

Frequently Asked Questions

Abstract

This document answers questions frequently asked about Slackware Linux. If you have any specific topic in mind for the FAQ, create a heading for the FAQ entry, under an appropriate section, and provide your answer. If you cannot answer it, then somebody else might have the answer. Either way, the FAQ is a great way to build up tidbits of knowledge that might otherwise not fit into an entire article.

If the answer is too long to fit comfortably into two or three paragraphs, consider adding a link in the answer to an existing page that answers the question or create a new page if one doesn't exist on the topic and the question is important enough to justify a page.

General Slackware

What is Slackware Linux?

See the [Slackware Linux](#) page.

How can I get a copy of Slackware Linux?

Slackware Linux can be obtained *for free* by downloading the disc images directly from the [internet\(FTP\)](#), [Internet mirrors](#) and via [BitTorrents](#) .

Can I buy copies of Slackware Linux?

Absolutely, just go to the [Slackware Store](#).

Can I buy Slackware gear?

You bet, just go to the [Slackware Store](#).

Why spend money if I can download Slackware Linux for free?

Indeed, Slackware Linux will forever be freely available for download, no strings attached. However, consider:

Slackware Linux is developed by Patrick Volkerding as a full-time job. The sale of Slackware Linux CD's, DVD's and paraphernalia is his only source of income. The core team of developers working with Patrick do not get paid, but should Patrick have to stop developing Slackware because he can no

longer support himself enough to do so, the distribution will then have a very uncertain future.

Who are the people behind Slackware Linux?

Patrick Volkerding is the founder, project co-ordinator and chief maintainer of Slackware Linux. He is also known as BDFL (Benevolent Dictator for Life). Several active contributors also help Patrick maintain the distribution.

LinuxQuestions.org did an [interview with Patrick Volkerding](#) recently. That interview covers a lot of ground, including mentions of past and present contributors, their involvement, and a brief history of Slackware Linux.

Present contributors include [Eric Hameleers](#) (aka AlienBOB), Robby Workman (aka rworkman), Stuart Winter (aka MoZes), Eric Jan Tromp (aka alphageek), Alan Hicks, Mark Post, Fred Emmott, Vincent Batts, Heinz Wiesinger (aka pprkut) and several more.

What is this talk about stable and current versions?

Slackware basically comes in two flavors: stable and “current”.

Stable versions are the numbered releases (12, 12.1, 13, 13.37, 14.0, 14.1 etc.). They are supported for a number of years after release, by way of security updates. Security updates are added to the the “/patches/packages/” subdirectory of a Slackware release tree on every public mirror. Functional updates are not added to past releases.

The “current” branch on the other hand, is a *development tree* which is always split-off from a recently released stable Slackware directory tree. Usually this split-off happens some weeks after the stable release.

It marks the start of a new development cycle towards the *next* stable release. At the end of a development cycle, a Slackware release is created by renaming the top-level “*slackware-current*” directory to “*slackware-NEWVERSION*”. Slackware-current is known to cause relatively frequent and potentially disruptive updates to the system. New users and users looking for a stable system for production use should *always* use a (fairly recent) stable release. The Slackware developers assume that anyone running slackware-current realizes that he/she is in essence, a *beta tester*.

To sum it up:

- *Stable* is built on rock-solid components, with well tested software. This recommended version will fit your desktop or server needs.
- *Current* is the testing ground for the upcoming release. The software you will find there will often be [W upstream](#)'s latest version. For more information on current, please visit the [dedicated page](#).

If you want to know when the next version will be released (at time of writing the most recent stable release is 14.1), it will be ready... when it's ready! There are no fixed release dates, as the Slackware goal is to deliver the most stable Linux experience.

Can Slackware be recompiled from scratch?

Short answer: no.

