Understanding UEFI Testing Webinar

Presented by UEFI Forum

Wednesday, January 29, 2020

Welcome & Introductions

Moderator: Brian Richardson
Firmware Ecosystem Development
Member Company: Intel Corporation
@intel_brian

Panelist: Dong Wei
UEFI Forum Vice President
and UTWG Work Group Chair
Member Company: Arm

Panelist: Alex Hung
Software Engineer and Maintainer of
Firmware Test Suite
Member Company: Canonical

Panelist: Supreeth Venkatesh
Software Engineer and Tech
Lead
Member Company: Arm

Agenda

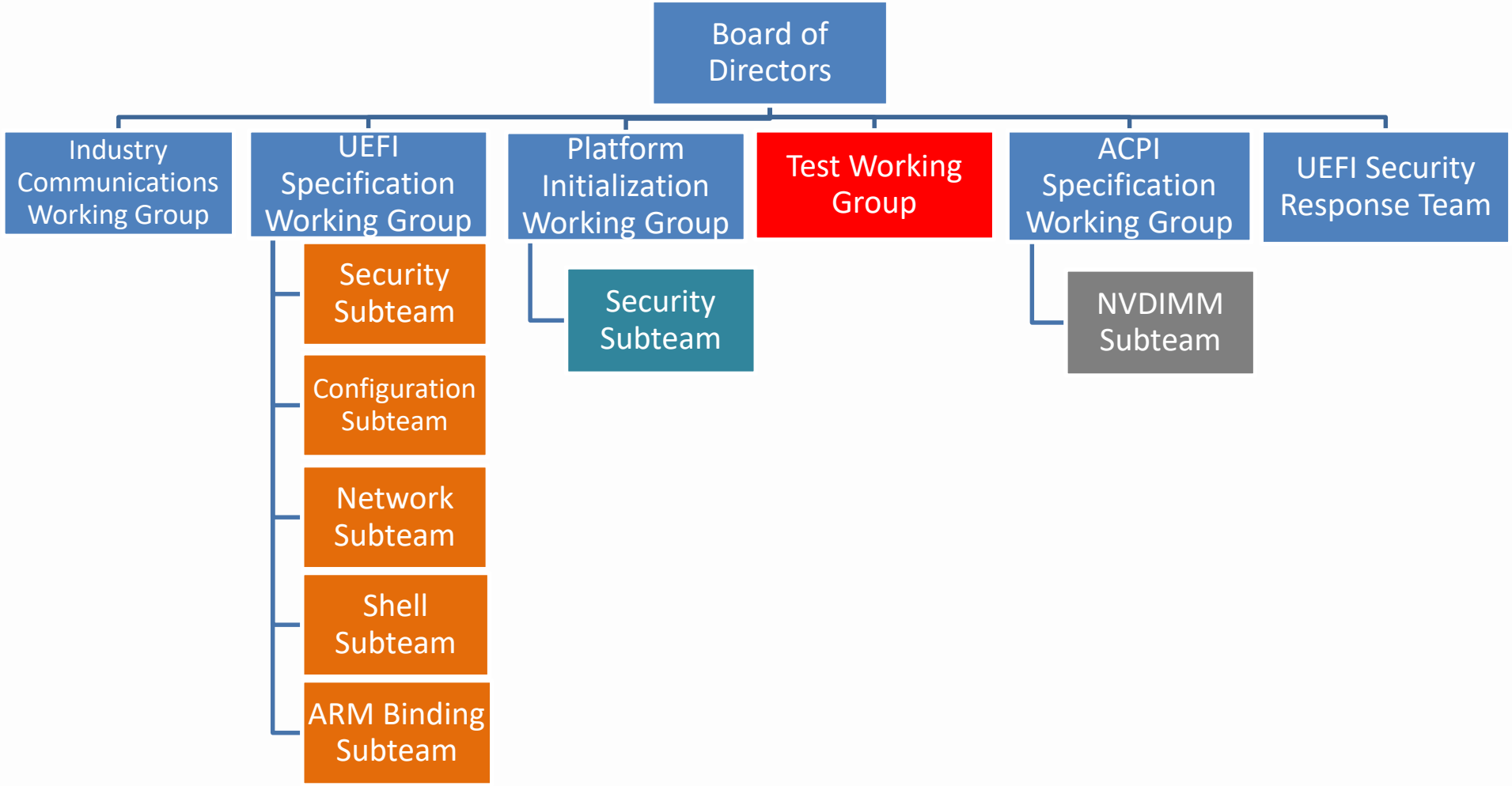
- UEFI Test Work Group
- Open Source Toolset and UEFI Self Certification Test
- Firmware Test Suite
- Q&A
- Save the Date: UEFI Forum Spring 2020 Plugfest

