

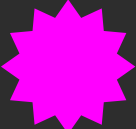


# UEFI & EDK II Training

## PLATFORM BUILD LAB WINDOWS EMULATOR

[tianocore.org](https://tianocore.org)

See also [LabGuide.md](#) for Copy & Paste examples in labs

# PLATFORM BUILD LABS

-  Pin Visual Studio Command Prompt to Windows Task Bar
-  Build a EDK II Platform using Emulator package
-  Run the Emulator in Windows

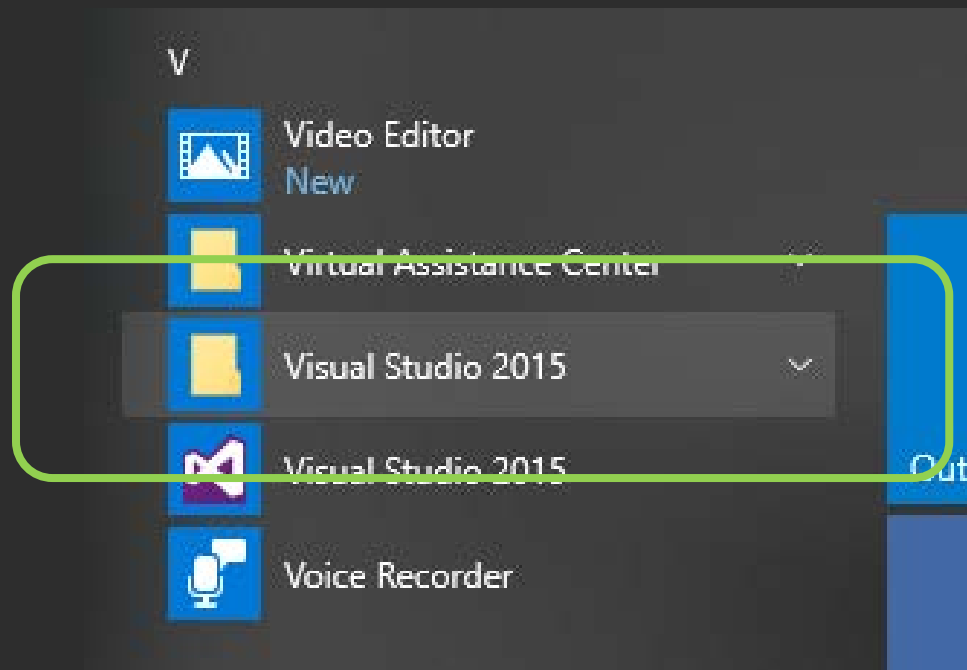
# PIN VS COMMAND PROMPT

Pin the Visual Studio Command prompt to Windows Task Bar

# Pin VS Command Prompt



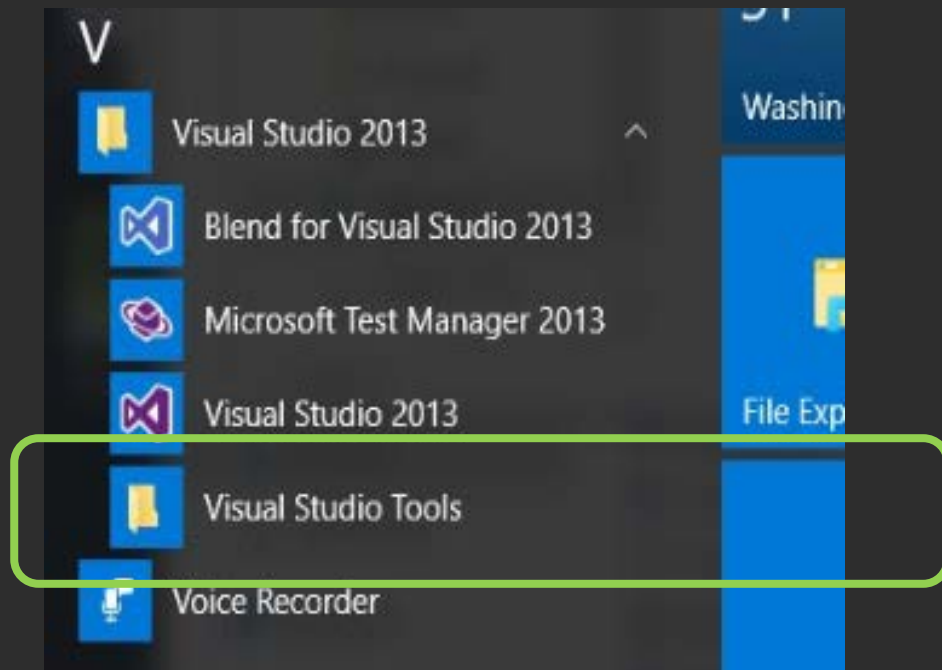
Windows 10



Steps to Pin Visual Studio Command Prompt to task bar for Windows 10

1. Using the Start menu in Windows 10, Left Click on “Windows Key” Lower Left 
2. Scroll down from the scroll bar on the right until “**Visual Studio 201<sup>n</sup>**”
3. Left Click “**Visual Studio 201<sup>n</sup>**”

# Pin VS Command Prompt



VS 2013

## 4. Left Click “Visual Studio Tools”

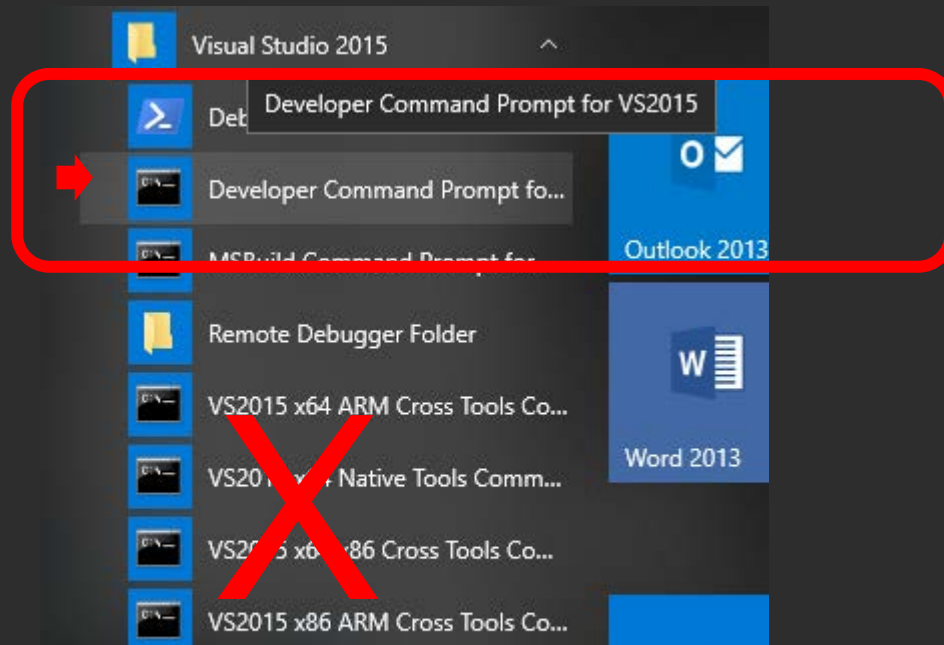
This will open another Windows file explorer window

