UEFI & EDK II Training

UEFI Driver Wizard Lab

tianocore.org

LESSON OBJECTIVE

- Setup the UEFI Driver Wizard
- Create a UEFI Driver Template

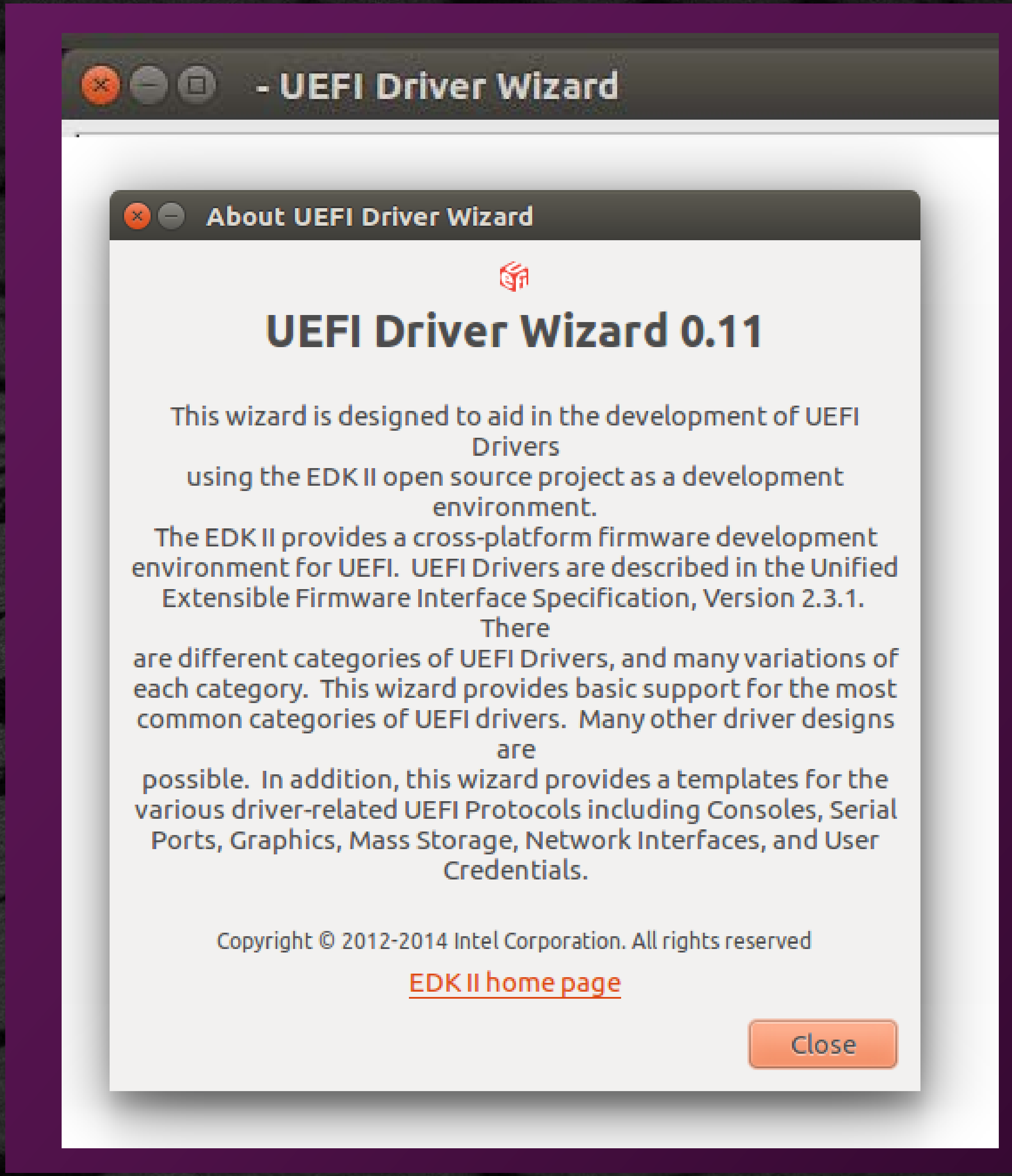
tianocore

UEFI DRIVER WIZARD

Creating a Template UEFI Driver with the UEFI Driver Wizard

UEFI Driver Wizard Overview

- ✓ Open source tool
- ✓ Based on *Driver Writer's Guide for UEFI 2.3.1* content
- ✓ Intel SSG engineers contributed
- ✓ Located on www.TianoCore.org



UEFI Driver Wizard 0.11

This wizard is designed to aid in the development of UEFI Drivers
using the EDK II open source project as a development environment.
The EDK II provides a cross-platform firmware development environment for UEFI. UEFI Drivers are described in the Unified Extensible Firmware Interface Specification, Version 2.3.1. There
are different categories of UEFI Drivers, and many variations of each category. This wizard provides basic support for the most common categories of UEFI drivers. Many other driver designs are
possible. In addition, this wizard provides a templates for the various driver-related UEFI Protocols including Consoles, Serial Ports, Graphics, Mass Storage, Network Interfaces, and User Credentials.

Copyright © 2012-2014 Intel Corporation. All rights reserved

EDK II home page

Installing Python for UEFI Driver Wizard

Requirements and Options

- Work space must contain BaseTools, MdePkg & MdeModulePkg Packages from UDK2017 for Driver development on Tianocore.org

- Uses previous lab's setup **`$HOME/src/edk2`**

- Python* scripts from Github Link then use instructions from README for Python and wxPython versions to install then run

```
bash$ python launch.py
```

Requirements for Your Driver

- UEFI Device Driver
- UEFI Version 2.7 (0x00020046)

```
#define EFI_2_70_SYSTEM_TABLE_REVISION ((2<<16) | (70DEC))
```

- Unloadable driver
- Support IA32 & x64 CPUs
- Returns component name information
- Test console device
- Option to produce strings & forms for setup

