

UEFI & EDK II Training

EDK II Debugging with Windows Lab – Simics® QSP

tianocore.org

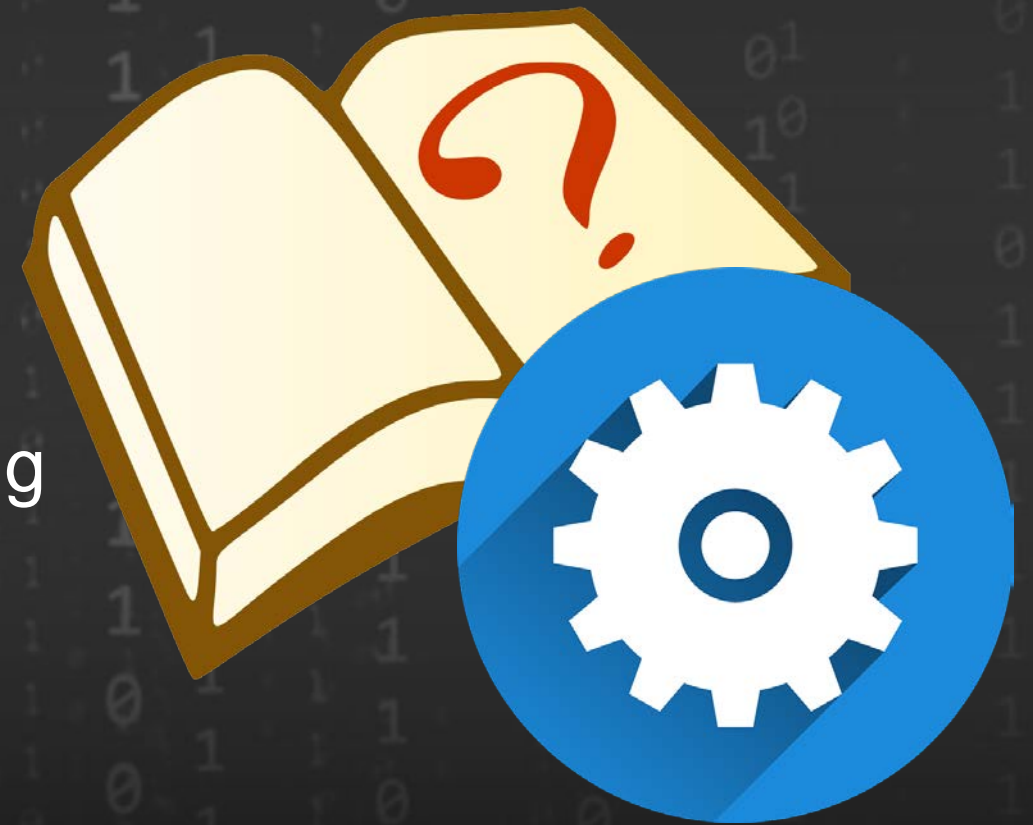
Copy and Paste [LabGuide.md](#)

Lesson Objective

- ✿ Using PCDs to Configure DebugLib – LAB 1 & 2
- ✿ Change the DebugLib instance to modify the debug output – LAB 3 & 4
- ✿ Debug EDK II Boot Flow – LAB 5

Catch up Lab 0

In this lab, you'll start where the previous Writing UEFI Applications left off.



Lab 0: Catch up from previous lab (1)

Skip the Catch up Lab if Lab Writing UEFI App Lab completed ([UEFI App Lab Guide](#))

- **Perform** Lab Setup from previous Labs ([Lab Guide](#))
- **Copy** contents of C:.. /FW/LabSampleCode/SampleAppDebug/, directory “MyPkg”, to C:/FW/edk2-ws/edk2/
- **Open**
edk2-platforms/Platform/Intel/SimicsOpenBoardPkg/BoardX58Ich10/OpenBoardPkg.dsc
- **Add** the following in the [Components . . .] section, Hint: add after comment:

```
# Add new modules here  
MyPkg/SampleApp/SampleApp.inf
```
- **Save** and close the file OpenBoard.dsc

Lab 0: Catch up from previous lab (2)

