

Linux Tips and Tricks

- Introduction -
- Manual Wifi configuration -
- Application-related -
- Adding repositories -
- Installing and removing packages -
- Packages from other distros -
- Adding commonly used applications to the panel -
- Making it look good -
- Themes -
- Other settings you can change -
- Set up everyday software -
- Web browser -
- Text Editor / IDE -
- File manager -
- Life savers / enhancers -
- Terminal aliases -
- Hardinfo -
- Data recovery -
- Deploying your system elsewhere -
- Using ddcutil to control your monitor -

Introduction

The more accurate title would be **the things I do after installing a Linux system to make my experience more comfortable**. Even with such a user-friendly distro as Salix (which this guide will assume - but most of it can be adapted to other distros), it is easy to get overwhelmed, especially for a recent Windows refugee. So, hopefully this article will provide some direction and save you time, effort, and possible frustration for when you break something and / or have to install Linux on another machine.

Manual Wifi configuration

Putting this in first since Internet access is required for much of the guide. Since NetworkManager is a famously terrible piece of junk and the Slackware scripts don't always pick the network up, here's how you can do it on your own (WIP):

- Type *“iwconfig”* or *“ifconfig”* to see the name of your interface (say, *“wlan0”*).
- Type *“sudo iwlist wlan0 scan”* to see the Wifi networks found. Take note of the one you want to connect to.
- Type *“sudo wpa_passphrase my_beautiful_network amazingly_strong_password”*, filling in the necessary values. Copy the output.
- Type *“sudo leafpad /etc/wpa_supplicant.conf”*. Paste the output and save.
- Type *“sudo leafpad /etc/rc.d/rc.inet1.conf”*. Add this at the end (of course, if you

already have another network with the number 4, use a higher number):

```
IFNAME[4]= "wlan0"  
IPADDR[4]= ""  
NETMASK[4]= ""  
USE_DHCP[4]= "yes"  
DHCP_HOSTNAME[4]= "my_beautiful_network"  
WLAN_WPA[4]= "wpa_supPLICant"  
WLAN_WPADriver[4]= "wext"
```

The Slackware scripts should pick it up on reboot now. To test it out, type “*sudo sh /etc/rc.d/rc.inet1 wlan0_restart*”. You can debug the terminal output if it still refuses to work, or inspect “*/var/log/messages*” for even more information. Failing that, we can bypass the Slackware scripts and connect with wpa_supPLICant directly by typing “*sudo wpa_supPLICant -iwlan0 -c/etc/wpa_supPLICant.conf&*”. This should be foolproof, but in case it's not - the wpa_supPLICant log resides in “*/var/log/wpa_supPLICant.log*”. If you want to connect with wpa_supPLICant at every boot (bypassing Slackware scripts), add these commands to “*/etc/rc.d/rc.local*”:

```
sh /etc/rc.d/rc.inet1 wlan0_stop  
wpa_supPLICant -iwlan0 -c/etc/wpa_supPLICant.conf&  
dhcpcd&
```

In case the network is still failing after reboot, the issue might be resolv.conf not being updated by dhcpcd. If that happens, type “*sudo resolvconf -l*” and copy the relevant nameservers. The name of the interface might not be the same as before (for example, I've had it become

“*wlan.ra*”), so take that into account. Paste the output into “*/etc/resolv.conf*”, which should look something like this (of course, the IPs will be different):

```
# resolv.conf from wlan0.ra  
# Generated by dhcpcd from wlan0.ra  
nameserver 55.55.55.55  
nameserver 33.33.33.33
```

Application-related

Adding repositories

Now, Salix repositories are quite large and should be enough to get you started, but someday you will need to branch out. The defaults are limited to packages from Slackware 14.2, while sometime you just might need a newer version (for example, certain python3 applications require 3.6 which Salix does not provide). Some packages are also not available at all, such as SpaceFM, my favorite file manager. In come the Slackel (http://www.slackel.gr/repo/x86_64/current/) and SlackOnly (https://packages.slackonly.com/pub/packages/14.2-x86_64/) repositories. Just start up Gslapt, go to Preferences - Sources, click Add, and copy those links. Remember, though, that **you can easily break shit** if you aren't careful - a repo can, for example, pull in a library that a package requires, and that one will replace the one you had before - breaking everything that was dependent on it. There are also other available repositories, such as Ponce or Alien - search around!

