



# Windows IoT Core Application Management

Created by Rick Lesniak

```
Administrator: Windows PowerShell
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup startup
Headed   : IoTCoreDefaultApp_kwmcxszfer2y!App
Headless : BlinkHeadless-imp_1.0.0.0_arm_n1q2psqd6svm2
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm_6efsg6e0hggwa
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

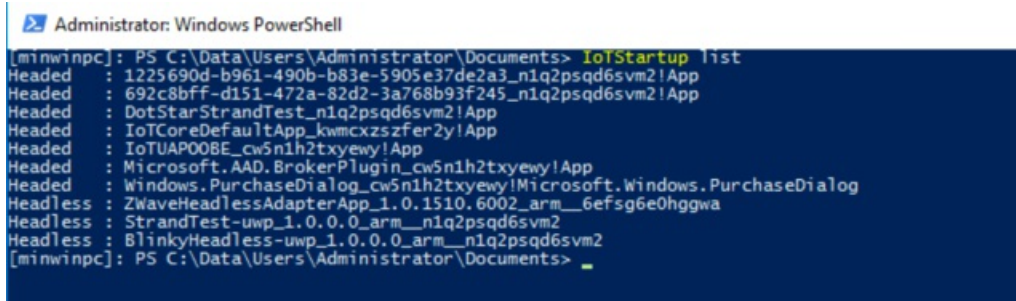
Last updated on 2018-08-22 03:55:11 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
PowerShell: Getting Started	4
PowerShell: Overview	8
Headless Apps at Startup	10
Headed Apps at Startup	11
Wait – what happened to the original Windows IoT Core GUI?	12
Help! BlinkyHeadless or BlinkyHeaded don't appear in the list of apps!	13

## Overview

In this tutorial, we introduce a new Windows tool, called *PowerShell*. We'll be using PowerShell to configure our Blinky apps to automatically run at system startup.



