Installing Xamarin.iOS on Windows

Contents

Overview

Xamarin.iOS for Visual Studio allows iOS applications to be written and tested on Windows computers, with a networked Mac providing the build and deployment service.

Developing for iOS inside Visual Studio lets you:

- Create cross-platform solutions for iOS, Android, and Windows applications.
- Use Visual Studio tools (such as *Resharper* and *Team Foundation Server*) for all your cross-platform projects, including iOS source code.
- Work with a familiar IDE, while taking advantage of Xamarin.iOS bindings of all Apple's APIs

Xamarin.iOS for Visual Studio supports configurations in which Visual Studio is running inside a Windows virtual machine on a Mac (using Parallels or VMWare), or when it is on a separate machine that is visible on the same network as a Mac. Regardless of which configuration works best for you, Visual Studio will connect to the Mac using SSH via the *Xamarin Mac Agent* (often referred to as XMA).

This article covers the steps to install and configure the Xamarin.iOS tools on both the Mac and Windows machine, as well as using the *Xamarin Mac Agent* so that we can build, debug, and deploy Xamarin.iOS applications using Visual Studio.

Requirements

Xamarin.iOS for Visual Studio accomplishes an amazing feat: it lets you create, build, and debug iOS applications on a Windows computer using the Visual Studio IDE. It cannot do this alone – iOS applications cannot be created without Apple's compiler, and they cannot be deployed without Apple's certificates and code-signing tools. This means that your Xamarin.iOS for Visual Studio installation requires a connection to a networked Mac OS X computer in order to perform these tasks. Once configured, Xamarin's tools will make the process as seamless as possible.

System Requirements

The system requirements are:

Windows

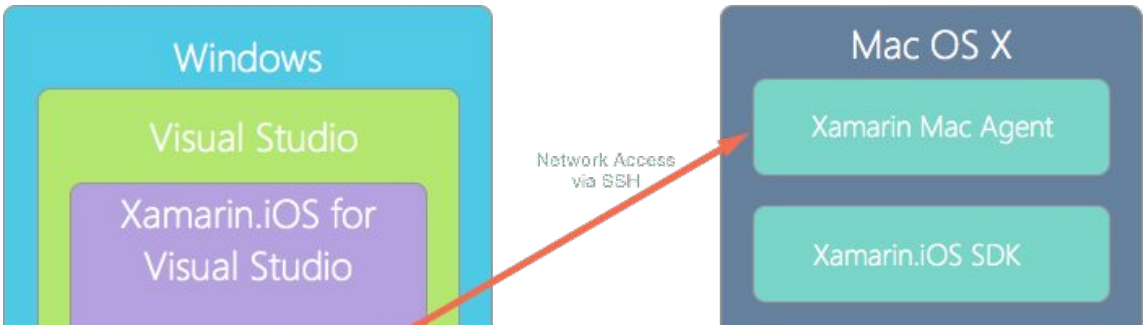
1. Windows 7 or higher.
2. Visual Studio 2012 Professional or higher.
3. Xamarin for Visual Studio.

You cannot use the Xamarin plug-in with Express editions of Visual Studio due to lack of plug-in support

Macintosh

1. A Mac running OS X Yosemite (10.10) or higher (although we recommend the latest stable version).
2. Xamarin iOS SDK.
3. Apple's Xcode(7+) IDE and iOS SDK (we recommend the latest version from the App Store).

The Windows computer must be able to reach the Mac via the network.



Apple Developer Account

To deploy applications to a device or to submit them to the App Store, an Apple Developer account is required. The relevant developer certificates and provisioning profiles must be created and installed on the networked Mac before Xamarin.iOS for Visual Studio can work. See the Device Provisioning article for steps to obtain a development certificate and to provision a device.

Features and limitations

Features

Xamarin.iOS for Visual Studio enables the creation, editing, building, and deployment of Xamarin.iOS projects from Windows. This includes the following features:

- Create new iOS projects.
- Edit iOS projects and cross-platform solutions that also include Xamarin.Android and Windows Phone projects.
- Compile iOS projects and cross-platform solutions that also include Xamarin.Android and Windows Phone projects.
- Storyboard and .xib support using the iOS Designer.
- Deploy and debug iOS applications, where the app itself runs in a simulator on the network-connected Mac, or on a device connected to the Mac.

Limitations

There are some tasks that Xamarion.iOS for Visual Studio currently cannot do:

- *No iOS simulator on Windows* – The iOS Simulator runs on Mac OS X, so it's necessary to switch to the Mac's screen when testing on the simulator.

