

NAME

environment.d – Definition of user session environment

SYNOPSIS

~/config/environment.d/*.conf

/etc/environment.d/*.conf

/run/environment.d/*.conf

/usr/lib/environment.d/*.conf

/etc/environment

DESCRIPTION

The environment.d directories contain a list of environment variable assignments for services started by the systemd user instance. **systemd-environment-d-generator**(8) parses them and updates the environment exported by the systemd user instance. See below for an discussion of which processes inherit those variables.

It is recommended to use numerical prefixes for file names to simplify ordering.

For backwards compatibility, a symlink to /etc/environment is installed, so this file is also parsed.

CONFIGURATION DIRECTORIES AND PRECEDENCE

Configuration files are read from directories in /etc/, /run/, /usr/local/lib/, and /lib/, in order of precedence, as listed in the SYNOPSIS section above. Files must have the ".conf" extension. Files in /etc/ override files with the same name in /run/, /usr/local/lib/, and /lib/. Files in /run/ override files with the same name under /usr/.

All configuration files are sorted by their filename in lexicographic order, regardless of which of the directories they reside in. If multiple files specify the same option, the entry in the file with the lexicographically latest name will take precedence. Thus, the configuration in a certain file may either be replaced completely (by placing a file with the same name in a directory with higher priority), or individual settings might be changed (by specifying additional settings in a file with a different name that is ordered later).

Packages should install their configuration files in /usr/lib/ (distribution packages) or /usr/local/lib/ (local installs). Files in /etc/ are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. It is recommended to prefix all filenames with a two-digit number and a dash, to simplify the ordering of the files.

If the administrator wants 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. If the vendor configuration file is included in the initrd image, the image has to be regenerated.

CONFIGURATION FORMAT

The configuration files contain a list of "*KEY=VALUE*" environment variable assignments, separated by newlines. The right hand side of these assignments may reference previously defined environment variables, using the "\${OTHER_KEY}" and "\$OTHER_KEY" format. It is also possible to use "\${FOO:-DEFAULT_VALUE}" to expand in the same way as "\${FOO}" unless the expansion would be empty, in which case it expands to *DEFAULT_VALUE*, and use "\${FOO:+ALTERNATE_VALUE}" to expand to *ALTERNATE_VALUE* as long as "\${FOO}" would have expanded to a non-empty value. No other elements of shell syntax are supported.

Each *KEY* must be a valid variable name. Empty lines and lines beginning with the comment character "#" are ignored.

Example

Example 1. Setup environment to allow access to a program installed in /opt/foo

/etc/environment.d/60-foo.conf:

```
FOO_DEBUG=force--software-gl,log-verbose
PATH=/opt/foo/bin:$PATH
LD_LIBRARY_PATH=/opt/foo/lib${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
XDG_DATA_DIRS=/opt/foo/share:${XDG_DATA_DIRS:-/usr/local/share:/usr/share/}
```

APPLICABILITY

Environment variables exported by the user manager (**systemd --user** instance started in the `user@uid.service` system service) apply to any services started by that manager. In particular, this may include services which run user shells. For example in the Gnome environment, the graphical terminal emulator runs as the `gnome-terminal-server.service` user unit, which in turn runs the user shell, so that shell will inherit environment variables exported by the user manager. For other instances of the shell, not launched by the user manager, the environment they inherit is defined by the program that starts them. Hint: in general, **systemd.service**(5) units contain programs launched by `systemd`, and **systemd.scope**(5) units contain programs launched by something else.

Specifically, for ssh logins, the **sshd**(8) service builds an environment that is a combination of variables forwarded from the remote system and defined by **sshd**, see the discussion in **ssh**(1). A graphical display session will have an analogous mechanism to define the environment. Note that some managers query the `systemd` user instance for the exported environment and inject this configuration into programs they start, using **systemctl show--environment** or the underlying D-Bus call.

SEE ALSO

systemd(1), **systemd-environment-d-generator**(8), **systemd.environment-generator**(7)