

# Xamarin インストール・事前作業手順書

(2017/3/3 更新)



JAPAN  
XAMARIN  
USER  
GROUP

Japan Xamarin User Group

田淵義人 @エクセルソフト

Twitter: [@ytabuchi](https://twitter.com/ytabuchi)

facebook: [ytabuchi.xlsoft](https://www.facebook.com/ytabuchi.xlsoft)

お疲れ様でした

ハンズオン会場でお待ちしております(^ ^)

# はじめに

本ドキュメントでは、Xamarin のインストールから Xamarin.Forms のデバッグ実行までをカバーします。トレーニング、ハンズオンを行う前に、Xamarin.Forms アプリを実行するまでを行ってください。

特に Android 開発をする方は、初回ビルドに大量のダウンロードが発生しますので、従量制限なしのネットワーク環境で作業を行っていただくことを強くお勧めいたします。

# 目次

Xamarin インストール

SDK更新や開発者登録

Xamarin でビルドのテスト

Microsoft Azure アカウントの準備(必要に応じて)

# Xamarin 対応環境一覧

Xamarin は様々な OS 向けのアプリを各種 IDE で開発できます。  
○の付いているアプリが開発可能です。

	Windows	Mac OS X
	Visual Studio	Xamarin Studio
iOS	○（要Mac）	○
Android	○	○
Mac	×	○
UWP	○	×
Windows Phone	○	×
Windowsストア	○	×

# 必要な作業

Xamarin のインストール、アップデート

SDK 更新や開発者登録

Android SDK 更新 (Windows / Androidの場合)

Apple Developer Program 加入、プロビジョニングプロファイルの準備 (Mac / iOSの場合)

実機または Simulator / Emulator でデバッグ実行

# Xamarin インストール

# Xamarin インストール

## Windows (新規)

次のスライド以降の手順を参照してください

## Windows (追加)

VS 2015 の場合は [Update 3](#) に  
VS 2013 の場合は [Xamarin をダウンロード](#)

[I already have Visual Studio installed] をチェック

[Windows に Xamarin.Android をインストール : XLsoft エクセルソフト](#) を参考にインストールします

## Mac OS X

最新の Xcode と iOS SDK をダウンロード、インストールします

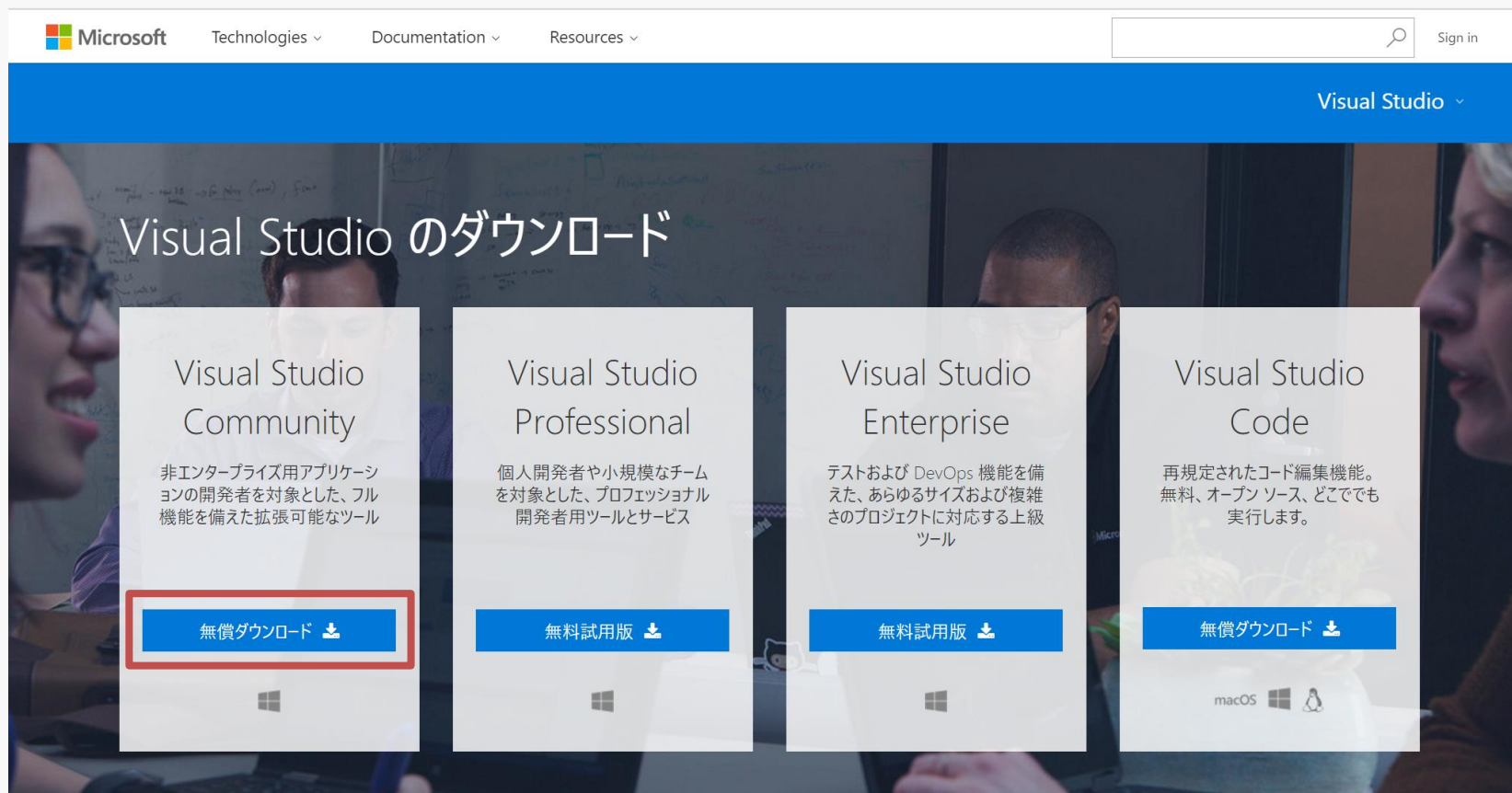
[Xamarinをダウンロードします](#)

[Mac に Xamarin.iOS をインストール : XLsoft エクセルソフト](#) を参考にインストールします



# Visual Studio 2015 Community のダウンロード

<https://www.visualstudio.com/ja/downloads/> にアクセスし、Community の「無料ダウンロード」をクリックします。Visual Studio の有償ライセンスをお持ちの方はそちらをご利用ください。



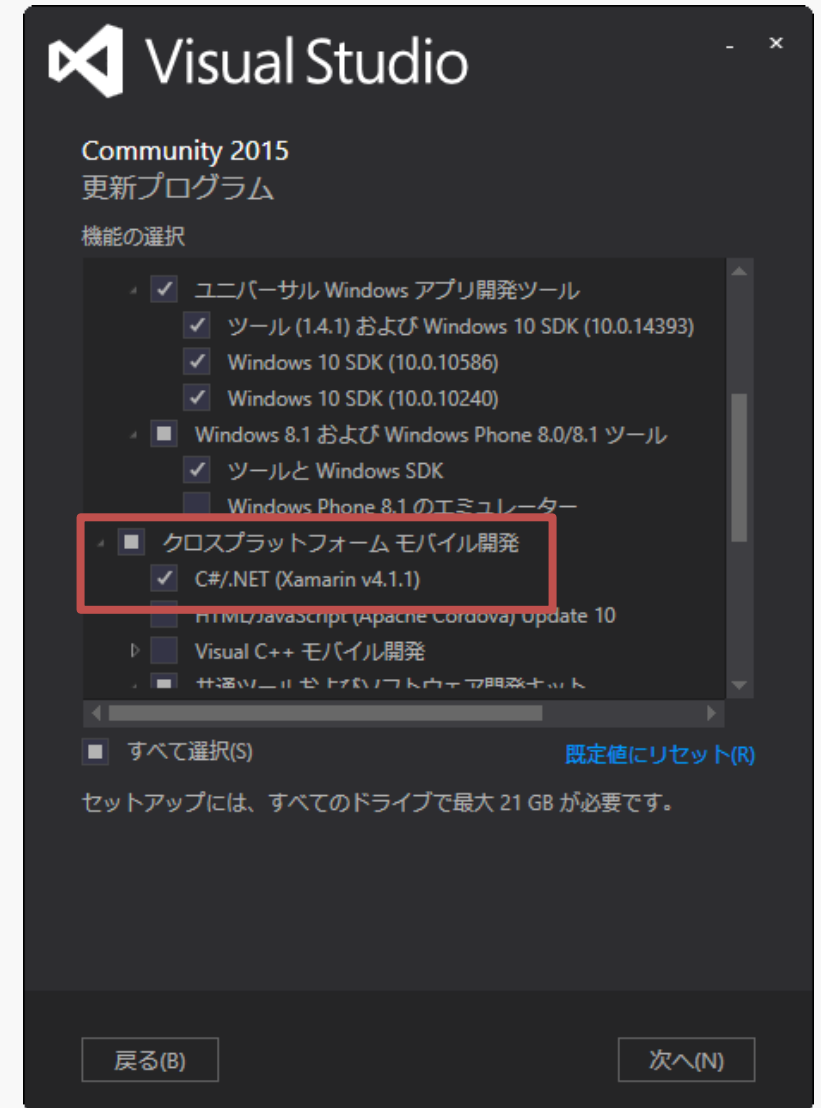
# Visual Studio インストール

Visual Studio 2015 には Xamarin のインストーラーが含まれています。

「カスタム」を選択し、「クロスプラットフォームモバイル開発 > C#/.NET (Xamarin v4.x)」をチェックしてインストールします。

自動的に、Android ネイティブ開発キット(Android NDK)、Android SDK、Android SDK(API 各種)、Java SE 開発キット(7.x)がインストールされます。

Visual Studio Emulator for Android はお勧めしません。(後述)



# Visual Studio インストール時の注意点 1

## Android SDK が既にインストールされている場合

Visual Studio のインストールで Xamarin をインストールせずに、後で Xamarin だけをインストールします。

<http://xamarin.com/download> にアクセスし、「I already have Visual Studio installed」で「Yes」を選択します。

Xamarin のインストーラーでインストールする際に、インストール済みの Android SDK の Path を指定してください。

Download Visual Studio.

