tianocore

UEFI & EDK II Training

EDK II Debugging with Windows Lab

tianocore.org

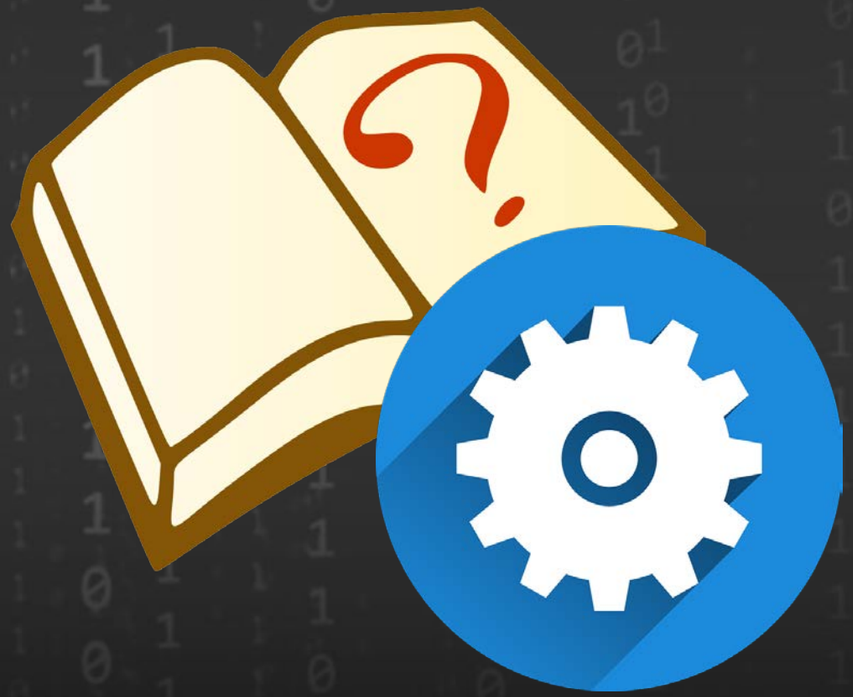
Copy and Paste LabGuide.md

LESSON OBJECTIVE

- Using PCDs to Configure `DebugLib` - LAB
- Change the `DebugLib` instance to modify the debug output - LAB
- Debug EDK II using VS Debugger - LAB

Lab 1 – Adding Debug Statements

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



Lab 1: Catch up from previous lab

Skip if Lab Writing UEFI App Lab completed (Lab Guide)

- Perform Lab Setup from previous Labs (Lab Guide)
- Create a Directory under the workspace `C:/FW/edk2-ws/edk2` “`SampleApp`”
- Copy contents of `C:../FW/LabSampleCode/SampleAppDebug` to `C:/FW/edk2-ws/edk2/SampleApp`
- Open `C:/FW/edk2/EmulatorPkg/EmulatorPkg.dsc`
- Add the following to the [Components] section:

```
# Add new modules here
 SampleApp/SampleApp.inf
```

- Save and close the file `EmulatorPkg.dsc`

Lab 1: Add debug statements to SampleApp

- Open a VS Command Prompt and type: `cd C:/FW/edk2-ws` then

```
C:/FW/edk2-ws > setenv.bat
C:/FW/edk2-ws > cd edk2
C:/FW/edk2-ws/edk2 > edksetup
```

- Open `C:/FW/edk2-ws/edk2/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>
```

LabGuide.md Slide for Copy and paste

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 Slide 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))  );
```