Dong Wei

UEFI Forum Vice President and UTWG Work Group Chair, Arm

What does the UEFI Test Work Group (UTWG) do?

UEFI Forum Overview



UTWG Charter

The group is responsible for

(1) Recommending a release of the open source **FWTS** for a version of the ACPI Spec

(2) Recommending a release of the open source **UEFI SCT** for a version of the UEFI Spec

(3) Recommending a release of the open source PI SCT for a version of the PI Spec

Results:

https://uefi.org/testtools

UEFI Forum Test Tools

Basics

- Self Certifying, No UEFI Forum Certification Program
- Testing Interface Compliance, Not Functional

Extensions

- Other entities can leverage and extend beyond the UEFI Forum usages
  - Arm builds Enterprise ACS on top to check for SBSA/SBBR Compliance to form the basis of the Arm ServerReady Certification

https://developer.arm.com/architectures/platform-design/server-systems

Supreeth Venkatesh

Arm

What is the open source toolset and what is the UEFI Self Certification Test (SCT)?

Introduction

- Modern, feature-rich, cross-platform firmware test environment for the UEFI specification
- Test harness for executing built-in compliance tests
- Scope for integrating user-defined tests

System Requirements

- The target system must have an X64 platform, an IA-32, or an ARM platform
- The target system firmware must have UEFI implemented per the UEFI Specification
- The UEFI implementation on the target system must include an UEFI Shell
- The target system must have at least 1000MB of disk space in the UEFI file system to contain the SCT test and log files

UEFI SCT Open Source Status

- Latest stable binary version - UEFI SCT 2.7B is being published at https://uefi.org/testtools
- UEFI SCT 2.7B – edk2-test-stable201910.
- Open source repo is https://github.com/tianocore/edk2-test
- Stable tag each quarter

Build UEFI SCT

- Steps for building uefi-sct binaries are documented here: https://github.com/tianocore/edk2-test/tree/master/uefi-sct/HowToBuild
- Builds currently available for
  - X64/IA32 platforms
  - AARCH64 platforms

Installation

- Ensuring that the target system is configured to boot to the UEFI Shell upon power-on/reset without user intervention
- Setting the boot options is usually done using UEFI Boot Manager during the target system's UEFI implementation
- Installing the UEFI SCT executable files into a default directory in the UEFI file system of the target system
- The default directory must be on a Read/Write storage medium

