



Linux Ubuntu 26.04 Manual Pages on command 'perf-annotate.1'

\$ man perf-annotate.1

PERF-ANNOTATE(1) perf Manual PERF-ANNOTATE(1)

NAME

perf-annotate - Read perf.data (created by perf record) and display annotated code

SYNOPSIS

perf annotate [-i <file> | --input=file] [symbol_name]

DESCRIPTION

This command reads the input file and displays an annotated version of the code. If the object file has debug symbols then the source code will be displayed alongside assembly code.

If there is no debug info in the object, then annotated assembly is displayed.

OPTIONS

-i, --input=<file>

Input file name. (default: perf.data unless stdin is a fifo)

-d, --dsos=<dso[,dso...]>

Only consider symbols in these dsos.

-s, --symbol=<symbol>

Symbol to annotate.

-f, --force

Don't do ownership validation.

-v, --verbose

Be more verbose. (Show symbol address, etc)

-q, --quiet

Do not show any warnings or messages. (Suppress -v)

-n, --show-nr-samples

Show the number of samples for each symbol

-D, --dump-raw-trace

Dump raw trace in ASCII.

-k, --vmlinux=<file>

vmlinux pathname.

--ignore-vmlinux

Ignore vmlinux files.

--itrace

Options for decoding instruction tracing data. The options are:

- i synthesize instructions events
- y synthesize cycles events
- b synthesize branches events
- c synthesize branches events (calls only)
- r synthesize branches events (returns only)
- x synthesize transactions events
- w synthesize ptwrite events
- p synthesize power events (incl. PSB events for Intel PT)
- o synthesize other events recorded due to the use
 of aux-output (refer to perf record)
- I synthesize interrupt or similar (asynchronous) events
 (e.g. Intel PT Event Trace)
- e synthesize error events
- d create a debug log
- f synthesize first level cache events
- m synthesize last level cache events
- M synthesize memory events
- t synthesize TLB events
- a synthesize remote access events
- g synthesize a call chain (use with i or x)
- G synthesize a call chain on existing event records
- I synthesize last branch entries (use with i or x)
- L synthesize last branch entries on existing event records

s skip initial number of events
q quicker (less detailed) decoding
A approximate IPC
Z prefer to ignore timestamps (so-called "timeless" decoding)
T use the timestamp trace as kernel time

The default is all events i.e. the same as `--itrace=iybxwpe`,
except for perf script where it is `--itrace=ce`

In addition, the period (default 100000, except for perf script where it is 1)
for instructions events can be specified in units of:

i instructions
t ticks
ms milliseconds
us microseconds
ns nanoseconds (default)

Also the call chain size (default 16, max. 1024) for instructions or
transactions events can be specified.

Also the number of last branch entries (default 64, max. 1024) for
instructions or transactions events can be specified.

Similar to options g and l, size may also be specified for options G and L.

On x86, note that G and L work poorly when data has been recorded with
large PEBS. Refer `linkperf:perf-intel-pt[1]` man page for details.

It is also possible to skip events generated (instructions, branches, transactions,
ptwrite, power) at the beginning. This is useful to ignore initialization code.

`--itrace=i0nss1000000`

skips the first million instructions.

The 'e' option may be followed by flags which affect what errors will or
will not be reported. Each flag must be preceded by either '+' or '-'.

The flags are:

o overflow
l trace data lost

If supported, the 'd' option may be followed by flags which affect what
debug messages will or will not be logged. Each flag must be preceded
by either '+' or '-'. The flags are:

- a all perf events
- e output only on errors (size configurable - see linkperf:perf-config[1])
- o output to stdout

If supported, the 'q' option may be repeated to increase the effect.

To disable decoding entirely, use --no-itrace.

-m, --modules

Load module symbols. WARNING: use only with -k and LIVE kernel.

-l, --print-line

Print matching source lines (may be slow).

-P, --full-paths

Don't shorten the displayed pathnames.

--stdio

Use the stdio interface.

--stdio2

Use the stdio2 interface, non-interactive, uses the TUI formatting.

--stdio-color=<mode>

always, never or auto, allowing configuring color output via the command line, in addition to via "color.ui" .perfconfig. Use --stdio-color always to generate color even when redirecting to a pipe or file. Using just --stdio-color is equivalent to using always.

--tui

Use the TUI interface. Use of --tui requires a tty, if one is not present, as when piping to other commands, the stdio interface is used. This interfaces starts by centering on the line with more samples, TAB/UNTAB cycles through the lines with more samples.

--gtk

Use the GTK interface.

-C, --cpu=<cpu>

Only report samples for the list of CPUs provided. Multiple CPUs can be provided as a comma-separated list with no space: 0,1. Ranges of CPUs are specified with -: 0-2.

Default is to report samples on all CPUs.

--asm-raw

Show raw instruction encoding of assembly instructions.

--show-total-period

Show a column with the sum of periods.

--source

Interleave source code with assembly code. Enabled by default, disable with --no-source.

--symfs=<directory>

Look for files with symbols relative to this directory.

-M, --disassembler-style=

Set disassembler style for objdump.

--addr2line=<path>

Path to addr2line binary.

--objdump=<path>

Path to objdump binary.

--prefix=PREFIX, --prefix-strip=N

Remove first N entries from source file path names in executables and add PREFIX. This allows to display source code compiled on systems with different file system layout.

--skip-missing

Skip symbols that cannot be annotated.

--group

Show event group information together

--demangle

Demangle symbol names to human readable form. It's enabled by default, disable with --no-demangle.

--demangle-kernel

Demangle kernel symbol names to human readable form (for C++ kernels).

--percent-type

Set annotation percent type from following choices: global-period, local-period, global-hits, local-hits

The local/global keywords set if the percentage is computed in the scope of the function (local) or the whole data (global).

The period/hits keywords set the base the percentage is computed on - the samples period or the number of samples (hits).

--percent-limit

Do not show functions which have an overhead under that percent on stdio or stdio2

(Default: 0). Note that this is about selection of functions to display, not about lines within the function.

`--data-type[=TYPE_NAME]`

Display data type annotation instead of code. It infers data type of samples (if they are memory accessing instructions) using DWARF debug information. It can take an optional argument of data type name. In that case it'd show annotation for the type only, otherwise it'd show all data types it finds.

`--type-stat`

Show stats for the data type annotation.

`--skip-empty`

Do not display empty (or dummy) events.

`--code-with-type`

Show data type info in code annotation (for memory instructions only).

SEE ALSO

`perf-record(1)`, `perf-report(1)`

`perf`

05/25/2026

PERF-ANNOTATE(1)