Lab 1: Build, Run and Test Result

At the VS Command Prompt

```
$> Build
$> RunEmulator.bat
```

Run the application from the shell

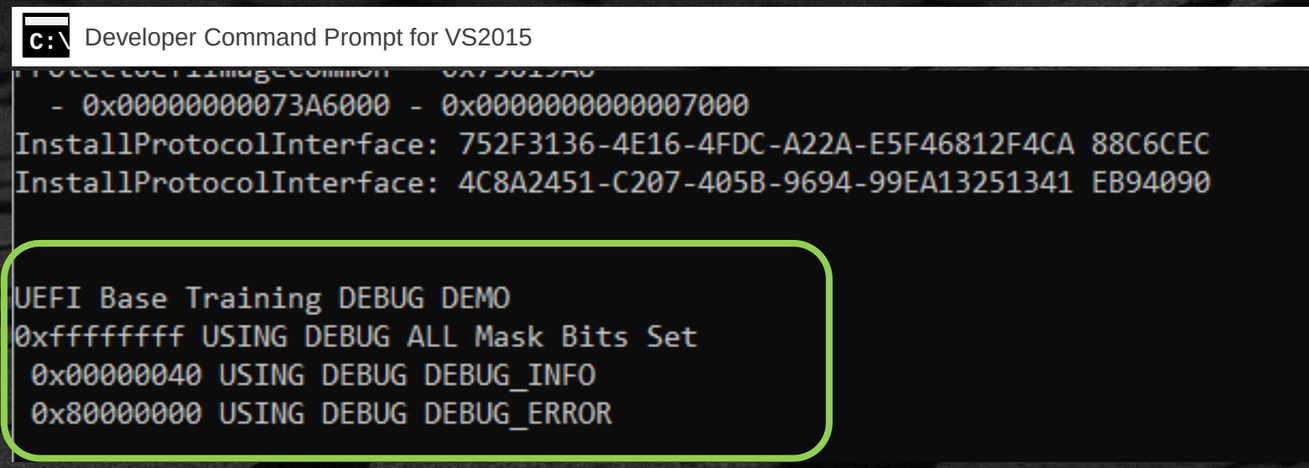
```
Shell> SampleApp
```

Check the VS Debug output

Exit

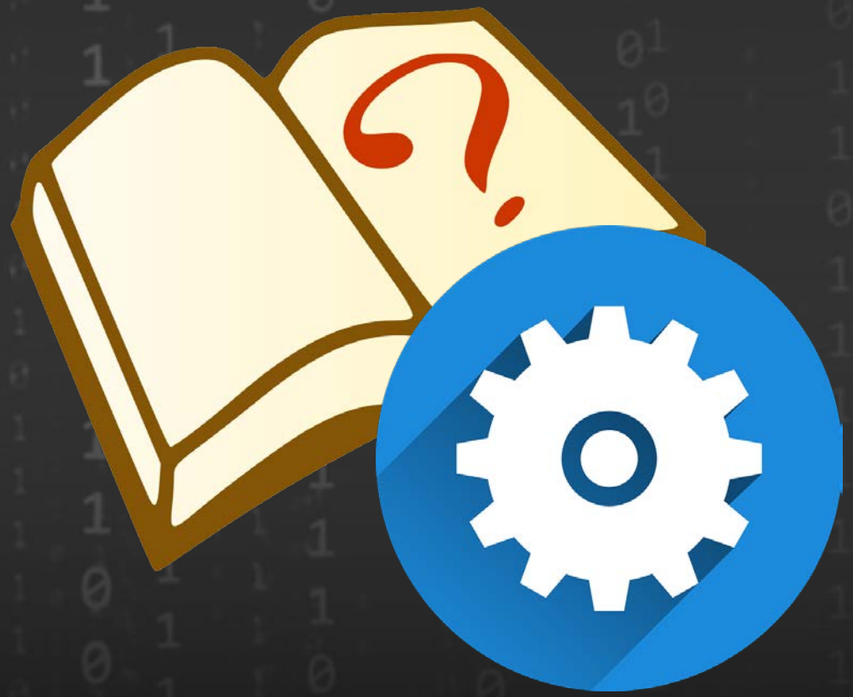
```
Shell> Reset
```

Visual Studio command prompt window output



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/EmulatorPkg/EmulatorPkg.dsc`
Replace `SampleApp/SampleApp.inf` with the following:

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

Save and **close** `EmulatorPkg.dsc`

LabGuide.md Slide for Copy and paste

Lab 2: Build, Run and Test Result

At the VS Command Prompt

```
$> Build
$> RunEmulator.bat
```

Run the application from the shell

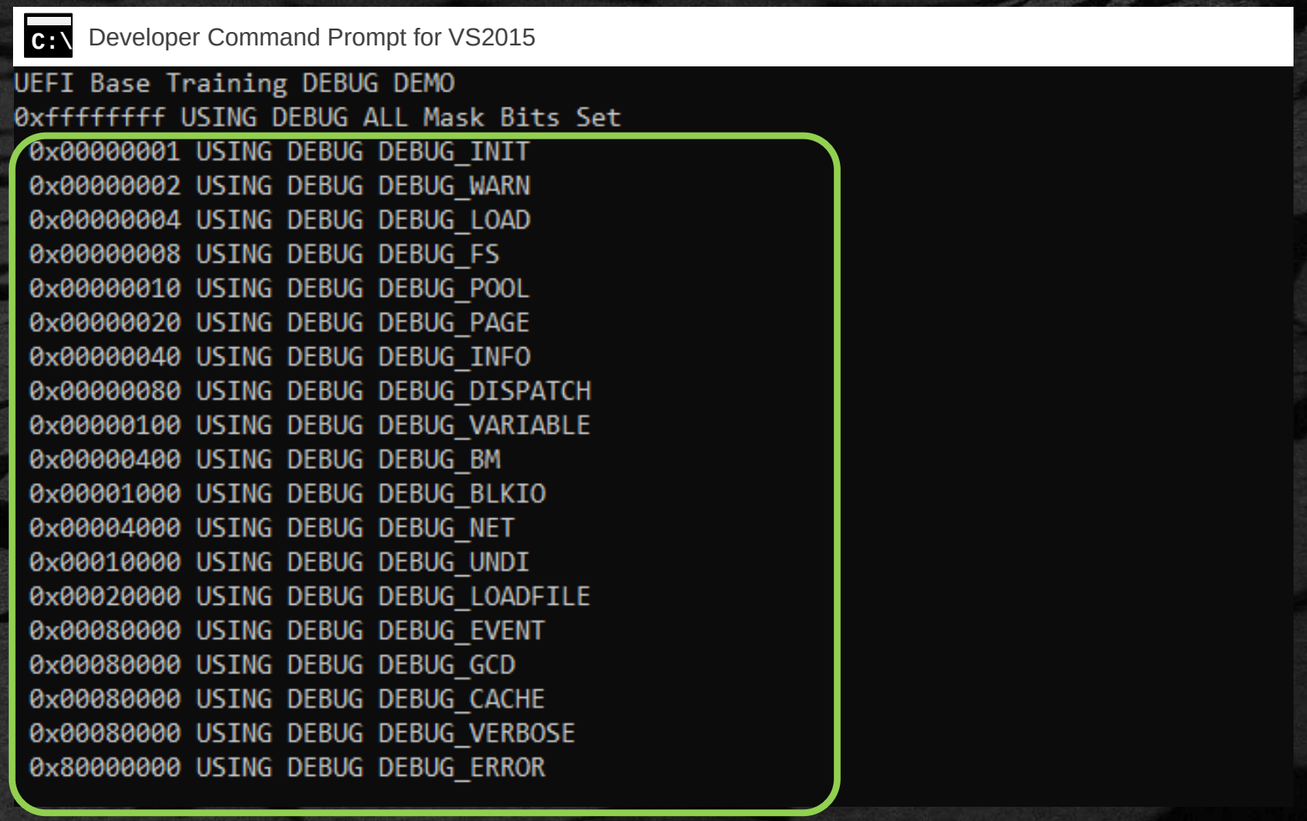
```
Shell> SampleApp
```

Check the VS Debug output

Exit

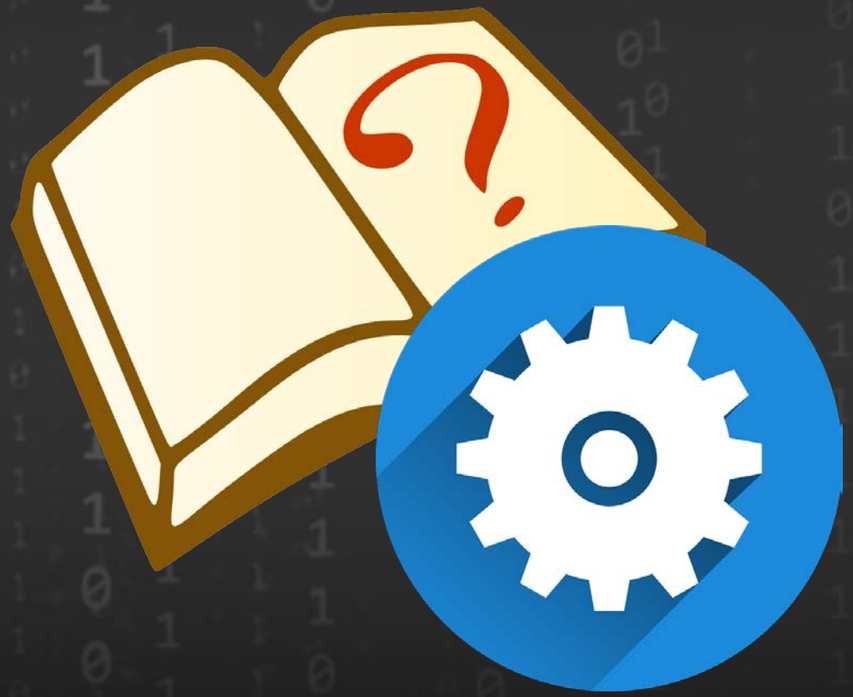
```
Shell> Reset
```

Visual Studio command prompt window output



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-ws/edk2/EmulatorPkg/EmulatorPkg.dsc`
Replace `SampleApp/SampleApp.inf { . . .}` with the following:

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

