NAME

systemd-tmpfiles, systemd-tmpfiles-setup.service, systemd-tmpfiles-setup-dev.service, systemd-tmpfiles-clean.service, systemd-tmpfiles-clean.timer − Creates, deletes and cleans up volatile and temporary files and directories

SYNOPSIS

systemd−tmpfiles [OPTIONS...] [*CONFIGFILE*...]

System units:

systemd−tmpfiles−setup.service
systemd−tmpfiles−setup−dev.service
systemd−tmpfiles−clean.service
systemd−tmpfiles−clean.timer

User units:

systemd−tmpfiles−setup.service
systemd−tmpfiles−clean.service
systemd−tmpfiles−clean.timer

DESCRIPTION

systemd−tmpfiles creates, deletes, and cleans up volatile and temporary files and directories, using the configuration file format and location specified in **tmpfiles.d**(5). It must be invoked with one or more options **−−create**, **−−remove**, and **−−clean**, to select the respective subset of operations.

By default, directives from all configuration files are applied. When invoked with **−−replace=***PATH*, arguments specified on the command line are used instead of the configuration file *PATH*. Otherwise, if one or more absolute filenames are passed on the command line, only the directives in these files are applied. If "−" is specified instead of a filename, directives are read from standard input. If only the basename of a configuration file is specified, all configuration directories as specified in **tmpfiles.d**(5) are searched for a matching file and the file found that has the highest priority is executed.

System services (systemd−tmpfiles−setup.service, systemd−tmpfiles−setup−dev.service, systemd−tmpfiles−clean.service) invoke **systemd−tmpfiles** to create system files and to perform system wide cleanup. Those services read administrator−controlled configuration files in tmpfiles.d/ directories. User services (systemd−tmpfiles−setup.service, systemd−tmpfiles−clean.service) also invoke **systemd−tmpfiles**, but it reads a separate set of files, which includes user−controlled files under ~/.config/user−tmpfiles.d/ and ~/.local/share/user−tmpfiles.d/, and administrator−controlled files under /usr/share/user−tmpfiles.d/. Users may use this to create and clean up files under their control, but the system instance performs global cleanup and is not influenced by user configuration. Note that this means a time−based cleanup configured in the system instance, such as the one typically configured for /tmp/, will thus also affect files created by the user instance if they are placed in /tmp/, even if the user instance's time−based cleanup is turned off.

To re−apply settings after configuration has been modified, simply restart systemd−tmpfiles−clean.service, which will apply any settings which can be safely executed at runtime. To debug **systemd−tmpfiles**, it may be useful to invoke it directly from the command line with increased log level (see *$SYSTEMD_LOG_LEVEL* below).

OPTIONS

The following options are understood:

−−create
: If this option is passed, all files and directories marked with *f*, *F*, *w*, *d*, *D*, *v*, *p*, *L*, *c*, *b*, *m* in the configuration files are created or written to. Files and directories marked with *z*, *Z*, *t*, *T*, *a*, and *A* have their ownership, access mode and security labels set.

−−clean
: If this option is passed, all files and directories with an age parameter configured will be cleaned up.

−−remove
If this option is passed, the contents of directories marked with *D* or *R*, and files or directories themselves marked with *r* or *R* are removed.

−−user
Execute "user" configuration, i.e. tmpfiles.d files in user configuration directories.

−−boot
Also execute lines with an exclamation mark.

−−prefix=*path*
Only apply rules with paths that start with the specified prefix. This option can be specified multiple times.

−−exclude−prefix=*path*
Ignore rules with paths that start with the specified prefix. This option can be specified multiple times.

−E
A shortcut for "−−exclude−prefix=/dev −−exclude−prefix=/proc −−exclude−prefix=/run −−exclude−prefix=/sys", i.e. exclude the hierarchies typically backed by virtual or memory file systems. This is useful in combination with **−−root=**, if the specified directory tree contains an OS tree without these virtual/memory file systems mounted in, as it is typically not desirable to create any files and directories below these subdirectories if they are supposed to be overmounted during runtime.

