

NAME

fwts – a firmware test suite to identify firmware bugs.

SYNOPSIS

fwts [*options*] [*test(s)*]

DESCRIPTION

This manual page documents briefly the **fwts** firmware test suite. The tool **fwts** is comprised of over fifty tests that are designed to examine and test different aspects of PC firmware. Many of these tests need super user access to extract tables and interact with the firmware and ACPI, so running **fwts** using `sudo` is required.

Running **fwts** with no options will run through all the batch tests that require no user interaction. However, one can select just specific tests to run if required.

By default **fwts** outputs the test results into the log file **results.log** however a different log file name can be specified and if required, output to `stderr` or `stdout` can be selected.

Note that there a variety of tests, including tests that can potentially hang a machine (such as a suspend/hibernate/resume).

OPTIONS

fwts options are as follow:

– output results to `stdout`.

--acpica

enable ACPICA execution mode options. These can be specified as a comma separated list of one or more options. Available options are: `serialized` (serialized execution of AML), `slack` (run in less pedantic mode), `ignore-errors` (ignore ACPICA exception errors), `disable-auto-repair` (disable ACPICA from automatically fixing broken ACPICA controls). Note that the `slack` mode will turn on implicit returns of zero on control methods to attempt to allow buggy AML to work on non-Windows systems.

--acpica-debug

enable ACPICA debug warning and error messages when invoking the ACPICA subsystem. This is mainly for fwts developers to help track down any ACPICA interfacing issues with fwts.

-a, --all

run all the tests.

-b, --batch

run the non-interactive batch tests. Batch tests require no user interaction.

--batch-experimental

run only batch experimental tests.

--disassemble-aml

disassemble AML (ACPI machine language) byte code. This attempts to disassemble AML in DSDT and SSDT tables and generates `DSDT.dsl` and `SSDTx.dsl` sources.

-d, --dump

extracts firmware data and dumps it into log files. This generates:

`acpidump.log` – containing a hex dump of the ACPI tables (which can be read using `acpixtract`).

`dmesg.log` – containing the current kernel log messages.

`dmidecode.log` – containing the output from `dmidecode`.

`lspci.log` – containing the output from `lspci -vv -nn`

`cpuinfo.log` – containing the output from `cat /proc/cpuinfo`

`README.txt` – containing a timestamp and kernel version information.

- dumpfile=acpidump.log**
load ACPI tables from output generated from acpidump or from sudo fwts ---dump. The latter is preferred as fwts ---dump is able to dump more tables than acpidump. This allows one to dump tables from one machine and processes them with fwts on another machine.
- uefi-get-var-multiple**
specifies the number of times to get a variable in the uefirtvariable get variable stress test.
- uefi-set-var-multiple**
specifies the number of times to set a variable in the uefirtvariable set variable stress test.
- uefi-query-var-multiple**
specifies the number of times to query a variable in the uefirtvariable query variable stress test.
- filter-error-discard**
specifies the errors that one wants to silently ignore. One supplies a comma sperated list of fwts error message labels that one wants fwts to not report as errors. fwts will run the test but if there is a test failure and the label matches the one supplied in this list fwts will then just ignore this error. This cannot be used with ---filter-error-keep.
- filter-error-keep**
specifies the errors that one wants to keep, all other errors are silently ignored. One supplies a comma sperated list of fwts error message labels that one wants fwts report as errors, other test failures will be not reported and silently ignored. This cannot be used with ---filter-error-discard.
- f, ---force-clean**
creates a new results log file, rather than just appending to any existing one (default).
- h, ---help**
outputs the internal help page.
- i, ---interactive**
run the interactive tests. These tests require user interaction.
- interactive-experimental**
run only interactive experimental tests.
- j, ---json-data-path**
specifies the path to the fwts json data files. These files contain json formatted configuration tables, for example klog scanning patterns.
- k, ---klog=file**
read the kernel log from the specified file rather than from the kernel log ring buffer. This allows one to run the kernel log scanning tests such as klog against pre-gathered log data.
- log-fields**
show the available log filtering fields. Specifying these fields with ---log-filter to select which fields one wants to log.
- log-filter**
specify which particular types of log data to be output into the log file. Each line of log data is tagged with a special marker depending on what type of log information is being output. The available types can be see by using ---log-fields. Specify the desired log types with comma separated list. To disable a field, prefix the name with ~, for example:
---log-filter=RES,SUM logs just the results and summary lines.
---log-filter=ALL,~INF logs all lines except for the information lines.
- log-format**
specify the information in each log line. The following specifiers are available:
%date – date
%time – time
%field – log-filter fields
%owner – name of the test routine