Save and **close** `EmulatorPkgPkg.dsc`

LabGuide.md Slide for Copy and paste

Lab 3: Build, Run and Test Result

At the VS Command Prompt

```
$> Build
$> RunEmulator.bat
```

Run the application from the shell

```
Shell> SampleApp
```

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

Exit

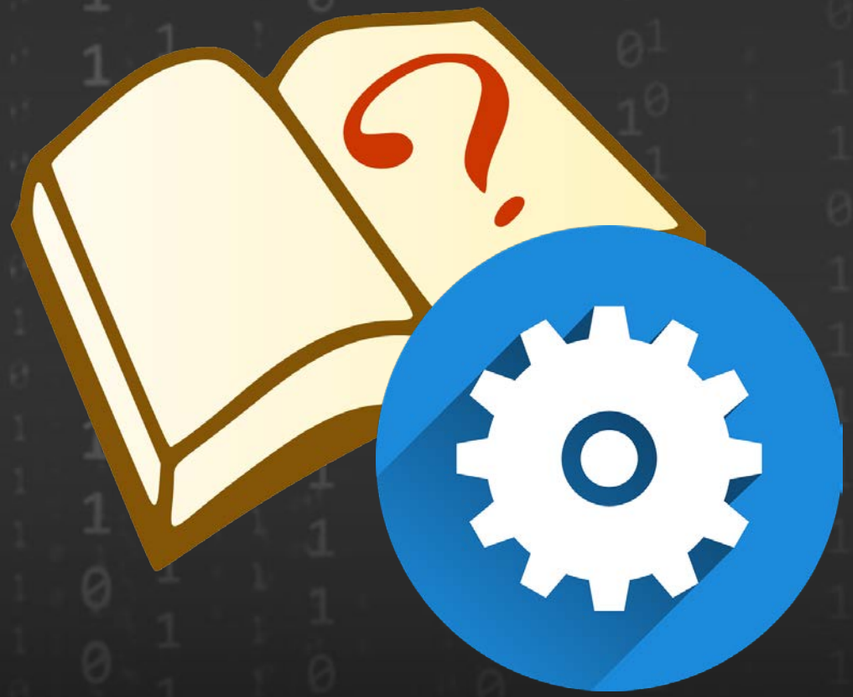
```
Shell> Reset
```

Debug output to console

```
Shell> sampleapp

                                                    EmulatorPkg

UEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set
 0x00000040 USING DEBUG DEBUG_INFO
 0x80000000 USING DEBUG DEBUG_ERROR
System Table: 0xB7A7C018

Press any Key to continue :
```

