

*presented by*

**CANONICAL**



# **Building Customized Tests with Firmware Test Suite**

Fall 2018 UEFI Plugfest

October 17, 2018

Presented by Alex Hung (Canonical Group Ltd.)

# Agenda



- Introduction
- Customize Tests in Scripts
- Customize Tests in GUI
- Create “Builds” for Your Team
- Build Your FWTS on Launchpad



# Introduction

# Before talking about FWTS...





# The Ultimate Question

- We know all ingredients (specs)
- But do we bake bread (firmware) right?
- Firmware Test Suite (FWTS) is
  - A test suite runs in runtime / Linux
  - The recommended ACPI SCT
  - A tool that verifies functions and definitions in various specs against implementations

# FWTS Framework



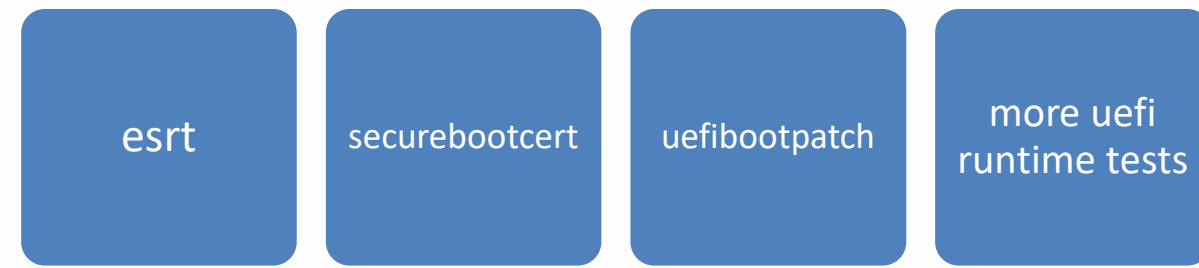
## Firmware Test Suite



## ACPI



## UEFI





# Customize Tests in Scripts

# Combine FWTS and Bash Scripts



- FWTS has different forms of options
  - Specify tests: **fwts test1 [test2 ... testN]**
  - Specify a group of tests: **fwts --uefitests**
  - Full list of arguments: **fwts --show-tests**
- Scripts can be used to run specific tests





# A Real Example

- I run fwts based on “*BIOS vendor*”

```
14 RECOMMEND_TESTS="version cpufreq maxfreq msr mtrr nx virt aspm dmidecode apicedge klog oops esrt --uefitests --acpitests"
15 RECOMMEND_TESTS_IFV="version cpufreq maxfreq msr mtrr nx virt aspm dmidecode apicedge klog oops esrt --uefitests --acpitests --ifv"
16 UPDATE_FWTS=false
17
18 readonly IFV_LIST=( American Byosoft INSYDE Intel Phoenix )
```

```
70 VENDOR=$(sudo dmidecode --string bios-vendor | awk -F ' ' '{ printf $1; }')
71 for i in "${IFV_LIST[@]}"
72 do
73     [ "$VENDOR" = "$i" ] && RECOMMEND_TESTS="$RECOMMEND_TESTS_IFV"
74 done
75
76 sudo fwts $RECOMMEND_TESTS
```



# Customize Tests in GUI (fwts-frontend-text)

# Remember FWTS-LIVE?



## Select Tests

This will run a suite of firmware tests that will check the BIOS and ACPI tables. It can also find issues that can cause Linux problems.

The default below is to run just all the Batch Tests, but you can select more tests below if required.

Please select below (using cursor up/down and space) and press enter to continue.

<input checked="" type="checkbox"/> <b>ALL</b>	<b>All Batch Tests</b>
<input type="checkbox"/> <b>ACPI</b>	ACPI Tests
<input type="checkbox"/> <b>UEFI</b>	UEFI Tests
<input type="checkbox"/> <b>SBBR</b>	ARM SBBR Tests
<input type="checkbox"/> <b>Recommended</b>	Recommended Tests
<input type="checkbox"/> <b>Recommended for IFV</b>	Recommended Tests for F/W Vendors
<input type="checkbox"/> <b>Selected</b>	Select Individual Tests
↓(+)	<b>87%</b>