Several other distros (including [Linux From Scratch](#)) are capable of being rebuilt from scratch using the publicly available sources and scripts. Fedora habitually recompiles itself completely from time to time. Slackware is *not* like those other distros: there is no “master build script”. The Slackware distribution grows and evolves “organically” and the utmost care is taken that all software in the distro is in working condition. Slackware does not need “rebuild from scratch” in order to achieve that goal. To the contrary: recompiling an older package may introduce new bugs or incompatibilities when the recompiled package picks up undesired new dependencies and/or functionality.

That does not make Slackware a “binary distribution”, not does this philosophy violate any license. In order to comply with Open Source licenses such as the GPL (and also because, why not?), Slackware makes the complete source code and build scripts available for all the packages that are part of the distro. You will find these sources in the `./source`, `./extra/source`, `./pasture/source` and `./patches/source` subdirectories of any release. These sources and scripts are exactly what was used to build the accompanying package at the time of its release - no more and no less.

Note: It is essential that you understand these sources and scripts have produced a package using the system libraries available at the time of its creation. Over time, with all the library updates and introduction of new packages, the source code of any package may lose the capability of being recompiled into that same package successfully. But that is perfectly normal: a Slackware package will *never* be recompiled unless its binaries stop working (due to dynamic library conflicts for instance) or because that package is upgraded to a new version. Only then, the sources and scripts for any package are re-evaluated. If patches are required at this time - to make the sources compile successfully or to make the resulting binaries work correctly - then patches will be added.

Installation and Support

How can I verify my copy of Slackware Linux?

All files in a Slackware Linux distribution can be verified against the [GPG key](#) of the distribution. For instance, if you want to verify if the ISO image you downloaded is an official, non-modified ISO image, then you run the following sequence of commands (example ISO for Slackware 13.37 for x86_64 platform):

```
$ wget http://slackware.com/gpg-key
...
2012-08-25 20:04:01 (81.3 MB/s) - 'gpg-key' saved [1357/1357]
$ gpg --import gpg-key
gpg: key 40102233: public key "Slackware Linux Project
<security@slackware.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1
$ gpg --verify slackware64-13.37-iso/slackware64-13.37-install-dvd.iso.asc
gpg: Signature made Mon 25 Apr 2011 07:14:02 PM CEST using DSA key ID
40102233
```

```
gpg: Good signature from "Slackware Linux Project <security@slackware.com>"
```

This message (gpg: Good signature) means that the ISO file is the same file which was cryptographically signed by Patrick Volkerding.

How do I install Slackware Linux?

Read the [installation guide](#).

During network installation I only see the "A" package series

You are attempting an installation of Slackware from a HTTP or FTP server. After entering the server's hostname and the Slackware package directory, you see the usual messages like "INITIALIZING PACKAGE TREE", but then during the "PACKAGE SERIES SELECTION", you can only select "A - BASE LINUX SYSTEM". Where are all the other package series?

You will experience this scenario when you are using a **32-bit** bootable installer image to install a **64-bit** Slackware from a webserver, or *vice versa*¹⁾: you attempt an installation of **32-bit** Slackware using a **64-bit** installation image.