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-ws/edk2/EmulatorPkg/EmulatorPkg.dsc`
Replace `SampleApp/SampleApp.inf { . . .}` with the following:

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

Save and **close** `EmulatorPkg.dsc`

LabGuide.md Slide for Copy and paste

Lab 4: Build, Run and Test Result

At the VS Command Prompt

```
$> Build
$> RunEmulator.bat
```

Run the application from the shell

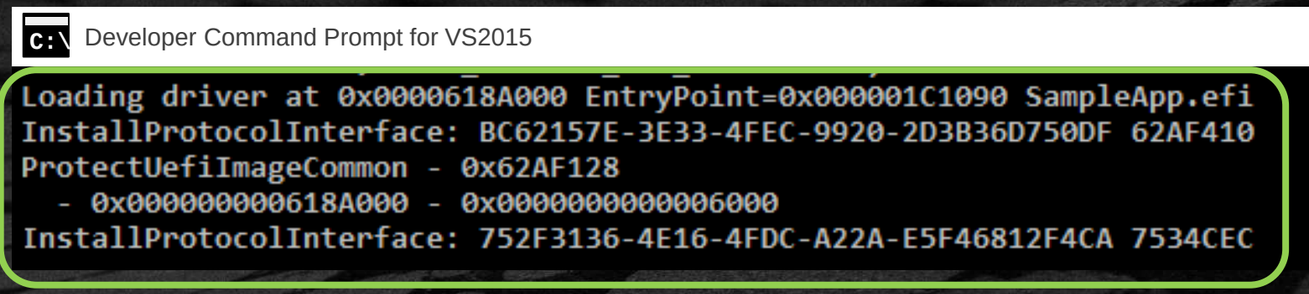
```
Shell> SampleApp
```

Check – now **NO** Debug output

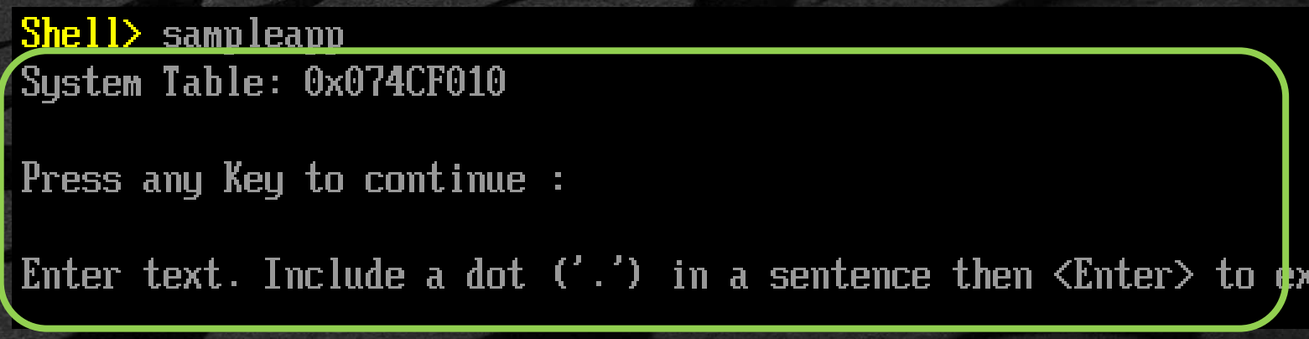
Exit

```
Shell> Reset
```

Visual Studio command prompt window output – NO DEBUG

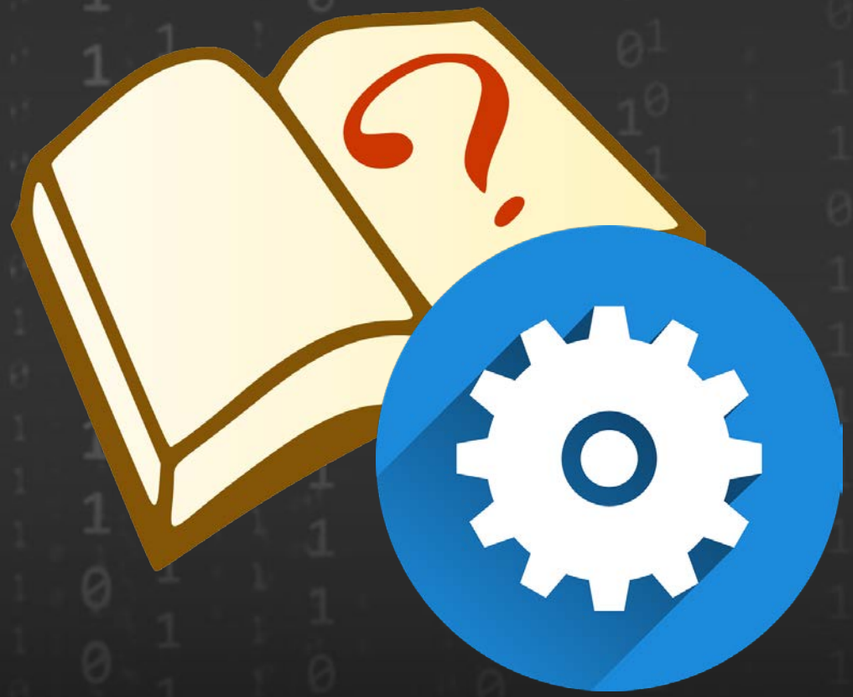


Console window – NO DEBUG



Lab 5: Debugging EDK II with VS Debugger

In this lab, you'll learn how setup the VS to debug the EDK II emulation



Lab 5: Debug with VS

Edit the SampleApp.c and add an “ASSERT_EFI_ERROR” :
Add the following:

```
EFI_STATUS      Status;
Status = EFI_NO_RESPONSE;
      . . .
ASSERT_EFI_ERROR(Status);
```

```
EFI_STATUS Status;
Status = EFI_NO_RESPONSE;  // or any EFI Error

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

ASSERT_EFI_ERROR(Status);
```