Template File Contents

Proper UEFI driver entry point

Template File Contents

Proper UEFI driver entry point

Basic driver libraries/headers

Template File Contents

Proper UEFI driver entry point

Basic driver libraries/headers

Skeletons for common driver functions

Template File Contents

Proper UEFI driver entry point

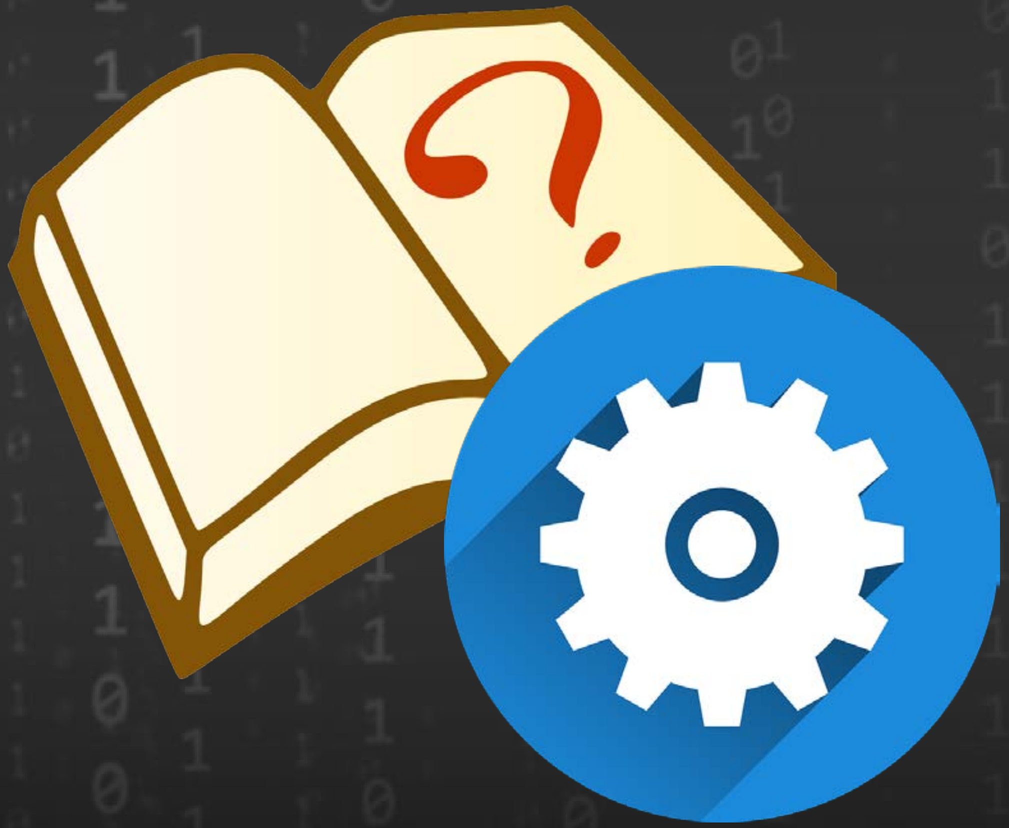
Basic driver libraries/headers

Skeletons for common driver functions

Error values until ported
EFI_UNSUPPORTED, EFI_DEVICE_ERROR

Lab 1: Create a UEFI Driver with the UEFI Driver Wizard

- In this lab, you'll create a new UEFI driver using the UEFI Driver Wizard.
- This will create a set of "c" code files to be used as a template UEFI Driver used in the subsequent driver labs



Lab 1: Install UEFI Driver Wizard, Python & wxPython

1. Perform Lab Setup from previous Labs

2. From the ~FW/DriverWizard folder, copy and paste folder “~FW/DriverWizard/UefiDriverWizard” to ~$Home

3. Check if version 2.7.x is the default of Python from Terminal Prompt

```
   bash$ python –V
```

```
  Python 2.7.12
```

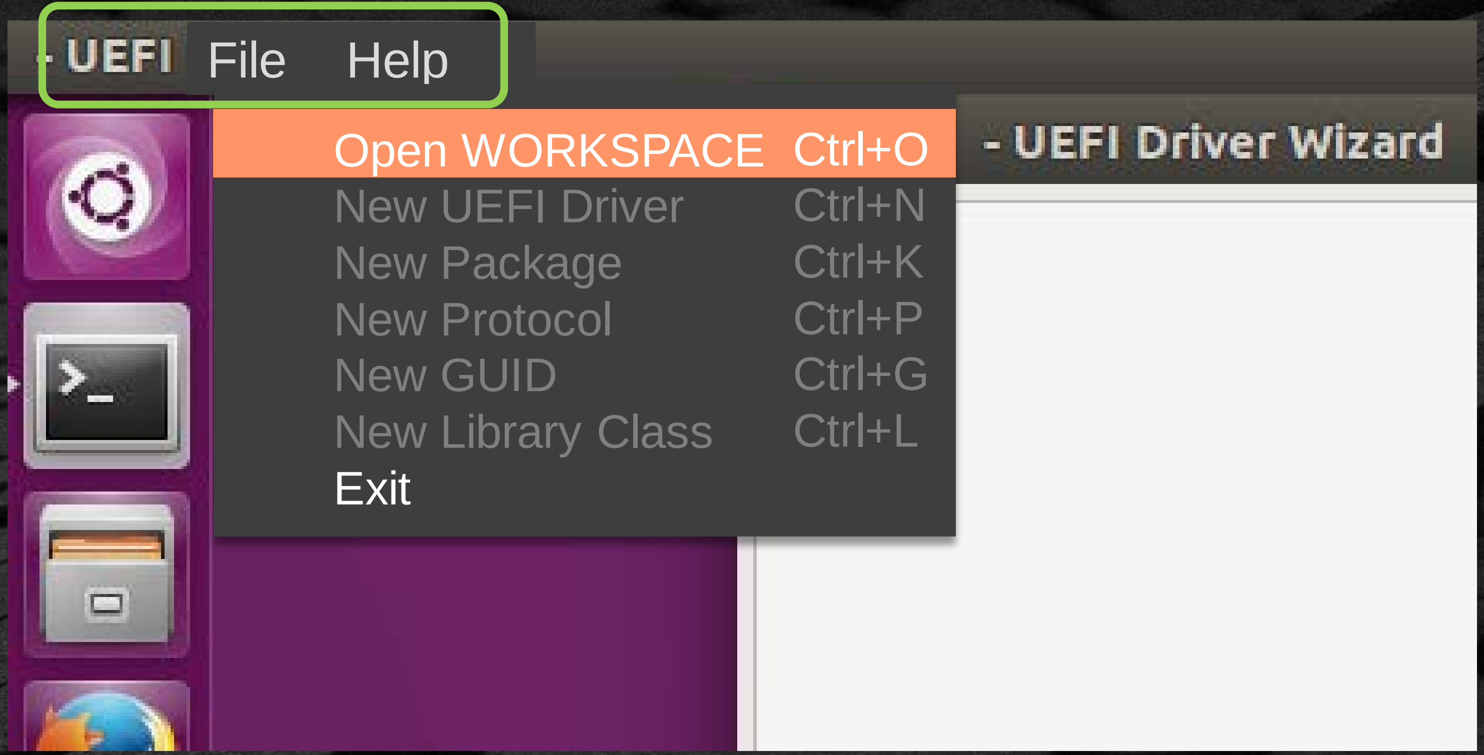
3. Install the wxPython (Version 3.0)

```
 bash$ sudo apt-get install python-wxgtk3.0
```