Usage Model

- Invoked as an UEFI application from the UEFI Shell
- Syntax

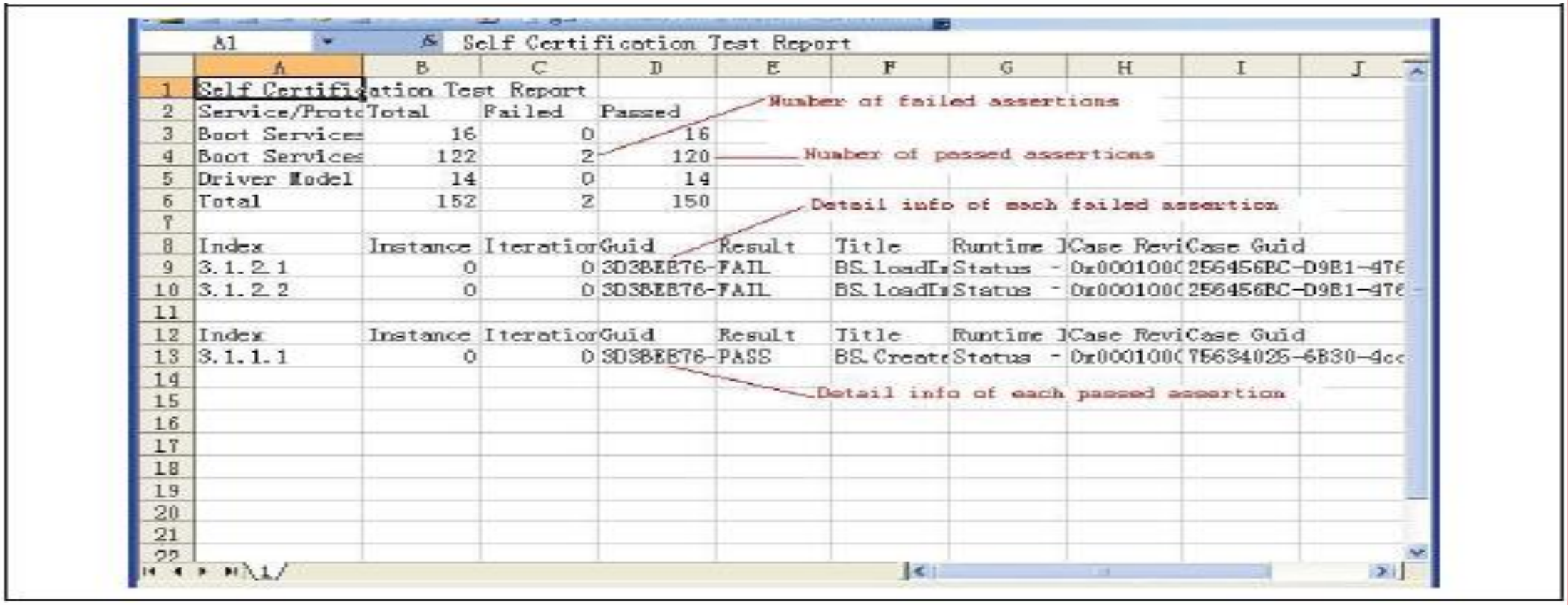
SCT [-a | -c | -s <seq> | -u] | -p <MNP | IP4 | Serial>] [-r] [-g <report>][-v]

Table 1. SCT Parameters

Options	Description
–a	Execute all test cases that are recognized by the UEFI SCT Test Harness.
–c	Continue execution of the test case in progress. This option is used to continue execution of test cases that perform system resets as part of their test routine.
–g <report>	Generate test report in .CSV format. The filename of the report is specified by report.
–r	Resets the environment for a fresh execution of the tests. This option removes results of previous test executions. Generally, it is used with the –a or –s options.
–s <seq>	Execute test cases in the sequence specified in the file seq.
–u	Start the Test Harness with the menu-driven interface.
–p	Passive Mode with specified communication layer
–f	Force the operation execution, no confirmation from user.
–v	Disables the display of test log information on the screen.

Test Report

- A summary of SCT test results is recorded into a test report file in CSV format



	A	B	C	D	E	F	G	H	I	J
1	Self Certifi	ation Test Report								
2	Service/Protc	Total	Failed	Passed						
3	Boot Service:	16	0	16						
4	Boot Service:	122	2	120						
5	Driver Model	14	0	14						
6	Total	152	2	150						
7										
8	Index	Instance	Iteratior	Guid	Result	Title	Runtime]	Case Revi	Case Guid	
9	3.1.2.1	0	0	3D3BEB76-	FAIL	BS.loadIn	Status -	0x000100(	256456BC-D9B1-47(	
10	3.1.2.2	0	0	3D3BEB76-	FAIL	BS.loadIn	Status -	0x000100(	256456BC-D9B1-47(	
11										
12	Index	Instance	Iteratior	Guid	Result	Title	Runtime]	Case Revi	Case Guid	
13	3.1.1.1	0	0	3D3BEB76-	PASS	BS.Create	Status -	0x000100(	75634025-6B30-4cc	
14										
15										
16										
17										
18										
19										
20										
21										
22										

Reporting Issues

- UEFI-SCT follows same process as edk2 (TianoCore)
- Bugzilla for reporting bugs or requesting new feature - https://bugzilla.tianocore.org/
- File a new bug or feature request in the “UEFI-SCT” component of the “Edk2 Test” product

Call to Action

- Edk2-test follows the same contribution guidelines as edk2 (TianoCore) project - https://github.com/tianocore/tianocore.github.io/wiki/How-To-Contribute
- SCT source is under 2-Clause BSD License
  https://github.com/tianocore/edk2-test/blob/master/License.txt

- General questions on UEFI SCT with the subject [edk2-test] can be sent to edk2-list devel@edk2.groups.io
- UEFI SCT open source project solicits your contributions with a humble thank you!