Configuring The Mac

Installation

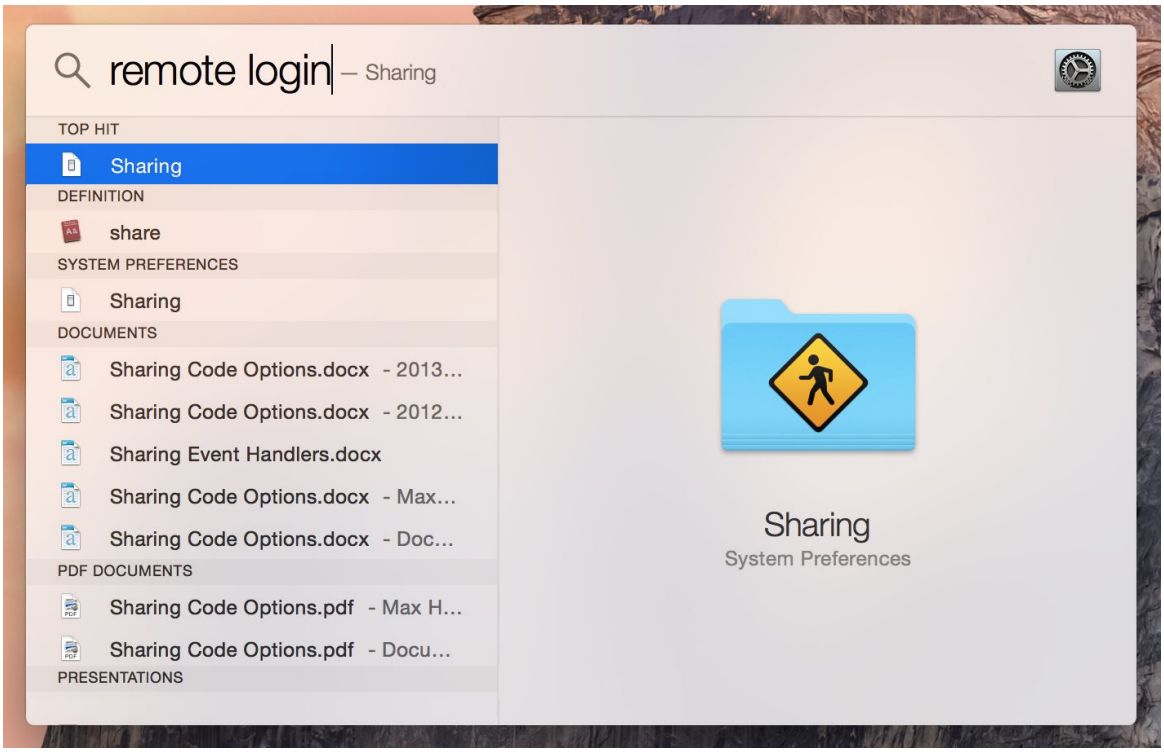
The simplest way to install Xamarin.iOS on your Mac is via the Xamarin Universal Installer, which will also install the necessary prerequisites. Follow these instructions to install Xamarin.iOS tools on your Mac host and to activate your Xamarin license.

Once the software is installed, follow the steps in the next section to configure Xamarin.iOS on OS X to allow Xamarin for Visual Studio to connect to it.

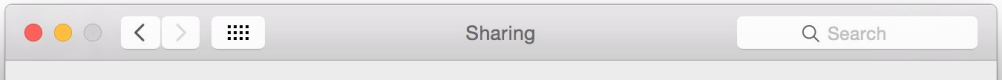
Configuration

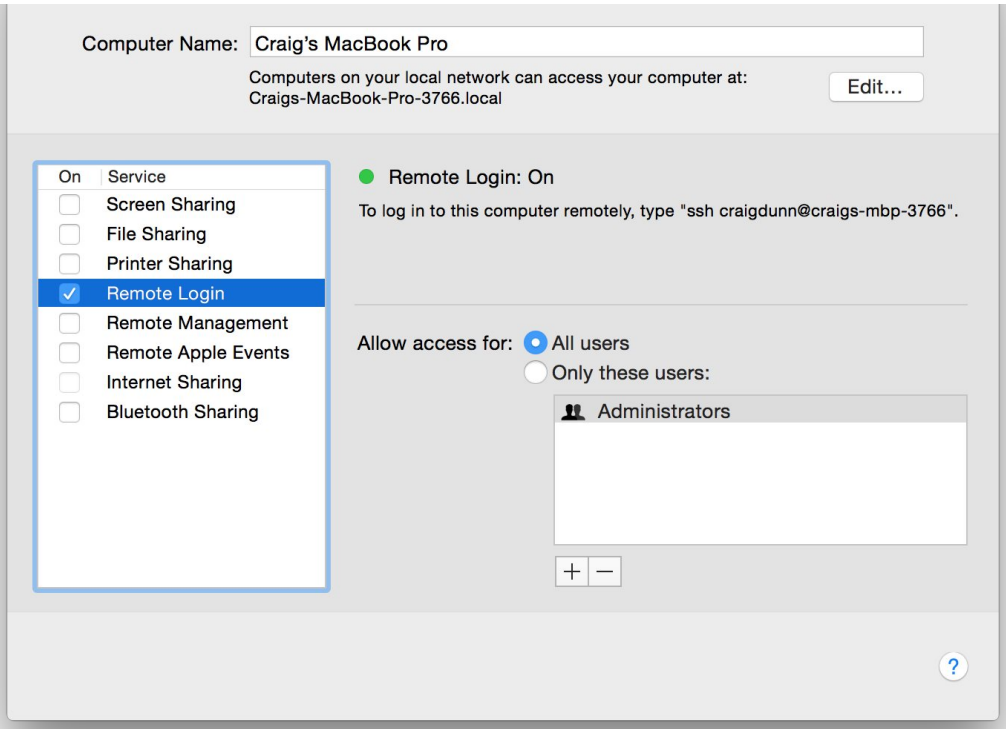
To access communication between the Xamarin extension for Visual Studio and your Mac, you will need to allow **Remote Login** on your Mac. Follow the steps below to set this up:

1. Open *Spotlight* (`Cmd-Space`) and search for **Remote Login** and then select the **Sharing** result. This will open the **System Preferences** at the **Sharing** panel.



2. Tick the **Remote Login** option in the **Service** list on the left in order to allow Xamarin for Visual Studio to connect to the Mac.