< **OK** >      <Cancel>      < Help >

FWTS-LIVE  
||  
Ubuntu Live  
+  
fwts-frontend-text  
+  
FWTS

# fwts-frontend-text is a Bash Script!



```
'All')
  case "${UNAME_PLATFORM}" in
    ppc*)
      FWTS_OPTIONS="--batch"
      ;;
    *)
      FWTS_OPTIONS="--batch --uefitests --skip-test=s3,s4"
      ;;
  esac
  do_test "${FWTS_OPTIONS}" 'Running Batch Tests'
  done_tests_finish
  ;;
'ACPI')
  do_test "--acpittests --skip-test=s3,s4" 'Running ACPI Tests'
  done_tests_finish
  ;;
'UEFI')
  do_test "--uefitests --skip-test=s3,s4" 'Running UEFI Tests'
  done_tests_finish
  ;;
'SBBR')
  do_test "--sbbbr --skip-test=s3,s4" 'Server Base Boot Requirements (SBBR) Tests'
  done_tests_finish
  ;;
'Recommended')
  case "${UNAME_PLATFORM}" in
    ppc*)
      FWTS_OPTIONS="--batch"
      ;;
    *)
      FWTS_OPTIONS="version cpufreq maxfreq msr mtrr nx virt aspm dmichk apicedge klog oops --acpites"
      ;;
  esac

  do_test "${FWTS_OPTIONS}" 'Running Recommended Tests'
  done_tests_finish
  ;;
```

# An Sample Patch



```
18 --- a/live-image/fwts-frontend-text
19 +++ b/live-image/fwts-frontend-text
20 @@ -260,6 +260,7 @@ if [ $? -ne 0 ]; then
21     fi
22
23     UNAME_PLATFORM=`(uname -m) 2>/dev/null` || UNAME_PLATFORM=unknown
24 +CHASSIS_TYPE=`(dmidecode -s chassis-type) 2>/dev/null` || CHASSIS_TYPE=unknown
25
26     while true
27     do
28 @@ -283,6 +284,7 @@ do
29         "ACPI" "ACPI Tests" off \
30         "UEFI" "UEFI Tests" off \
31         "SBBR" "ARM SBBR Tests" off \
32 +         "Chassis" "Chassis Tests" off \
33         "Recommended" "Recommended Tests" off \
34         "Recommended for IFV" "Recommended Tests for F/W Vendors" off \
35         "Selected" "Select Individual Tests" off \
36 @@ -320,6 +322,21 @@ do
37         do_test "--sbbbr --skip-test=s3,s4" 'Server Base Boot Requirements (SBBR) Tests'
38         done_tests_finish
39         ;;
40 +         'Chassis')
41 +         case "${CHASSIS_TYPE}" in
42 +             Desktop)
43 +                 FWTS_OPTIONS="dmicheck fadt bgrt hest"
44 +                 ;;
45 +             Notebook)
46 +                 FWTS_OPTIONS="acpi_ac acpi_battery acpi_lid"
47 +                 ;;
48 +             *)
49 +                 FWTS_OPTIONS="--uefitests"
50 +                 ;;
51 +         esac
52 +         do_test "${FWTS_OPTIONS}" 'Running Chassis Tests'
53 +         done_tests_finish
54 +         ;;
55     'Recommended')
56         case "${UNAME_PLATFORM}" in
57             ppc*)
```



# Create “Builds” for Your Team



# Run FWTS on Products

- FWTS includes tests for various specs
  - FWTS keeps up with spec updates too
- However, real products
  - Cover part of specs
  - Must (not) include certain parts of specs
  - Implement other inhouse requirements



# Implement “Builds” of Tests

- FWTS already includes some “builds”
  - acpittests, ifv, uefittests, **sbbr**
- ARM Server Base Boot Requirements (**SBBR**) covers necessary parts of ACPI, SMBIOS and UEFI
  - **sbbr** build is contributed by Arm
  - Commit 463ae8584





# Other Possibilities

- By Chassis: desktop, laptop, or server
  - Check SMBIOS and ACPI tables
- By OS: Linux or Windows
  - Include or exclude certain ACPI tables
- An example patch is available on [https://github.com/alexhungce/uefi\\_plugfest](https://github.com/alexhungce/uefi_plugfest)