Note: *VS 2013 example, other version of VS maybe different*

# Pin VS Command Prompt

VS 2015

VS 2017 is similar

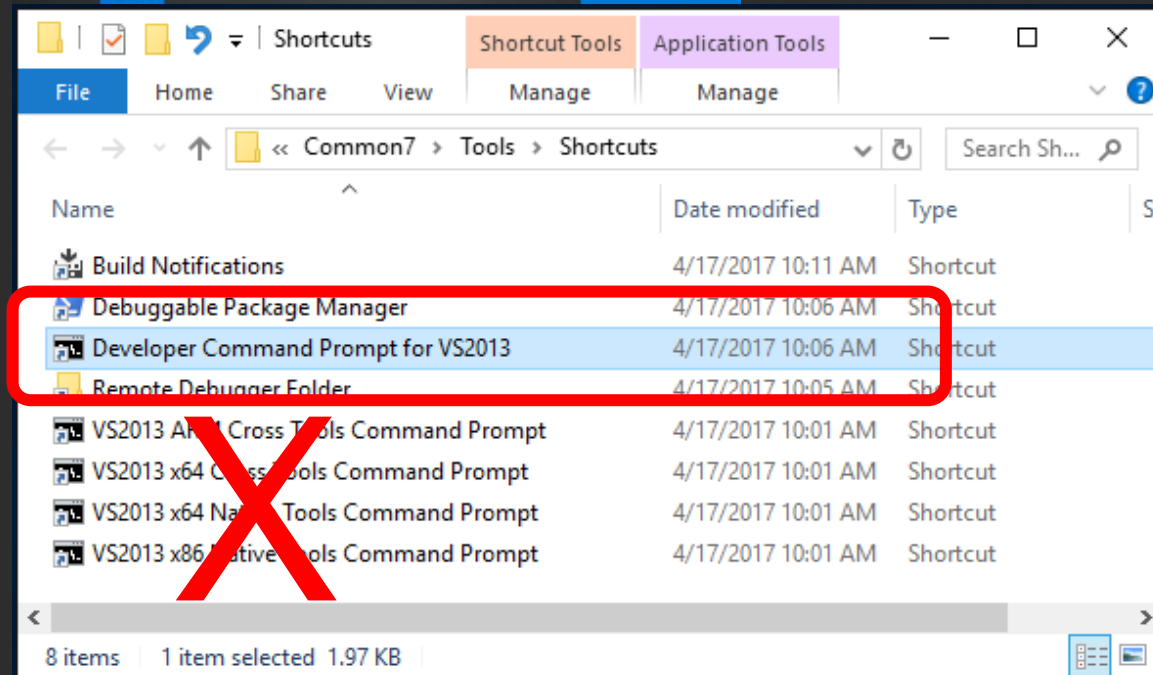


5. Select  
“Developer Command Prompt for VS201n”

6. Right Click to open Windows dialog box

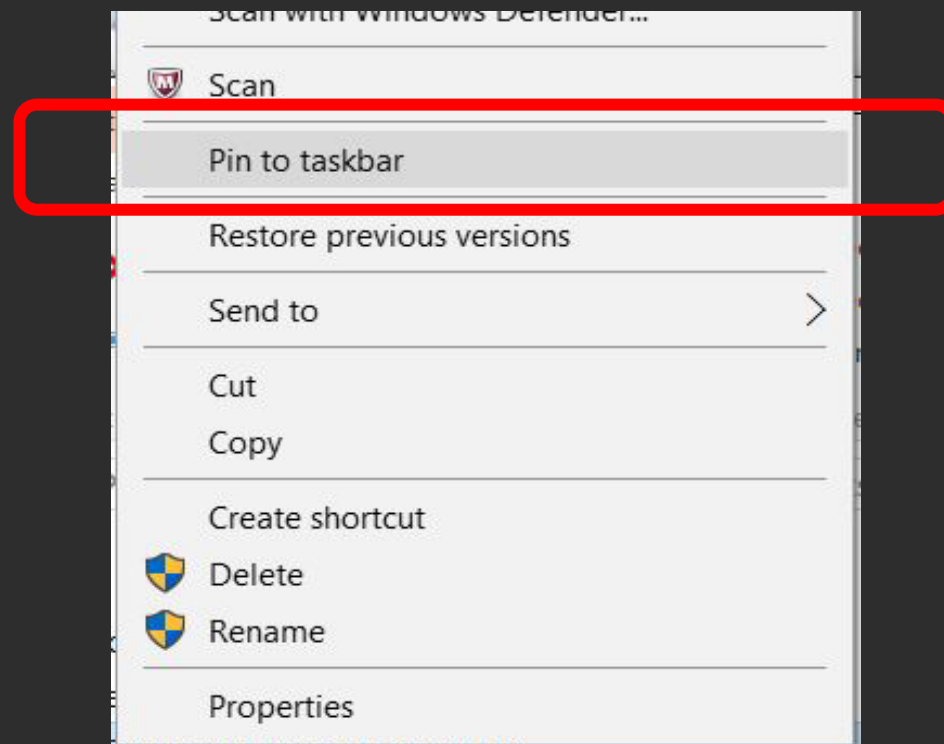
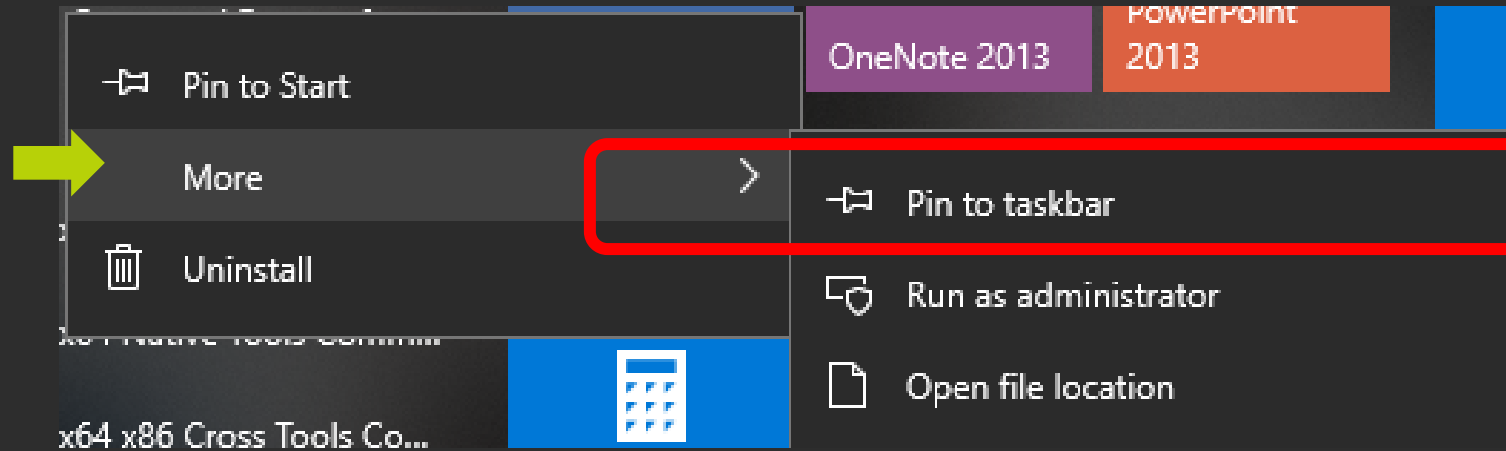
**Do not use any of the other  
“.. Command Prompts”**