Lab 1: UefiDriverWizard -Select Work Space

Terminal Prompt (Cnt-Alt-T)

```
bash$ cd ~UefiDriverWizard
bash$ python launch.py
```

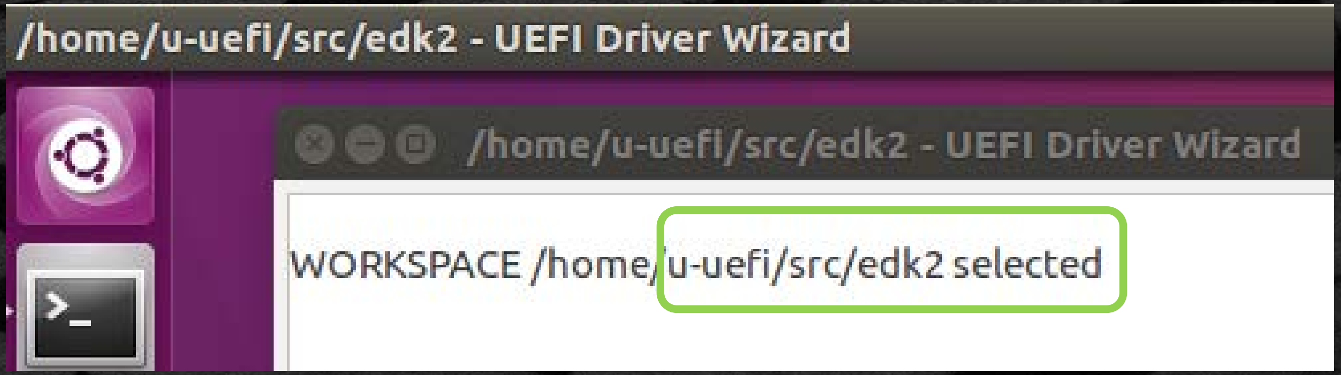


Select a Work Space

Control+O – to browse for a directory

Browse to ~src/edk2

Select Open

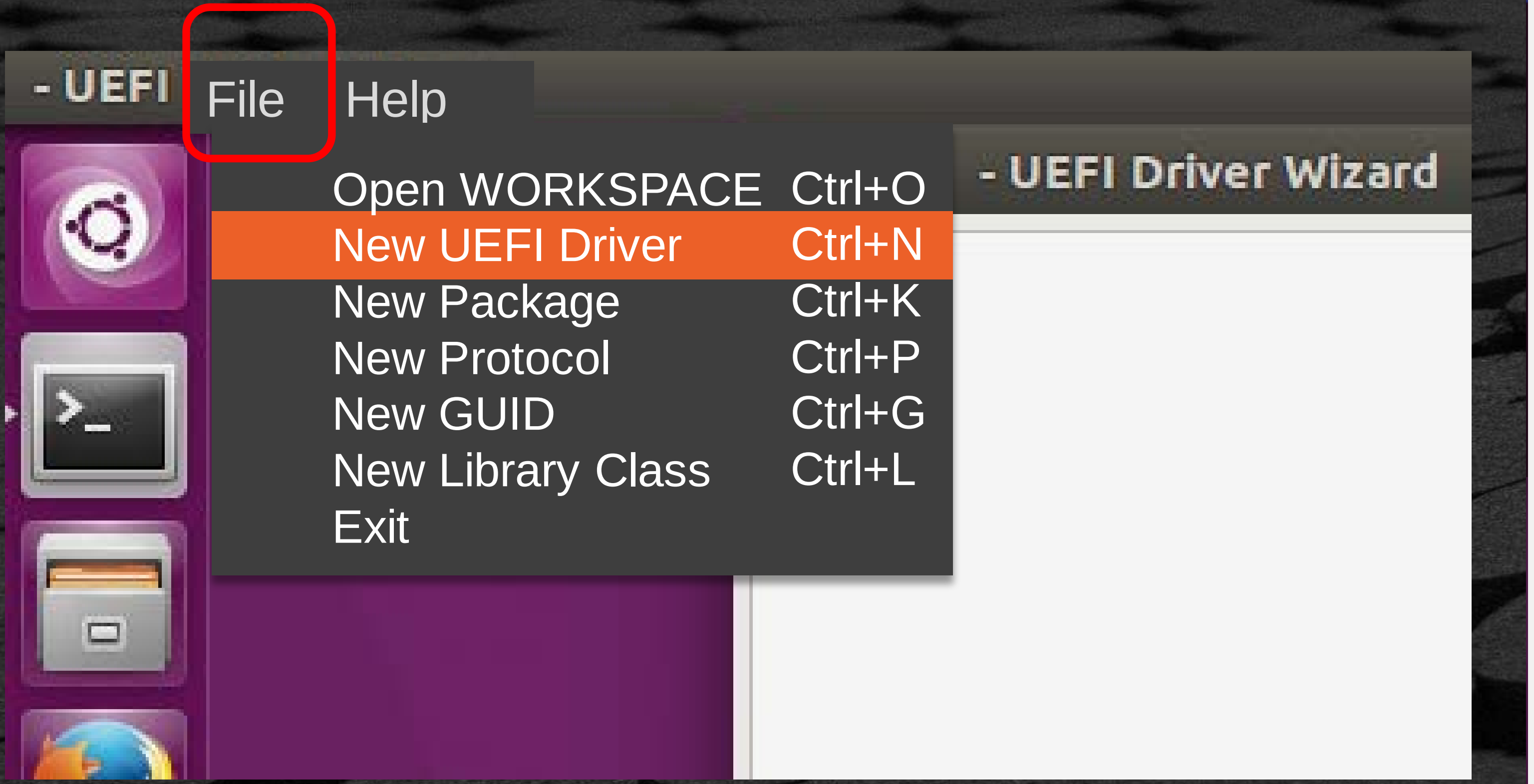

/home/u-uefi/src/edk2 - UEFI Driver Wizard
/home/u-uefi/src/edk2 - UEFI Driver Wizard

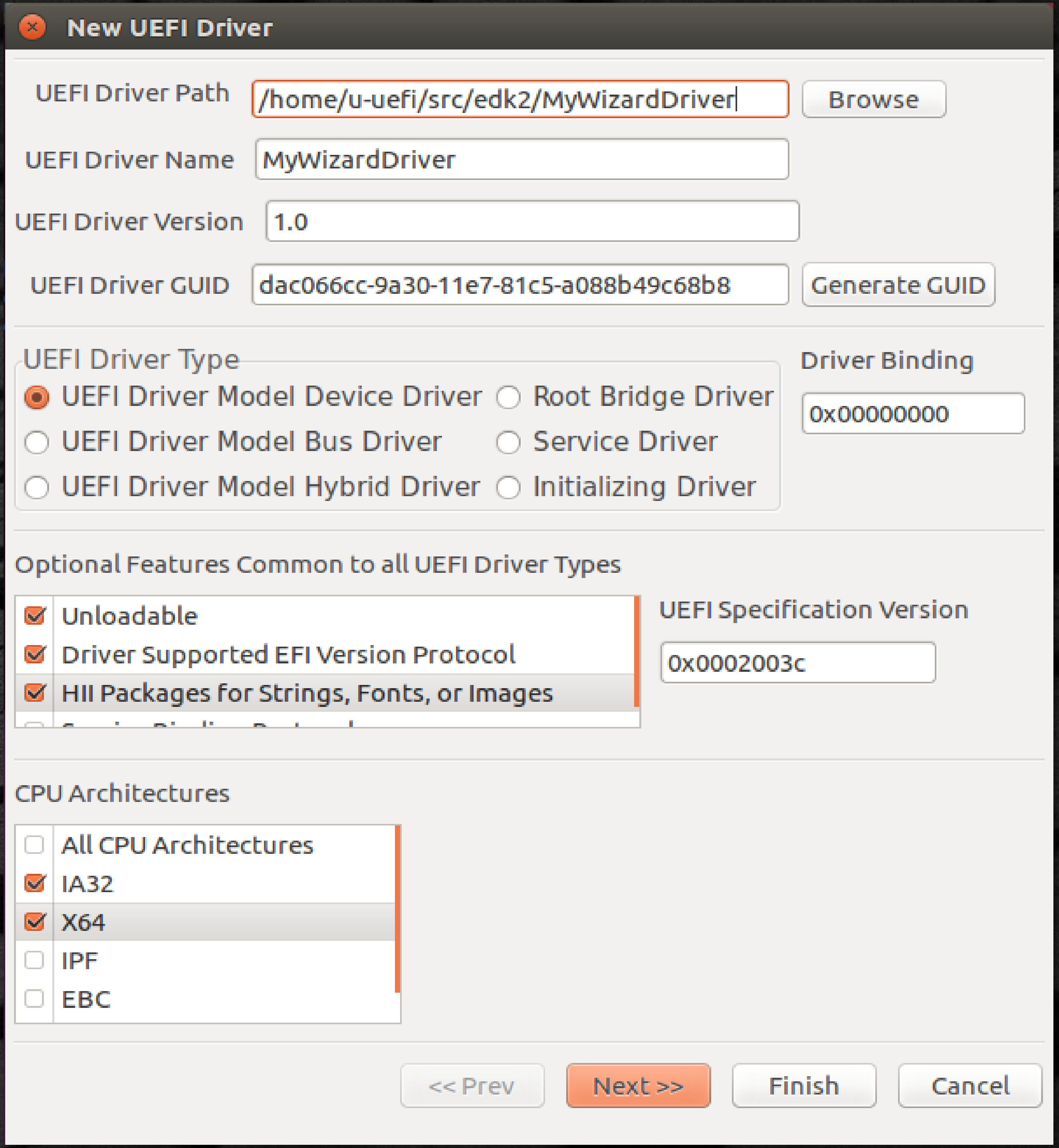
```
WORKSPACE /home/u-uefi/src/edk2 selected
```

Note: the environment for EDK II must be setup with edksetup.sh

Lab 1: Create a New UEFI Driver

Control+N – to Open Menu





New UEFI Driver

UEFI Driver Path: /home/u-uefi/src/edk2/MyWizardDriver

UEFI Driver Name: MyWizardDriver

UEFI Driver Version: 1.0

UEFI Driver GUID: dac066cc-9a30-11e7-81c5-a088b49c68b8

UEFI Driver Type
- ● UEFI Driver Model Device Driver
- ○ UEFI Driver Model Bus Driver
- ○ UEFI Driver Model Hybrid Driver
- ○ Root Bridge Driver
- ○ Service Driver
- ○ Initializing Driver

Driver Binding: 0x00000000

Optional Features Common to all UEFI Driver Types
- ☑ Unloadable
- ☑ Driver Supported EFI Version Protocol
- ☑ HII Packages for Strings, Fonts, or Images

