UEFI & EDK II Training

EDK II Debugging with Windows Lab

tianocore.org

Lesson EDK II Debugging

tianocore

LESSON OBJECTIVE

- Define **DebugLib** and its attributes
- List the ways to debug
- Using PCDs to Configure DebugLib - LAB
- Change Compiler & Linker Flags for debugging
- Change the **DebugLib** instance to modify the debug output - LAB
- Debug EDK II using VS Debugger - LAB

tianocore

DEBUGGING OVERVIEW

tianocore

Debug Methods

DEBUG and ASSERT macros in EDK II code

DEBUG instead of Print functions

Software/hardware debuggers

Shell commands to test capabilities for simple debugging

tianocore

EDK II `DebugLib` Library

`Debug` and **`Assert`** macros in code

Enable/disable when compiled (**`target.txt`**)

Connects a Host to capture debug messages

tianocore

DEBUGGING WITH PCDS

tianocore

Using PCDs to Configure DebugLib

MdePkg Debug Library Class

```
[PcdsFixedAtBuild. PcdsPatchableInModule]
      • • •
  gEfiMdePkgTokenSpaceGuid.PcdDebugPropertyMask|0x1f
  gEfiMdePkgTokenSpaceGuid.PcdDebugPrintErrorLevel|0x80000040
```

PcdDebugPropertyMask Values

Debugging *Features* Enabled

```
#define DEBUG_PROPERTY_DEBUG_ASSERT_ENABLED       0x01
#define DEBUG_PROPERTY_DEBUG_PRINT_ENABLED        0x02
#define DEBUG_PROPERTY_DEBUG_CODE_ENABLED         0x04
#define DEBUG_PROPERTY_CLEAR_MEMORY_ENABLED       0x08
#define DEBUG_PROPERTY_ASSERT_BREAKPOINT_ENABLED  0x10
#define DEBUG_PROPERTY_ASSERT_DEADLOOP_ENABLED    0x20
```

Default value in OvmfPkg is 0x2f

tianocore

PcdDebugPrintErrorLevel Values

Debug *Messages* Displayed

```
#define DEBUG_INIT       0x00000001   // Initialization
#define DEBUG_WARN       0x00000002   // Warnings
#define DEBUG_LOAD       0x00000004   // Load events
#define DEBUG_FS         0x00000008   // EFI File system
#define DEBUG_POOL       0x00000010   // Alloc & Free's  Pool
#define DEBUG_PAGE       0x00000020   // Alloc & Free's  Page
#define DEBUG_INFO       0x00000040   // Verbose
#define DEBUG_DISPATCH   0x00000080   // PEI/DXE Dispatchers
#define DEBUG_VARIABLE   0x00000100   //Variable
#define DEBUG_BM         0x00000400   // Boot Manager
#define DEBUG_BLKIO      0x00001000   // BlkIo Driver
#define DEBUG_NET        0x00004000   // SNI Driver
#define DEBUG_UNDI       0x00010000   // UNDI Driver
#define DEBUG_LOADFILE   0x00020000   // Load File
#define DEBUG_EVENT      0x00080000   // Event messages
#define DEBUG_ERROR      0x80000000   // Error
```

Aliases EFI_D_INIT == DEBUG_INIT, etc..

Default value in OvmfPkg is 0x8000004f

Changing PCD Values

Change all instances of a PCD in platform DSC

```
[PcdsFixedAtBuild.IA32]
gEfiMdePkgTokenSpaceGuid.PcdDebugPrintErrorLevel|0x00000000
```

Changing PCD Values

Change all instances of a PCD in platform DSC

```
[PcdsFixedAtBuild.IA32]
gEfiMdePkgTokenSpaceGuid.PcdDebugPrintErrorLevel|0x00000000
```

Change a single module’s PCD values in DSC

```
MyPath/MyModule.inf {
<PcdsFixedAtBuild>
gEfiMdePkgTokenSpaceGuid.PcdDebugPrintErrorLevel|0x80000000
}
```

tianocore

Other Debug Related Libraries

`ReportStatusCodeLib` –Progress codes

`gEfiMdePkgTokenSpaceGuid.PcdReportStatusCodePropertyMask`

`PostCodeLib` – Enable Post codes

`gEfiMdePkgTokenSpaceGuid.PcdPostCodePropertyMask`

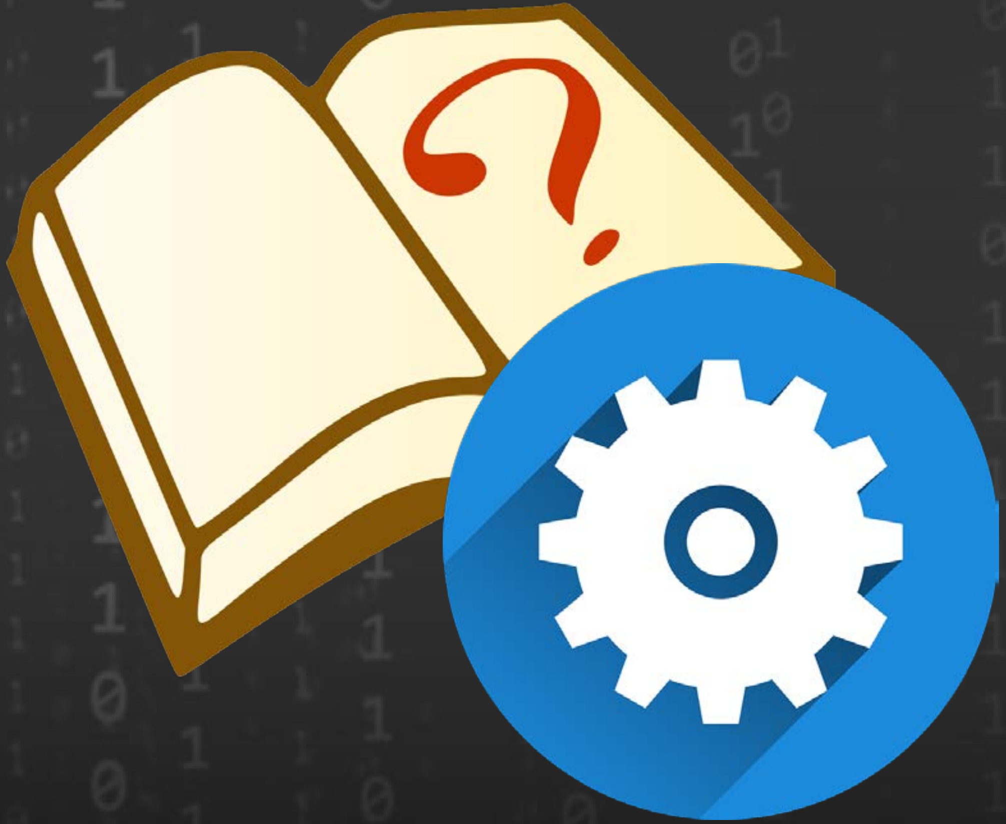
`PerformanceLib` – Enable Measurement

`gEfiMdePkgTokenSpaceGuid.PcdPerformanceLibraryPropertyMask`

tianocore

Lab 1 – Adding Debug Statements

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



tianocore

Lab 1: Catch up from previous lab

Skip if Lab Writing UEFI App Lab completed

- Perform Lab Setup from previous Labs
- Create a Directory under the workspace C:/FW/edk2 “SampleApp”
- Copy contents of C:../FW/LabSampleCode/SampleAppDebug to C:/FW/edk2/SampleApp
- Open C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc
- Add the following to the [Components] section:

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

- Save and close the file C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

tianocore

Lab 1: Add debug statements to SampleApp

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

```
C:/FW/edk2> edksetup
```

- Open C:/FW/edk2/SampleApp/SampleApp.c

- Add the following to the include statements at the top of the file after below the last “**`include`**” statement:

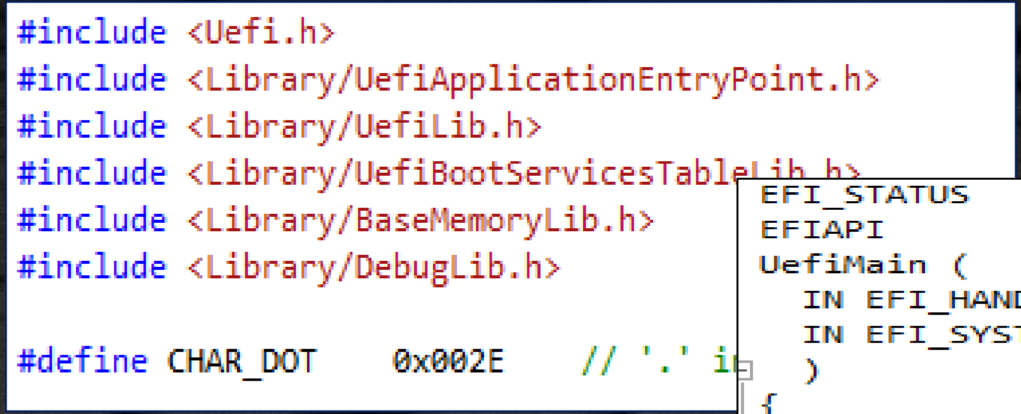
```
#include <Library/DebugLib.h>
```

Lab 1: Add debug statements to SampleApp

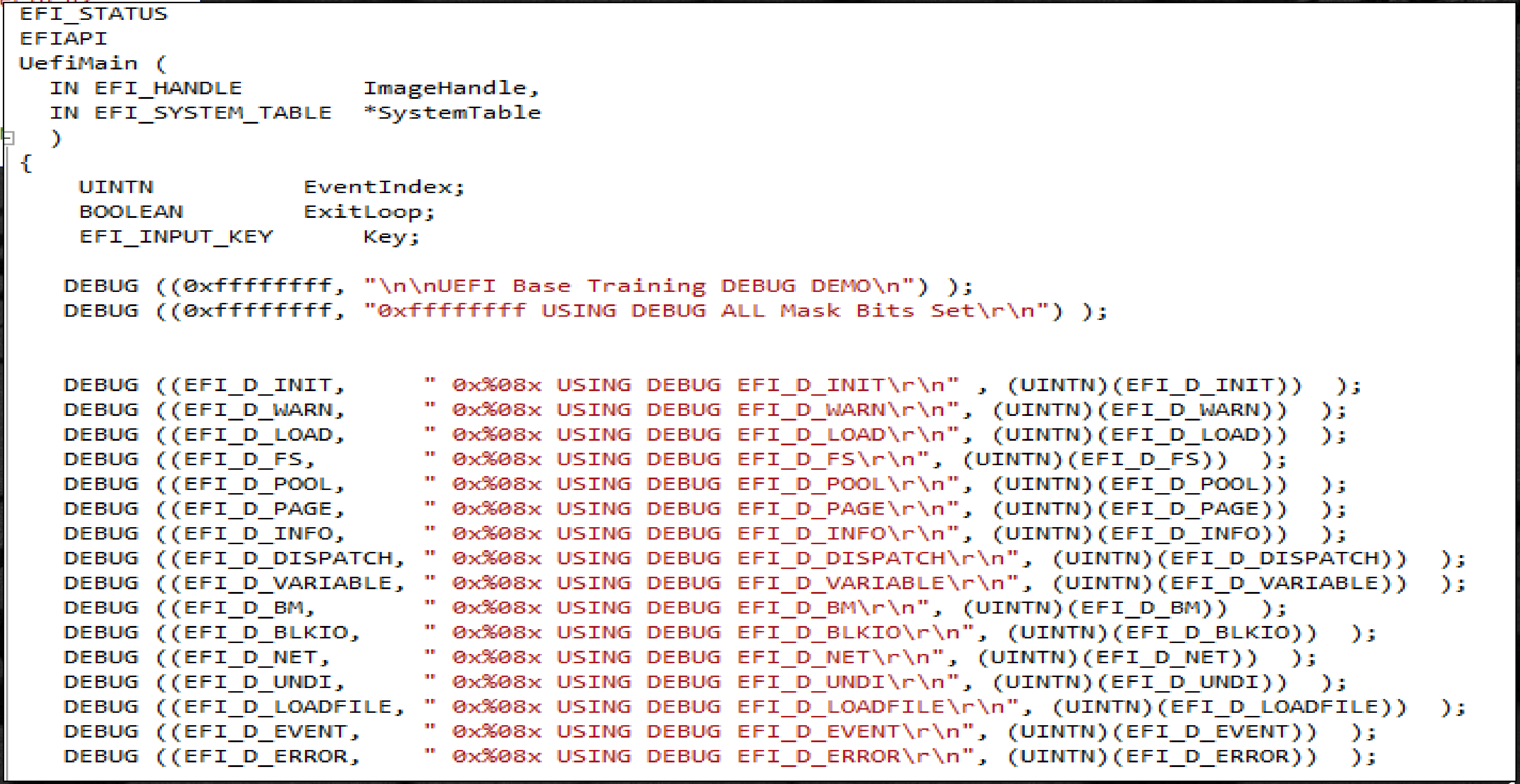
Locate the UefiMain function. Then copy and paste the following code after the “EFI_INPUT_KEY KEY;” statement: and before the first **`Print()`** statement as shown in the screen shot below:

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

DEBUG ((EFI_D_INIT,     " 0x%08x USING DEBUG EFI_D_INIT\n" , (UINTN)(EFI_D_INIT))  );
DEBUG ((EFI_D_WARN,     " 0x%08x USING DEBUG EFI_D_WARN\n", (UINTN)(EFI_D_WARN))  );
DEBUG ((EFI_D_LOAD,     " 0x%08x USING DEBUG EFI_D_LOAD\n", (UINTN)(EFI_D_LOAD))  );
DEBUG ((EFI_D_FS,       " 0x%08x USING DEBUG EFI_D_FS\n", (UINTN)(EFI_D_FS))  );
DEBUG ((EFI_D_POOL,     " 0x%08x USING DEBUG EFI_D_POOL\n", (UINTN)(EFI_D_POOL))  );
DEBUG ((EFI_D_PAGE,     " 0x%08x USING DEBUG EFI_D_PAGE\n", (UINTN)(EFI_D_PAGE))  );
DEBUG ((EFI_D_INFO,     " 0x%08x USING DEBUG EFI_D_INFO\n", (UINTN)(EFI_D_INFO))  );
DEBUG ((EFI_D_DISPATCH, " 0x%08x USING DEBUG EFI_D_DISPATCH\n",(UINTN)(EFI_D_DISPATCH)));
DEBUG ((EFI_D_VARIABLE, " 0x%08x USING DEBUG EFI_D_VARIABLE\n",(UINTN)(EFI_D_VARIABLE)));
DEBUG ((EFI_D_BM,       " 0x%08x USING DEBUG EFI_D_BM\n", (UINTN)(EFI_D_BM))  );
DEBUG ((EFI_D_BLKIO,    " 0x%08x USING DEBUG EFI_D_BLKIO\n", (UINTN)(EFI_D_BLKIO))  );
DEBUG ((EFI_D_NET,      " 0x%08x USING DEBUG EFI_D_NET\n", (UINTN)(EFI_D_NET))  );
DEBUG ((EFI_D_UNDI,     " 0x%08x USING DEBUG EFI_D_UNDI\n", (UINTN)(EFI_D_UNDI))  );
DEBUG ((EFI_D_LOADFILE, " 0x%08x USING DEBUG EFI_D_LOADFILE\n",(UINTN)(EFI_D_LOADFILE)));
DEBUG ((EFI_D_EVENT,    " 0x%08x USING DEBUG EFI_D_EVENT\n", (UINTN)(EFI_D_EVENT))  );
DEBUG ((EFI_D_ERROR,    " 0x%08x USING DEBUG EFI_D_ERROR\n", (UINTN)(EFI_D_ERROR))  );
```

Lab 1: Add debug statements to SampleApp



```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
#include <Library/UefiLib.h>
#include <Library/UefiBootServicesTableLib.h>
#include <Library/BaseMemoryLib.h>
#include <Library/DebugLib.h>

#define CHAR_DOT    0x002E    // '.' i
```



```
EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
    UINTN           EventIndex;
    BOOLEAN         ExitLoop;
    EFI_INPUT_KEY       Key;

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


   DEBUG ((EFI_D_INIT,     " 0x%08x USING DEBUG EFI_D_INIT\r\n" , (UINTN)(EFI_D_INIT))  );
   DEBUG ((EFI_D_WARN,     " 0x%08x USING DEBUG EFI_D_WARN\r\n", (UINTN)(EFI_D_WARN))  );
   DEBUG ((EFI_D_LOAD,     " 0x%08x USING DEBUG EFI_D_LOAD\r\n", (UINTN)(EFI_D_LOAD))  );
   DEBUG ((EFI_D_FS,       " 0x%08x USING DEBUG EFI_D_FS\r\n", (UINTN)(EFI_D_FS))  );
   DEBUG ((EFI_D_POOL,     " 0x%08x USING DEBUG EFI_D_POOL\r\n", (UINTN)(EFI_D_POOL))  );
   DEBUG ((EFI_D_PAGE,     " 0x%08x USING DEBUG EFI_D_PAGE\r\n", (UINTN)(EFI_D_PAGE))  );
   DEBUG ((EFI_D_INFO,     " 0x%08x USING DEBUG EFI_D_INFO\r\n", (UINTN)(EFI_D_INFO))  );
   DEBUG ((EFI_D_DISPATCH, " 0x%08x USING DEBUG EFI_D_DISPATCH\r\n", (UINTN)(EFI_D_DISPATCH))  );
   DEBUG ((EFI_D_VARIABLE, " 0x%08x USING DEBUG EFI_D_VARIABLE\r\n", (UINTN)(EFI_D_VARIABLE))  );
   DEBUG ((EFI_D_BM,       " 0x%08x USING DEBUG EFI_D_BM\r\n", (UINTN)(EFI_D_BM))  );
   DEBUG ((EFI_D_BLKIO,    " 0x%08x USING DEBUG EFI_D_BLKIO\r\n", (UINTN)(EFI_D_BLKIO))  );
   DEBUG ((EFI_D_NET,      " 0x%08x USING DEBUG EFI_D_NET\r\n", (UINTN)(EFI_D_NET))  );
   DEBUG ((EFI_D_UNDI,     " 0x%08x USING DEBUG EFI_D_UNDI\r\n", (UINTN)(EFI_D_UNDI))  );
   DEBUG ((EFI_D_LOADFILE, " 0x%08x USING DEBUG EFI_D_LOADFILE\r\n", (UINTN)(EFI_D_LOADFILE))  );
   DEBUG ((EFI_D_EVENT,    " 0x%08x USING DEBUG EFI_D_EVENT\r\n", (UINTN)(EFI_D_EVENT))  );
   DEBUG ((EFI_D_ERROR,    " 0x%08x USING DEBUG EFI_D_ERROR\r\n", (UINTN)(EFI_D_ERROR))  );
```

Lab 1: Build, Run and Test Result

At the VS Command Prompt