VS 2013



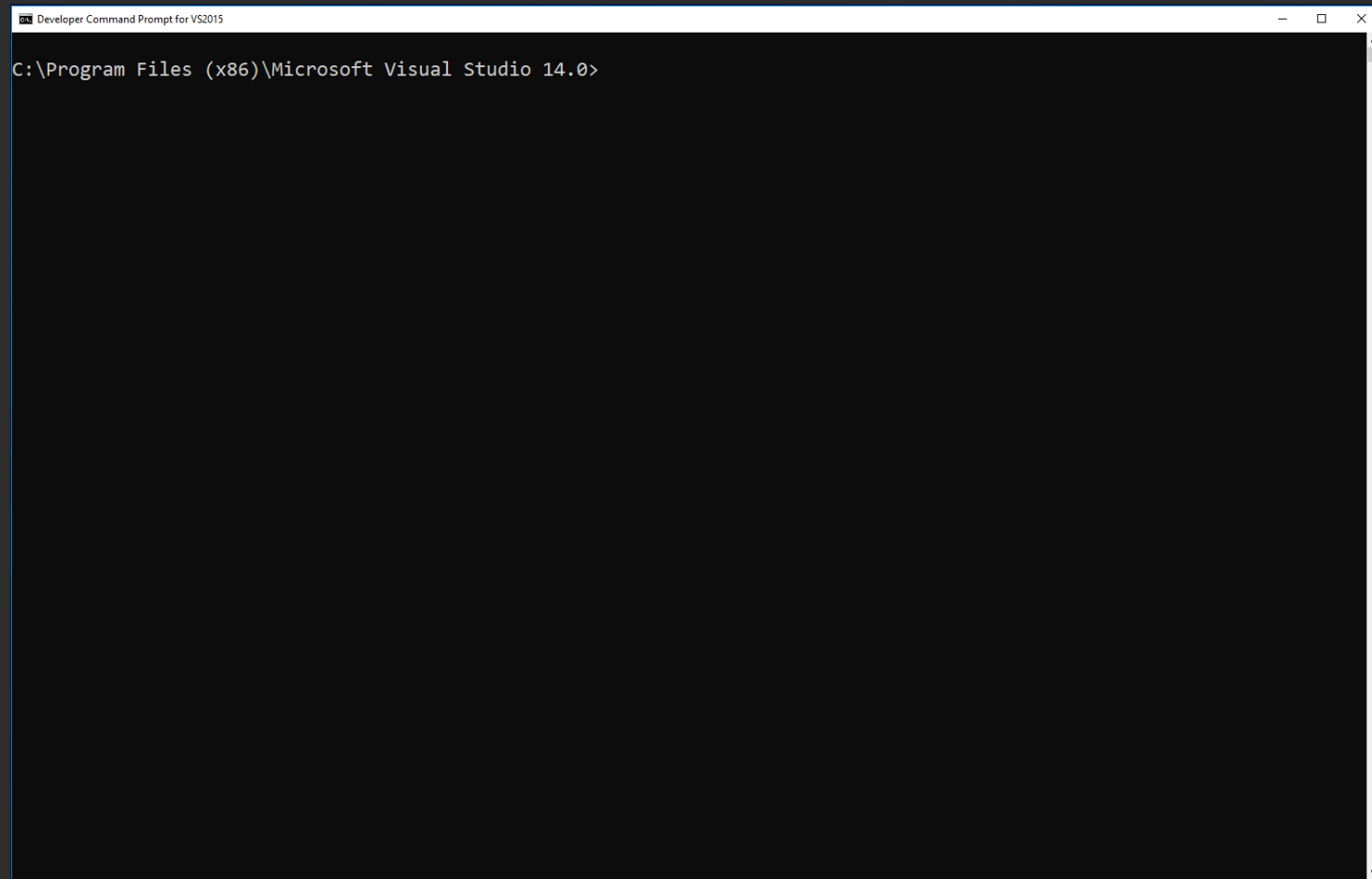
# Pin VS Command Prompt

7. Left Click on  
“Pin to taskbar”





# Pin VS Command Prompt



## 8. Open VS Command Prompt”

All Windows Labs use this short-cut to Build Edk II platforms and projects using Windows Visual Studio :  
2013 / 2015 /2017 or 2019







# END OF PIN VS PROMPT

# BUILD EMULATOR

Setup EmulatorPkg to build and run emulation with Windows

# Prerequisites

– Done Before Class

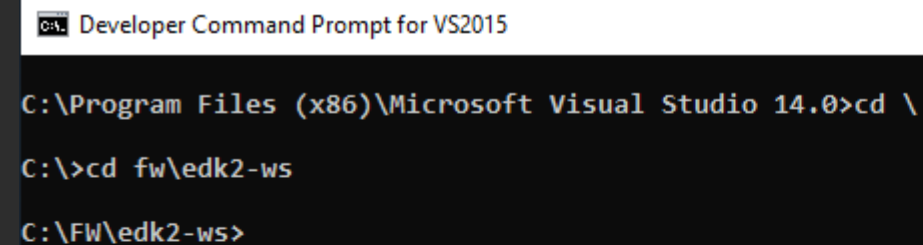
- Windows 10: 
  - Continuous Integration (CI) - Stuart CI Build with Visual Studio VS2017 or VS2019 
  - Non-Stuart CI - Visual Studio VS2015, VS2017 or VS2019
  - Windows SDK (for rc)
  - Windows WDK (for Capsules)
- Python 3.7.x or greater and /Scripts directories on Path: [Link](#) to download  python™
- Git for Windows on Path : [Link](#) 
- NASM for Win64 : [Link](#)

# Create Workspace Directory

Open Windows Command Prompt

Make new directory for Workspace:

```
$ cd /  
$ Mkdir FW  
$ cd FW  
$ Mkdir edk2-ws  
$ cd edk2-ws
```



```
Developer Command Prompt for VS2015  
C:\Program Files (x86)\Microsoft Visual Studio 14.0>cd \  
C:\>cd fw\edk2-ws  
C:\FW\edk2-ws>
```

# Download the EDK II Source Code

Download the open source EDK II from Github 

Note if behind a firewall, set PROXYS FIRST (example shows for Intel corporation)

```
$ git config --global https.proxy proxy-dmz.intel.com:912  
$ git config --global http.proxy proxy-dmz.intel.com:911
```


From the command prompt use “git clone” to download

```
> git clone -b Edk2Lab_22Q1 https://github.com/tianocore-training/edk2.git  
> git clone https://github.com/tianocore/edk2-libc.git
```