Nice! You are about to download Visual Studio with Xamarin so you can write your apps entirely in C# and share the same code on iOS, Android, Windows, Mac and more.

If you already have Visual Studio, the installer will add Xamarin to your existing installation.

Yoshito Tabuchi

ytabuchi@xlsoft.com

XLsoft K.K.

I already have Visual Studio installed

Yes

☒ I agree to the [Terms of Use](#) and [Privacy Statement](#)

Download Now

[or download Xamarin Studio for OS X](#)

# Visual Studio インストール時の注意点 2

## Visual Studio Emulator for Android について

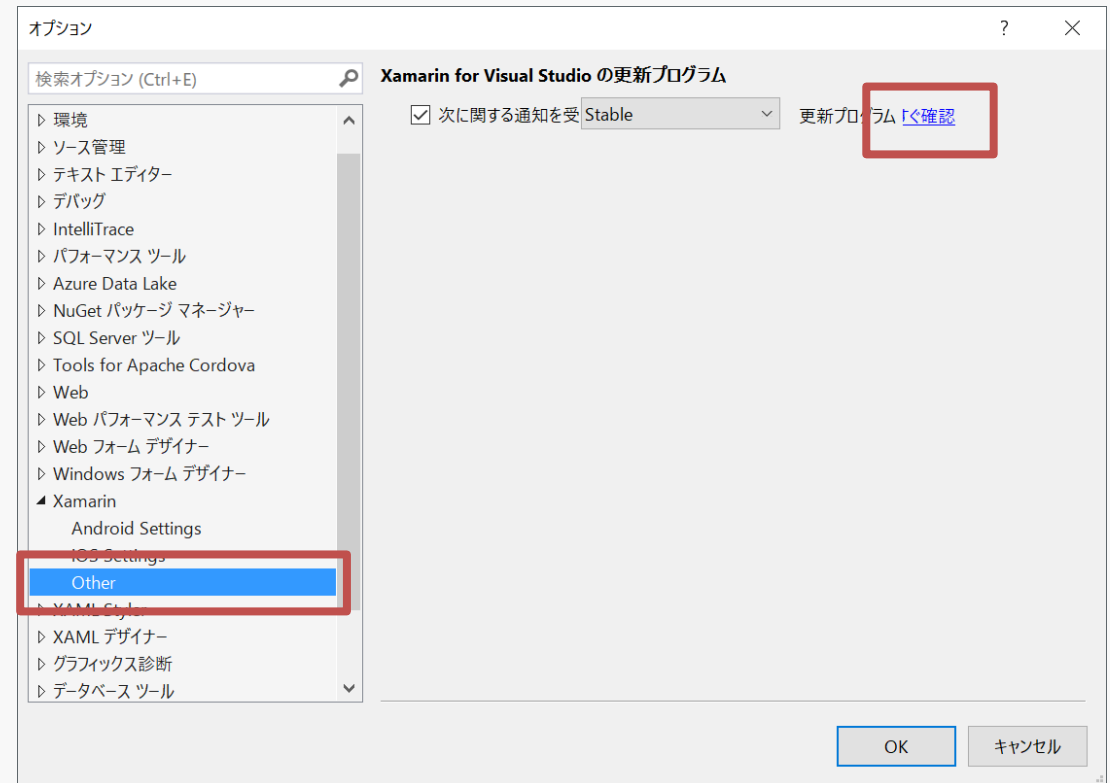
標準設定で 2GB のメモリを使用する点、HYPER-V が必須（Windows 10 Homeでは使えない）な点から、JXUG では使用をお勧めしません。

実機でのデバッグか、Android SDK に付属する Intel HAXM を使用した x86 Emulator でのデバッグを強くお勧めします。

Google Map や Ad などの Google Play Service を使用するアプリをデバッグする場合は、Google API のイメージを使用してください。

# Xamarin を最新版にアップデート

インストール後、Visual Studio のメニューから「ツール > オプション」で表示される「オプション」ダイアログで「Xamarin > Other > すぐ確認」をクリックして最新版にアップデートします。



# Xamarin Studio インストール

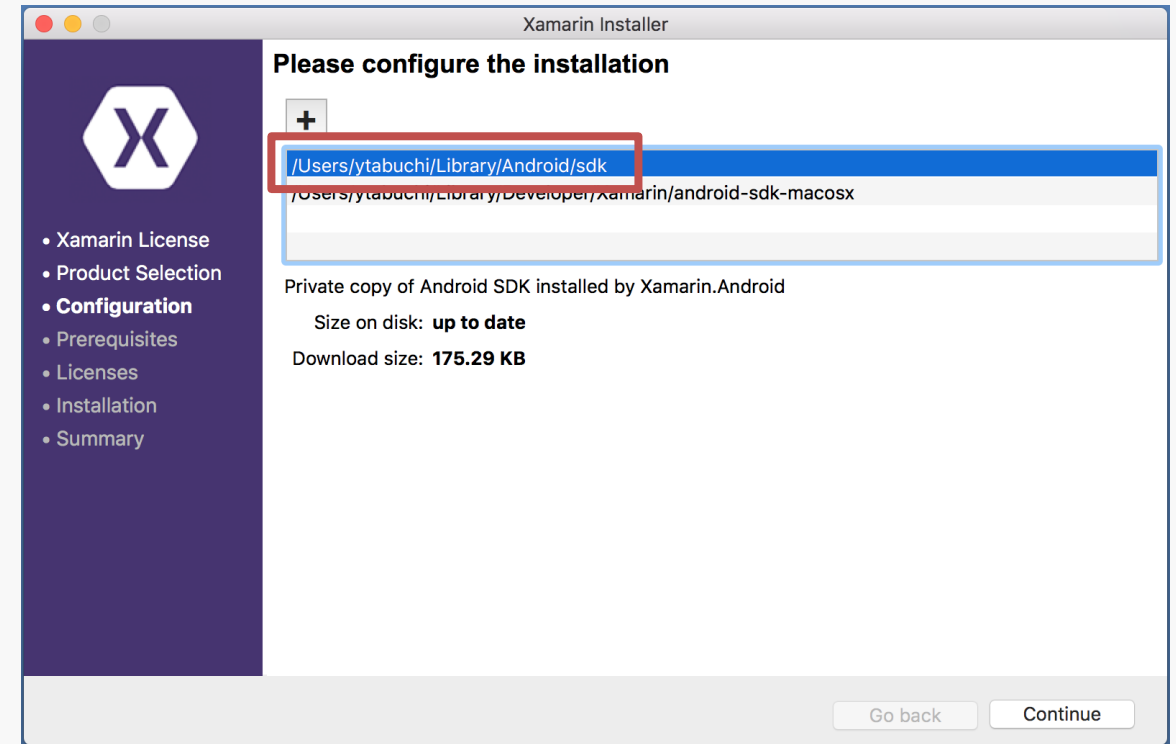
Xamarin Studio をインストールする前に、以下のシステム要件を満たしていることを確認してください。

macOS が Yosemite (10.10) 以上

Xcode の最新版が入っている

<http://xamarin.com/download> から Xamarin インストーラーをダウンロードしてインストールします。

既に Android SDK をインストール済みの場合は既存の SDK を指定します。認識していない場合は + ボタンでパスを指定します。



# Xamarin Studio の更新

Xamarin Studio をインストールしたらまずはメニューの「Xamarin Studio Community > 更新の確認」で最新版を入手、インストールします。



# Visual Studio for Mac / Visual Studio 2017 について

## Visual Studio for Mac

Xamarin Studio をリブランドしたもので、基本は Xamarin Studio と同じです。

Xamarin Studio との共存も可能ですが、インストールされているプロダクトによっては Xamarin Studio のアンインストールが必要になる可能性があるため、お勧めしません。

## Visual Studio 2017 RC

Visual Studio のインストーラーが、Visual Studio 2015 に含まれる Xamarin を**削除**します。

Visual Studio 2015 を使用しましょう。



# SDK更新や開発者登録

# Java 8 インストール

Android 7.0 Nougat から Java 8 がサポートされたため、API 24 以上をインストールする場合は Java 8 の JDK が必要です。

<http://www.oracle.com/technetwork/java/javase/downloads/index.html> から JDK をダウンロード、インストールしてください。