3. Make sure that **Remote Login** is set to allow access for **All users**, or that your Mac username or group is included in the list of allowed users in the list on the right.

Your Mac should now be discoverable by Visual Studio if it's on the same network.

In addition to this, if you have the OS X firewall set to block signed applications by default, you may need to allow mono-sgen to receive incoming connections. An alert dialog will appear to prompt you if this is the case.

iOS Developer Setup

For iOS development, it is important that your Mac machine is configured with the relevant signing identities. This allows you to correctly sign your apps so that they will be distributed either via the App Store or Ad Hoc. Follow the link below for instructions on setting up a Mac for iOS development with Xamarin. These are the same steps regardless of whether you are using Xamarin Studio or Visual Studio – if you're already a Xamarin.iOS developer then you've probably already taken these steps:

- Device Provisioning

Once your Mac is configured, it's time to set up your Windows computer.

Windows Installation

Visual Studio 2015 allows you to install Xamarin's tools for VS during setup. Refer to Microsoft's Install Xamarin with Visual Studio guide for more information on how to do this.

To install the tools for other versions of Visual Studio (and Xamarin Studio) the Xamarin Universal Installer, which is found here, provides the simplest way to install Xamarin's software on your Windows machine.

The installer includes a *Xamarin Bonjour Service* (Apple's network discovery protocol), which is installed at the end of the process. Installing the Xamarin Bonjour Service requires Administrator rights, so expect to be prompted twice to allow installation as an Administrator.

Note: on Windows 8 you must run the installer as an Administrator. The simplest way to do this is start a Command Prompt by right-clicking on `cmd.exe` and choosing to Run as Administrator… and execute the installer MSI on the command line:

```
msiexec /i:some_path_to\XamarinSetup.Universal.exe
```

Be sure the file name is correct.

This does not apply to Windows 8.1

Installation Process

The Xamarin Universal Installer will guide you through the necessary steps to install Xamarin Software which, amongst others, includes the following:

- Xamarin for Visual Studio (This is already included in Visual Studio 2015)
- Xamarin Studio

Note that on Windows, Xamarin Studio can only be used for Xamarin.Android development.

The Downloads page can also be used to download the current version, and previous versions, of Xamarin.iOS for Visual Studio.

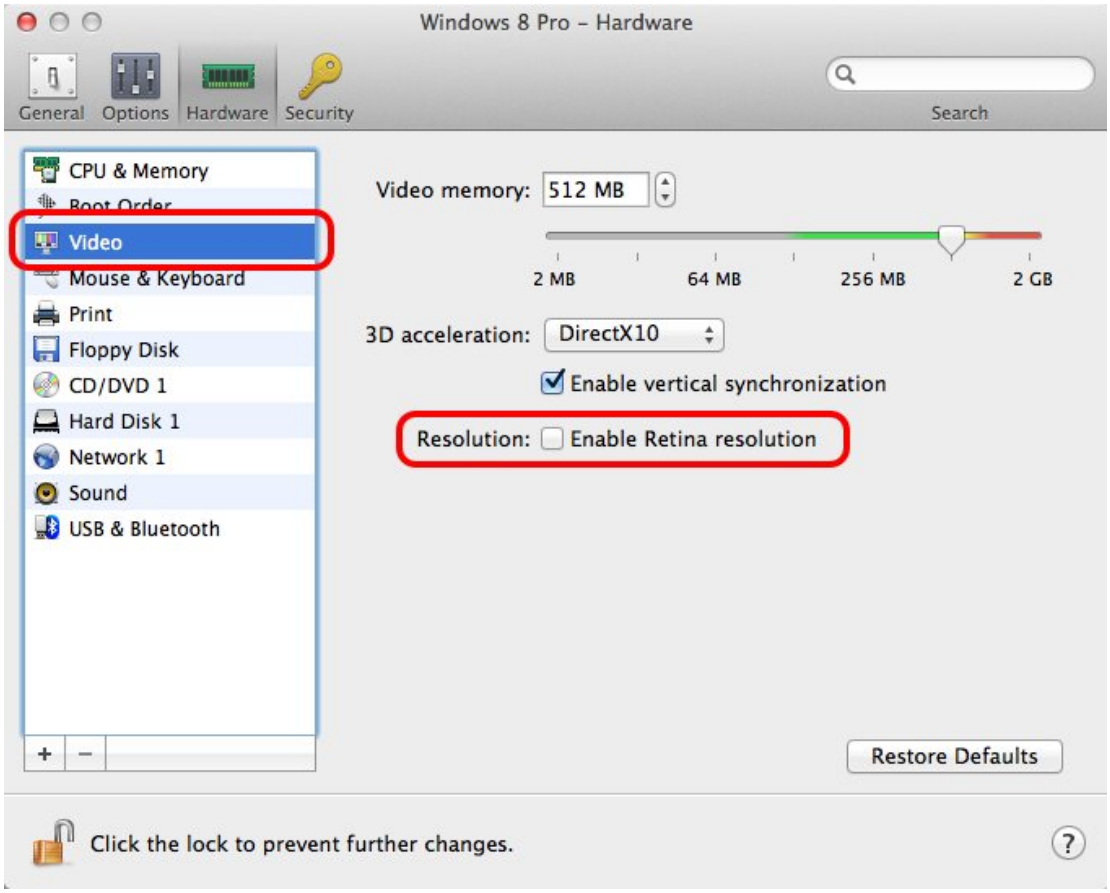
Using a Windows Virtual Machine

Xamarin.iOS for Visual Studio can be run inside a Windows virtual machine on OS X, using applications such as *Parallels* or *VMWare*.

