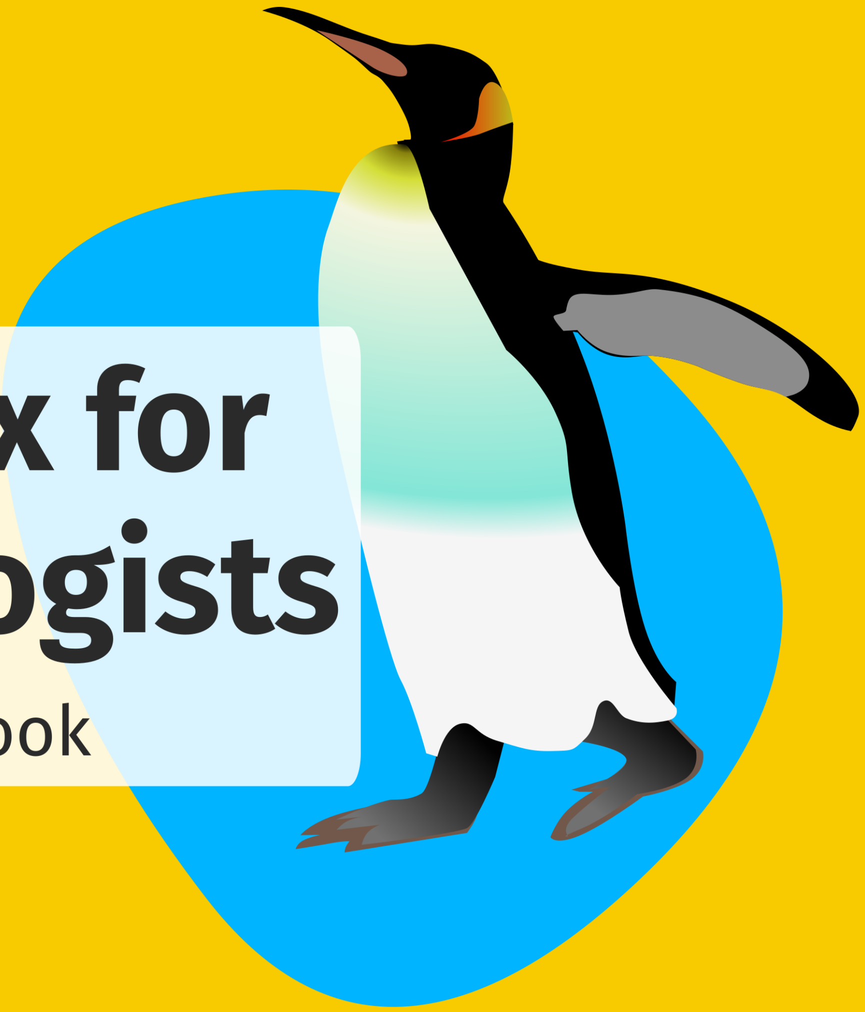
Linux for Biologists

A Cookbook



Vimalkumar Velayudhan

Linux for Biologists

A Cookbook

Vimalkumar Velayudhan

First edition

June 9, 2021

For Shanthi

Thanks

I would like to express my gratitude to my mentors, colleagues, students, friends and family. Without their support and encouragement, this book wouldn’t have been possible.

Thanks also to the wonderful world of Linux and open source software and the community around it.

Contents

1 About this book

The target audience for this book, what you will need to follow the included recipes and a note on the author.

1.1 Who is it for?

If you are a student or a researcher in biological sciences and are new to Linux, this book will help you get started quickly.

The primary focus is in using graphical user interfaces (GUIs) to accomplish tasks, but I have also included methods using the command line interface.