```
Administrator: Windows PowerShell
[adminwinc] : PS C:\Data\Users\Administrator\Documents> IoTStartup list
Headed      : 1225690d-b961-490b-b83e-5905e37de2a3_n1q2psqd6svm2!App
Headed      : 692c8bff-d151-472a-82d2-3a768b93f245_n1q2psqd6svm2!App
Headed      : DotStarStrandTest_n1q2psqd6svm2!App
Headed      : IoTCoreDefaultApp_kwmcxszf2y!App
Headed      : IoTUAP008E_cw5n1h2txyewy!App
Headed      : Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy!App
Headed      : Windows.PurchaseDialog_cw5n1h2txyewy!Microsoft.Windows.PurchaseDialog
Headless    : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm__6efsg6e0hggwa
Headless    : StrandTest-uwp_1.0.0.0_arm__n1q2psqd6svm2
Headless    : BlinkyHeadless-uwp_1.0.0.0_arm__n1q2psqd6svm2
[adminwinc] : PS C:\Data\Users\Administrator\Documents> _
```

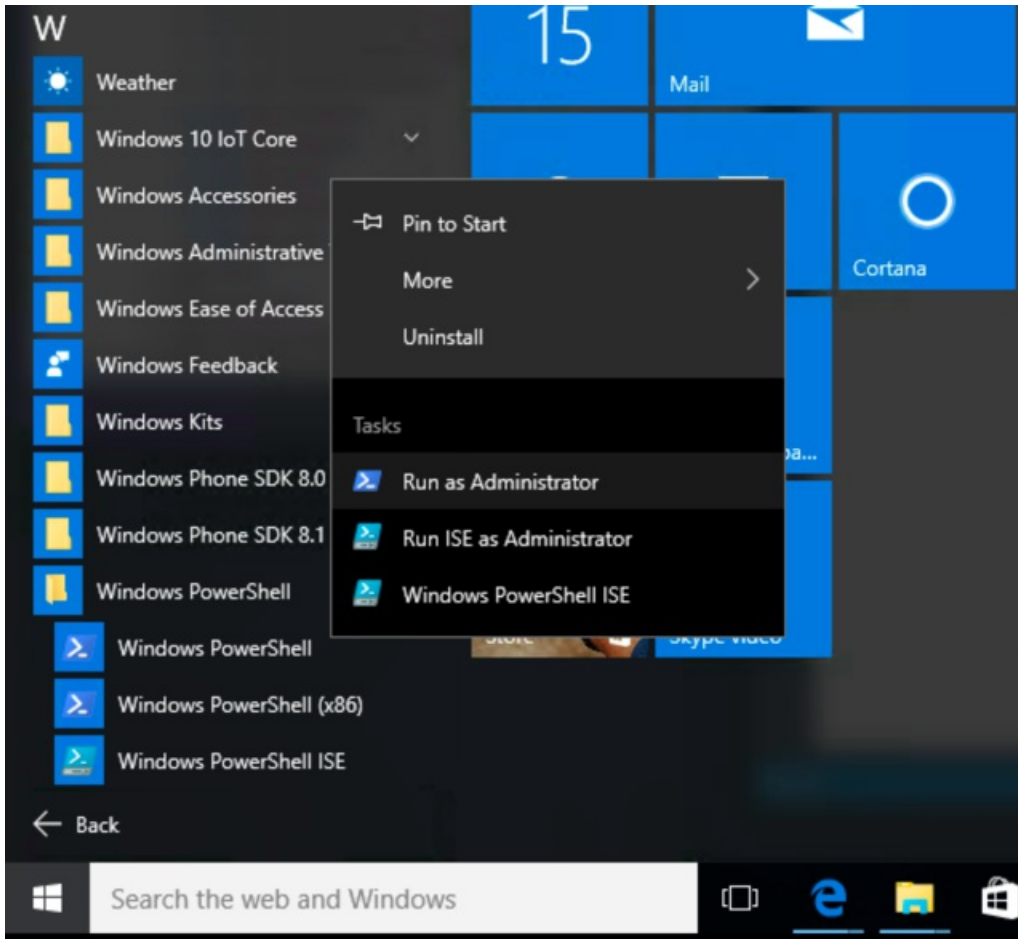
## PowerShell: Getting Started

Windows PowerShell is a command-line utility that runs on your Windows 10 computer, and allows you to execute commands on a Windows IoT device, like a Raspberry Pi.

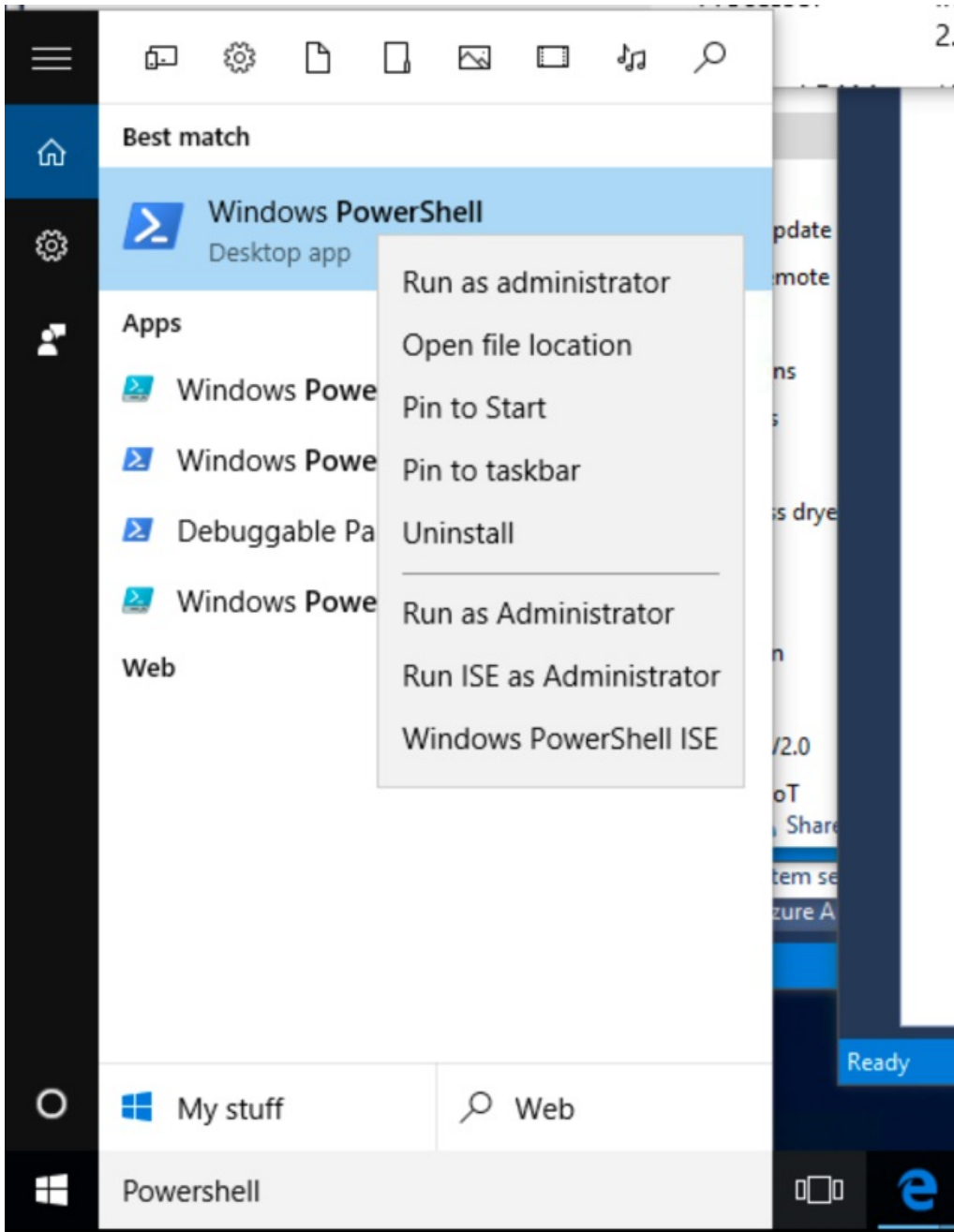
PowerShell has lots of uses, but we'll focus on setting applications to run at startup in Windows IoT.

PowerShell should appear in the **All Apps** submenu of your Start menu, as **Windows PowerShell**

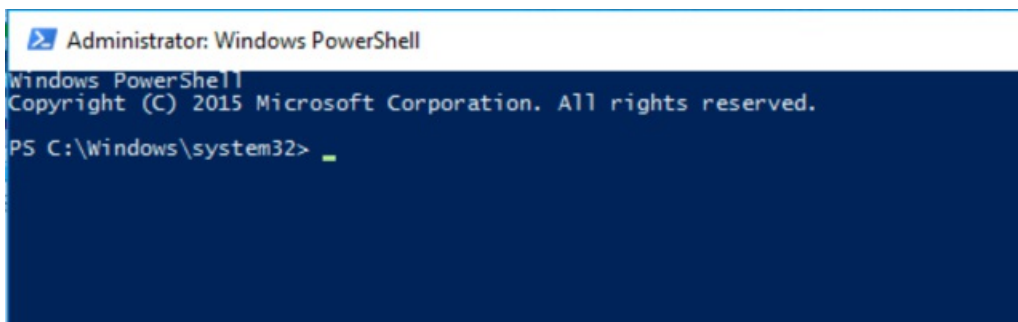
Make sure you run PowerShell as "Administrator". To do this, right-click on the PowerShell icon and select **Run as Administrator** from the popup.



If you can't find it there, you can type **Windows PowerShell** into the Cortana box (lower left of your screen) to have Windows find it for you.



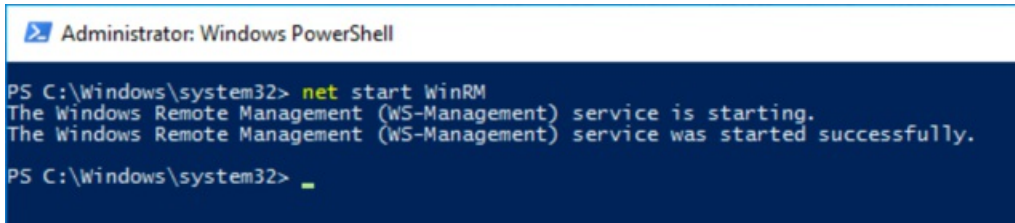
You'll be asked if you want to allow PowerShell to make changes to your system. This is because you ran it as Administrator. Click OK. The PowerShell console window will open.



Initially, PowerShell is open you the `windows\system32` directory on your computer. So, the first step we need to do is to connect PowerShell to your Windows IoT device.

PowerShell needs the **WinRM** service to be running on your computer. On the PowerShell command line, type