Parallels and Retina Macbook Pros

When using a Windows Virtual Machine inside Parallels 10 on a Macbook Pro with Retina Display, you should check the Video Hardware Settings to ensure Enable Retina resolution is **not** checked (in previous versions of Parallels you would select the **Scaled** option). If this is not set correctly

some elements of the IDE will be poorly rendered and difficult to use.



Installation Complete

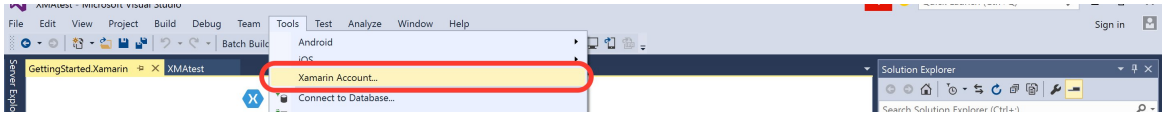
After the installation process is complete, there are still a few more steps required to get everything working:

- *Link Your Xamarin Account* – This ensures that all features you are entitled to are enabled. You can create a new Xamarin Account if required.
- *Connect Visual Studio to the Xamarin Mac Agent* – Visual Studio must be connected to the Mac build host using the Xamarin Mac Agent before it can build Xamarin.iOS projects.
- *Configure the Visual Studio Toolbar* – This will let you easily access Xamarin.iOS features in Visual Studio.

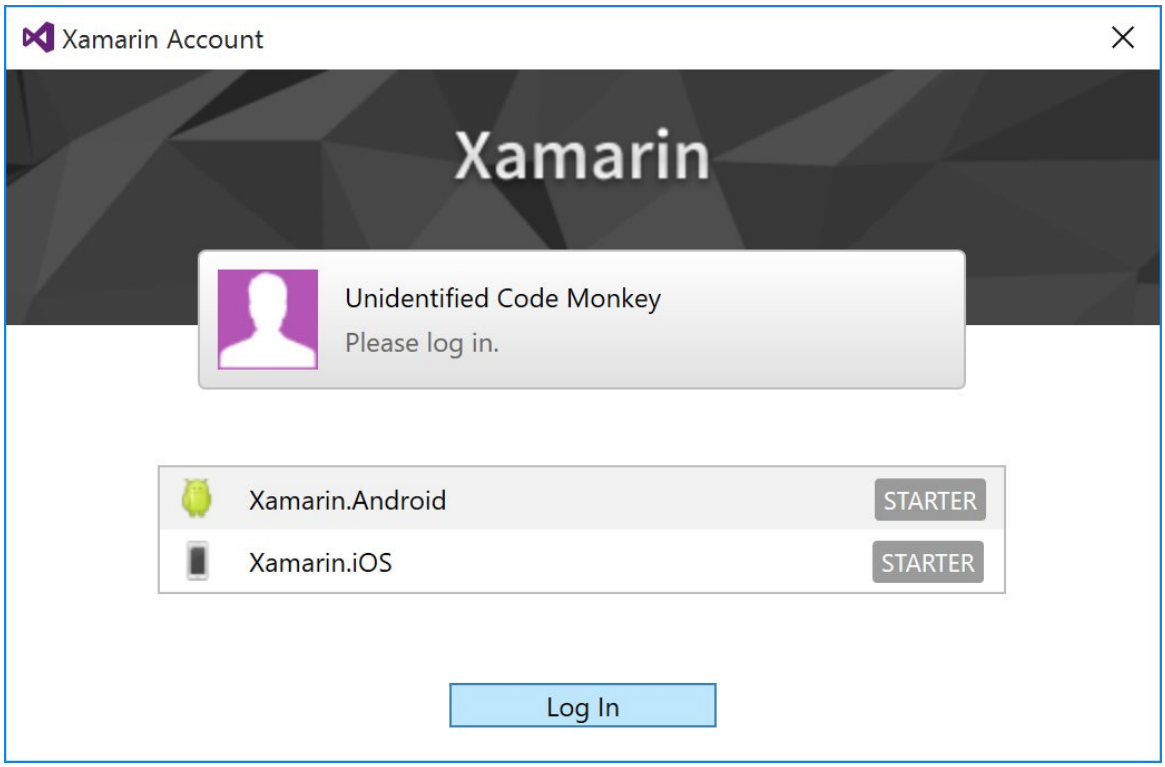
Linking to your Xamarin Account

You must link your Xamarin account to Visual Studio by signing in to access all the features you are entitled to, and to activate your license. Follow the steps below to do this.