Alex Hung

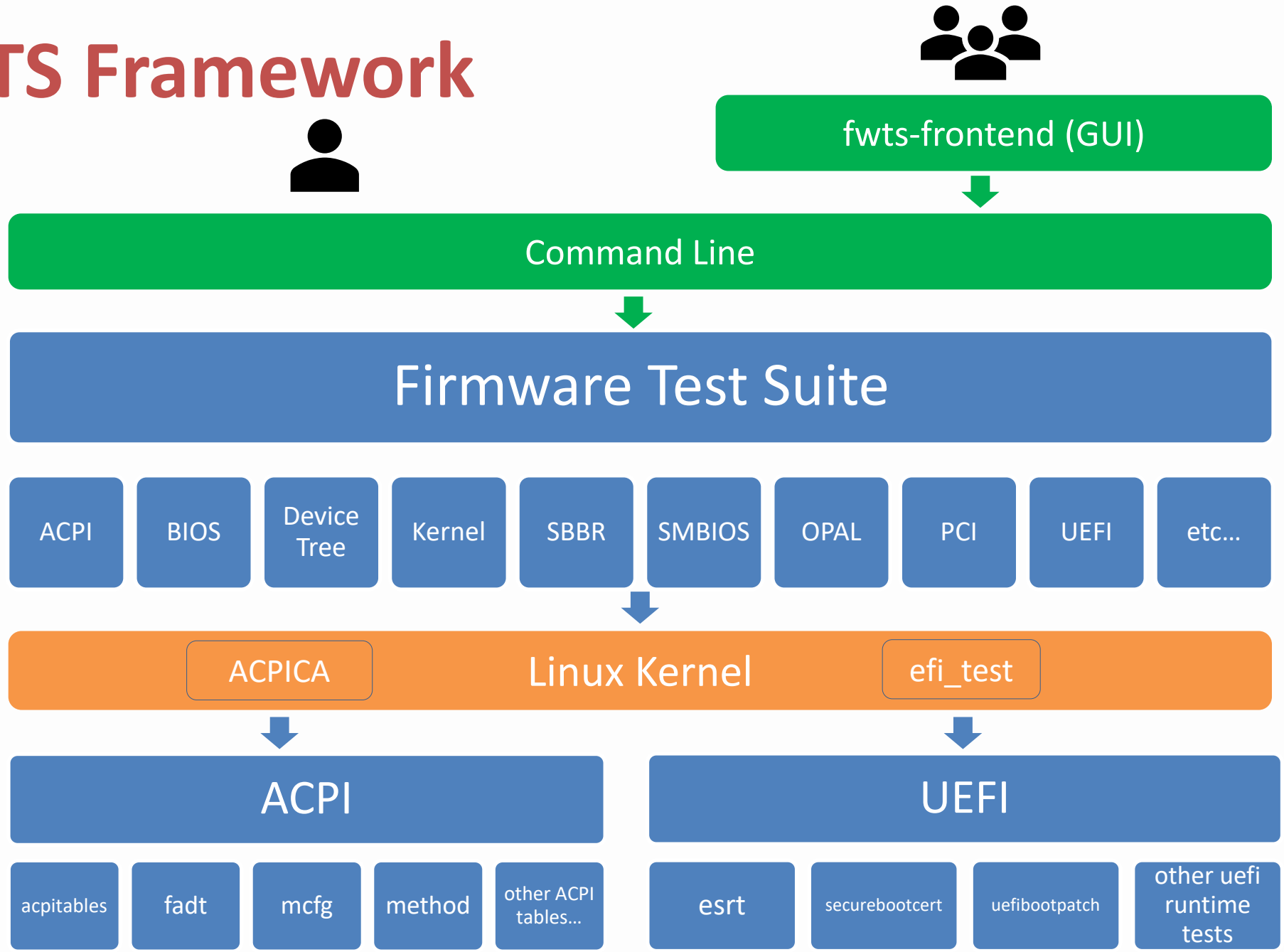
Canonical

What is the Firmware Test Suite (FWTS)?

