tianocore

UEFI & EDK II Training

How to Write a UEFI Application w/ Windows Lab

tianocore.org

LESSON OBJECTIVE

- UEFI Application with PCDs
- Simple UEFI Application
- Add functionality to UEFI Application
- Using EADK with UEFI Application

UEFI APPLICATION W/ PCDS

tianocore

Review

EDK II PCD's Purpose and Goals

Documentaton : MdeModulePkg/Universal/PCD/Dxe/Pcd.inf

Purpose

- Establishes platform common definitions
- Build-time/Run-time aspects
- Binary Editing Capabilities

Goals

- Simplify porting
- Easy to associate with a module or platform

PCD SYNTAX

Review

PCDs can be located anywhere within the Workspace even though a different package will use those PCDs for a given project

.DEC

Define PCD

.INF

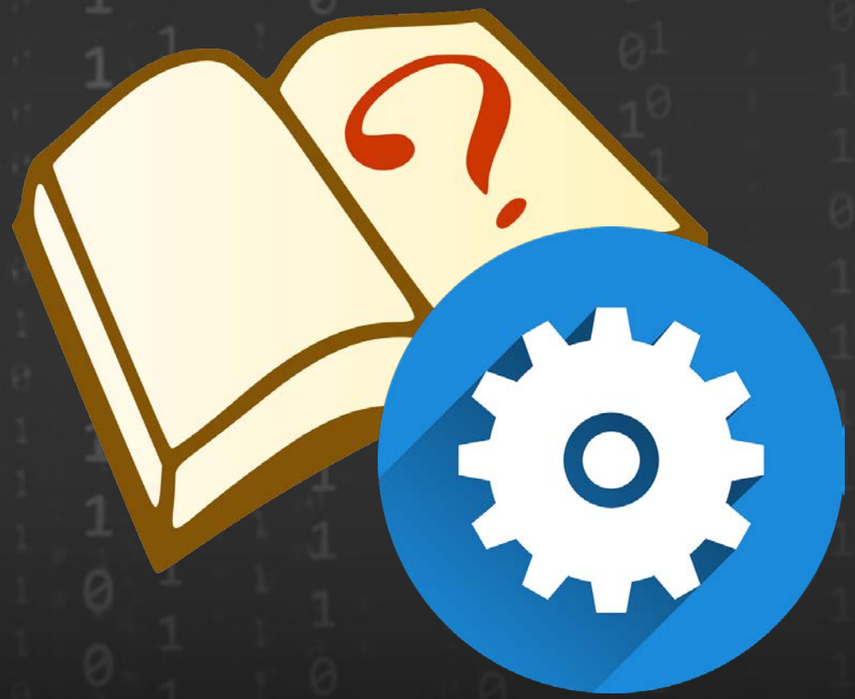
Reference PCD

.DSC

Modify PCD

Lab 1: Writing UEFI Applications with PCDs

In this lab, you'll learn how to write UEFI applications with PCDs.



EDK II HelloWorld App Lab

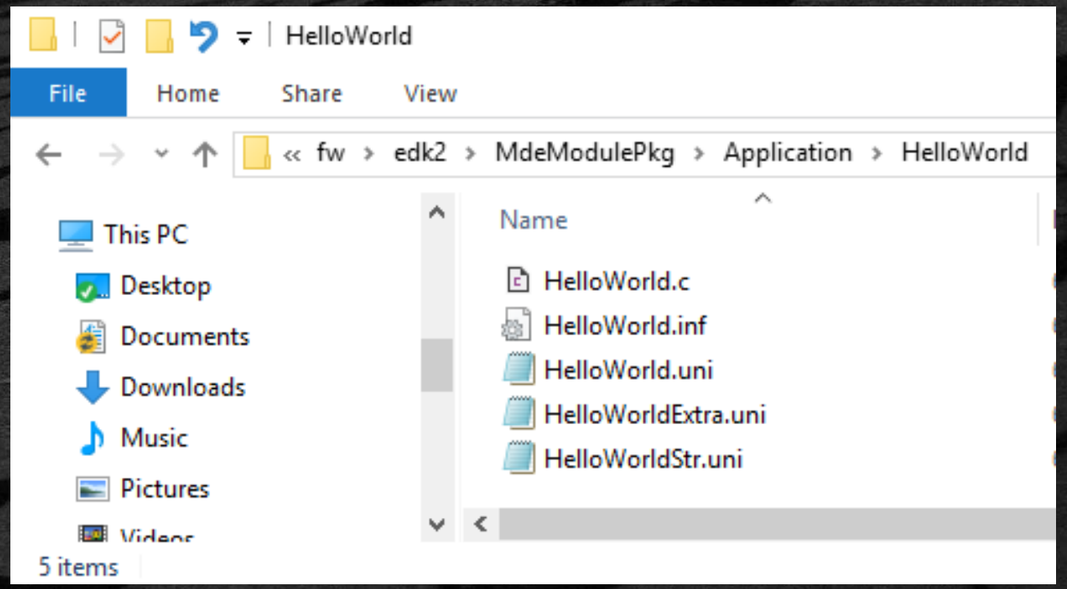
First Setup for Building EDK II for Nt32, See Lab Setup

Locate and Open MdeModulePkg\Application\HelloWorld\HelloWorld.c

Notice the PCD values

Build Nt32 Emulation

Then Run HelloWorld



EDK II HelloWorld App Lab

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

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

Build the Nt32 Emulation

```
C:/FW/edk2> Build –D BUILD_NEW_SHELL
C:/FW/edk2> Build Run
```

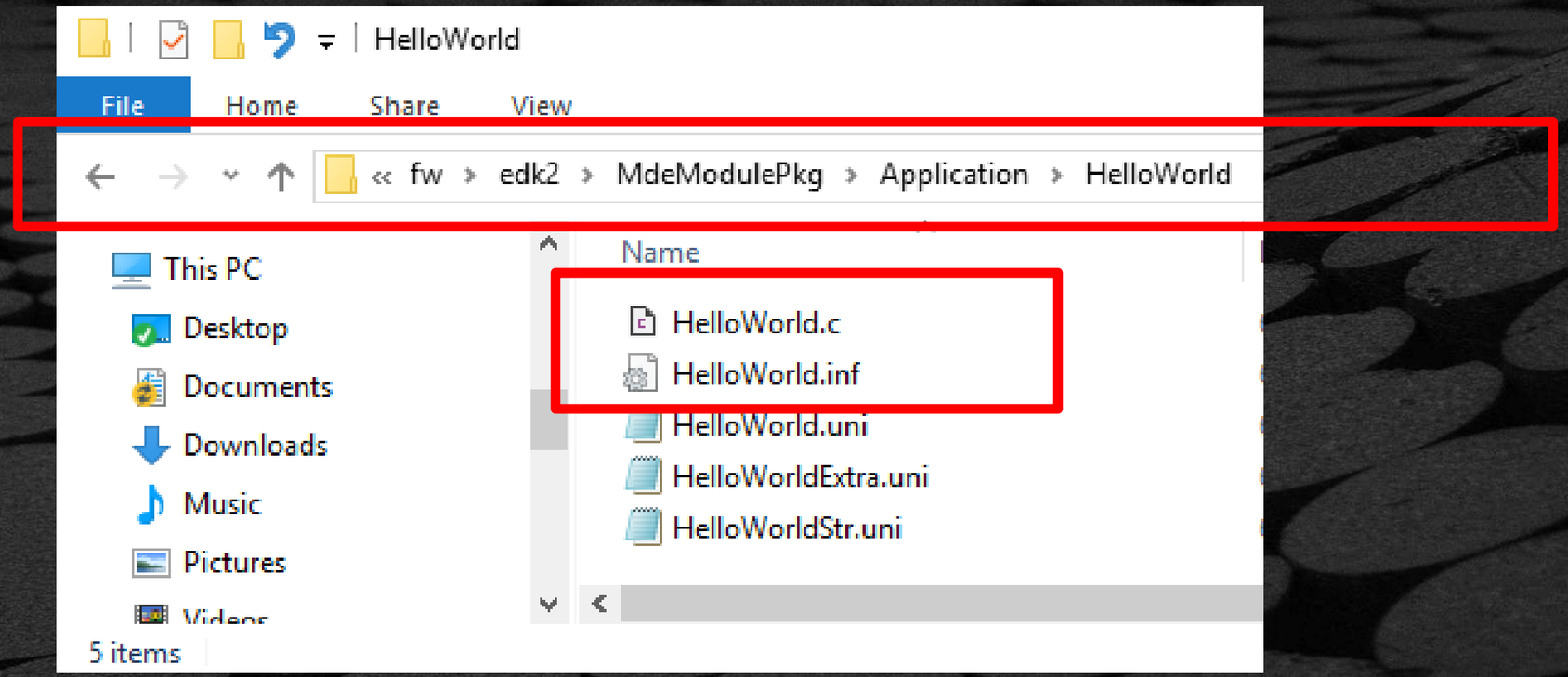
At the UEFI Shell prompt

```
Shell> Helloworld
UEFI Hello World!
Shell>
```

How can we force the HelloWorld application to print out 3 times ?