Download the Submodules and Checkout the Lab Branch

```
C:\FW\edk2-WS> Cd edk2  
C:\FW\edk2-WS\edk2> git submodule update --init  
C:\FW\edk2-WS> Cd ..
```

# Download Lab Material

Download the Lab\_Material\_FW.zip from :  [github.com Lab\\_Material\\_FW.zip](https://github.com/tianocore-training/Lab_Material_FW.zip)  
OR

Use git clone to download the Lab\_Material\_FW

```
C:\> git clone https://github.com/tianocore-training/Lab_Material_FW.git
```

Directory Lab\_Material\_FW will be created

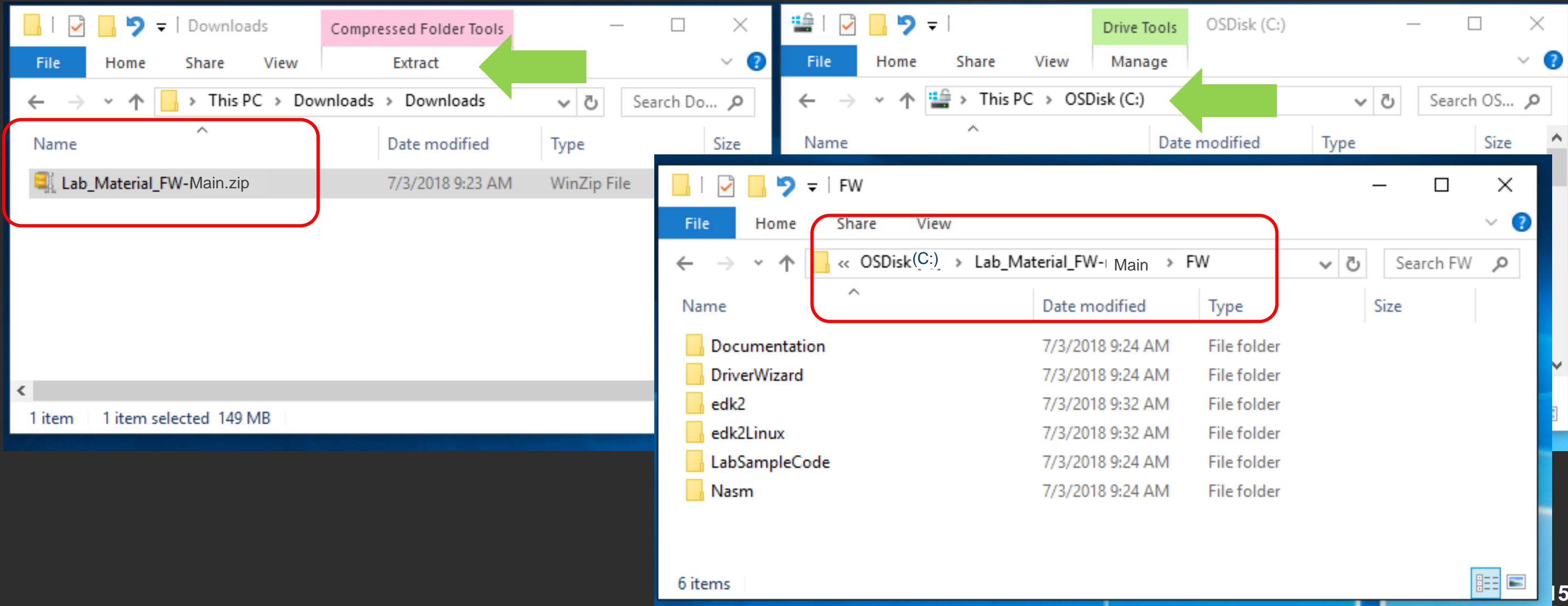
FW

- Documentation
- DriverWizard
- edk2-ws
- LabSampleCode
- Nasm

# Build EDK II

## -Extract the Source

Extract the Downloaded Lab\_Material\_FW-main.zip to C:\



The screenshot illustrates the process of extracting the downloaded Lab\_Material\_FW-main.zip file. It consists of two overlapping File Explorer windows.

**Left Window (Downloads):**

- Address bar: This PC > Downloads > Downloads
- Tab: Compressed Folder Tools
- Action: Extract (highlighted with a green arrow)
- File list:
 

Name	Date modified	Type	Size
Lab_Material_FW-Main.zip	7/3/2018 9:23 AM	WinZip File	
- Status bar: 1 item | 1 item selected 149 MB

**Right Window (OSDisk (C:)):**

- Address bar: This PC > OSDisk (C:) (highlighted with a green arrow)
- Tab: Drive Tools
- Action: Manage
- File list:
 

Name	Date modified	Type	Size
Documentation	7/3/2018 9:24 AM	File folder	
DriverWizard	7/3/2018 9:24 AM	File folder	
edk2	7/3/2018 9:32 AM	File folder	
edk2Linux	7/3/2018 9:32 AM	File folder	
LabSampleCode	7/3/2018 9:24 AM	File folder	
Nasm	7/3/2018 9:24 AM	File folder	
- Status bar: 6 items

The right window also shows a nested view of the extracted files, with the address bar indicating the path: << OSDisk(C:) > Lab\_Material\_FW-main > FW. The file list in this view is empty, suggesting the files are currently hidden or the view is not yet fully updated.

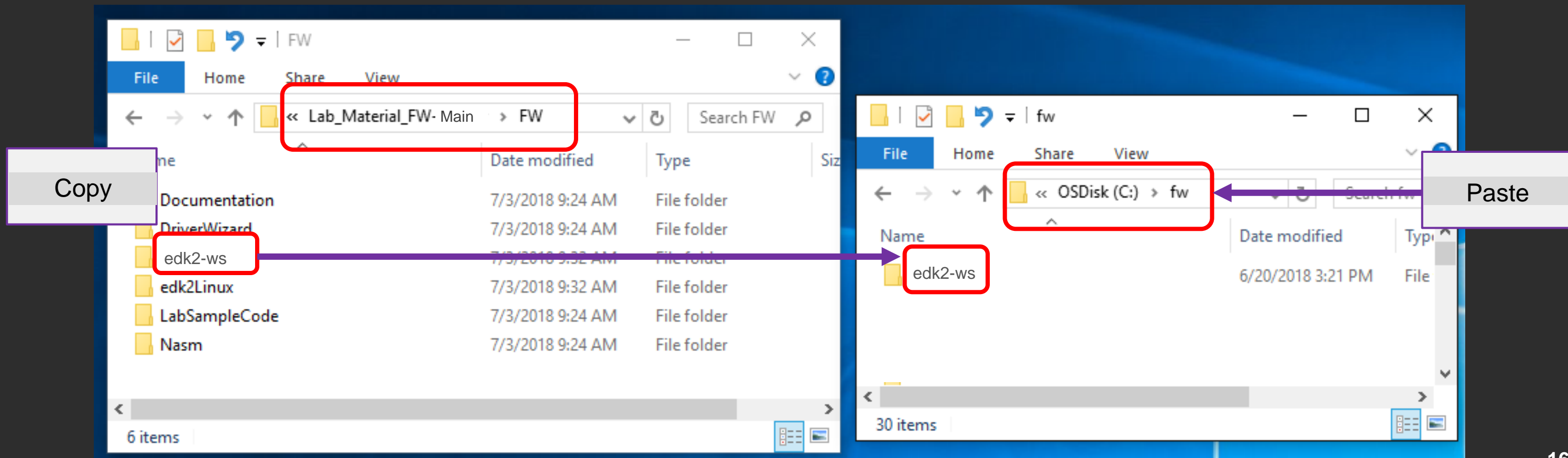


# Build EDK II

## - Copy edk2-ws

From the downloaded Lab\_Material\_FW folder, **copy** and **paste** folder “..\edk2-ws” to C:/FW

