# Setting up your development environment.

## Prerequisites

The following is required to complete this exercise.

* Windows 10 (Version 10.0.14393 or later) running on your development machine. Type “ver” at the command prompt if you are unsure.
* Visual Studio 2015 (with at least update 3). Any edition.

## Task 1 – Download & Install Device Explorer

To register your device in the Azure IoT Hub Service and to monitor the communication between them you need to install the Azure Device Explorer. This is a sample application written using the IoT SDK and utilizing the IoT Registry APIs to allow the registering of new devices.

1. Follow this link to download the **SetupDeviceExplorer.msi** file: <https://github.com/Azure/azure-iot-sdks/releases>. (A copy is also available in the lab assets folder). If you wish to you can also download and review the source code for this utility which is available on the same page. It makes a good learning resource on its own.
2. Install the downloaded application (it’s a “Setup->Next->Next->Finish” job!). If the application is ready installed on your machine, you can skip this step.
3. From the *C:\Program Files (x86)\Microsoft\DeviceExplorer* folder launch **DeviceExplorer.exe** so ensure its works.

## Task 2 – Download & Install IoT Dashboard

The Windows 10 IoT Core Dashboard allows you to easily discover and access your tiny devices running Windows IoT Core 10.0.10586 version or later. Once these have been deployed on the same network as your development PC (connected via either Ethernet or Wi-Fi) you’ll see them appear in the dashboard. This is the same tool which is used to flash an SD Card with Windows 10 IoT Core before it is inserted into a device.

1. Download the dashboard from here: <http://go.microsoft.com/fwlink/?LinkID=708576> (should point to <https://iottools.blob.core.windows.net/iotdashboard/setup.exe>).
2. Run the setup (it’s another “Setup->Next->Next->Finish” job!).
3. Once complete, from the Start Menu type “dashboard”. Click on **“Windows 10 IoT Core Dashboard”** to launch the tool.
4. To view your device on the IoT Dashboard you’ll need to ensure it has network connectivity. Whilst the device can be connected via Ethernet, Wi-Fi or Bluetooth the latter two often require manual setup steps. **The surest way to connect to your device is either via an Ethernet cable plugged into your switch/router or via “Connection Sharing” on your development PC**. So long as you have a spare Ethernet port you can connect your device directly via a normal Ethernet cable. See *Setup your device to Internet Share.pdf* in the Assets folder for steps on how to setup Connection Sharing.
5. Providing your device is setup correctly it will appear in the “My Devices” tab of the Dashboard. *Note: It can take up to 5 minutes for a new device to appear in the list.*

|  |  |
| --- | --- |
|  |  |

1. You can test connectivity to the device in 1 of 2 ways:
   * Select “Launch PowerShell” from the context menu of your selected device. After entering the device Administrator password, you will be able to control the device via Remote PowerShell.

|  |  |
| --- | --- |
|  |  |

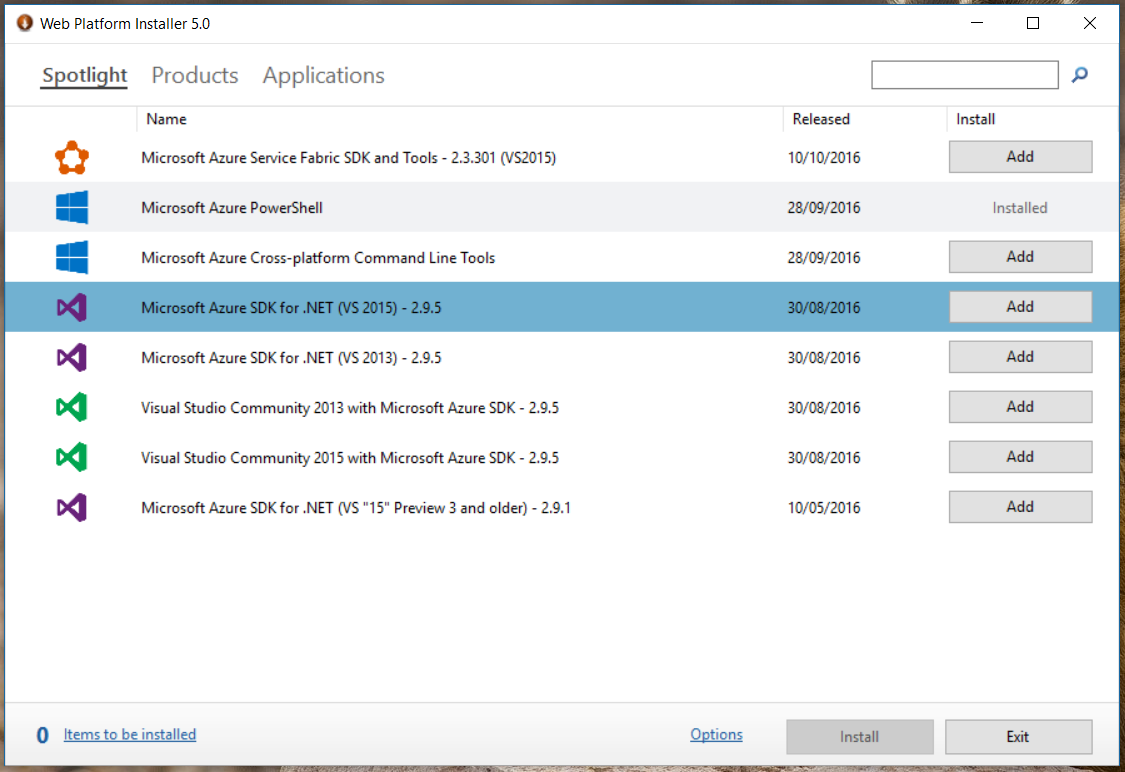
* + Choose “Open in Device Portal” from the context menu to open the web based administration interface.