```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

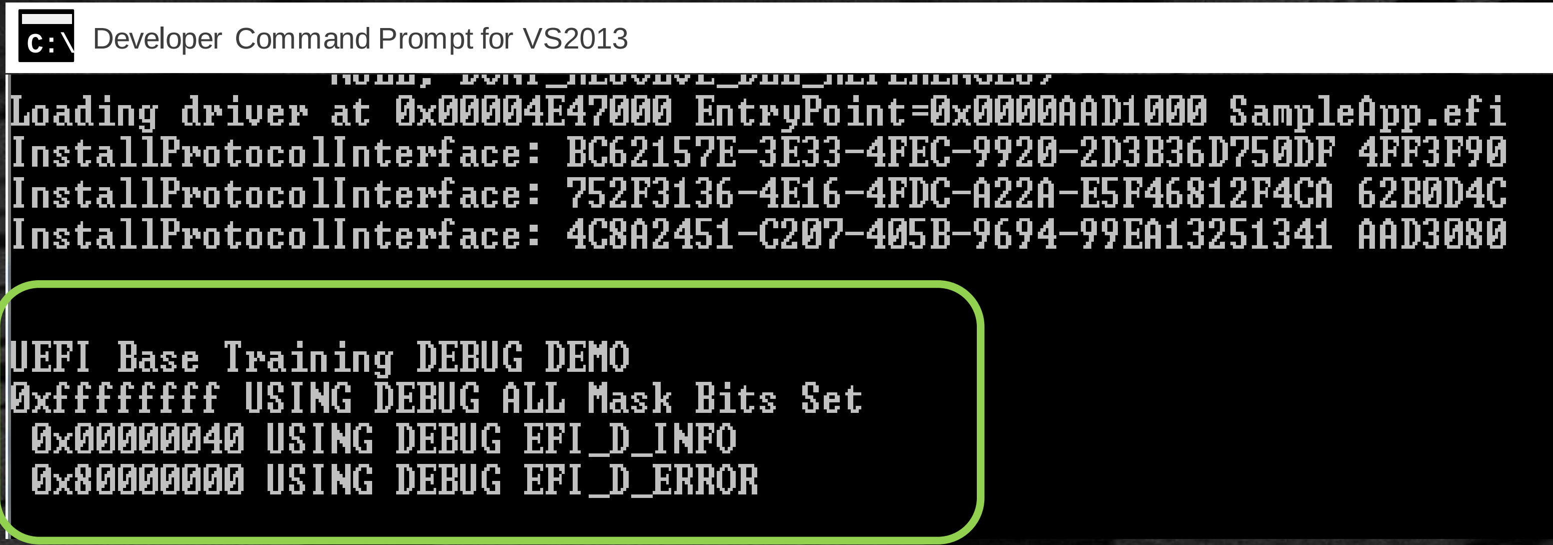
```
Shell> SampleApp
```

Check the VS Debug output

Exit

```
Shell> Reset
```

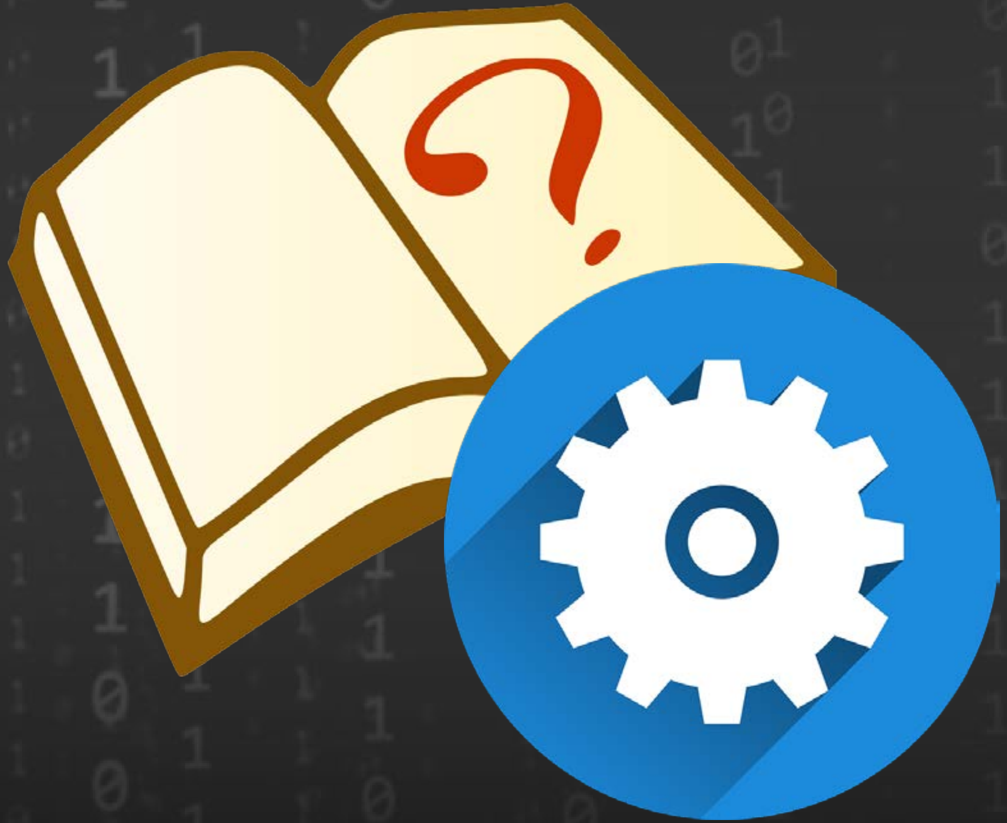
Visual Studio command prompt window output



```
Loading driver at 0x00004E47000 EntryPoint=0x0000AAD1000 SampleApp.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 4FF3F90
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 62B0D4C
InstallProtocolInterface: 4C8A2451-C207-405B-9694-99EA13251341 AAD3080

UEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set
 0x00000040 USING DEBUG EFI_D_INFO
 0x80000000 USING DEBUG EFI_D_ERROR
```

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/Nt32Pkg/Nt32Pkg.dsc
Replace SampleApp/SampleApp.inf with the following:

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

Save and close C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

tianocore

Lab 1: Build, Run and Test Result

At the VS Command Prompt

```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

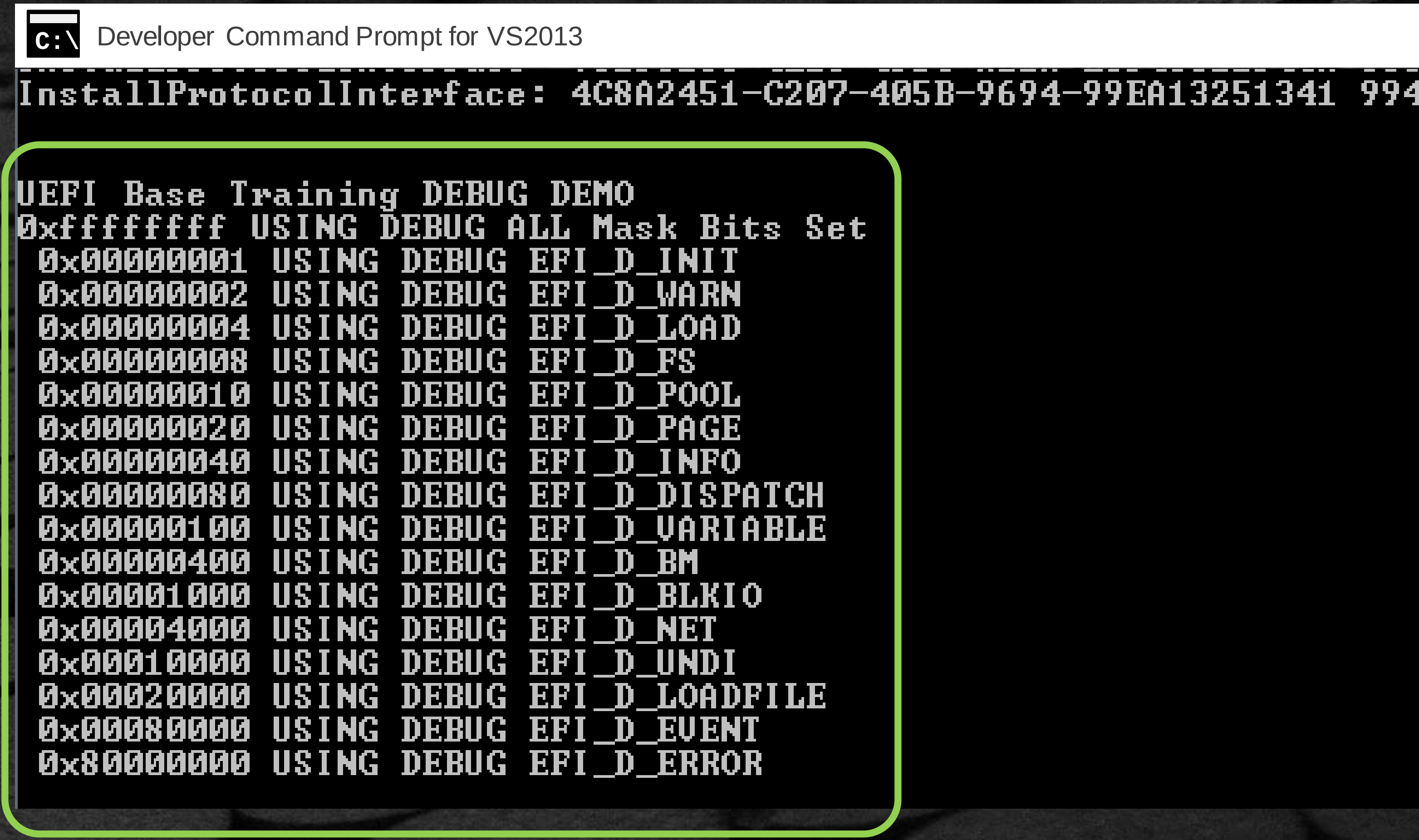
```
Shell> SampleApp
```

Check the VS Debug output

Exit

```
Shell> Reset
```

Visual Studio command prompt window output



Developer Command Prompt for VS2013

```
InstallProtocolInterface: 4C8A2451-C207-405B-9694-99EA13251341 994

UEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set
 0x00000001 USING DEBUG EFI_D_INIT
 0x00000002 USING DEBUG EFI_D_WARN
 0x00000004 USING DEBUG EFI_D_LOAD
 0x00000008 USING DEBUG EFI_D_FS
 0x00000010 USING DEBUG EFI_D_POOL
 0x00000020 USING DEBUG EFI_D_PAGE
 0x00000040 USING DEBUG EFI_D_INFO
 0x00000080 USING DEBUG EFI_D_DISPATCH
 0x00000100 USING DEBUG EFI_D_VARIABLE
 0x00000400 USING DEBUG EFI_D_BM
 0x00001000 USING DEBUG EFI_D_BLKIO
 0x00004000 USING DEBUG EFI_D_NET
 0x00010000 USING DEBUG EFI_D_UNDI
 0x00020000 USING DEBUG EFI_D_LOADFILE
 0x00080000 USING DEBUG EFI_D_EVENT
 0x80000000 USING DEBUG EFI_D_ERROR
```

tianocore

CHANGING FLAGS

Changing Compiler & Linker Flags

Precedence for Debug Flags Hierarchy

Tools_def.txt

Precedence for Debug Flags Hierarchy

DSC [BuildOptions] section
(platform scope)

tianocore

Precedence for Debug Flags Hierarchy

INF [BuildOptions] section

tianocore

Precedence for Debug Flags Hierarchy

DSC <BuildOptions>
under a specific module

tianocore

Precedence for Debug Flags Hierarchy

Tools_def.txt

DSC [BuildOptions] section (platform scope)

INF [BuildOptions] section

DSC <BuildOptions> under a specific module

1. Tools_def.txt
2. DSC [BuildOptions] section (platform scope)
3. INF [BuildOptions] section (module scope)
4. DSC <BuildOptions> under a specific module

tianocore

Compiler / Linker Flags

Example from Microsoft* compiler to turn off optimization

“`/O2`” to “`/O1`” requires ”`/Od /O1`” flags

tianocore

Compiler / Linker Flags

Example from Microsoft* compiler to turn off optimization

"**/O2**" to "**/O1**" requires "**/Od /O1**" flags

Change common flags in platform DSC