EDK II HelloWorld App Lab

MdeModulePkg/Application/HelloWorld



EDK II HelloWorld App

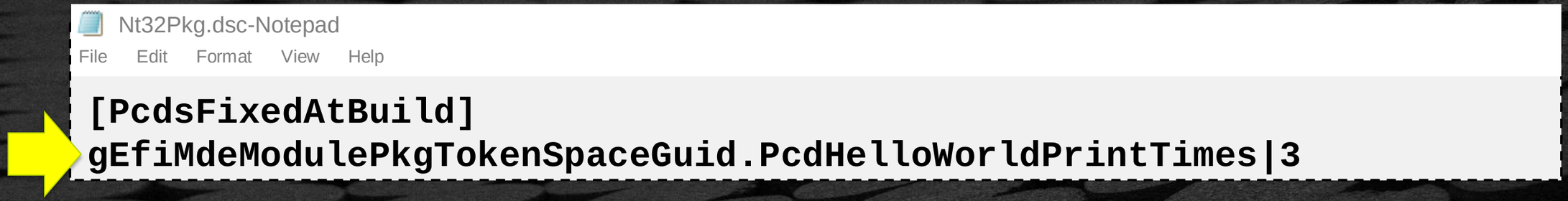
Source HelloWorld.c

```
EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
  UINT32 Index;
  Index = 0;
  // Three PCD type (FeatureFlag, UINT32
  // and String) are used as the sample.
  if (FeaturePcdGet (PcdHelloWorldPrintEnable)) {
  for (Index = 0; Index < PcdGet32   (PcdHelloWorldPrintTimes); Index ++) {

    // Use UefiLib Print API to print
      // string to UEFI console

      Print ((CHAR16*)PcdGetPtr (PcdHelloWorldPrintString));
    }
  }

  return EFI_SUCCESS;
}
```

EDK II HelloWorld App Solution

1. Edit the file C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

After the section `[PcdsFixedAtBuild]` (search for **"PcdsFixedAtBuild"** or **"Hello"**)



2. Re-Build – Cd to C:/FW/edk2

```
C:/FW/edk2: > build  -D BUILD_NEW_SHELL
```

EDK II HelloWorld App Solution

3. Run Nt32 Emulation

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

4. At the Shell prompt

```
Shell> Helloworld
UEFI Hello World!
UEFI Hello World!
UEFI Hello World!
Shell>
```

5: Exit Nt32 Emulation

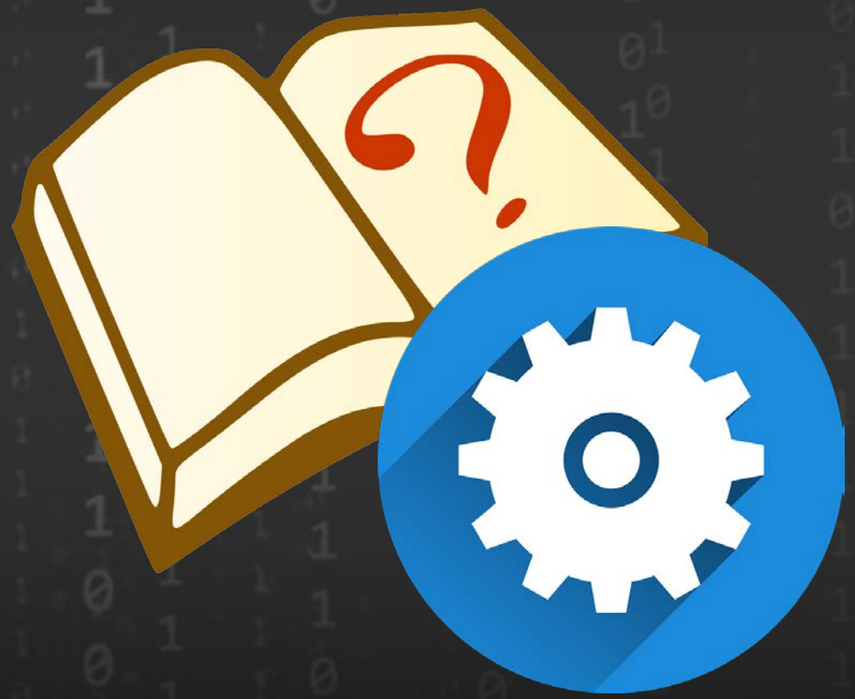
```
Shell> Reset
```

How can we change the **string** of the HelloWorld application?

Also see ~src/edk2/MdeModulePkg/MdeModulePkg.Dec

Lab 2: Write a Simple UEFI Applications

In this lab, you'll learn how to write simple UEFI applications.



LAB 2 Writing a Simple UEFI Application

In this lab, you'll learn how to write simple UEFI applications.

"C" file

```
EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
  return EFI_SUCCESS;
}
```

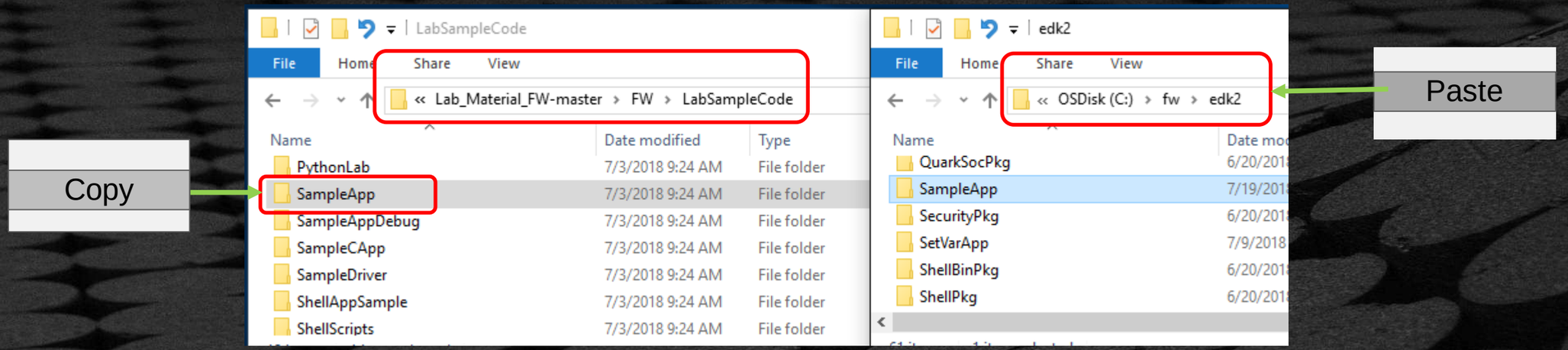
.inf file

```
[Defines]
  INF_VERSION         =
  BASE_NAME           =
  FILE_GUID           =
  MODULE_TYPE         =
  VERSION_STRING      =
  ENTRY_POINT         =

[Sources]

[Packages]

[LibraryClasses]
```

- What goes into a Simplest "C"
- Start with what should go into the Simplest .INF file

Application Lab –start with .c and .inf template

Copy the **`LabSampleCode/SampleApp`** directory to **`C:/FW/edk2`**



Edit **`SampleApp.inf`**

- Look in the INF for **“XXXXXXXXXX”** sections that will need information
- Create Name & GUID, and then fill in the MODULE_TYPE

Lab 2: Sample Application INF file

SampleApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION      = 0x00010005
  BASE_NAME        = XXXXXXXXXXX
  FILE_GUID        = XXXXXXXXXXX
  MODULE_TYPE      = XXXXXXXXXXX
  VERSION_STRING   = 1.0
  ENTRY_POINT      = UefiMain

[Sources]
  XXXXXXXXX
[Packages]
  #XXXXXXXX

[LibraryClasses]
  #XXXXXXXXXXXX

[Guids]
 # . . .
```

SampleApp

Get a GUID

UEFI_APPLICATION

SampleApp.c

Get a GUID guidgenerator.com/

Lab 2: Sample Application ‘C’ file

SampleApp.c - Notepad

File Edit Format View Help

```
/** @file
  This is a simple shell application
**/
EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
  return EFI_SUCCESS;
}
```

Does not do anything but return Success

Lab 2: Will it compile now?

Not yet …

1. Need to add headers to the .C file
2. Need to add a reference to INF from the platform DSC
3. Need to add a few Package dependencies and libraries to the .INF

Application Lab – Update Files

1. .DSC (Nt32Pkg/Nt32Pkg.dsc)

```
[Components . . .]
```

Add INF to components section, before build options

Hint: add after comment: `# Add new modules here`

```
SampleApp/SampleApp.inf
```

2. .INF File (**SampleApp/SampleApp.inf**)

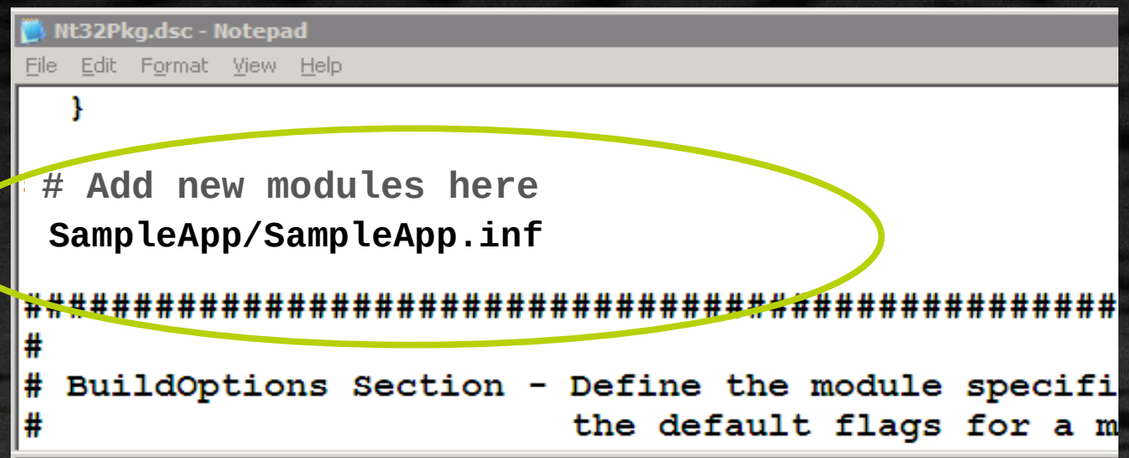
Packages (all depend on MdePkg)

```
[Packages]
    MdePkg/MdePkg.dec
[LibraryClasses]
    UefiApplicationEntryPoint
```

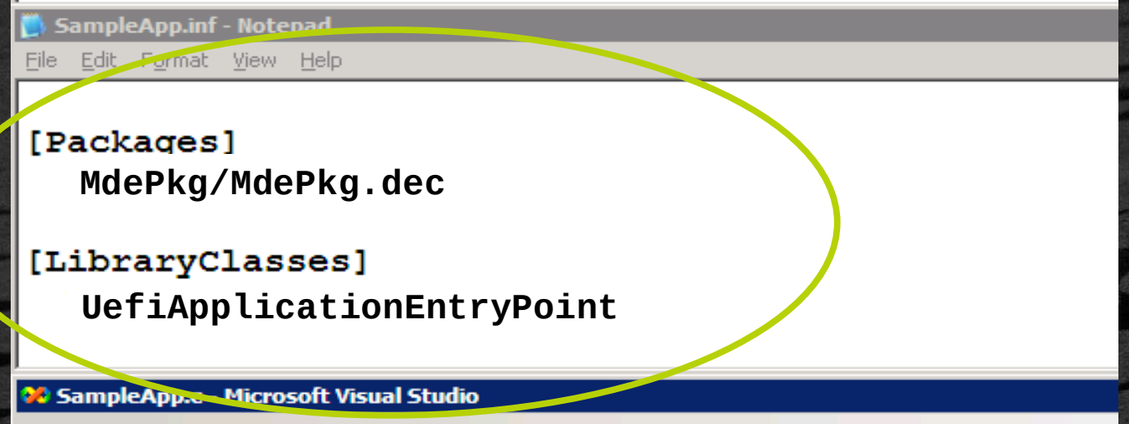
3. .C file - Header references File (**SampleApp/SampleApp.c**)

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
```