```
net start WinRM
```



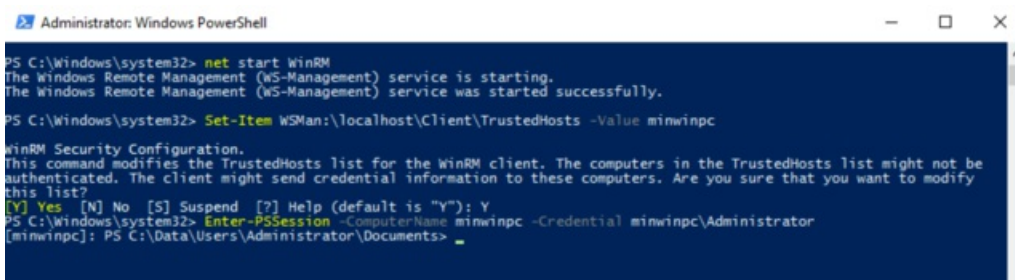
```
Administrator: Windows PowerShell
PS C:\Windows\system32> net start WinRM
The Windows Remote Management (WS-Management) service is starting.
The Windows Remote Management (WS-Management) service was started successfully.
PS C:\Windows\system32> _
```

The next step is to establish a “trusted” relationship between your computer and the Windows IoT device. This is done with the **Set\_Item** command:

```
Set-Item WSMan:\localhost\Client\TrustedHosts -Value <machine-name or IP address>
```

Substitute the name of your IoT device, or use its IP address. Both are available on the standard Windows IoT start screen. Our Raspberry Pi still has the default name of **minwinpc**:

Windows will ask you to confirm that you want to modify the list of trusted computers. Reply ‘Y’ to confirm:

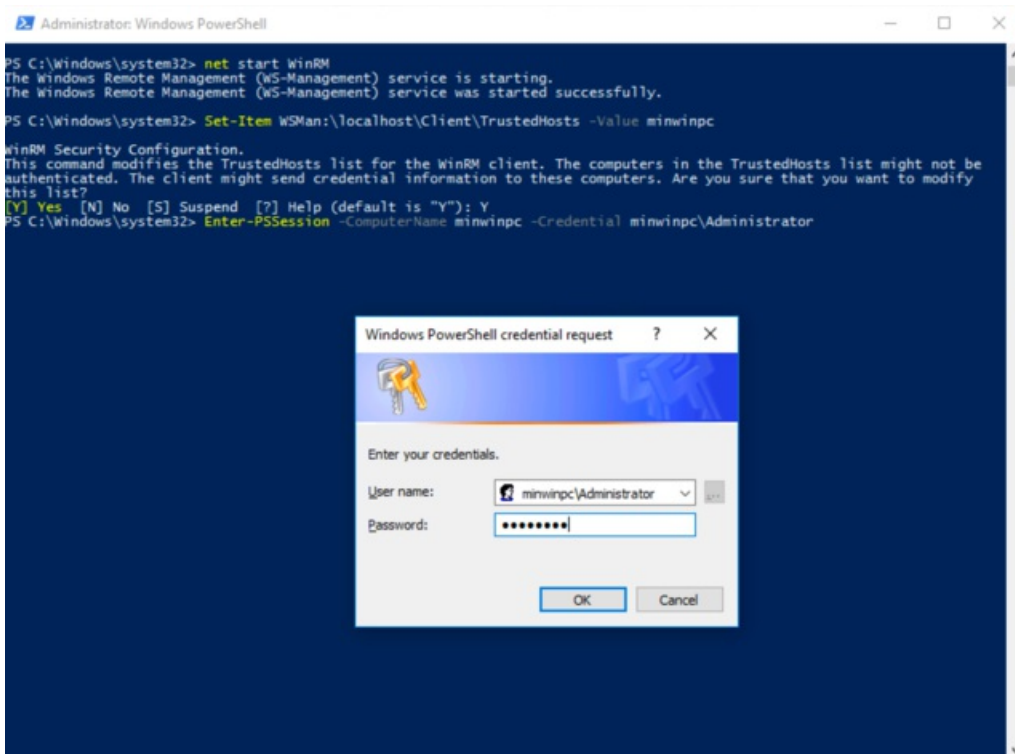


