Session 2: FLASHING IMAGE AND USE DEVELOP MENT TOOLS

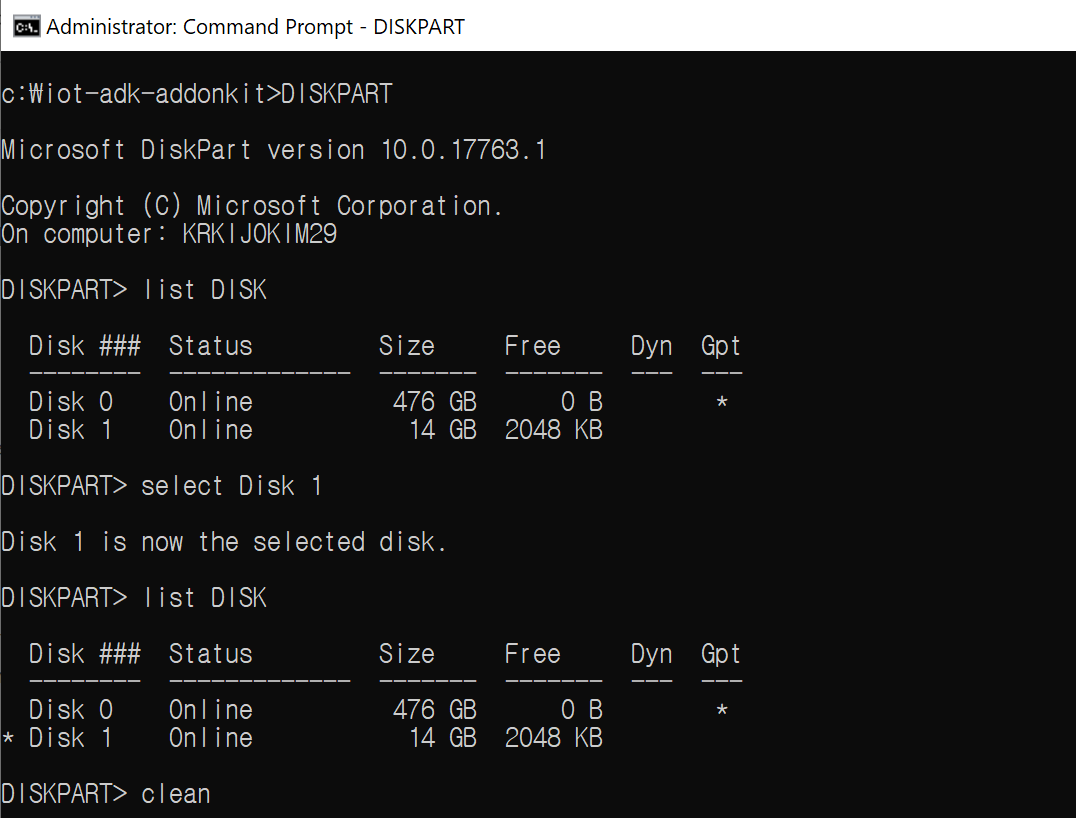
# Flashing image and use development tools

**PURPOSE: Understanding how to flash image to the device and get familiar with many development tools for IoT Core**

Reference – [Flashing Image](https://github.com/MicrosoftDocs/windows-iotcore-docs/blob/fabricam/windows-iotcore/manufacturing-guide/05-FlashingImage.md)

**FLASHING an IMAGE**

* Insert a USB Card Reader with your microSD card
* Run cmd as administrator and run DISKPART
* In DISKPART, clean disk partition of microSD
  + Run **list disk** to see the list of available disks
  + Run **select disk X**, where X corresponds to the disk number of your USB drive



