6.2.1 Yellowdog Updater, Modified (YUM)

Click one of the buttons to take you to that part of the video.

Yellowdog Updater, Modified (YUM) 0:00-1:38

In addition to RPM, you can also use the yum command at the shell prompt to install RPM packages. YUM stands for Yellowdog Updater, Modified. Where they came up with that name, I have no idea.

The yum command is really just a front end for RPM, but it has several key advantages over using just the rpm command alone.

First of all, it allows you to install a package along with all of its dependencies in one single command. Let me tell you, this is huge. The key problem here is the fact that the package dependency chain can get insanely long.

Here's what happens. You decide that you want to install package A on your system, and you're going to use the RPM command to do it. Great. No problem.

When you run RPM, you are informed that package A is actually dependent upon packages B and C being installed before it can be installed on the system. Before you can install package A, you've got to install packages B and C first. No problem.

You go and download them, and then you try to install them with the RPM command again. When you do that, you discover that package B is dependent upon packages D, E, and F being installed on the system first. You think "Great, I'll just move onto package C. We'll come back to B in a minute."

You use RPM to try to install package C, and you find out that it's dependent upon packages G, H, and I being installed on the system first. Of course, as we go to install these packages, we find out that they each have their own set of dependencies as well.

YUM 1:39-3:37

In my quest to install package A on the system, I end up having to research, download, and install 25 or 30 different packages just to get package A installed on the system. It can end up being a total nightmare.

This is where YUM really shines, because YUM will save you tons of time by automatically calculating and installing all of these dependent packages for you when you try to install package A.

The second advantage of YUM is the fact that it locates packages automatically for you. It does this by searching through package repositories on the internet. This is so helpful.

I can't tell you how many hours I've wasted before I learned about YUM trying to find a particular version of a particular package in order to meet a dependency requirement. Say I need version 2.0 of a package. I go on the internet and the only version I can find is 1.98, which doesn't work.

With YUM, there's no more poking around the web trying to find just the right package to resolve some dependency issue. It just does everything automatically. Frankly, once you start using YUM, you'll probably never use RPM again unless you absolutely have to.

When you run yum from the command prompt, it will go out and find the latest version of the package you requested. It will pull it down from the repository. It will check it for dependencies and install all of those dependencies automatically for you, and then install the requested package itself. All you have to is sit and watch.

Be aware that YUM is not supported, unfortunately, by all distributions. It was originally developed for Red Hat distributions, so it's supported on Fedora, but it's not supported on other distributions such as openSUSE.

Fortunately, most of these other distributions have their equivalent of YUM. For example, on openSUSE, you use the Zypper command instead of YUM. It uses pretty much the same syntax. It provides almost identical functionality to YUM on a Fedora system.

DNF 3:38-5:05

Before we go on, you need to be aware that YUM is actually being replaced. It's on its way out. In fact, on many newer Linux distributions you're going to see that it uses the dnf command instead of yum.

Don't get too uptight, because dnf is really just an enhanced version of yum. It has a different name, but the syntax is almost identical to that used by yum. In fact, DNF is an acronym that stands for Dandified YUM. It's just YUM on steroids. It works in exactly the same way, uses almost identical command syntax.

All you have to do is replace yum in your commands on these distributions with dnf. In fact, if you run yum on these distributions, you'll see that it actually just calls dnf to perform whatever task it was you want it to do.

The syntax for using yum is shown here. You run yum. You specify an option. You can specify a command. And then the name of the package that you want to perform that action on.

For example, if I needed to install the findutils package on my system, all I would have to do is run this command right here: yum install findutils. Then the YUM utility will search through the repositories that has been configured to search through.

It will locate the latest version of the findutils package. It will calculate dependencies. Then it will go ahead and download the RPM package and install it for you.

---

yumdownloader 5:06-8:27

Be aware that if you want to only download the package and not actually install it, you can use the yumdownloader command instead of yum. The syntax is pretty simple. You just enter yumdownloader and the name of the package.

The yumdownloader utility will then query those same repositories that YUM uses. It will pull down the latest version of the package, and it will save it in the current directory where you ran the yumdownloader command from. Then you can install it manually whenever you want to.

You can do a lot of other things with YUM besides just install packages. For example, you can use the yum remove command to uninstall a package.

You can use yum list installed to view all the packages that are currently installed on your system. You can use yum list install followed by a package name to check and see if a particular package is already installed on the system. You can run yum list updates to generate a list of all available updates for all installed packages on the system.

If you just want to check for updates for a particular package, you enter yum list update followed by the package name. If you want to view information about a package--such as its version number, dependencies, and so on--you enter yum info followed by the package name.

If you need to identify a package that provides a particular file, you enter yum whatprovides followed by the file name.

yum and yumdownloader don't magically know where to find these RPM packages that you want to install. We have to tell it where to look. This is done using two different configuration files. The first one is /etc/yum.conf.

This is the main configuration file for yum. It defines such things as the URLs for software repositories. It defines the directory where downloaded packages will be saved during the installation process--that's right here. It also tells us where the yum log file will be saved--that's right here.

I want to point out right here that we said just a second ago you can put the URL to your repository in this file. Notice that it tells us that we can also put in separate files named with an extension of .repo.

Understand that the second option is to store your repository information within separate files in the /etc/yum.repos.d/ directory. This is actually the preferred option. What you do is create a separate file for each repository that you want your yum utility to use. You name it appropriately and save it in this directory.

Here's an example of several that were configured by default from my Fedora system. We have fedora.repo. We have fedora-updates.repo and fedora-updates-testing.repo. The syntax used within one of these repo files is shown here.

We're looking at the main fedora.repo file that was configured by default on my system. First of all, we have the name of the repository in brackets, and then we have the name directive right here. Both of these just basically define a name for the repository.

Then we have the baseurl parameter right here, which defines a URL for where the packages are located on the internet. In this case, we are pointing to a server out on the internet, downloadfedoraproject.org, and then the path on that server to where those files can be found.

Custom Repository Configuration 8:28-9:28

You can do this if you want, but you can also create your own custom repository configuration files that points to a server on your local network. In this case, you could configure one server on your network to go out and pull files down from the internet, and then point all of your other systems to use that server.

Instead of going out on the internet, you just download the files locally. That enables you to reduce a lot of your redundant internet traffic. Instead of everybody going out and downloading the same files from the internet, we can just download them once to a local server and then distribute that file to all of our local systems.

We have the enabled parameter right here that either enables or disables this repository. We also have gpgcheck right here which either turns on or turns off GPG security key checking in order to validate the integrity of your repository files.

It's set to one. That turns it on. Zero turns it off. Then finally, we have the location of the GPG key for this particular repository.

Summary 9:28-9:43

That's it for this lesson. In this lesson we talked about using YUM instead of RPM to manage RPM packages on your system. We first talked about how YUM works. We then reviewed how to install and remove packages with YUM. Then we ended this lesson by talking about how to configure the YUM utility itself.