1.2 What you will learn

Using the included recipes, you will learn how to:

- Run Linux on your computer
- Use the desktop and included software
- Work with files and directories
- Search and install software
- Run some basic commands
- Run the Galaxy platform on your computer

Note: *If you need help while following this book...*

You can visit the forums of this book on Leanpub. If your query is not answered there, please start a new discussion.

https://community.leanpub.com/c/linuxforbiologi

1.3 What you will need

To follow the recipes in this book, you will need:

1. A Linux desktop
2. Administrator privileges

1.3.1 Linux desktop

The Linux and open-source world is full of choice.

There are different versions (or distributions) of Linux that are available for download and use.

To ensure that you can follow this book successfully, *Linux Mint* 20.1 is used as a *reference* distribution. All the included recipes have been tested on this distribution.

Virtual machine image

If you do not have access to a Linux desktop, you can download a virtual machine from the website of this book and use it to run a virtual machine on your computer (Fig. 1).

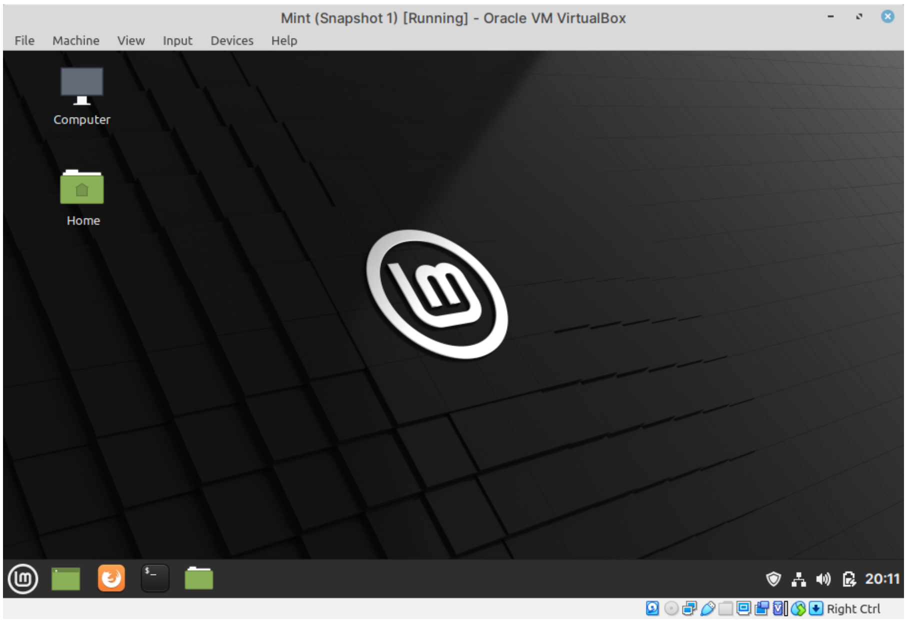


Fig. 1: Linux Mint running in VirtualBox

Follow the steps in *Running a Linux virtual machine* to get started.

What about other Linux distributions?

Although most of the recipes will work on other Linux distributions, you will need to make changes in some cases.

For example, the applications you use might be different — instead of the *Files* file manager used here, you might use Nautilus or Dolphin.

If there are problems, I recommend you download and use the *virtual machine*.

1.3.2 Administrator privileges

The software installation steps in this book will need administrator privileges. In the default configuration, only *root* or members of the `sudo` user group will be able to install software on a Linux system.

Note: If you are using the *virtual machine*, distributed with this book, you do not need to do anything.

If you are not using the virtual machine, you will need to ensure your user account is part of the `sudo` group, to be able to install software.

1.4 About the author

Hello. I'm Vimal.

I got introduced to Linux and open source software while pursuing my Master's degree in Biotechnology at Madurai Kamaraj University, India.

The idea of being able to use and modify software without restrictions was new and exciting to me. I learnt to install Linux, compile software from source code and even create my own Linux live CD!

This experience became valuable in academia, where I worked primarily in research support for a period of over 9 years. During this time, I had installed and managed Linux servers, databases, web applications and websites, contributed in the development of scientific software, published software in repositories, and taught beginner level courses in Python programming and Bioinformatics.