```
[BuildOptions]
    DEBUG_*_IA32_CC_FLAGS = /Od /Oy-
```

tianocore

Compiler / Linker Flags

Example from Microsoft* compiler to turn off optimization

"**/O2**" to "**/O1**" requires "**/Od /O1**" flags

Change common flags in platform DSC

```
[BuildOptions]
    DEBUG_*_IA32_CC_FLAGS = /Od /Oy-
```

Change a single module's flags in DSC

```
MyPath/MyModule.inf {
<BuildOptions>
   DEBUG_*_IA32_CC_FLAGS = /Od /Oy-
}
```

DebugLib USAGE

tianocore

Interface

The DebugLib Class

```
MdePkg\Include\Library\DebugLib.h
```

Macros

(where PCDs are checked)

```
    ASSERT (Expression)
    DEBUG (Expression)
    ASSERT_EFI_ERROR (StatusParameter)
    ASSERT_PROTOCOL_ALREADY_INSTALLED(…)
```

Advanced Macros

```
    DEBUG_CODE (Expression)
    DEBUG_CODE_BEGIN() & DEBUG_CODE_END()
    DEBUG_CLEAR_MEMORY(…)
```

tianocore

Implementation

DebugLib Instances (1)

BaseDebugLibSerialPort

- Instance of DebugLib
- Uses SerialPortLib class to send debug output to serial port
- Default for many platforms: BaseDebugLibNull
- OVMF uses it with Switch DEBUG_ON_SERIAL_PORT

Implementation

DebugLib Instances (2)

`UefiDebugLibConOut`   `UefiDebugLibStdErr`

- Instances of **`DebugLib`** (for apps and drivers)
- Send all debug output to console/debug console

tianocore

Implementation

DebugLib Instances (3)

`PeiDxeDebugLibReportStatusCode`

- Sends ASCII String specified by Description Value to the **`ReportStatusCode()`**
- May also use the **`SerialPortLib`** class to send debug output to serial port
- **`BaseDebugLibNull`** - Resolves references

Default for most platforms

tianocore

Changing Library Instances

Change common library instances in the platform DSC by module type

```
[LibraryClasses.common.IA32]
DebugLib|MdePkg/Library/BaseDebugLibNull/BaseDebugLibNull.inf
```

Changing Library Instances

Change common library instances in the platform DSC by module type

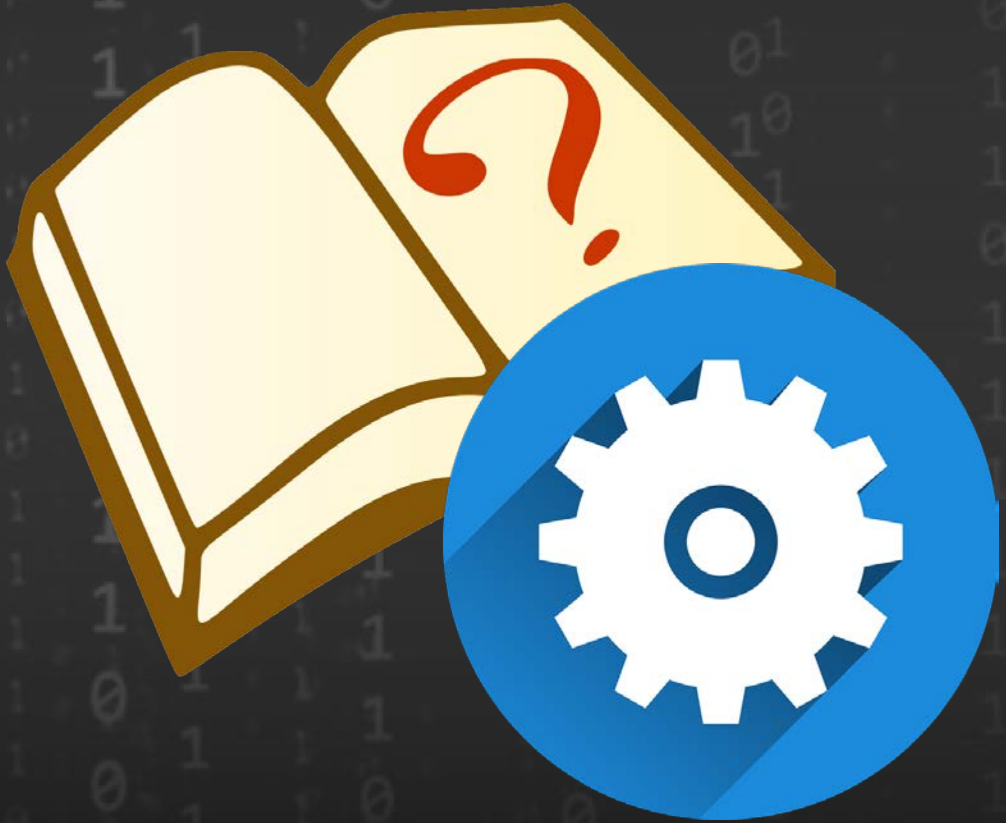
```
[LibraryClasses.common.IA32]
DebugLib|MdePkg/Library/BaseDebugLibNull/BaseDebugLibNull.inf
```

Change a single module’s library instance in the platform DSC

```
MyPath/MyModule.inf {
<LibraryClasses>
DebugLib|MdePkg/Library/BaseDebugLibSerialPort.inf
}
```

Lab 2 – Library Instances for Debugging

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



tianocore

Lab 3: Using Library Instances for Debugging

Open C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc
Replace SampleApp/SampleApp.inf { . . .} with the following:

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

Save and close C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

Lab 3: Build, Run and Test Result

At the VS Command Prompt