Save SampleApp.c

LabGuide.md Slide for Copy and paste

Lab 5: Debug with VS - ASSERT

At the VS Command Prompt

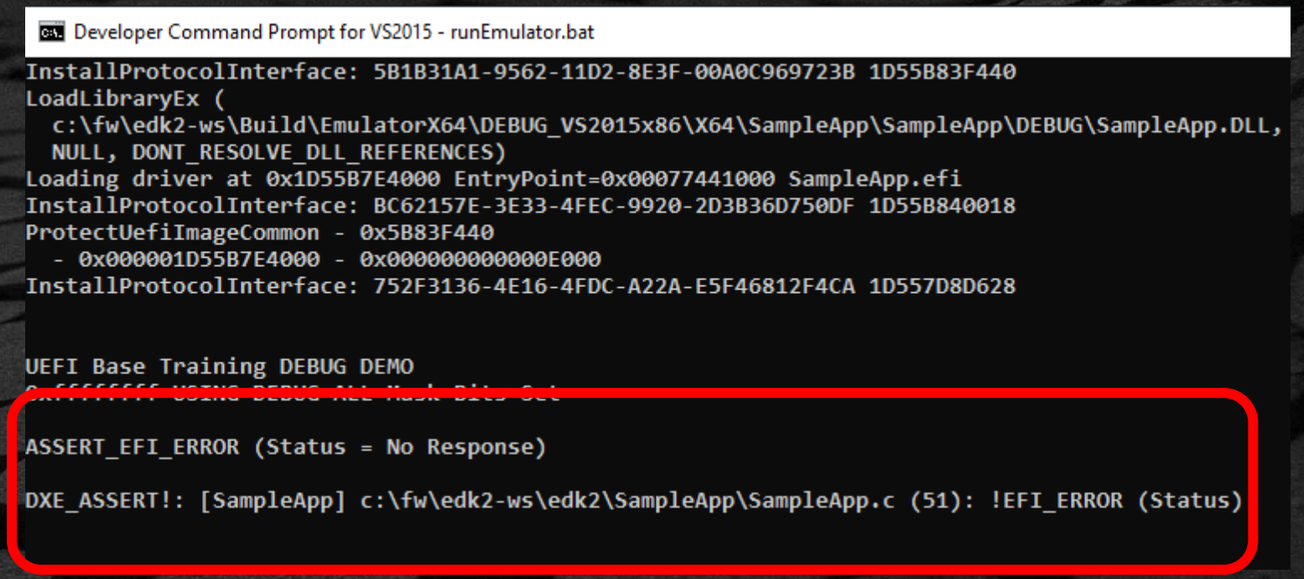
```
$> Build
$> RunEmulator.bat
```

Run the application from the shell

```
Shell> SampleApp
```

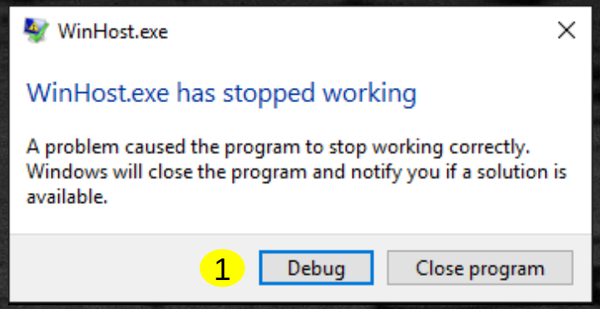
Assert in VS Command Prompt

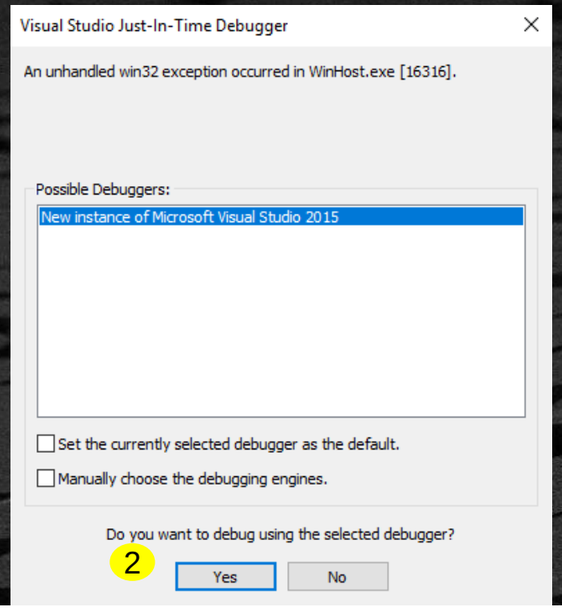
Visual Studio command prompt window output

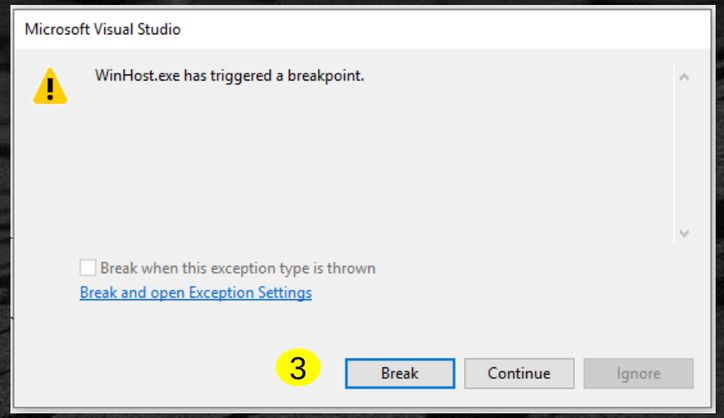


Lab 5: Debug with VS - ASSERT

Windows* VS Debugger Will Pop UP








```
24    **/
25    VOID
26    EFIAPI
27    CpuBreakpoint (
28      VOID
29      )
30    {
31      __debugbreak ();
32    }
33
```

4

“F10” - Step over

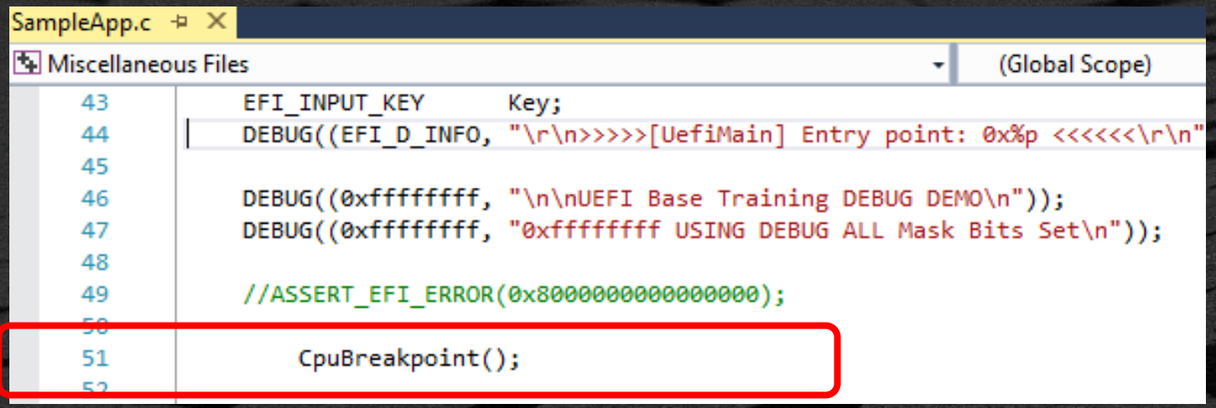
```
43      EFI_INPUT_KEY      Key;
44      EFI_STATUS      Status;
45      Status = EFI_NO_RESPONSE;  // or any other EFI Error
46
47    DEBUG ((0xffffffff, "\n\nUEFI Base Training DEBUG DEMO\n") );
48    DEBUG ((0xffffffff, "0xffffffff USING DEBUG ALL Mask Bits Set\r\n") );
49
50    ASSERT_EFI_ERROR(Status);  ≤4ms elapsed
51
```

SampleApp.c

Lab 5: Debug with VS - CpuBreakpoint

Edit the SampleApp.c and add "CpuBreakpoint();" Statement and comment out the "ASSERT":

```
CpuBreakpoint();
```