This book is a compilation of the experiences I had gained over time in my interactions with students, colleagues, and the users of our services. Thank you for your interest in reading it. I hope you will find it useful.

Vimalkumar Velayudhan

Website: https://vimalkvn.com

Email: vimal@disroot.org

2 Getting started with Linux

In this section, you will learn how to:

1. Run Linux on your computer
2. Use the desktop and included software
3. Work with files and directories
4. Start using the command-line and learn some basic commands

2.1 Linux — an overview

Linux is an *operating system* developed by Linus Torvalds in 1991. It is released under an open-source licence and the source code is freely available for everyone.

2.1.1 Linux distribution

A Linux distribution includes the Linux operating system with additional software.

Examples: Debian[3], Ubuntu[4], Linux Mint[5], Fedora[6], openSUSE[7] etc.,

[3] https://www.debian.org/
[4] https://ubuntu.com/
[5] https://linuxmint.com/
[6] https://getfedora.org/
[7] https://www.opensuse.org/

2.1.2 Desktop environment

A desktop environment is a *graphical user interface* that provides easy access to files and applications.

Examples:

- GNOME (Fig. 2)
- KDE Plasma (Fig. 3)
- Xfce (Fig. 4)
- MATE (Fig. 5)
- Cinnamon (Fig. 6)

Linux distributions might offer additional variants for download, depending on the desktop environment in use.