UEFI Specification Version: 0x0002003c

CPU Architectures
- □ All CPU Architectures
- ☑ IA32
- ☑ X64
- □ IPF
- □ EBC

Lab 1: New UEFI Driver Menu

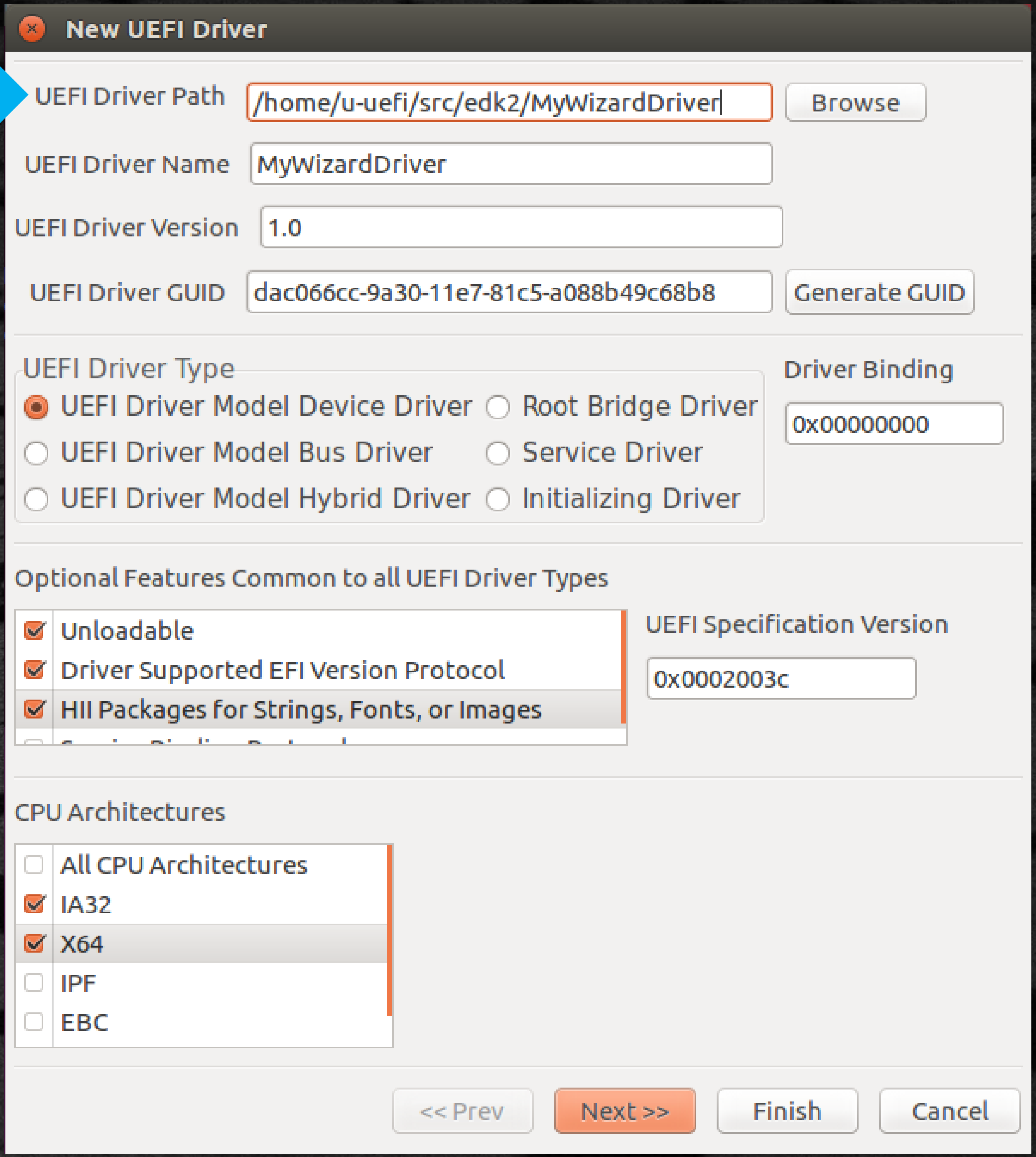
- UEFI Driver Path" – Type: "**MyWizardDriver**"

  Note: "UEFI Driver Name" is filled in.

- **Ensure** all the forms, radio buttons, and boxes are filled in and **selected *exactly*** like the image to the right.

- ***Note***: A new, specific driver GUID will populate, so it will be different than this image

Click Next >>



New UEFI Driver

UEFI Driver Path: /home/u-uefi/src/edk2/MyWizardDriver

UEFI Driver Name: MyWizardDriver

UEFI Driver Version: 1.0

UEFI Driver GUID: dac066cc-9a30-11e7-81c5-a088b49c68b8

UEFI Driver Type

● UEFI Driver Model Device Driver
○ Root Bridge Driver
○ UEFI Driver Model Bus Driver
○ Service Driver
○ UEFI Driver Model Hybrid Driver
○ Initializing Driver

Driver Binding: 0x00000000

Optional Features Common to all UEFI Driver Types

☑ Unloadable
☑ Driver Supported EFI Version Protocol
☑ HII Packages for Strings, Fonts, or Images