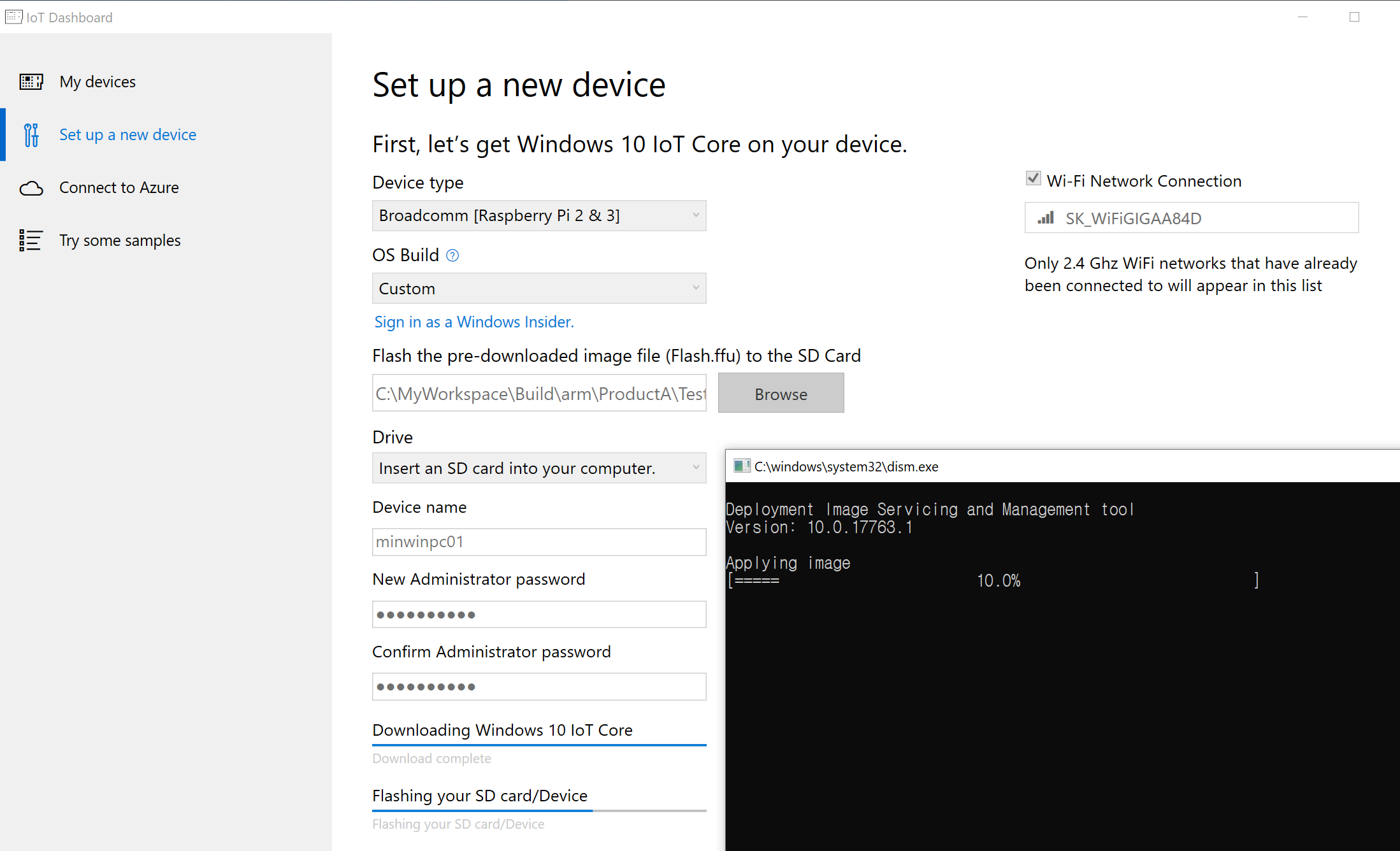
* + Run ***clean*** to clean the selected disk
  + Run ***create partition primary*** to create a primary partition on the disk
  + Run ***format fs=fat32 quick*** to format the drive
  + Run *assign* to assign the drive
  + Run *exit* to exit from diskpart.