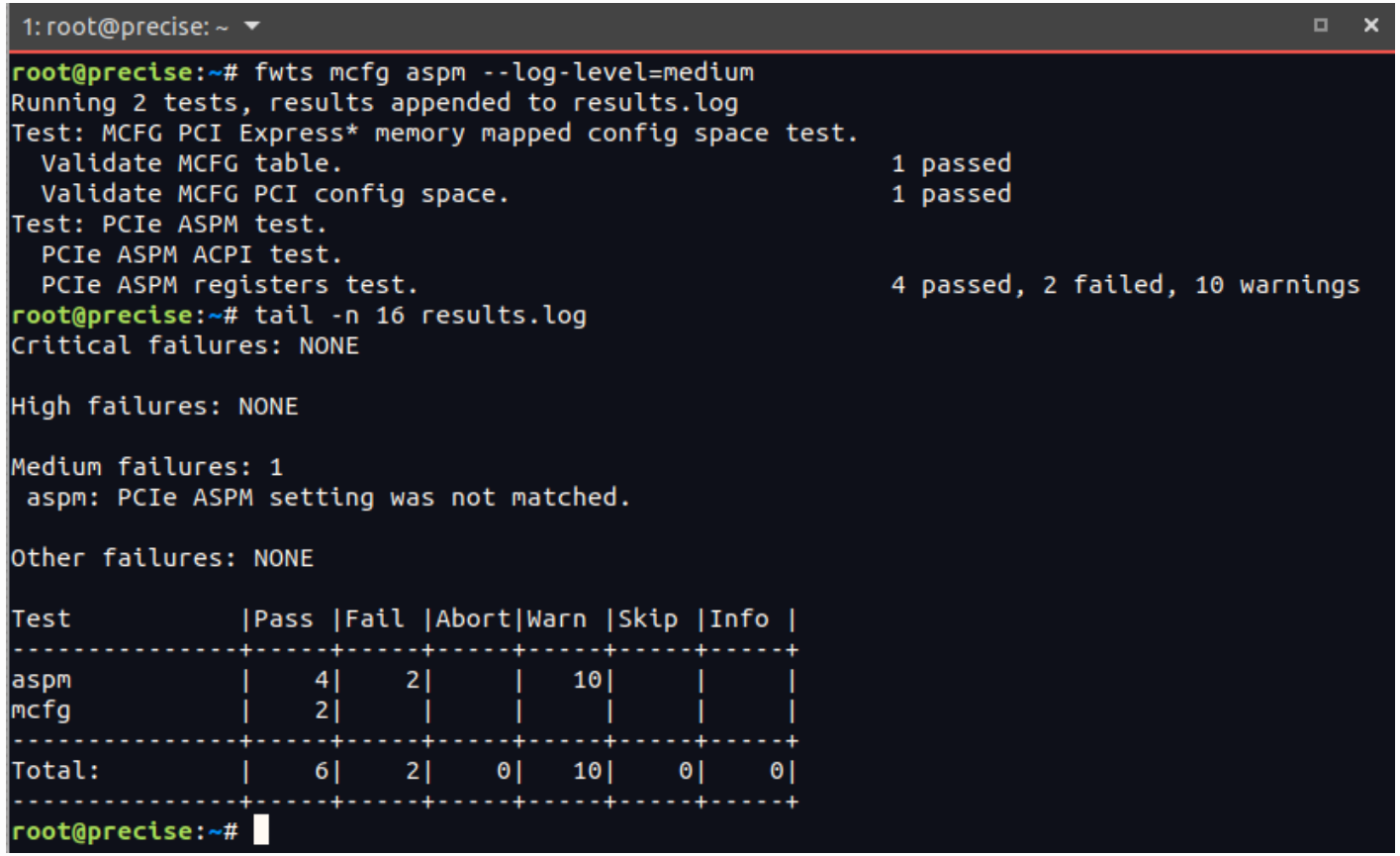
Firmware Test Suite (FWTS)