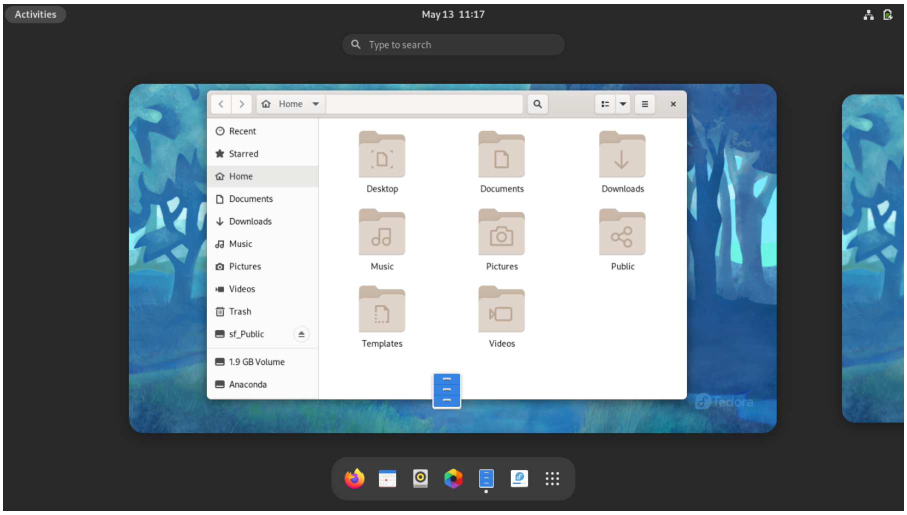


Fig. 2: GNOME

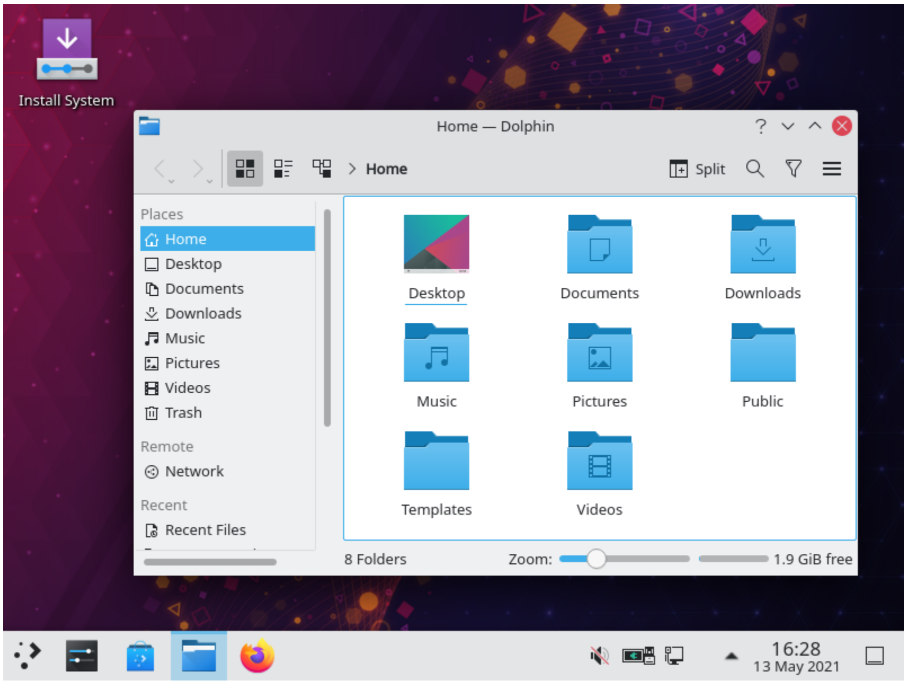


Fig. 3: KDE Plasma

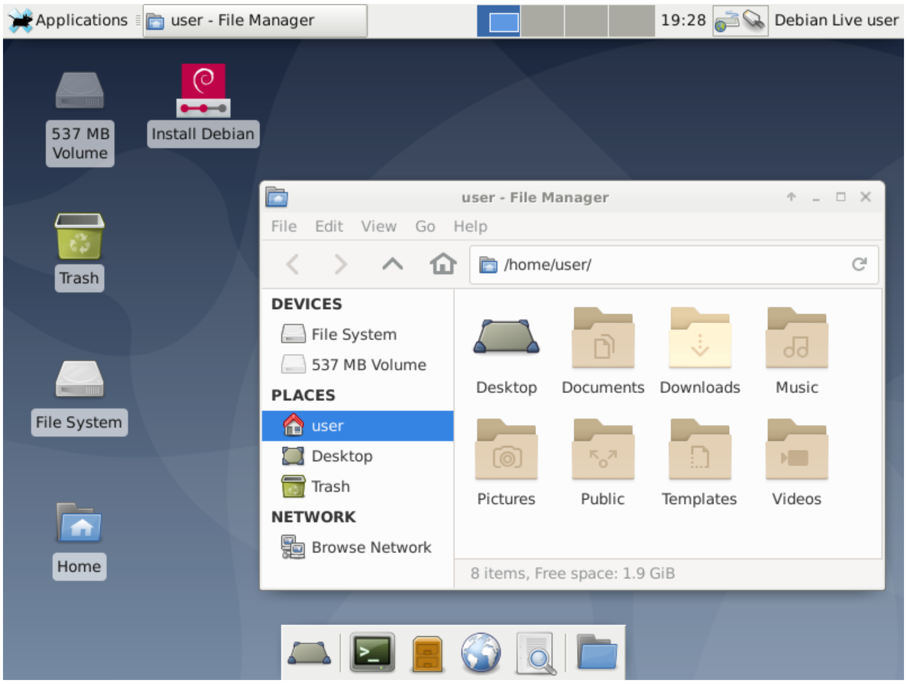

Applications
user - File Manager
19:28
Debian Live user
537 MB Volume
Install Debian
Trash
File System
Home
user - File Manager
File
Edit
View
Go
Help
/home/user/
DEVICES
File System
537 MB Volume
PLACES
user
Desktop
Trash
NETWORK
Browse Network
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
8 items, Free space: 1.9 GiB