# An Sample Patch



```
59 --- a/src/lib/include/fwts_framework.h
60 +++ b/src/lib/include/fwts_framework.h
61 @@ -61,7 +61,9 @@ typedef enum {
62     FWTS_FLAG_SHOW_TESTS_FULL           = 0x00200000,
63     FWTS_FLAG_SHOW_TESTS_CATEGORIES    = 0x00400000,
64     FWTS_FLAG_TEST_COMPLIANCE ACPI     = 0x00800000,
65 -    FWTS_FLAG_TEST_SBBR                = 0x01000000
66 +    FWTS_FLAG_TEST_SBBR                = 0x01000000,
67 +    FWTS_FLAG_TEST_DESKTOP             = 0x02000000,
68 +    FWTS_FLAG_TEST_NOTEBOOK           = 0x04000000
69 } fwts_framework_flags;
70
71 #define FWTS_FLAG_TEST_MASK            \
72 diff --git a/src/lib/src/fwts_framework.c b/src/lib/src/fwts_framework.c
73 index bedc1ee9..b0c61c6e 100644
74 --- a/src/lib/src/fwts_framework.c
75 +++ b/src/lib/src/fwts_framework.c
76 @@ -136,6 +136,8 @@ static fwts_option fwts_framework_options[] = {
77     { "sbb",          "", 0, "Run ARM SBBR tests." },
78     { "ifv",         "", 0, "Run tests in firmware-vendor modes." },
79     { "clog",        "", 1, "Specify a coreboot logfile dump" },
80 +    { "desktop",    "", 0, "Run tests as desktop systems." },
81 +    { "notebook",   "", 0, "Run tests as mobile systems." },
82     { NULL, NULL, 0, NULL }
83 };
84
85 @@ -1338,6 +1340,13 @@ int fwts_framework_options_handler(fwts_framework *fw, int argc, char * const ar
86     break;
87     case 48: /* --coreboot-log */
88         fwts_framework_strdup(&fw->clog, optarg);
89 +    break;
90 +    case 49: /* --desktop */
91 +    fw->flags |= FWTS_FLAG_TEST_DESKTOP;
92 +    break;
93 +    case 50: /* --notebook */
94 +    fw->flags |= FWTS_FLAG_TEST_NOTEBOOK;
95 +    break;
96
97     }
98     break;
```

# An Sample Patch (cont'd)



```
21 --- a/src/acpi/fadt/fadt.c
22 +++ b/src/acpi/fadt/fadt.c
23 @@ -425,6 +425,14 @@ static void acpi_table_check_fadt_pm_profile(fwts_framework *fw)
24     else
25         fwts_failed(fw, LOG_LEVEL_MEDIUM, "FADTPMProfile",
26                     "FADT preferred PM profile is invalid.");
27 +
28 +     if (fw->flags & FWTS_FLAG_TEST_DESKTOP && fadt->preferred_pm_profile != 1)
29 +         fwts_failed(fw, LOG_LEVEL_MEDIUM, "FADTPMProfile",
30 +                     "FADT preferred PM profile is not desktop.");
31 +
32 +     if (fw->flags & FWTS_FLAG_TEST_NOTEBOOK && fadt->preferred_pm_profile != 2)
33 +         fwts_failed(fw, LOG_LEVEL_MEDIUM, "FADTPMProfile",
34 +                     "FADT preferred PM profile is not mobile.");
35 }
36
37 static void acpi_table_check_fadt_reduced_hardware(fwts_framework *fw)
38 diff --git a/src/dmi/dmichkcheck/dmichkcheck.c b/src/dmi/dmichkcheck/dmichkcheck.c
39 index d0096bd4..9737ba3b 100644
40 --- a/src/dmi/dmichkcheck/dmichkcheck.c
41 +++ b/src/dmi/dmichkcheck/dmichkcheck.c
42 @@ -1196,6 +1196,14 @@ static void dmichkcheck_entry(fwts_framework *fw,
43     break;
44 }
45
46 +     if (fw->flags & FWTS_FLAG_TEST_DESKTOP && (data[5] & ~0x80) != 0x03)
47 +         fwts_failed(fw, LOG_LEVEL_MEDIUM, "DMIBadChassisType",
48 +                     "Chassis type is not desktop.");
49 +
50 +     if (fw->flags & FWTS_FLAG_TEST_NOTEBOOK && (data[5] & ~0x80) != 0x0a)
51 +         fwts_failed(fw, LOG_LEVEL_MEDIUM, "DMIBadChassisType",
52 +                     "Chassis type is not mobile.");
53 +
54     dmi_min_max_mask_uint8_check(fw, table, addr, "Chassis Lock", hdr, 0x5, 0x0, 0x1, 0x7, 0x1);
55     dmi_str_check(fw, table, addr, "Version", hdr, 0x6);
56     dmi_str_check(fw, table, addr, "Serial Number", hdr, 0x7);
```



