

# UEFI & EDK II Training

Continuous Integration (CI) Unit Test Framework for  
Developer Validation Lab

[tianocore.org](https://tianocore.org)

See also [LabGuide.md](#) for Copy & Paste examples in labs



This lab will show how to build and run a unit test sample code in the host-based environment.

- Step by step guide for the Stuart CI build and run for the Sample Unit Test from UnitTestFrameworkPkg
- Steps to build for the Non-Stuart CI build and run
- Create a Host Unit Test Framework for a simple function
- Add a UEFI Shell Unit Test Framework using the EmulatorPkg



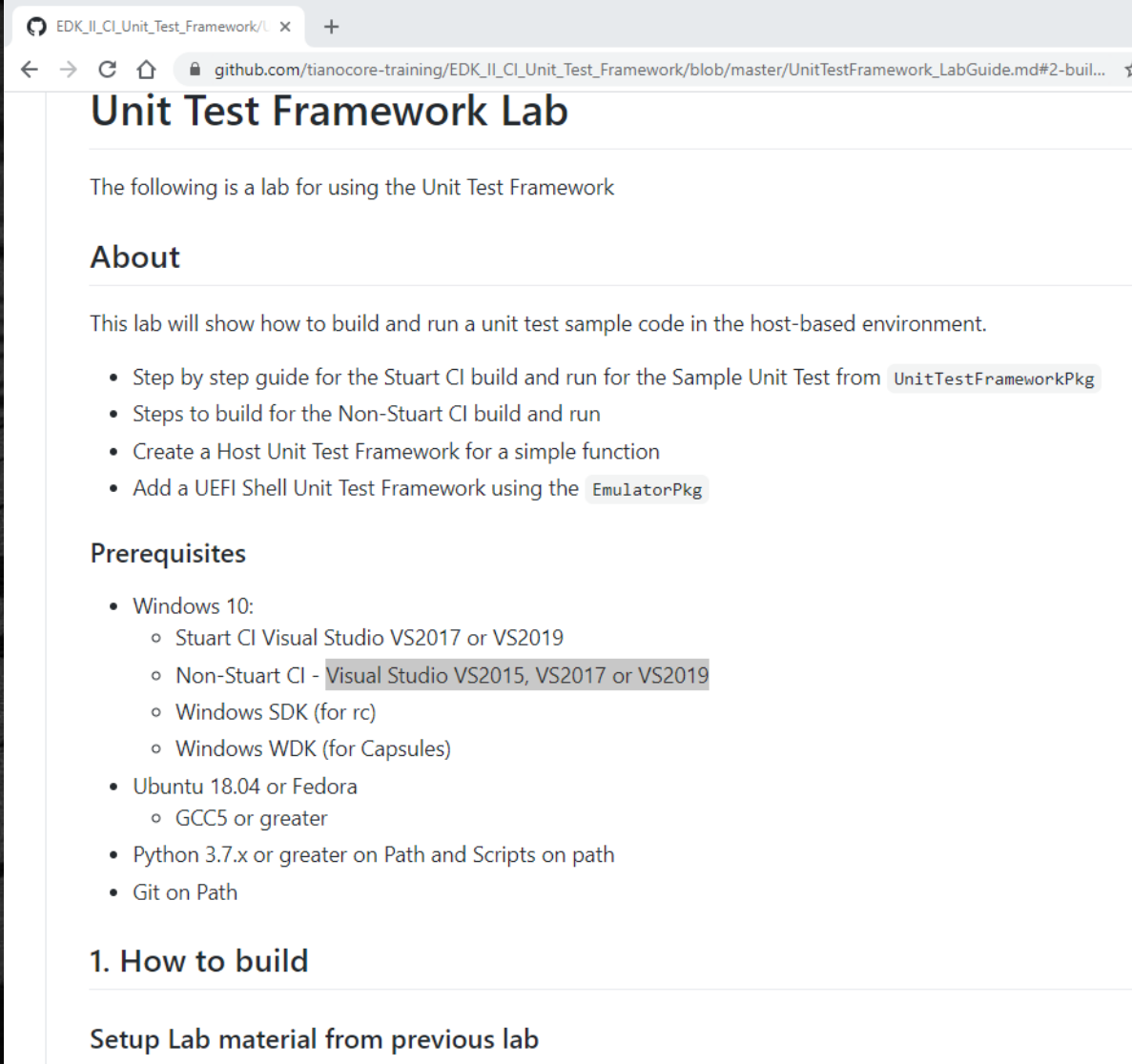
- Windows 10:
  - Stuart CI - Visual Studio VS2017 or VS2019
  - Non-Stuart CI - Visual Studio VS2015, VS2017 or VS2019
  - Windows SDK (for rc)
  - Windows WDK (for Capsules)
- Ubuntu 18.04 or Fedora
  - GCC5 or greater
- Python 3.7.x or greater on Path
- Git on Path