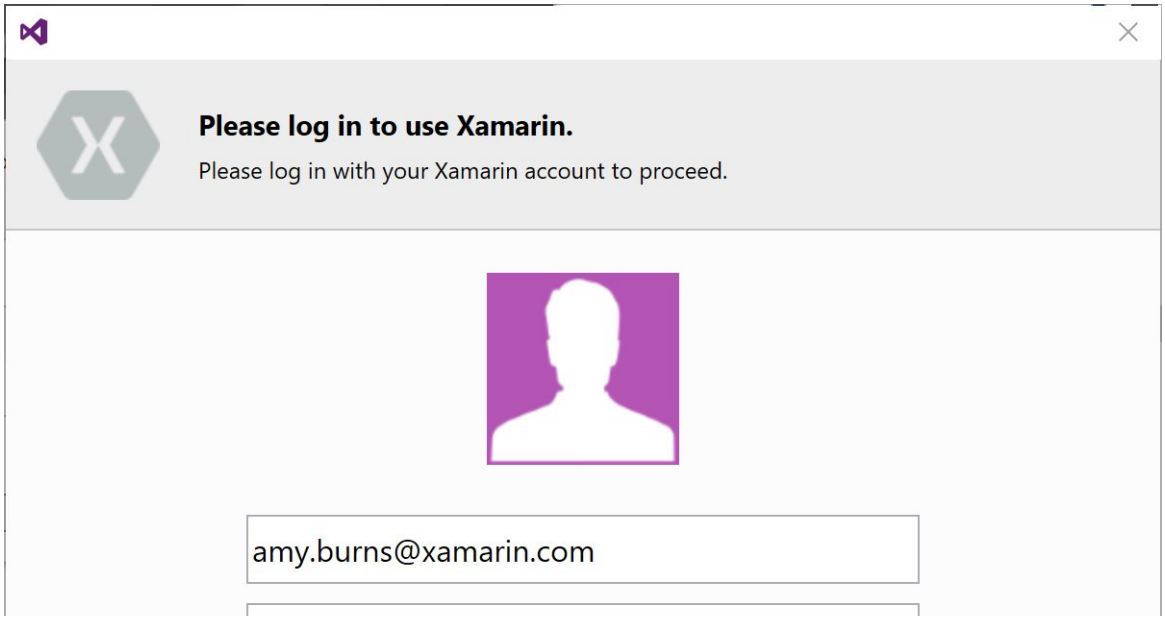
In Visual Studio, go to the Tools > Xamarin Account… menu item:

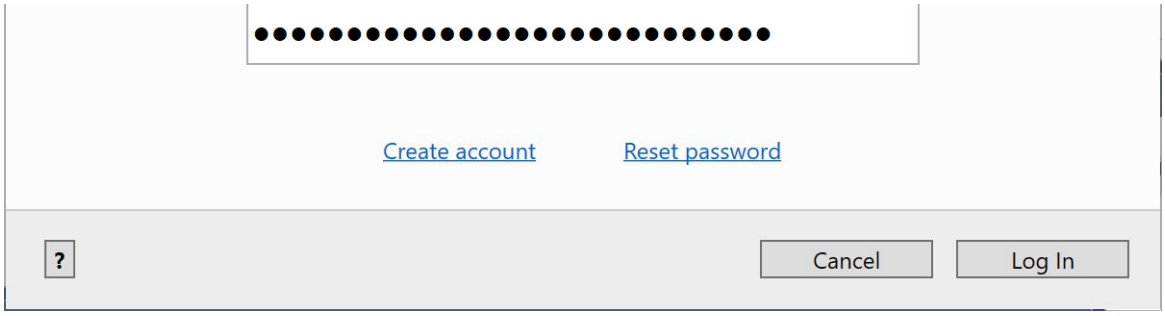


The Xamarin Account dialog will appear, click Log In to enter your Xamarin credentials. If you don't already have a Xamarin account, the next screen will provide a link allowing you to create one:



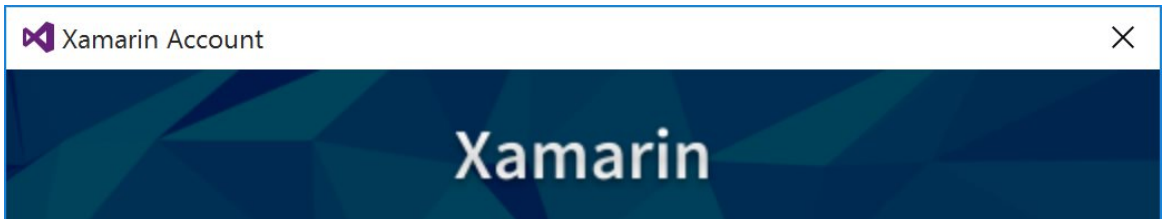
Enter your Xamarin Account details and click Log In, or create a new account:

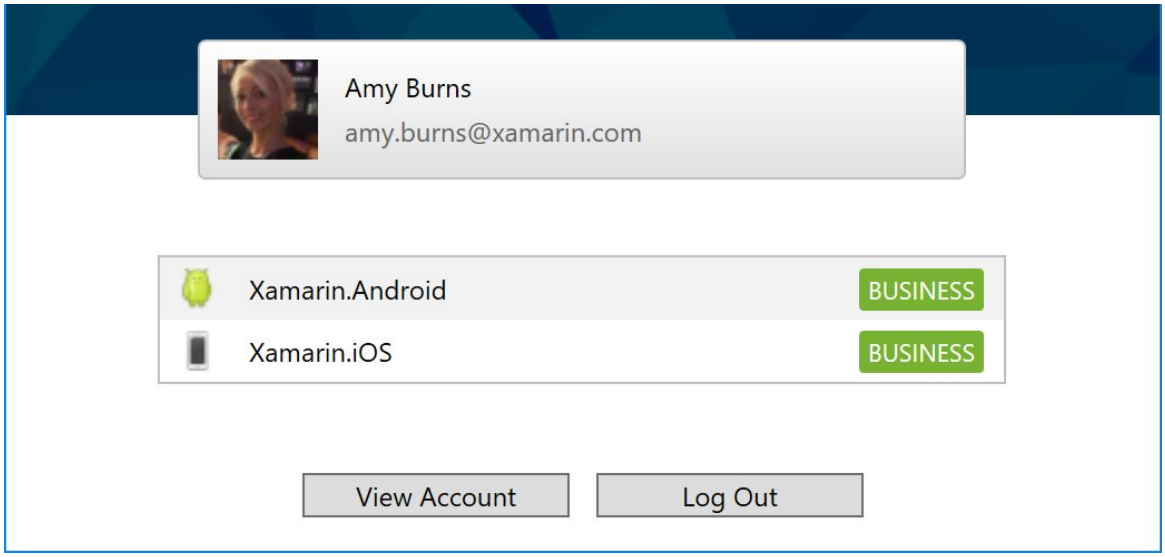




After a few seconds, your Xamarin subscription will be activated with the relevant subscription tier that you have purchased:

You'll now be returned to the Xamarin Account dialog screen indicating that your Xamarin Account has been successfully linked to the Xamarin Plug-in for Visual Studio:



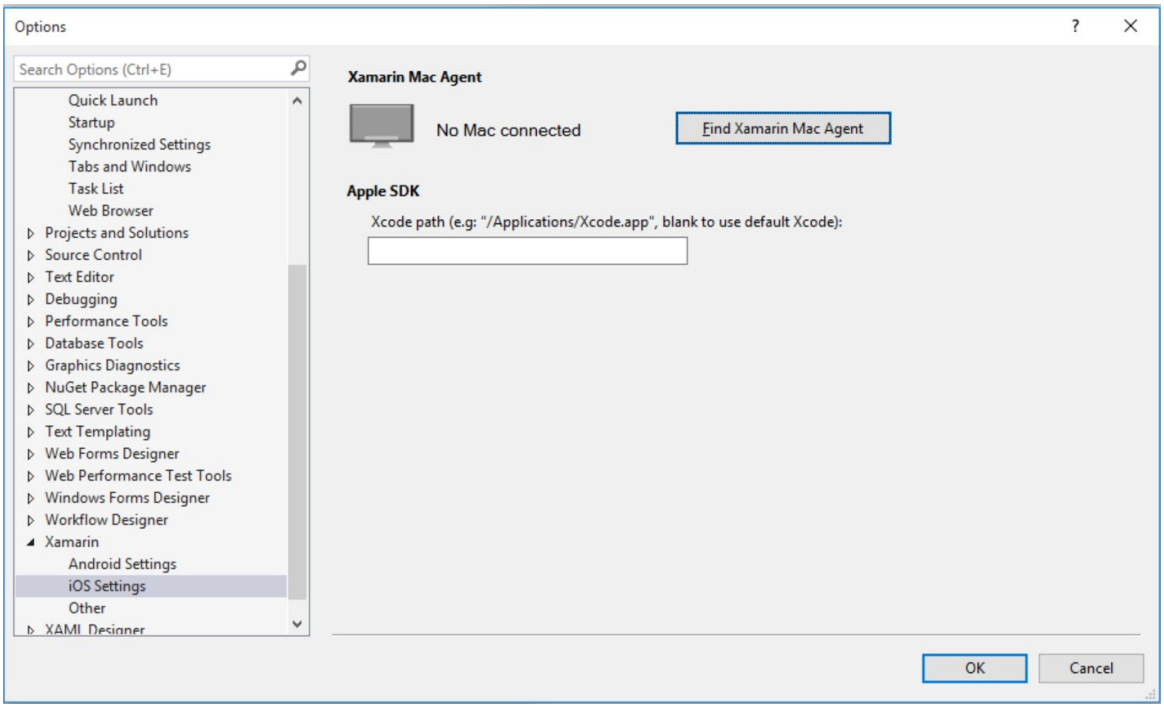


Connecting to the Mac Using XMA

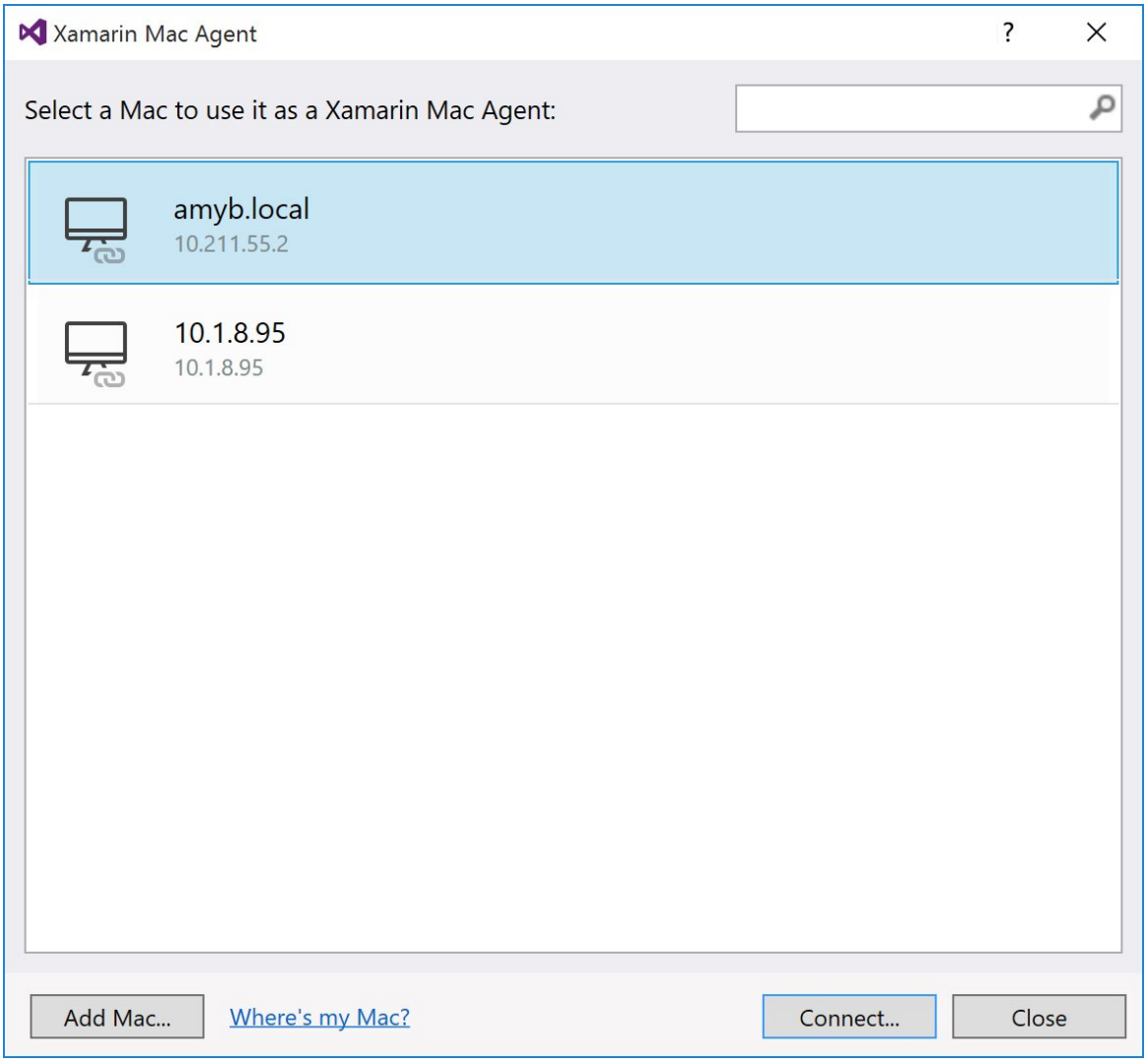
The Xamarin Mac Agent has completely replaced the Mac Build Host application, allowing a connection from Xamarin.iOS for Visual Studio to your Mac machine. For more information on the Xamarin Mac Agent, refer to the XMA guide.