|  |  |
| --- | --- |
|  |  |

## Task 3 – Install Azure PowerShell & Azure SDK.

Your Windows 10 development machine comes with PowerShell 5.0 as standard however the Azure PowerShell commands need to be installed separately. Visual Studio 2015 comes with a version of the Azure SDK. These are both however updated frequently (sometimes twice a month) so it pays to always have the latest versions installed as updates often include bug fixes, new features and commands or updates necessary to support backend Azure changes. The easiest way to install these is via the Web Platform Installer Tool, a package manager for Microsoft development tools and various open source technologies.

1. If Visual Studio 2015 is open, close it now.
2. Browse to <https://www.microsoft.com/web/downloads/platform.aspx> and click the large green button on the right-hand side of the page marked “Free Download”.
3. Save the installer to your downloads folder then double click to launch.
4. Once loaded the “Spotlight” home page shows the status of installed components. You will need to click **Add** on “**Microsoft Azure PowerShell**” and on “**Microsoft Azure SDK for .Net (VS2015**)”. *Notice the machine shown below already has the latest “Microsoft Azure PowerShell” installed but not the latest “Microsoft Azure SDK for Visual Studio 2015”.*



1. Once you have added the two components, click “Install”. This may take several minutes to complete.

**It is a good practice to keep these components updated. You can launch the “Web Platform” installer tool from the start menu at any time. If new updates are available, they will be displayed with the “Add” button showing ready for you to click. This is true even if you have installed a previous version of the components.**

**In the future, you might find it easier to install these components from the command Line.**

***List products already installed and those available for installation.***

**C:\Program Files\Microsoft\Web Platform Installer\WebPICmd.exe /list /listoption:all**

**The output contains a short name and description for each product available. Use the short names to specify what you want installed.**

***Install the latest Azure SDK and PowerShell Azure tools.***

**C:\Program Files\Microsoft\Web Platform Installer\WebPICmd.exe /install /products:"VWDOrVs2015AzurePack,WindowsAzurePowershellGet"**

**Read more at:** <https://www.iis.net/learn/install/web-platform-installer/web-platform-installer-v4-command-line-webpicmdexe-rtw-release>

## Task 4 – Enable Developer Mode.

Windows 10 allows new style Universal Applications to be installed via the app store, however developers can install their own applications directly onto their own machine for testing – this is known as Side Loading. Since applications for Window 10 IoT Core are also Universal Applications, it makes sense to test these first on the local machine and enabling “Developer Mode” makes this possible.

*If you have previously opened Visual Studio 2015 and tried to create a Universal Application, you will have already been prompted to enable developer mode. In that case, you can skip this task.*

Setting Developer mode can be done via PowerShell or *Settings->Updates & Security->For Developers*. For this exercise, you will make the changes via PowerShell.

1. Open a PowerShell prompt as Administrator.
2. Type the following command:

Set-ItemProperty "hklm:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppModelUnlock" -Name "AllowDevelopmentWithoutDevLicense" -Value "1"

*Note this is one long command shown on three lines for formatting purposes, don’t split it up over three lines when you type it.*

See <https://msdn.microsoft.com/en-us/windows/uwp/get-started/enable-your-device-for-development> for more details on setting up Developer Mode.

## Task 5 – IoT Core Templates.

Windows 10 IoT Core can run two different types of apps:

* + A Foreground App which as a user interface. Windows 10 IoT Core can only run 1 Foreground App at a time because it has no shell (aka Desktop).
  + A Background App template for when you do not need a user interface. Multiple Background Apps can be run at the same time.

Since Background Apps are the most common, a Visual Studio template is available to help create them. It must be downloaded and installed separately.

1. Close Visual Studio 2015 if it’s already open.
2. Download the template from the Visual Studio Gallery - <https://visualstudiogallery.msdn.microsoft.com/55b357e1-a533-43ad-82a5-a88ac4b01dec>
3. Run the setup (it’s another “Setup->Next->Next->Finish” job!).
4. Once complete, open Visual Studio then go **File->Project->Visual C#->Windows->Windows IoT Core** and check you have a **Background Application (IoT)** template available.

## Task 6 – Install Connected Service for Azure IoT Hub Visual Studio Extension

Installing this extension will provide the ability in Visual Studio to right click on a project and add a reference to an IoT Hub. It will create a number of .cs files which contain boiler-plate code which will vastly improve the speed with which you can connect a device to IoT Hub and send & receive messages.

1. Ensure that Visual Studio is currently shutdown. You cannot install extensions when it is running.
2. Open <https://marketplace.visualstudio.com/items?itemName=MicrosoftIoT.ConnectedServiceforAzureIoTHub> and click the green “Download” button at the top of the page.
3. Double click on the downloaded file to begin the installation.