systemd.v

- Description
- Filtering and Sorting
- Example
- See Also

systemd

systemd.v

7

systemd.v

Directory with Versioned Resources

Description

In various places systemd components accept paths whose trailing components have the `.v/` suffix, pointing to a directory. These components will then automatically look for suitable files inside the directory, do a version comparison and open the newest file found (by version). Available since version v256. Specifically, two expressions are supported:

- When looking for files with a suffix <.SUFFIX>, and a path `…PATH/NAME.SUFFIX.v/` is specified, then all files `…PATH/NAME.SUFFIX.v/NAME_*.SUFFIX` are enumerated, filtered, sorted and the newest file used. The primary sorting key is the *variable part*, here indicated by the wildcard `*`.
- When a path `…PATH.v/NAME___.SUFFIX` is specified (i.e. the penultimate component of the path ends in `.v` and the final component contains a triple underscore), then all files `…PATH.v/NAME_*.SUFFIX` are enumerated, filtered, sorted and the newest file used (again, by the *variable part*, here indicated by the wildcard `*`).

To illustrate this in an example, consider a directory `/var/lib/machines/mymachine.raw.v/`, which is populated with three files:

- `mymachine_7.5.13.raw`
- `mymachine_7.5.14.raw`
- `mymachine_7.6.0.raw`

Invoke a tool such as `systemd-nspawn(1)` with a command line like the following:

```
# systemd-nspawn --image=/var/lib/machines/mymachine.raw.v --boot
```

Then this would automatically be resolved to the equivalent of:

```
# systemd-nspawn --image=/var/lib/machines/mymachine.raw.v/mymachine_7.6.0.raw --boot
```

Much of systemd's functionality that expects a path to a disk image or OS directory hierarchy support the `.v/` versioned directory mechanism, for example `systemd-nspawn(1)`, `systemd-dissect(1)` or the `RootDirectory=`/`RootImage=` settings of service files (see `systemd.exec(5)`).

Use the `systemd-vpick(1)` tool to resolve `.v/` paths from the command line, for example for usage in shell scripts.

Filtering and Sorting

The variable part of the filenames in the `.v/` directories are filtered and compared primarily with a version comparison, implementing UAPI.10 Version Format Specification. However, additional rules apply:

- If the variable part is suffixed by one or two integer values ("tries left" and "tries done") in the formats `+LEFT` or `+LEFT-DONE`, then these indicate usage attempt counters. The idea is that each time before a file is attempted to be used, its "tries left" counter is decreased, and the "tries done" counter increased (simply by renaming the file). When the file is successfully used (which for example could mean for an OS image: successfully booted) the counters are removed from the file name, indicating that the file has been validated to work correctly. This mechanism mirrors the boot assessment counters defined by Automatic Boot Assessment. Any filenames with no boot counters or with a non-zero "tries left" counter are sorted before filenames with a zero "tries left" counter.
- Preceding the use counters (if they are specified), an optional CPU architecture identifier may be specified in the filename (separated from the version with an underscore), as defined in the architecture vocabulary of the `ConditionArchitecture=` unit file setting, as documented in `systemd.unit(5)`. Files whose name indicates an architecture not supported locally are filtered and not considered for the version comparison.
- The rest of the variable part is the version string.

Or in other words, the files in the `.v/` directories should follow one of these naming structures:

- `NAME_VERSION.SUFFIX`
- `NAME_VERSION_ARCHITECTURE.SUFFIX`
- `NAME_VERSION+LEFT.SUFFIX`
- `NAME_VERSION+LEFT-DONE.SUFFIX`
- `NAME_VERSION_ARCHITECTURE+LEFT.SUFFIX`
- `NAME_VERSION_ARCHITECTURE+LEFT-DONE.SUFFIX`

Example

Here's a more comprehensive example, further extending the one described above. Consider a directory `/var/lib/machines/mymachine.raw.v/`, which is populated with the following files:

- `mymachine_7.5.13.raw`
- `mymachine_7.5.14_x86-64.raw`
- `mymachine_7.6.0_arm64.raw`
- `mymachine_7.7.0_x86-64+0-5.raw`

Now invoke the following command on an x86-64 machine:

```
$ systemd-vpick --suffix=.raw /var/lib/machines/mymachine.raw.v/
```

This would resolve the specified path to `/var/lib/machines/mymachine.raw.v/mymachine_7.5.14_x86-64.raw`. Explanation: even though `mymachine_7.7.0_x86-64+0-5.raw` has the newest version, it is not preferred because its tries left counter is zero. And even though `mymachine_7.6.0_arm64.raw` has the second newest version it is also not considered in this case, because we operate on an x86_64 system and the image is intended for arm64 CPUs. Finally, the `mymachine_7.5.13.raw` image is not considered because it is older than `mymachine_7.5.14_x86-64.raw`.

See Also

`systemd(1)`, `systemd-vpick(1)`, `systemd-nspawn(1)`, `systemd-dissect(1)`, `systemd.exec(5)`, `systemd-sysupdate(8)`

□□□ #2
□ □□□ □□□ 2026-10-31 15:07:59 UTC
□ □□□ □□□ 2026-10-31 15:08:00 UTC