Save SampleApp.c

LabGuide.md Slide for Copy and paste

Lab 5: Debug with VS

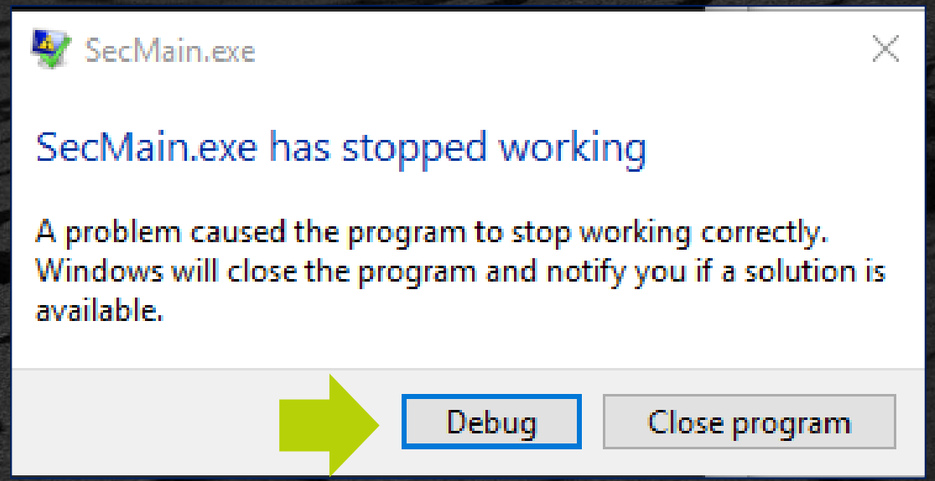
At the VS Command Prompt

```
$> Build
$> RunEmulator.bat
```

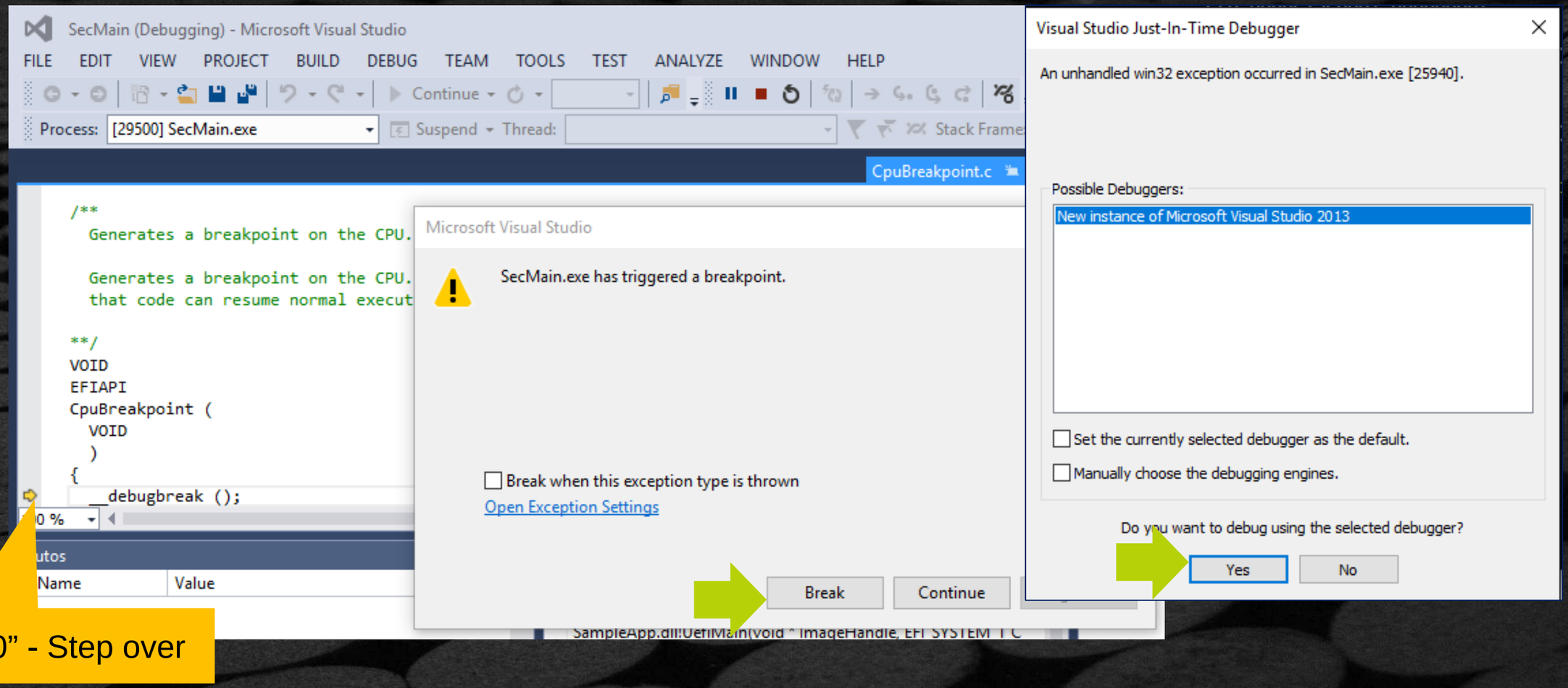
Run the application from the shell

```
Shell> SampleApp
```

