

## Windows IoT Core Lab Setup

### Bring Your Own Device Setup Overview

Assuming you have Windows 10 installed then you will need to allocate about an hour for setup. If you also have Visual Studio Update 2 installed, then allocate about 20 minutes.

#### Overview of software requirements for the labs.

1. **Windows 10** (running on the metal ie not virtualized)

As at April 2016 build 10.0.10586. Up to date and fully patched. You can upgrade to Windows 10 from [here](#). If you are already running Windows 10, you can find your current build number by clicking the start button, typing "winver", and hitting enter.

1. You will need Admin rights on the PC
2. Be able to setup Internet connection sharing

Enable Internet Connection Sharing on each development PC so the Raspberry Pi can pass through internet requests – see [Enabling Internet Connection Sharing](#).

3. Developer mode in Windows 10 enabled. [Instructions](#).

2. **Install Visual Studio 2015 Update 2** ([Community Edition](#) is sufficient)

1. Install the Windows IoT Core Project Templates

3. **Provision an Azure Account (free)**

4. **Install**

1. Windows 10 IoT Core Dashboard
2. IoT Hub Device Explorer
3. Windows IoT Remote Client from the Windows 10 App Store

5. **Pre-Cache Windows IoT Core NuGet Packages**

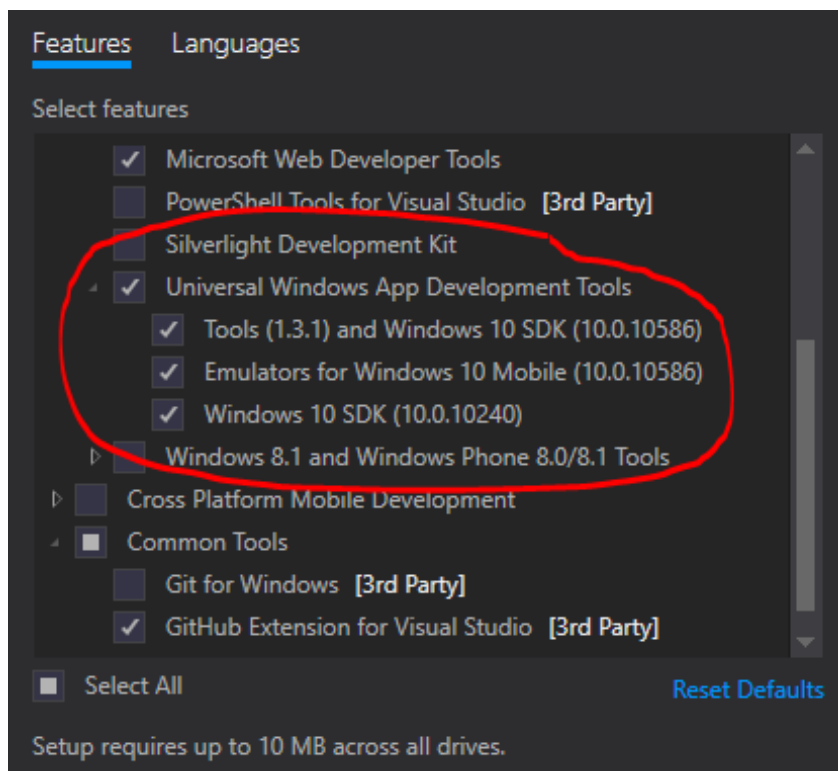
#### Windows 10 Development PC Hardware requirements

- 1) A PC with minimum 4gb RAM, 1GB free disk space
- 2) Two network interfaces
  - a. One wireless
  - b. One wired Ethernet port (or a USB Ethernet dongle) to connect to the Raspberry Pi
- 3) Cross over Ethernet cable to connect the Raspberry Pi to the Windows 10 development machine

## Install Visual Studio 2015 Update 2

Visual Studio 2015 Update 2 or above ([Community Edition](#) is sufficient).

**NOTE:** Ensure you do a **Custom** install and select **Universal Windows App Development Tools** -> **Tools and Windows SDK**. Also be sure to install the Web Development Tools.