**[Note]** If you encounter the following error, run *convert* mbr: “ERROR: Failed to format "F:"; DiskPart errorlevel -2147212244”

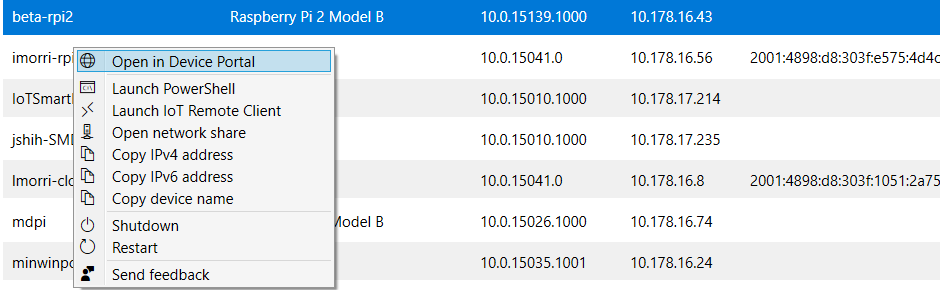
* Start Windows IoT Core Dashboard
* Plug your microSD card into the technician PC, and select it in the tool
* From the *Setup a New Device* section, select Broadcomm [Raspberry Pi 2 & 3] under Device Type
* Select *Custom* under OS Build
* Click Browse and navigate and select the FFU file you created earlier
* Verify your microSD card is listed under the Drive selection
* [**!Important**] You must set the device name to minwinpc[Your Kit Number] to identify your device and Administrator Password selections for your device



* Check “I accept the software license terms checkbox” and click Install



* If Raspberry RPi3 is running on our recommended SD card takes 3-4 minutes for first boot. On the same Pi with poor quality SD card, we have seen boot times longer than 15 minutes.
* Confirm your RPi3 is **connected the Ethernet Hub with LAN** and your technical PC is in the same subnet
* After RPi3 boot, connect to your device, Right click and select *Open in Device Portal*. This will launch the Windows Device Portal page and is the easiest way to interact and manage your device.

[](https://github.com/MicrosoftDocs/windows-iotcore-docs/blob/fabricam/windows-iotcore/media/IoTDashboard/IoTDashboard_RightClickMenu.PNG)

* If devices are not listed then ensure that your windows10iotcoredashboard.exe is allowed to communicate through Windows Firewall by following the steps below:
  + Open Network and Sharing Center and then find the type of network (Domain/Private/Public) your PC is connected to.
  + Open Control Panel and click System and Security.
  + Click Allow an app through Windows Firewall under Windows Firewall.
  + Click Change settings.
  + Find windows10iotcoredashboard.exe in Allowed apps and features and then enable the appropriate network check box

**[Tip]** If you are unable to find your device in the dashboard

Try typing your [IP Address] and [:8080] into the browser to get Windows Device Portal up and running. To get your device to show in the dashboard, try rebooting your device