Fig. 4: Xfce

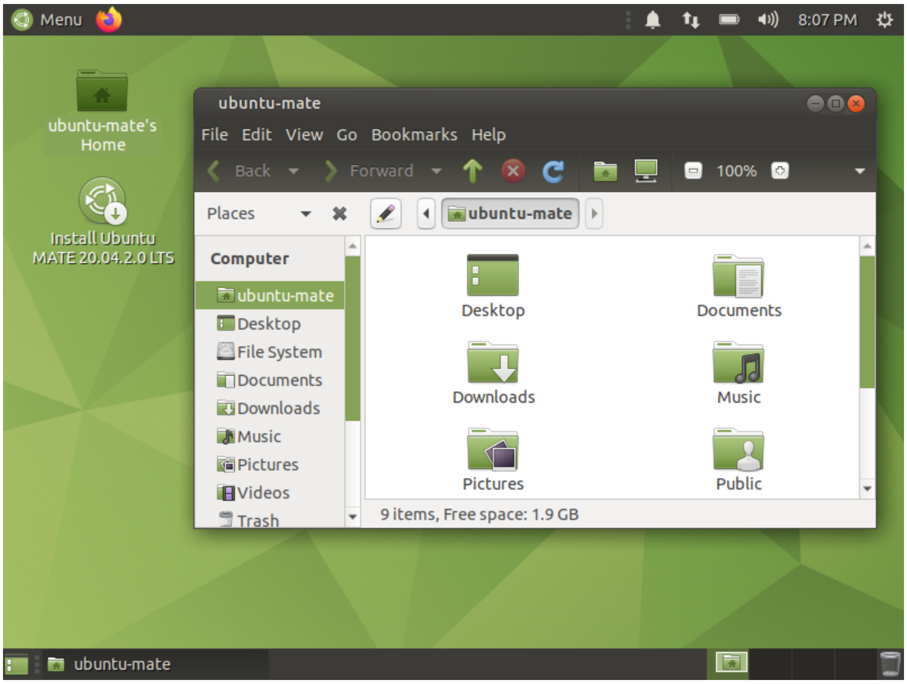


Fig. 5: MATE

This book uses Linux Mint (Cinnamon)

This book will use Linux Mint 20.1 with the Cinnamon desktop environment as the reference distribution (Fig. 6).

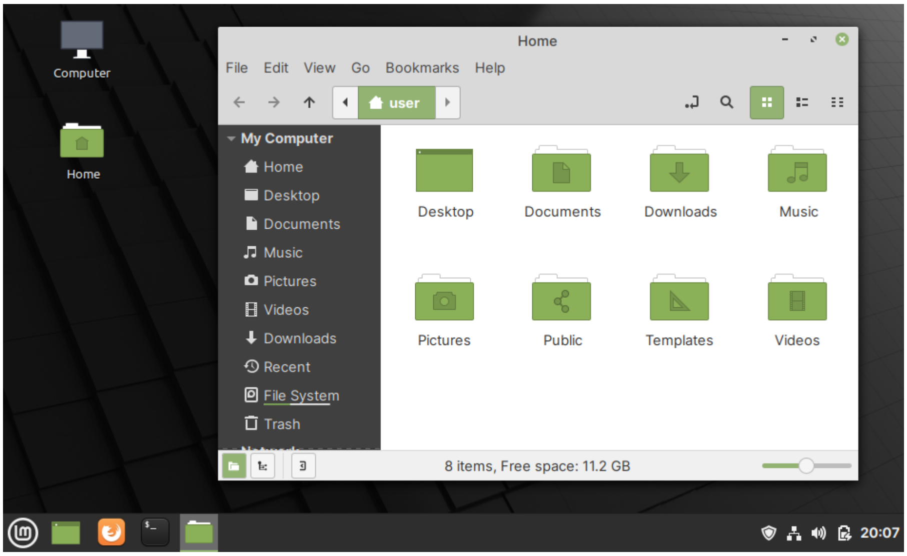


Fig. 6: Linux Mint with Cinnamon desktop environment

All the included recipes have been tested on this distribution.