To connect your Mac, follow the steps below:

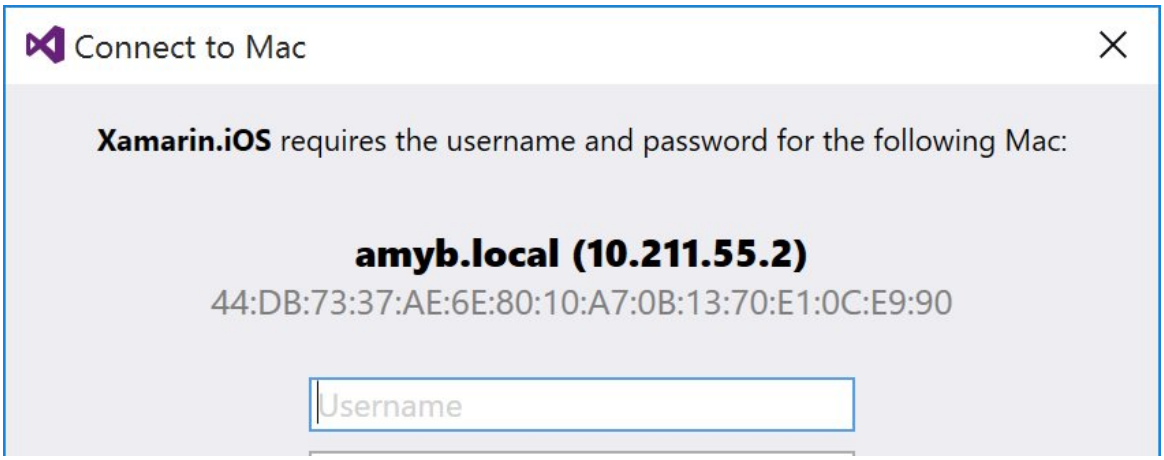
- Browse to *Tools* > *Options* and under *Xamarin* select *iOS Settings*:

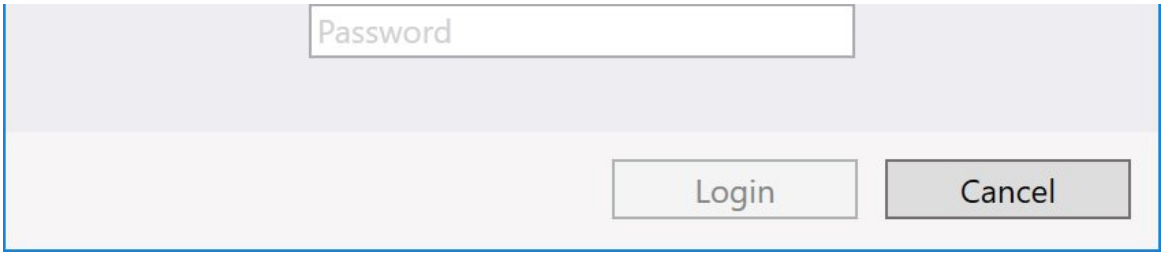


- Providing the Mac has been correctly configured to allow **Remote Login**, you should be able to select your Mac in the list:



- This will prompt for the administrative credentials of your Mac host:



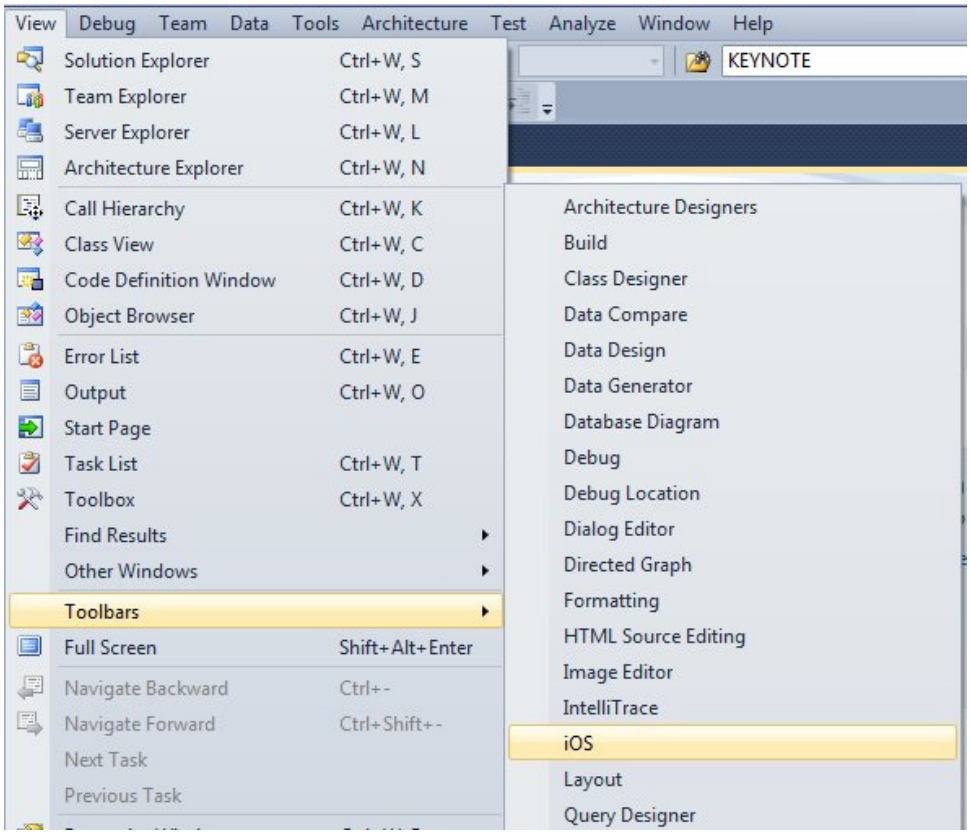


- When XMA is successfully connected, it will display the 'Connection Successful' icon next to the machine name:

XMA will remember your credentials for next time, and will reconnect each time you open Visual Studio.

Visual Studio Toolbar Configuration

You must manually configure the Visual Studio toolbars, as they will be necessary to perform certain operations. First, open the View > Toolbars menu and make sure the iOS entry is selected. Choose the menu item as shown in this screenshot—it should be ticked to indicate that the toolbar is visible:



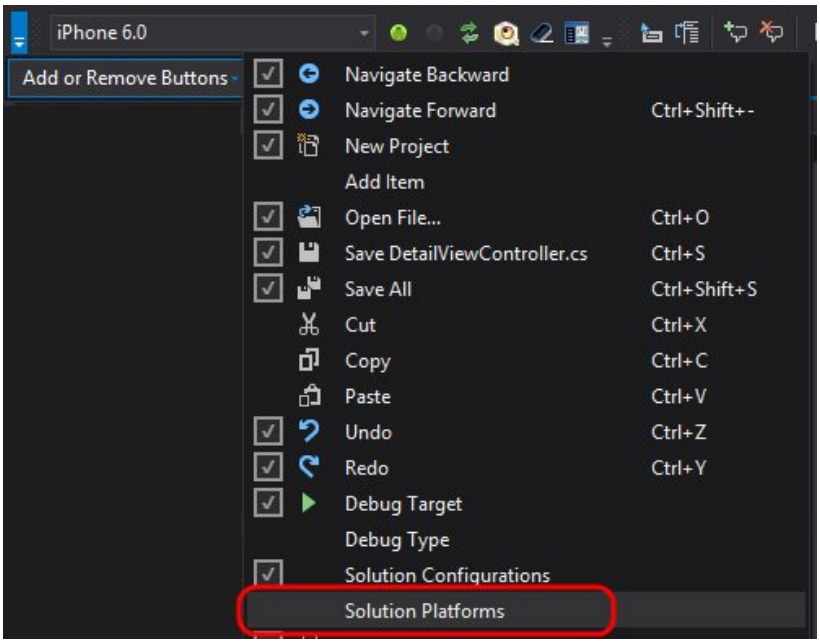
Add the Solution Platforms button to the Standard toolbar, following the instructions below. This allows an iOS

Device or the iOS Simulator to be selected when debugging.

Visual Studio 2012 & 2013

Click the menu button at the right side of the Standard bar:

- Choose Add or Remove Buttons
- Select Solution Platforms



The Standard and iOS toolbars should now resemble this screenshot:

Once the toolbar configuration is complete, you are ready to begin using Xamarin iOS for Visual Studio.

Visual Studio 2015

If the Xamarin tools are installed through Visual Studio as outlined in the Install Xamarin with Visual Studio guide, the toolbars will be configured by default. If they are not, follow the same steps as Visual Studio 2012 and 2013 to configure them.

Summary

This article presented a step-by-step guide to installing, configuring, and using Xamarin iOS for Visual Studio.

It covered installing and configuring the prerequisite tools on Windows and Mac OS X.