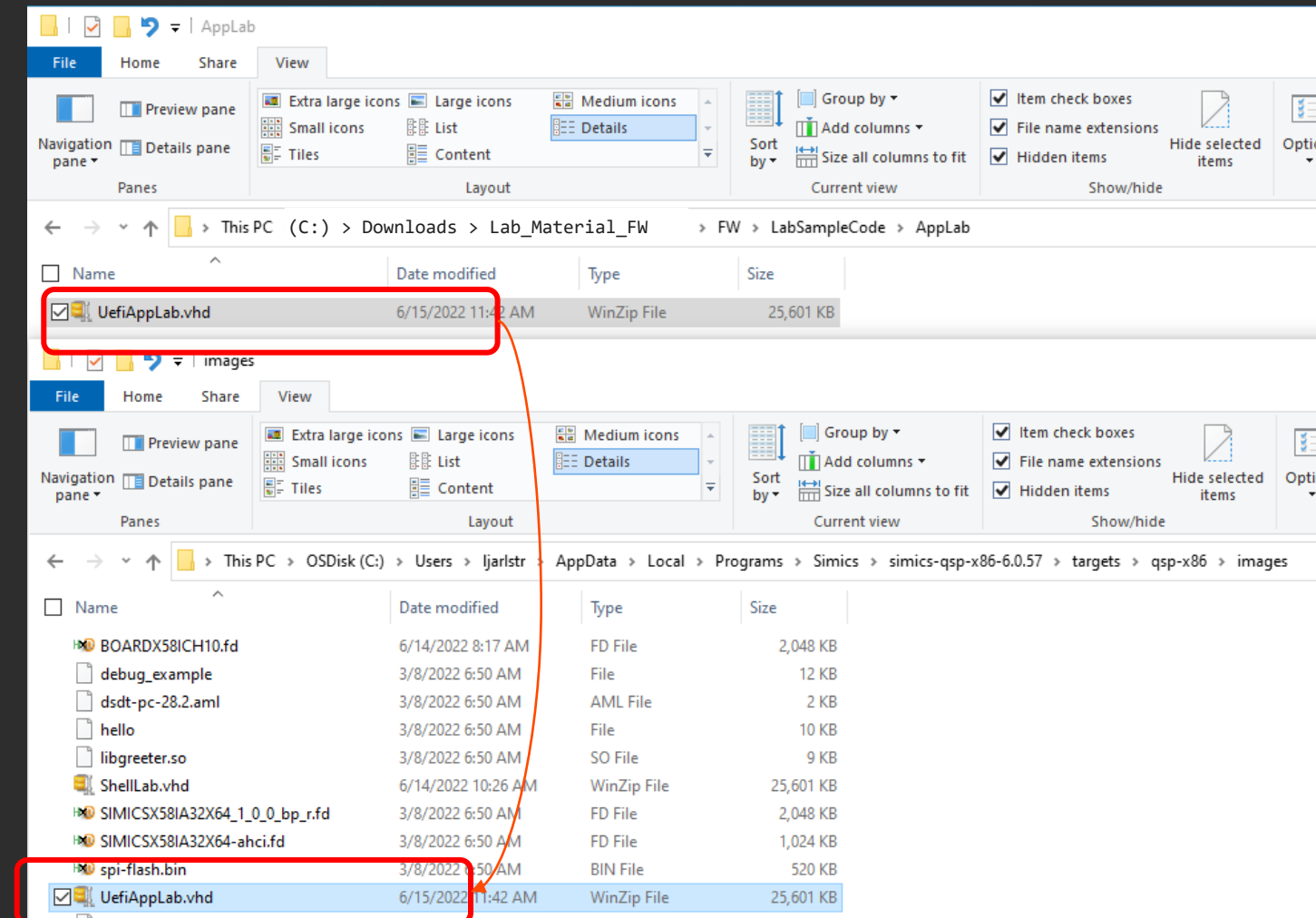
Copy the UefiAppLab.vhd

From:

.../Lab_Material_FW/FW/LabSampleCode/
AppLab/UefiAppLab.vhd

to

%USERPROFILE%\AppData\Local\Pr
ograms\Simics\simics-qsp-x86-
6.0.57\targets\qsp-x86\images



Lab 0: Catch up from previous lab (3)

Update the Simics Script to Use the UefiAppLab.vhd image as a file system

Edit the file: qsp-modern-core.simics from

%USERPROFILE%

\AppData\Local\Programs\Simics\simics-qsp-cpu-6.0.4\targets\qsp-x86\qsp-modern-core.simics

Add the following Line:

\$disk1_image="%simics%/targets/qsp-x86/images/UefiAppLab.vhd"

Before the “run-command-file” line

Save qsp-modern-core.simics

File: qsp-modern-core.simics

```
Decl{
decl {
! Script that runs the Quick Start Platform (QSP) with a modern
!   processor core.

params from "%simics%/targets/qsp-x86/qsp-clear-linux.simics"
default cpu_comp_class = "x86QSP2"
default num_cores = 2
default num_threads = 2
}
$disk1_image="%simics%/targets/qsp-x86/images/UefiAppLab.vhd"

run-command-file "%simics%/targets/qsp-x86/qsp-clear-linux.simics"
```


Lab 1 – Adding Debug Statements

In this lab, you'll add debug statements to the previous lab's SampleApp UEFI Shell application



Lab 1: Add debug statements to SampleApp

1. Mount the UefiAppLab.vhd using Disk Manager: [How to Mount VHD](#)
2. Open C:/FW/edk2-ws/edk2/MyPkg/SampleApp/SampleApp.c
 - Add the following to the include statements at the top of the file after below the last “include” statement:

```
#include <Library/DebugLib.h>
```


Lab 1: Add debug statements to SampleApp

Locate the UefiMain function. Then copy and paste the following code after the “EFI_INPUT_KEY KEY;” statement: and before the first Print() statement as shown in the screen shot below:

[LabGuide.md](#) for Copy and paste

```
DEBUG ((0xffffffff, "/n/nUEFI Base Training DEBUG DEMO/n") );
DEBUG ((0xffffffff, "0xffffffff USING DEBUG ALL Mask Bits Set/n") );

DEBUG ((DEBUG_INIT,      " 0x%08x USING DEBUG DEBUG_INIT/n" , (UINTN)(DEBUG_INIT)) );
DEBUG ((DEBUG_WARN,      " 0x%08x USING DEBUG DEBUG_WARN/n", (UINTN)(DEBUG_WARN)) );
DEBUG ((DEBUG_LOAD,      " 0x%08x USING DEBUG DEBUG_LOAD/n", (UINTN)(DEBUG_LOAD)) );
DEBUG ((DEBUG_FS,        " 0x%08x USING DEBUG DEBUG_FS/n", (UINTN)(DEBUG_FS)) );
DEBUG ((DEBUG_POOL,      " 0x%08x USING DEBUG DEBUG_POOL/n", (UINTN)(DEBUG_POOL)) );
DEBUG ((DEBUG_PAGE,      " 0x%08x USING DEBUG DEBUG_PAGE/n", (UINTN)(DEBUG_PAGE)) );
DEBUG ((DEBUG_INFO,      " 0x%08x USING DEBUG DEBUG_INFO/n", (UINTN)(DEBUG_INFO)) );
DEBUG ((DEBUG_DISPATCH,  " 0x%08x USING DEBUG DEBUG_DISPATCH/n", (UINTN)(DEBUG_DISPATCH)));
DEBUG ((DEBUG_VARIABLE,  " 0x%08x USING DEBUG DEBUG_VARIABLE/n", (UINTN)(DEBUG_VARIABLE)));
DEBUG ((DEBUG_BM,        " 0x%08x USING DEBUG DEBUG_BM/n", (UINTN)(DEBUG_BM)) );
DEBUG ((DEBUG_BLKIO,     " 0x%08x USING DEBUG DEBUG_BLKIO/n", (UINTN)(DEBUG_BLKIO)) );
DEBUG ((DEBUG_NET,       " 0x%08x USING DEBUG DEBUG_NET/n", (UINTN)(DEBUG_NET)) );
DEBUG ((DEBUG_UNDI,      " 0x%08x USING DEBUG DEBUG_UNDI/n", (UINTN)(DEBUG_UNDI)) );
DEBUG ((DEBUG_LOADFILE,  " 0x%08x USING DEBUG DEBUG_LOADFILE/n", (UINTN)(DEBUG_LOADFILE)));
DEBUG ((DEBUG_EVENT,     " 0x%08x USING DEBUG DEBUG_EVENT/n", (UINTN)(DEBUG_EVENT)) );
DEBUG ((DEBUG_GCD,       " 0x%08x USING DEBUG DEBUG_GCD/n", (UINTN)(DEBUG_EVENT)) );
DEBUG ((DEBUG_CACHE,     " 0x%08x USING DEBUG DEBUG_CACHE/n", (UINTN)(DEBUG_CACHE)) );
DEBUG ((DEBUG_VERBOSE,   " 0x%08x USING DEBUG DEBUG_VERBOSE/n", (UINTN)(DEBUG_VERBOSE)) );
DEBUG ((DEBUG_ERROR,     " 0x%08x USING DEBUG DEBUG_ERROR/n", (UINTN)(DEBUG_ERROR)) );
```

SAVE and CLOSE SampleApp.c

Update UefiAppLab.vhd File

Build the Simics QSP Board

At the VS Command Prompt, Build BoardX58Ich10

```
$> cd C:\FW\edk2-ws\edk2-platforms\Platform\Intel\
$> python build_bios.py -p BoardX58Ich10 -t VS20XX
```

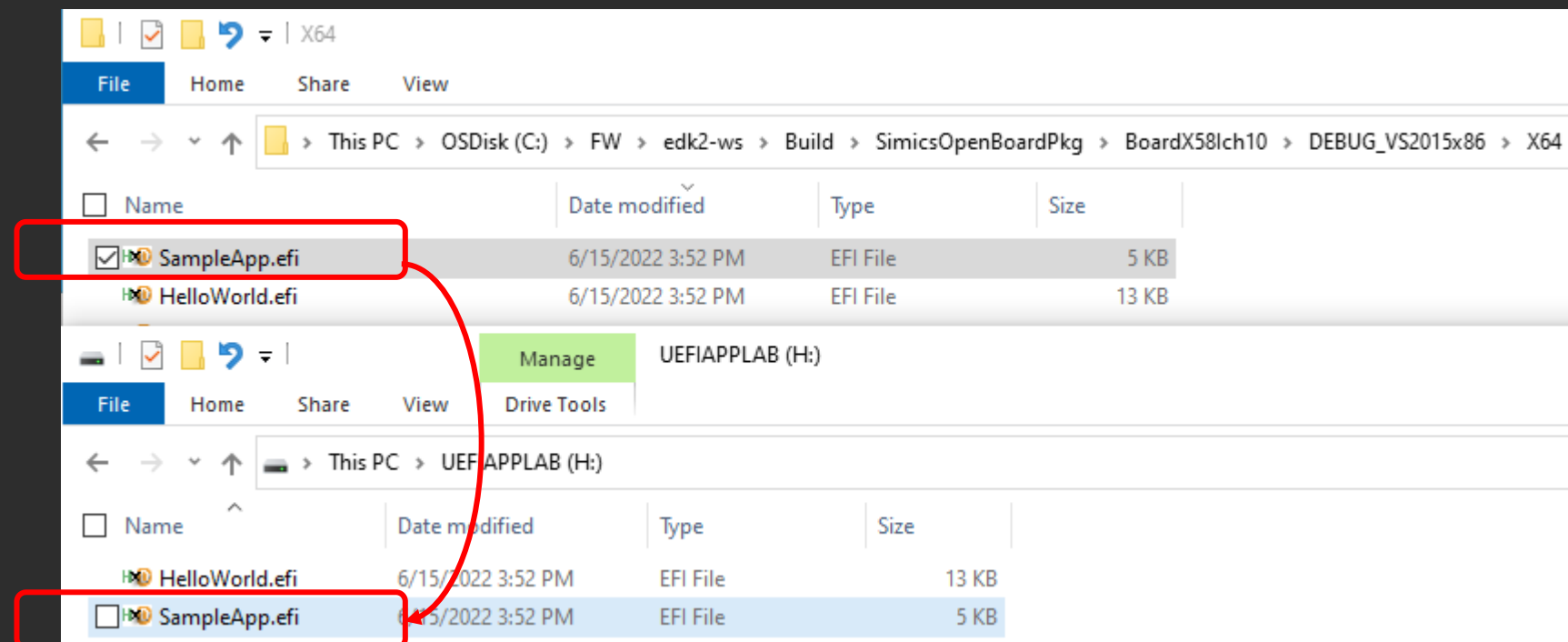
Copy

C:\FW\edk2-ws\Build\SimicsOpenBoardPkg\BoardX58Ich10\DEBUG_VS2015x86\X64\SampleApp.efi