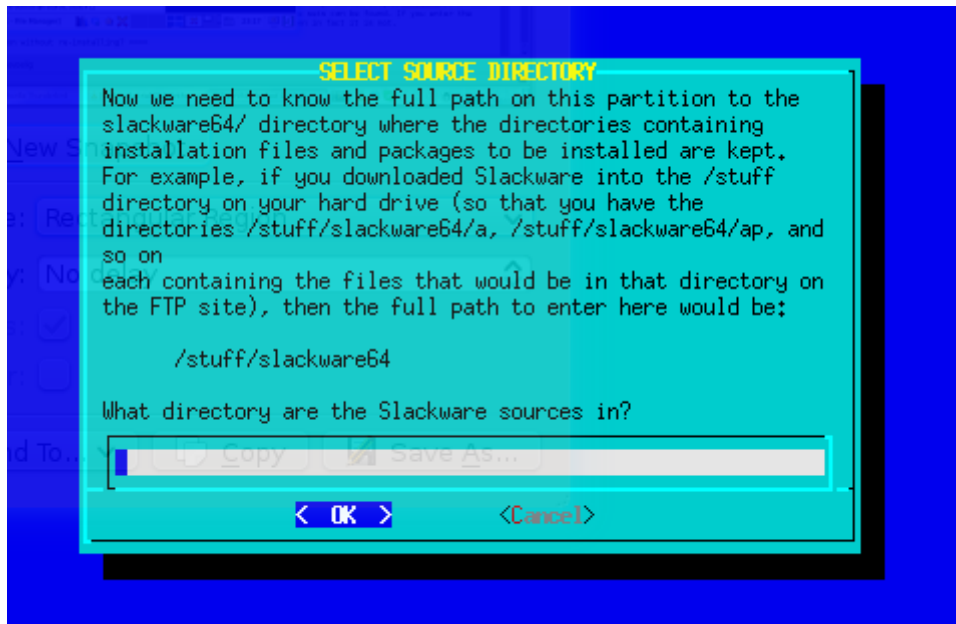
Solution: use the same architecture for the installer image and for the architecture of the Slackware release that you want to install from a webserver.

The package installation took 3 seconds and now what?

[Installing more than 1000 packages should take between 10 and 45 minutes, depending on the computer you are installing them on. If the installer claims it is finished after only a few seconds, that means it did in fact not install anything at all. So what happened here?](#)

This kind of issue occurs when you are not installing from the DVD or CD you booted from, but selected one of the alternative installation methods: a pre-mounted directory, or a loop-mounted ISO, *etcetera*. In all those cases, you have to enter a directory path to the location where the subdirectories representing the Slackware package sets can be found (the 'a', 'ap', 'd', ..., 'y' directories).

If you enter the wrong directory there, then the installer will not complain about your error. It will pretend that it installs packages when in fact it is not. Look at this message closely:



How do I upgrade an existing Slackware Linux installation without re-installing?

Read the page on [Slackpkg](#), particularly the [full system upgrade](#) section.

To track the "Current" branch, read the page on [Slackware-Current](#).

If you wish to perform a major upgrade from an earlier release to the latest stable version, you should read and follow the instructions in `UPGRADE.TXT` and `CHANGES_AND_HINTS.TXT` files provided in the official Slackware CD/DVD or internet mirror for the appropriate version (that is, the latest version).

Before upgrading the system, it is recommended that you always make a complete backup of your data.

Where do I get more information on Slackware Linux online?

A bunch of links to valuable online sources of Slackware information can be found on [this page](#). The officially recognized web forum for the Slackware community is at [LinuxQuestions.org forums](#).

Where can I find commercial support for Slackware Linux?

The official Slackware website [lists some companies](#) which offer Slackware Linux technical support and consulting. It is probably better to make local inquiries in your city or town for Linux related tech services if you need hands-on support. The local LUG (Linux User Group) might also prove to be a valuable source of advice/information.

Does Slackware Linux have a special "desktop" edition, "server" edition and...

No, there is only one edition of Slackware. Slackware is a general purpose distribution that can be configured and customized to the end user's needs. See also the [Slackware philosophy](#) for a lengthier explanation for adhering to the KISS philosophy.

When will Slackware Linux "x.y" version be released?

Slackware Linux does not have a fixed release schedule. The principle behind this is to *release when ready* and not before. There are no six-month release cycles like a few other popular Linux distributions. See the [philosophy](#) page for more information.

Does Slackware Linux work with my hardware or peripheral?

In short, if the Linux kernel supports a piece of hardware, so does Slackware. The distribution's "huge" kernel is compiled with maximum support for most devices and is a pristine, unpatched kernel. So if any Linux distribution is likely to have support for a given hardware device, Slackware should. Be aware that, even if a device is supported by a kernel driver, end user applications and userspace tools to utilize the full functionality of the hardware may be lacking. This is a rather complex topic and may require persistence and patience to research and figure out, especially in the case of low-cost or obscure devices.