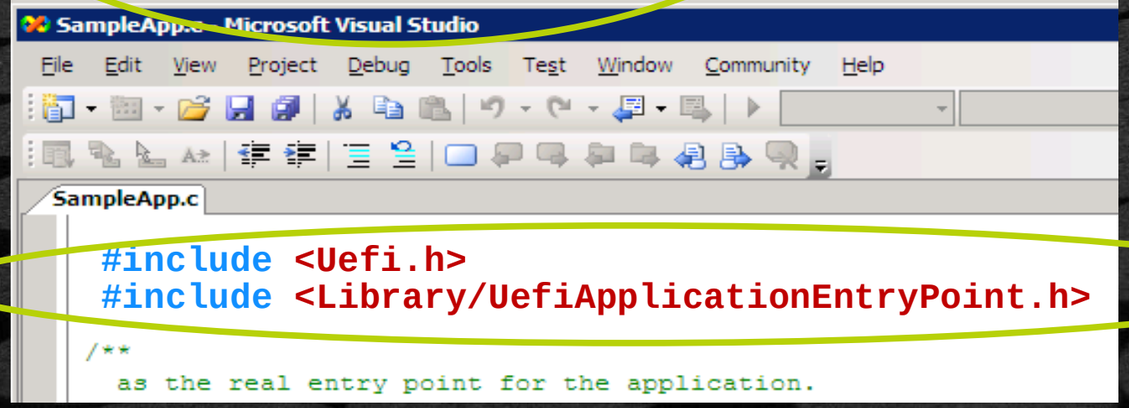
Lab 2: cont. Solution



Nt32Pkg/Nt32Pkg.dsc



SampleApp/SampleApp.inf



SampleApp/SampleApp.c

Lab 2 : Will it compile now?

At the VS Command Prompt

```
C:/FW/edk2> Build -D BUILD_NEW_SHELL
C:/FW/edk2> Build Run
```

Run the application from the shell

```
Shell> SampleApp
Shell>
```

Notice that the program will immediately unload because the main function is empty

Exit

```
Shell> Reset
```

Possible Build Errors

Error on SampleApp.inf

Developer Command Prompt for VS2013

```
Processing meta-data ..

build...
c:\fw\edk2\SampleApp\SampleApp.inf(21): error 3000: No value specified
        FILE_GUID                      =


- Failed -
Build end time: 09:11:30, Jul.25 2018
Build total time: 00:00:03


C:\fw\edk2>
```

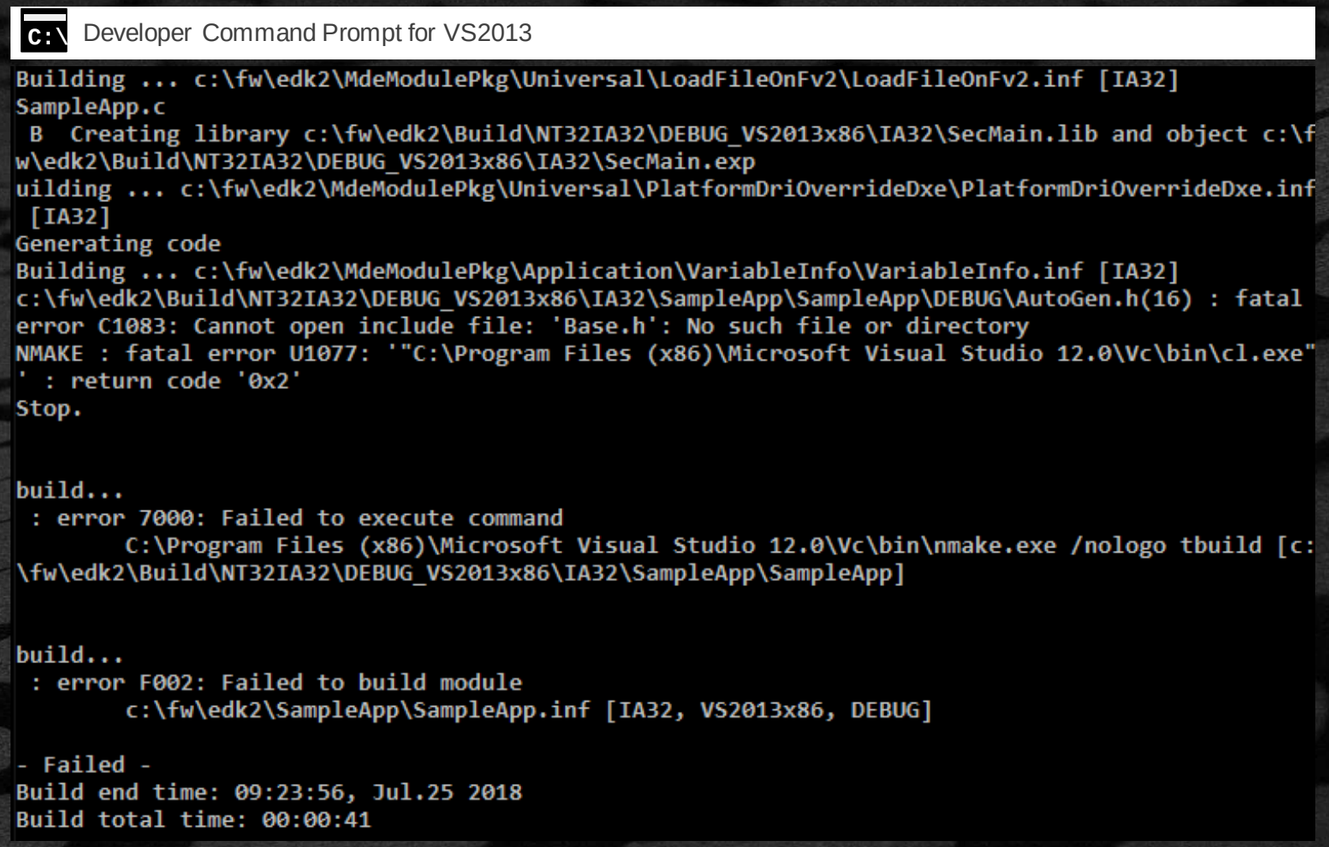
Developer Command Prompt for VS2013

```
Processing meta-data ......................


build...
 : error C0DE: Unknown fatal error when processing [c:\fw\edk2\SampleApp\SampleApp.inf]

(Please send email to edk2-devel@lists.01.org for help, attaching following call stack trace!)

(Python 2.7.14 on win32) Traceback (most recent call last):
  File "build\build.py", line 2493, in Main
  File "build\build.py", line 2226, in Launch
  File "build\build.py", line 2047, in _MultiThreadBuildPlatform
  File "c:\Users\Public\Documents\BuildPool\BaseTools\build\Source\Python\AutoGen\AutoGen.py",
line 4391, in CreateCodeFile
  File "c:\Users\Public\Documents\BuildPool\BaseTools\build\Source\Python\AutoGen\AutoGen.py",
line 3604, in _GetAutoGenFileList
  File "c:\Users\Public\Documents\BuildPool\BaseTools\build\Source\Python\AutoGen\GenC.py", lin
e 2075, in CreateCode
  File "c:\Users\Public\Documents\BuildPool\BaseTools\build\Source\Python\AutoGen\GenC.py", lin
e 2033, in CreateHeaderCode
  File "c:\Users\Public\Documents\BuildPool\BaseTools\build\Source\Python\Common\Misc.py", line
 308, in GuidStringToGuidStructureString
IndexError: list index out of range


- Failed -
Build end time: 09:15:55, Jul.25 2018
Build total time: 00:00:24
```