Installing and removing packages

It is hard for a distro to contain everything you need and nothing else - in fact, it is impossible (but Salix comes close). So you will always have to add and remove something. You can use the repositories you've added in the previous step - or, another nice way to install a particular package you need is to use SlakFinder (<http://slakfinder.org/> - here are my [search plugins](#) for its easier usage). This website allows you to type a package name (even partial) and it will show you all the repositories you can download it from (including the links!). You will have to use the **installpkg** command to use those. This way will also not bring in any dependencies, which has its advantages (no breakage) and disadvantages (package might not work). Note: if you want to skip installing dependencies, you can use the **Download packages only** option in Gslapt - or find a repository that doesn't contain dependency information. I will assume you've managed to install everything you need without breaking anything (harder than it sounds sometimes). Slapt-get (what Gslapt uses) keeps all its installed packages in /var/slapt-get/, so I recommend copying those to a pendrive. This way, if you ever install Salix elsewhere, you can immediately bring it to your preferred state by running the command "sudo installpkg ./*" while inside the folder you've stored them in. And you won't even have to connect to the Internet anymore to download those packages.

Packages from other distros

Unfortunately for us Slackware-based distribution users, Ubuntu / Debian are the primary target for many Linux developers. Therefore, many packages **cannot be found** in the Slackware .t*z format, but can in .deb or sometimes Red Hat's .rpm. What to do then? **deb2tgz** and **rpm2tgz** to the rescue. These scripts allow you to convert any .deb or .rpm package so that it is compatible

with Slackware-based distros. deb2tgz is found in the SlackOnly repository, while rpm2tgz - in the Salix one.

Add commonly-used applications to the panel

By default, Salix includes only a few generic icons on the panel - the terminal, text editor, file manager and web browser. But of course you will use so much more of them - so why not put the most used ones in there as well? Let's add a music player - right click the panel, hover over Panel, and click Add new items. Now choose Launcher and click Add. An empty icon will appear in there - right click it and choose "Properties". Now click the plus button. Type the name of your music player (for example QuodLibet) or find it in the list of applications, then click on it and use the Add option to put it in the panel. Close the launcher editor, right click the icon that appeared, and Move it to wherever it is convenient for you. You will probably install packages pretty often, so let's add Gslapt in there as well. Then the mail client, painting program, IDE, and movie player. You're all set! The result should look something like this:



This should heavily boost your productivity - no more clicking around to find the program you're looking for. The panel settings (as well as other Xfce-related ones) are stored in `/home/YourUsername/.config/xfce4/`, so you can easily copy them to a pendrive or something. And if you break Linux on this machine, you can just reinstall it and keep the home partition, so the settings will be carried over.

Making it look good

Themes

Let's face it, Xfce's default look **sucks**. And all its default themes also **suck**. Now you can install the murrine engine and its associated themes, but they are only slightly better. Chances are, if you want to actually enjoy looking at your computer, you will need to peruse the website <https://xfce-look.org/>. But first, go to Settings > Settings manager and let's see how the look of Xfce is actually decided. You can divide it into five parts:

- Theme (the color scheme of everything except the title bar - set through the Appearance menu)
- Icons (also Appearance menu)
- Wallpaper (in Desktop)
- Title bar (colour and buttons; in Window Manager menu)
- Cursors (in Mouse and Touchpad > Theme).

Now wouldn't it be nice to have all these put together, instead of scattered around? Download the package Xfce-theme-manager (find it through the aforementioned SlakFinder). Now a new section will appear in your Settings Manager. There you can modify all the settings as well as some additional ones - and save them as actual whole themes. Anyway, visit <https://xfce-look.org/> and look around. First, time to dump the boring single-colored cursors. Let's have some fun - how about a **fucking Katana** as your cursor - download it by - you guessed it - clicking the Download btton. To install the cursor set, move the archive to /usr/share/icons, open it in Engrampa and extract the folder that is INSIDE - just extracting the archive will create a folder with its name, **and the theme will not work**. Some of my favorites are Charged, Dragon's Jewel, and Dragon Claw. Keep in mind anyone could submit a theme, so there's