Use an Internet search engine such as [Google](#) or [DuckDuckGo](#), or use a community contributed hardware compatibility list like the [LinuxQuestions HCL](#) to find out if your piece of hardware works on Linux or not.

While this can be a considerably daunting task for new Linux users, here are some references to get started:

- For printers, [CUPS](#) and [OpenPrinting.org](#) are good sources to look for device compatibility. HP printers and MFDs (Multi-Function Devices) have their own open source drivers at [HPLIP](#).
- For supported scanners, take a look at the [SANE](#) project.
- For webcams, and USB video class devices, [Linux UVC drivers & tools](#) can help.
- Linux supports several models of Wacom pen tablets and devices. See the [linuxwacom](#) project for more information. Many non-wacom tablets are supported by the [Wizardpen](#) driver.

Software and Package Management

How do I install/upgrade/remove software in Slackware Linux?

Slackware Linux comes with its own package management tools, namely `pkgtool`, `installpkg`, `upgradepkg` and `removepkg` for installing, upgrading and removing software packages. `makepkg` can be used to create packages. See also [slackpkg](#) and the page on [installing software](#).

Why doesn't Slackware Linux have my favourite "XYZ" software included in the CD/DVD?

The official distribution is small enough for various reasons, the most important one being constraints

on developers' time and resources. The short answer is that the official Slackware distribution cannot carry every possible open source application that might be needed by its users. However, Slackware still provides a large number of [applications](#) that are needed by most users.

Other sources for obtaining software not found in the Slackware CD/DVD (both in source and binary form):

Note that the licensing of some software may only permit redistribution in source form. Please do not redistribute or make available online any binary packages of software you compile from source unless you are sure the license permits it. Also avoid packaging and redistributing proprietary software without the creator's permission.

- There are quite a few trusted and well-respected individual Slackware contributors who make their SlackBuild scripts and binary packages available online. One source is [AlienBOB's repository](#) and another is [Robby Workman's packages](#).
- [SlackBuilds.org](#) is a community source of high quality and tested build scripts that make it easy to install third-party software on Slackware, and [sbopkg](#), makes processing SlackBuilds easier.
- Another option is to use binary packages from [Slacky.eu](#) or other binary Slackware package sources. Note that packages are contributed by the community and the quality of packaging may vary from person to person.
- Still another option is to convert rpm packages to Slackware packages by using the tool [rpm2tgz](#).
- [src2pkg](#) can be used to build packages from source, and to convert other package formats (.deb, .rpm) to Slackware packages. The [cpan2tgz](#) program is able to build Perl modules into packages.
- Last but not the least, intermediate to advanced users generally prefer to compile from upstream sources or write their own [SlackBuild scripts](#).

Why doesn't Slackware's package manager do dependency handling?

This is a huge debate in the Open Source community, bordering on *vi vs emacs* like flame wars, in reference to package management and dependency handling; but for several reasons, summarized shortly below, Slackware does not prefer “[automagic](#)” dependency resolution:

- Automatic dependency handling requires constant manual developer maintenance, and adds potential for dependency hell.
- The official Slackware Linux distribution is anyway meant to act as a cohesive whole. Hence, dependency management is largely moot as installing the entire distribution (the recommended way) takes care of most dependency problems.
- Several popular Open Source applications can be compiled with different dependencies based on compile-time configuration switches. This makes dependency handling harder and more error-prone for binary redistribution of third party software.
- Slackware Linux official distribution does not have the resources or manpower to manage dependency handling for third party software, which is a complex undertaking, requiring a lot of testing and is prone to errors as already noted above.

However, there are still solutions for third-party software automatic dependency handling for those who want it. [slapt-get](#) is a package manager that adds dependency handling for third party package sources like [LinuxPackages.net](#) and [Slacky.eu](#).

[Salix OS](#) is a Slackware derived distribution that incorporates dependency handling.

