

Package Management System

The name "package" is used in Linux for an application with all of its supported files. A **package management system** is a collection of tools to automate the process of installing, upgrading and removing software packages from a computer. Linux systems rely heavily on it.

Extension	Description	Related Commands
.bz	Burrows-Wheeler algorithmic Compression	bzip2, bunzip
.gz	Lempel-Ziv algorithmic Compression	gzip, gunzip

Tar Command

Usually it's needed to be able to combine many files into a single file that can easily be split back to its original files again. The **tar** command can create a new file based on the combined contents of several other files. The name tar originally stood for "tape archive" so one reason to use this command is to put files onto a tape. Before the machines were networked together, this was almost the only way to move a set of files from one UNIX/Linux machine to the other. The tar command puts the data into the same format as would be required for a tape, but the data is placed into a file. This can only be done if you use the `-f` option to specify which file you would like the data placed into.

The command to create a new file which is the tarring of a list of other files would be:

```
% tar -cvf new.file.tar first.file second.file third.file
```

The tar command is called with three options:

C: to tell it you want to create the tar file

V: to tell it you would like to see verbose output

F: to tell it you would like it to deal with file rather than tape

Here the three commands are shown all using the same dash (-), this can be done for most commands that are being called with more than one option. The file named immediately after the `-f` is the name of the tar file that will be created. The other files named will have their contents copied into the tar file being created (these files themselves will not be modified in anyway).

```
% tar -tvf new.file.tar
```

This use of the tar command will display a list of files contained in the tar file named immediately after the `-f` option.

```
% tar -xvf new.file.tar
```

This use of the tar command will extract the files contained in the tar file named immediately after the -f option. The tar file will not be modified in any way. The files created will be named as specified in the tar file; in files with those names already exist, their contents will be replaced.

```
% tar -rvf new.file.tar another.file
```

This use of tar command adds a new file to an already tarred file.

```
% tar -vf new.file.tar ---delete toBeRemoved.file
```

This use of tar command deletes an specific file form a tarred file.

Compressing files:

List of options for tar:

◇ c=Create

◇ x=extract

◇ t=table of contents

◇ v=verbose

◇ f=file

◇ r = to add a new file to an already tarred file

◇ --delete= to delete an specific file from a tarred file

◇ z to zip a file with gz extension

◇ j= to zip a file with bz2 extension

Debian Package Management:

To find information about download, install or remove packages you either use a graphical tool or the text modus. The graphical tools provided by distributions do not differ very much.

Package management with dpkg:

dpkg - - update - avail	: Updates the list of available packages
- l [- - list]	: Lists all installed packages
- i [- - install]	: Installs one or more package(s)
- r [- - remove]	: Removes one or more package(s) but the config files remain
- p [- - purge]	: Like - - remove but also the config files

are removed

Package management with APT:

APT is the *Advanced Package Tool* and provides the `apt-get` program. `apt-get` provides a simple way to retrieve and install packages from multiple sources using the command line. Unlike `dpkg`, `apt-get` does not understand `.deb` files, it works with the packages proper name and can only install `.deb` archives from a source specified in

`/etc/apt/sources.list`. `apt-get` will call `dpkg` directly after downloading the `.deb` archives from the configured sources.

Some common ways to use `apt-get` are:

- To update the list of package known by your system, you can run:
 `apt-get update`
(you should execute this regularly to update your package lists)
- To upgrade all the packages on your system, run:
 `apt-get upgrade`
- To install the *foo* package and all its dependencies, run:
 `apt-get install foo`
- To remove the *foo* package from your system, run:
 `apt-get remove foo`
- To remove the *foo* package and its configuration files from your system, run:
 `apt-get --purge remove foo`
- To upgrade all the packages on your system to a new Debian GNU/Linux release, run:
 `apt-get dist-upgrade`

Note that you must be logged in as root to perform any commands that modify the system packages.

The `apt` tool suite also includes the `apt-cache` tool to query the package lists. You can use it to find packages providing specific functionality through simple text or regular expression queries and through queries of dependencies in the package management system. Some common ways to use `apt-cache` are:

- To find packages whose description contains *word*:
 `apt-cache search word`

Sources to get packages from:

New packages can be fetched from different sources like CD- ROM, local directories or the internet. Sources to be used are defined in a list on your local machine.

The list is based in `/etc / apt / s o u r ces.list` and can be edited with a text editor.

Before doing so it's strongly recommended to archive the original file:

```
# cp /etc / ap t / s o u r ces.list /etc / apt / s o u r ces.list.orig
```

If you want to use sources from the Internet you should inform yourself about trustworthy URLs. Most distributions come along with a list of sources in the basic installation. To activate the preferred source in

`sources.list`, simply delete the `#` on the beginning of the line.

To add CD- ROM to the `sources.list` file, run the command:

```
#apt - cdrom add
```

Then comment all other lines in `/etc / apt / s o u r ces.list` file.

Redhat/SUSE Package Management:

The installed package

- | | |
|---------------------------|--|
| - V [- - verify] pkgName | : Checks if the package was downloaded correctly |
| - i [- - install] pkgName | : Installs the package |
| - - test pkgName | : The installation is only Simulated |
| - F [- - freshen] pkgName | : Updates installed packages |
| - e [- - erase] pkgName | : Deinstalls a package |

Package management with yum:

Yum is an automatic updater and package installer / remover for rpm systems. It makes it easier to maintain groups of machines without having to manually update each one using rpm.

1. Before you use yum for the first time, type following command to download headers.

Faculty Network Administrators' Training Course

yum check - update

2. To install a package which is not installed already, use "install" option with yum.

yum install [package - name]

3. To remove a package which installed already, use "remove" option with yum.

yum remove [package - name]

4. To install any updates available, use "update" option with yum.

yum update

5. To search a package by name (wildcard ok), use "search" option with yum.

yum search [any- name]