−−root=*root*
Takes a directory path as an argument. All paths will be prefixed with the given alternate *root* path, including config search paths.

When this option is used, the libc Name Service Switch (NSS) is bypassed for resolving users and groups. Instead the files /etc/passwd and /etc/group inside the alternate root are read directly. This means that users/groups not listed in these files will not be resolved, i.e. LDAP NIS and other complex databases are not considered.

Consider combining this with **−E** to ensure the invocation does not create files or directories below mount points in the OS image operated on that are typically overmounted during runtime.

−−image=*image*
Takes a path to a disk image file or block device node. If specified all operations are applied to file system in the indicated disk image. This is similar to **−−root=** but operates on file systems stored in disk images or block devices. The disk image should either contain just a file system or a set of file systems within a GPT partition table, following the **Discoverable Partitions Specification**[1]. For further information on supported disk images, see **systemd-nspawn**(1)'s switch of the same name.

Implies **−E**.

−−replace=*PATH*
When this option is given, one ore more positional arguments must be specified. All configuration files found in the directories listed in **tmpfiles.d**(5) will be read, and the configuration given on the command line will be handled instead of and with the same priority as the configuration file *PATH*.

This option is intended to be used when package installation scripts are running and files belonging to that package are not yet available on disk, so their contents must be given on the command line, but the admin configuration might already exist and should be given higher priority.

−−cat−config
Copy the contents of config files to standard output. Before each file, the filename is printed as a comment.

−−no−pager
Do not pipe output into a pager.

−h, **−−help**
Print a short help text and exit.

−−version
Print a short version string and exit.

It is possible to combine **−−create**, **−−clean**, and **−−remove** in one invocation (in which case removal and cleanup are executed before creation of new files). For example, during boot the following command line is executed to ensure that all temporary and volatile directories are removed and created according to the configuration file:

systemd−tmpfiles −−remove −−create

ENVIRONMENT

$SYSTEMD_LOG_LEVEL
The maximum log level of emitted messages (messages with a higher log level, i.e. less important ones, will be suppressed). Either one of (in order of decreasing importance) **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, **debug**, or an integer in the range 0...7. See **syslog**(3) for more information.

$SYSTEMD_LOG_COLOR
A boolean. If true, messages written to the tty will be colored according to priority.

This setting is only useful when messages are written directly to the terminal, because **journalctl**(1) and other tools that display logs will color messages based on the log level on their own.

$SYSTEMD_LOG_TIME
A boolean. If true, console log messages will be prefixed with a timestamp.

This setting is only useful when messages are written directly to the terminal or a file, because **journalctl**(1) and other tools that display logs will attach timestamps based on the entry metadata on their own.

$SYSTEMD_LOG_LOCATION
A boolean. If true, messages will be prefixed with a filename and line number in the source code where the message originates.

Note that the log location is often attached as metadata to journal entries anyway. Including it directly in the message text can nevertheless be convenient when debugging programs.

$SYSTEMD_LOG_TARGET
The destination for log messages. One of **console** (log to the attached tty), **console−prefixed** (log to the attached tty but with prefixes encoding the log level and "facility", see **syslog**(3), **kmsg** (log to the kernel circular log buffer), **journal** (log to the journal), **journal−or−kmsg** (log to the journal if available, and to kmsg otherwise), **auto** (determine the appropriate log target automatically, the default), **null** (disable log output).

$SYSTEMD_PAGER
Pager to use when **−−no−pager** is not given; overrides *$PAGER*. If neither *$SYSTEMD_PAGER* nor *$PAGER* are set, a set of well−known pager implementations are tried in turn, including **less**(1) and **more**(1), until one is found. If no pager implementation is discovered no pager is invoked. Setting this environment variable to an empty string or the value "cat" is equivalent to passing **−−no−pager**.