Distro X and Y does package management with dependency handling and...

Yes, many distributions have package managers that do dependency handling and do them well. Indeed, there are many choices in that regard. The Linux community still needs a stable, tried and tested distribution that **doesn't**. Slackware certainly fills a niche in the Linux community in that regard. But as always, preferences vary and many Linux users do prefer the bare simplicity of Slackware. One thing to bear in mind is that the core simplicity of the package management in Slackware has one significant side benefit: there will always be third-party enhancements and solutions to commonly faced issues.

Some may call it a bug, others may call it a feature. Be it so, Slackware remains committed to its core principles and this issue is one aspect of its core principles.

Miscellaneous

Why does Slackware take so long to boot up?

By default, Slackware installs the huge kernel, which has built-in support for all possible hardware drivers. This may increase you boot time as the kernel probes for hardware at startup. You can switch to the generic kernel by following the instructions in the file `/boot/README.initrd`. See also: [how to set up the generic kernel](#).

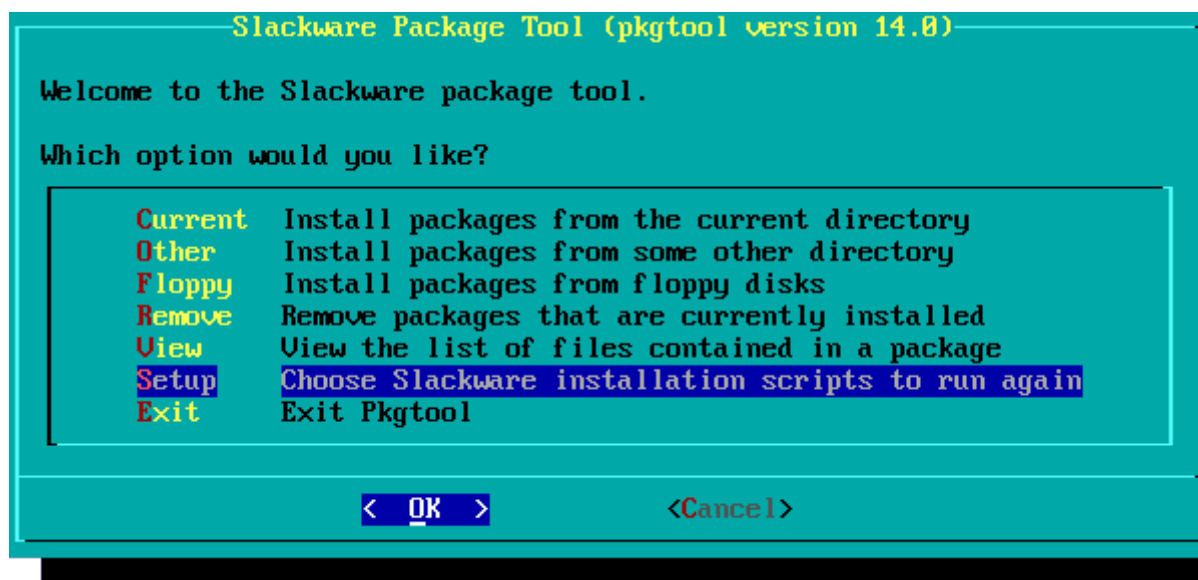
You may also want to disable unused services in `/etc/rc.d/` directory by disabling the executable bit of the startup script. For instance, to disable the Apache daemon, simply run (as root)

```
chmod -x rc.httpd
```

