



Linux Ubuntu 26.04 Manual Pages on command 'trace-cmd-restore.1'

\$ man trace-cmd-restore.1

TRACE-CMD-RESTORE(1) libtracefs Manual TRACE-CMD-RESTORE(1)

NAME

trace-cmd-restore - restore a failed trace record

SYNOPSIS

trace-cmd restore [OPTIONS] [command] cpu-file [cpu-file ...]

DESCRIPTION

The trace-cmd(1) restore command will restore a crashed trace-cmd-record(1) file. If for some reason a trace-cmd record fails, it will leave a the per-cpu data files and not create the final trace.dat file. The trace-cmd restore will append the files to create a working trace.dat file that can be read with trace-cmd-report(1).

When trace-cmd record runs, it spawns off a process per CPU and writes to a per cpu file usually called trace.dat.cpuX, where X represents the CPU number that it is tracing. If the -o option was used in the trace-cmd record, then the CPU data files will have that name instead of the trace.dat name. If a unexpected crash occurs before the tracing is finished, then the per CPU files will still exist but there will not be any trace.dat file to read from. trace-cmd restore will allow you to create a trace.dat file with the existing data files.

OPTIONS

-c

Create a partial trace.dat file from the machine, to be used with a full trace-cmd restore at another time. This option is useful for embedded devices. If a server contains the cpu files of a crashed trace-cmd record (or trace-cmd listen), trace-cmd restore can be executed on the embedded device with the -c option to get all the stored

information of that embedded device. Then the file created could be copied to the server to run the trace-cmd restore there with the cpu files.

If *-o* is not specified, then the file created will be called 'trace-partial.dat'. This is because the file is not a full version of something that trace-cmd-report(1) could use.

-t tracing_dir

Used with -c, it overrides the location to read the events from. By default, tracing information is read from the debugfs/tracing directory. -t will use that location instead. This can be useful if the trace.dat file to create is from another machine. Just tar -cvf events.tar debugfs/tracing and copy and untar that file locally, and use that directory instead.

-k kallsyms

Used with -c, it overrides where to read the kallsyms file from. By default, /proc/kallsyms is used. -k will override the file to read the kallsyms from. This can be useful if the trace.dat file to create is from another machine. Just copy the /proc/kallsyms file locally, and use -k to point to that file.

-o output'

By default, trace-cmd restore will create a trace.dat file (or trace-partial.dat if -c is specified). You can specify a different file to write to with the -o option.

-i input

By default, trace-cmd restore will read the information of the current system to create the initial data stored in the trace.dat file. If the crash was on another machine, then that machine should have the trace-cmd restore run with the -c option to create the trace.dat partial file. Then that file can be copied to the current machine where trace-cmd restore will use -i to load that file instead of reading from the current system.

EXAMPLES

If a crash happened on another box, you could run:

```
$ trace-cmd restore -c -o box-partial.dat
```

Then on the server that has the cpu files:

```
$ trace-cmd restore -i box-partial.dat trace.dat.cpu0 trace.dat.cpu1
```

This would create a trace.dat file for the embedded box.

trace-cmd(1), trace-cmd-record(1), trace-cmd-report(1), trace-cmd-start(1),
trace-cmd-stop(1), trace-cmd-extract(1), trace-cmd-reset(1), trace-cmd-split(1),
trace-cmd-list(1), trace-cmd-listen(1)

AUTHOR

Written by Steven Rostedt, <rostedt@goodmis.org[1]>

RESOURCES

<https://git.kernel.org/pub/scm/utils/trace-cmd/trace-cmd.git/>

COPYING

Copyright (C) 2010 Red Hat, Inc. Free use of this software is granted under the terms of the
GNU Public License (GPL).

NOTES

1. rostedt@goodmis.org

mailto:rostedt@goodmis.org

libtracefs

01/22/2026

TRACE-CMD-RESTORE(1)