## Validate your Visual Studio Installation

You can validate your Visual Studio installation by selecting Help > About Microsoft Visual Studio. The required version of Visual Studio is 14.0.25123.00 Update 2. The required version of Visual Studio Tools for Universal Windows Apps (aka Tools (1.3.1) and Windows 10 SDK (10.0.10586)) is 14.0.25208.00.

## Install Windows IoT Core Project Templates

Download the template from [here](#). Alternatively, the templates can be found by searching for Windows IoT Core Project Templates in the [Visual Studio Gallery](#) or directly from Visual Studio in the Extension and Updates dialog (Tools > Extensions and Updates > Online).

## Provision an Azure Account

If you don't already have an Azure account, then you will need to provision one.

There are currently two free trial offers – either good for the purposes of the workshop.

1) [Visual Studio Dev Essentials](#) Sign up for free. \$25 a month for a year. More slow and steady over an extended period of time.

2) [Free one-month trial](#). Sign up for free and get \$200 to spend on all Azure services. Great if you really want to exercise lots of Azure capabilities for a limited period of time.

Valid credit card information is required for identity verification purposes only. Your credit card will not be charged for this offer unless you explicitly remove the spending limit.

If you sign up for Dev Essentials, then be sure to select the Azure offering highlighted below.

The screenshot shows the 'My Benefits' page for Visual Studio Dev Essentials. The page has a dark header with navigation links: Downloads, News, Support, Marketplace, and Documentation. The main heading is 'Welcome to Visual Studio Dev Essentials!' with the tagline 'Everything you need to build and deploy your app on any platform'. Below the header, there are eight benefit cards arranged in a 2x4 grid. The first card is 'Visual Studio Community' (Full-featured, extensible IDE). The second is 'Visual Studio Code' (Modern lightweight editor). The third is 'Visual Studio Team Services' (Basic level). The fourth is 'Pluralsight' (6-month subscription). The fifth card, 'Azure' (\$25 monthly Azure credit), is circled in red. It offers '12 months of access to your own personal sandbox for dev/test. Provision virtual machines, cloud services, and...'. The sixth card is 'Microsoft R Server Developer Edition'. The seventh is 'Xamarin University Training' (Free on-demand access). The eighth is 'Microsoft SQL Server Developer Edition'. Each card includes an icon, a title, a description, and links to download or activate the benefit.

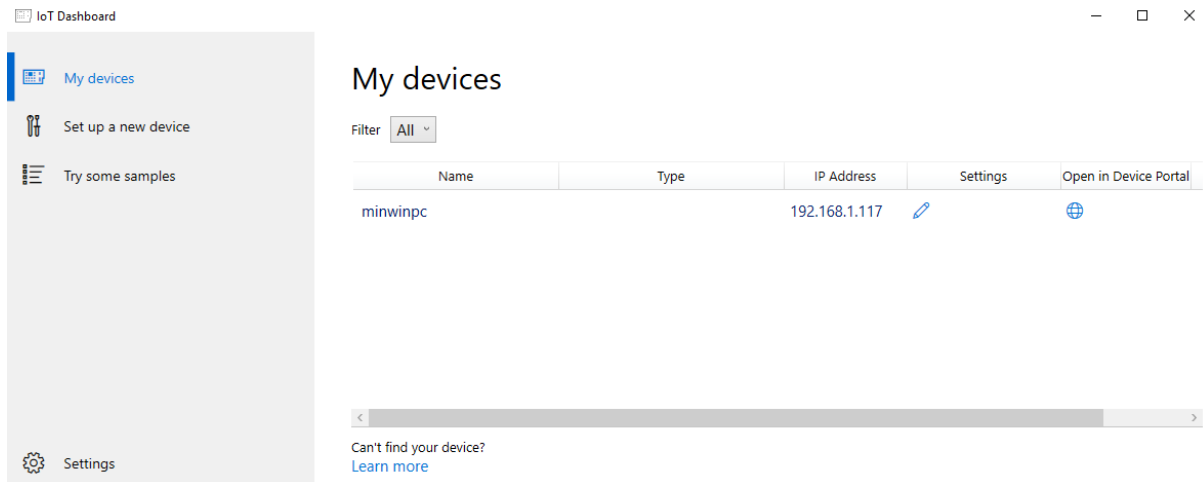
Benefit	Description	Link
Visual Studio Community	Full-featured, extensible IDE	Download   Learn more
Visual Studio Code	Modern lightweight editor	Download the beta   Learn more
Visual Studio Team Services	Basic level	Get started   Learn more
Pluralsight	6-month subscription	Get code   Take training
<b>Azure</b>	<b>\$25 monthly Azure credit</b> 12 months of access to your own personal sandbox for dev/test. Provision virtual machines, cloud services, and...	Activate   Go to your account
Microsoft R Server	Developer Edition	Download
Xamarin University Training	Free on-demand access	Get code   Get code
Microsoft SQL Server	Developer Edition	Download   Learn more