UEFI Specification Version: 0x0002003c

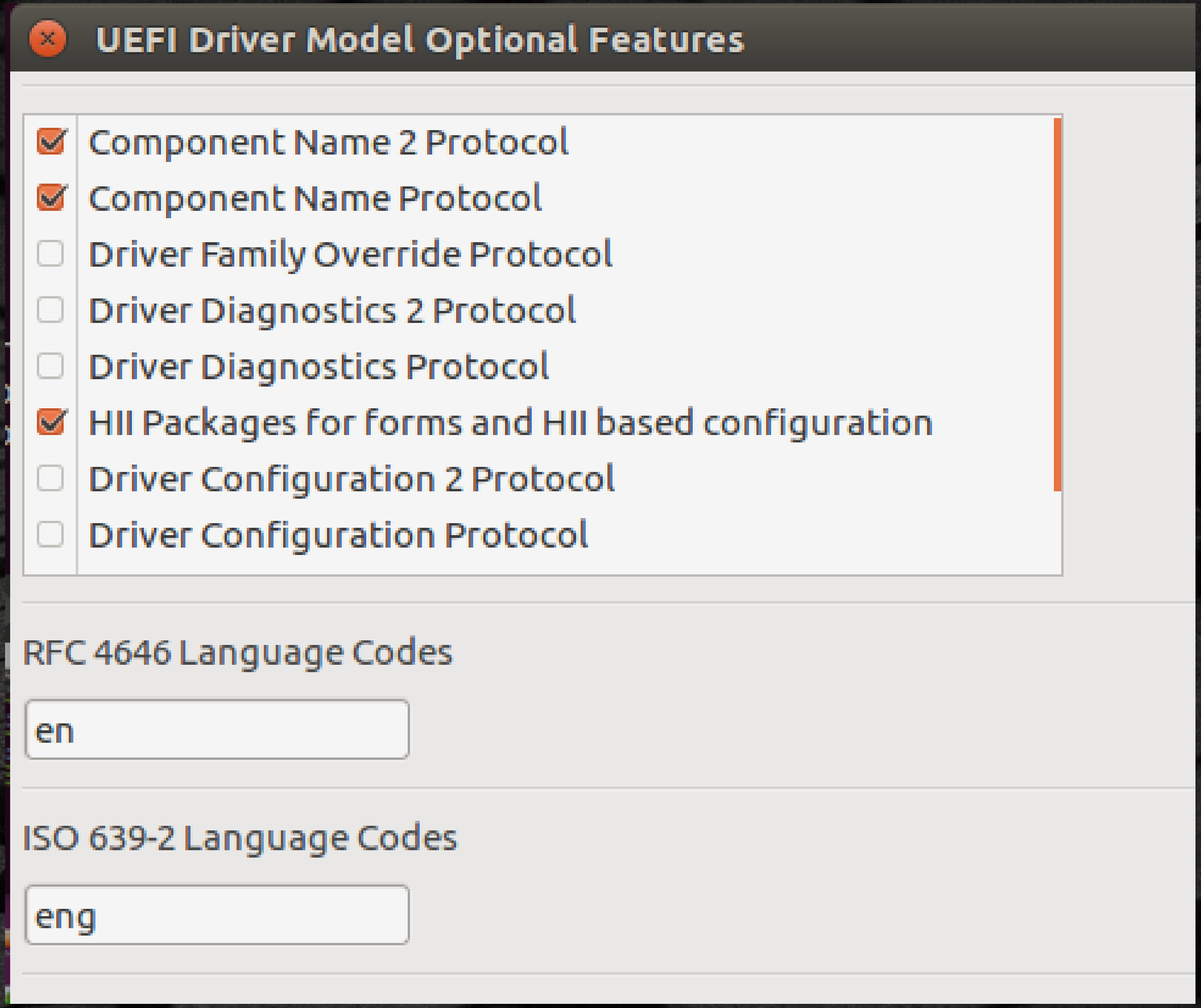
CPU Architectures

□ All CPU Architectures
☑ IA32
☑ X64
□ IPF
□ EBC

Lab 1: UEFI Driver Model Optional Features

Ensure all the forms, radio buttons, and boxes are filled in and **selected *exactly*** like the image to the right.

- ✓ "Componnt Name 2 Prorocol"
- ✓ "Componnt Name Prorocol"
- ✓ "HII Packages for Forms . . ."



UEFI Driver Model Optional Features

☑ Component Name 2 Protocol
☑ Component Name Protocol
□ Driver Family Override Protocol
□ Driver Diagnostics 2 Protocol
□ Driver Diagnostics Protocol
☑ HII Packages for forms and HII based configuration
□ Driver Configuration 2 Protocol
□ Driver Configuration Protocol

RFC 4646 Language Codes: en

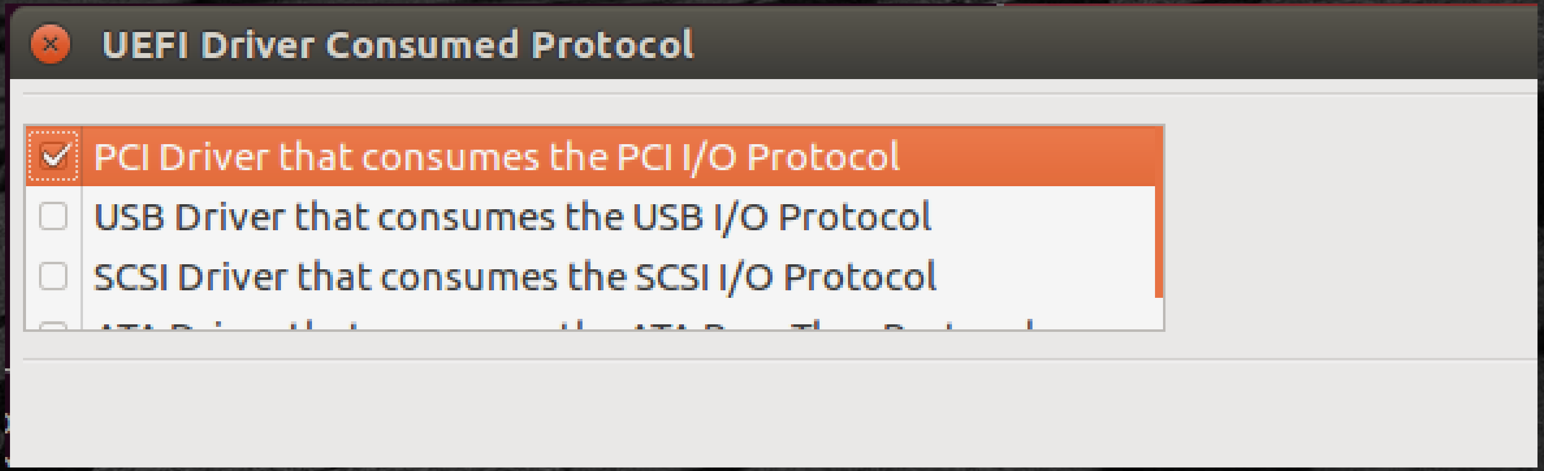
ISO 639-2 Language Codes: eng

Click Next >>

Lab 1: UEFI Driver Consumed Protocol

Select

✓ “PCI Driver that consumes the PCI I/O Protocol”