# Steps for this Lab

1. How to build
2. Build and Run for Stuart CI Locally
3. Or Build and Run for Windows Non-Stuart CI
4. Run the Host Unit test locally
5. Create and Add a unit test case to test a function
6. Add the Unit test case the UEFI Shell

Solutions: See in the LabMaterial\_FW/FW/  
/LabSampleCode/LessonU\_Unit\_Test



EDK\_II\_CI\_Unit\_Test\_Framework/ x +

github.com/tianocore-training/EDK\_II\_CI\_Unit\_Test\_Framework/blob/master/UnitTestFramework\_LabGuide.md#2-buil... ☆

## Unit Test Framework Lab

The following is a lab for using the Unit Test Framework

### About

This lab will show how to build and run a unit test sample code in the host-based environment.

- Step by step guide for the Stuart CI build and run for the Sample Unit Test from `UnitTestFrameworkPkg`
- Steps to build for the Non-Stuart CI build and run
- Create a Host Unit Test Framework for a simple function
- Add a UEFI Shell Unit Test Framework using the `EmulatorPkg`

### Prerequisites

- Windows 10:
  - Stuart CI Visual Studio VS2017 or VS2019
  - Non-Stuart CI - Visual Studio VS2015, VS2017 or VS2019
  - Windows SDK (for rc)
  - Windows WDK (for Capsules)
- Ubuntu 18.04 or Fedora
  - GCC5 or greater
- Python 3.7.x or greater on Path and Scripts on path
- Git on Path