# Install Windows 10 IoT Core Dashboard

See [Get Started with Windows IoT](#).

From here click “Download Windows 10 IoT Core Dashboard”

To start press the Windows key and type “iot dashboard” run the app and for convenience pin it to your Start screen or taskbar.

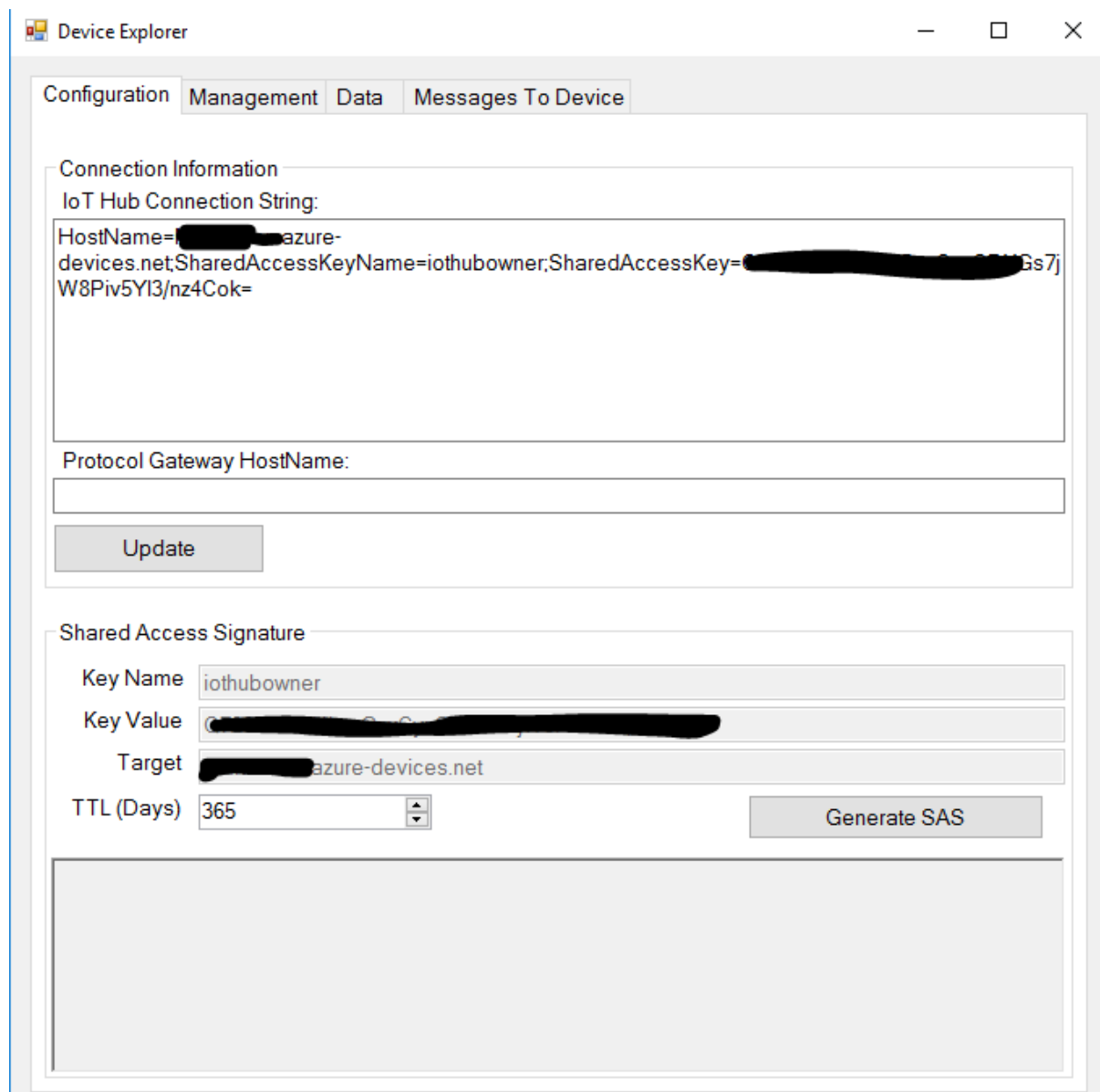


## Install IoT Hub Device Explorer

Azure IoT Hub only allows connections from known devices that present proper credentials. In this lab series you will use the *DeviceExplorer* utility to provision a device for use in Azure IoT Hub.

A pre-built version of the Device Explorer application for Windows can be downloaded by clicking on this link: [Downloads](#) (Scroll down for **SetupDeviceExplorer.msi**).

To start press the Windows key and type “device explorer” run the app and for convenience pin it to your Start screen or taskbar.



The screenshot shows the 'Device Explorer' application window with the 'Configuration' tab selected. The window has a title bar with standard Windows controls (minimize, maximize, close). Below the title bar are four tabs: 'Configuration', 'Management', 'Data', and 'Messages To Device'. The 'Configuration' tab is active and contains two main sections: 'Connection Information' and 'Shared Access Signature'.

**Connection Information**

IoT Hub Connection String:

HostName=[redacted]azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=[redacted]Gs7jW8Piv5Yl3/nz4Cok=

Protocol Gateway HostName:

[Empty text box]

[Update button]

**Shared Access Signature**

Key Name: iothubowner

Key Value: [redacted]

Target: [redacted]azure-devices.net

TTL (Days): 365 [dropdown arrow]

[Generate SAS button]

[Large empty text box for SAS output]

# Install Windows IoT Remote Client

From the Windows App Store install the Windows IoT Remote Client.