```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

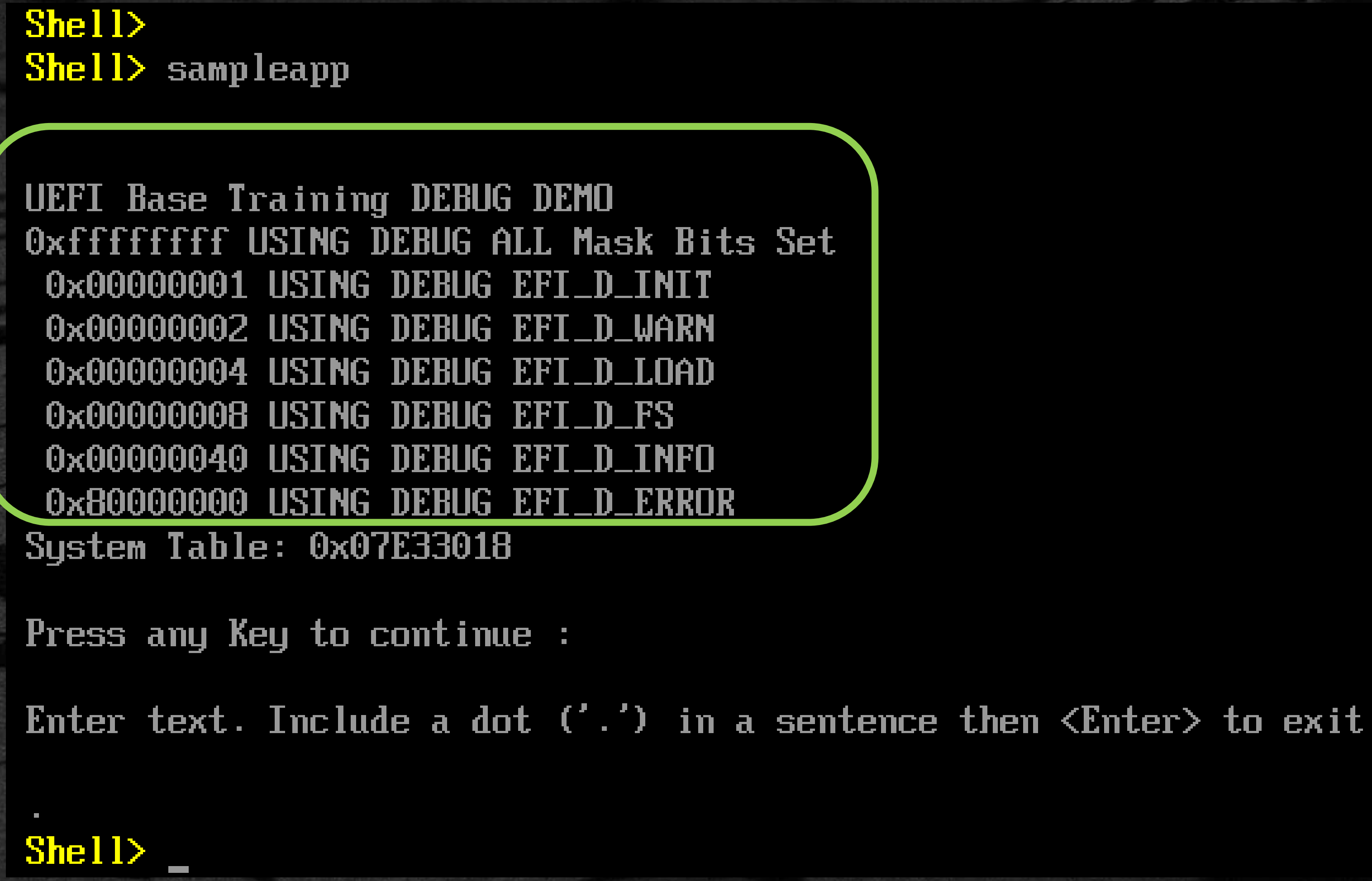
```
Shell> SampleApp
```

See that the output from the Debug statements now goes to the Nt32 console

Exit

```
Shell> Reset
```

Debug output to console

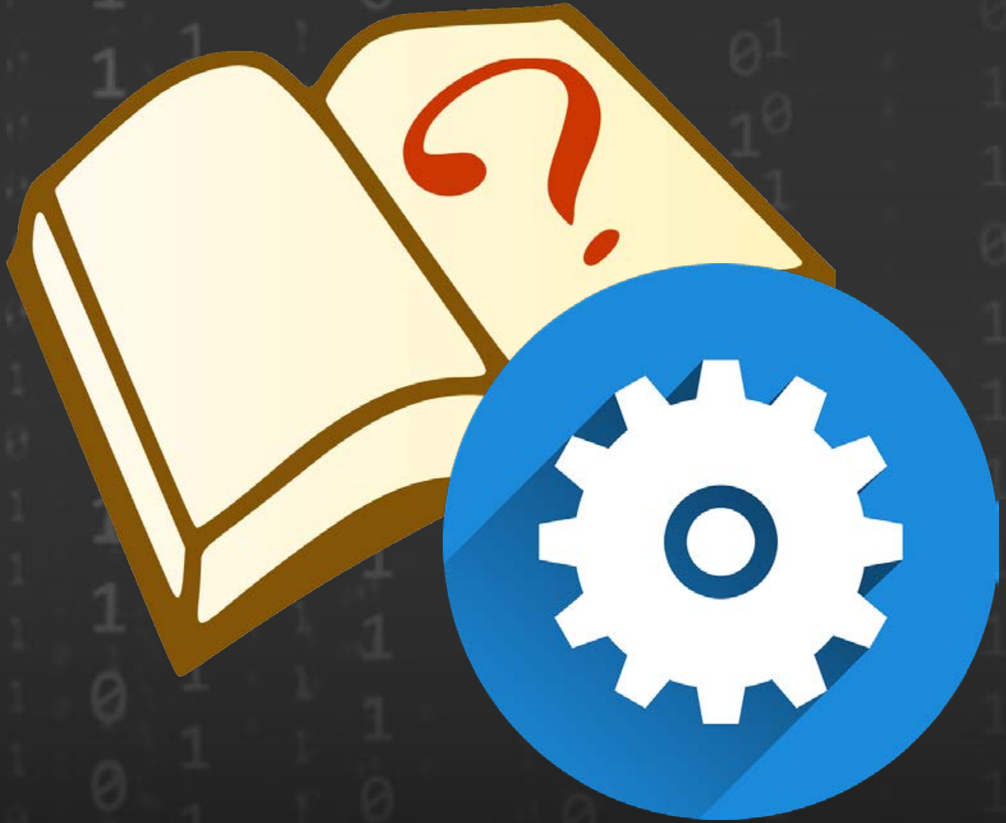


```
Shell>
Shell> sampleapp


UEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set
 0x00000001 USING DEBUG EFI_D_INIT
 0x00000002 USING DEBUG EFI_D_WARN
 0x00000004 USING DEBUG EFI_D_LOAD
 0x00000008 USING DEBUG EFI_D_FS
 0x00000040 USING DEBUG EFI_D_INFO
 0x80000000 USING DEBUG EFI_D_ERROR
System Table: 0x07E33018

Press any Key to continue :

Enter text. Include a dot ('.') in a sentence then <Enter> to exit


.
Shell> _
```

tianocore

Lab 4: Null Instance of DebugLib

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



tianocore

Lab 4: Using Null Library Instances

Open C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc
Replace SampleApp/SampleApp.inf { . . .} with the following:

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

Save and close C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

Lab 4: Build, Run and Test Result

At the VS Command Prompt

```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

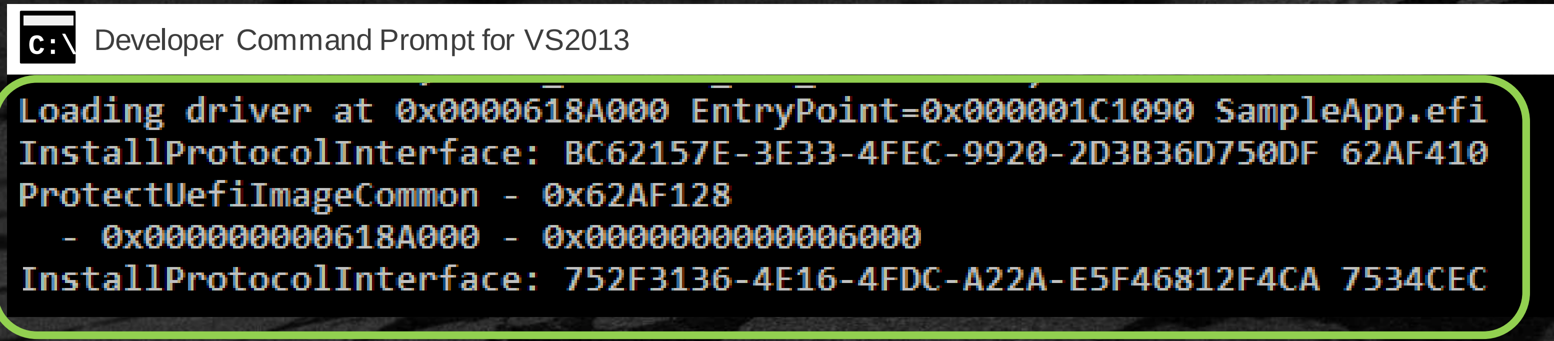
```
Shell> SampleApp
```

Check – now **NO** Debug output

Exit

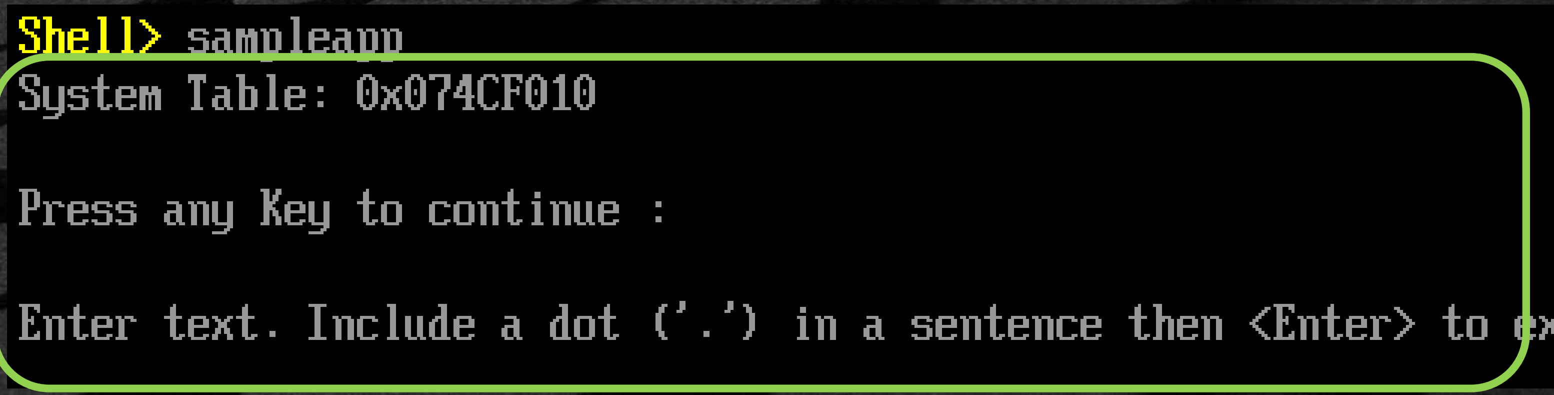
```
Shell> Reset
```

Visual Studio command prompt window output – NO DEBUG


Developer Command Prompt for VS2013

```
Loading driver at 0x0000618A000 EntryPoint=0x000001C1090 SampleApp.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 62AF410
ProtectUefiImageCommon - 0x62AF128
  - 0x000000000618A000 - 0x0000000000006000
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 7534CEC
```

Nt32 console window – NO DEBUG

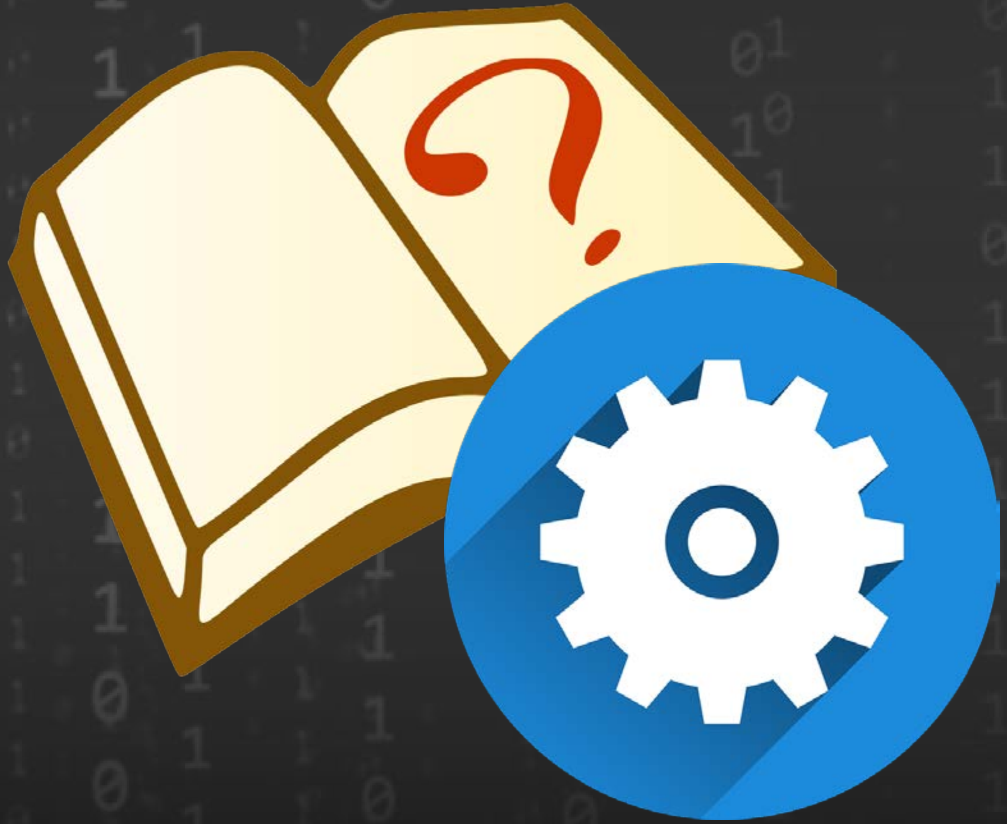


