freedesktop.org

www/ Software/ systemd/ Incompatibilities

Edit Page History Repo Info

Back to systemd

Compatibility with SysV

systemd provides a fair degree of compatibility with the behavior exposed by the SysV init system as implemented by many distributions. Compatibility is provided both for the user experience and the SysV scripting APIs. However, there are some areas where compatibility is limited due to technical reasons or design decisions of systemd and the distributions. All of the following applies to SysV init scripts handled by systemd, however a number of them matter only on specific distributions. Many of the incompatibilities are specific to distribution-specific extensions of LSB/SysV init.

- If your distribution removes SysV init scripts in favor of systemd unit files typing "/etc/init.d/foobar start" to start a service will not work, since the script will not be available. Use the more correct "/sbin/service foobar start" instead, and your command will be forwarded to systemd. Note that invoking the init script directly has always been suboptimal since too much of the caller's execution context (environment block, umask, resource limits, audit trails, ...) ended up being inherited by the service, and invocation via "/sbin/service" used to clean this up at least partially. Invocation via /sbin/service works on both SysV and systemd systems. Also, LSB only standardizes invocation via "/sbin/service" anyway. (Note that some distributions ship both systemd unit files and SysV scripts for the services. For these invoking the init scripts will work as expected and the request be forwarded to systemd in any case.)
- LSB header dependency information matters. The SysV implementations on many distributions did not use the dependency information encoded in LSB init script headers, or used them only in very limited ways. Due to that they are often incorrect or incomplete. systemd however fully interprets these headers and follows them closely at runtime (and not at installation time like some implementations).
- Timeouts apply to all init script operations in systemd. While on SysV systems a hanging init script could freeze the system on systemd all init script operations are subject to a timeout of 5min.
- Services are executed in completely clean execution contexts, no context of the invoking user session is inherited. Not even $HOME or similar are set. Init scripts depending on these will not work correctly.
- Services cannot read from stdin, as this will be connected to /dev/null. That means interactive init scripts are not supported (i.e.

Debian's X-Interactive in the LSB header is not supported either.) Thankfully most distributions do not support interaction in init scripts anyway. If you need interaction to ask disk or SSL passphrases please consider using the minimal password querying framework systemd supports. (details, manual page)

- Additional verbs for init scripts are not supported. If your init script traditionally supported additional verbs for your init script simply move them to an auxiliary script.
- Additional parameters to the standard verbs (i.e. to "start", "stop" and "status") are not supported. This was an extension of SysV that never was standardized officially, and is not supported in systemd.
- systemd only stops running services. On traditional SysV a K link installed for shutdown was executed when going down regardless whether the service was started before or not. systemd is more strict here and does not stop service that weren't started in the first place.
- Runlevels are supported in a limited fashion only. SysV runlevels are mapped to systemd target units, however not all systemd target units map back to SysV runlevels. This is due to the fact that systemd targets are a lot more flexible and expressive than SysV runlevels. That means that checks for the current runlevel (with /sbin/runlevel or so) may well return "N" (i.e. unknown runlevel) during normal operation. Scripts that rely on explicit runlevel checks are incompatible with many setups. Avoid runlevel checks like these.
- Tools like /sbin/chkconfig might return misleading information when used to list enablement status of services. First of all, the tool will only see SysV services, not native units. Secondly, it will only show runlevel-related information (which does not fully map to systemd targets). Finally, the information shown might be overridden by a native unit file.
- By default runlevels 2,3,4 are all aliases for "multi-user.target". If a service is enabled in one of these runlevels, they'll be enabled in all of these. This is only a default however, and users can easily override the mappings, and split them up into individual runlevels if they want. However, we recommend moving on from runlevels and using the much more expressive target units of systemd.
- Early boot runlevels as they are used by some distributions are no longer supported. i.e. "fake", distribution-specific runlevels such as "S" or "b" cannot be used with systemd.
- By default, System V services are not permitted to acquire realtime scheduling, even with root privileges. See MyServiceCantGetRealtime for details, and what to do about it.

Note that there are some areas where systemd currently provides a certain amount of compatibility where we expect this compatibility to be removed eventually.

Last edited Sun Oct 6 14:37:19 2013