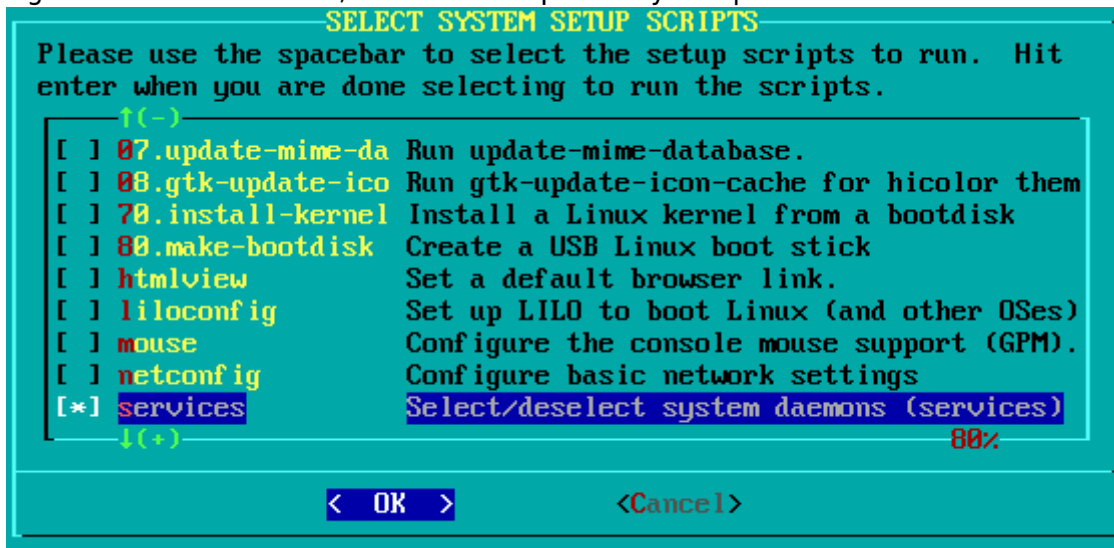
If you prefer screen menus to do same task you can run (as root) `pkgtool`

```
# pkgtool
```

navigate to Setup item and press Enter



then navigate to services item, mark it with Space key and press Enter



then clear mark in front of [] httpd item with Space key and press Enter.



Finally, select Exit end press Enter.

Additionally, if you use LILO, Slackware's default boot loader, you can add the compact option to /etc/lilo.conf to speed up the boot process by merging read requests from adjacent sectors:

```

# LILO configuration file
# generated by 'liloconfig'
#
# Start LILO global section
# Append any additional kernel parameters:
append=" vt.default_utf8=1"
boot = /dev/sda
compact

```

Also, Slackware runs a bunch of X/GTK related scripts at startup for multiuser runlevels (fc-cache, update-mime-database, gtk-update-icon-cache, update-gtk-immodules, update-gdk-pixbuf-loaders, update-pango-querymodules etc.). These may be disabled manually by

commenting out those lines which refer to them in the file `/etc/rc.M` but don't disable anything unless you're absolutely sure you know what you are doing! These scripts are essential for GTK applications to render fonts and icons properly.

Unless you are a fairly advanced user and know what you are doing, do not attempt to modify or disable any of the boot scripts. Does a few extra seconds gained while booting really make a big difference?

Slackware Linux has a reputation for being tough for newbies. Is this true?

Read [The Slackware way](#) to find out what you can expect from a distribution like Slackware. While Slackware certainly is different from many other distributions, whether easy or difficult to use is a matter to be decided entirely by the individual user.

Some things to bear in mind, though:

- Slackware has a text-menu based (ncurses) installer and requires several technical decisions to be made at install-time, some of which can be potentially destructive to the data on the hard disk. In other words, when in doubt, don't perform any potentially destructive actions and *do* consult the help documentation.
- A new Slackware installation initially boots into a text console by default and can be [configured to boot into a GUI](#) later, as opposed to desktop-based distributions that boot into a GUI login screen by default.
- Basic knowledge of standard shell (bash) commands and a text-mode editor such as vim is necessary to start using Slackware comfortably.
- Some knowledge of the locations of commonly used configuration files and scripts found in *nix-like systems would make things much more comfortable.

[slackware](#)

¹⁾

the other way round

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
<https://docs.slackware.com/slackware:faq>

Last update: **2017/08/13 14:30 (UTC)**

