NAME

systemd.preset − Service enablement presets

SYNOPSIS

/etc/systemd/system−preset/*.preset

/run/systemd/system−preset/*.preset

/lib/systemd/system−preset/*.preset

/etc/systemd/user−preset/*.preset

/run/systemd/user−preset/*.preset

/usr/lib/systemd/user−preset/*.preset

DESCRIPTION

Preset files may be used to encode policy which units shall be enabled by default and which ones shall be disabled. They are read by **systemctl preset** which uses this information to enable or disable a unit. Depending on that policy, **systemctl preset** is identical to **systemctl enable** or **systemctl disable**. **systemctl preset** is used by the post install scriptlets of rpm packages (or other OS package formats), to enable/disable specific units by default on package installation, enforcing distribution, spin or administrator preset policy. This allows choosing a certain set of units to be enabled/disabled even before installing the actual package. For more information, see **systemctl**(1).

It is not recommended to ship preset files within the respective software packages implementing the units, but rather centralize them in a distribution or spin default policy, which can be amended by administrator policy, see below.

If no preset files exist, **systemctl preset** will enable all units that are installed by default. If this is not desired and all units shall rather be disabled, it is necessary to ship a preset file with a single, catchall "disable *" line. (See example 1, below.)

PRESET FILE FORMAT

The preset files contain a list of directives consisting of either the word "enable" or "disable" followed by a space and a unit name (possibly with shell style wildcards), separated by newlines. Empty lines and lines whose first non−whitespace character is "#" or ";" are ignored. Multiple instance names for unit templates may be specified as a space separated list at the end of the line instead of the customary position between "@" and the unit suffix.

Presets must refer to the "real" unit file, and not to any aliases. See **systemd.unit**(5) for a description of unit aliasing.

Two different directives are understood: "enable" may be used to enable units by default, "disable" to disable units by default.

If multiple lines apply to a unit name, the first matching one takes precedence over all others.

Each preset file shall be named in the style of <priority>−<policy−name>.preset. Files in /etc/ override files with the same name in /usr/lib/ and /run/. Files in /run/ override files with the same name in /lib/. Packages should install their preset files in /lib/. Files in /etc/ are reserved for the local administrator, who may use this logic to override the preset files installed by vendor packages. All preset files are sorted by their filename in lexicographic order, regardless of which of the directories they reside in. If multiple files specify the same unit name, the entry in the file with the lexicographically earliest name will be applied. 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 preset file supplied by the vendor, the recommended way is to place a symlink to /dev/null in /etc/systemd/system−preset/ bearing the same filename.

EXAMPLES

Example 1. Default to off

/lib/systemd/system−preset/99−default.preset

disable *

This disables all units. Due to the filename prefix "99−", it will be read last and hence can easily be overridden by spin or administrator preset policy.

Example 2. Enable multiple template instances

```
# /lib/systemd/system−preset/80−dirsrv.preset

enable dirsrv@.service foo bar baz
```

This enables all three of dirsrv@foo.service, dirsrv@bar.service and dirsrv@baz.service.

Example 3. A GNOME spin

```
# /lib/systemd/system−preset/50−gnome.preset

enable gdm.service
enable colord.service
enable accounts−daemon.service
enable avahi−daemon.*
```

This enables the three mentioned units, plus all avahi−daemon regardless of which unit type. A file like this could be useful for inclusion in a GNOME spin of a distribution. It will ensure that the units necessary for GNOME are properly enabled as they are installed. It leaves all other units untouched, and subject to other (later) preset files, for example like the one from the first example above.

Example 4. Administrator policy

```
# /etc/systemd/system−preset/00−lennart.preset

enable httpd.service
enable sshd.service
enable postfix.service
disable *
```

This enables three specific services and disables all others. This is useful for administrators to specifically select the units to enable, and disable all others. Due to the filename prefix "00−" it will be read early and override all other preset policy files.

MOTIVATION FOR THE PRESET LOGIC

Different distributions have different policies on which services shall be enabled by default when the package they are shipped in is installed. On Fedora all services stay off by default, so that installing a package will not cause a service to be enabled (with some exceptions). On Debian all services are immediately enabled by default, so that installing a package will cause its services to be enabled right−away.

Even within a single distribution, different spins (flavours, remixes, whatever you might want to call them) of a distribution also have different policies on what services to enable, and what services to leave off. For example, Fedora Workstation will enable **gdm** as display manager by default, while the Fedora KDE spin will enable **sddm** instead.

Different sites might also have different policies what to turn on by default and what to turn off. For example, one administrator would prefer to enforce the policy of "**sshd** should be always on, but everything else off", while another one might say "**snmpd** always on, and for everything else use the distribution policy defaults".

Traditionally, policy about which services shall be enabled were implemented in each package individually. This made it cumbersome to implement different policies per spin or per site, or to create software packages that do the right thing on more than one distribution. The enablement mechanism was also encoding the enablement policy.

The preset mechanism allows clean separation of the enablement mechanism (inside the package scriptlets, by invoking **systemctl preset**) and enablement policy (centralized in the preset files), and lifts the configuration out of individual packages. Preset files may be written for specific distributions, for specific spins or for specific sites, in order to enforce different policies as needed. It is recommended to apply the policy encoded in preset files in package installation scriptlets.

SEE ALSO

systemd(1), **systemctl**(1), **systemd-delta**(1)

daemon(7) has a discussion of packaging scriptlets.

Fedora page introducing the use of presets: **Features/PackagePresets**[1].

NOTES

1. Features/PackagePresets
   https://fedoraproject.org/wiki/Features/PackagePresets