bound to be a few low quality or even incomplete ones. Now do the same for Icon Themes and XFCE/XFWM4 Themes (extracted in /usr/share/themes) and you're all set! Note: while seeking themes, try to find ones that have GTK3 support instead of just GTK2; otherwise, the GTK3 programs will use the default look and seem out of place compared to the GTK2 ones (edit: or trash gtk3 altogether, then you can use all the gtk2-only themes). As for the wallpaper - well, you can get it from anywhere - any png/jpg image will work. Put them in /usr/share/xfce4/backdrops/ to have them displayed in the settings manager. It is also possible to edit a theme (or create your own, but that's way too much effort) - for that, go to your theme's folder, then gtk2.0 and open the “*gtkrc*” file in your favorite text editor. There's really no proper documentation for doing this, so you'll have to figure out what stuff means on your own.

Other settings you can change

Lots! Let's go through them one by one:

- **Screensaver**

- you're a lucky guy; there are a lot of them **by default!** Even a fucking Pacman one. Find them in Settings > Screensaver. Some fun ones are Kumppa, Fireworkx, or BinaryRing.

- **Transparency**

- a common criticism of "ricing" is that you never see your wallpaper. Well, not if you have this enabled! Go to Window Manager Tweaks > Compositor, enable it if it isn't already, and choose your preferred transparency levels for inactive and moving windows. I recommend making them mostly transparent. While we're here, turn on your Terminal emulator (if you're new to

Linux - trust me, you will use this a lot) and change its transparency as well (Edit > Preferences > Appearance. Background > Transparent).

- **Screen resolution**

- self explanatory. It's in the Display section of the Settings Manager.

- **The look during Alt+Tabbing**

- it's in Window Manager Tweaks > Compositor, specifically the **Show windows preview in place of icons when cycling..** I think this should be disabled, because it's fucking ugly.

- **Font family and size**

- Appearance > Fonts.

- **Desktop icons**

- Desktop > Icons. I recommend deleting them all, to see the wallpaper in all its glory.