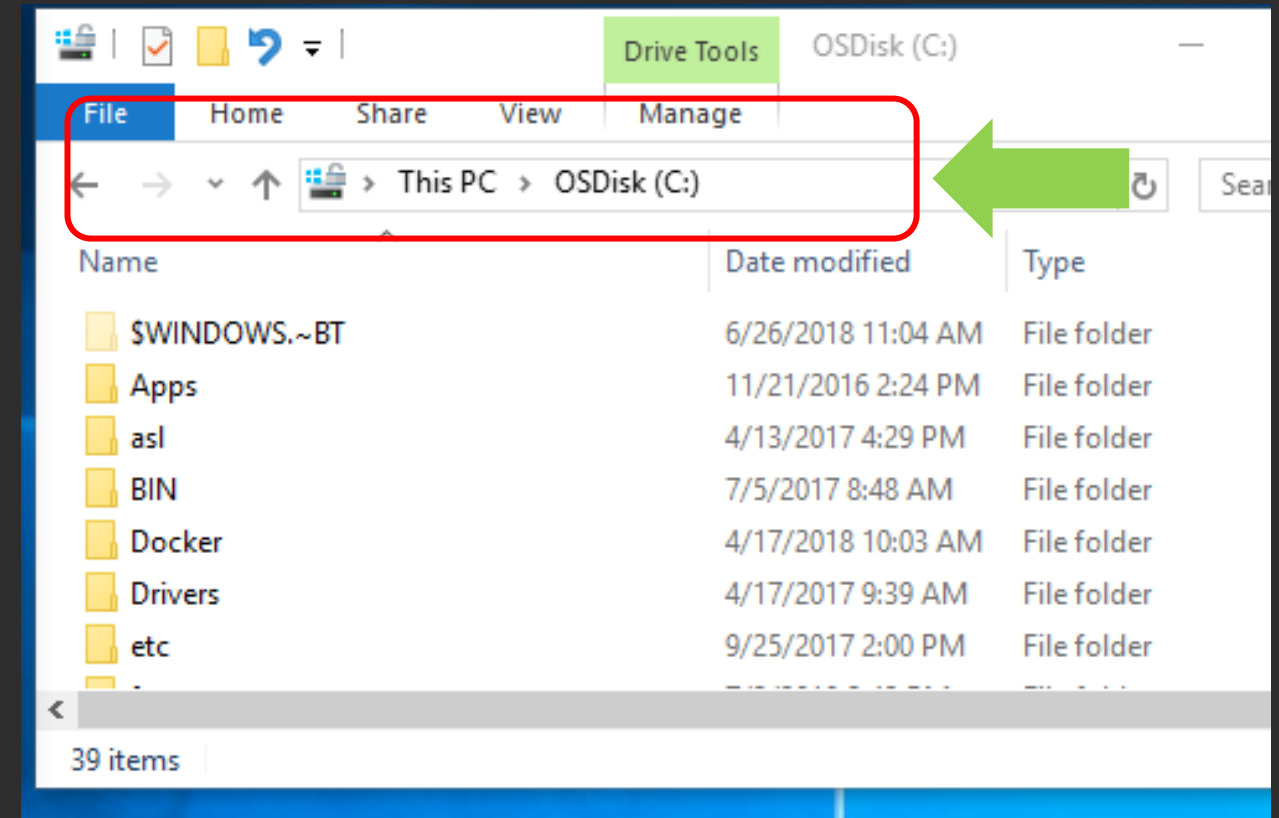
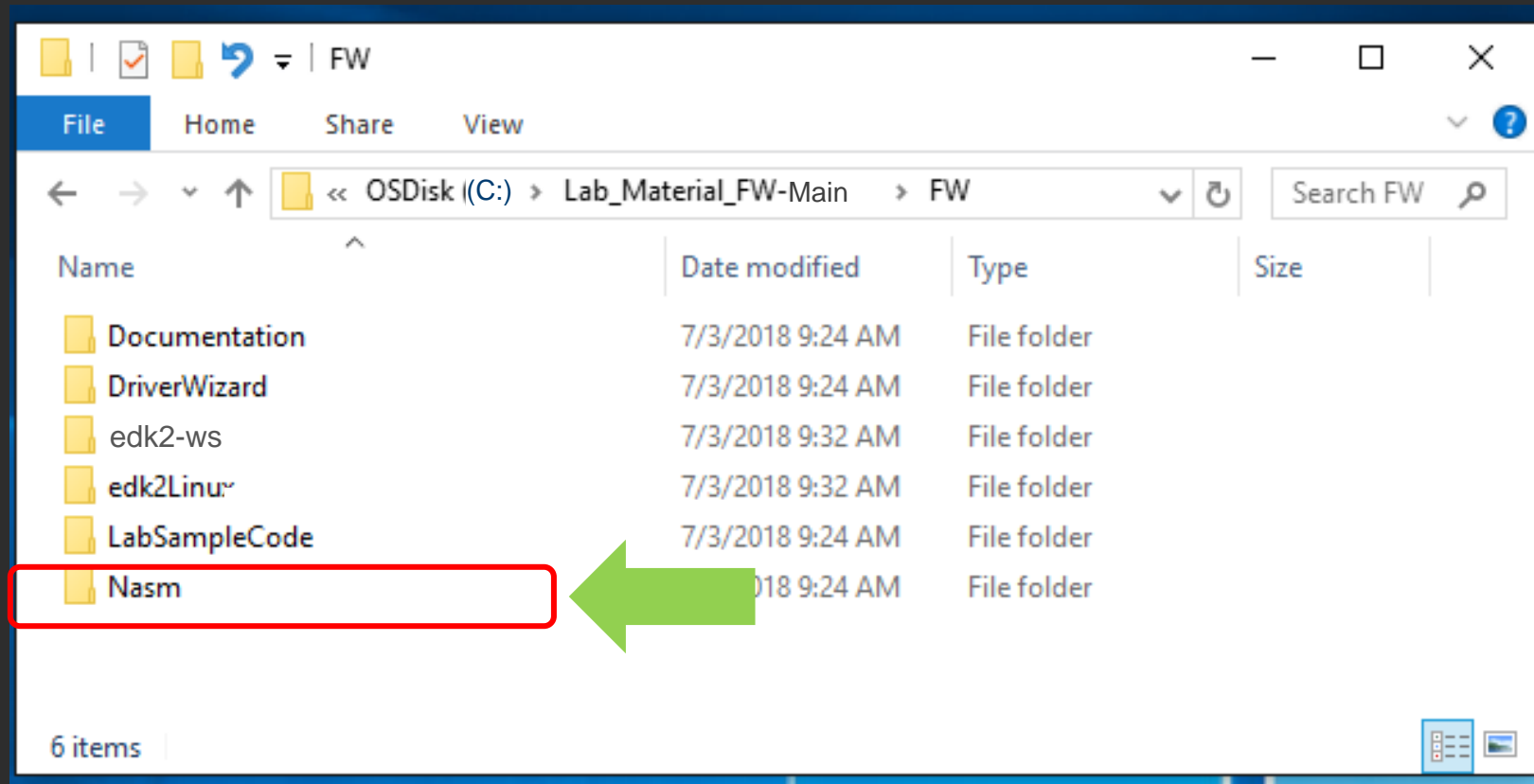
*Note:* Overwrite existing files and directories