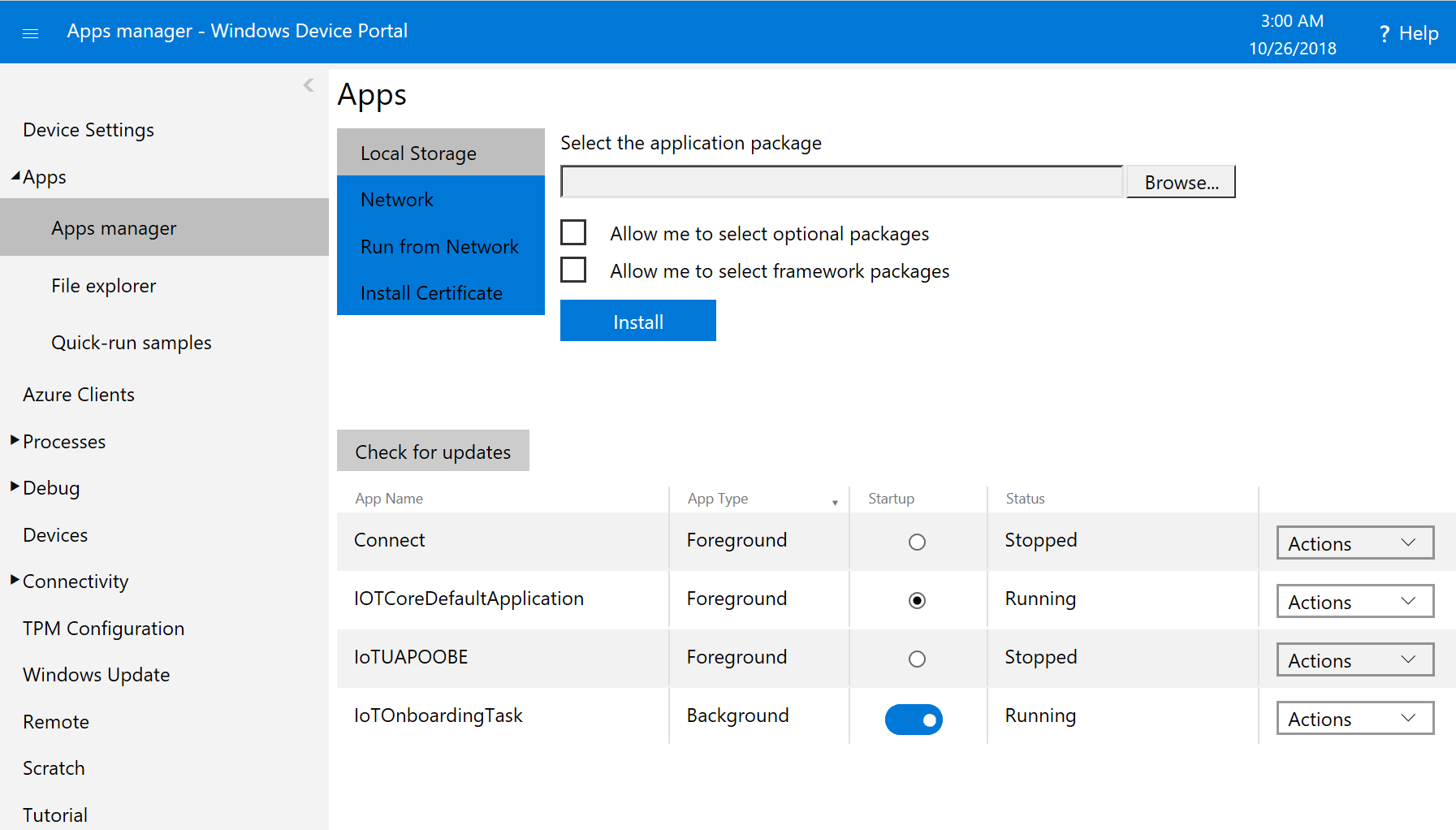
# use development tools - DEVICE PORTAL & WINDBG

**PURPOSE: Get familiar with Device Portal tool and Windbg for dump investigation**

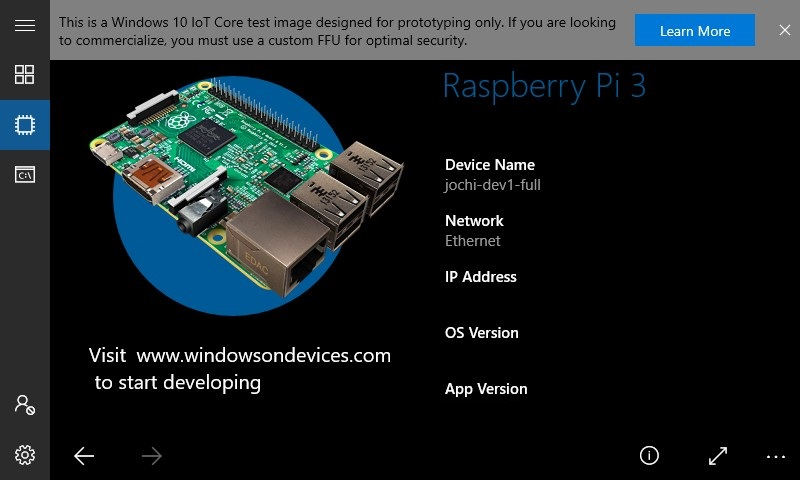
Reference – [Flashing Image](https://github.com/MicrosoftDocs/windows-iotcore-docs/blob/fabricam/windows-iotcore/manufacturing-guide/05-FlashingImage.md)

**DEVICE PORTAL & WINDBG**

* On Device Portal, go to Apps -> Apps Manager and select package files in **IoTBrowserApp** under your HOL directory. And, add dependent packages as framework package.
* Try to run and set the default app.



