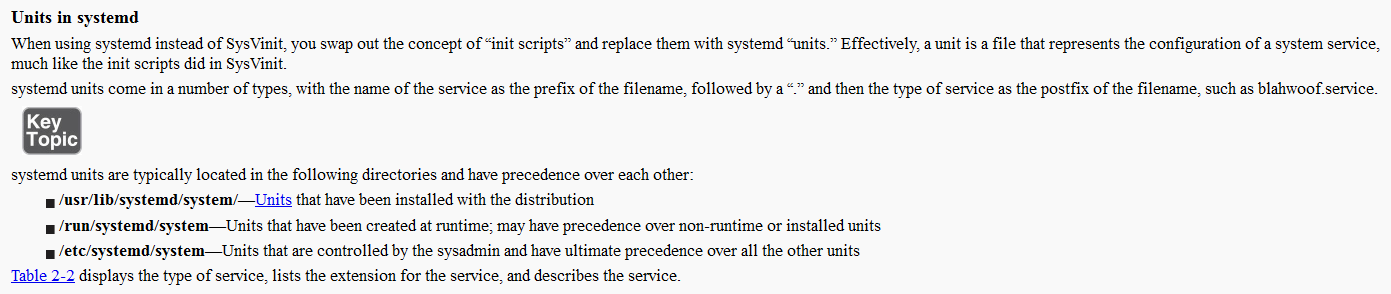
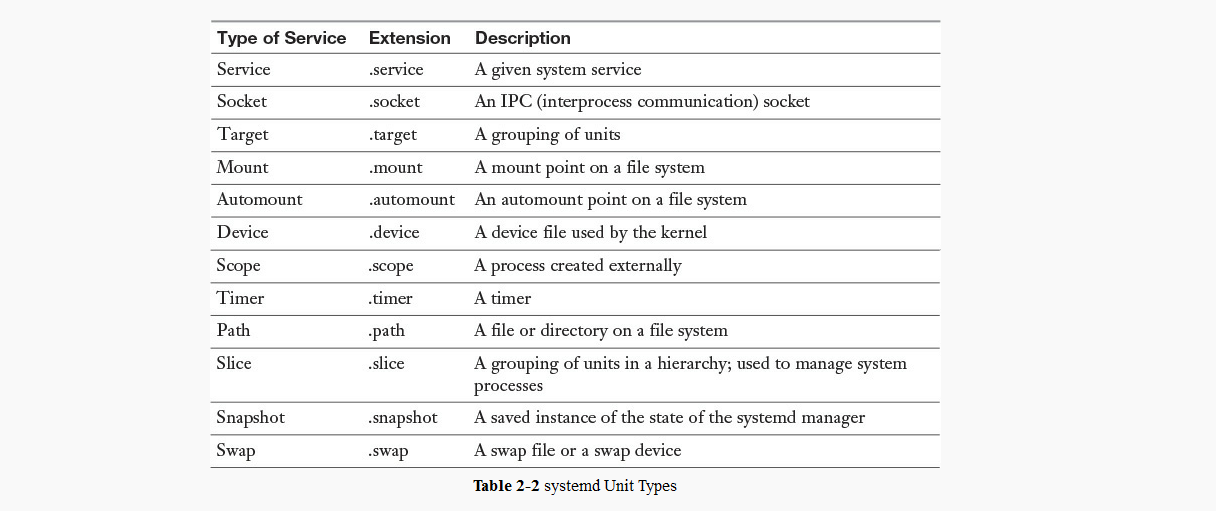
/etc/system/system > /run/system/system > /usr/lib/systemd/system

Units are the objects that systemd knows how to manage. These are basically a standardized representation of system resources that can be managed by the suite of daemons and manipulated by the provided utilities.





**Wants and Requires**

The systemd procedure uses a set of options or requirements called “wants” and “requires” to indicate dependencies between units and groups of units.

For example, if you were to look at the typical contents of the /usr/lib/system/system/graphical.target file, you might find the following lines:

Requires=multi-user.target  
After=multi-user.target  
Wants=display-manager.service

### [Unit] Section Directives

The first section found in most unit files is the [Unit] section. This is generally used for defining metadata for the unit and configuring the relationship of the unit to other units.

Although section order does not matter to systemd when parsing the file, this section is often placed at the top because it provides an overview of the unit. Some common directives that you will find in the [Unit] section are:

**Requires=**: This directive lists any units upon which this unit essentially depends. If the current unit is activated, the units listed here must successfully activate as well, else this unit will fail. These units are started in parallel with the current unit by default.

**Wants=**: This directive is similar to Requires=, but less strict. Systemd will attempt to start any units listed here when this unit is activated. If these units are not found or fail to start, the current unit will continue to function. This is the recommended way to configure most dependency relationships. Again, this implies a parallel activation unless modified by other directives.

 **Before=**: The units listed in this directive will not be started until the current unit is marked as started if they are activated at the same time. This does not imply a dependency relationship and must be used in conjunction with one of the above directives if this is desired.

 **After=**: The units listed in this directive will be started before starting the current unit. This does not imply a dependency relationship and one must be established through the above directives if this is required.

 **Conflicts=**: This can be used to list units that cannot be run at the same time as the current unit. Starting a unit with this relationship will cause the other units to be stopped. After=multi-user.target => multi-user.target is started first, then /usr/lib/system/system/graphical.target is started

### [Install] Section Directives

On the opposite side of unit file, the last section is often the [Install] section. This section is optional and is used to define the behavior or a unit if it is enabled or disabled. Enabling a unit marks it to be automatically started at boot. In essence, this is accomplished by latching the unit in question onto another unit that is somewhere in the line of units to be started at boot.

Because of this, only units that can be enabled will have this section. The directives within dictate what should happen when the unit is enabled:

 **WantedBy=**: The WantedBy= directive is the most common way to specify how a unit should be enabled. This directive allows you to specify a dependency relationship in a similar way to the Wants= directive does in the [Unit] section. The difference is that this directive is included in the ancillary unit allowing the primary unit listed to remain relatively clean. When a unit with this directive is enabled, a directory will be created within /etc/systemd/system named after the specified unit with .wants appended to the end. Within this, a symbolic link to the current unit will be created, creating the dependency. For instance, if the current unit has WantedBy=multi-user.target, a directory called multi-user.target.wants will be created within /etc/systemd/system (if not already available) and a symbolic link to the current unit will be placed within. Disabling this unit removes the link and removes the dependency relationship.

 **RequiredBy=**: This directive is very similar to the WantedBy= directive, but instead specifies a required dependency that will cause the activation to fail if not met. When enabled, a unit with this directive will create a directory ending with .requires.

Each unit requires a configuration file that defines what program it starts and how it should  
start the program. The systemd system stores unit configuration files in the /lib/systemd  
/system folder.  
Here’s an example of the sshd.service unit configuration file used in Fedora 20:  
# **cat sshd.service**  
[Unit]  
Description=OpenSSH server daemon  
After=syslog.target network.target auditd.service  
[Service]  
EnvironmentFile=/etc/sysconfig/sshd  
ExecStartPre=/usr/sbin/sshd-keygen  
ExecStart=/usr/sbin/sshd -D $OPTIONS  
ExecReload=/bin/kill -HUP $MAINPID  
KillMode=process  
Restart=on-failure  
RestartSec=42s  
[Install]  
WantedBy=multi-user.target  
#

The sshd.service configuration file defines the program to start (/usr/sbin/sshd),  
along with some other features, such as what services should run before the sshd service  
starts (the After line), what target level the system should be in (the WantedBy line), and  
how to reload the program (the Restart line).