The FILE_GUID was invalid or not updated from “XXX…” to a proper formatted GUID

Possible Build Errors

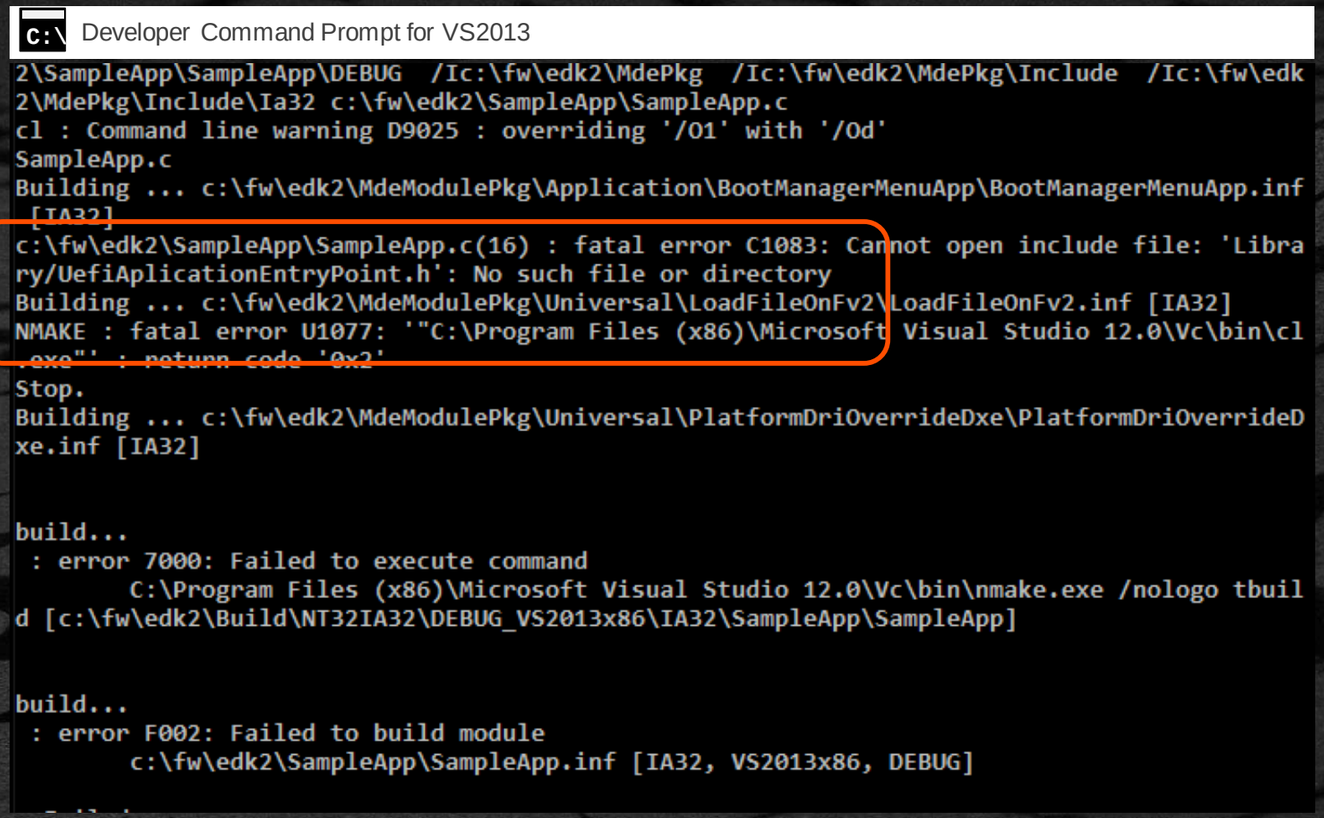
Error on SampleApp.inf



The [Packages] was invalid or did not specify MdePkg/MdePkg.dec properly

Possible Build Errors

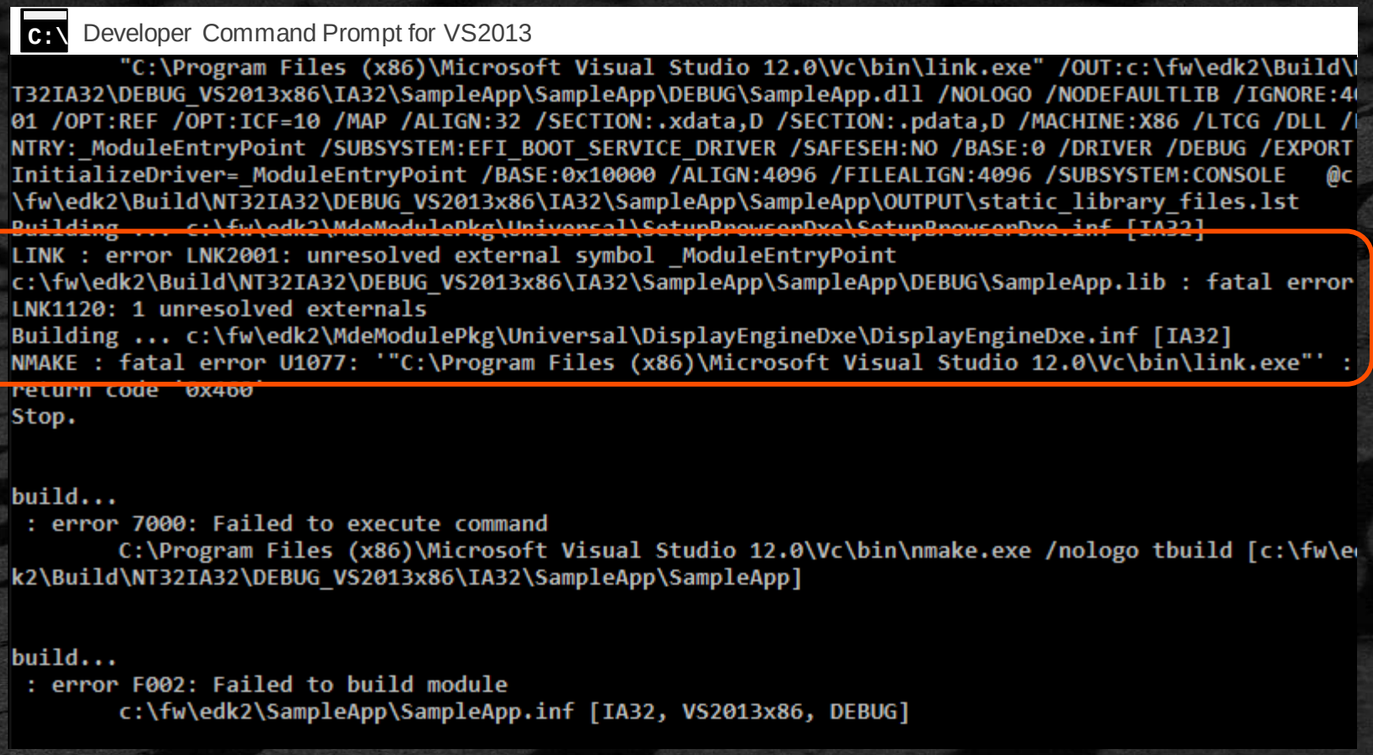
Compiler Error on SampleApp.c



The #include <Library/UefiApplicationEntryPoint.h> has a typo (“Application” not “Aplication”)

Possible Build Errors

Compile Linker Error on unresolved reference



The SampleApp.inf section `[LibraryClasses]` did not reference `UefiApplicationEntryPoint`

Possible Build Errors

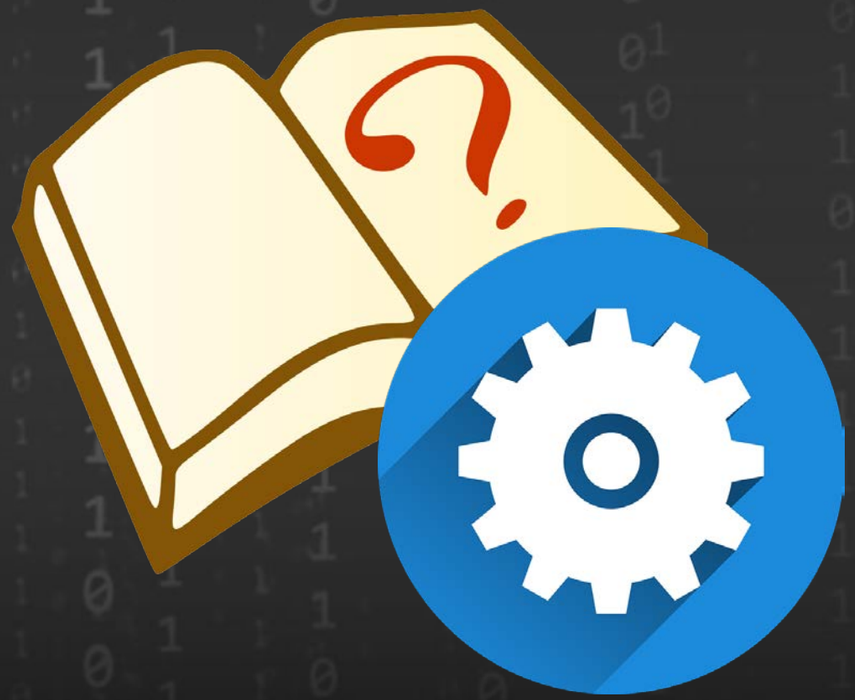
Error at the Shell prompt

```
Press ESC in 4 seconds to skip startup.nsh or any other key to continue.
2.0 Shell> SampleApp
'SampleApp' is not recognized as an internal or external command, operable progr
am, or script file.
2.0 Shell> FS0:
2.0 FS0:\> LS SampleApp.efi
Error. No matching files were found.
2.0 FS0:\> _
```

Ensure the SampleApp.inf BaseName is SampleApp

Lab 2.1: Build Switches

In this lab, you'll remove the build switch BUILD_NEW_SHELL to be always TRUE



Lab 2.1: Compiling w/out Build Switch

At the VS Command Prompt build without the –D switch

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

```
Shell> ver
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Shell> _
```

Check the Shell version with “Ver” command

Build with the –D BUILD_NEW_SHELL

```
C:/FW/edk2> Build -D BUILD_NEW_SHELL
C:/FW/edk2> Build Run
```

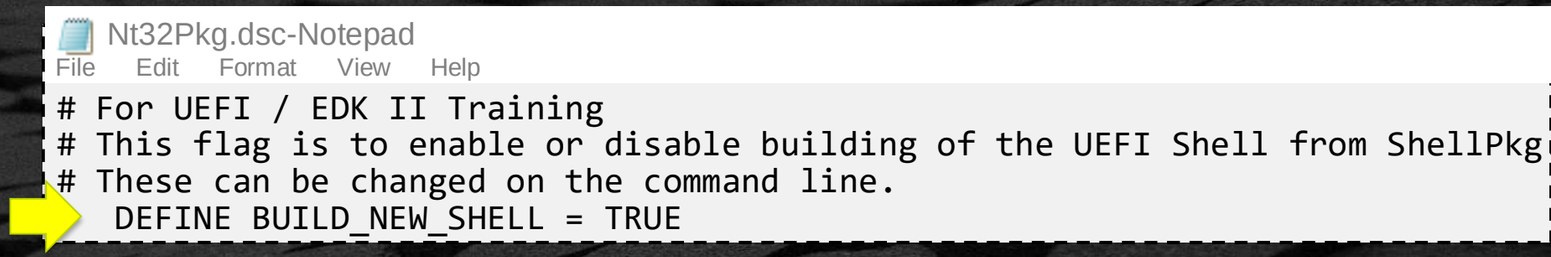
```
Shell> ver
UEFI Interactive Shell v2.2 -From BUILD_NEW_SHELL Switch
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Shell> _
```

Check the Shell version with “Ver” command

Lab 2.1: Compiling w/out Build Switch

Edit the file C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc

Change the `DEFINE BUILD_NEW_SHELL = FALSE` to “TRUE”



Re-Build – Cd to C:/FW/edk2

```
C:/FW/edk2:> build
C:/FW/edk2:> build run
```

```
Shell> ver
UEFI Interactive Shell v2.2 -From BUILD_NEW_SHELL Switch
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Shell> _
```