2.1.3 Ways to run a Linux desktop

To run a Linux desktop on your computer, you can:

1. Install a *Linux distribution* or
2. Run a virtual machine

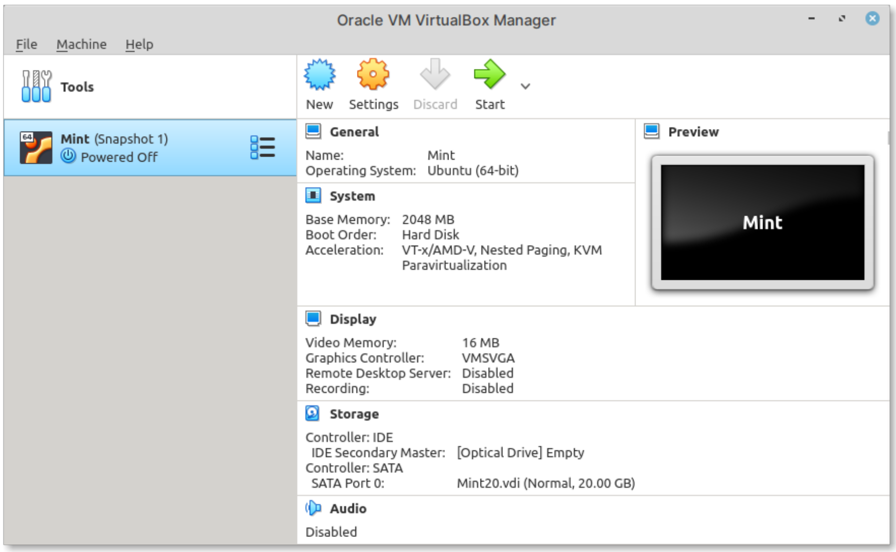
I will follow the *second* method in this book, as it is a lot easier for new users to get started.

Note: There are other possibilities, but the methods above are the most common.

2.2 Running a Linux virtual machine

To run virtual machines of other operating systems (guests) on your computer (host), you will need to first install a *hypervisor* — software that runs virtual machines.

Examples of hypervisors include VirtualBox, Parallels, VMware Workstation etc., I will use VirtualBox in this book (Fig. 7).



General
Name: Mint
Operating System: Ubuntu (64-bit)

System
Base Memory: 2048 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Mint20.vdi (Normal, 20.00 GB)

Audio
Disabled

Fig. 7: Main window of VirtualBox

2.2.1 Requirements

VirtualBox installed on your computer

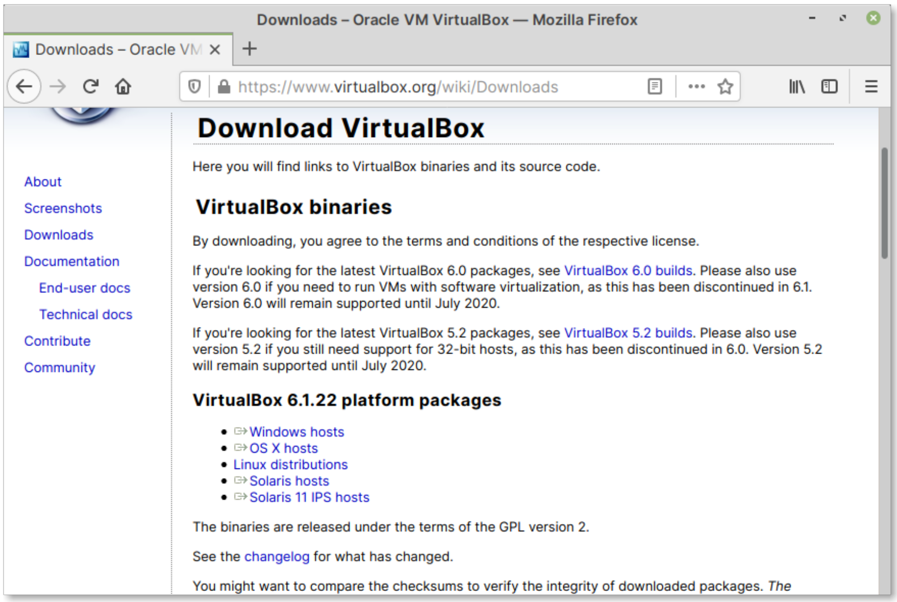
You can install VirtualBox by following the instructions below depending on your operating system.