To

X:\UEFIAPPLAB\

(where X is the VHD Drive)



Build Directory

VHD Disk

Lab 1: Build, Run and Test Result

Open another Windows Command Prompt

```
$> cd simics-projects\my-simics-project-1
```

Run the qsp-modern-core script :

```
$> .\simics targets/qsp-x86/qsp-modern-core.simics  
simics> run
```

(Press “F2” at the logo, then Select “Boot Manger” followed by “EFI Internal Shell”)

Invoke SampleApp at the Shell prompt

```
Shell> Fs1:  
FS1:\> SampleApp
```

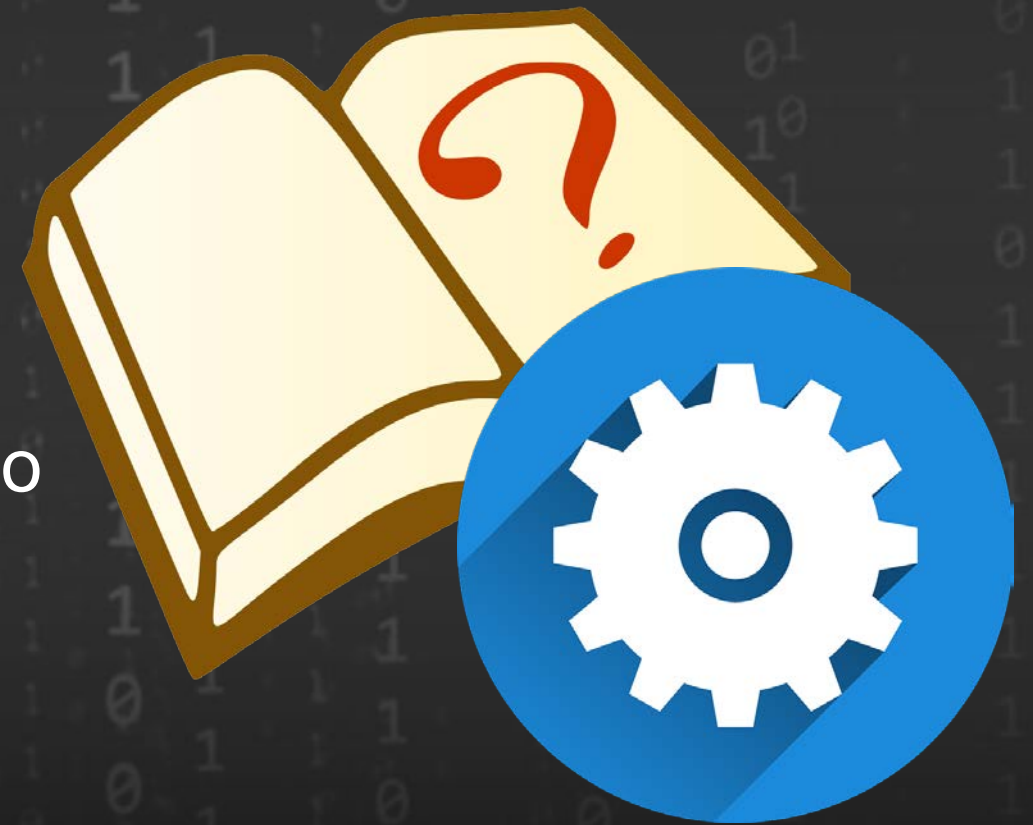
See that the output from the Debug statements goes to the Simics Serial Console

Exit Simics `simics> Stop` then `simics> quit`

```
-----  
Loading driver at 0x000DD279000 EntryPoint=0x000DD279344 SampleApp.efi  
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF DDF99818  
ProtectUefiImageCommon - 0xDDF901C0  
- 0x00000000DD279000 - 0x0000000000004020  
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA DF303818  
  
>>>>[UefiMain] Entry point: 0xDD279560 <<<<<<  
  
UEFI Base Training DEBUG DEMO  
0xffffffff USING DEBUG ALL Mask Bits Set  
0x00000001 USING DEBUG DEBUG_INIT  
0x00000002 USING DEBUG DEBUG_WARN  
0x00000004 USING DEBUG DEBUG_LOAD  
0x00000008 USING DEBUG DEBUG_FS  
0x00000040 USING DEBUG DEBUG_INFO  
0x80000000 USING DEBUG DEBUG_ERROR  
FSOpen: Open '\\' Success  
█
```

Lab 2 – Changing PCD Value

In this lab, you'll learn how to use PCD values to change debugging capabilities.



Lab 2: Change PCDs for SampleApp

Open

C:/FW/edk2-ws/edk2-platforms/Platform/Intel/SimicsOpenBoardPkg/BoardX58Ich10/OpenBoardPkg.dsc

Replace MyPkg/SampleApp/SampleApp.inf with the following:

```
MyPkg/SampleApp/SampleApp.inf {  
  <PcdsFixedAtBuild>  
    gEfiMdePkgTokenSpaceGuid.PcdDebugPropertyMask|0xff  
    gEfiMdePkgTokenSpaceGuid.PcdDebugPrintErrorLevel|0xffffffff  
}
```

Save and close OpenBoardPkg.dsc

[LabGuide.md](#) for Copy and paste

Lab 2 : Build and Test SampleApp

1. At the VS Command Prompt, Re-Build BoardX58Ich10

```
$> Cd C:\FW\edk2-ws\edk2-platforms\Platform\Intel\
$> python build_bios.py -p BoardX58Ich10 -t VS20XX
```

