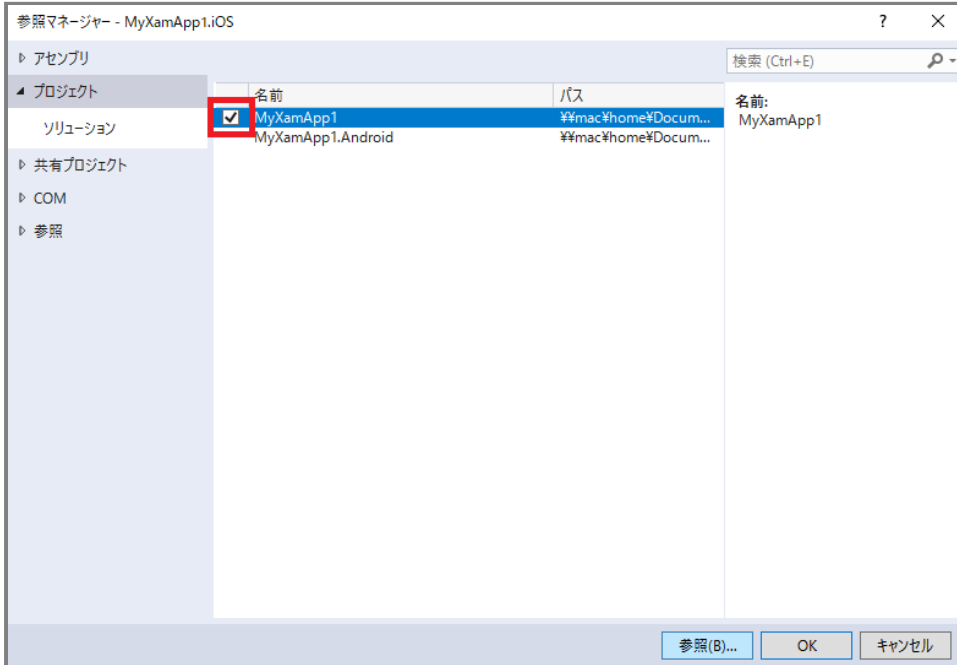


図11. [MyXamApp1] のチェックをオン

同様に、MyXamApp1.Androidプロジェクトでも [参照] を右クリックし、 [参照の追加] をクリックします (図12)。

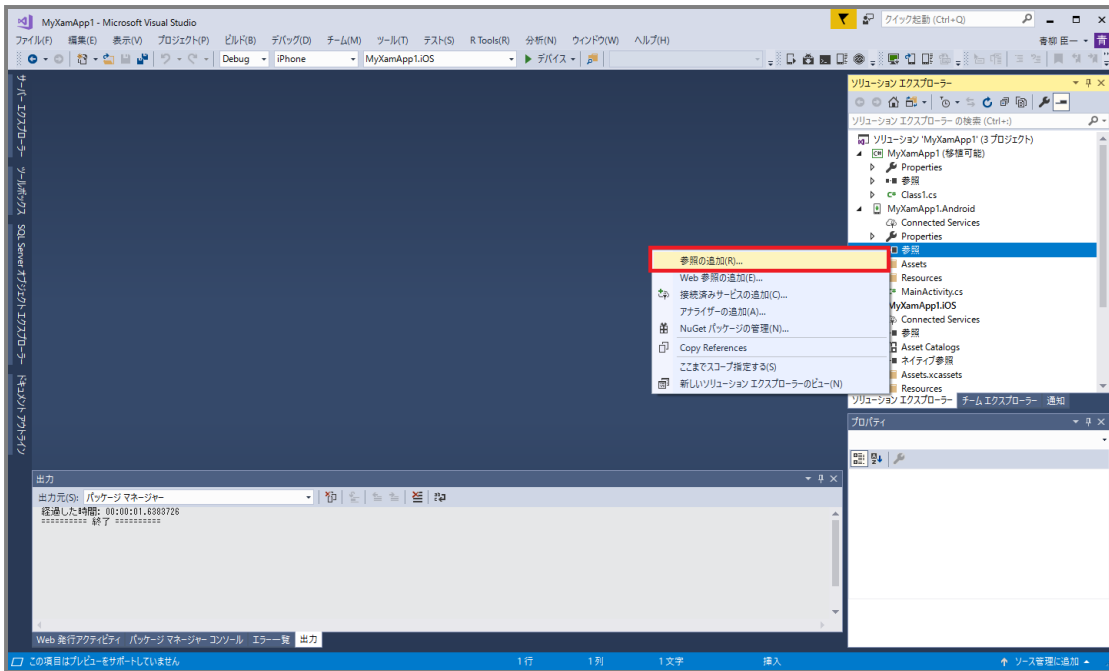


図12. [参照の追加] をクリック

表示された「参照マネージャー」で [プロジェクト] を選択し、[MyXamApp1] のチェックをオンにして、[OK] ボタンをクリックします (図13)。

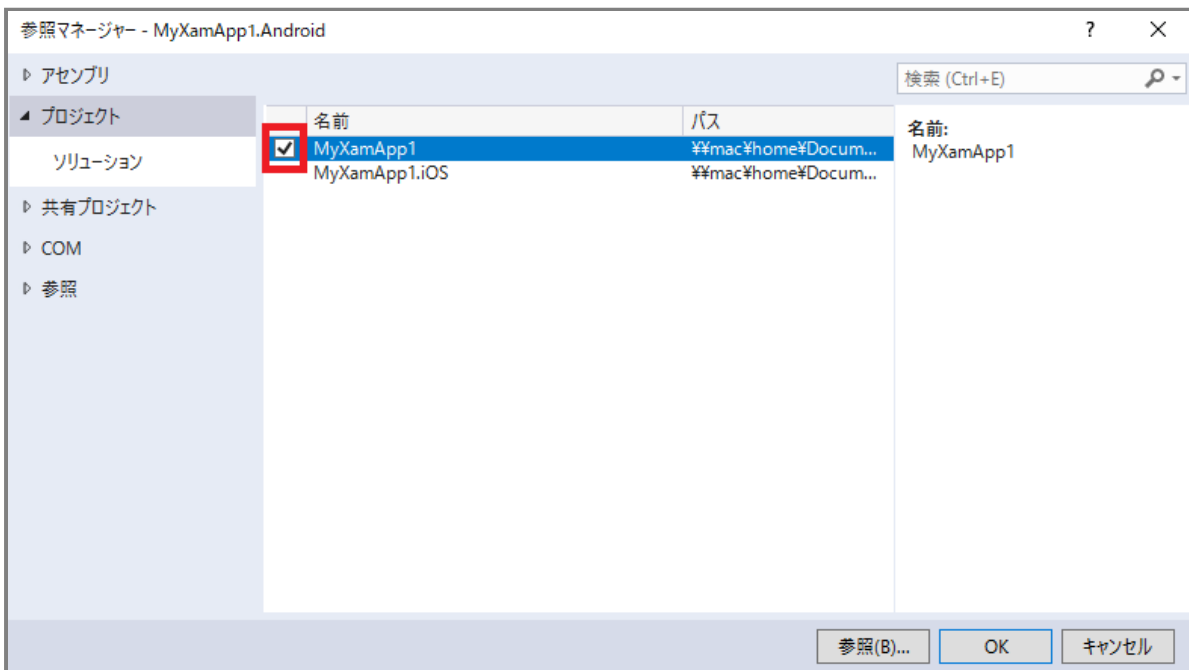


図13. [MyXamApp1] のチェックをオン

●3.6. できあがり

以上の手順で、書籍で利用しているテンプレート相当のプロジェクトを作成することができました。

●補足1. PCLの名称

本書執筆時点では、移植可能プロジェクトとしてPCL（ポータブルクラスライブラリ）という技術が使用されるのが通常でした。これが今では、**.NET Standard**と呼ばれる技術に置き換えられつつあります。

.NET Standardは、「PCLの進化系のようなもの」と捉えていただければよいかと思います。

そのため、現在のプロジェクトテンプレートでは、PCLは「クラスライブラリ(レガシポータブル)」という名前になっています。

これに対して、.NET Standard版の移植可能プロジェクトは「クラスライブラリ(.NET Standard)」です。

XamarinはPCL/.NET Standardどちらの移植可能プロジェクトにも対応しているため、どちらも使用することができます。本稿では、書籍にあわせてPCLで作成しました。

●補足2. Androidプロジェクトで「名前空間に関するエラー」が発生する場合

Androidプロジェクトにおいて、以下のエラーが発生する場合があります。

```
CS0234 型または名前空間の名前 'Support' が名前空間 'MyXamApp1.Android' に存在しません
(アセンブリ参照があることを確認してください)。
```

このエラーは、プロジェクト名を「MyXamApp1.Android」のようなものにした場合に起こります。ソースコード内の`Android.Support.V7.Widget.Toolbar`などの先頭にある`Android`が`MyXamApp1.Android`の`Android`だと解釈されてしまうためです。

このエラーを回避するためには、ソースコードで`global::Android.Support.V7.Widget.Toolbar`のように、名前空間名の先頭に`global::`を追加します。

`global::`を追加することで、`Android`が`MyXamApp1.Android`の`Android`ではなく、「グローバル名前空間の`Android`である」と明示することができます。

テンプレートとは無関係の問題ですが、名前空間名の付け方によって発生しうるエラーなので、同様のエラーが発生した場合には参考にしてください。