詳しくは <http://ytabuchi.hatenablog.com/entry/2016/10/03/180000> をご参照ください。

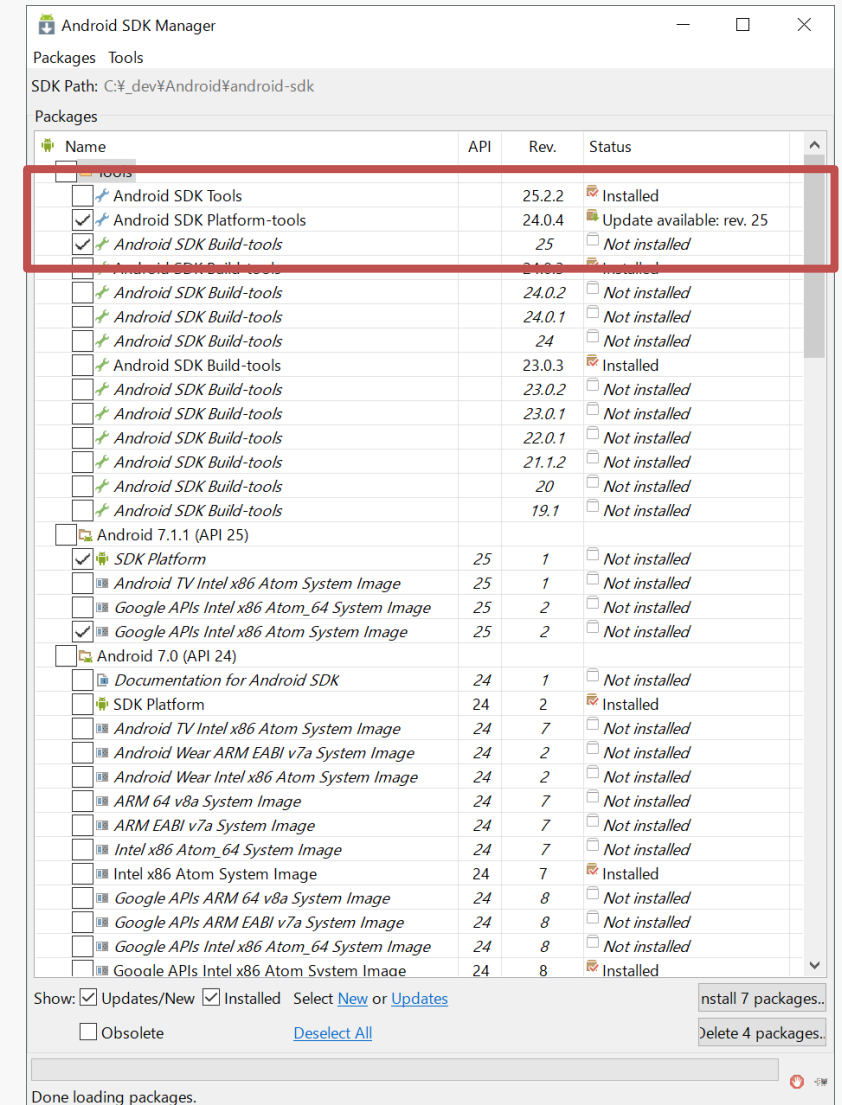
# Android SDK 更新

## Android SDK 更新

Visual Studio メニューから [ツール > Android > Android SDK Manager] をクリックして、  
[Android SDK Tools] [Android SDK Platform-tools] を最新にします。

[Android SDK Build-tools] の最新版をインストールします。

特定のバージョンでビルドできるようにするためには、最新バージョンと、2, 3個前のバージョンも入れておいても良いでしょう。2017年3月時点では、25, 24, 23, 22 を入れておくの良いと思います。(22 を入れておく Lollipop でビルドできるので、ランタイムパーミッションを回避してビルドできる...)



# Android SDK 更新

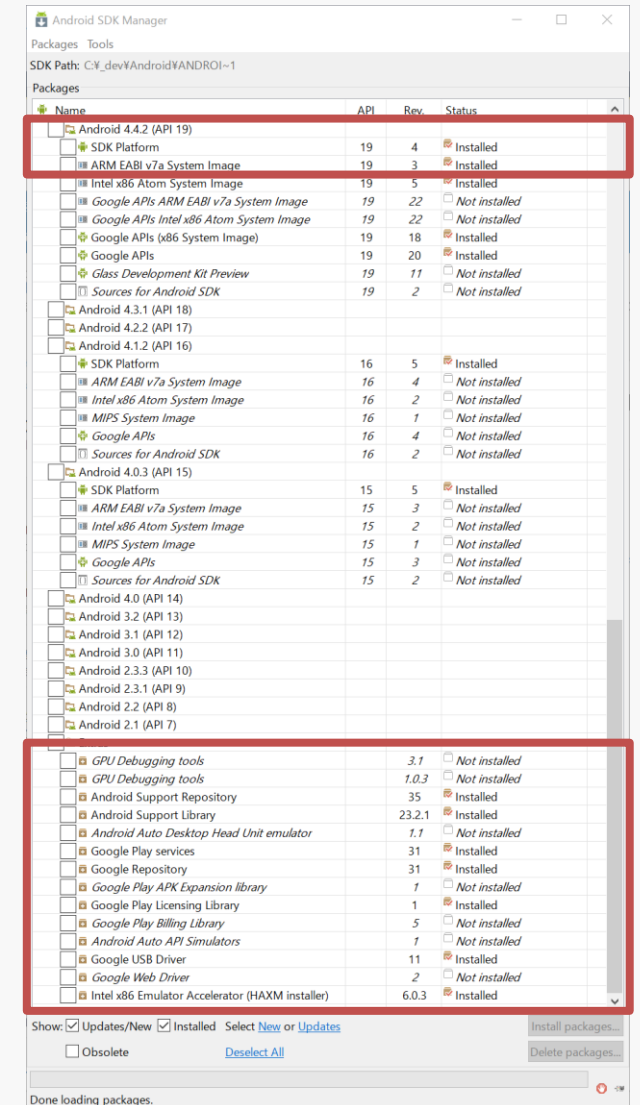
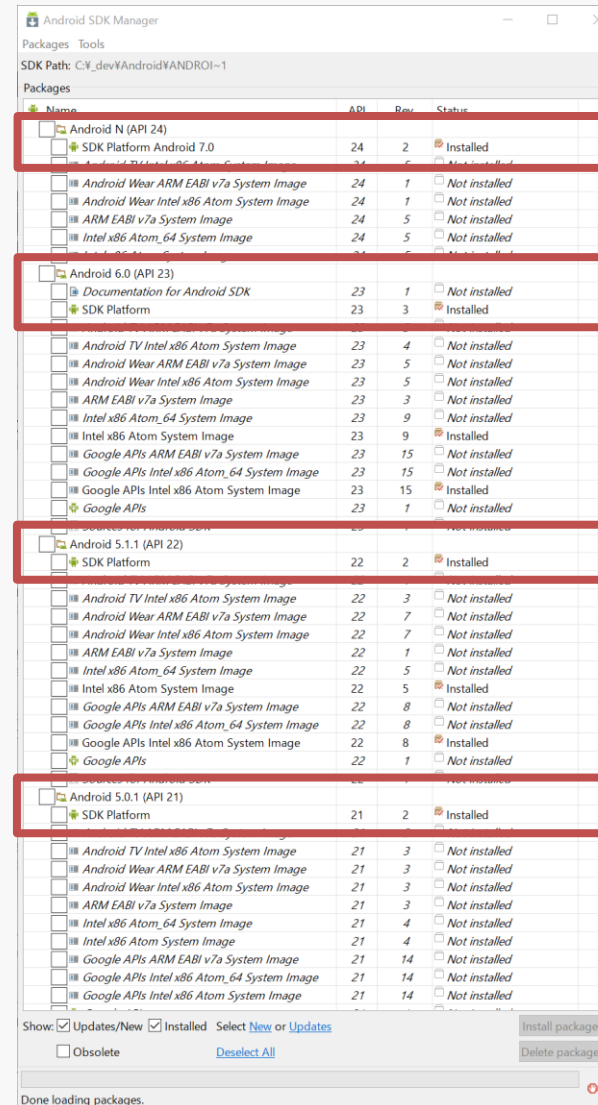
[SDK Platform] をインストールします。

7.1, 7.0, 6.0, 5.1, 5.0, 4.4

Extra をインストールします。

Android Support Repository,  
Android Support Library, Google  
Play services, Google Repository,  
Google USB Driver

その他の Library は必要に応じてインストール



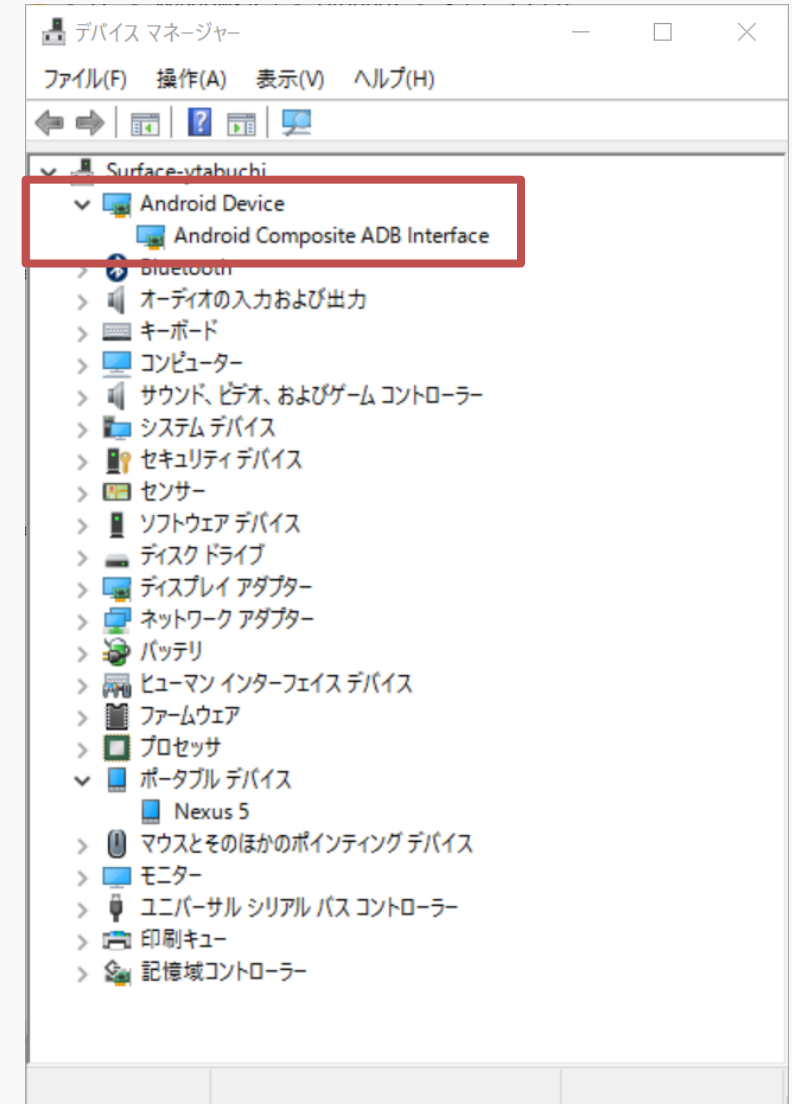
# 実機用ドライバーインストール（Windowsのみ）

## Android Device の確認

実機を使用する場合は、デバイスマネージャーを開き、  
「Android Device」に「Android Composite ADB Interface」があるか確認します