```
Shell> sampleapp
System Table: 0x074CF010

Press any Key to continue :

Enter text. Include a dot ('.') in a sentence then <Enter> to ex
```

tianocore

Lab 5: Debugging EDK II with VS Debugger

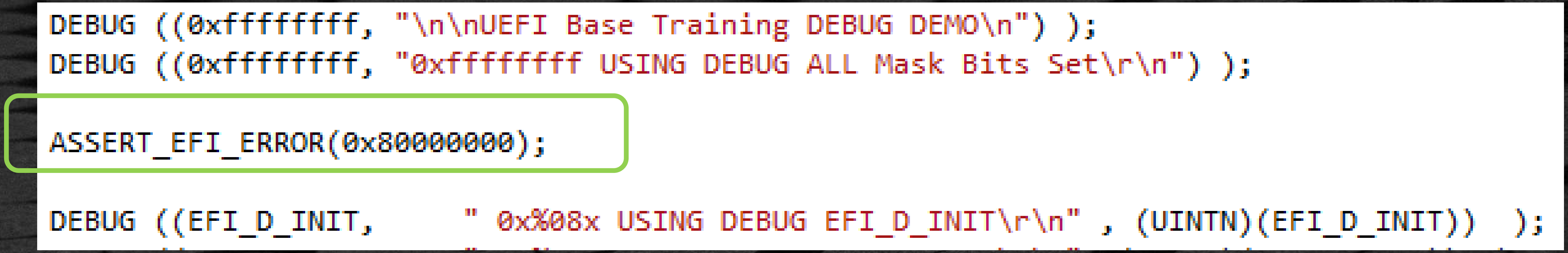
In this lab, you'll learn how setup the VS to debug the EDK II Nt32 emulation



Lab 5: Nt32 Debug with VS

Edit the SampleApp.c and add an “**`ASSERT_EFI_ERROR`**” Statement :

```
ASSERT_EFI_ERROR(0x80000000);
```



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

ASSERT_EFI_ERROR(0x80000000);

DEBUG ((EFI_D_INIT,     " 0x%08x USING DEBUG EFI_D_INIT\r\n" , (UINTN)(EFI_D_INIT))  );
```

Save SampleApp.c

tianocore

Lab 5: Nt32 Debug with VS

At the VS Command Prompt

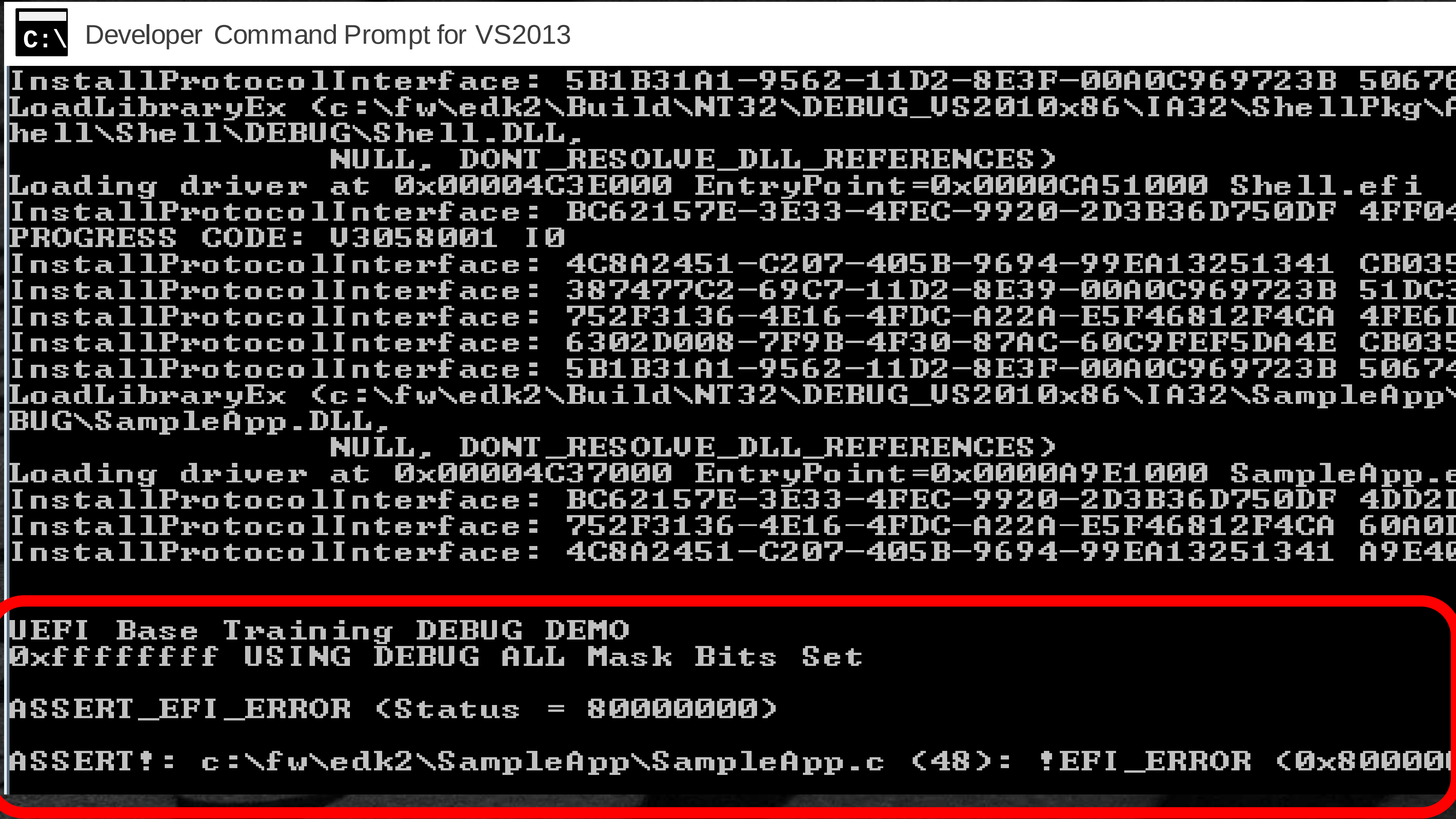
```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

```
Shell> SampleApp
```

Assert in VS Command Prompt

Visual Studio command prompt window output

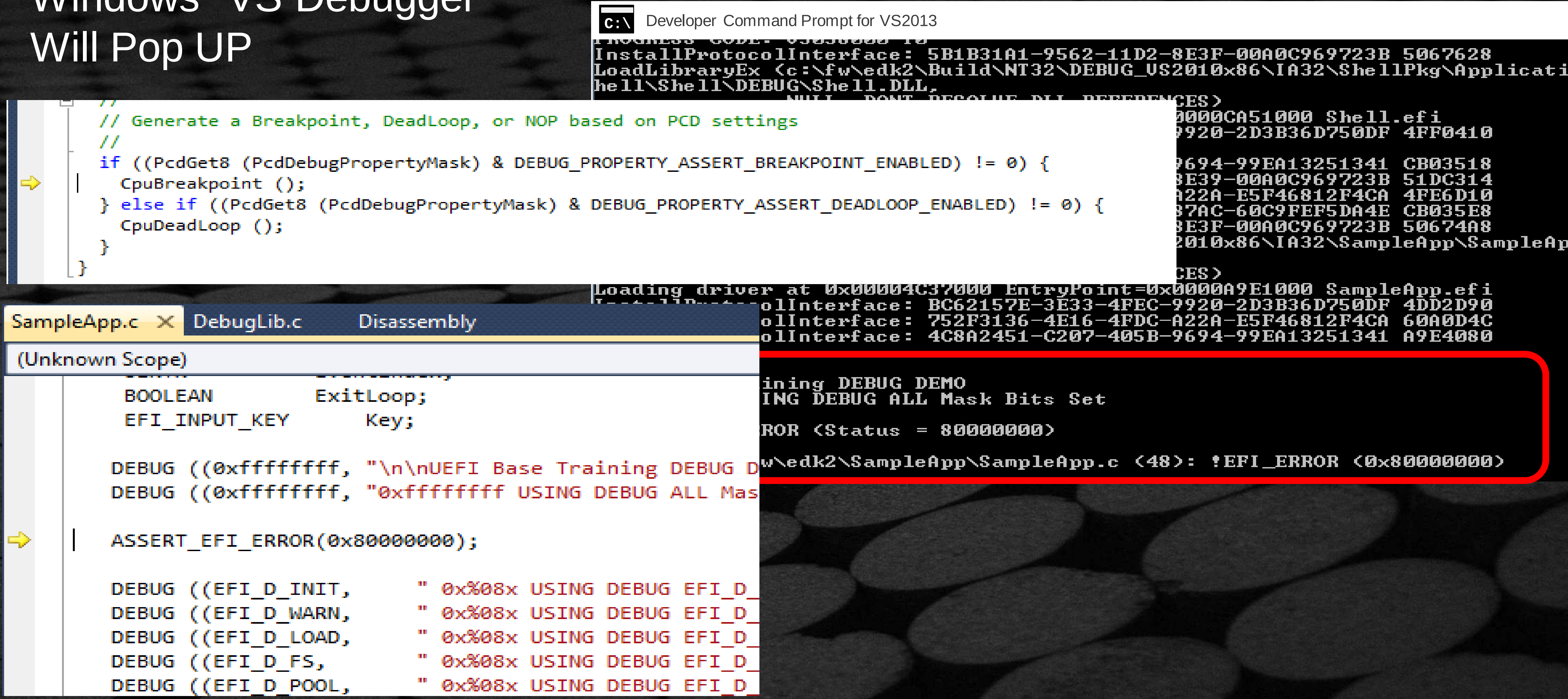


Developer Command Prompt for VS2013

