



Linux Ubuntu 26.04 Manual Pages on command 'gprofng-collect-app.1'

\$ man gprofng-collect-app.1

GPROFNG-COLLECT-APP(1)

User Commands

GPROFNG-COLLECT-APP(1)

NAME

gprofng-collect-app - Collect performance data for the target application

SYNOPSIS

gprofng collect app [option(s)] target [target-option(s)]

DESCRIPTION

Collect performance data on the target program. In addition to Program Counter (PC)

sampling, hardware event counters and various tracing options are supported.

For example, this command collects performance data for an executable called a.out and

stores the data collected in an experiment directory with the name example.er.

```
$ gprofng collect app -o example.er ./a.out
```

OPTIONS

--version

Print the version number and exit.

--help

Print usage information and exit.

-v, --verbose

By default, verbose mode is disabled. This option enables it.

-p {off | on | lo[w] | hi[gh] | <value>}

Disable (off) or enable (on) clock profiling using a default sampling granularity, or

enable clock profiling implicitly by setting the sampling granularity (lo[w], hi[gh], or

a specific value in ms). By default, clock profiling is enabled (-p on).

-h <ctr_def>[,<ctr_def>]

Enable hardware event counter profiling and select one or more counter(s). To see the supported counters on this system, use the -h option without other arguments.

-o <exp_name>

Specify the name for the experiment directory. The name has to end with .er and may contain an absolute path (e.g. /tmp/experiment.er). An existing experiment with the same name will not be overwritten.

-O <exp_name>

This is the same as the -o option, but unlike this option, silently overwrites an existing experiment directory with the same name.

-C <comment_string>

Add up to 10 comment strings to the experiment. These comments appear in the notes section of the header and can be retrieved with the gprofng display text command using the -header option.

-j {on | off | <path>}

Controls Java profiling when the target is a JVM machine. The allowed values for this option are:

on Record profiling data for the JVM machine, and recognize methods compiled by the Java HotSpot virtual machine. Also record Java call stacks.

off Do not record Java profiling data. Profiling data for native call stacks is still recorded.

<path>

Records profiling data for the JVM, and use the JVM as installed in <path>.

The default is -j on.

-J <jvm-option(s)>

Specifies one or more additional options to be passed to the JVM used. The jvm-option(s) list must be enclosed in quotation marks if it contains more than one option. The items in the list need to be separated by spaces or tabs. Each item is passed as a separate option to the JVM. Note that this option implies -j on.

-t <duration>[m|s]

Collects data for the specified duration. The duration can be a single number, optionally followed by either m to specify minutes, or s to specify seconds, which is the default.

The duration can also consists of two numbers separated by a minus (-) sign. If a

single number is given, data is collected from the start of the run until the given time. If two numbers are given, data is collected from the first time to the second. In case the second time is zero, data is collected until the end of the run. If two non-zero numbers are given, the first must be less than the second.

-n This is used for a dry run. Several run-time settings are displayed, but the target is not executed and no performance data is collected.

-F {off|on|=regex}

Control whether descendant processes should have their data recorded. To disable/enable this feature, use off/on. Use =regex to record data on those processes whose executable name matches the regular expression. Only the basename of the executable is used, not the full path. If spaces or characters interpreted by the shell are used, enclose the regex in single quotes. The default is -F on.

-a {off|on|ldobjects|src|usedldobjects|usedsrc}

Specify archiving of binaries and other files. In addition to disable this feature (off), or enable archiving off all loadobjects and sources (on), the other options support a more refined selection.

All of these options enable archiving, but the keyword controls what exactly is selected: all load objects (ldobjects), all source files (src), the loadobjects associated with a program counter (usedldobjects), or the source files associated with a program counter (usedsrc). The default is -a ldobjects.

-S {off|on|<seconds>}

Disable (off), or enable (on) periodic sampling of process-wide resource utilization.

By default, sampling occurs every second. Use the <seconds> option to change this. The default is -S on.

-y <signal>[,r]

Controls recording of data with the signal named <signal>, referred to as the pause-resume signal. Whenever the given signal is delivered to the process, switch between paused (no data is recorded) and resumed (data is recorded) states.

By default, data collection begins in the paused state. If the optional r is given, data collection begins in the resumed state and data collection begins immediately.

SIGUSR1 or SIGUSR2 are recommended for this use, but any signal that is not used by the target can be used.

-l <signal>

Specify a signal that will trigger a sample of process-wide resource utilization. When the named <signal> is delivered to the process, a sample is recorded.

The signal can be specified using the full name, without the initial letters "SIG", or the signal number. Note that the kill command can be used to deliver a signal.

If both the -l and -y options are used, the signal must be different.

-s <option>[,<API>]

Enable synchronization wait tracing, where <option> is used to define the specifics of the tracing (on, off, <threshold>, or all). The API is selected through the setting for <API>: n selects native/Pthreads, j selects Java, and nj selects both. The default is -s off.

-H {off|on}

Disable (off), or enable (on) heap tracing. The default is -H off.

-i {off|on}

Disable (off), or enable (on) I/O tracing. The default is -i off.

NOTES

Any executable in the ELF (Executable and Linkable Format) object format can be used for profiling with gprofng. If debug information is available, gprofng can provide more details, but this is not a requirement.

SEE ALSO

gprofng(1), gprofng-archive(1), gprofng-display-html(1), gprofng-display-src(1),
gprofng-display-text(1)

The user guide for gprofng is maintained as a Texinfo manual. If the info and gprofng programs are correctly installed, the command info gprofng should give access to this document.

COPYRIGHT

Copyright (c) 2022-2026 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".