UEFI Driver Consumed Protocol

☑ PCI Driver that consumes the PCI I/O Protocol
□ USB Driver that consumes the USB I/O Protocol
□ SCSI Driver that consumes the SCSI I/O Protocol

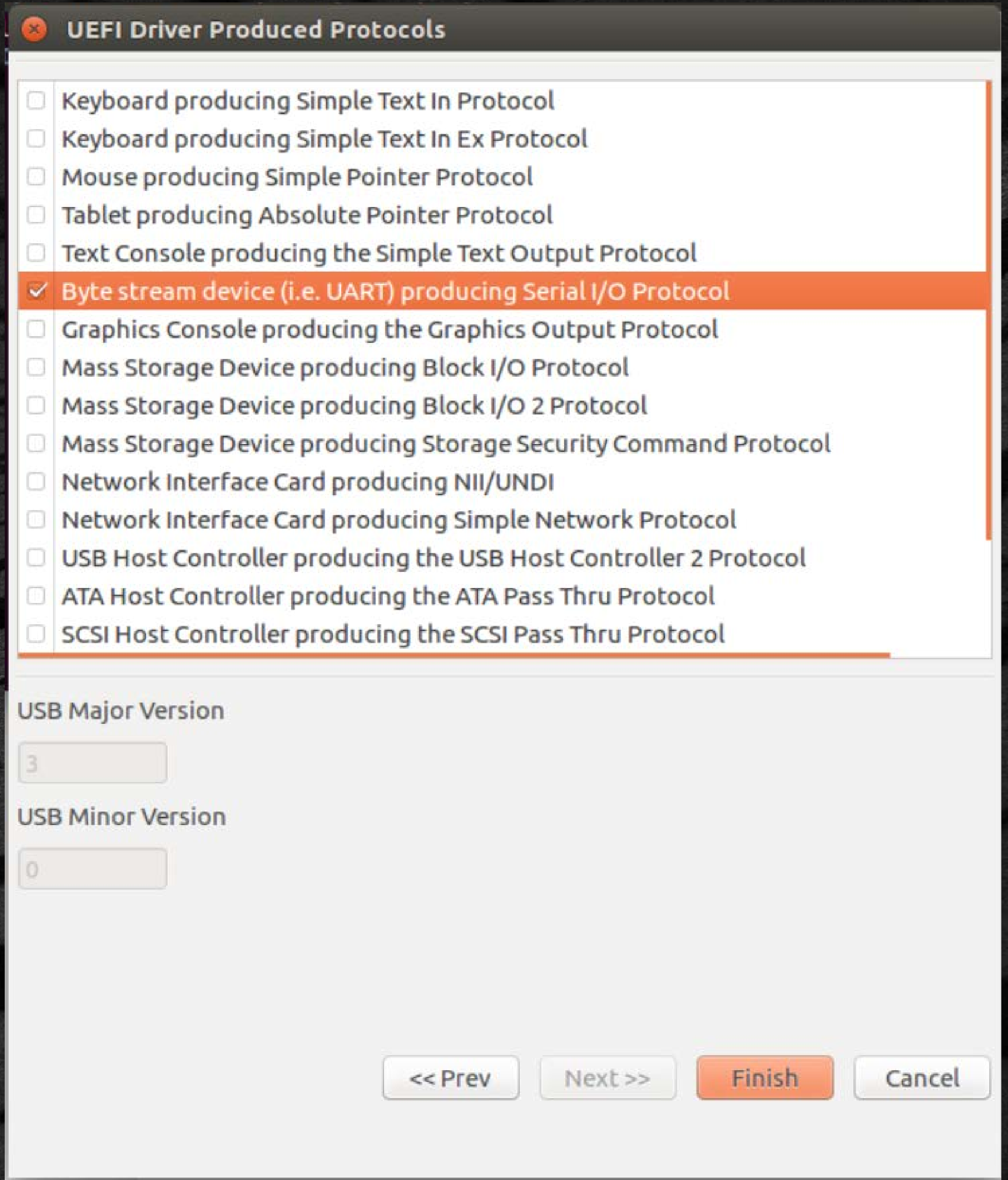
Click Next >>

Lab1: UEFI Driver Produced Protocols

Select

✓ "Byte stream device (i.e.UART) producing Serial I/O Protocol”

Click Finish



UEFI Driver Produced Protocols

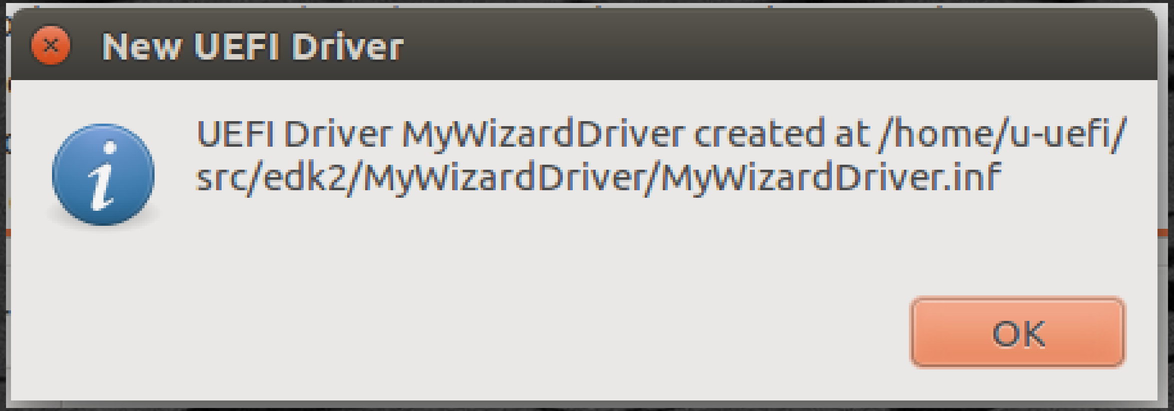
- □ Keyboard producing Simple Text In Protocol
- □ Keyboard producing Simple Text In Ex Protocol
- □ Mouse producing Simple Pointer Protocol
- □ Tablet producing Absolute Pointer Protocol
- □ Text Console producing the Simple Text Output Protocol
- ☑ Byte stream device (i.e. UART) producing Serial I/O Protocol
- □ Graphics Console producing the Graphics Output Protocol
- □ Mass Storage Device producing Block I/O Protocol
- □ Mass Storage Device producing Block I/O 2 Protocol
- □ Mass Storage Device producing Storage Security Command Protocol
- □ Network Interface Card producing NII/UNDI
- □ Network Interface Card producing Simple Network Protocol
- □ USB Host Controller producing the USB Host Controller 2 Protocol
- □ ATA Host Controller producing the ATA Pass Thru Protocol
- □ SCSI Host Controller producing the SCSI Pass Thru Protocol