- The recommended ACPI SCT
- A Linux command-line tool that automates firmware checking
  - Identify UEFI, ACPI and many other errors
  - Explain the errors and give advice to fix firmware bugs
- A GPLv2 open-source software tool
- Install FWTS on Ubuntu Linux
  - *add-apt-repository ppa:firmware-testing-team/ppa-fwts-stable*
  - *apt install fwts fwts-frontend*
- More information
  - https://wiki.ubuntu.com/FirmwareTestSuite
  - https://wiki.ubuntu.com/FirmwareTestSuite/Reference

FWTS Framework



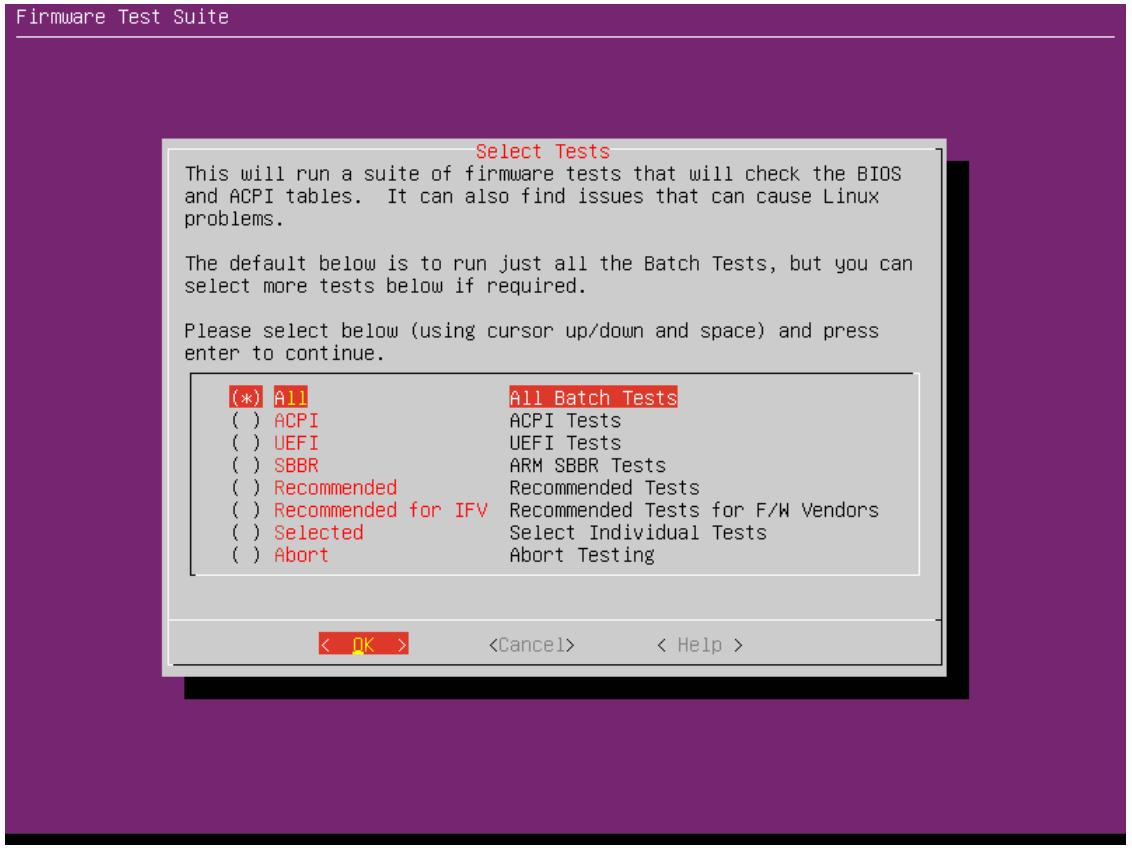
FWTS Commands



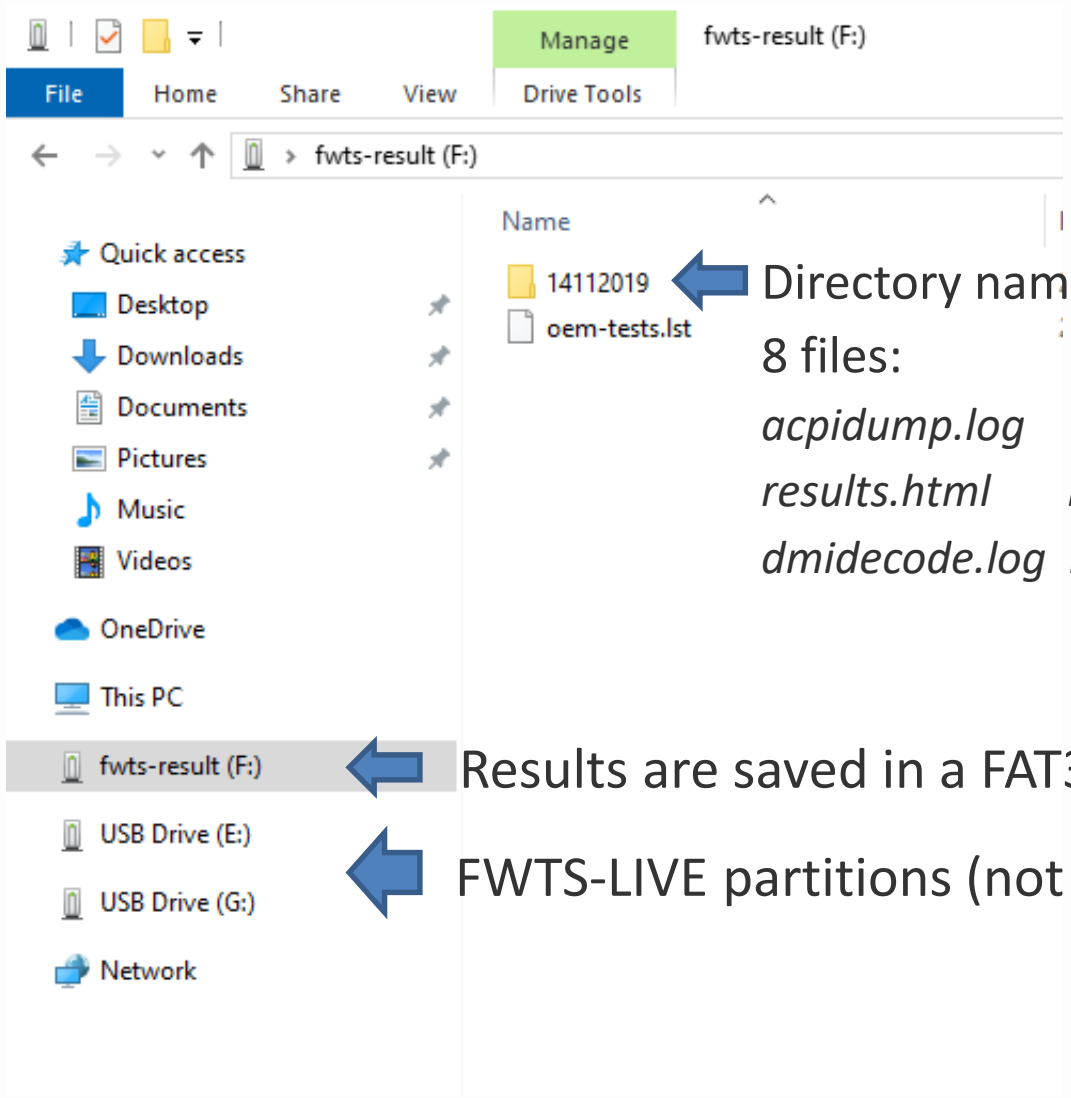
- Specify test(s): fwts test1 [test2 … testN]
- Specify a group of tests: fwts --uefitests
- Show all arguments: fwts-show-tests-full

FWTS-live

- A USB image that includes Ubuntu-Live + FWTS + fwts-frontend
  - Downloads: http://fwts.ubuntu.com/fwts-live/
  - More info: https://wiki.ubuntu.com/FirmwareTestSuite/FirmwareTestSuiteLive