2. Copy SampleApp.efi from the build directory to the VHD Disk

Copy ..\Build\SimicsOpenBoardPkg\BoardX58Ich10\DEBUG_VS20XX\X64\SampleApp.efi UefiAppLab Disk

3. Run the qsp-modern-core script from Windows Command Prompt

```
$> .\simics targets/qsp-x86/qsp-modern-core.simics
simics> run
```

4. At the UEFI Shell prompt

```
Shell> Fs1:
FS1:\> SampleApp.efi
```

See that the output from ALL the Debug statements goes to the Simics Serial Console

5. Exit Simics simics> stop, simics> quit

```
board.mb.sb.com[0] - serial console
Edit View Settings

JEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set
0x00000001 USING DEBUG DEBUG_INIT
0x00000002 USING DEBUG DEBUG_WARN
0x00000004 USING DEBUG DEBUG_LOAD
0x00000008 USING DEBUG DEBUG_FS
0x00000010 USING DEBUG DEBUG_POOL
0x00000020 USING DEBUG DEBUG_PAGE
0x00000040 USING DEBUG DEBUG_INFO
0x00000080 USING DEBUG DEBUG_DISPATCH
0x00000100 USING DEBUG DEBUG_VARIABLE
0x00000400 USING DEBUG DEBUG_BM
0x00001000 USING DEBUG DEBUG_BLKIO
0x00004000 USING DEBUG DEBUG_NET
0x00010000 USING DEBUG DEBUG_UNDI
0x00020000 USING DEBUG DEBUG_LOADFILE
0x00080000 USING DEBUG DEBUG_EVENT
0x00080000 USING DEBUG DEBUG_GCD
0x00200000 USING DEBUG DEBUG_CACHE
0x00400000 USING DEBUG DEBUG_VERBOSE
0x80000000 USING DEBUG DEBUG_ERROR
FSOpen: Open '\' Success
```


Lab 3 – Library Instances for Debugging

In this lab, you'll learn how to add specific debug library instances.



Lab 3: Using Library Instances for Debugging

Open

C:/FW/edk2-platforms/Platform/Intel/SimicsOpenBoardPkg/BoardX58Ich10/OpenBoardPkg.dsc

Replace MyPkg/SampleApp/SampleApp.inf { . . . } with the following:

```
MyPkg/SampleApp/SampleApp.inf {  
    <LibraryClasses>  
        DebugLib|MdePkg/Library/UefiDebugLibConOut/UefiDebugLibConOut.inf  
}
```

Save and close OpenBoardPkg.dsc

Lab 3 : Build and Test SampleApp

1. At the VS Command Prompt, Re-Build BoardX58Ich10

```
$> Cd C:\FW\edk2-ws\edk2-platforms\Platform\Intel\
$> python build_bios.py -p BoardX58Ich10 -t VS20XX
```

2. Copy SampleApp.efi from the build directory to the VHD Disk

Copy ..\Build\SimicsOpenBoardPkg\BoardX58Ich10\DEBUG_VS20XX\X64\SampleApp.efi UefiAppLab Disk

3. Run the qsp-modern-core script from Windows Command Prompt :

```
$> .\simics targets/qsp-x86/qsp-modern-core.simics
simics> run
```

4. At the UEFI Shell prompt

```
Shell> Fs1:
FS1:\> SampleApp.efi
```

See that the output from the Debug statements now goes to the Simics VGA console

5. Exit Simics simics> stop, simics> quit

board.mb.gpu.vga - graphics console

Edit View Settings

```
Shell> fs1:
```

```
FS1:\> SampleApp.efi
```

```
>>>>>[UefiMain] Entry point: 0xDD2F4670 <<<<<<
```

```
UEFI Base Training DEBUG DEMO
```

```
0xffffffff USING DEBUG ALL Mask Bits Set
```

```
0x00000001 USING DEBUG DEBUG_INIT
```

```
0x00000002 USING DEBUG DEBUG_WARN
```

```
0x00000004 USING DEBUG DEBUG_LOAD
```

```
0x00000008 USING DEBUG DEBUG_FS
```

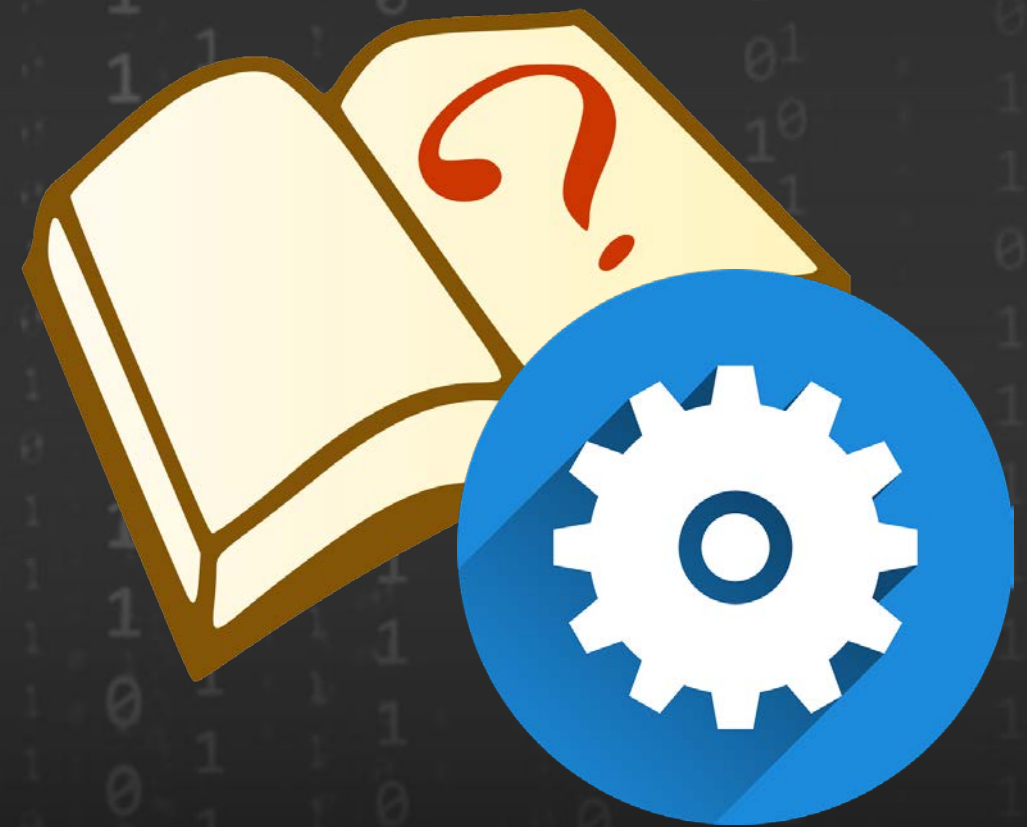
```
0x00000040 USING DEBUG DEBUG_INFO
```

```
0x80000000 USING DEBUG DEBUG_ERROR
```

```
System Table: 0xDEDED018
```