Check the Shell version with “Ver” command

Lab 2: If there are build errors …

See class files for the solution

- `. . .FW/LabSampleCode/LessonB.2`
- Copy the .inf and .c files to `C:/FW/edk2/SampleApp`
- Search sample DSC for reference to `SampleApp.inf` and add this line to your workspace DSC file `C:/FW/edk2/Nt32Pkg/Nt32Pkg.dsc`

```
  SampleApp/SampleApp.inf
```

Invoke “ `build`” again and check the solution

ADD FUNCTIONALITY

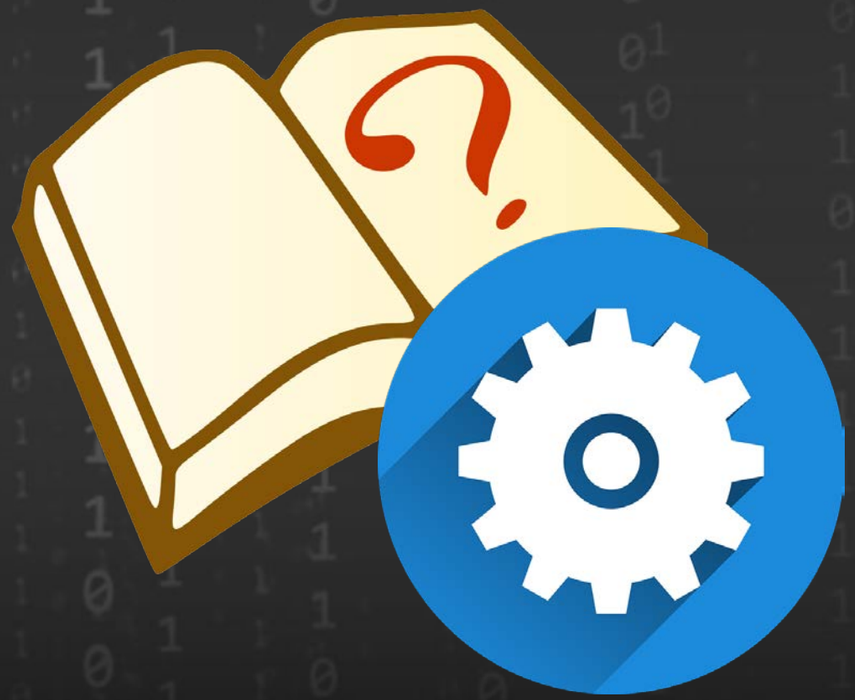
Add Functionality to the Simple UEFI Application :
Next 3 Labs

Lab 3: Print the UEFI System Table
Lab 4: Wait for an Event
Lab 5: Create a Simple Typewriter function

Solutions in .../FW/LabSampleCode/LabSolutions/LessonB.n

Lab 3: Print the UEFI System Table

Add code to print the hex address of the EFI System Table pointer to the console.



Lab 3 : Add System Table Code

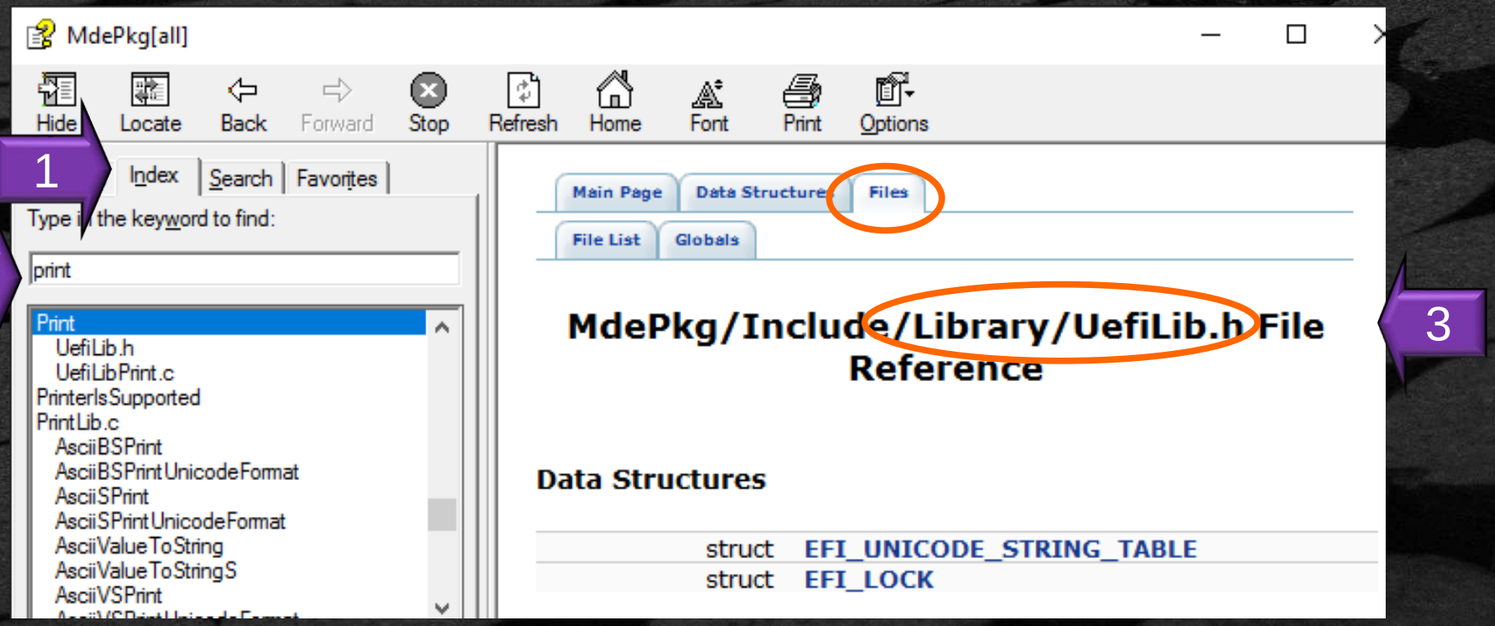
Add code to print to the console the hex address of the system table pointer

- Where is the “print” function?
- Where does the app get the pointer value?
  (compared to **`mem`** command below)

```
Valid EFI Header at Address 0000000007E34018
----------------------------------------------------
System: Table Structure size 00000078 revision 0002003C
ConIn (0000000007CB0550) ConOut (0000000006EEEB20) StdErr (0000000007CB0310)
Runtime Services 0000000007E34B98
Boot Services    0000000007EC8480
SAL System Table 0000000000000000
ACPI Table       0000000007E40000
ACPI 2.0 Table   0000000007E40014
MPS Table        0000000000000000
SMBIOS Table     0000000007E12000
Shell> sampleapp
System Table: 0x07E34018
Shell> _
```

Lab 3 : Locating the `Print()` Function

1. Search the MdePkg.chm and find that the Print function by clicking on the “Index” tab
2. Type “Print” and double click
3. Scroll to the top in the right window to see that the print function is in the UefiLib.h file



Lab 3 : Modifying .C & .INF Files

SampleApp.c - Notepad

File Edit Format View Help

SampleApp.c

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

EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
  Print(L"System Table: 0x%08x\n“, SystemTable);
  return EFI_SUCCESS;
}
```

SampleApp.inf - Notepad

File Edit Format View Help

SampleApp.inf

```
[LibraryClasses]
  UefiApplicationEntryPoint
  UefiLib
```

Note: Solution files are in the lab materials directory

Lab 3 : Modifying .C & .INF Files

- SampleApp.c

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
// Lab 3
#include <Library/UefiLib.h>

EFI_STATUS
EFIAPI
UefiMain (
 IN EFI_HANDLE ImageHandle,
 IN EFI_SYSTEM_TABLE *SystemTable
 )
{
// Lab 3
 Print(L"System Table: 0x%08x\n", SystemTable); //
 return EFI_SUCCESS;
}
```

- SampleApp.inf

```
 [LibraryClasses]
 UefiApplicationEntryPoint
# Lab 3
 UefiLib
```

Lab 3 : Build and Test SampleApp

At the VS Command Prompt

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

Run the application from the shell

```
Shell> SampleApp
System Table: 0x07E34018
Shell>
```