# Build EDK II

## -Get Nasm

Copy Nasm directory to C:\  
(creating C:\Nasm directory)



# NON-STUART CI BUILD EMULATORPKG

**Skip** if doing Stuart CI

Note: will need to update conf/target.txt for other labs

# Non-Stuart CI Build EDK II

## - build BaseTools

Open VS Command prompt & Cd to workspace directory

```
$> cd C:\FW\edk2-ws
```

Setup the local environment: (see batch file [setenv.bat](#) )

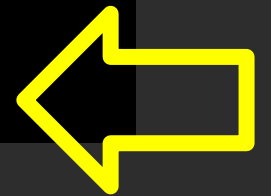
```
$> set WORKSPACE=%CD%
```

```
$> set PACKAGES_PATH=%WORKSPACE%\edk2;%WORKSPACE%\edk2-libc
```

Invoke Edksetup.bat from directory C:/FW/edk2-ws/edk2 to Build BaseTools

```
$> cd edk2
```

```
$> edksetup.bat Rebuild
```



Building BaseTools only needs to be done once but setting up local environment and edksetup.bat needs to be done each new VS prompt session

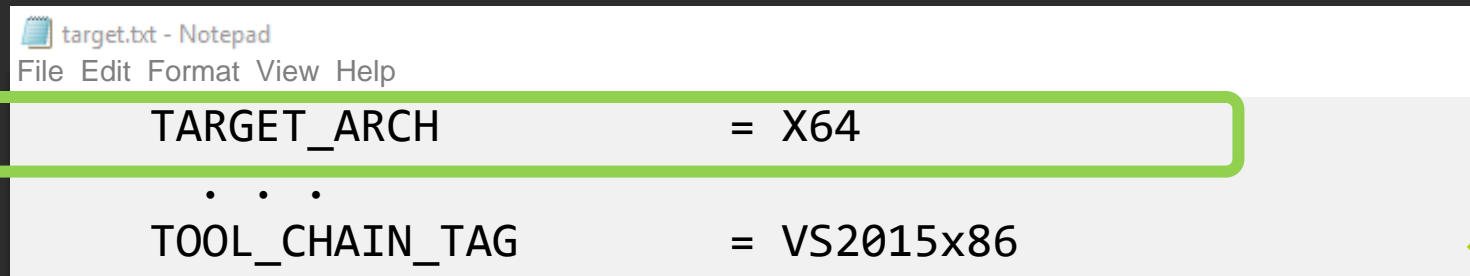
# Non Stuart CI Build EDK II

## EmulatorPkg – Non Stuart CI Build with edk2 -Update Target.txt

Invoke Edksetup.bat

```
$> cd C:\FW\edk2-ws\edk2
$> edksetup.bat
```

Edit the file Conf/target.txt (*change* TOOL\_CHAIN\_TAG)  
notepad Conf/target.txt



VS version	TOOL_CHAIN_TAG
2015	VS2015x86
2017	VS2017
2019	VS2019

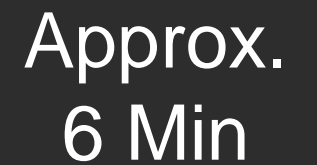
Save and Exit

Build EmulatorPkg

```
$> build -D ADD_SHELL_STRING -a X64
```

# Possible Build Errors

1. If you get a BUILD Error: Error “C:/Program “ not found
  - First check that you have opened Visual Studio and installed the “C++”
  - Open Visual Studio and create a “C++” project
  - (This will take some time to install)
2. If you get a BUILD Error: Check if RC.Exe compiler not found is the error -[here](#)
3. If you get a BUILD Error: fatal error C1041: cannot open program database ... Check [here](#)