Lab 4: Null Instance of DebugLib

In this lab, you'll change the DebugLib to the Null instance.



Lab 4: Using Null Library Instances

Open

C:/FW/edk2-platforms/Platform/Intel/SimicsOpenBoardPkg/BoardX58Ich10/OpenBoardPkg.dsc

Replace MyPkg/SampleApp/SampleApp.inf { . . . } with the following:

```
MyPkg/SampleApp/SampleApp.inf {  
    <LibraryClasses>  
    DebugLib|MdePkg/Library/BaseDebugLibNull/BaseDebugLibNull.inf  
}
```

Save and close OpenBoardPkg.dsc

Lab 4 : Build and Test SampleApp

1. At the VS Command Prompt, Re-Build BoardX58Ich10

```
$> Cd C:\FW\edk2-ws\edk2-platforms\Platform\Intel\
$> python build_bios.py -p BoardX58Ich10 -t VS20XX
```

2. Copy SampleApp.efi from the build directory to the VHD Disk

Copy ..\Build\SimicsOpenBoardPkg\BoardX58Ich10\DEBUG_VS20XX\X64\SampleApp.efi UefiAppLab Disk

3. Run the qsp-modern-core script from Windows Command Prompt :

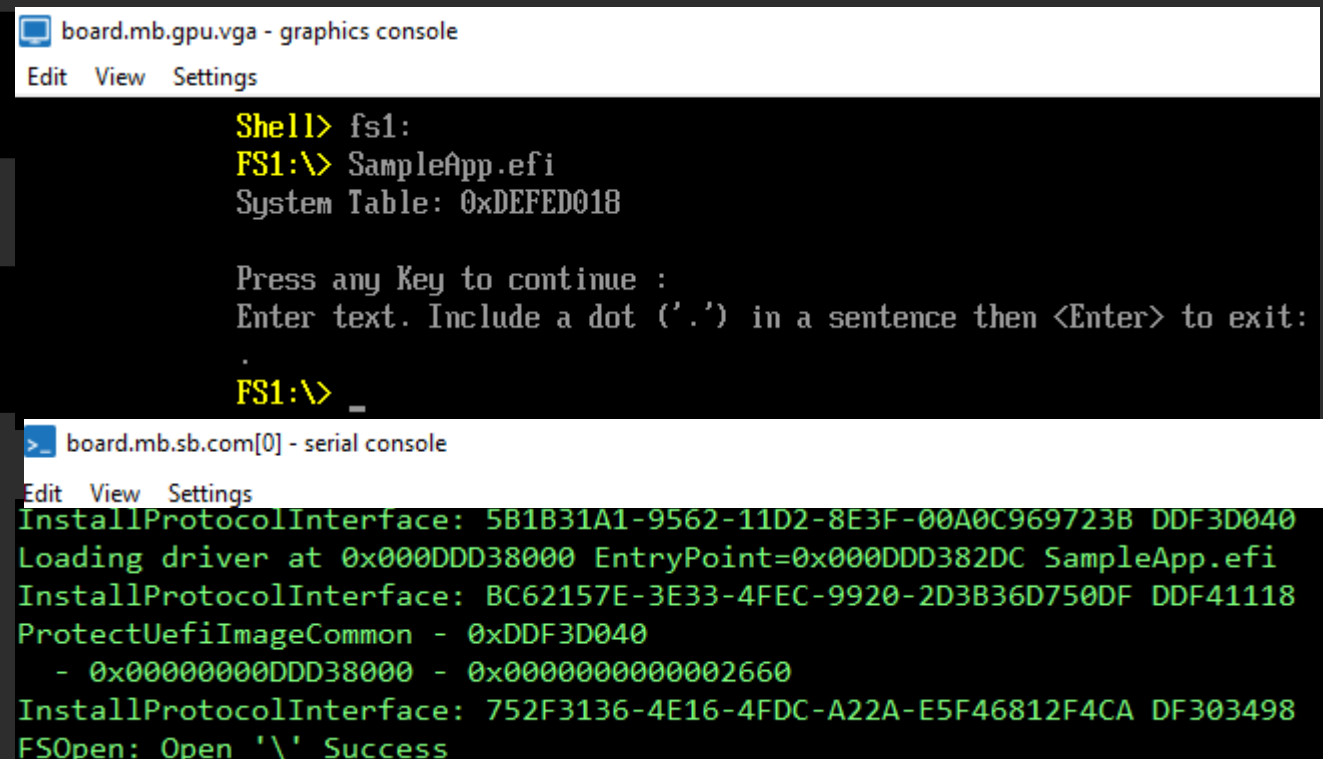
```
$> .\simics targets/qsp-x86/qsp-modern-core.simics
simics> run
```