無い場合は、お持ちの Android 端末のメーカーサイトから、ドライバーをダウンロードしてインストールしてください

Nexus シリーズの端末は[こちら](#)からダウンロードできます



# Android 端末を開発者モードに

## 開発者モードの確認

Android 端末の「設定」画面を開きます。「開発者向けオプション」があるかを確認します

無い場合は、「端末情報」(に準じたメニュー)をタップして、「ビルド番号」を7回タップして開発者モードにします



# Android Emulator を使用する場合

ご利用の仮想環境によって最適な Emulator を選択してください。  
詳しくは以下をご参照ください。

Windows で動作する Android 仮想環境の比較まとめ

2016年10月現在では、Google 純正の x86 Emulator を強くお勧めします。インストール方法は以降のスライドをご参照ください。

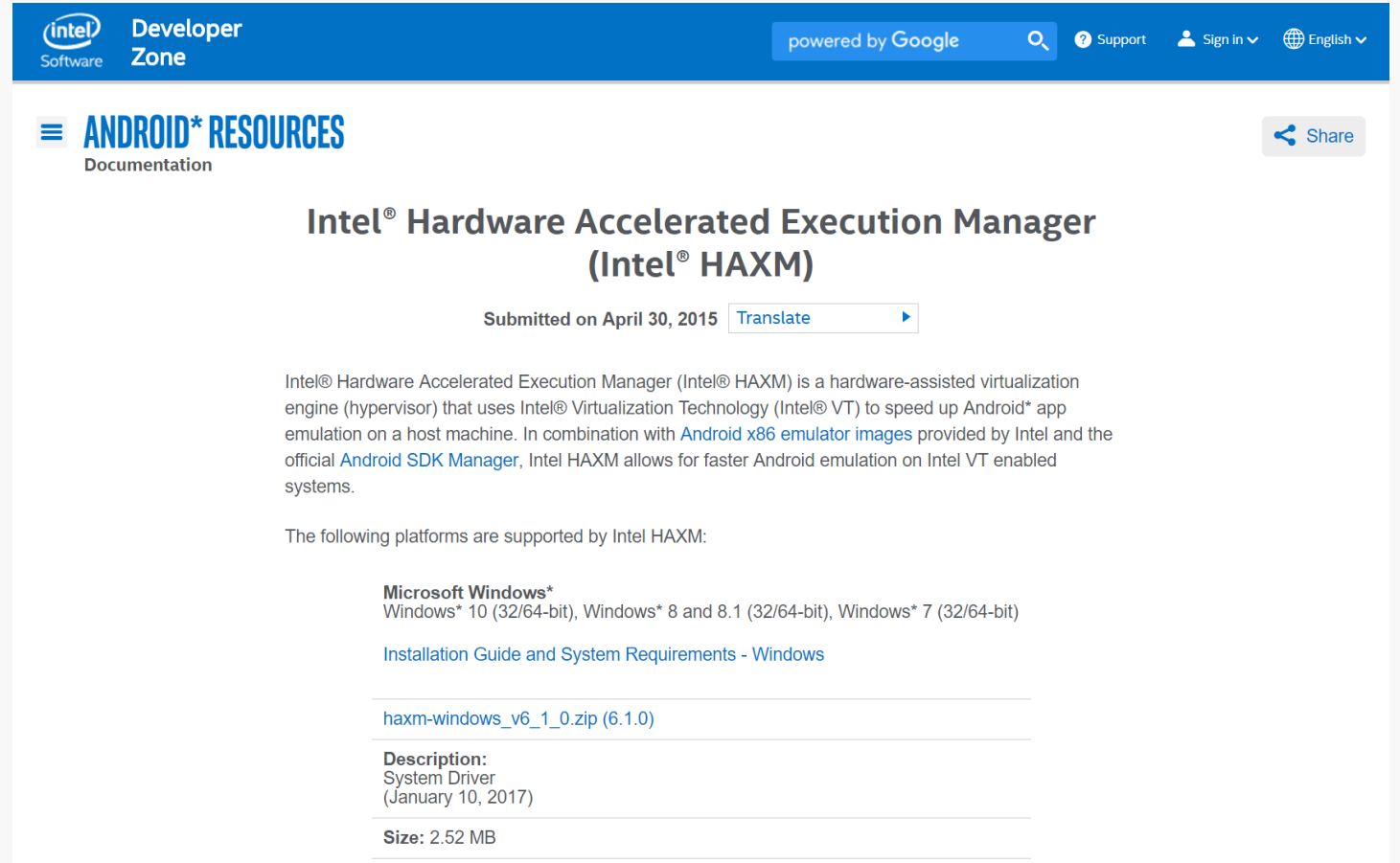
# Intel x86 Emulator Accelerator

## Intel x86 Emulator Accelerator (HAXM Installer) をインストール

SDK Manager の一番下の Intel x86 Emulator Accelerator (HAXM Installer) が、2017年3月現在ではインストールできなくなっています。

直接[インテルのサイト](https://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manager)からダウンロード、インストールします。

<https://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manager>



The screenshot shows the Intel Developer Zone page for the Intel Hardware Accelerated Execution Manager (Intel HAXM). The page is titled "Intel® Hardware Accelerated Execution Manager (Intel® HAXM)" and is part of the "ANDROID\* RESOURCES" section. It includes a "Submitted on April 30, 2015" date and a "Translate" button. The main text describes HAXM as a hardware-assisted virtualization engine (hypervisor) that uses Intel® Virtualization Technology (Intel® VT) to speed up Android\* app emulation on a host machine. It also mentions that HAXM allows for faster Android emulation on Intel VT enabled systems. The page lists supported platforms: Microsoft Windows\* (Windows\* 10 (32/64-bit), Windows\* 8 and 8.1 (32/64-bit), Windows\* 7 (32/64-bit)). There is a link to the "Installation Guide and System Requirements - Windows" and a download link for "haxm-windows\_v6\_1\_0.zip (6.1.0)". The description states it is a "System Driver" and was updated on "January 10, 2017". The size is listed as "2.52 MB".

Intel Software Developer Zone powered by Google Support Sign in English

≡ ANDROID\* RESOURCES Documentation Share

### Intel® Hardware Accelerated Execution Manager (Intel® HAXM)

Submitted on April 30, 2015 [Translate](#)

Intel® Hardware Accelerated Execution Manager (Intel® HAXM) is a hardware-assisted virtualization engine (hypervisor) that uses Intel® Virtualization Technology (Intel® VT) to speed up Android\* app emulation on a host machine. In combination with [Android x86 emulator images](#) provided by Intel and the official [Android SDK Manager](#), Intel HAXM allows for faster Android emulation on Intel VT enabled systems.

The following platforms are supported by Intel HAXM:

- Microsoft Windows\***  
Windows\* 10 (32/64-bit), Windows\* 8 and 8.1 (32/64-bit), Windows\* 7 (32/64-bit)

[Installation Guide and System Requirements - Windows](#)

[haxm-windows\\_v6\\_1\\_0.zip \(6.1.0\)](#)

**Description:**  
System Driver  
(January 10, 2017)

**Size:** 2.52 MB



# x86 Emulator

右の例を参考に AVD を作成します。

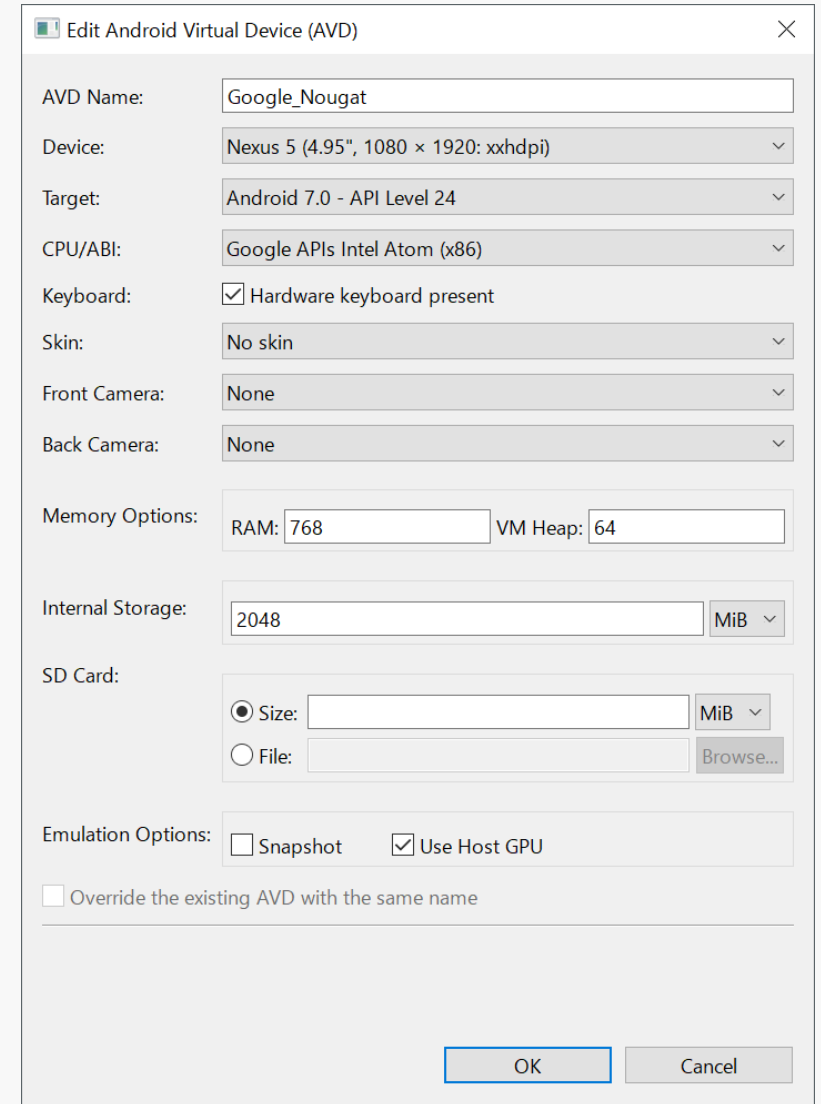
CPU/ABI を「Intel Atom (x86)」か「Google APIs Intel Atom (x86)」にします。