%level – test failure level
 %line – log line
 e.g. `--log-format="%date %time [%field] (%owner): "`

- log-level [critical|high|medium|low|info|all]**
 specify the test failure level to log. Test failure levels equal to and higher than the specified are logged and recorded as failures. The default is 'all' (which is identical to 'info'). For example, a log level of 'medium' will just log test failures of level 'medium', 'high' and 'critical', where as a log level of 'critical' will just log 'critical' level failures.
- log-type**
 specify the log type. Currently plaintext, json and xml log types are available and the default is plaintext.
- lspci=path**
 specify the full path and filename to the the lspci binary.
- P, --power-states**
 run S3 and S4 power state tests (s3, s4 tests)
- results-no-separators**
 no pretty printing of horizontal separators in the results log file.
- r, --results-output=filename**
 specify the results output log file. One can also specify stdout and stderr to redirect to these output streams.
- R, --rsdp=physaddr**
 specify the physical address of ACPI RSDP. This is useful on some systems where it cannot be automatically detected.
- pm-method=method**
 specify the power method to use to enter S3 or S4 (or autodetection will be used). The following specifiers are available:
 logind – the default method, where available (requires dbus and logind).
 pm-utils – the previous default method, now deprecated.
 sysfs – the fallback, used when logind is not available.
 e.g. `--pm-method=sysfs`
- s3-delay-delta=N**
 time to be added onto delay between each S3 iteration.
- s3-device-check**
 check differences between device configurations over a S3 cycle. Note this adds 15 seconds delay after each s3 resume to allow wifi to re-associate.
- s3-device-check-delay**
 specify the time to wait while devices re-configure (e.g. wifi to re-associate, ethernet to connect..) before a device configuration check is run. The default is 15 seconds. If this option is used the device checking is assumed so one does not also need to use the `--s3-device-check` flag.
- s3-hybrid**
 enables fwts to run Hybrid Sleep.
- s3-min-delay=N**
 minimum time between S3 iterations.
- s3-max-delay=N**
 maximum time between S3 iterations.
- s3-multiple=N**
 specified the number of multiple S3 suspend/resume tests to run. The default is 2 tests.

- s3-quirks=--quirk[,--quirk]**
specify a comma separated list of quirk arguments to pass to pm-suspend, for example:
`--s3-quirks=--quirk-s3-bios,--quirk-save-pci`
- s3-sleep-delay=N**
sleep N seconds from the start of the suspend to the wakeup time. Note that this time **MUST** be longer than the time it takes to suspend the machine otherwise the wakeup timer will fire during the suspend state. The default is 30 seconds.
- s3-suspend-time=N**
specify the maximum allowed suspend time in seconds. If suspend takes longer than this then an error is logged.
- s3-resume-time=N**
specify the maximum allowed resume time in seconds. If resume takes longer than this then an error is logged.
- s3power-sleep-delay=N**
specify the suspend duration in seconds. The higher the value the more accurate the s3power test result. Durations less than 10 minutes are not recommended.
- s4-delay-delta=N**
time to be added onto delay between each S4 iteration.
- s4-device-check**
check differences between device configurations over a S4 cycle. Note this adds 15 seconds delay after each s3 resume to allow wifi to re-associate.
- s4-device-check-delay**
specify the time to wait while devices re-configure (e.g. wifi to re-associate, ethernet to connect..) before a device configuration check is run. The default is 15 seconds. If this option is used the device checking is assumed so one does not also need to use the `--s4-device-check` flag.
- s4-min-delay=N**
minimum time between S4 iterations.
- s4-max-delay=N**
maximum time between S4 iterations.
- s4-multiple=N**
specified the number of multiple S4 hibernate/resume tests to run. The default is 2 tests.
- s4-quirks=--quirk[,--quirk]**
specify a comma separated list of quirk arguments to pass to pm-hibernate, for example:
`--s4-quirks=--quirk-save-pci`
- s4-sleep-delay=N**
sleep N seconds from the start of the hibernate to the wakeup time. Note that this time **MUST** be longer than the time it takes to hibernate the machine otherwise the wakeup timer will fire during the hibernate state. The default is currently 90 seconds.
- p, --show-progress**
show the progress of the tests being run. Each test will identified as it is being run. For long tests, a percentage of completion time will be displayed. As of fwts 0.19.06 this is enabled by default and can be disabled with `--quiet` (or `-q`).
- q, --quiet**
run quietly with no output to stdout.
- D, --show-progress-dialog**
output the progress of tests being run in a form that can be piped into the dialog tool with the `--gauge` option.