4. At the UEFI Shell prompt

```
Shell> Fs1:
FS1:\> SampleApp.efi
```

See that there is **NO** Debug output

5. Exit Simics simics> stop, simics> quit

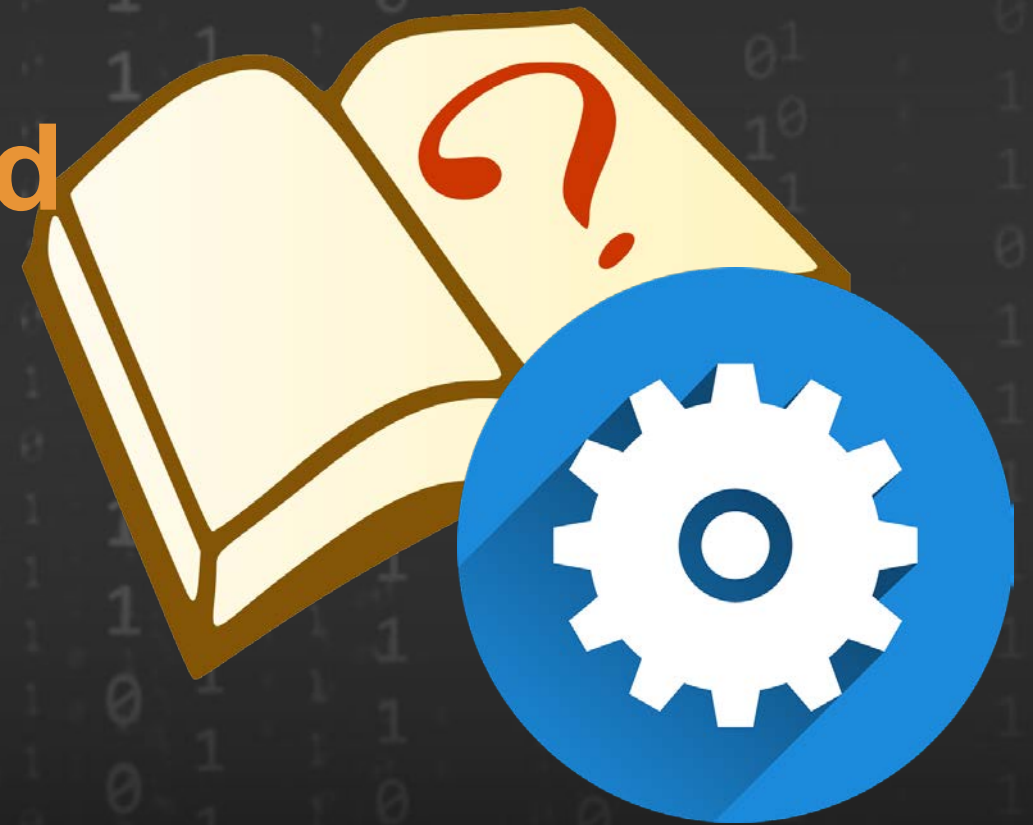


The first screenshot shows the UEFI Shell interface for 'board.mb.gpu.vga - graphics console'. It displays the prompt 'Shell> fs1:', followed by 'FS1:\> SampleApp.efi' and 'System Table: 0xDEFED018'. Below this, it says 'Press any Key to continue : Enter text. Include a dot ('.') in a sentence then <Enter> to exit:' and shows 'FS1:\> .'.

The second screenshot shows the serial console output for 'board.mb.sb.com[0] - serial console'. It displays the following text: 'InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B DDF3D040', 'Loading driver at 0x000DDD38000 EntryPoint=0x000DDD382DC SampleApp.efi', 'InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF DDF41118', 'ProtectUefiImageCommon - 0xDDF3D040', '- 0x00000000DDD38000 - 0x0000000000002660', 'InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA DF303498', and 'FSOpen: Open '\'' Success'.

Lab 5: Debugging EDK II add Debug to Boot Flow

In this lab, you'll learn how to add Debug statements to the EDK II Boot flow and check the debug log output



Lab 5: Debug Boot Flow

Edit the MdeModulePkg/Core/Pei/PeiMain/PeiMain.c

Add a “DEBUG” print ~line 489 before the call to the PeiDispatcher:

```
DEBUG((DEBUG_INFO, "***** ***** *****Before call to Pei Dispatcher ***** ***** *****\n"));
```

Save PeiMain.c

```
487 // Call PEIM dispatcher
488 //
489 DEBUG((DEBUG_INFO, "***** ***** *****Before call to Pei Dispatcher ***** ***** *****\n"));
490 PeiDispatcher (SecCoreData, &PrivateData);
491
```

Lab 5 : Build and Test the Boot Flow

1. At the VS Command Prompt, Re-Build BoardX58Ich10

```
$> Cd C:\FW\edk2-ws\edk2-platforms\Platform\Intel\
$> python build_bios.py -p BoardX58Ich10 -t VS20XX
```

2. Copy the Simics QSP Board .FD file

C:\fw\edk2-ws\Build\SimicsOpenBoardPkg\BoardX58Ich10\DEBUG_VS20XX\FV\BOARDX58ICH10.fd To
 %USERPROFILE%\AppData\Local\Programs\Simics\simics-qsp-x86-6.0.57\targets\qsp-x86\images

