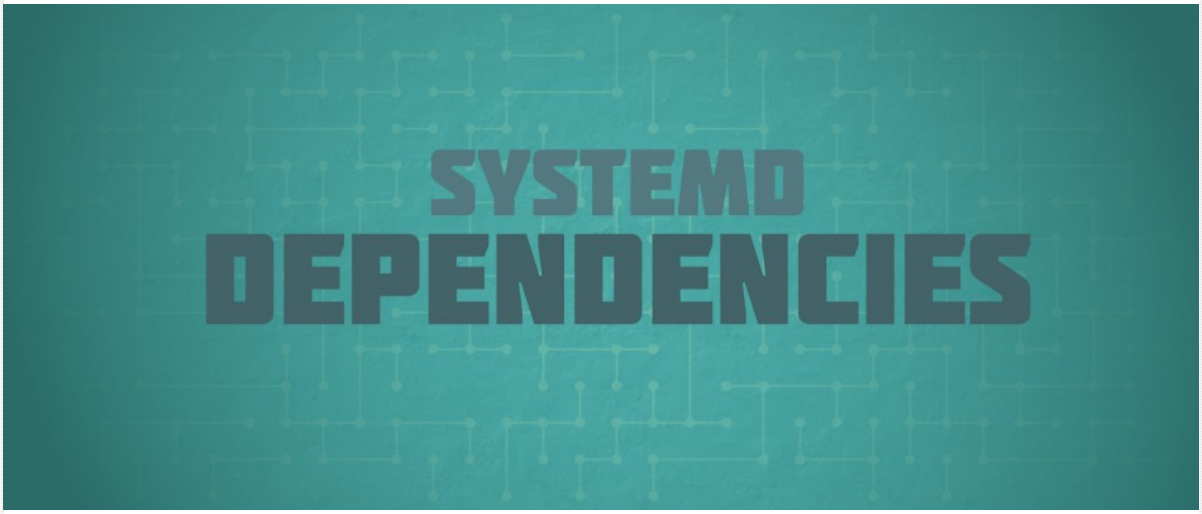
systemd: Unit dependencies and order



Welcome back to our continuing series on systemd features. As you've guessed from our previous articles, systemd brings more power and flexibility to service startup and management. One of the most important changes in systemd from legacy SysVinit is how it starts up units.

You may have heard from casual users that systemd starts everything together. Some people believe this is true, and that's why the system starts faster. But the reality is not quite that simple. Let's look a little more deeply at how systemd understands unit relationships.

Unit dependencies

Unit files include the feature of *dependencies*. Any unit may *want* or *require* one or more other units before it can run. These dependencies are set in unit files with the directives *Wants* and *Requires*. The difference between these is simple.

- If unit1 has *Wants=unit2* as a dependency, when unit1 is run, unit2 will be run as well. But whether unit2 starts successfully does not affect unit1 running

successfully.

- When unit1 has *Requires=unit2*, however, again both units will run, but if unit2 does not succeed, unit1 is also deactivated. This happens regardless of whether the processes of unit1 would otherwise have worked fine.

Did you notice something subtle about this description? It doesn't talk about order. When systemd starts your system, it loads all unit files and reads through them to determine dependencies like this. When unit1 runs in these examples, unit2 is run **at the same time**. It's important to know that dependencies and ordering are two different things to systemd.

Here's an example of dependencies in part of the *sshd.service* unit file:

```
Wants=sshd-keygen.service
```

So when *sshd.service* runs, the *sshd-keygen.service* will run as well. However, *sshd-keygen.service* does not need to succeed for *sshd.service* to run successfully.

Why would a dependency like this need to exist? In this case, the *sshd-keygen.service* creates new SSH keys for a server if they do not exist. We always want to check whether these keys exist before the SSH server starts. If they already exist, the service unit exits with an error indicating so. But in that case of course, we still want the SSH server to run as usual.

Unit order

This doesn't mean systemd can't put things in proper order. Without any other instructions, systemd would run a group of units at the same time. This is probably why some people believe systemd starts everything at the same time (or "in parallel"). It is sometimes necessary, of course, for processes to run in a certain order.

Fortunately, systemd also has unit directives for this issue as well, *Before* and *After*. These directives work pretty much as you'd expect:

- If unit1 has the directive *Before=unit2,* then if both units are run, unit1 will be executed fully before unit2 starts.
- If unit1 has the directive *After=unit2,* then if both units are run, unit2 will be executed fully before unit1 starts.

Once again, note how this ordering does not affect dependencies. Neither case causes unit2 to run. Let's look at the *sshd.service* unit again:

```
Wants=sshd-keygen.service
```

```
After=network.target sshd-keygen.service
```

The ordering directive *After* ensures that the SSH server will not run until after the host key generation unit, and after the network is up. (The *network.target* unit ensures that various units bring up the network.) Dependencies like *Wants* and *Requires* are often used together with *After* to preserve proper dependencies *and* order together.

Faster boot time

You may recall from earlier in this series that SysVinit started every service in sequence, based on numbering. However, systemd automatically sorts through unit files to read dependency and ordering information. It uses this information to allow many services to start almost simultaneously, while preserving order where needed.

This handling is one reason startup can be faster under systemd. Faster boot time isn't primarily why systemd was invented, but it's often a side benefit of how it handles units.

Paul W. Frields

Paul W. Frields has been a Linux user and enthusiast since 1997, and joined the Fedora Project in 2003, shortly after launch. He was a founding member of the Fedora Project Board, and has worked on documentation, website publishing, advocacy, toolchain development, and maintaining software. He joined Red Hat as Fedora Project Leader from February 2008 to July 2010, and remains with Red Hat as an engineering manager. He currently lives with his wife and two children in Virginia.

November 25, 2015

For System Administrators, For Users

Previous post

Next post

5 Comments

ADD YOURS

Stan Sykes
November 25, 2015 at 05:27

Just a quick note to say a big “Thank You” for this series of articles. Very useful and informative.

Kind regards,
Stan

Steven Snow
November 25, 2015 at 11:12

Hi Paul,

Thank you for these articles. It is nice to understand the under the hood bits of Fedora to better customize it. Keep it up!

David
November 25, 2015 at 15:30

Unfortunately, in newer versions of systemd, network.target doesn’t work as an ordering milestone, and you have to use network-online.target instead if you want it to not start until the network connection is active. There are still some unit files that need to be fixed to adapt to this strange change.

Samuel Sieb
December 3, 2015 at 19:01

If you think there is some unit that requires the network to be fully online before starting and doesn’t have the right target, then please file a bug to get it fixed.

Leslie Satenstein

November 25, 2015 at 17:56

Paul, thank you for such a clear explanation about some of the “design jewels” created within the systemd architecture. When a design is just “Wow!, great!”, I call it a jewel.

Leave a Reply

Your email address will not be published.

Name

Email

Website

Post Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.

SUBSCRIBE TO FEDORA MAGAZINE

Search form 

Subscribe with RSS

or

Enter your email address below to receive notifications of new posts by email.

Email Address

Email Address

Subscribe