```
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 5067
LoadLibraryEx (c:\fw\edk2\Build\NT32\DEBUG_VS2010x86\IA32\ShellPkg\
hell\Shell\DEBUG\Shell.DLL,
               NULL, DONT_RESOLVE_DLL_REFERENCES)
Loading driver at 0x00004C3E000 EntryPoint=0x0000CA51000 Shell.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 4FF0
PROGRESS CODE: V3058001 I0
InstallProtocolInterface: 4C8A2451-C207-405B-9694-99EA13251341 CB03
InstallProtocolInterface: 387477C2-69C7-11D2-8E39-00A0C969723B 51DC
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 4FE6
InstallProtocolInterface: 6302D008-7F9B-4F30-87AC-60C9FEF5DA4E CB03
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 5067
LoadLibraryEx (c:\fw\edk2\Build\NT32\DEBUG_VS2010x86\IA32\SampleApp\
BUG\SampleApp.DLL,
               NULL, DONT_RESOLVE_DLL_REFERENCES)
Loading driver at 0x00004C37000 EntryPoint=0x0000A9E1000 SampleApp.e
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 4DD2
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 60A0
InstallProtocolInterface: 4C8A2451-C207-405B-9694-99EA13251341 A9E4

UEFI Base Training DEBUG DEMO
0xffffffff USING DEBUG ALL Mask Bits Set

ASSERT_EFI_ERROR (Status = 80000000)

ASSERT!: c:\fw\edk2\SampleApp\SampleApp.c (48): !EFI_ERROR (0x80000
```

Lab 5: Nt32 Debug with VS

Windows* VS Debugger Will Pop UP



Developer Command Prompt for VS2013

```
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 5067628
LoadLibraryEx (c:\fw\edk2\Build\NT32\DEBUG_VS2010x86\IA32\ShellPkg\Applicati
hell\Shell\DEBUG\Shell.DLL,
                                                          CES)
                                                          0000CA51000 Shell.efi
                                                          9920-2D3B36D750DF 4FF0410

                                                          9694-99EA13251341 CB03518
                                                          8E39-00A0C969723B 51DC314
                                                          A22A-E5F46812F4CA 4FE6D10
                                                          87AC-60C9FEF5DA4E CB035E8
                                                          8E3F-00A0C969723B 50674A8
                                                          2010x86\IA32\SampleApp\SampleAp

                                                          CES)
Loading driver at 0x00004C37000 EntryPoint=0x0000A9E1000 SampleApp.efi
                      olInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 4DD2D90
                      olInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 60A0D4C
                      olInterface: 4C8A2451-C207-405B-9694-99EA13251341 A9E4080

                      ining DEBUG DEMO
                      ING DEBUG ALL Mask Bits Set

                      ROR (Status = 80000000)

                      w\edk2\SampleApp\SampleApp.c (48): !EFI_ERROR (0x80000000)
```

```c
  // Generate a Breakpoint, DeadLoop, or NOP based on PCD settings
  //
  if ((PcdGet8 (PcdDebugPropertyMask) & DEBUG_PROPERTY_ASSERT_BREAKPOINT_ENABLED) != 0) {
    CpuBreakpoint ();
  } else if ((PcdGet8 (PcdDebugPropertyMask) & DEBUG_PROPERTY_ASSERT_DEADLOOP_ENABLED) != 0) {
    CpuDeadLoop ();
  }
}
```

SampleApp.c | DebugLib.c | Disassembly

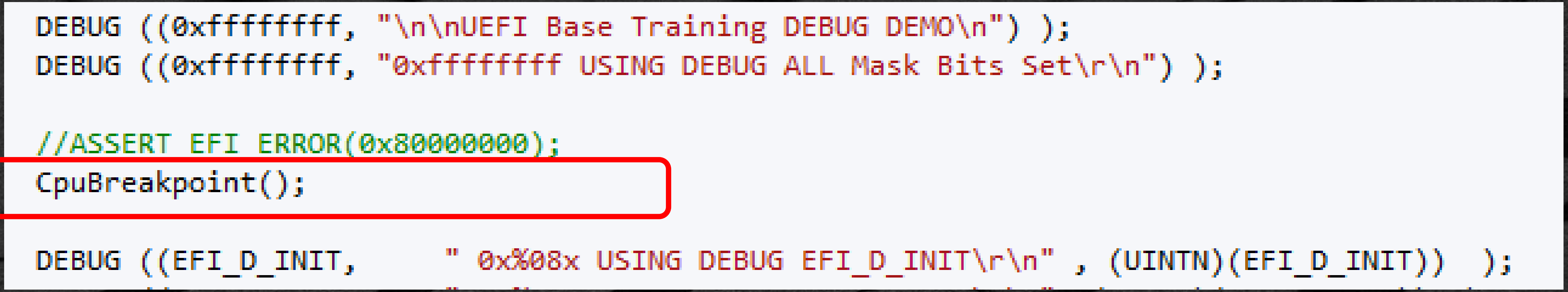
(Unknown Scope)

```c
    BOOLEAN         ExitLoop;
    EFI_INPUT_KEY       Key;

  DEBUG ((0xffffffff, "\n\nUEFI Base Training DEBUG D
  DEBUG ((0xffffffff, "0xffffffff USING DEBUG ALL Mas

  ASSERT_EFI_ERROR(0x80000000);

  DEBUG ((EFI_D_INIT,      " 0x%08x USING DEBUG EFI_D_
  DEBUG ((EFI_D_WARN,      " 0x%08x USING DEBUG EFI_D_
  DEBUG ((EFI_D_LOAD,      " 0x%08x USING DEBUG EFI_D_
  DEBUG ((EFI_D_FS,        " 0x%08x USING DEBUG EFI_D_
  DEBUG ((EFI_D_POOL,      " 0x%08x USING DEBUG EFI_D_
```

Lab 5: Nt32 Debug with VS

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

```
CpuBreakpoint();
```



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

//ASSERT_EFI_ERROR(0x80000000);
CpuBreakpoint();

DEBUG ((EFI_D_INIT,     " 0x%08x USING DEBUG EFI_D_INIT\r\n" , (UINTN)(EFI_D_INIT))  );
```

Save SampleApp.c

tianocore

Lab 5: Nt32 Debug with VS

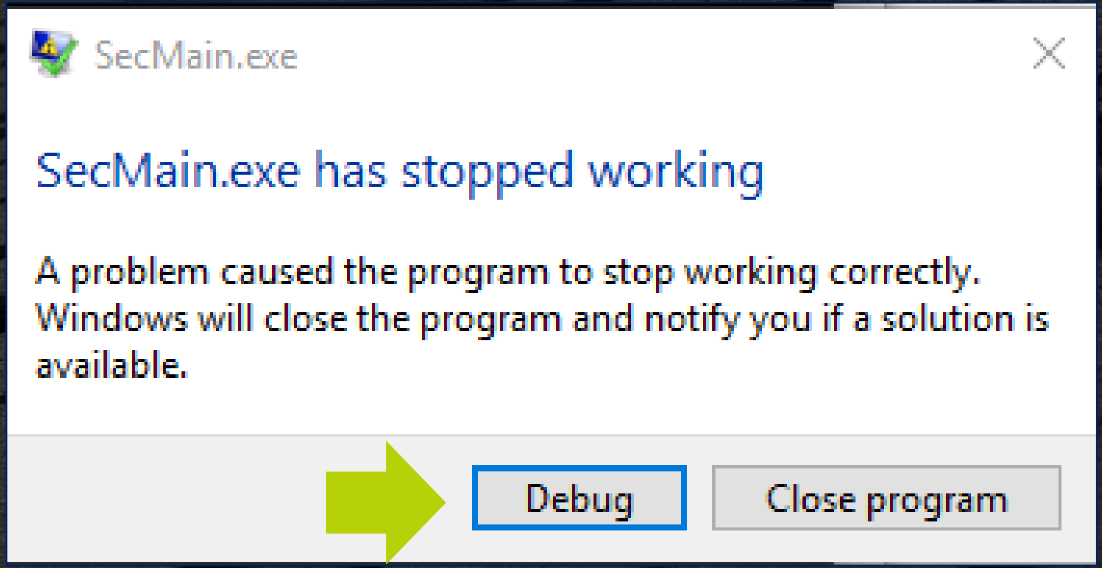
At the VS Command Prompt

```
C:/FW/edk2> Build
C:/FW/edk2> Build Run
```