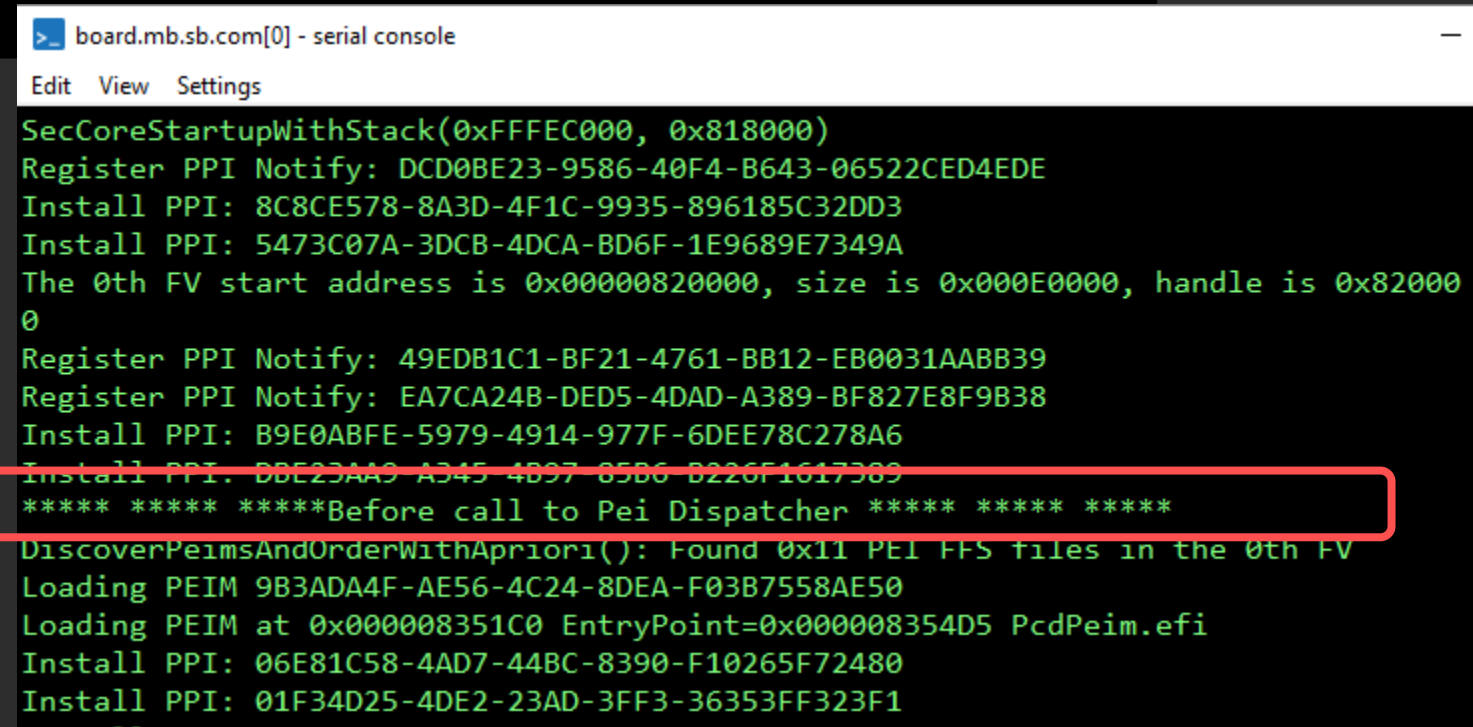
3. Run the qsp-modern-core script from Windows Command Prompt :

```
$> .\simics targets/qsp-x86/qsp-modern-core.simics
simics> run
```

4. Scroll back in the Simics Serial Console to find the Debug statement before the PEI Dispatcher. This would be a place to debug a PEIM with a Debugger and add a "CpuDeadLoop" or "CpuBreakPoint"

5. Exit Simics

```
simics> stop, simics> quit
```



```
board.mb.sb.com[0] - serial console
Edit View Settings
SecCoreStartupWithStack(0xFFFFEC000, 0x818000)
Register PPI Notify: DCD0BE23-9586-40F4-B643-06522CED4EDE
Install PPI: 8C8CE578-8A3D-4F1C-9935-896185C32DD3
Install PPI: 5473C07A-3DCB-4DCA-BD6F-1E9689E7349A
The 0th FV start address is 0x00000820000, size is 0x000E0000, handle is 0x820000
Register PPI Notify: 49EDB1C1-BF21-4761-BB12-EB0031AABB39
Register PPI Notify: EA7CA24B-DED5-4DAD-A389-BF827E8F9B38
Install PPI: B9E0ABFE-5979-4914-977F-6DEE78C278A6
Install PPI: DBE23AA9-A345-4B97-85B6-B226F1617389
***** ***** Before call to Pei Dispatcher ***** *****
DiscoverPeimsAndOrderWithApriori(): Found 0x11 PEI FFS files in the 0th FV
Loading PEIM 9B3ADA4F-AE56-4C24-8DEA-F03B7558AE50
Loading PEIM at 0x000008351C0 EntryPoint=0x000008354D5 PcdPeim.efi
Install PPI: 06E81C58-4AD7-44BC-8390-F10265F72480
Install PPI: 01F34D25-4DE2-23AD-3FF3-36353FF323F1
```

Summary

- ✿ Using PCDs to Configure DebugLib – LAB 1 & 2
- ✿ Change the DebugLib instance to modify the debug output – LAB 3 & 4
- ✿ Debug EDK II Boot flow- LAB 5

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) 2022, Intel Corporation. All rights reserved.