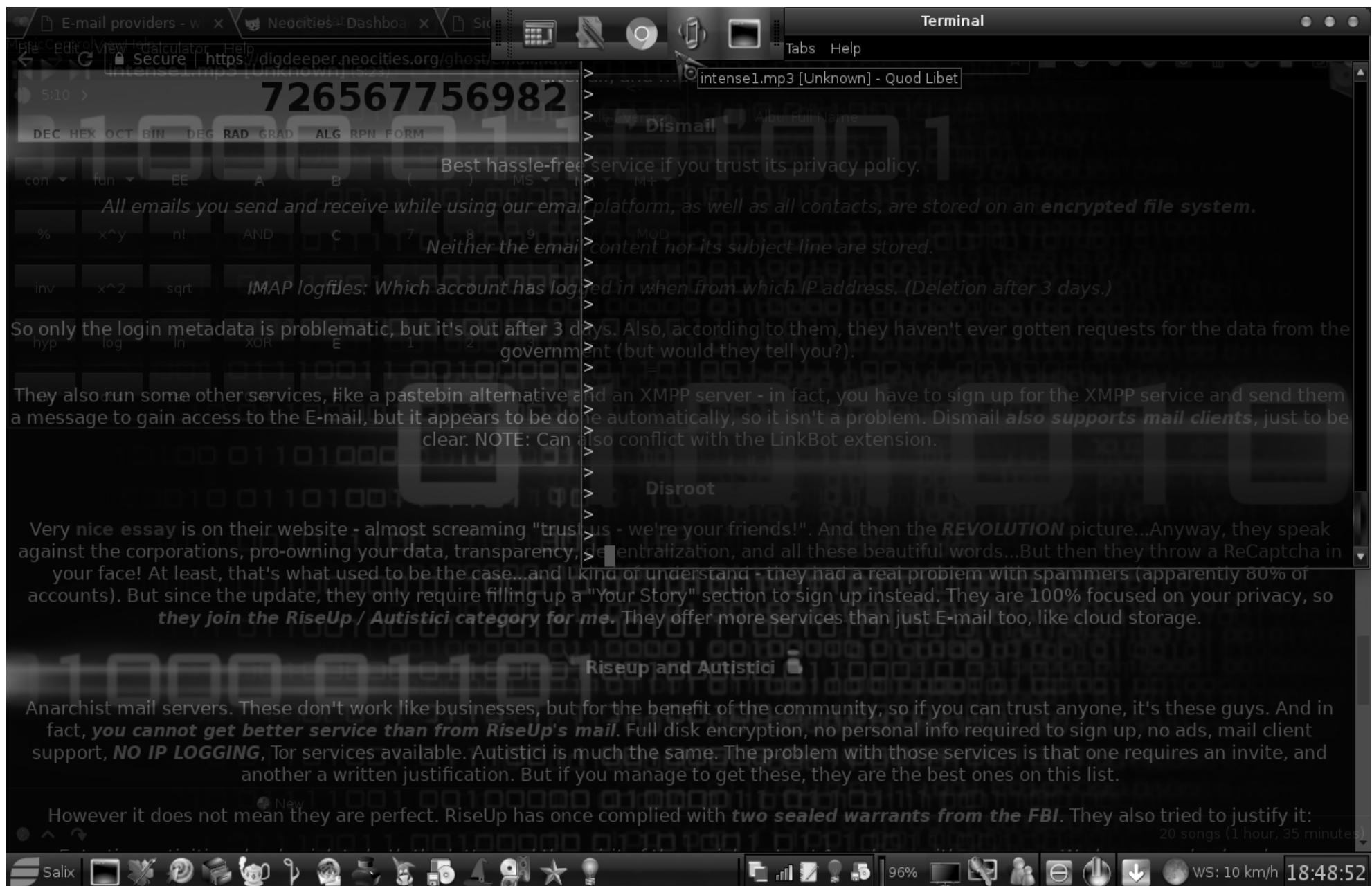
- **Terminal colors**

- Lime on black, hacker style ^_^

- **Changing the look of specific icons** - You can do that too. Right click on an application on the panel, Edit Launcher and click on its icon. By using the "Image files" option and combing through the folders in /usr/share/icons/, you can use icons from

many different themes at the same time, or even your own custom ones.

If you want to create a new panel, right click an existing one and navigate to Panel > Panel Preferences, then click the + button. There you can choose its properties such as size or whether it will hide itself when the cursor is away. You can use it, for example, to display the currently active applications at the top while you hover over there. Okay, that's about as much as you can do with pure Xfce. The final result might look something like this:



There are some other tools for more heavy ricing that I might describe someday; for now though, let's move on to functionality.

Set up everyday software

Web Browser

If you're like me, you will be spending much of your computer time browsing the Internet. So why not prepare it for this - as we will see - gargantuan task now? First of all - the browser choice. Refer to my other article - [How to choose a browser for your daily use?](#) - for advice on this topic. Time to harden it with addons (if it supports them - but if you read the linked article, you will hopefully see how essential they are). [Essential privacy \(and other\) addons](#). Now let's set up our search engines. Pale Moon will detect most of them when visited and you can add them then. To test it out, go to <https://metager.de>, press the small black arrow near the search bar, and then "Add Metager". Now you can select it from the dropdown list and it will become the default in the search bar. Another way to add them is finding a plugin from <https://mycroftproject.com/> for the website you want to search through. After you've added all the ones you want, use the Manage Search Engines menu (black arrow again) to set keyboard shortcuts for them. This will allow you to use any search engine in the address bar by typing their shortcut (e.g ddg for DuckDuckGo) followed by the query - very convenient. If you've settled on a Chrome-based browser, it will **detect whenever you use a search engine**, and automatically add it to the settings menu. Go there and click "Manage search engines". There, you will have a list of all the ones you've ever used - so if you searched for something with SlakFinder, you can now set a keyword for it. Click on the three dots to the right of it, Edit, and put "sf" as the keyword. Now you can perform a SlakFinder search by typing **sf your-search-term** in the address bar. Pretty convenient, huh? Now do the same for StartPage (keyword sp) and DuckDuckGo (keyword ddg), as well as any others you want. Note: you should also configure a VPN and / or Tor for your web browsing. Refer to [How and why to VPNize your](#)

entire traffic and Darknet setup in Slackware-based distros.