## Finished build





# RUN THE EMULATOR

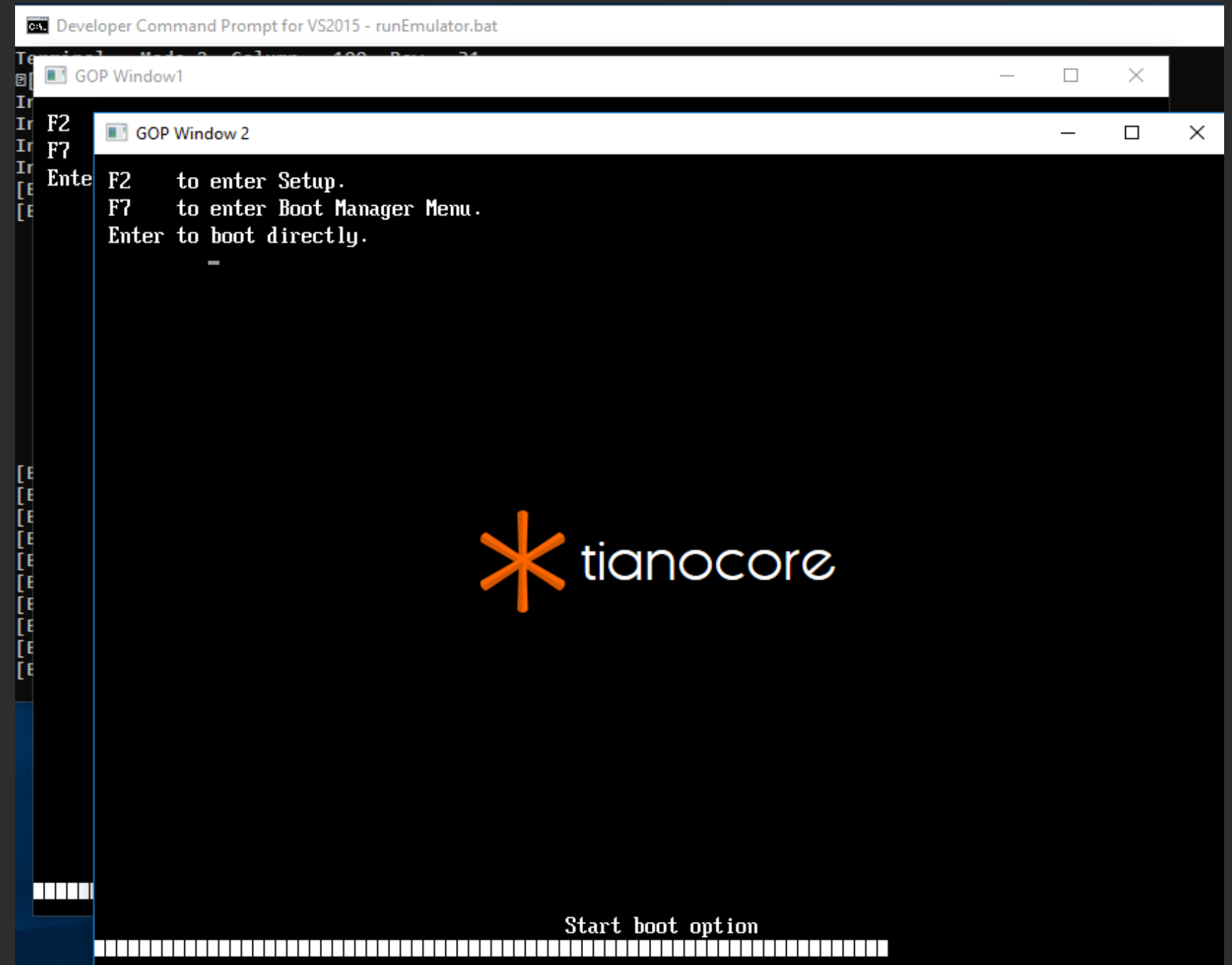
# Invoke Emulation

From the command prompt

```
$> RunEmulator.bat
```

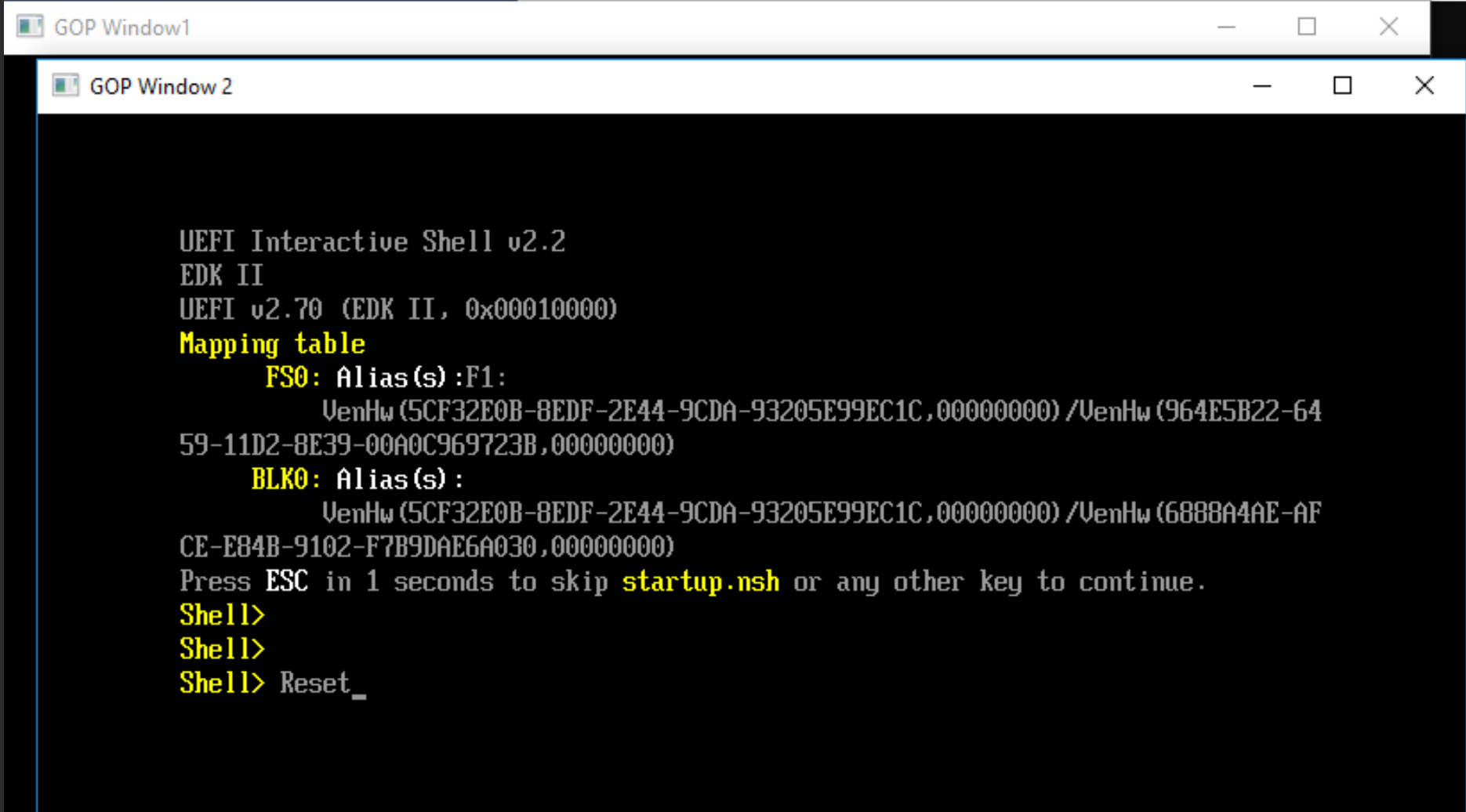
Or  
run **WinHost.exe** from:  
Build/ . . ./X64 directory

**Notice 2** “GOP Window n” opened



# Emulator at Shell Prompt

The Emulator will be running on top of Windows with the UEFI Shell prompt available.



The screenshot shows a window titled 'GOP Window 2' containing the UEFI Interactive Shell v2.2. The text displayed is as follows:

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s) :F1:
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (964E5B22-64
59-11D2-8E39-00A0C969723B,000000000)
  BLK0: Alias(s) :
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (6888A4AE-AF
CE-E84B-9102-F7B9DAE6A030,000000000)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell>
Shell>
Shell> Reset_
```

# Show the UEFI Boot Variables

At the Shell Prompt:

Shell> FS0:

FS0:> BCFG Boot Dump

```
Desc      - UEFI BootManagerMenuApp
DevPath   - Fv (6D99E806-3D38-42C2-A095-5F4300BFD7DC) /FuFile (EEC25BDC-67F2-4D95-B
1D5-F81B2039D11D)
Optional- N
Option: 02. Variable: Boot0002
Desc      - UEFI Misc Device
DevPath   - VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (6888A4AE-
AFCE-E84B-9102-F7B9DAE6A030,000000000)
Optional- Y
Option: 03. Variable: Boot0003
Desc      - UEFI Non-Block Boot Device
DevPath   - VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (964E5B22-
6459-11D2-8E39-00A0C969723B,000000000)
Optional- Y
Option: 04. Variable: Boot0004
Desc      - UEFI BootManagerMenuApp
DevPath   - Fv (6D99E806-3D38-42C2-A095-5F4300BFD7DC) /FuFile (EEC25BDC-67F2-4D95-B
1D5-F81B2039D11D) /BootManagerMenuApp
Optional- Y
Option: 05. Variable: Boot0000
Desc      - UEFI Enter Setup
DevPath   - Fv (6D99E806-3D38-42C2-A095-5F4300BFD7DC) /FuFile (462CAA21-7614-4503-B
36E-8AB6F4662331) /Enter Setup
Optional- N
FS0:\> _
```

# Use the Dmpstore to Show the Boot Order

At the Shell Prompt:

**FS0:>** Dmpstore BootOrder