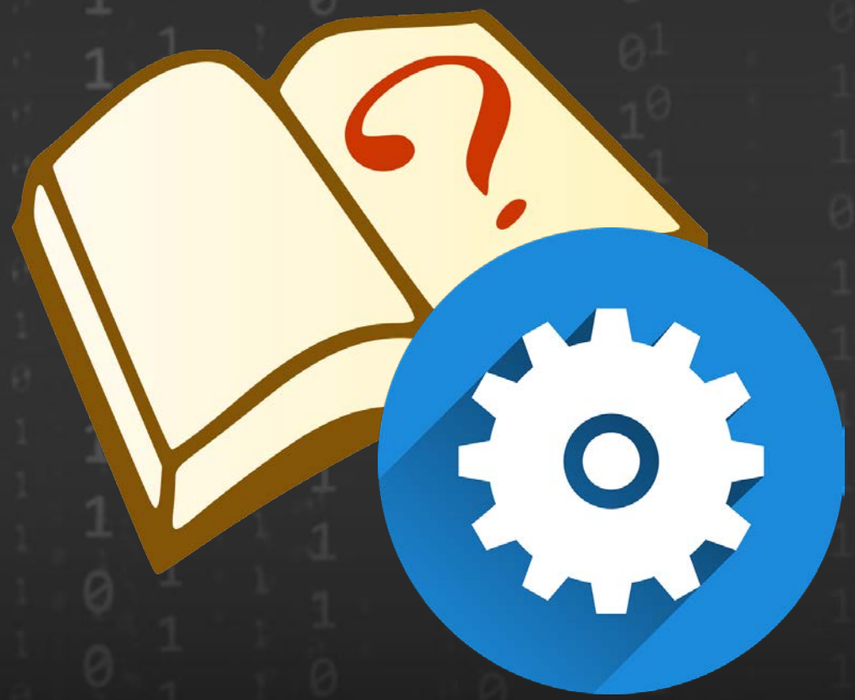
Verify by using the “mem” command

Exit

```
Shell> Reset
```

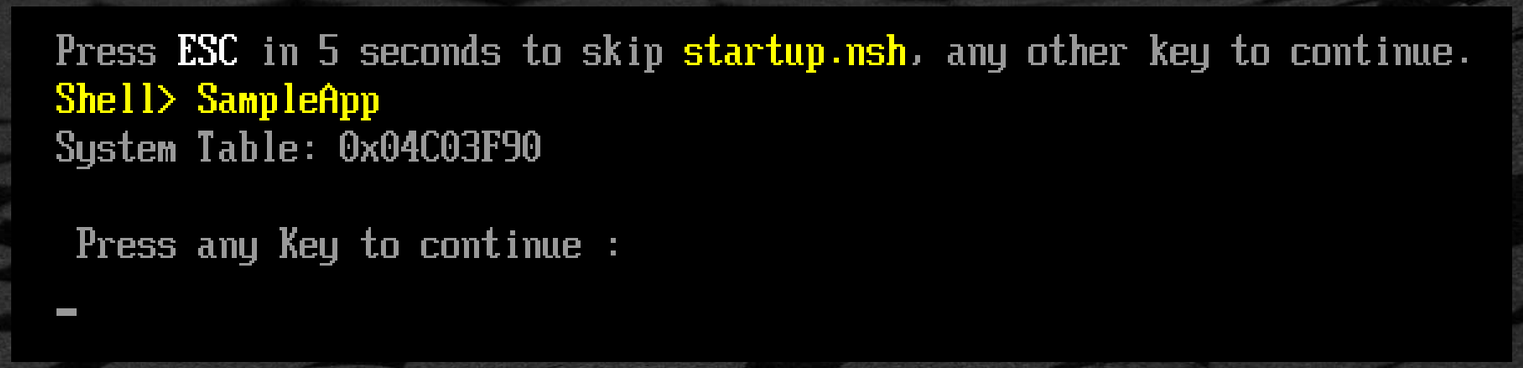
Lab 4: Waiting for an Event

In this lab, you'll learn how to locate code and .chm files to help write EFI code for waiting for an event



Lab 4 : Add Wait for Event

Add code to make your application wait for a key press event (`WaitForEvent / WaitForKey`)



- Where are these functions located?
- What else can you do with the key press?

Lab 4: HOW?

Locate Functions: `WaitForEvent / WaitForKey`

- Search MdePkg.chm- "MdePkg Document With Libraries.chm" located in ... Lab_Material_FW/FW/Documentation
  - Locate `WaitForEvent` in Boot Services
  - Locate `WaitForKey` and find (`EFI_SIMPLE_TEXT_INPUT_PROTOCOL` will be part of ConIn)
- Check the UEFI Spec for parameters needed:
  - `WaitForEvent` is referenced via Boot Services pointer, which is referenced via EFI System Table
  - `WaitForKey` can be referenced through the EFI System Table passed into the application
- **OR** Search the working space for `WaitForEvent` for an example
- One can be found in MdePkg/Library/UefiLib/Console.c ~ ln 569:

Lab 4 : Update the C File for `WaitForKey`

Search the work space and find the following MdePkg/Library/UefiLib/Console.c ~ ln 569:

Console.c - Notepad

File Edit Format View Help

```
  UINTN                     EventIndex;

  . . .

   // If we encounter error, continue to read another key in.
   //
      if (Status != EFI_NOT_READY) {
       continue;
     }
     gBS->WaitForEvent (1, &gST->ConIn->WaitForKey, &EventIndex);
   }

  . . .
```

Line 416

Line 569

Add the following to `SampleApp.c`

SampleApp.c - Notepad

File Edit Format View Help

```
UINTN                           EventIndex;
Print(L"System Table: 0x%08x",SystemTable);
Print(L"\nPress any Key to continue : \n");
gBS->WaitForEvent (1, &gST->ConIn->WaitForKey, &EventIndex);
```

Copy and Paste

Lab 4 : Update the C File for `WaitForKey`

Add the following "Lab 4" statements to SampleApp.c

```
EFI_STATUS
EFIAPI
UefiMain (
 IN EFI_HANDLE ImageHandle,
 IN EFI_SYSTEM_TABLE *SystemTable
 )
{
// Lab 4
 UINTN EventIndex;

// Lab 3
 Print(L"System Table: 0x%08x",SystemTable);

// Lab 4
 Print(L"\nPress any Key to continue : \n");
 gBS->WaitForEvent (1, &gST->ConIn->WaitForKey, &EventIndex);
```

Lab 4: Test Compile

However, this won't compile … gBS and gST are not defined.

```
Building ... c:\fw\edk2\MdeModulePkg\Application\BootManagerMenuApp\BootManagerMenuApp.inf [IA32]
c:\fw\edk2\SampleApp\SampleApp.c(42) : error C2065: 'gBS' : undeclared identifier
c:\fw\edk2\SampleApp\SampleApp.c(42) : error C2223: left of '->WaitForEvent' must point to struct/unic
n
c:\fw\edk2\SampleApp\SampleApp.c(42) : error C2065: 'gST' : undeclared identifier
c:\fw\edk2\SampleApp\SampleApp.c(42) : error C2223: left of '->ConIn' must point to struct/union
Building ... c:\fw\edk2\MdeModulePkg\Universal\LoadFileOnFv2\LoadFileOnFv2.inf [IA32]
NMAKE : fatal error U1077: '"C:\Program Files (x86)\Microsoft Visual Studio 12.0\Vc\bin\cl.exe"' : ret
urn code '0x2'
Stop.
Building ... c:\fw\edk2\MdeModulePkg\Universal\PlatformDriOverrideDxe\PlatformDriOverrideDxe.inf [IA32
]


build...
 : error 7000: Failed to execute command
        C:\Program Files (x86)\Microsoft Visual Studio 12.0\Vc\bin\nmake.exe /nologo tbuild [c:\fw\edk
2\Build\NT32IA32\DEBUG_VS2013x86\IA32\SampleApp\SampleApp]
```

Search the MdePkg.chm for "**gBS**" and "**gST**" – they are located in **UefiBootServicesTableLib.h**

Add the boot services lib to **SampleApp.c** …
#include <Library/UefiBootServicesTableLib.h>

(hint: **Lesson B.4** has the solution)

Lab 4: Update for gBS & gST

SampleApp.c - Notepad

File Edit Format View Help

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
#include <Library/UefiLib.h>
#include <Library/UefiBootServicesTableLib.h>
// . . .
EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
  UINTN                EventIndex;
  Print(L"System Table: 0x%08x\n“, SystemTable);
  Print(L"\nPress any Key to  continue :\n");
  gBS->WaitForEvent (1, &gST->ConIn->WaitForKey, &EventIndex);
  return EFI_SUCCESS;
}
```

Lab 4: Update for gBS & gST

SampleApp.c Should have the following for Lab 4:

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
#include <Library/UefiLib.h>
// Lab 4
#include <Library/UefiBootServicesTableLib.h>

EFI_STATUS
EFIAPI
UefiMain (
 IN EFI_HANDLE        ImageHandle,
 IN EFI_SYSTEM_TABLE *SystemTable
 )
{
 UINTN           EventIndex;

// Lab 3
 Print(L"System Table: 0x%08x\n",SystemTable);

//Lab 4
 Print( L"\nPress any Key to continue : \n\n");
 gBS->WaitForEvent (1, &gST->ConIn->WaitForKey,        &EventIndex);
 return EFI_SUCCESS;
}
```

Lab 4 : Build and Test SampleApp

At the VS Command Prompt

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

Run the application from the shell

```
Shell> SampleApp
System Table: 0x07E34018
Press any key to continue:
Shell>
```

Notice that the SampleApp will wait until a key press to continue

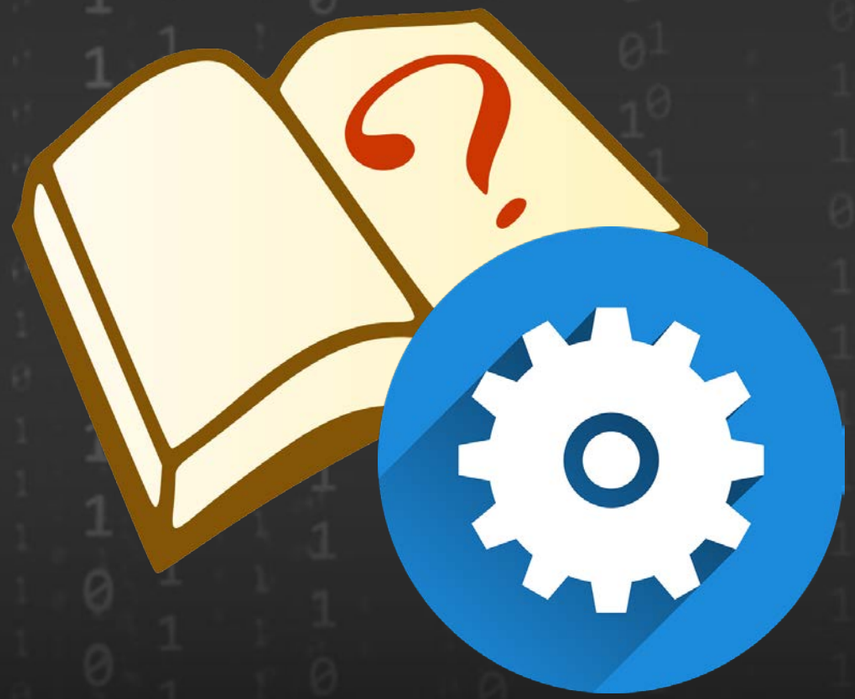
Exit

```
Shell> Reset
```

tianocore

Lab 5: Creating a Simple Typewriter Function

In this lab, you'll learn how to create a simple typewriter function that retrieves the keys you type and subsequently prints each one back to the console



Lab 5 : Typewriter Function

Create a Simple Typewriter Function using the SampleApp from Lab 4

Requirements:

- Retrieve keys entered from keyboard (*Like* Lab 4)
- Print back each key entered to the console
- To exit, press “.” (DOT) and then <Enter>

Lab 5 : Typewriter Function

Create a Simple Typewriter Function using the SampleApp from Lab 4

How:

1. Add a Loop using `WaitForEvent` with `WaitForKey`
2. Use the `ReadKeyStroke` function from `ConIn`
3. Print back each key to console
4. Exit when DOT “.” character is followed by an <Enter> key

Lab 5: How Process (Hints)

- Use the same procedure as with Lab 4 to find "ReadKeyStroke" in the work space: MdePkg/Library/UefiLib/Console.c ~ ln 558

```
Status = gST->ConIn->ReadKeyStroke (gST->ConIn, Key);
```

- ReadKeyStroke uses buffer called EFI_INPUT_KEY ~ ln 399

```
OUT EFI_INPUT_KEY  *Key,
```

- TIP: Good Idea to zero out a buffer in your function –
  - Use MdePkg.chm to find ZeroMem function
  - Use ZeroMem on your variable buffer "Key" of type EFI_INPUT_KEY
- Use Boolean flag "ExitLoop" to exit your loop once the user enters a DOT "." character.