$SYSTEMD_LESS
Override the options passed to **less** (by default "FRSXMK").

Users might want to change two options in particular:

K
This option instructs the pager to exit immediately when Ctrl+C is pressed. To allow **less** to handle Ctrl+C itself to switch back to the pager command prompt, unset this option.

If the value of *$SYSTEMD_LESS* does not include "K", and the pager that is invoked is **less**, Ctrl+C will be ignored by the executable, and needs to be handled by the pager.

X

This option instructs the pager to not send termcap initialization and deinitialization strings to the terminal. It is set by default to allow command output to remain visible in the terminal even after the pager exits. Nevertheless, this prevents some pager functionality from working, in particular paged output cannot be scrolled with the mouse.

See **less**(1) for more discussion.

$SYSTEMD_LESSCHARSET

Override the charset passed to **less** (by default "utf−8", if the invoking terminal is determined to be UTF−8 compatible).

$SYSTEMD_PAGERSECURE

Takes a boolean argument. When true, the "secure" mode of the pager is enabled; if false, disabled. If *$SYSTEMD_PAGERSECURE* is not set at all, secure mode is enabled if the effective UID is not the same as the owner of the login session, see **geteuid**(2) and **sd_pid_get_owner_uid**(3). In secure mode, **LESSSECURE=1** will be set when invoking the pager, and the pager shall disable commands that open or create new files or start new subprocesses. When *$SYSTEMD_PAGERSECURE* is not set at all, pagers which are not known to implement secure mode will not be used. (Currently only **less**(1) implements secure mode.)

Note: when commands are invoked with elevated privileges, for example under **sudo**(8) or **pkexec**(1), care must be taken to ensure that unintended interactive features are not enabled. "Secure" mode for the pager may be enabled automatically as describe above. Setting *SYSTEMD_PAGERSECURE=0* or not removing it from the inherited environment allows the user to invoke arbitrary commands. Note that if the *$SYSTEMD_PAGER* or *$PAGER* variables are to be honoured, *$SYSTEMD_PAGERSECURE* must be set too. It might be reasonable to completely disable the pager using **−−no−pager** instead.

$SYSTEMD_COLORS

Takes a boolean argument. When true, **systemd** and related utilities will use colors in their output, otherwise the output will be monochrome. Additionally, the variable can take one of the following special values: "16", "256" to restrict the use of colors to the base 16 or 256 ANSI colors, respectively. This can be specified to override the automatic decision based on *$TERM* and what the console is connected to.

$SYSTEMD_URLIFY

The value must be a boolean. Controls whether clickable links should be generated in the output for terminal emulators supporting this. This can be specified to override the decision that **systemd** makes based on *$TERM* and other conditions.

UNPRIVILEGED −−CLEANUP OPERATION

systemd−tmpfiles tries to avoid changing the access and modification times on the directories it accesses, which requires **CAP_FOWNER** privileges. When running as non−root, directories which are checked for files to clean up will have their access time bumped, which might prevent their cleanup.

EXIT STATUS

On success, 0 is returned. If the configuration was syntactically invalid (syntax errors, missing arguments, ...), so some lines had to be ignored, but no other errors occurred, **65** is returned (**EX_DATAERR** from /usr/include/sysexits.h). If the configuration was syntactically valid, but could not be executed (lack of permissions, creation of files in missing directories, invalid contents when writing to /sys/ values, ...), **73** is returned (**EX_CANTCREAT** from /usr/include/sysexits.h). Otherwise, **1** is returned (**EXIT_FAILURE** from /usr/include/stdlib.h).

Note: when creating items, if the target already exists, but is of the wrong type or otherwise does not match the requested state, and forced operation has not been requested with "+", a message is emitted, but the

failure is otherwise ignored.

SEE ALSO

systemd(1), **tmpfiles.d**(5)

NOTES

1. Discoverable Partitions Specification
   https://systemd.io/DISCOVERABLE_PARTITIONS