VS option go to VS Debugger

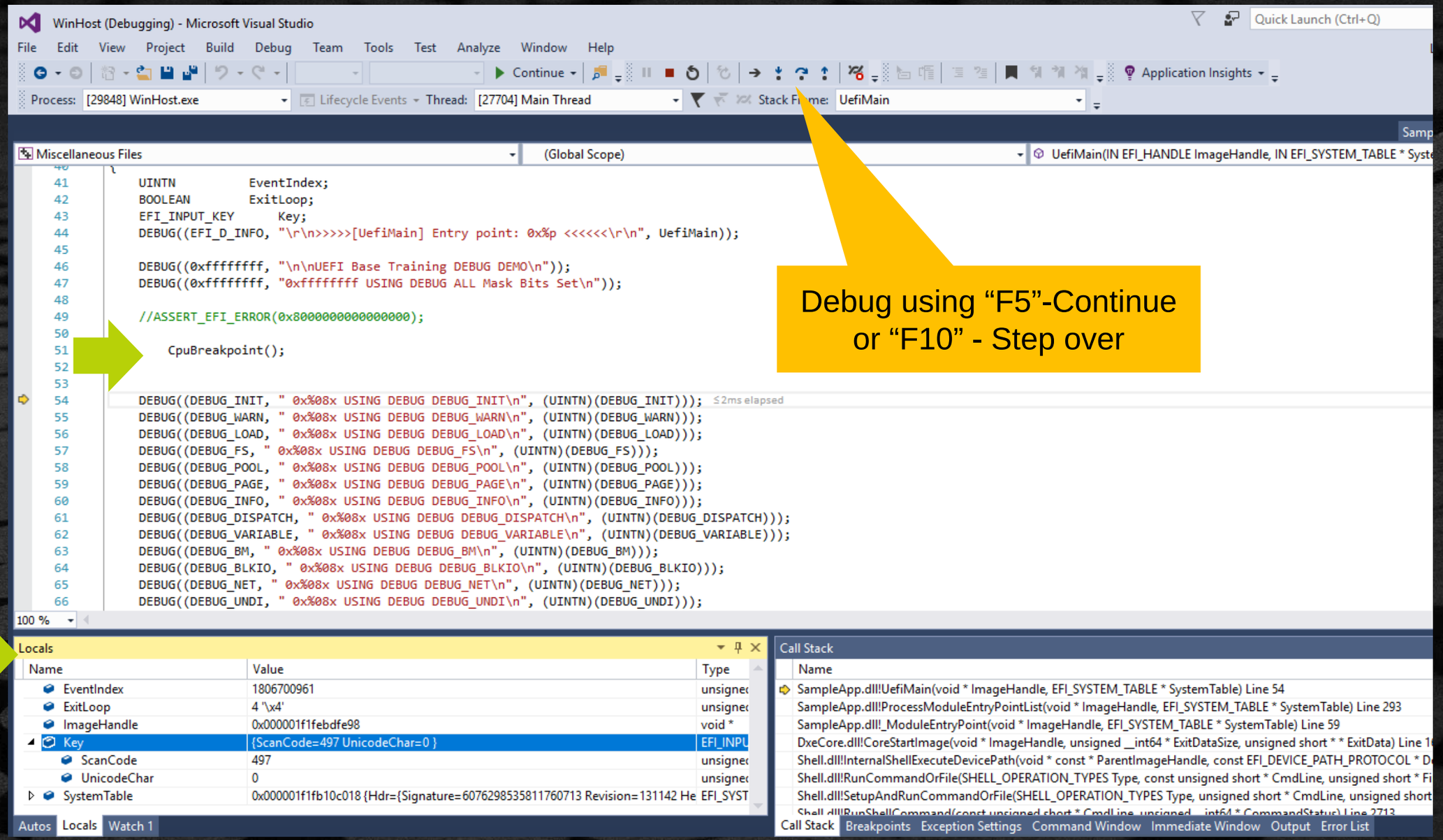


Invoke Windows Visual Studio Debugger



"F10" - Step over

Invoke Windows Visual Studio Debugger



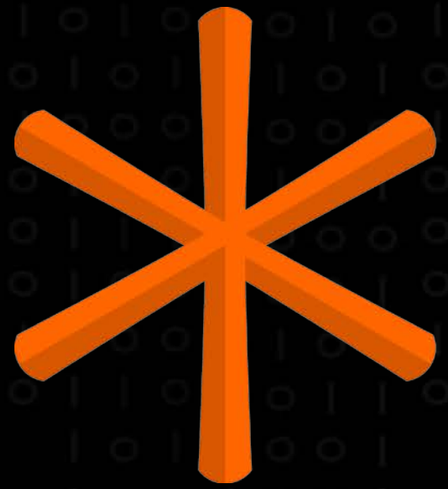
SUMMARY

- Using PCDs to Configure `DebugLib` - LAB
- Change the `DebugLib` instance to modify the debug output - LAB
- Debug EDK II using VS Debugger - LAB

Questions?

Return to Main Training Page

Return to Training Table of contents for next presentation link



tianocore

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) 2018, Intel Corporation. All rights reserved.

tianocore

BACK UP

ISSUE:

Debugging in Emulatior with Windows 7 and Visual Studio does not work?

Symptom: With Windows 7 a **`CpuBreakpoint()`** or **`ASSERT`** just exits with an error from the “Build Run” command.

Link to fix this issue:
https://github.com/tianocore/tianocore.github.io/wiki/NT32#Debugging_in_Nt32_Emulation_with_Windows_7_and_Visual_Studio_does_not_work

1. Run the RegEdt32
2. Navigate to the HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\AeDebug
3. Add a string value entry called "Auto" with a value of "1“

Windows 10 Visual Studio does not seem to have this issue