USB Major Version: 3

USB Minor Version: 0

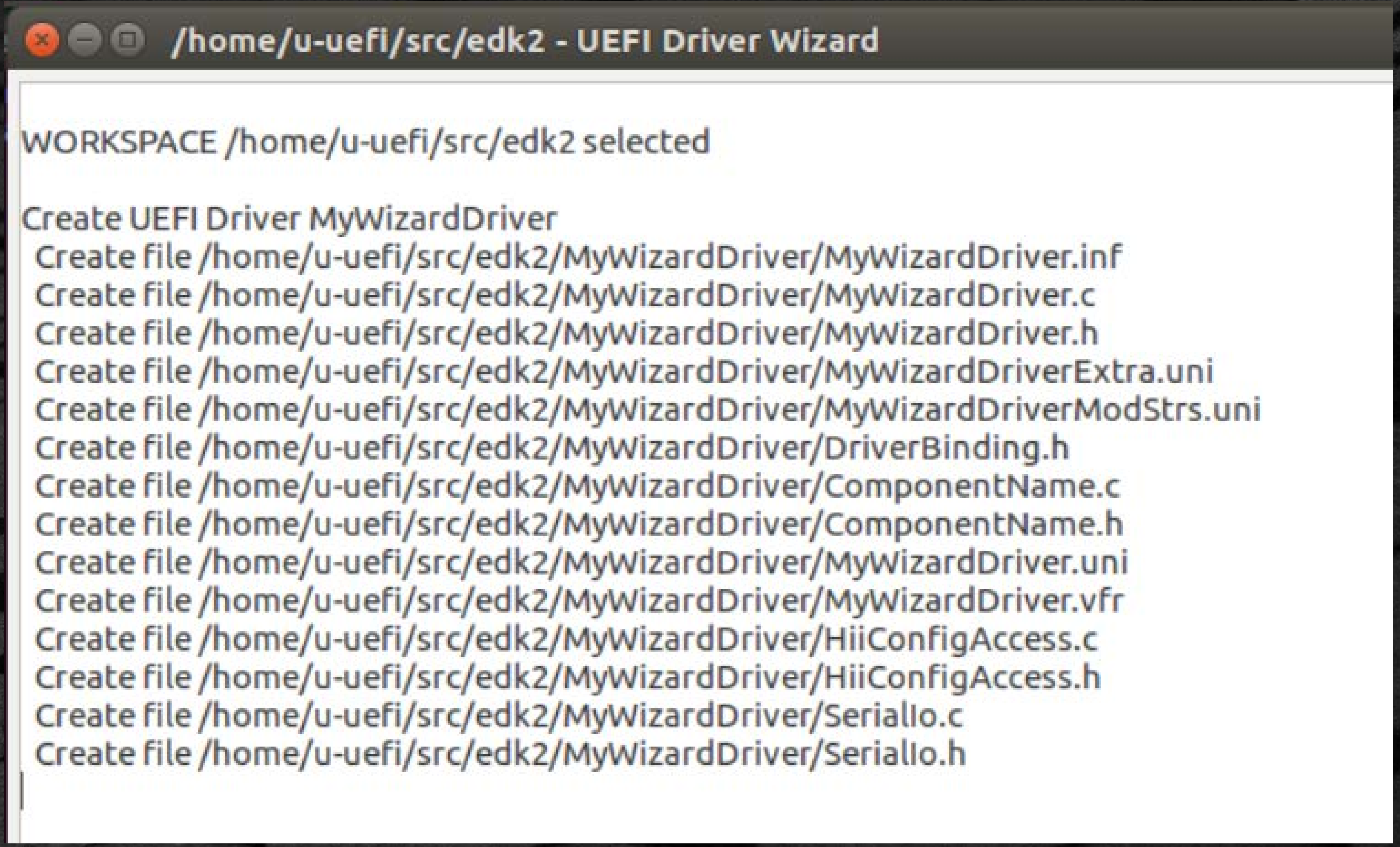
Lab 1: UEFI Driver Created

UEFI Driver template created



New UEFI Driver

UEFI Driver MyWizardDriver created at /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.inf



/home/u-uefi/src/edk2 - UEFI Driver Wizard

```
WORKSPACE /home/u-uefi/src/edk2 selected

Create UEFI Driver MyWizardDriver
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.inf
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.c
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.h
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriverExtra.uni
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriverModStrs.uni
 Create file /home/u-uefi/src/edk2/MyWizardDriver/DriverBinding.h
 Create file /home/u-uefi/src/edk2/MyWizardDriver/ComponentName.c
 Create file /home/u-uefi/src/edk2/MyWizardDriver/ComponentName.h
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.uni
 Create file /home/u-uefi/src/edk2/MyWizardDriver/MyWizardDriver.vfr
 Create file /home/u-uefi/src/edk2/MyWizardDriver/HiiConfigAccess.c
 Create file /home/u-uefi/src/edk2/MyWizardDriver/HiiConfigAccess.h
 Create file /home/u-uefi/src/edk2/MyWizardDriver/SerialIo.c
 Create file /home/u-uefi/src/edk2/MyWizardDriver/SerialIo.h
```

SUMMARY

- Setup the UEFI Driver Wizard
- Create a UEFI Driver Template

Questions?

tianocore