```
Administrator: Windows PowerShell
PS C:\Windows\system32> net start WinRM
The Windows Remote Management (WS-Management) service is starting.
The Windows Remote Management (WS-Management) service was started successfully.
PS C:\Windows\system32> Set-Item WSMan:\localhost\Client\TrustedHosts -Value minwinpc
WinRM Security Configuration.
This command modifies the TrustedHosts list for the WinRM client. The computers in the TrustedHosts list might not be authenticated. The client might send credential information to these computers. Are you sure that you want to modify this list?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
PS C:\Windows\system32> Enter-PSSession -ComputerName minwinpc -Credential minwinpc\Administrator
[minwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

With that out of the way, we can start a command line session on the Raspberry Pi. The command looks like this

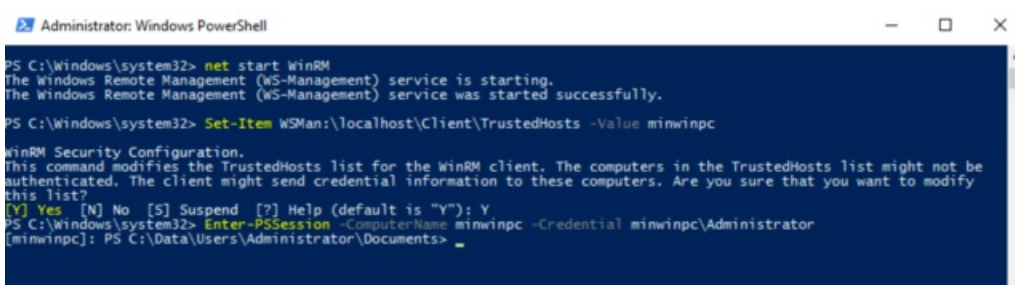
```
Enter-PSSession -ComputerName <machine-name or IP Address> -Credential <machine-name or IP Address or loc
```

Again, substitute the name or IP address of your Raspberry Pi. Mine is still **minwinpc**.



When you execute the `Enter-PSession` command, an authentication window will pop up. Enter the Windows IoT password for your Raspberry Pi. The default password is `p@ssw0rd`.

Enter your password and click **OK**. It may take a minute or so for the command to complete. When it does, you'll see a new command-line prompt, showing that you are in a `Documents` directory. You are now ready to execute PowerShell commands on your Windows IoT Core Raspberry Pi:



## PowerShell: Overview

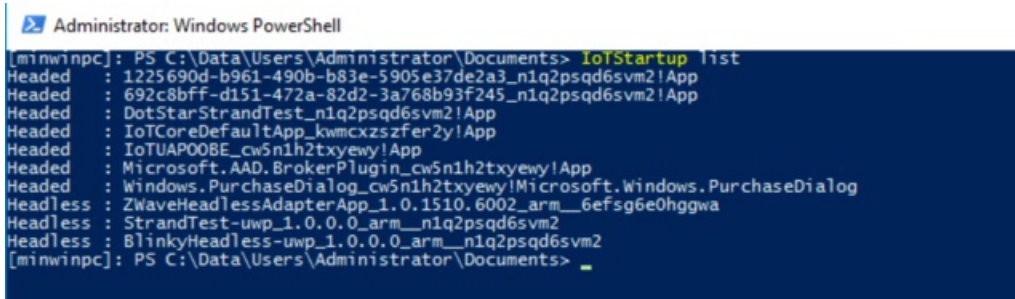
We'll be using PowerShell to set our startup applications, but it can do a lot more. See this page for a summary of the Windows IoT PowerShell commands

<https://adafru.it/pse>

<https://adafru.it/pse>

Meanwhile, let's get started by listing the apps installed on the Windows IoT Raspberry Pi. In PowerShell, enter the command

```
IoTStartup list
```