Run the application from the shell

```
Shell> SampleApp
```

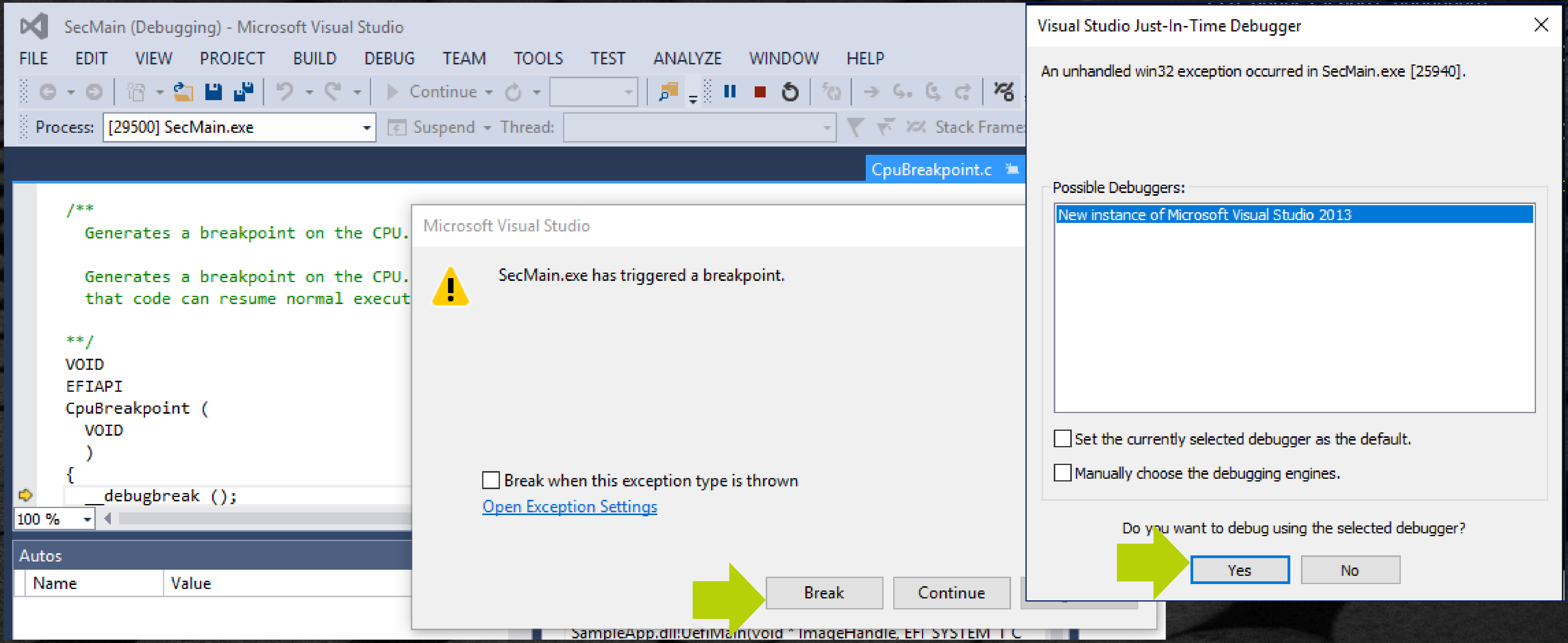
VS option go to VS Debugger


SecMain.exe

SecMain.exe has stopped working

A problem caused the program to stop working correctly. Windows will close the program and notify you if a solution is available.

Invoke Windows Visual Studio Debugger


SecMain (Debugging) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Continue
Process: [29500] SecMain.exe
Suspend
Thread:
Stack Frame:
CpuBreakpoint.c
100 %
Autos
Name
Value

```
/**
  Generates a breakpoint on the CPU.

  Generates a breakpoint on the CPU.
  that code can resume normal execut

**/
VOID
EFIAPI
CpuBreakpoint (
  VOID
  )
{
  __debugbreak ();
```

SampleApp.dll!UefiMain(void * ImageHandle, EFI_SYSTEM_T C

Microsoft Visual Studio

SecMain.exe has triggered a breakpoint.

☐ Break when this exception type is thrown

Open Exception Settings

Visual Studio Just-In-Time Debugger

An unhandled win32 exception occurred in SecMain.exe [25940].

Possible Debuggers:

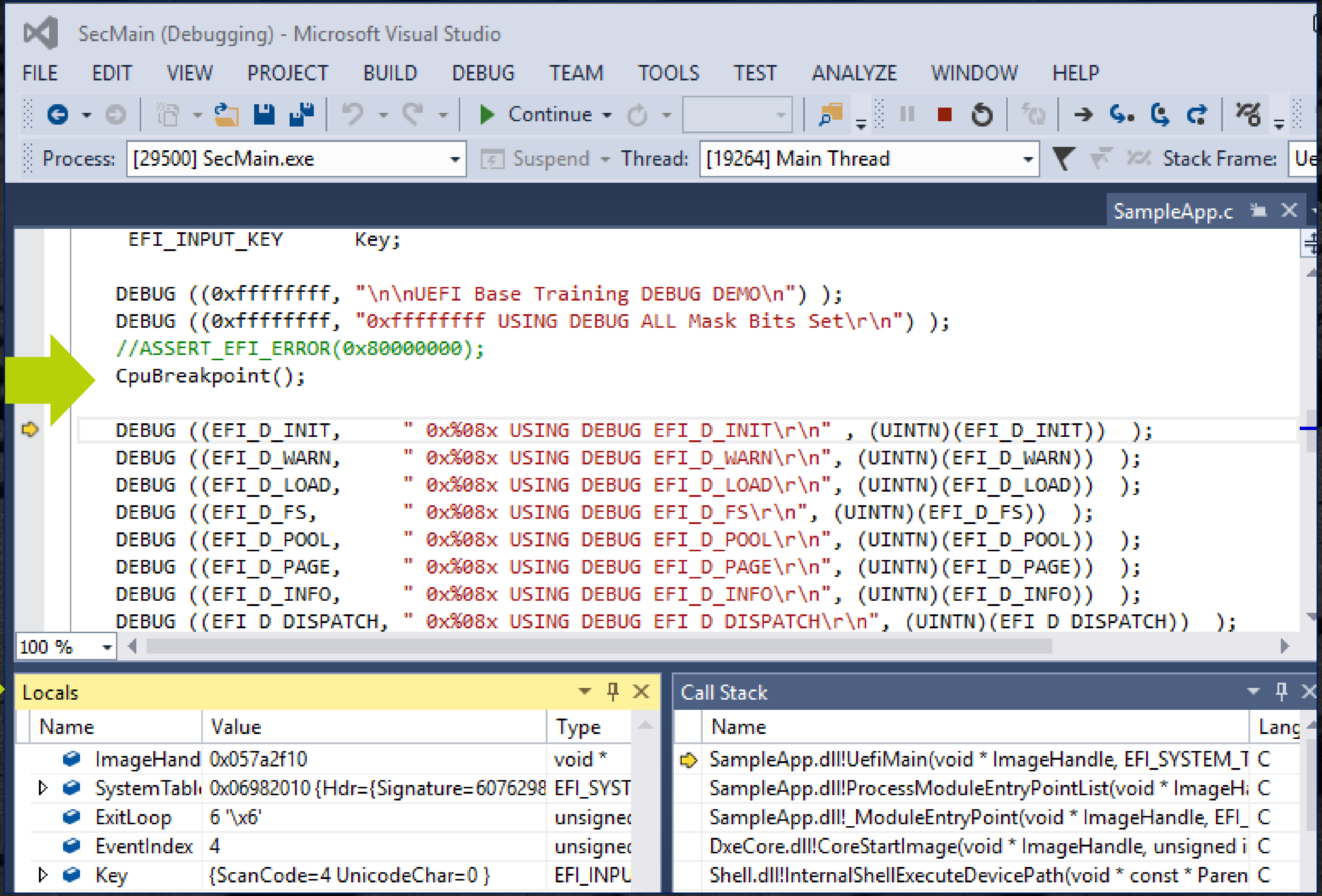
New instance of Microsoft Visual Studio 2013

☐ Set the currently selected debugger as the default.

☐ Manually choose the debugging engines.

Do you want to debug using the selected debugger?

Invoke Windows Visual Studio Debugger


SecMain (Debugging) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Continue
Process: [29500] SecMain.exe
Suspend
Thread: [19264] Main Thread
Stack Frame:
SampleApp.c
100 %

```
 EFI_INPUT_KEY      Key;

DEBUG ((0xffffffff, "\n\nUEFI Base Training DEBUG DEMO\n") );
DEBUG ((0xffffffff, "0xffffffff USING DEBUG ALL Mask Bits Set\r\n") );
//ASSERT_EFI_ERROR(0x80000000);
CpuBreakpoint();

DEBUG ((EFI_D_INIT,     " 0x%08x USING DEBUG EFI_D_INIT\r\n" , (UINTN)(EFI_D_INIT))  );
DEBUG ((EFI_D_WARN,     " 0x%08x USING DEBUG EFI_D_WARN\r\n", (UINTN)(EFI_D_WARN))  );
DEBUG ((EFI_D_LOAD,     " 0x%08x USING DEBUG EFI_D_LOAD\r\n", (UINTN)(EFI_D_LOAD))  );
DEBUG ((EFI_D_FS,       " 0x%08x USING DEBUG EFI_D_FS\r\n", (UINTN)(EFI_D_FS))  );
DEBUG ((EFI_D_POOL,     " 0x%08x USING DEBUG EFI_D_POOL\r\n", (UINTN)(EFI_D_POOL))  );
DEBUG ((EFI_D_PAGE,     " 0x%08x USING DEBUG EFI_D_PAGE\r\n", (UINTN)(EFI_D_PAGE))  );
DEBUG ((EFI_D_INFO,     " 0x%08x USING DEBUG EFI_D_INFO\r\n", (UINTN)(EFI_D_INFO))  );
DEBUG ((EFI_D_DISPATCH, " 0x%08x USING DEBUG EFI_D_DISPATCH\r\n", (UINTN)(EFI_D_DISPATCH))  );
```

Locals

Name	Value	Type
ImageHand	0x057a2f10	void *
SystemTabl	0x06982010 {Hdr={Signature=6076298	EFI_SYST
ExitLoop	6 '\x6'	unsigne
EventIndex	4	unsigne
Key	{ScanCode=4 UnicodeChar=0 }	EFI_INPU

Call Stack

Name	Lang
SampleApp.dll!UefiMain(void * ImageHandle, EFI_SYSTEM_T	C
SampleApp.dll!ProcessModuleEntryPointList(void * ImageH	C
SampleApp.dll!_ModuleEntryPoint(void * ImageHandle, EFI_	C
DxeCore.dll!CoreStartImage(void * ImageHandle, unsigned i	C
Shell.dll!InternalShellExecuteDevicePath(void * const * Paren	C

tianocore

SUMMARY

- Define **`DebugLib`** and its attributes
- List the ways to debug
- Using PCDs to Configure DebugLib - LAB
- Change Compiler & Linker Flags for debugging
- Change the **`DebugLib`** instance to modify the debug output - LAB
- Debug EDK II using VS Debugger - LAB

tianocore

tianocore

tianocore

www.tianocore.org

BACK UP

tianocore

ISSUE:

Debugging in Nt32 Emulation 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