```
FS0:\> dmpstore bootorder
Variable NU+RT+BS 'EFIGlobalVariable:BootOrder' DataSize = 0x0C
  00000000: 05 00 01 00 02 00 03 00-04 00 00 00          *.....*
FS0:\> _
```

# Use the BCFG to Move a boot item

Use BCFG to Move the 5th boot item  
too 1st location.

Then verify using the “dmpstore”

(Hint: use BCFG -? -b for help menu)

The dmpstore output should look like  
the screen shot



## Result

```
FS0:\> dmpstore bootorder
Variable NU+RT+BS 'EFIGlobalVariable:BootOrder' DataSize = 0x0C
00000000: 00 00 05 00 01 00 02 00-03 00 04 00 *.....
```

# Use the BCFG to Add a boot item

From Windows File explorer, Copy the file from the

%WORKSPACE%\edk2\ShellPkg\OldShell\Shell\_FullX64.efi to the directory  
%WORKSPACE%\Build\ . . .\X64

Use BCFG to Add a 06 entry for a new boot option with Shell\_FullX64.efi

Then verify using the “BCFG Boot Dump”

Hint: make sure Shell\_FullX64.efi is in the FS0: directory by doing:

FS0:\> Dir

After the bcfg add, The output should look like



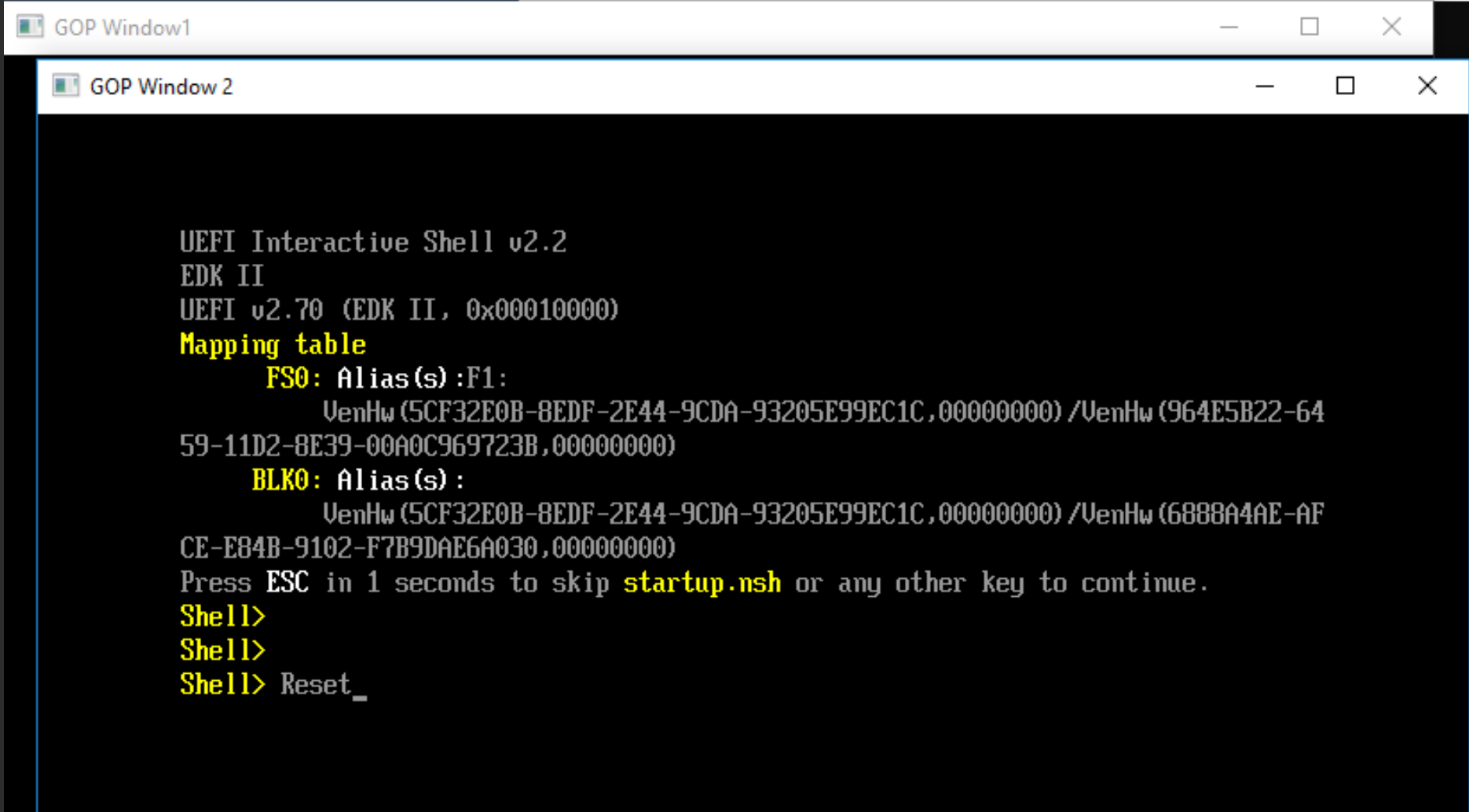
```
FS0:\> dir shell*.efi
Directory of: FS0:\
09/28/2021  12:32                1,425,408  Shell.efi
08/26/2021  15:33                771,136  Shell_FullX64.efi
```

Optional- Y	Result
Option: 06. Variable: Boot0006	
Desc - Olde EFI Shell 1.0	
DevPath - VenHw(5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000)/VenHw(6459-11D2-8E39-00A0C969723B,00000000)/\Shell_FullX64.efi	
Optional- N	
FS0:\> _	



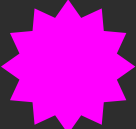


# Emulator at Shell Prompt

Type : “Reset” to exit



```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s) :F1:
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (964E5B22-64
59-11D2-8E39-00A0C969723B,000000000)
  BLK0: Alias(s) :
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,000000000) /VenHw (6888A4AE-AF
CE-E84B-9102-F7B9DAE6A030,000000000)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell>
Shell>
Shell> Reset_
```

# SUMMARY

-  Pin Visual Studio Command Prompt to Windows Task Bar
-  Build a EDK II Platform using Emulator package
-  Run the Emulator in Windows

# Questions?



# Return to Main Training Page



Return to Training Table of contents for next presentation [link](#)





# ACKNOWLEDGEMENTS

Redistribution and use in source (original document form) and 'compiled' forms (converted to PDF, epub, HTML and other formats) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (original document form) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, epub, HTML and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY TIANOCORE PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TIANOCORE PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2021-2022, Intel Corporation. All rights reserved.

**BACK UP**