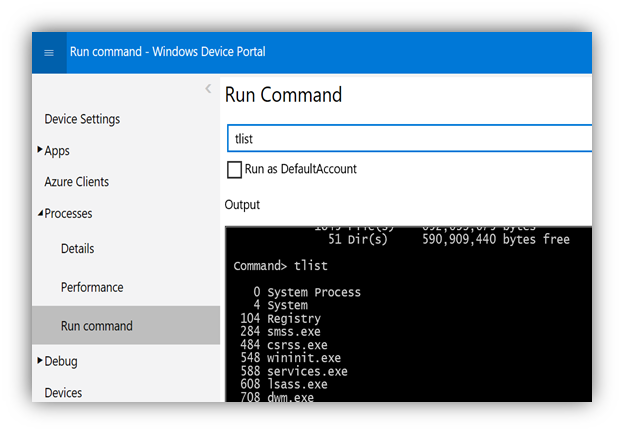
* Keep the IoTBrowserApp for future lab and change start up to IOTCoreDefaultApplication



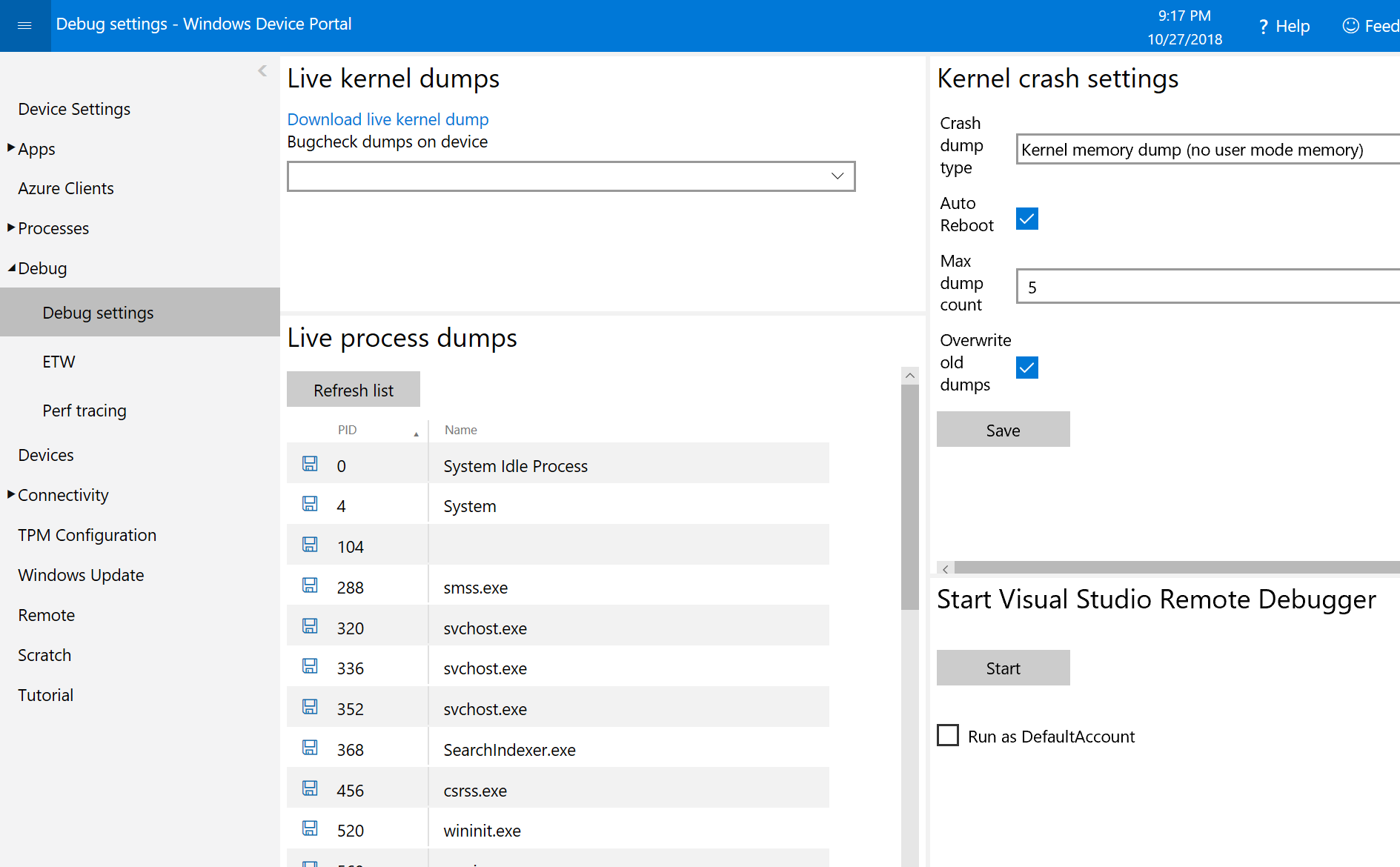
* Go to Processes tab, check process information