FWTS-LIVE Log Files



Directory name is /date/time/

8 files:

acpidump.log *dmesg.log* *lspci.log*

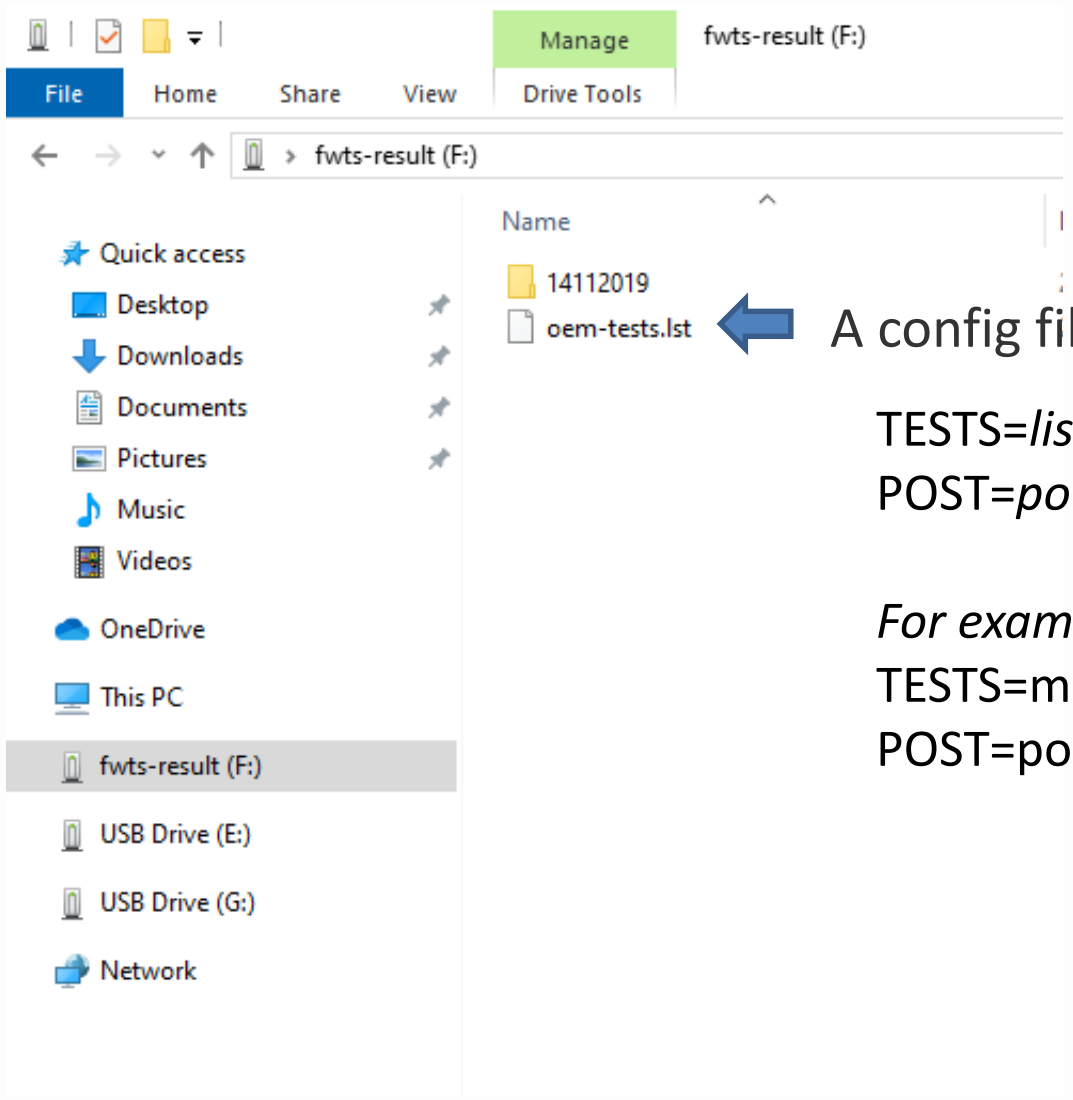
results.html *results.log* *cpuinfo.log*

dmidecode.log *README.txt*

Results are saved in a FAT32 partition

FWTS-LIVE partitions (not accessible in Windows)

FWTS-LIVE Automation



A config file for auto-test

TESTS=*list of tests to be executed*
POST=*poweroff|reboot*

For example:
TESTS=mcfg aspm
POST=poweroff

Understanding UEFI Testing Panel Discussion

Save the Date!

UEFI Forum Spring 2020 Plugfest

March 30 – April 3, 2020

Hillsboro, OR

Register Here:

https://uefiplugfest.regfox.com/spring2020uefiplugfest