The screenshot shows the Windows App Store interface. At the top, there's a navigation bar with 'Store', 'Home', 'Apps', 'Games', 'Music', and 'Movies & TV'. A search bar on the right contains 'Windows IoT Remote Client'. The main content area features the app's tile for 'Windows IoT Remote Client' by Microsoft Corporation, which is marked as 'Built for Windows 10'. Below the tile, a description states: 'The Windows IoT Remote Client application is a part of a remote display and sensor technology available for the Insider build of Windows 10 IoT Core. With a device...'. A 'More' link is present. Below this, it says 'This product is installed.' and an 'Open' button. To the right of the app tile is a preview window showing the 'Connect' screen of the application, which includes fields for 'Select a discovered device', 'Enter an IP address', and a 'Connect' button. Below the main content, the 'What's new in this version' section for Version 1.0.8 lists updates: 'Introduces Gyro and Magnetometer redirection', 'Improves handling of display resolution', and 'Adds more error handling.' A 'Troubleshooting' section follows, explaining that the Raspberry Pi 2 does not have GPU support and that the Raspberry Pi 3 can have improved performance by lowering resolution. A note at the bottom advises ensuring the IoT device is on the same network and running the Insider build 14295 or later. On the right, the 'People also like' section recommends 'Token2Shell/MD' (5 stars, \$36.79 to \$3.69) and 'IoT Explorer for AllJoyn' (5 stars, Owned).

Store

Home Apps Games Music Movies & TV

Windows IoT Remote Client

Windows IoT Remote Client

Microsoft Corporation

★★★★★

Share

Built for Windows 10

The Windows IoT Remote Client application is a part of a remote display and sensor technology available for the Insider build of Windows 10 IoT Core. With a device...

More

This product is installed.