Text Editor / IDE

If you are a developer, Geany is a great, lean tool to use and installed by default in Salix. What I like to do here is first to **create some projects** using the *“Project > New”* option. With this, you will be able to switch through different projects with three clicks and have all their files loaded. Very useful if you - for example - both develop a website and create software. The nice thing about this is that you can **search and replace a string in all loaded files** (Ctrl+H -> Replace all in session). Makes it much more convenient to do mass modifications. Another feature I like to use is **templates** - if you are going to be creating documents that look the same just with different content, templates are invaluable. Create a skeleton of your file with all the content removed and put it in `/usr/share/geany/templates/files/`. Then, anytime you want to make a new document you can do that in 3 clicks with File -> New with template and then choose the template. Very convenient! Macros are another thing you should really try to learn (Tools -> Edit macros) - these allow you to make more complex modifications to a document, but they can be kind of hard to use. Though with a little bit of effort they can automate lots of tedious work. Another thing to take note of are the navigation keyboard shortcuts (move to next word, delete current line, etc) which again streamline the development. **UPDATE April 2025:** newest Geany seems to have removed the underscore as a "word boundary" (basically any character that "stops" jumping between words with CTRL+left / right arrow). This was unacceptable to me as my images are all named like "my_awesome_image.png" and sometimes it's really necessary for me to quickly change the "awesome" to "horrible" or something like that. So I set out on the journey to figure this out and I did. Go to *Tools - Configuration files -> filetypes.common* and find this section:

```
#~ [settings]
#~ # which characters should be skipped when moving (or included when deleting) to
word boundaries
#~ should always include space and tab (\s\t)
#~ whitespace_chars=\s\t!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~
#~
wordchars=_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

First, uncomment the first line, which enables the entire config section. Then, uncomment the whitespace and wordchars lines. As you can see, the last line contains the underscore, meaning it's going to be considered a "letter" by geany, and won't stop the "fast travel". Delete it. Then add an underscore to the previous line, which defines the characters that actually **do** stop the "fast travel". I like adding it to the end of the line. The final section will look like this:

```
[settings]
#~ # which characters should be skipped when moving (or included when deleting) to
word boundaries
#~ # should always include space and tab (\s\t)
whitespace_chars=\s\t!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~_
wordchars=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

After that, go to *Tools* and click *Reload configuration*. Everything should work properly after that.

File manager

Another one you'll be spending some time on. This guide will assume you are using **SpaceFM**, since it is simply the most powerful one. What can it do?

- **Create custom commands** - see [here](#).
- **Set up key shortcuts** - for like everything. Right click any item and then Key Shortcut.
- **Create bookmarks** - Navigate to a folder you want to add, then click Bookmarks -> New Bookmark. Right click the bookmark that has appeared to set its icon or key shortcut.
- **Set global mount options for devices** - right click on a device, then Settings -> Mount Options. If you want to do that for a specific device, use the Re/Mount option. Some useful ones are rw and sync especially for MP3 players (without sync, the files don't seem to be saved to the player at all, even if you sync manually).
- **Modify the toolbar** - right click it to see all the options. Can add bookmarks, commands, applications, move stuff around, remove buttons already there, etc.
- **Lots and lots of others!** SpaceFM's power is truly **galactical** - I don't even know half of it. Explore!

Life savers / enhancers

Terminal aliases

Let's face it - it can get tedious to type the same long terminal commands all the time. Aliases to the rescue! In your home directory, create a file named **.bash_aliases** - yes, with the dot - and put this line there: **alias sli= "sudo slapt-get install"**. Since you will most likely be using slapt-

get quite a bit, this will bind it to the shortcut **sli**. And you can do that for any command just by putting an alias in a new line. Keep in mind you have to log out to have the shortcuts be applied.

Hardinfo

Just a program that shows you all the hardware you have - something that should really be there by default, but somehow isn't. Available in the default repos.

Data recovery

It is inevitable that you will lose some of your data sooner or later - whether through power, hardware, or system failure; accidentally (or on purpose - and then regretting it) deleting it, or through an overzealous BSD installer eating your old partitions. Easy to say **back it up!** - and you should - but a backup can get eaten by a dog, too. There are two tools available for data recovery - extundelete and PhotoRec. Now, the former only goes after the files **still present in your filesystem** - even then, I've always had problems actually getting it to do anything; the latter, on the other hand, goes straight to the underlying data. It will recover anything that can be - so let's learn how to use it.