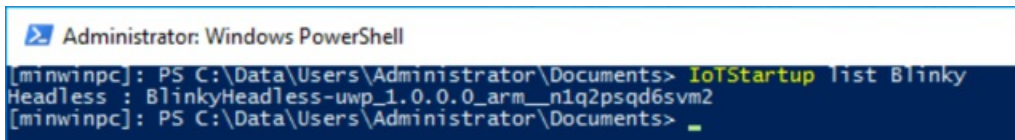
```
Administrator: Windows PowerShell
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup list
Headed : 1225690d-b961-490b-b83e-5905e37de2a3_n1q2psqd6svm2!App
Headed : 692c8bff-d151-472a-82d2-3a768b93f245_n1q2psqd6svm2!App
Headed : DotStarStrandTest_n1q2psqd6svm2!App
Headed : IoTCoreDefaultApp_kwmcxszfer2y!App
Headed : IoTUAP00BE_cw5n1h2txyewy!App
Headed : Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy!App
Headed : Windows.PurchaseDialog_cw5n1h2txyewy!Microsoft.Windows.PurchaseDialog
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm__6efsg6e0hggwa
Headless : StrandTest-uwp_1.0.0.0_arm__n1q2psqd6svm2
Headless : BlinkyHeadless-uwp_1.0.0.0_arm__n1q2psqd6svm2
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

This shows us that we have several Headed apps and a few Headless apps as well. The Headless apps include the BlinkyHeadless app, and a Dotstar app we've been working on for a future project tutorial.

In general, the application names are kind of dense looking. Lots of characters and numbers. You can pretty much ignore the clutter, because PowerShell can do pattern matching. So, instead of laboriously typing the entire app name into a command, you can just type part of it. For instance, we can enter the command

```
IoTStartup list Blinky
```

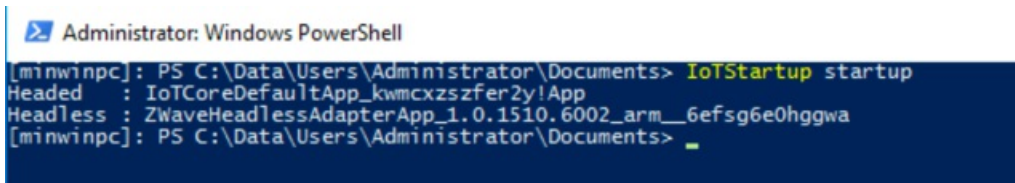
and PowerShell will just list the apps that match:



```
Administrator: Windows PowerShell
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup list Blinky
Headless : BlinkyHeadless-uwp_1.0.0.0_arm__n1q2psqd6svm2
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

In this tutorial, we're interested in setting startup apps. We can list the currently registered startup apps with the `IoTStartup startup` command

```
IoTStartup startup
```



```
Administrator: Windows PowerShell
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup startup
Headed : IoTCoreDefaultApp_kwmcxszfer2y!App
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm__6efsg6e0hggwa
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> _
```



The default headed startup app is *IoTCoreDefaultApp*. This is the app that put up the information screen on HDMI when you first booted your Pi. *ZWaveHeadlessAdapterApp* implements Z-Wave, a wireless communications protocol designed to allow devices in the home (lighting, appliances, etc.) to communicate with one another for home automation (see here: <https://developer.microsoft.com/en-us/windows/iot/samples/zwavetutorial> (<https://adafru.it/pFA>))

In the next section, we'll register BlinkyHeadless as a startup app.

## Headless Apps at Startup

If you haven't yet gone through the BlinkyHeadless tutorial, now is a good time to do so.

You should have BlinkyHeadless running from Visual Studio. That means that the app is already “deployed” on the Raspberry Pi.

<https://adafru.it/psf>

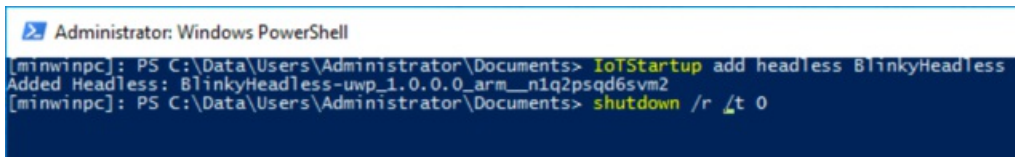
<https://adafru.it/psf>

In PowerShell, enter the command

```
IoTStartup add headless BlinkyHeadless
```

Now, restart your pi from PowerShell. The easy way to do this is to enter the command

```
shutdown /r /t 0
```



```
Administrator: Windows PowerShell
[MinWinPC]: PS C:\Data\Users\Administrator\Documents> IoTStartup add headless BlinkyHeadless
Added Headless: BlinkyHeadless-uwp_1.0.0.0_arm__n1q2psqd6svm2
[MinWinPC]: PS C:\Data\Users\Administrator\Documents> shutdown /r /t 0
```

Your Pi will perform an orderly shutdown, and will then reboot. This may take a few minutes, but you should see your LED start blinking when the restart is complete.