### 1. How to build

Setup Lab material from previous lab



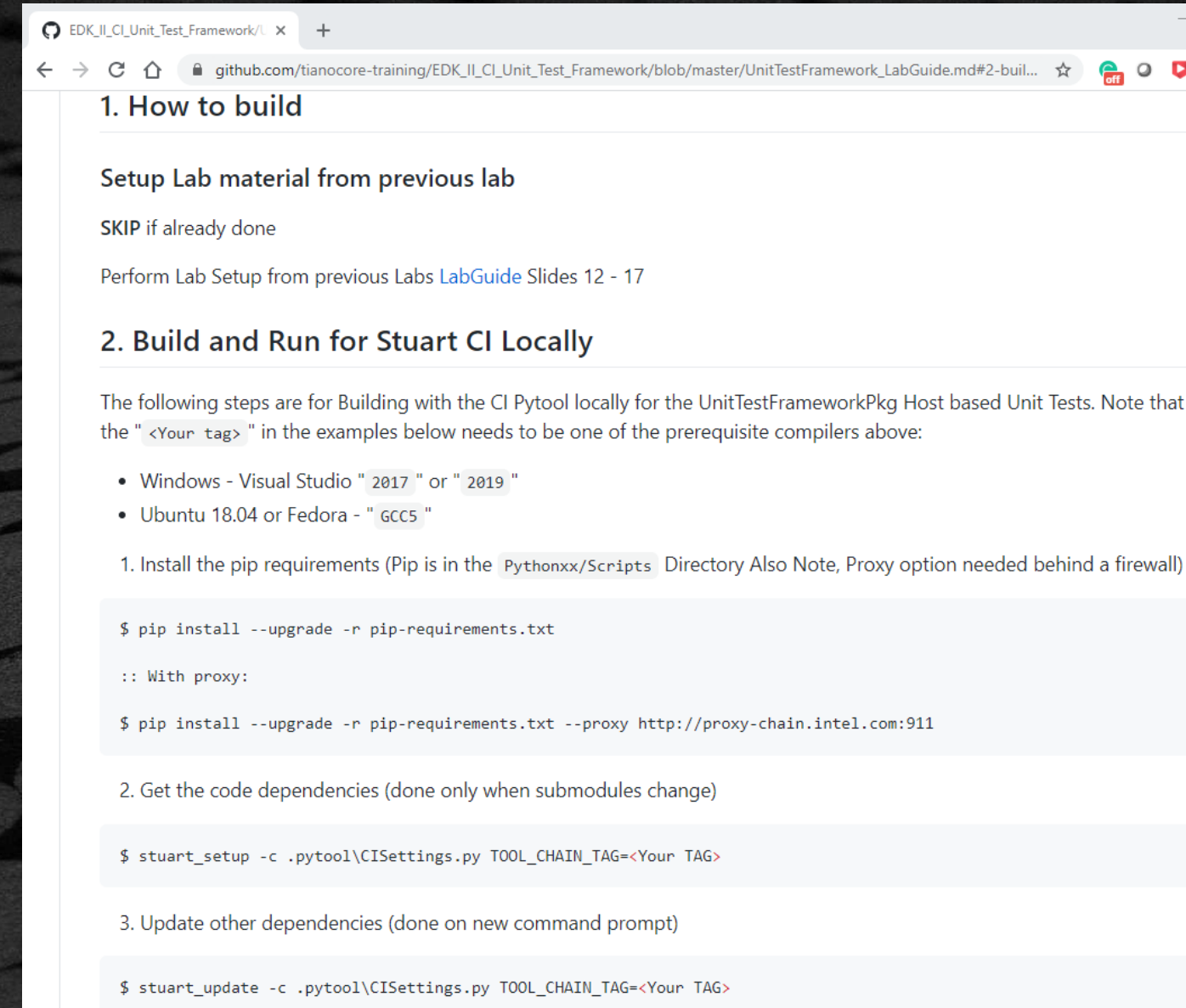
## Setup Lab material from previous lab

**SKIP** if already done

- Perform Lab Setup from previous Labs [LabGuide](#) Slides 12 - 17

**Continue** following the Lab Guide Starting at [How to Build](#)

Build either Stuart CI or Non-Stuart CI



EDK\_II\_CI\_Unit\_Test\_Framework/... x +

github.com/tianocore-training/EDK\_II\_CI\_Unit\_Test\_Framework/blob/master/UnitTestFramework\_LabGuide.md#2-buil...

### 1. How to build

Setup Lab material from previous lab

SKIP if already done

Perform Lab Setup from previous Labs [LabGuide](#) Slides 12 - 17

### 2. Build and Run for Stuart CI Locally

The following steps are for Building with the CI Pytool locally for the UnitTestFrameworkPkg Host based Unit Tests. Note that the "<Your\_tag>" in the examples below needs to be one of the prerequisite compilers above:

- Windows - Visual Studio "2017" or "2019"
- Ubuntu 18.04 or Fedora - "GCC5"

1. Install the pip requirements (Pip is in the Pythonxx/Scripts Directory Also Note, Proxy option needed behind a firewall)

```
$ pip install --upgrade -r pip-requirements.txt
```

:: With proxy:

```
$ pip install --upgrade -r pip-requirements.txt --proxy http://proxy-chain.intel.com:911
```

2. Get the code dependencies (done only when submodules change)

```
$ stuart_setup -c .pytool\CISettings.py TOOL_CHAIN_TAG=<Your_TAG>
```

3. Update other dependencies (done on new command prompt)

```
$ stuart_update -c .pytool\CISettings.py TOOL_CHAIN_TAG=<Your_TAG>
```



## Unit Test Framework Package Overview

– [Link](#)

Continuous Integration (CI) Configuring  
for Unit Tests – [Link](#)

## Code Examples of Unit Test Cases

- [Sample Unit Test](#)
- [BaseSafeIntLib Unit Test](#)
- [BaseLib Unit Test](#)
- [DxeResetSystemLib Unit Test](#)
- [MtrrLibUnitTest](#)
- Cmocka Edk II Unit Test ChefCook  
example: [link](#)



# Questions?





# Return to Main Training Page



Return to Training Table of contents for next presentation

[Link](#)







# ACKNOWLEDGMENTS

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) 2020, Intel Corporation. All rights reserved.