Memory Options を 768MB にします。(Windows の場合は 768MB 以上にすると問題起きるよ！という Warning が表示されているはず)

Internal Storage を 1GB や 2GB にします。

Xamarin.Android のアプリは Debug ビルドで Shared Runtime や各 API を先に端末にインストールしますが、合計で 50MB 以上あるので、標準で指定されている 200MB ではとてもじゃないけど足りません。

Use Host GPU をチェックします。



AVD Name: Google\_Nougat

Device: Nexus 5 (4.95", 1080 × 1920: xxhdpi)

Target: Android 7.0 - API Level 24

CPU/ABI: Google APIs Intel Atom (x86)

Keyboard: ☒ Hardware keyboard present

Skin: No skin

Front Camera: None

Back Camera: None

Memory Options: RAM: 768 VM Heap: 64

Internal Storage: 2048 MiB

SD Card: ☒ Size: MiB ☐ File: Browse...

Emulation Options: ☐ Snapshot ☒ Use Host GPU

☐ Override the existing AVD with the same name

OK Cancel

# Apple Developer Program 登録 (Mac)

## Apple Developer Program 登録

開発した iOS アプリを iOS デバイスにデプロイするには、Apple Developer Program への加入をお勧めします。iOS Simulator へのデプロイであれば加入不要です。

加入後は [Xcode 6.x で実機デバッグするまでの簡単な流れ](#) を参照して、Provisioning Profile 登録してください。

現在は Free Provisioning でもデプロイ出来るようになりましたが、Xamarin Studio, Visual Studio では手間がかかります。

詳しくは [Xcode 7 と Xamarin Studio Starter で 1 円も払わずに自作 iOS アプリを実機確認する](#) を参照してください。

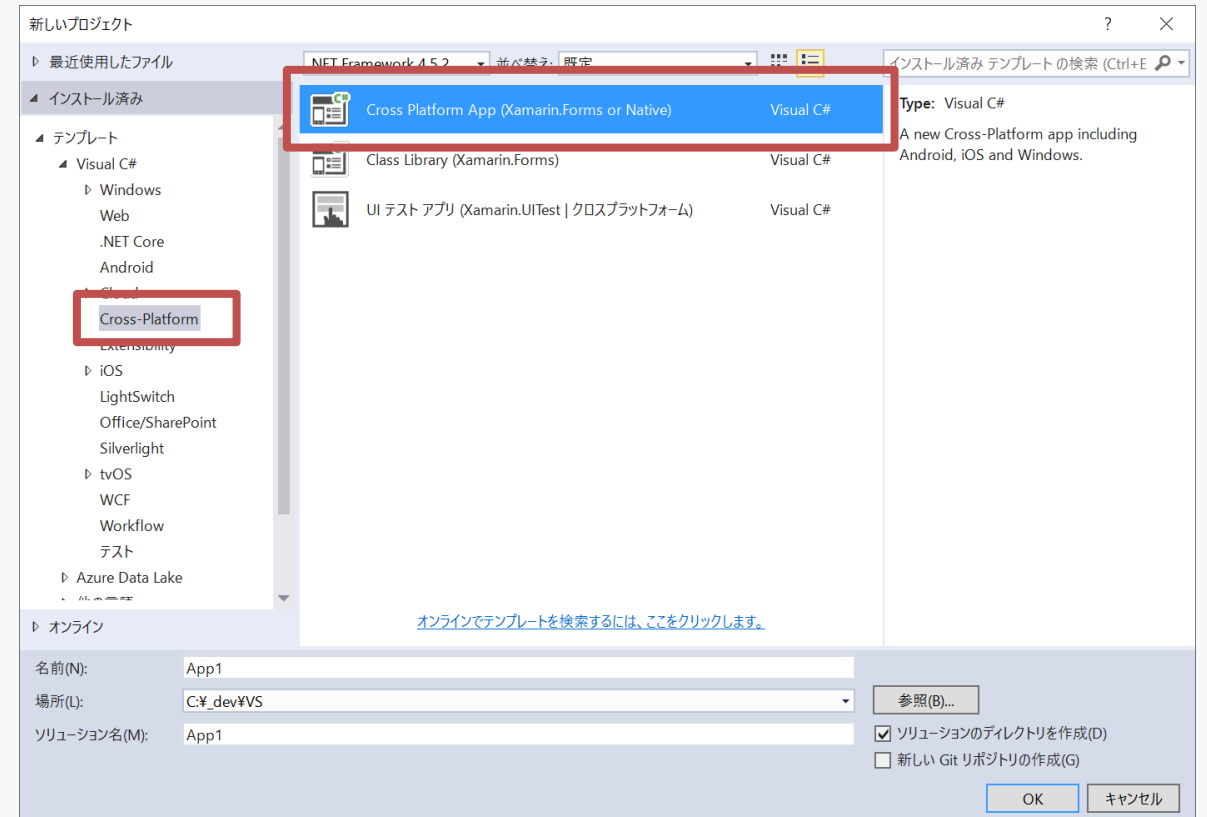
# Xamarin でビルドのテスト

# Xamarin.Forms アプリを作成 (Windows) - 1

## Blank App 作成

Visual Studio メニューの「ファイル > 新規作成 > プロジェクト」をクリックし、  
「新しいプロジェクト」ダイアログで  
「Visual C# > Cross-Platform」を選択  
します。

中央のテンプレートから「Cross Platform  
App (Xamarin.Forms or Native)」を選  
択して、[OK] をクリックします。



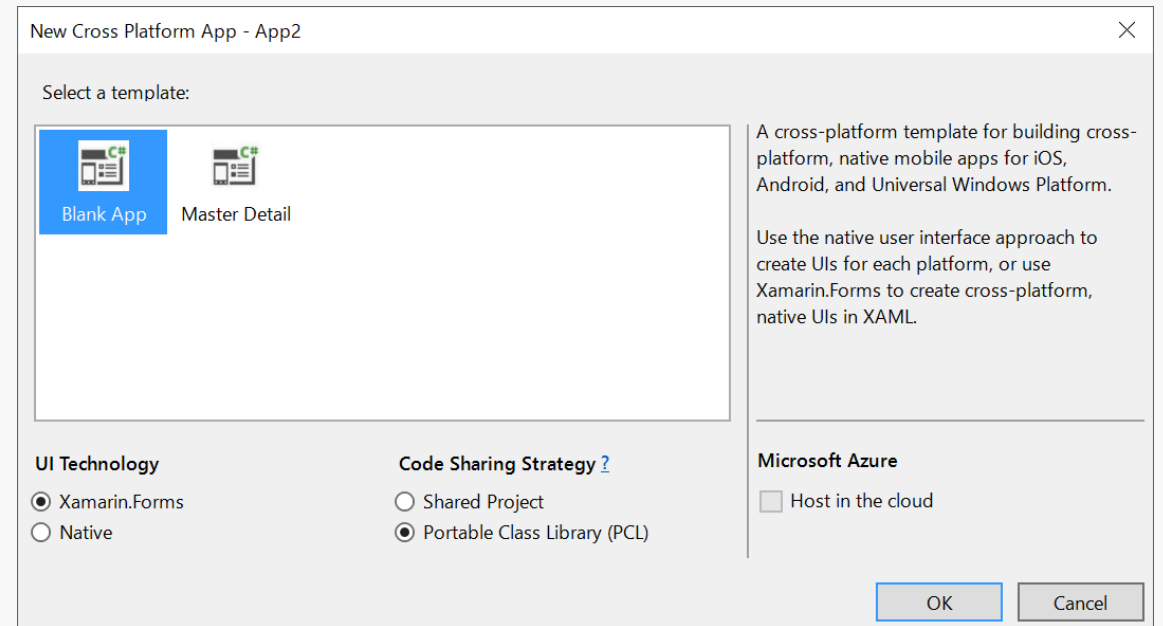
# Xamarin.Forms アプリを作成 (Windows) - 2

作成するアプリのタイプを選択します。

Templateから「Blank App」を選択  
UI Technologyから「Xamarin.Forms」を選択

Code Sharing Strategyから任意の方法を選択します。ここでは「Portable Class Library (PCL)」を選択しています。

(2017年3月時点で Xamarin は「Shared Project」を推しているようです。.NET Standard への移行期間のため、PCL で動作しないライブラリが多いためだと思われます。)



