Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'systemd.scope.5' command

$ man systemd.scope.5

SYSTEMD.SCOPE(5) systemd.scope SYSTEMD.SCOPE(5)

NAME

systemd.scope - Scope unit configuration

SYNOPSIS

scope.scope

DESCRIPTION

Scope units are not configured via unit configuration files, but are only created programmatically using the bus interfaces of systemd. They are named similar to filenames. A unit whose name ends in ".scope" refers to a scope unit. Scopes units manage a set of system processes. Unlike service units, scope units manage externally created processes, and do not fork off processes on its own.

The main purpose of scope units is grouping worker processes of a system service for organization and for managing resources.

systemd-run --scope may be used to easily launch a command in a new scope unit from the command line.

See the New Control Group Interfaces[1] for an introduction on how to make use of scope units from programs.

Note that, unlike service units, scope units have no "main" process: all processes in the scope are equivalent. The lifecycle of the scope unit is thus not bound to the lifetime of one specific process, but to the existence of at least one process in the scope. This also means that the exit statuses of these processes are not relevant for the

scope unit failure state. Scope units may still enter a failure state, for example due to resource exhaustion or stop timeouts being reached, but not due to programs inside of them terminating uncleanly. Since processes managed as scope units generally remain children of the original process that forked them off, it is also the job of that process to collect their exit statuses and act on them as needed.

AUTOMATIC DEPENDENCIES

Implicit Dependencies

Implicit dependencies may be added as result of resource control parameters as documented in systemd.resource-control(5).

Default Dependencies

The following dependencies are added unless DefaultDependencies=no is set:

? Scope units will automatically have dependencies of type Conflicts= and Before= on shutdown.target. These ensure that scope units are removed prior to system shutdown. Only scope units involved with early boot or late system shutdown should disable DefaultDependencies= option.

OPTIONS

Scope files may include a [Unit] section, which is described in systemd.unit(5).

Scope files may include a [Scope] section, which carries information about the scope and the units it contains. A number of options that may be used in this section are shared with other unit types. These options are documented in systemd.kill(5) and systemd.resource-control(5). The options specific to the [Scope] section of scope units are the following:

OOMPolicy=

Configure the out-of-memory (OOM) killing policy for the kernel and the userspace OOM killer systemd-oomd.service(8). On Linux, when memory becomes scarce to the point that the kernel has trouble allocating memory for itself, it might decide to kill a running process in order to free up memory and reduce memory pressure. Note

that systemd-oomd.service is a more flexible solution that aims to prevent out-of-memory situations for the userspace too, not just the kernel, by attempting to terminate services earlier, before the kernel would have to act.

This setting takes one of continue, stop or kill. If set to continue and a process in the unit is killed by the OOM killer, this is logged but the unit continues running. If set to stop the event is logged but the unit is terminated cleanly by the service manager. If set to kill and one of the unit's processes is killed by the OOM killer the kernel is instructed to kill all remaining processes of the unit too, by setting the memory.oom.group attribute to 1; also see kernel documentation[2].

Defaults to the setting DefaultOOMPolicy= in systemd-system.conf(5) is set to, except for units where Delegate= is turned on, where it defaults to continue.

Use the OOMScoreAdjust= setting to configure whether processes of the unit shall be considered preferred or less preferred candidates for process termination by the Linux OOM killer logic. See systemd.exec(5) for details.

This setting also applies to systemd-oomd. Similarly to the kernel OOM kills, this setting determines the state of the unit after systemd-oomd kills a cgroup associated with it.

RuntimeMaxSec=

Configures a maximum time for the scope to run. If this is used and the scope has been active for longer than the specified time it is terminated and put into a failure state. Pass "infinity" (the default) to configure no runtime limit.

RuntimeRandomizedExtraSec=

This option modifies RuntimeMaxSec= by increasing the maximum runtime by an evenly distributed duration between 0 and the specified value (in seconds). If RuntimeMaxSec= is unspecified, then this feature will be disabled.

Check systemd.unit(5), systemd.exec(5), and systemd.kill(5) for more

settings.

SEE ALSO

systemd(1), systemd-run(1), systemd.unit(5), systemd.resource-control(5), systemd.service(5), systemd.directives(7).

NOTES

1. New Control Group Interfaces

   https://www.freedesktop.org/wiki/Software/systemd/ControlGroupInterface

2. kernel documentation

   https://docs.kernel.org/admin-guide/cgroup-v2.html