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

* Run command to run windows commands



* Go to Debug tab, capture a user memory dump for any live process and captured kernel dump by Crash control setting. You can also change Kernel crash setting to full, kernel or mini dump. Try to capture any user dump or download captured kernel dumps



* Run Screen capture tool and check current UX.

**[OPTIONAL]**

* Run windbg and load the dump file and try windbg commands
* Run cmd as Administrator
* If your host computer is x64 then use windbg(x64) otherwise use windbg(x86)

|  |  |
| --- | --- |
| X64 | X86 |
| *C:\windows\system32> set Path=%Path%;C:\Program Files (x86)\Windows Kits\10\Debuggers\x64*  *C:\windbg* | *C:\windows\system32> set Path=%Path%;****C:\Program Files (x86)\Windows Kits\10\Debuggers\x86***  ***C:\windbg*** |

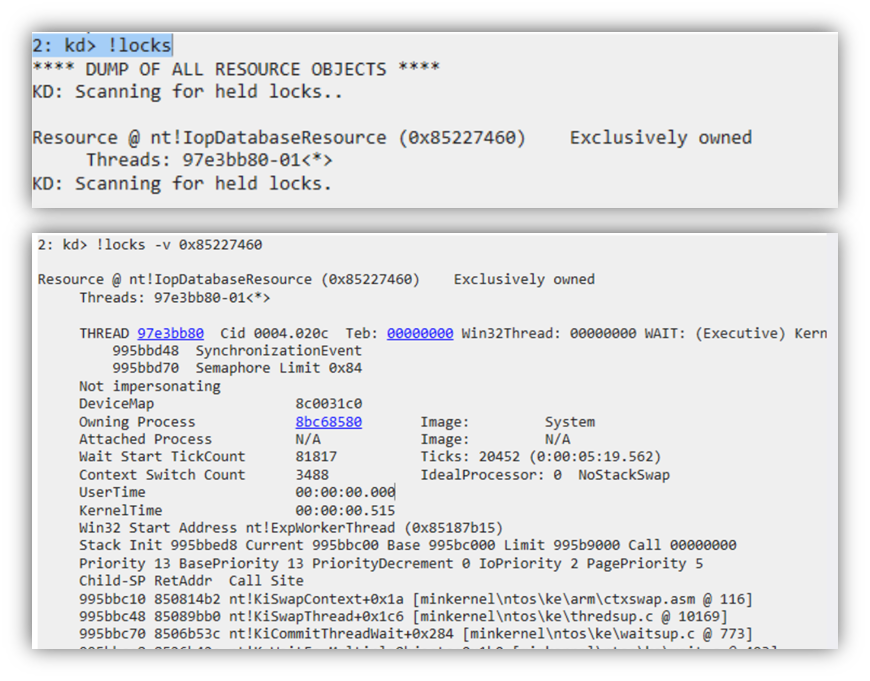
* Set *\_NT\_SYMBOL\_PATH* to set public symbol server

*set \_NT\_SYMBOL\_PATH=srv\*DownstreamStore\*https://msdl.microsoft.com/download/symbols*