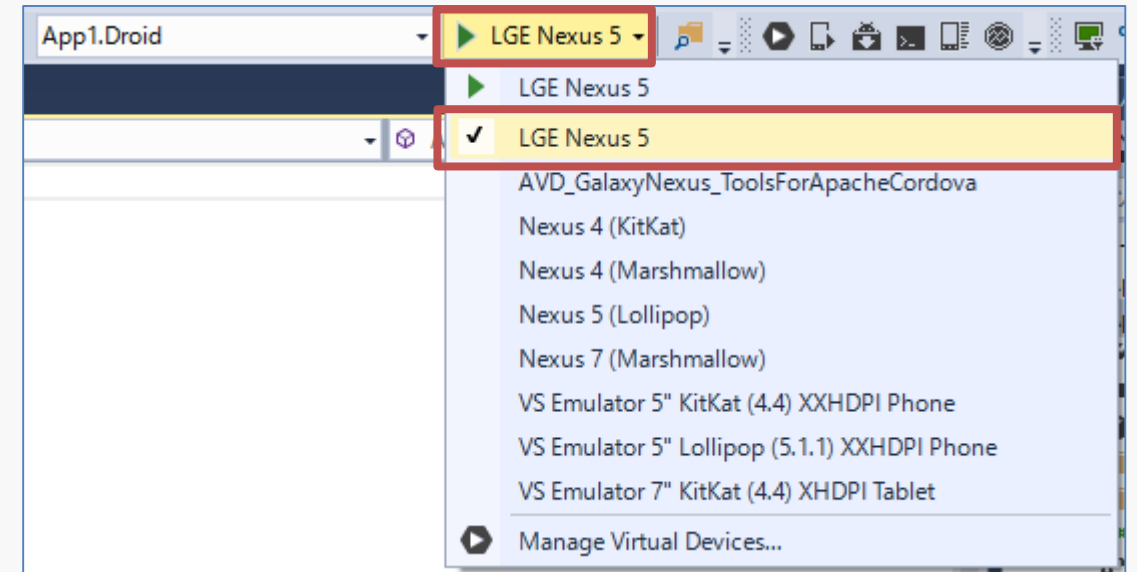
# Xamarin.Forms アプリをデバッグ実行 (Windows)

## デバッグ実行

ドロップダウンから実機または Emulator を選択し、[▶] ボタンでデバッグ実行します。

Xamarin.Forms の Android アプリをビルドすると、初回に必要な Android Support Library をダウンロードするため、かなり時間がかかります。

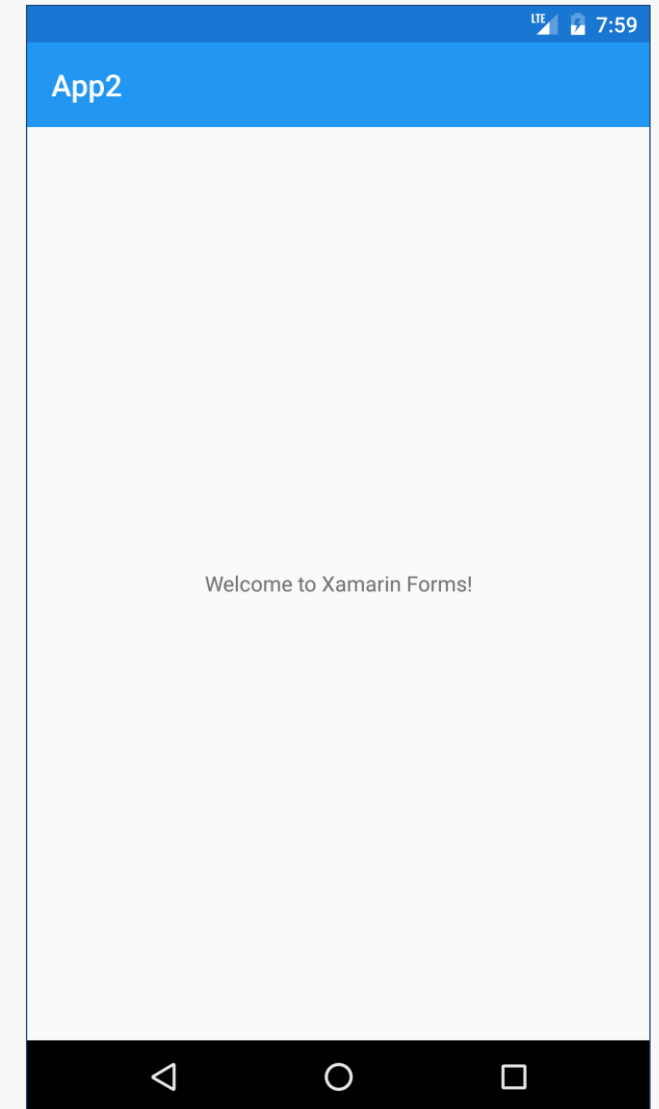
ビルドを停止してしまうとその後のビルドが失敗する場合がありますので、終了までじっくり待ってください。