- s, --show-tests**
show the names of available tests. By default will show all tests. Use the `--batch`, `--interactive`, `--batch-experimental`, `--interactive-experimental`, `--utils` options to show these specific tests.
- show-tests-full**
show all the available tests listed by minor test description. By default will show all tests. Use the `--batch`, `--interactive`, `--batch-experimental`, `--interactive-experimental` options to show these specific tests.
- show-tests-categories**
show all the available tests and the categories they belong to.
- skip-test=test[,test..]**
specify tests to skip over and not run. List must be comma separated.
- stdout-summary**
output SUCCESS or FAILED to stdout at end of tests.
- t, --table-path=path**
specify the path containing ACPI tables. These tables need to be named in the format: table-name.dat, for example DSDT.dat, for example, as extracted using `acpidump` or `fwts --dump` and then `acpixtract`.
- u, --utils**
run utilities. Designed to dump system information, such as annotated ACPI tables, CMOS memory, Int 15 E820 memory map, firmware ROM data.
- v, --version**
output version number and build date of the `fwts` tool.
- w, --width=N**
specify the width in characters of the output logfile. The default is 130.

EXAMPLES

- Run all the batch tests and append the results into the default log results.log:
`sudo fwts`
- Run all the interactive tests and start a clean results log called interactive.log:
`sudo fwts -i -f -r interactive.log`
- Run all the tests, interactive and batch:
`sudo fwts -i -b`
- Run just the battery and cpufreq tests:
`sudo fwts battery cpufreq`
- Run all the batch tests and define a new log format using just the date and line number:
`sudo fwts --log-format="%date %line: "`
- Run all the interactive tests and log just the results, info and summary data:
`sudo fwts -i --log-filter=RES,INF,SUM`
- Dump all the interesting firmware information into log files for analysis later:
`sudo fwts --dump`
- View kernel and ACPI driver version and BIOS information:
`sudo fwts -w 80 -r stdout version bios_info --log-filter=INF --log-format=""`
- Show the batch and batch experimental tests:
`fwts --show-tests --batch --batch-experimental`
- Run multiple S3 tests with delay between each test ranging from 1 second to 10 seconds with a delay delta per test of 0.2 seconds
`sudo fwts s3 --s3-multiple=100 --s3-min-delay=1 --s3-max-delay=10`

--s3-delay-delta=0.2

SEE ALSO

iasl(1), acpextract(1), acpidump(1), dmidecode(8), lspci(8)

AUTHOR

fwts was originally written by Colin King with much of the original test code derived from the Intel Linux Firmware test kit. Many thanks also for contributions from (in alphabetical order): AceLan Kao, Al Stone, Alberto Milone, Alex Hung, Anthony Wong, Chris Van Hoof, David Ward, Fu Wei, Heyi Guo, Ivan Hu, Jeremy Kerr, Jiri Vohanka, Kamal Mostafa, Keng-Yu Lin, Matt Fleming, Naresh Bhat, Phidias Chiang, Pradeep Gaddam, Ricardo Neri, Robert Hooker, Seth Forshee, Yang Kun, Yi Li and Zhang Rui.

This manual page was written by Colin King for the Ubuntu project (but may be used by others).

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.