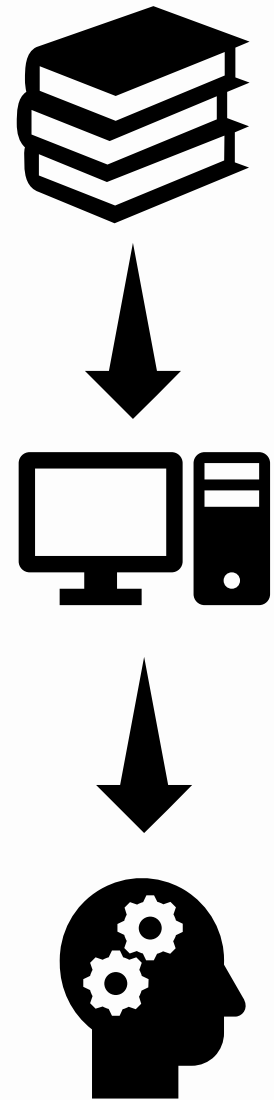
# Build Your FWTS on Launchpad

# Build Your Customized FWTS

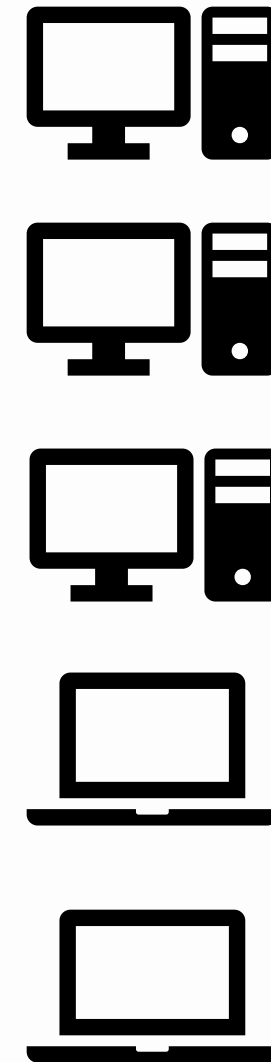
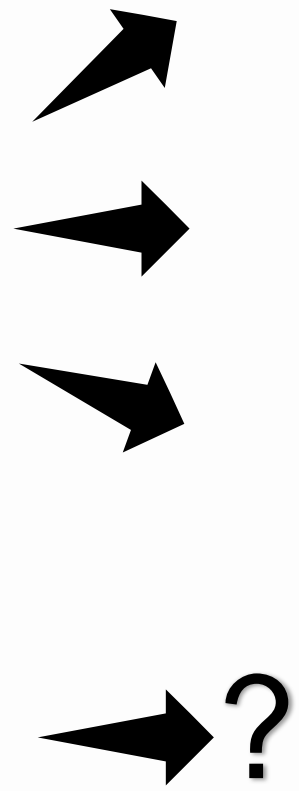
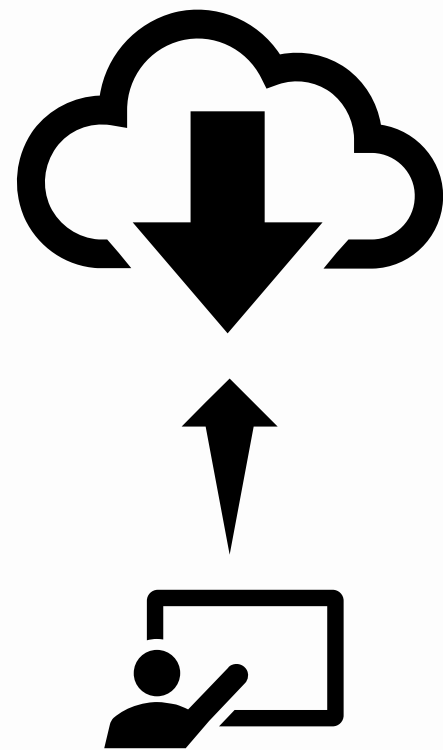


- Maintainers do not accept your patches
- You do not share patches
- A solution: FWTS is open source, and you can fork it freely

# Personal Package Archives (PPA)



Launchpad.net



x86

x64

ARM32

ARM64

POWER

NOTE: Source code are visible on public PPA

# FWTS Stable PPA



*<https://launchpad.net/~firmware-testing-team/+archive/ubuntu/ppa-fwts-stable>*

- `sudo add-apt-repository ppa:firmware-testing-team/ppa-fwts-stable`
- `sudo apt-get update`
- `sudo apt-get install fwts`

- Launchpad users can have PPA
- An example script for uploading to PPA:  
*[https://github.com/alexhungce/uefi\\_plugfest](https://github.com/alexhungce/uefi_plugfest)*



# Resources

- Documentation: <https://wiki.ubuntu.com/FirmwareTestSuite>
- Sample scripts: [https://github.com/alexhungce/uefi\\_plugfest](https://github.com/alexhungce/uefi_plugfest)
- Source code
  - *git clone git://kernel.ubuntu.com/hwe/fwts.git*
  - *git clone https://github.com/ColinIanKing/fwts*
- fwts-frontend-text (FWTS GUI)
  - Installation: `sudo apt install fwts-frontend`
    - Binary location: `/usr/bin/fwts-frontend-text`
  - Bash source code: `live-image/fwts-frontend-text`



Thanks for attending the Fall 2018  
UEFI Seminar and Plugfest

For more information on the Unified  
EFI Forum and UEFI Specifications,  
visit <http://www.uefi.org>

*presented by*

CANONICAL