# Xamarin.Forms アプリをデバッグ実行 (Windows)

次の画面が実機に表示されれば完了です

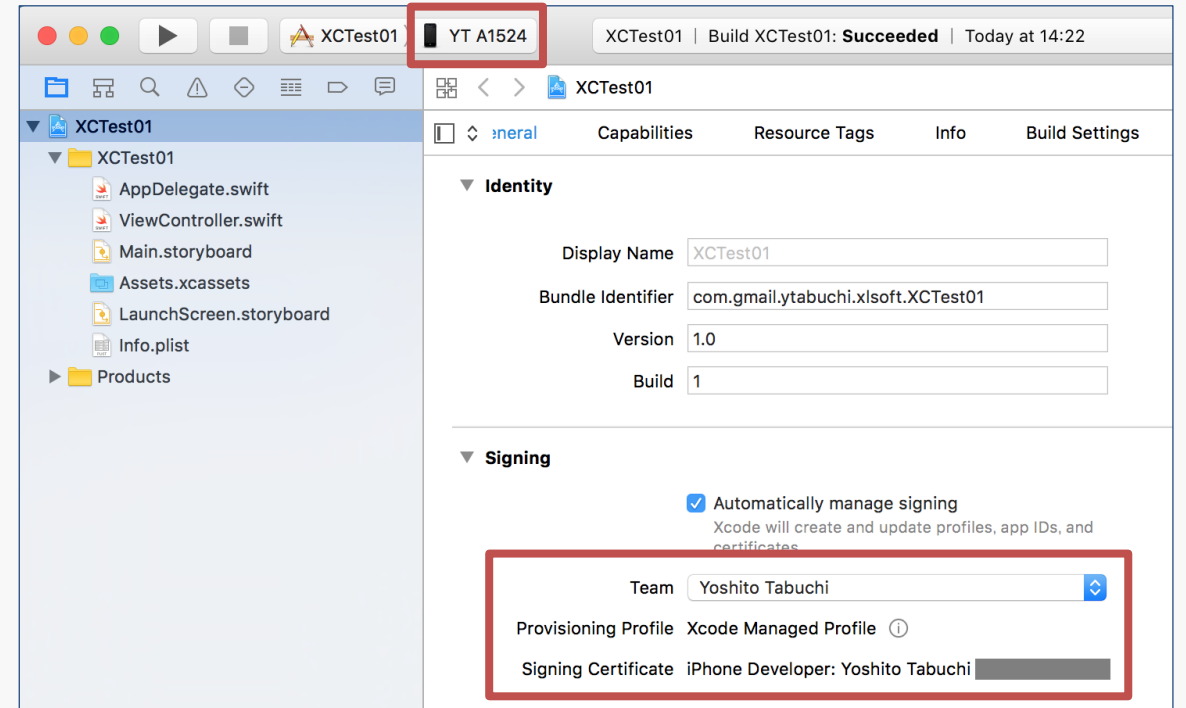
写真は Android 6.0 がインストールされた Nexus 5 の実機



# Xamarin Studio から iOS デバイスにデプロイ (Mac)

まず最初に Xcode で正しいプロビジョニングプロファイルで実機にデプロイできることを確認します。

次に Xamarin Studio で作業をします。

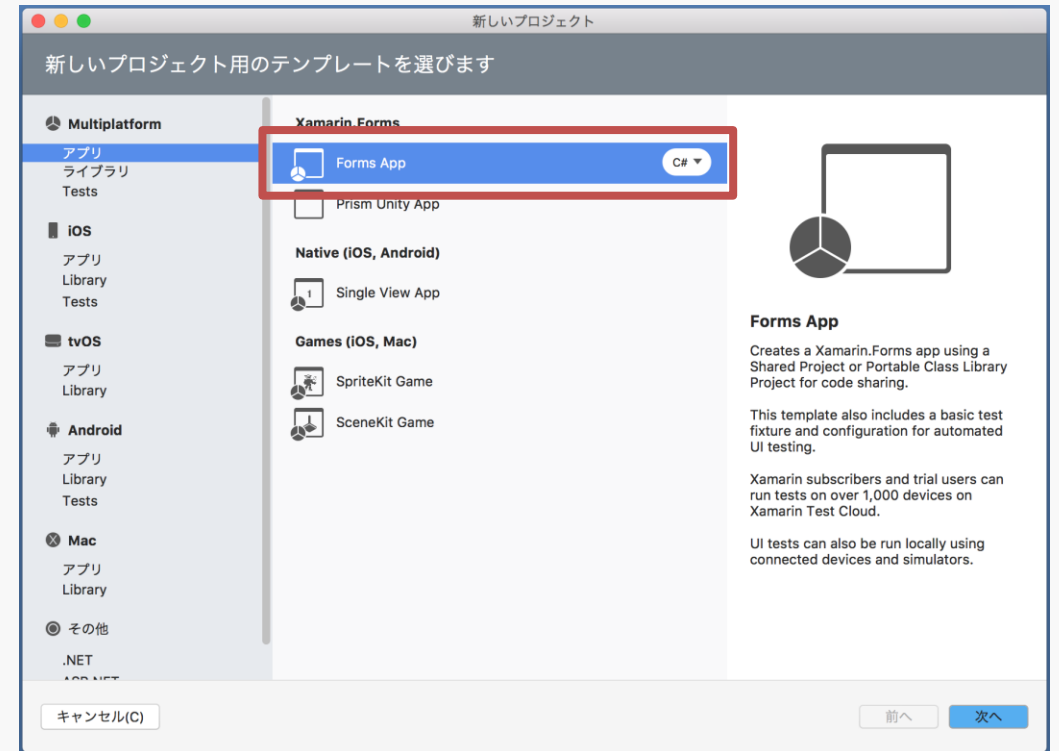




# Xamarin.Forms アプリを作成 (Mac)

メニューから [ファイル > 新しいソリューション] をクリックします。

「Multiplatform」の「アプリ」から「Forms App」を選択し、[次へ] をクリックします。

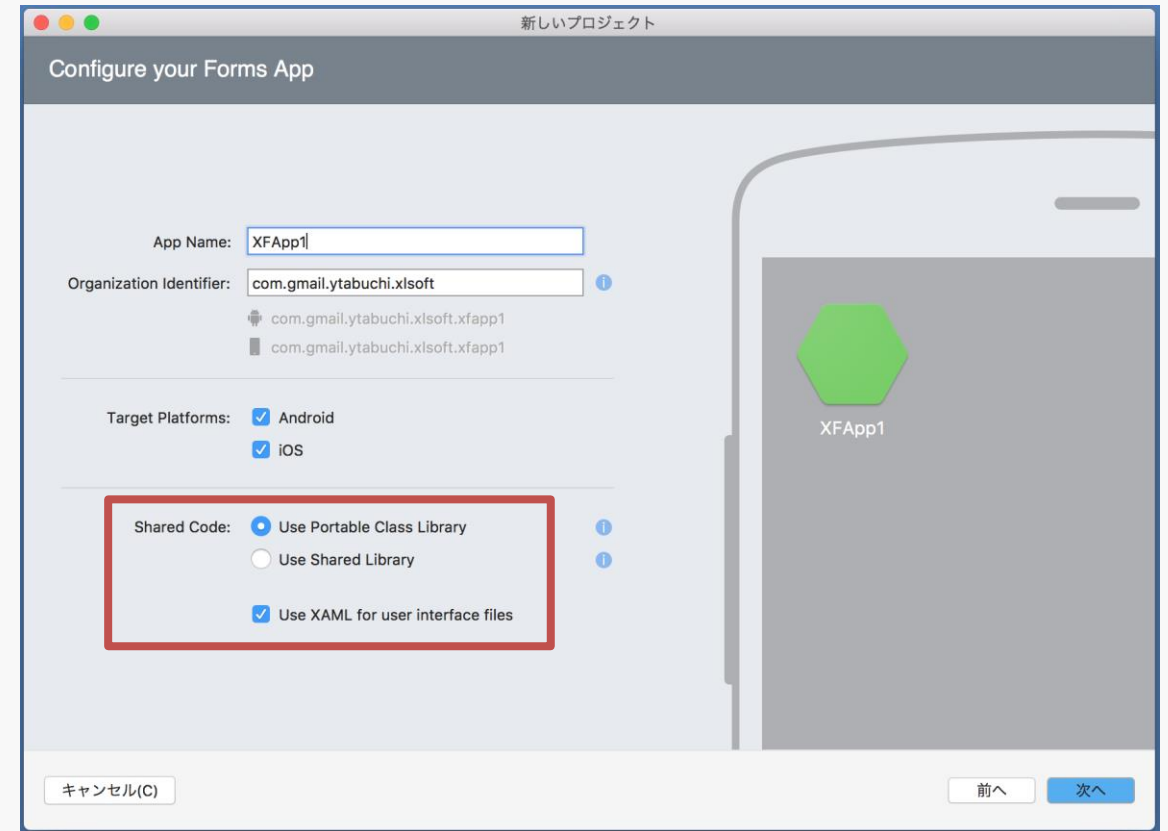


# Xamarin.Forms アプリを作成 (Mac)

「App Name」に任意の名前を入力します。

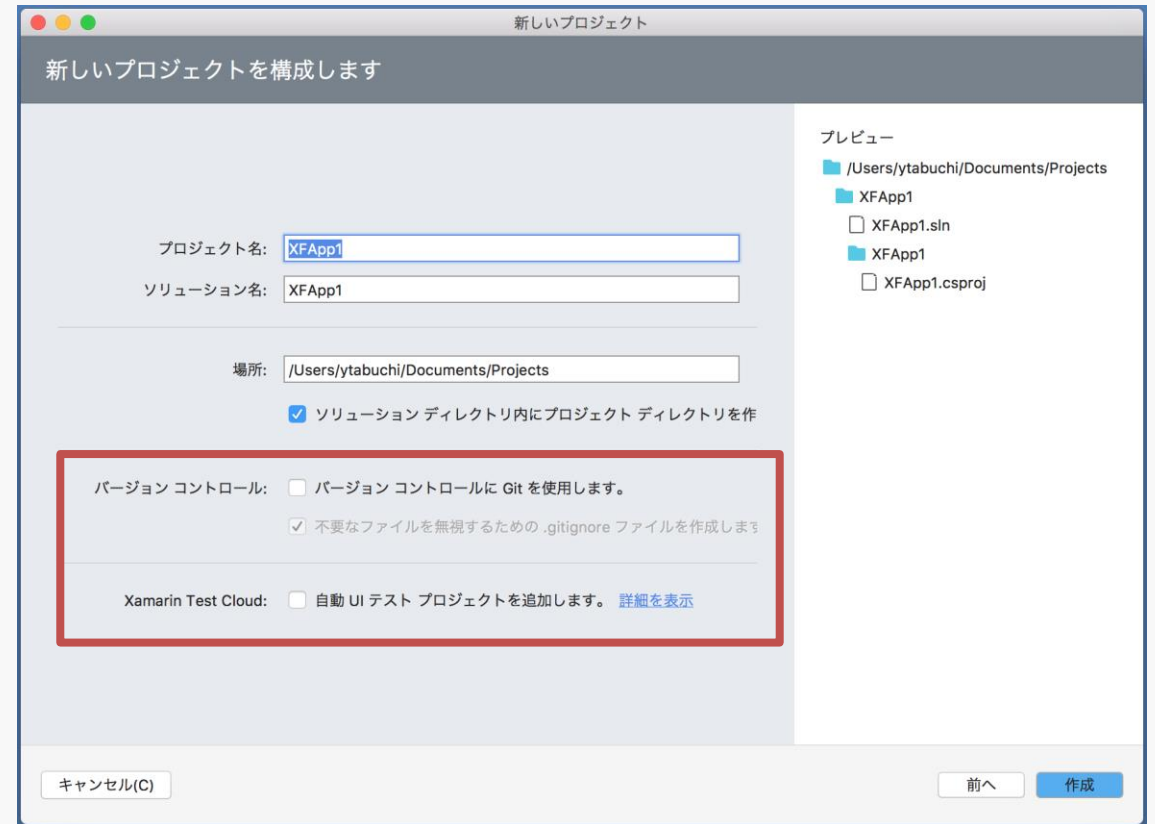
「Shared Code」欄の「User Portable Class Library」にチェックが入っていることを確認し [次へ] をクリックします

「Use XAML for user interface files」にチェックが入っていると、App.xaml／App.xaml.cs のエントリーポイントが作成されます



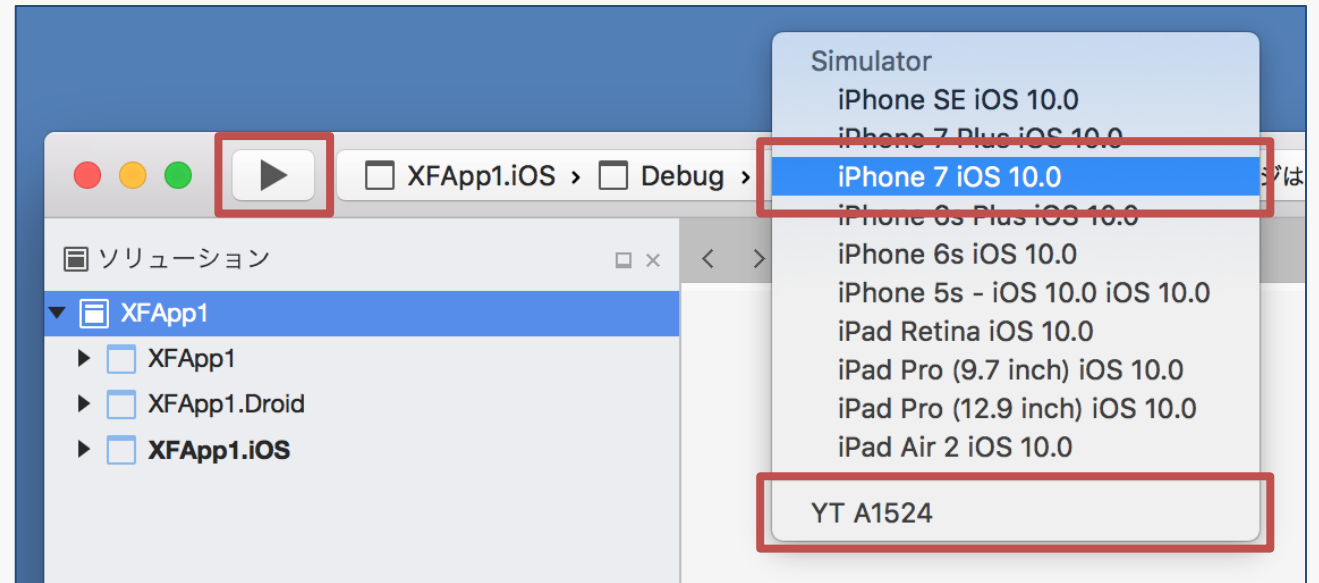
# Xamarin.Forms アプリを作成 (Mac)

