



## ***Linux Ubuntu 26.04 Manual Pages on command 'tracecmd\_buffer\_instance\_handle.3'***

***\$ man tracecmd\_buffer\_instance\_handle.3***

LIBTRACECMD(3)                      libtracefs Manual                      LIBTRACECMD(3)

### NAME

tracecmd\_buffer\_instances, tracecmd\_buffer\_instance\_name, tracecmd\_buffer\_instance\_handle -  
Read tracing instances from a trace file.

### SYNOPSIS

```
#include <trace-cmd.h>

int tracecmd_buffer_instances(struct tracecmd_input *handle);

const char *tracecmd_buffer_instance_name(struct tracecmd_input *handle, int indx);

struct tracecmd_input *tracecmd_buffer_instance_handle(struct tracecmd_input *handle, int indx);
```

### DESCRIPTION

This set of APIs can be used to get information and read tracing data from tracing instances stored in a trace file.

The tracecmd\_buffer\_instances() function gets the number of tracing instances recorded in a trace file. The top instance is not counted. The handle is a tracecmd\_input handler returned by tracecmd\_open\_head().

The tracecmd\_buffer\_instance\_name() function gets the name of the tracing instance with given index indx, recorded in a trace file. The indx is a number in the interval [0 ..

count-1], where count is the number returned by `tracecmd_buffer_instances()`. The handle is a `tracecmd_input` handler returned by `tracecmd_open_head()`.

The `tracecmd_buffer_instance_handle()` allocates and initializes a `tracecmd_input` handle, associated with trace instance with index `indx` from a trace file. The handle is a `tracecmd_input` handler returned by `tracecmd_open_head()`. The `indx` is a number in the interval [0 .. count-1], where count is the number returned by `tracecmd_buffer_instances()`.

## RETURN VALUE

The `tracecmd_buffer_instances()` function returns the number of tracing instances recorded in a trace file.

The `tracecmd_buffer_instance_name()` function returns a string, the name of a tracing instance, or NULL in case of an error. The string must not be freed.

The `tracecmd_buffer_instance_handle()` function returns a pointer to newly allocated `tracecmd_input` handler or NULL in case of an error. The returned handler must be closed by `tracecmd_close()`(3)

## EXAMPLE

```
#include <trace-cmd.h>

...

struct tracecmd_input *handle = tracecmd_open_head("trace.dat");

if (!handle) {
    /* Failed to open trace.dat file */
}

...

int num = tracecmd_buffer_instances(handle);

while(num) {
    struct tracecmd_input *h;
    char *name;
```

```

    name = tracecmd_buffer_instance_name(handle, num);
    if (!name) {
        /* Failed to get name of instance num */
    }
    h = tracecmd_buffer_instance_handle(handle, num);
    if (!h) {
        /* Failed to initialize handler for instance num */
    }

    ...

    tracecmd_close(h);
    num--;
}

...

tracecmd_close(handle);

```

## FILES

trace-cmd.h

Header file to include in order to have access to the library APIs.

-ltracecmd

Linker switch to add when building a program that uses the library.

## SEE ALSO

libtracefs(3), libtraceevent(3), trace-cmd(1) trace-cmd.dat(5)

## AUTHOR

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

Tzvetomir Stoyanov <tz.stoyanov@gmail.com[2]>

## REPORTING BUGS

Report bugs to <linux-trace-devel@vger.kernel.org[3]>

## LICENSE

libtracecmd is Free Software licensed under the GNU LGPL 2.1

## RESOURCES

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

## COPYING

Copyright (C) 2020 VMware, 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)

<mailto:rostedt@goodmis.org>

2. [tz.stoyanov@gmail.com](mailto:tz.stoyanov@gmail.com)

<mailto:tz.stoyanov@gmail.com>

3. [linux-trace-devel@vger.kernel.org](mailto:linux-trace-devel@vger.kernel.org)

<mailto:linux-trace-devel@vger.kernel.org>

libtracefs

01/22/2026

LIBTRACECMD(3)