Let's say you've had some hacking ^_^ ebooks stored on a pendrive - but you've decided to use it for something else - and now you'd like to bring them back. Type **sudo photorec** in the terminal and choose the relevant drive. Now press the right arrow twice to enter File Options. Press S to disable all the file types, then scroll down to **pdf**, press Space and Enter to go back to the main menu. Forget about specific partitions - it might have been repartitioned a thousand times since the data loss - so use the **No partition - Whole disk** option. Again, filesystem doesn't

matter if we're looking through the whole disk, so just select ext3, move the cursor back to Search, and press Enter. Last thing you have to do is choose where the recovered files will be dumped. You can create a new folder "Recovery" inside /home if you want to. Before we start, I should warn you that the dumped files can take quite a lot of space - even **all of it** - so watch out. Okay, let's press C, and the process can begin.

Right away PhotoRec will begin dumping a bunch of files (if it found anything) - and you can explore them real-time. The recovery can take quite long, so be patient. Now remember - PhotoRec goes **after the underlying data** - so you can find some partial or corrupted files - and the filenames are ignored since they are stored in the filesystem. You might be surprised at what PhotoRec finds - I've seen system-related stuff from like five installations ago, or old pictures I've used for shitposting on chans that I forgot have even existed. To make it easier to find the file you want, sort them by size so the useless <1KB ones will be shown first. If it was a text file, you can also type the command *"grep -rnw '/home/YourUsername/Recovery' -e 'TextToBeFound'"*, which will search for files containing a particular string inside the Recovery folder. Okay, that's about it for our brief PhotoRec guide (it can do more - search the 'net; the basics are sufficient though). Hook up some old drives and happy hunting!

Deploying your system elsewhere

Moved to a [separate article](#)

Using ddcutil to control your monitor

Since it's annoying to get up and press those unwieldy buttons anytime you want to change

color settings or such. First of all, enable the option *DDC/CI* in your monitor settings if it has that; **not all monitors support this standard** and laptops apparently don't. If yours doesn't, unfortunately none of this will work. Now install packages *ddcutil* and *i2c-tools* if they are not there already; then type “*sudo modprobe i2c-dev*” into the terminal. The command “*sudo ddcutil detect*” should then show you some information about your monitor. Type “*sudo ddcutil getvcp KNOWN*” and you'll get a long output which contains lines such as this:

VCP code 0x30 (Vertical Position (Phase)): current value = 50, max value = 100

VCP code 0x20 (Horizontal Position (Phase)): current value = 50, max value = 100

VCP code 0x12 (Contrast): current value = 75, max value = 100

Note the parts starting with “*0x*” - those are the codes for the values you want to edit. So - if you want to change the “*Contrast*” - you'll need the “*0x12*” code number. Now type:

sudo ddcutil setvcp 0x12 70

This changes the contrast to 70. Of course, the code can be changed to any feature that is supported by your monitor (many won't be and will still require the buttons). Changing the video input source to HDMI:

sudo ddcutil setvcp 0x60 3

This can be used to e.g enable playing with a Nintendo Switch in docked mode through one command, instead of pressing a bunch of annoying buttons (hell, I know most of the currently produced monitors have the horrible flat ones). Of course, the codes for your device will be different. By the way, if you intend to use this more often, you might want to add the *i2c-dev*

module to */etc/rc.d/rc.modules.local* so that it is loaded on boot.

Other things to keep in mind

- **ALWAYS** keep around an USB drive with a Linux Live iso on it! This will be essential if you render your current installation unbootable; even if you don't manage to recover it, you will be able copy the files you need, and reinstall. Even better - use the instructions from [this article](#) and keep THAT USB drive around.
- To make a command not steal the whole terminal window, add the “&” character after it (for example, “*gparted&*” or “*gslapt&*”). If you've already run a command without it, press Ctrl+Z in that terminal window and take note of the number in parentheses that appeared. Then, type “*bg that number*”, for example “*bg 3*”, which will send the command to the background. You can bring it back to terminal with “*fg 3*”.

[Back to the front page](#)