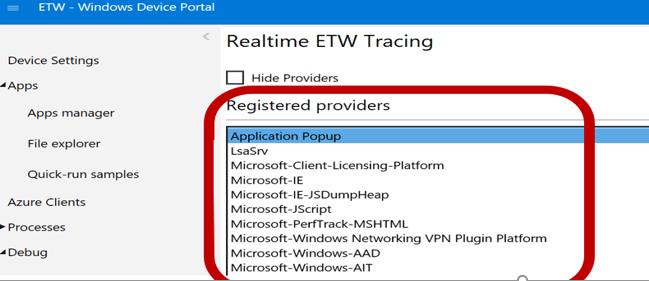
* Load dump file

*Windbg -z [path to your dump file]*

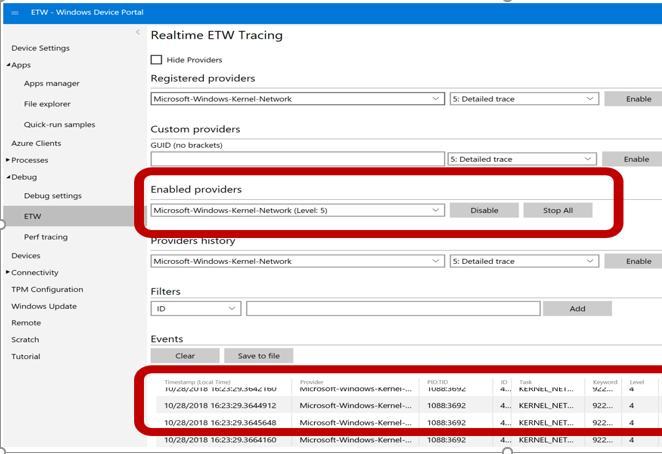
* Run *!analyze* and other windbg commands like *!locks, !kvm, !process 0 0, .dv*. More commands – [link](https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/commands)



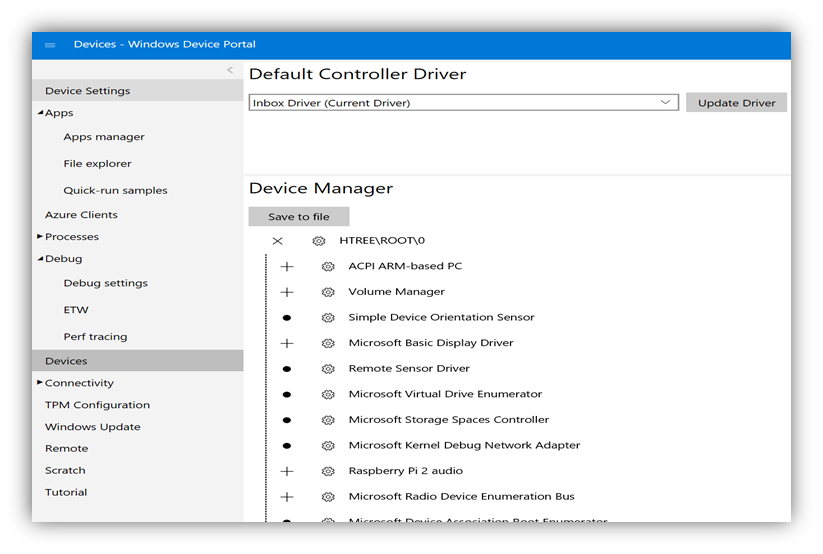
* Go to Debug > ETW tab, register provider from the list and enable



* Check enabled providers drop down box and then live event information displayed and can save it. Try USB or network logs.