You can have as many headless startup apps as you like.

## Headed Apps at Startup

The procedure for setting a headed app to run at startup is virtually identical to the procedure for setting up a headless app.

In the previous section, we rebooted the Pi, and (hopefully) the LED started blinking.

Have a look at the IoT Dashboard. When the restart is complete, you should see your Pi reappear in the **My devices** list. From the dashboard, open the **Device Portal** and navigate to the **Apps** page. You should see **BlinkyHeadless** in the list of Running apps.

Since we're going to replace BlinkyHeadless with BlinkyHeaded, find **BlinkyHeadless** in the list of running apps and click the **X** next to its name to stop it.

(We have to remove **BlinkyHeadless** because it uses the same GPIO pin as **BlinkyHeaded**. You can't run two apps that access the same pin!)

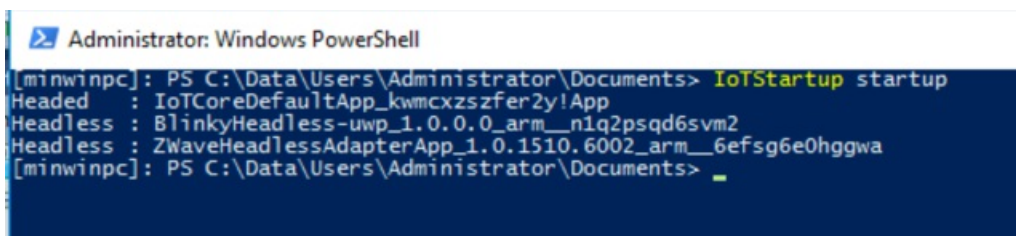
Now, make sure **BlinkyHeaded** runs correctly from Visual Studio (at the time of this writing there was a deployment bug in Visual Studio that happened when the Pi was restarted. The workaround is to close and reopen Visual Studio after the restart is complete).

Go back to PowerShell on your Windows computer. Since you rebooted, you'll have to reconnect PowerShell to the Pi.

Re-enter the **Set\_Item** command (which establishes a trusted relationship between the computer and the Pi). You'll be asked for your user name and password again. Then re-issue the **Enter-PSSession** command.

Both of these commands were explained in the PowerShell section of this tutorial, so refer back to that if you've forgotten them.

PowerShell should now be reconnected to your Pi. Enter the **IoTStartup** startup command to see that **BlinkyHeadless** is in the list of startup apps:



```
Administrator: Windows PowerShell
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup startup
Headed   : IoTCoreDefaultApp_kwmcxzszer2y!App
Headless : BlinkyHeadless-uwp_1.0.0.0_arm_niq2psqd6svm2
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm_6efsg6e0hggwa
[adminwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

The next step is to remove **BlinkyHeadless** as a startup app, and replace it with **BlinkyHeaded**.

Enter the following command to remove **BlinkyHeadless** from the startup list

```
IotStartup remove headless BlinkyHeadless
```

And enter the following command to add BlinkyHeaded to the startup list