Lab 5: Solution

SampleApp.c - Notepad

File Edit Format View Help

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
#include <Library/UefiLib.h>
#include <Library/BaseMemoryLib.h>
#include <Library/UefiBootServicesTableLib.h>
#define CHAR_DOT  0x002E    // '.' in Unicode

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

// Lab 3
 Print(L"System Table: 0x%08x\n",SystemTable);

//Lab 4
 Print( L"\nPress any Key to continue : \n\n");
 gBS->WaitForEvent (1, &gST->ConIn->WaitForKey,EventIndex);
```

(hint: Lesson B.5 has the solution)

```
// Lab 5
 Print(L"Enter text. Include a dot ('.') in a \
     sentence then <Enter> to exit:\n\n”);
 ZeroMem (&Key, sizeof (EFI_INPUT_KEY));
 gST->ConIn->ReadKeyStroke (gST->ConIn, &Key);
 ExitLoop = FALSE;
 do {
       gBS->WaitForEvent (1, &gST->ConIn->WaitForKey,
                  &EventIndex);
       gST->ConIn->ReadKeyStroke (gST->ConIn, &Key);
       Print(L"%c", Key.UnicodeChar);
       if (Key.UnicodeChar == CHAR_DOT){
            ExitLoop = TRUE;
       }
    } while (!(Key.UnicodeChar == CHAR_LINEFEED  ||
       Key.UnicodeChar == CHAR_CARRIAGE_RETURN) ||
       !(ExitLoop) );

 Print(L"\n");
 return EFI_SUCCESS;
}
```

Lab 5: Solution

SampleApp.c Should have the following for Lab 5:

```
#include <Uefi.h>
#include <Library/UefiApplicationEntryPoint.h>
#include <Library/UefiLib.h>
#include <Library/UefiBootServicesTableLib.h>
#include <Library/BaseMemoryLib.h>
#define CHAR_DOT  0x002E    // '.' in Unicode

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

// Lab 3
 Print(L"System Table: 0x%08x\n",SystemTable);

//Lab 4
 Print( L"\nPress any Key to continue : \n\n");
 gBS->WaitForEvent (1, &gST->ConIn->WaitForKey,
```

```
// Lab 5
 Print(L"Enter text. Include a dot ('.') in a sentence then
<Enter> to exit:\n\n”);
 ZeroMem (&Key, sizeof (EFI_INPUT_KEY));
 gST->ConIn->ReadKeyStroke (gST->ConIn, &Key);
 ExitLoop = FALSE;
 do {
       gBS->WaitForEvent (1, &gST->ConIn-
>WaitForKey,&EventIndex);
       gST->ConIn->ReadKeyStroke (gST->ConIn, &Key);
       Print(L"%c", Key.UnicodeChar);
       if (Key.UnicodeChar == CHAR_DOT){
            ExitLoop = TRUE;
       }
    } while (!(Key.UnicodeChar == CHAR_LINEFEED  ||
       Key.UnicodeChar == CHAR_CARRIAGE_RETURN) ||
       !(ExitLoop) );

 Print(L"\n");
 return EFI_SUCCESS;
}
```

Lab 5 :Build and Test SampleApp

At the VS Command Prompt

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

Run the application from the shell

```
Shell> sampleapp
System Table: 0x061CBF90


 Press any Key to continue :
Enter text. Include a dot ('.') in a sentence then <Enter> to exit:

This is text from the type writer function.
Shell> _
```

Exit

```
Shell> Reset
```

Bonus Exercise: Open Protocol Example

Write an Application using **`argv, argc`** parameters

- Captures command line parameters using Open Protocol
- Need to open SHELL_INTERFACE_PROTOCOL
- Note: Requires ShellPkg

Build SampleApp

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

Run the application form the shell

```
Shell> SampleApp  test1 test2
```

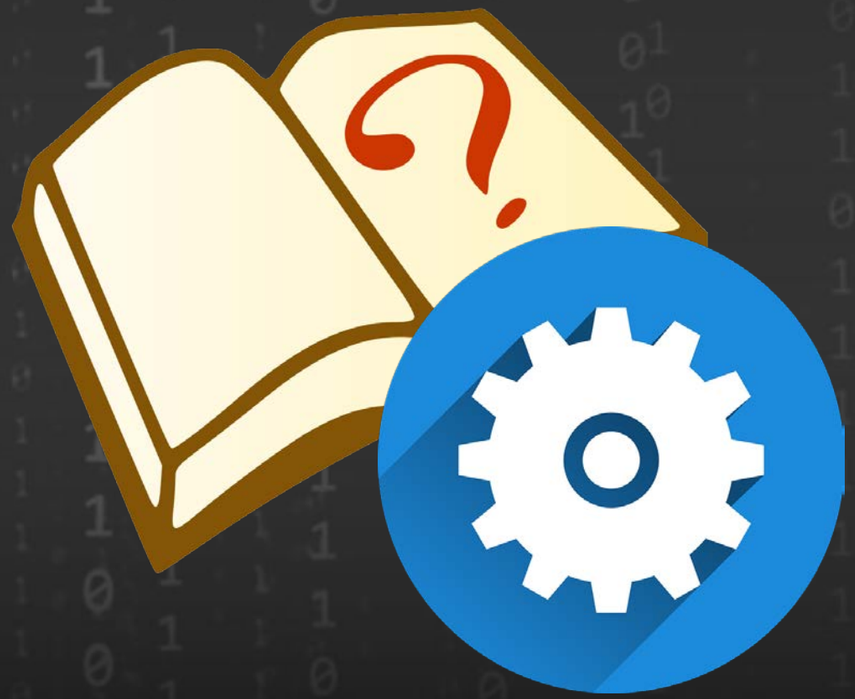
(hint: ~FW/LabSampleCode/ShellAppSample has the solution)

USING EADK

Using EADK with UEFI Application

Lab 6: Writing UEFI Applications with EADK

In this lab, you'll write an application with the same functionality as SampleApp.c using LibC from the EDK II Application Development Kit (EADK)



Lab 6: With EDK II EADK

Write the same application with the same functionality as SampleApp.c using the LibC from the EADK

```
Shell> fs0:
FS0:\> SampleCApp
System Table: 0x631bf90

Press any Key and then <Enter> to continue :


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

This is a sentence using my UEFI Application using the C library.


FS0:\> _
```

What libraries are needed

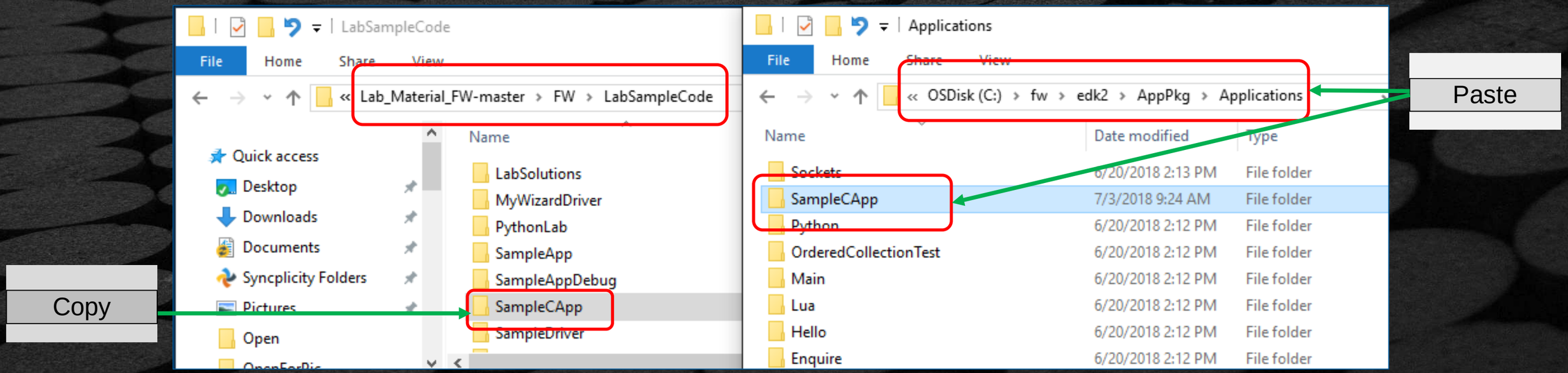
What differences are there using the LibC

Lab 6: EDK II using EADK

Start with the packages for EADK

- /edk2 - AppPkg - has directory Applications
- /edk2 - StdLib - contains the LibC libraries

- Copy and paste directory ../**FW/LabSampleCode/SampleCApp** to **C:/FW/edk2/AppPkg/Applications/SampleCApp**



Lab 6: EDK II using EADK

Check out AppPkg/Applications/SampleCApp

SampleCApp.c and SampleCApp.inf

SampleCApp.c - Notepad

File Edit Format View Help

```
#include <stdio.h>
// . . .
int
main (
  IN int Argc,
  IN char **Argv
)
{
   return 0;
}
```

SampleCApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION            = 1.25
  BASE_NAME              = SampleCApp
  FILE_GUID              = 4ea9…
  MODULE_TYPE            = UEFI_APPLICATION
  VERSION_STRING         = 0.1
  ENTRY_POINT            = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
```

Lab 6: EDK II using EADK

SampleCApp.c and SampleCApp.inf

“C” file

```
#include <stdio.h>
   // . . .
   int
   main (
     IN int Argc,
     IN char **Argv
   )
   {
      return 0;
   }
```