「バージョンコントロール」と「Xamarin Test Cloud」のチェックを外し、[作成] をクリックします



# Xamarin.Forms アプリを作成 (Mac)

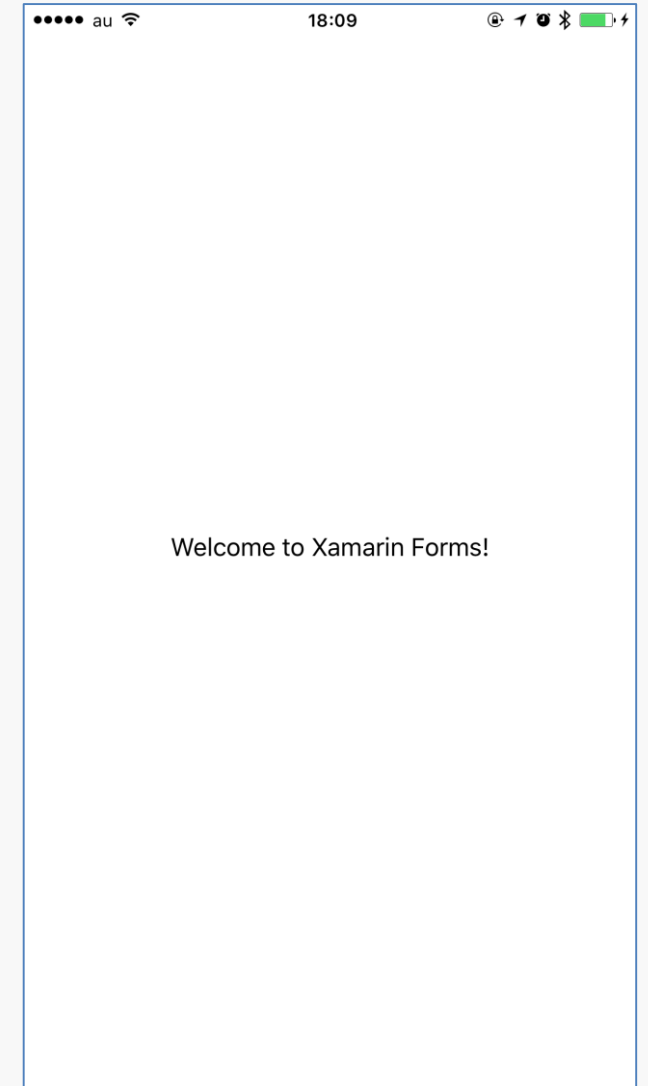
「Debug」の右側のドロップダウンから接続している実機または Simulator を選択し、[▶] でビルド、デプロイを開始します。



# Xamarin.Forms アプリを作成 (Mac)

次の画面が実機または Simulator に表示  
されれば完了です

写真は iPhone 6 Plus の実機



# Microsoft Azure アカウントの準備

Azure のハンズオンをやる方は必ず確認してください

# Microsoft Azure アカウントについて

Xamarin のハンズオンには Microsoft Azure を使用するコースが含まれているため、何らかの Azure のアカウントが必要です。

## MSDN

[MSDN Subscription](#) をお持ちの方は、そのアカウントを使用してください。

## 学生

[Microsoft Imagine \(旧DreamSpark\)](#) を利用してください。

## Visual Studio Dev Essentials

登録無料の開発者用プログラムです。1年間毎月\$25(現在は3,000円) のAzureが利用できて、更に Xamarin ユーザーには嬉しい Syncfusion も1年間無料で使えます。



# Visual Studio Dev Essentials 登録

<https://www.visualstudio.com/ja/dev-essentials/> にアクセスし、  
「今すぐ参加またはアクセス」をクリックします。





# Azure アクティベート その 1

Azure のアクティブ化をクリックします。

The screenshot shows the Visual Studio Dev Essentials website. The top navigation bar includes "Visual Studio", "My Benefits", and user information "Yoshito Tabuchi" with a "Sign out" link. Below the navigation bar, there are tabs for "特典", "ダウンロード", "プロダクト キー", "サブスクリプション", "サポート", and "Marketplace". The main header area displays "表示中: Visual Studio Dev Essentials" and "Visual Studio Dev Essentials へ ようこそ". Below this, a section titled "おすすめ情報 (6)" features six tiles. The "Azure" tile, which includes the text "Azure 毎月 25 ドルのクレジット (1 年間)" and "自分専用の開発/テスト用サンドボックスです! VM、クラウド サービス、その他。クレジットを既存の Aure ...", is highlighted with a red rectangular box. The button "アクティブ化" (Activate) is visible at the bottom of this tile. Other tiles include "Visual Studio Community", "Visual Studio Code", "Visual Studio Team Services", "Xamarin University トレーニング", and "Pluralsight". At the bottom of the page, a section titled "ツール (11)" is partially visible.

# Azure アクティベート その 2

電話番号を入力し、SMS か音声でコードを受け取ります。

1

⊖ 電話による本人確認

日本 (+81)

テキスト メッセージを受信

電話で確認コードを受け取る

コードの確認

# Azure アクティベート その 3

クレジットカード情報を入れます。勝手に課金されることはありませんのでご安心ください。

2

⊖ 支払情報 ⓘ

支払い方法

新しいクレジットカード/デビットカード ▾

VISA

MasterCard

JCB

AMERICAN EXPRESS

\* カード番号

\* 有効期限

2017 ▾

\* セキュリティコード ⓘ

\* クレジットカードの名義

YOSHITO TABUCHI

\* 郵便番号

\* 都道府県

▾

\* 市区町村

\* 住所 1

住所 2

- オプション -

電話番号

- 市外局番 -

- 番号 -

次へ

# Azure アクティベート その 4

購入をクリックしてしばらく待ちます

## 3 ① 契約

☒ サブスクリプション契約、プランの詳細、プライバシーに関する声明に同意します。

購入 ➡

# Azure アクティベート その5

「サービスの管理を開始する」をクリックして Azure Portal に移動します。

Microsoft Azure へようこそ。

サブスクリプション - 開発者プログラム特典

サブスクリプションの準備が整いました。

サービスの管理を開始する >

For Technical support, buy a support plan

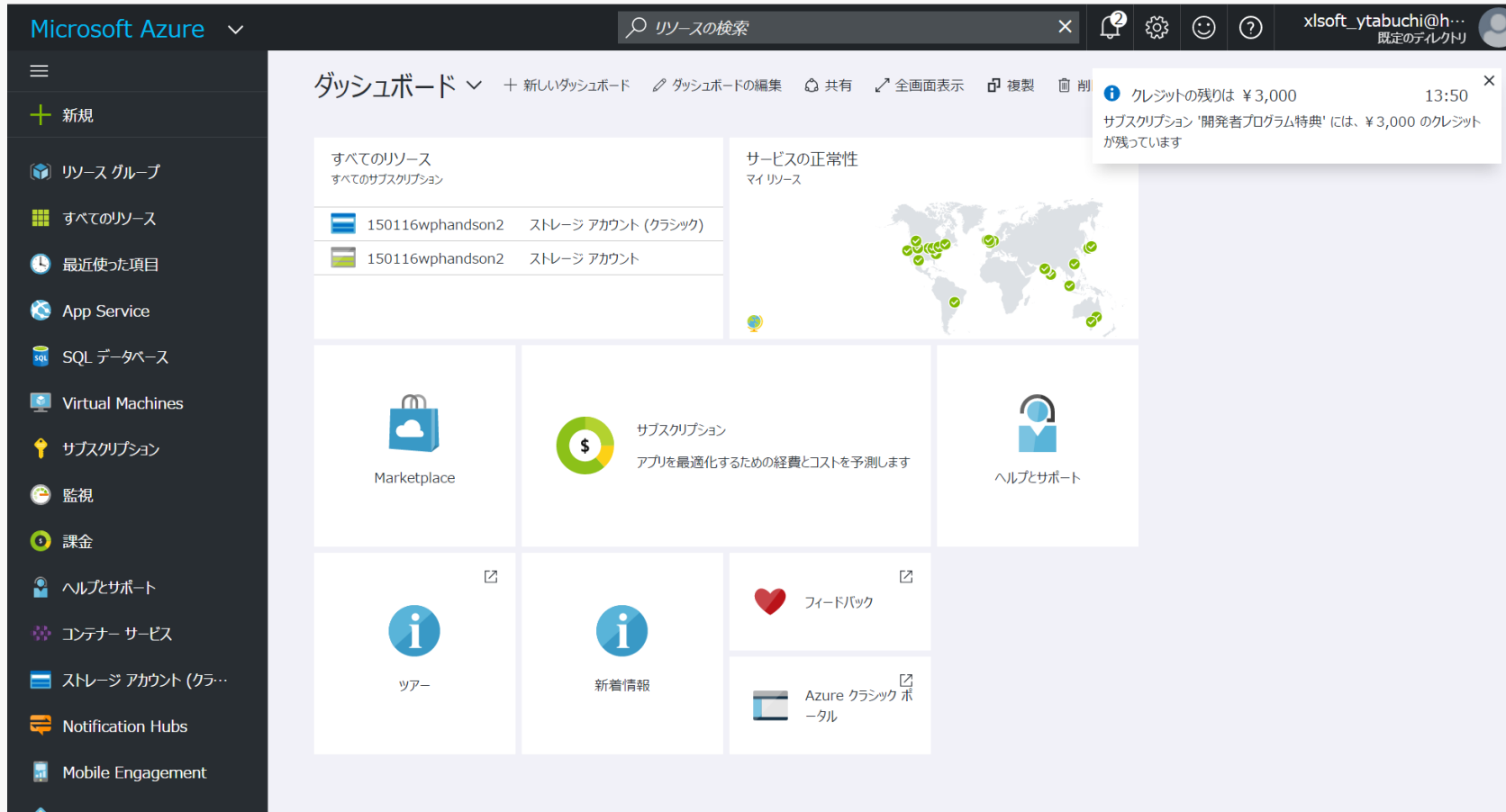
Your subscription includes unlimited 24 x 7 billing and subscription support.

For one-on-one technical support, you'll need a support plan.

Compare support plans ▶

# Azure Portal

<http://portal.azure.com> にアクセスできるようになりました(^ ^)



お疲れ様でした