

NAME

coredump.conf, coredump.conf.d – Core dump storage configuration files

SYNOPSIS

```
/etc/systemd/coredump.conf
/run/systemd/coredump.conf
/usr/local/lib/systemd/coredump.conf
/usr/lib/systemd/coredump.conf
/etc/systemd/coredump.conf.d/*.conf
/run/systemd/coredump.conf.d/*.conf
/usr/local/lib/systemd/coredump.conf.d/*.conf
/usr/lib/systemd/coredump.conf.d/*.conf
```

DESCRIPTION

These files configure the behavior of **systemd-coredump(8)**, a handler for core dumps invoked by the kernel. Whether **systemd-coredump** is used is determined by the kernel's *kernel.core_pattern* **sysctl(8)** setting. See **systemd-coredump(8)** and **core(5)** pages for the details.

CONFIGURATION DIRECTORIES AND PRECEDENCE

The default configuration is set during compilation, so configuration is only needed when it is necessary to deviate from those defaults. The main configuration file is loaded from one of the listed directories in order of priority, only the first file found is used: `/etc/systemd/`, `/run/systemd/`, `/usr/local/lib/systemd/` ^[1], `/usr/lib/systemd/`. The vendor version of the file contains commented out entries showing the defaults as a guide to the administrator. Local overrides can also be created by creating drop-ins, as described below. The main configuration file can also be edited for this purpose (or a copy in `/etc/` if it is shipped under `/usr/`), however using drop-ins for local configuration is recommended over modifications to the main configuration file.

In addition to the main configuration file, drop-in configuration snippets are read from `/usr/lib/systemd/*.conf.d/`, `/usr/local/lib/systemd/*.conf.d/`, and `/etc/systemd/*.conf.d/`. Those drop-ins have higher precedence and override the main configuration file. Files in the `/*.conf.d/` configuration subdirectories are sorted by their filename in lexicographic order, regardless of in which of the subdirectories they reside. When multiple files specify the same option, for options which accept just a single value, the entry in the file sorted last takes precedence, and for options which accept a list of values, entries are collected as they occur in the sorted files.

When packages need to customize the configuration, they can install drop-ins under `/usr/`. Files in `/etc/` are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. Drop-ins have to be used to override package drop-ins, since the main configuration file has lower precedence. It is recommended to prefix all filenames in those subdirectories with a two-digit number and a dash, to simplify the ordering. This also defines a concept of drop-in priorities to allow OS vendors to ship drop-ins within a specific range lower than the range used by users. This should lower the risk of package drop-ins overriding accidentally drop-ins defined by users. It is recommended to use the range 10–40 for drop-ins in `/usr/` and the range 60–90 for drop-ins in `/etc/` and `/run/`, to make sure that local and transient drop-ins take priority over drop-ins shipped by the OS vendor.

To disable a configuration file supplied by the vendor, the recommended way is to place a symlink to `/dev/null` in the configuration directory in `/etc/`, with the same filename as the vendor configuration file.

OPTIONS

All options are configured in the `[Coredump]` section:

Storage=

Controls where to store cores. One of "none", "external", and "journal". When "none", the core dumps may be logged (including the backtrace if possible), but not stored permanently. When "external" (the default), cores will be stored in `/var/lib/systemd/coredump/`. When "journal", cores will be stored in the journal and rotated following normal journal rotation patterns.

When cores are stored in the journal, they might be compressed following journal compression

settings, see **journal**.conf(5). When cores are stored externally, they will be compressed by default, see below.

Note that in order to process a coredump (i.e. extract a stack trace) the core must be written to disk first. Thus, unless *ProcessSizeMax*= is set to 0 (see below), the core will be written to */var/lib/systemd/coredump/* either way (under a temporary filename, or even in an unlinked file), *Storage*= thus only controls whether to leave it there even after it was processed.

Added in version 215.

Compress=

Controls compression for external storage. Takes a boolean argument, which defaults to "yes".

Added in version 215.

ProcessSizeMax=

The maximum size in bytes of a core which will be processed. Core dumps exceeding this size may be stored, but the stack trace will not be generated. Like other sizes in this same config file, the usual suffixes to the base of 1024 are allowed (B, K, M, G, T, P, and E). Defaults to 1G on 32-bit systems, 32G on 64-bit systems.

Setting *Storage=none* and *ProcessSizeMax=0* disables all coredump handling except for a log entry.

Added in version 215.

EnterNamespace=

For processes belonging to a PID namespace, controls whether **systemd-coredump**(8) shall attempt to process core dumps on the host, using debug information from the file system hierarchy (i.e. the mount namespace) of the process that crashed. Access to the process' file system hierarchy might be necessary to generate a fully symbolized backtrace. If set to "yes", **systemd-coredump** will obtain the tree of mounts from the crashing process' mount namespace and will try to generate the stack trace in host context using the debug information of binaries and libraries contained in the crashing process' hierarchy. Defaults to "no", i.e. no attempt is made to acquire external debug information from the process' mount namespace, in order to maximize security. This option has no effect on processes that are part of the host's PID namespace.

Note that the coredump of the namespaced process is still saved in */var/lib/systemd/coredump/* on the host even if *EnterNamespace*= is set to "no" (subject to *Storage*=).

Note that *EnterNamespace*= only has an effect if a core dump is generated by a container whose unit does not have *CoredumpReceive*= enabled.

Note that it's typically preferable to let containers and other namespace-based sandboxes process their own coredumps, if possible, for best security. This may be enabled on the container's unit via the *CoredumpReceive*= setting, see **systemd.resource-control**(5) for details.

Added in version 257.

ExternalSizeMax=, *JournalSizeMax*=

The maximum (compressed or uncompressed) size in bytes of a coredump to be saved in separate files on disk (default: 1G on 32-bit systems, 32G on 64-bit systems) or in the journal (default: 767M). Note that the journal service enforces a hard limit on journal log records of 767M, and will ignore larger submitted log records. Hence, *JournalSizeMax*= may be lowered relative to the default, but not increased. Unit suffixes are allowed just as in **ProcessSizeMax**=.

ExternalSizeMax=infinity sets the core size to unlimited.

Added in version 215.

MaxUse=, KeepFree=

Enforce limits on the disk space, specified in bytes, taken up by externally stored core dumps. Unit suffixes are allowed just as in **ProcessSizeMax=**. **MaxUse=** makes sure that old core dumps are removed as soon as the total disk space taken up by core dumps grows beyond this limit (defaults to 10% of the total disk size, capped at 4GiB, minimum 1MiB). **KeepFree=** controls how much disk space to keep free at least (defaults to 15% of the total disk size, limited to a maximum of 4GiB). Note that the disk space used by core dumps might temporarily exceed these limits while core dumps are processed. **MaxUse=** takes precedence. Note that old core dumps are also removed based on time via **systemd-tmpfiles(8)**. Set either value to 0 to turn off size-based cleanup.

Added in version 215.

Use

`systemd-analyze cat-config systemd/coredump.conf`

to display the full config.

SEE ALSO

systemd-journald.service(8), **coredumpctl(1)**, **systemd-tmpfiles(8)**

NOTES

1. Please note that those configuration files must be available at all times. If `/usr/local/` is a separate partition, it may not be available during early boot, and must not be used for configuration.