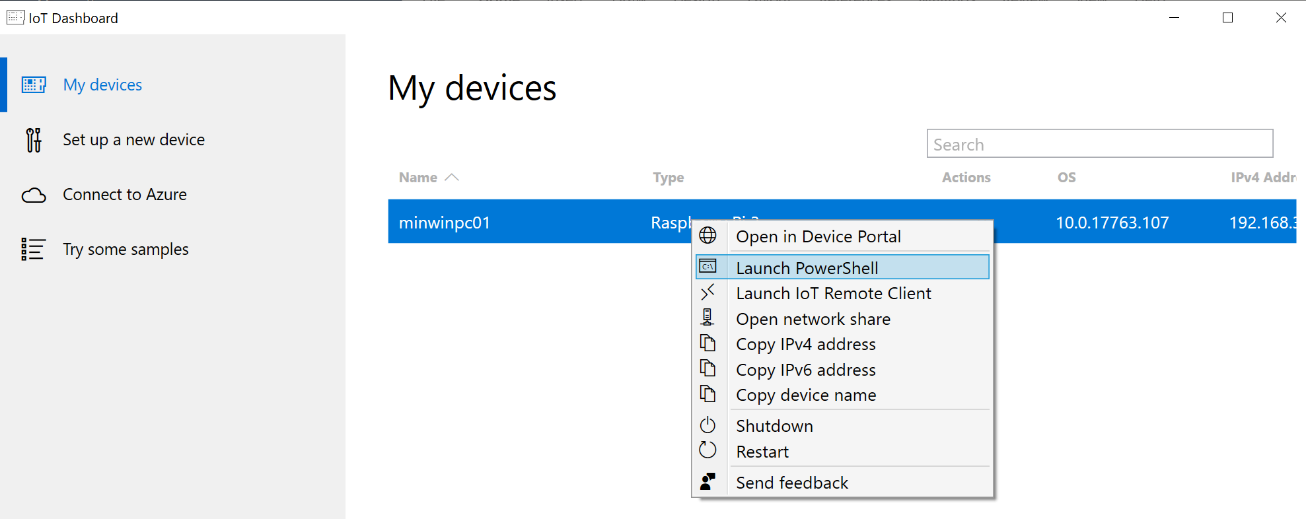


* Go to Devices and move to “Device Manager” and check enumerated drivers



**POWERSHELL**

* Connect to your device, Right click and select “Launch PowerShell”.



* You will get prompted logon dialog box
* Type in correct credential then powershell will be connected to the device
* If you failed to get powershell connection errors then try it as below
* To start PowerShell as an administrator, right-click Windows PowerShell and then select Run as administrator.
* You may need to start the WinRM service on your desktop to enable remote connections. To do so, from the PowerShell console, type the following command:

*net start WinRM*

* From the PowerShell console, type the following to put the target device into trusted host list an suggest useing IP address:

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

* Now you can start a session with your Windows IoT Core device. From you administrator PowerShell console, type:

*Enter-PSSession -ComputerName <machine-name or IP Address> -Credential <machine-name or IP Address>\Administrator*

* Enable FTP
* On Device with Powershell, start FTP

*PS C:\> start ftpd*

* On PC, connect with ftp://[ip] with file exploer
* On Device with Powershell, stop FTP

*PS C:\> kill -processname ftpd\**

* Filesharing
* File sharing server on your Windows IoT Core device starts automatically on boot by default
* On PC, File Explorer on your computer and type [*\\[Target*](file:///\\[Target) *Device IP or Name] \c$*
* If it’s not accessible then try to start the filesharing service

*net start Server*

* If you want to stop the file sharing service

*net stop Server /y*

* Try IOT Core command line utilities
  + Commonly used utilities
  + *IotStartup list* lists installed applications
  + *IotStartup list headed* lists installed headed applications
  + *IotStartup list headless* lists installed headless applications
  + *DeployAppx install MyApp.appx* installs the .APPX and the certificate of the same name if found.
  + *DeployAppx uninstall MyApp.appx* uninstalls any installed .APPX with a matching package family name.
  + List of commands and utilities that you can use with PowerShell, see the [Command Line Utils](https://github.com/MicrosoftDocs/windows-iotcore-docs/blob/fabricam/windows-iotcore/manage-your-device/CommandLineUtils.md) , Please check and try commands
* Try Windows Performance Recorder (WPR) & Performance recorder
  + List currently supported predefined profile

*wpr –profiles*

* + Start capturing with profile or custom profile for example for power

*xperf –start power*

* + Stop capturing with ETL file name

*xperf -stop <ETL file name>*

* Driver Verifier – Improve your deriver quality, enable driver verifier for specific driver, query and reset
  + Run “verifier /?”
  + Run “*verifier /standard /driver [my driver file]*” to enable driver verifier
  + Reboot the device “*shutdown /r /t 0”*
  + Run “*verifier /querysettings*” to query current setting
  + Run “*verifier /reset*” to reset driver verifier setting

**[Note]** Driver Verifier monitors Windows kernel-mode drivers and graphics drivers to detect illegal function calls or actions that might corrupt the system. [Link](https://docs.microsoft.com/en-us/windows-hardware/drivers/devtest/driver-verifier)

**[Note]** We also provide pp verifiers for application quality improvement, refer -[Link](https://docs.microsoft.com/en-us/windows/desktop/win7appqual/application-verifier)

* Close session.

*Exit-PSSession*

# quiz

**Q1: Try to flash your image with DSIM tool**

**Q2: Configure your RPi device connected to your AP with your Device Portal Tool**