```
IotStartup add headed BlinkyHeaded
```

<https://adafru.it/psf>

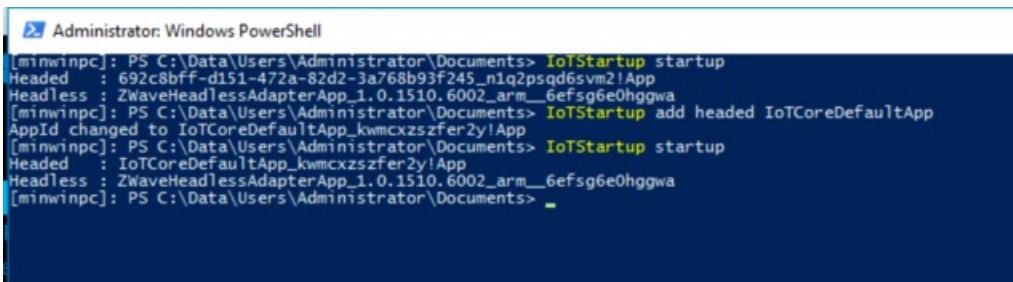
Let's review: We stopped BlinkyHeadless, we removed BlinkyHeadless from the list of startup apps, and we added BlinkyHeaded to the list of startup apps. Enter the restart command in PowerShell.

You should already have an HDMI monitor attached to the Pi, from when you first developed BlinkyHeaded. Restart the Pi with the `shutdown /r /t 0` command. When the boot completes, you should see the BlinkyHeaded GUI on the monitor, and the LED should be blinking.

## Wait – what happened to the original Windows IoT Core GUI?

When you set BlinkyHeaded as the startup headed app, you replaced the default headed startup app. There can be only one headed startup app. But you can go back to the original app. It's called `IoTCoreDefaultApp`, and you can set that back to be the headed startup app in PowerShell.

In this example, we issued the `IoTStartup startup` command to list the current startup apps. Then we issued the command `IoTStartup add headed IOTCoreDefaultApp` to set the headed startup app back to the default. Issuing the `IoTStartup startup` command again shows the change:



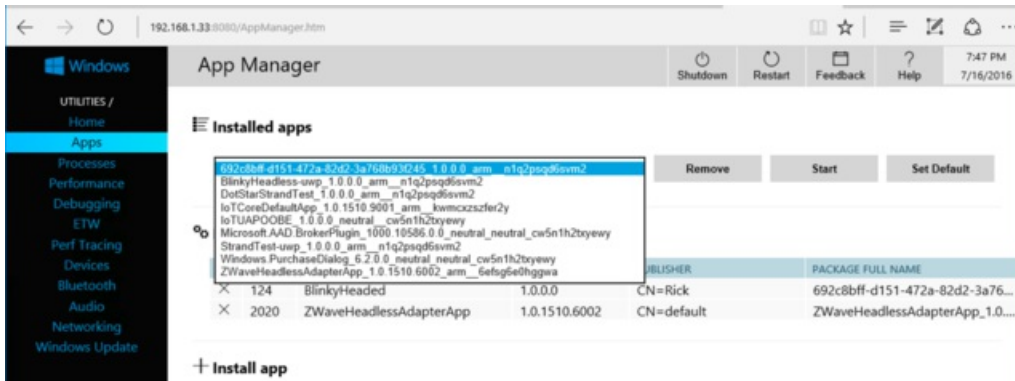
```
Administrator: Windows PowerShell
[Minwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup startup
Headed : 692c8bff-d151-472a-82d2-3a768b93f245_n1q2psqd6svm2!App
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm__6efsg6e0hggwa
[Minwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup add headed IOTCoreDefaultApp
AppId changed to IOTCoreDefaultApp_kwmcxszf2y!App
[Minwinpc]: PS C:\Data\Users\Administrator\Documents> IoTStartup startup
Headed : IOTCoreDefaultApp_kwmcxszf2y!App
Headless : ZWaveHeadlessAdapterApp_1.0.1510.6002_arm__6efsg6e0hggwa
[Minwinpc]: PS C:\Data\Users\Administrator\Documents> _
```

## Help! BlinkyHeadless or BlinkyHeaded don't appear in the list of apps!

You may run into a situation where PowerShell doesn't list an app matching the name BlinkyHeadless or BlinkyHeaded. In my system, for example, there is no app named BlinkyHeaded. So, we'll use BlinkyHeaded as an example in this section. Substitute BlinkyHeadless in this section if that's the one you're having trouble with.

Here's where the dense, encoded-looking application names come into play.

When you run BlinkyHeaded from Visual Studio, you'll see BlinkyHeaded in the list of running apps on the Device Portal Apps page:



See that "Remove" button in the App Manager, next to the list of installed apps? Unless you are very sure what you are doing, don't use that button. You could inadvertently remove an important app, forcing you to re-image your Windows IoT system!

You can see BlinkyHeaded in the running apps list, but it doesn't seem to appear in the list of installed apps. Well, actually, it does appear in the list of installed apps, but the name is encoded. In the picture, look at the "Package Full Name", all the way to the right of the BlinkyHeaded line. You'll see that the package full name starts with **692c8bff...** . Now look at the list of installed apps. You'll see that the first app in that list *also* starts with **692c8bff...** It's the BlinkyHeaded app in disguise!

Your BlinkyHeaded app will have a different full package name. Use that in PowerShell to set the Headed startup app. I'll use **692c8bff** here – but you should use the first few characters of your Blinky app's package name

```
IoTStartup add headed 692c8bff
```

It's possible that you won't have this problem on your system. But if you do, you can apply this solution to the problem.