Open

PC

What's new in this version

Version 1.0.8

Introduces Gyro and Magnetometer redirection

Improves handling of display resolution

Adds more error handling.

Troubleshooting

The Raspberry Pi 2 does not have GPU support on Windows 10 IoT Core, thus the framerate of the remote display experience is lower than on other boards. Enhanced CPU on the Raspberry Pi 3, as well as graphics capabilities on the MBM and Dragonboard, allow for a more stable experience. Lowering the resolution to 800x600 will increase performance. You can do this from the management page on your device, the home screen. If the Display Resolution box is blank, see the section below about the Raspberry Pi not rendering.

Make sure your IoT device is on the same network as your companion device. Check that you're running the Insider build 14295 or later of Windows 10 IoT Core with the

People also like

Show all

Token2Shell/MD

★★★★★

\$36.79 \$3.69

IoT Explorer for AllJoyn

★★★★★

Owned

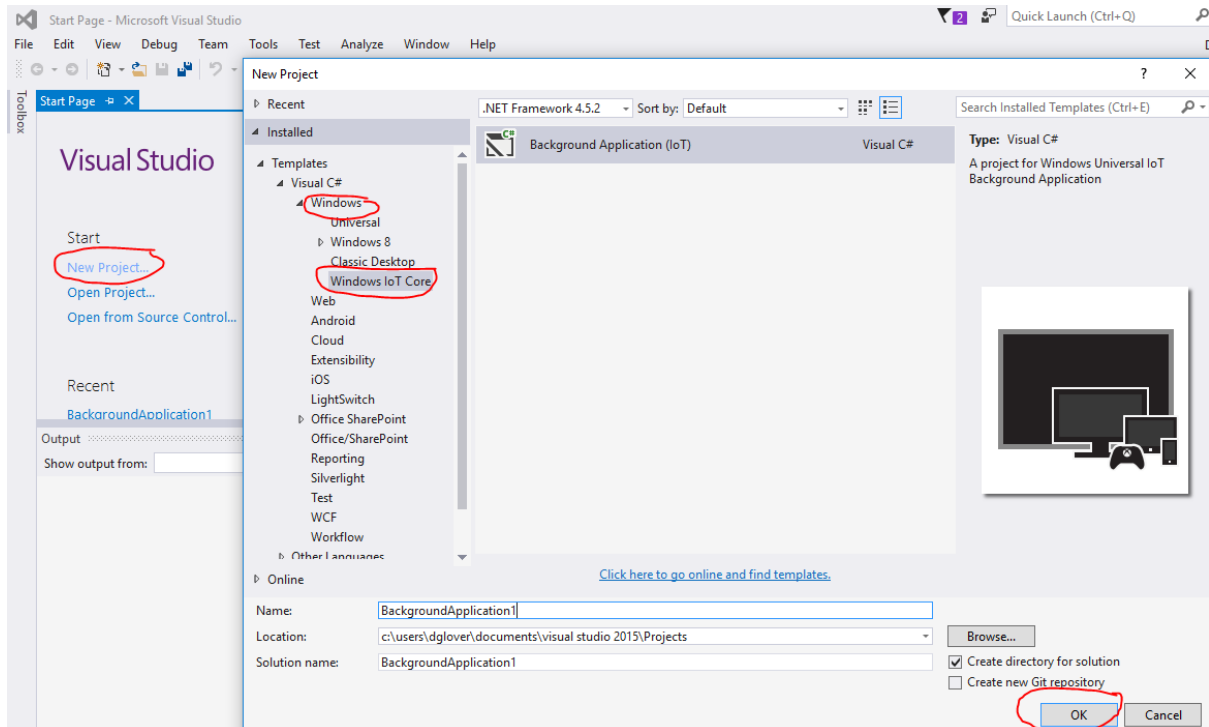
Dev Center

## Pre-Cache Windows IoT Core Nuget Packages

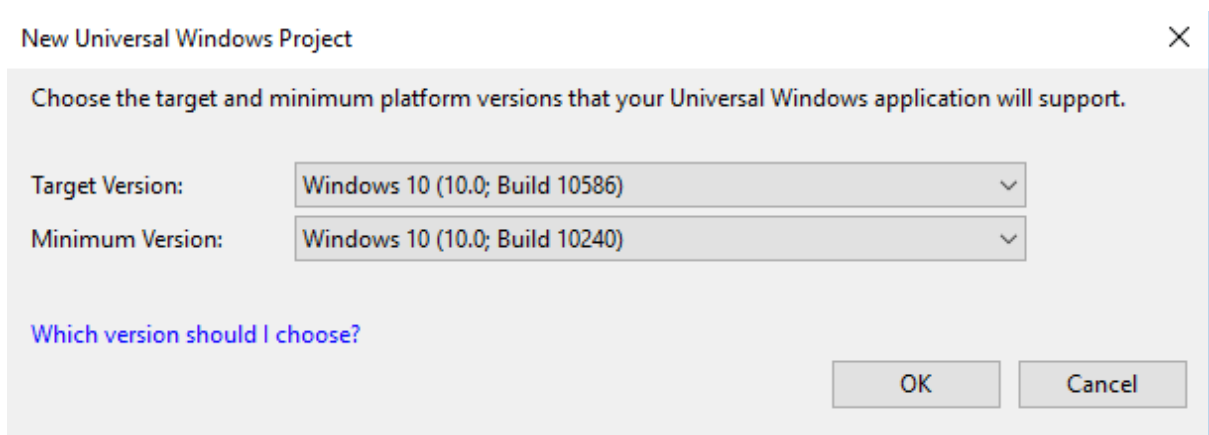
Create and Build a test Windows IoT Core project. The purpose is to pre-cache the required NuGet packages on to your PC.

### Start Visual Studio

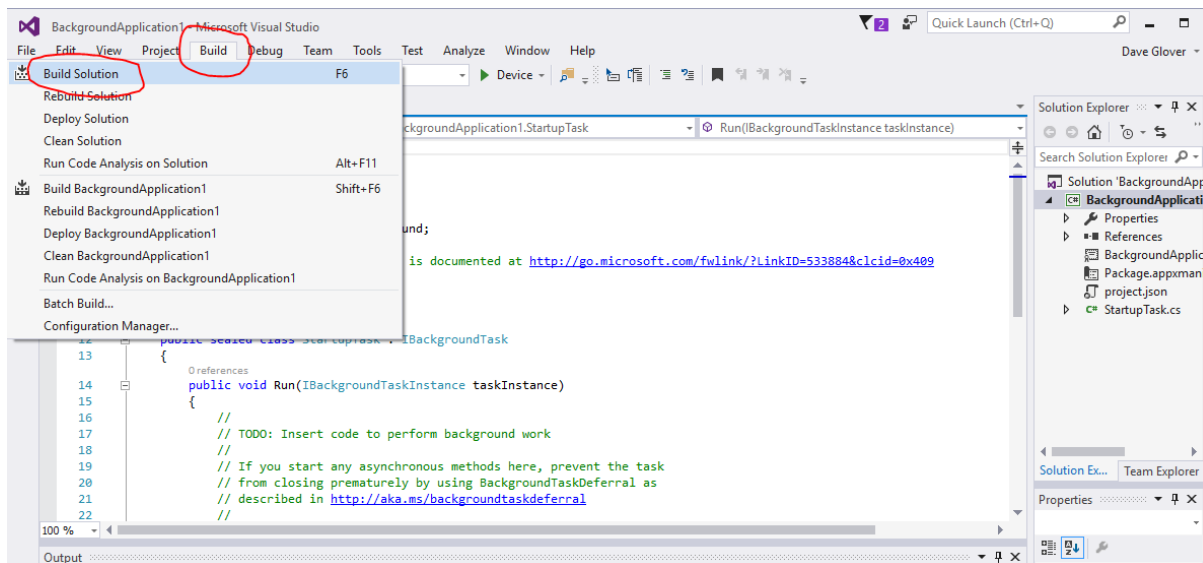
New Project -> Windows -> Windows IoT Core -> OK



### Select the Defaults



## Build the Project



This will download and cache the required NuGet packages.

## Ensure successful build