.inf file

```
[Defines]
  INF_VERSION    = 1.25
  BASE_NAME      = SampleCApp
  FILE_GUID      = 54321...
  MODULE_TYPE    = UEFI_APPLICATION
  VERSION_STRING = 0.1
  ENTRY_POINT    = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
```

Lab 6 : Update AppPkg.dsc

Edit the AppPkg/AppPkg.dsc and add SampleCApp.inf at the end of the components section

- (hint: search for "##### Sample Applications")
- AppPkg/Applications/SampleCApp/SampleCApp.inf

```
[Components]
#### Sample Applications.
AppPkg/Applications/Hello/Hello.inf             # No LibC includes or functions.
AppPkg/Applications/Main/Main.inf               # Simple invocation. No other LibC function
AppPkg/Applications/Enquire/Enquire.inf         #
AppPkg/Applications/ArithChk/ArithChk.inf       #

AppPkg/Applications/SampleCApp/SampleCApp.inf  # LAB 6
```

Lab 6 :Build and Test SampleCApp

Build the AppPkg at the VS Command Prompt

```
C:/FW/edk2> build -p AppPkg/AppPkg.dsc –m AppPkg/Applications/SampleCApp/SampleCApp.inf
```

Copy the built application to the Nt32 runtime directory (note VS Tool)

```
C:/FW/edk2> copy Build/AppPkg/DEBUG_VS2013x86/IA32/SampleCApp.efi
                 Build/Nt32IA32/DEBUG_VS2013x86/IA32
```

Run Nt32 emulation

```
C:/FW/edk2> build run
```

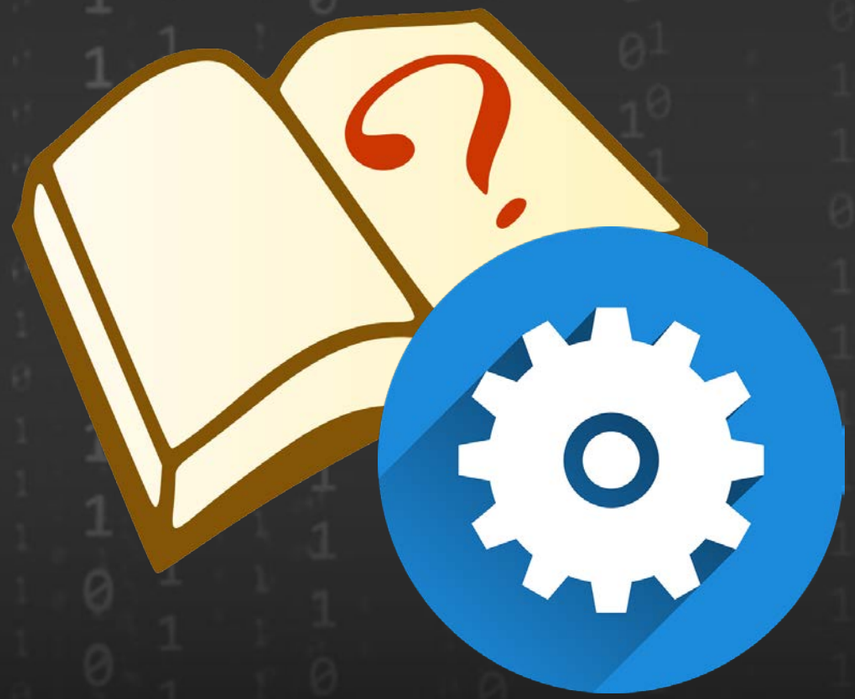
Run the application SampleCApp from the Shell

```
Shell> SampleCApp
Shell>
```

Notice that the program will immediately unload because the main function is empty

Lab 7: Adding Functionality to SampleCApp

In this lab, you'll add functionality to SampleCApp the same as in Lab 5. This lab will use EADK libraries so the coding style is similar to standard C.



Lab 7: Add the same functionally from Lab 5

SampleCApp.c and SampleCApp.inf

SampleCApp.c - Notepad

File Edit Format View Help

```
#include <stdio.h>
#include <Library/UefiBootServicesTableLib.h>
// . . .
   char c;

   printf("System Table: %p \n", gST) ;
   puts("Press any Key and then <Enter>
         to continue : ");
   c=(char)getchar();
   puts ("Enter text. Include a dot ('.') in a
         sentence then <Enter> to exit:");
   do {
      c=(char)getchar();
      } while (c != '.');
   puts ("\n");

   return 0;
 }
```

SampleCApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION            = 1.25
  BASE_NAME              = SampleCApp
  FILE_GUID              = 4ea9…
  MODULE_TYPE            = UEFI_APPLICATION
  VERSION_STRING         = 0.1
  ENTRY_POINT            = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
  UefiBootServicesTableLib
```

Lab 7: Add the same functionally from Lab 5

SampleCApp.c and SampleCApp.inf

SampleCApp.c - Notepad

File Edit Format View Help

```
#include <stdio.h>
#include <Library/UefiBootServicesTableLib.h>
// . . .
   char c;

   printf("System Table: %p \n", gST) ;
   puts("Press any Key and then <Enter>
         to continue : ");
   c=(char)getchar();
   puts ("Enter text. Include a dot ('.') in a
         sentence then <Enter> to exit:");
   do {
      c=(char)getchar();
      } while (c != '.');
   puts ("\n");

   return 0;
}
```

3

SampleCApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION            = 1.25
  BASE_NAME              = SampleCApp
  FILE_GUID              = 4ea9…
  MODULE_TYPE            = UEFI_APPLICATION
  VERSION_STRING         = 0.1
  ENTRY_POINT            = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
  UefiBootServicesTableLib
```

Lab 7: Add the same functionally from Lab 5

SampleCApp.c and SampleCApp.inf

SampleCApp.c - Notepad

File Edit Format View Help

```
#include <stdio.h>
#include <Library/UefiBootServicesTableLib.h>
// . . .
   char c;

   printf("System Table: %p \n", gST) ;
   puts("Press any Key and then <Enter>
         to continue : ");
   c=(char)getchar();
   puts ("Enter text. Include a dot ('.') in a
         sentence then <Enter> to exit:");
   do {
      c=(char)getchar();
      } while (c != '.');
   puts ("\n");

   return 0;
}
```

3

4

SampleCApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION            = 1.25
  BASE_NAME              = SampleCApp
  FILE_GUID              = 4ea9…
  MODULE_TYPE            = UEFI_APPLICATION
  VERSION_STRING         = 0.1
  ENTRY_POINT            = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
  UefiBootServicesTableLib
```

Lab 7: Add the same functionally from Lab 5

SampleCApp.c and SampleCApp.inf

SampleCApp.c - Notepad

File Edit Format View Help

```
#include <stdio.h>
#include <Library/UefiBootServicesTableLib.h>
// . . .
   char c;

   printf("System Table: %p \n", gST) ;
   puts("Press any Key and then <Enter>
         to continue : ");
   c=(char)getchar();
   puts ("Enter text. Include a dot ('.') in a
         sentence then <Enter> to exit:");
   do {
      c=(char)getchar();
      } while (c != '.');
   puts ("\n");

   return 0;
 }
```

3

4

5

SampleCApp.inf - Notepad

File Edit Format View Help

```
[Defines]
  INF_VERSION            = 1.25
  BASE_NAME              = SampleCApp
  FILE_GUID              = 4ea9…
  MODULE_TYPE            = UEFI_APPLICATION
  VERSION_STRING         = 0.1
  ENTRY_POINT            = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
  UefiBootServicesTableLib
```

Lab 7: Solution

SampleCApp.c and SampleCApp.inf

"C" file

```
#include <stdio.h>
#include <Library/UefiBootServicesTable
// . . .
   char c;

   printf("System Table: %p \n", gST) ;
   puts("Press any Key and then <Enter> to continue :
   c=(char)getchar();
   puts ("Enter text. Include a dot ('.') in a
   do {
      c=(char)getchar();
      } while (c != '.');
   puts ("\n");

   return 0;
 }
```

.inf file

```
[Defines]
  INF_VERSION    = 1.25
  BASE_NAME      = SampleCApp
  FILE_GUID      = 4ea9…
  MODULE_TYPE    = UEFI_APPLICATION
  VERSION_STRING = 0.1
  ENTRY_POINT    = ShellCEntryLib

[Sources]
  SampleCApp.c

[Packages]
  StdLib/StdLib.dec
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec

[LibraryClasses]
  LibC
  LibStdio
  UefiBootServicesTableLib
```

Lab 7 :Build and Test SampleCApp

Build the AppPkg

```
bash$ build -p AppPkg/AppPkg.dsc –m AppPkg/Applications/SampleCApp/SampleCApp.inf
```

Copy the built application to the run OVMF hda-contents directory

```
bash$ cp Build/AppPkg/DEBUG_GCC5/X64/SampleCApp.efi ~/run-ovmf/hda-contents
```

Test by Invoking Qemu

```
bash$ cd ~/run-ovmf
bash$ . RunQemu.sh
```

Run the application from the New Shell

```
Shell> SampleCApp
Press any Key and then <Enter> to Continue :

Enter text. Include a dot (‘.’) in a sentence then <Enter> to exit:
This is sample text.
Shell>
```

Lab 6 :Build and Test SampleCApp

Build the AppPkg at the VS Command Prompt

```
C:/FW/edk2> build -p AppPkg/AppPkg.dsc –m AppPkg/Applications/SampleCApp/SampleCApp.inf
```

Copy the built application to the Nt32 runtime directory (note VS Tool)

```
C:/FW/edk2> copy Build/AppPkg/DEBUG_VS2013x86/IA32/SampleCApp.efi
                 Build/Nt32IA32/DEBUG_VS2013x86/IA32
```

Run Nt32 emulation

```
C:/FW/edk2> build run
```

Run the application SampleCApp from the Shell

```
Shell> SampleCApp
Press any Key and then <Enter> to Continue :

Enter text. Include a dot (‘.’) in a sentence then <Enter> to exit:
This is sample text.
Shell>
```

tianocore

SUMMARY

- UEFI Application with PCDs
- Simple UEFI Application
- Add functionality to UEFI Application
- Using EADK with UEFI Application


Questions?


tianocore