Attention: *Installing software will require administrator privileges.*

Instructions for macOS and Windows

You can download an installer for macOS or Windows from the Downloads[8] section of the VirtualBox website (Fig. 8).

[8] https://www.virtualbox.org/wiki/Downloads



Downloads – Oracle VM VirtualBox — Mozilla Firefox

https://www.virtualbox.org/wiki/Downloads

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see VirtualBox 6.0 builds. Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds. Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0. Version 5.2 will remain supported until July 2020.

VirtualBox 6.1.22 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The*

Fig. 8: VirtualBox downloads page

Instructions for Linux distributions

Most Linux distributions include VirtualBox in their package repositories. It is recommended you install it using your package manager like *Synaptic* or *apt*, dnf etc.,

For example, to get VirtualBox working on Ubuntu 20.04 LTS, you will need to install the `virtualbox` and `virtualbox-guest-additions-iso` packages.

Note: If you cannot install from repositories, you can download an installer from the Linux Downloads[9] section of the project

[9] https://www.virtualbox.org/wiki/Linux_Downloads

website.

Virtual machine image of Linux Mint

You can download a ready-to-use virtual machine image of Linux Mint 20.1 from the website of this book:

https://vimalkvn.com/linuxforbiologists

It is distributed as an *OVA* format file.

Click on the download link and save the `linuxmint-20.1.ova` file to your computer.

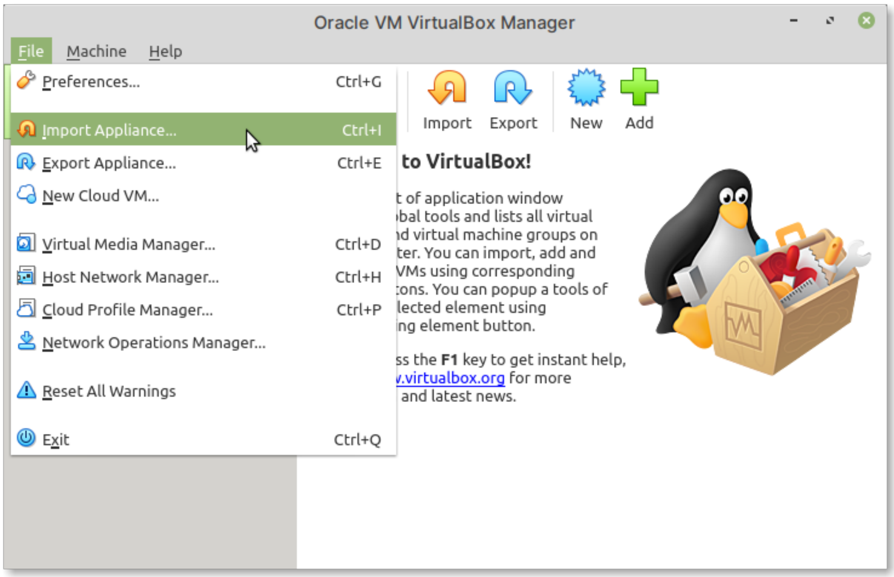
2.2.2 Importing the virtual machine image

Open VirtualBox on your computer and then follow the steps below to import the downloaded virtual machine image.

Step 1 — Select OVA file

From the main window of VirtualBox, select:

File → Import Appliance (Fig. 9)


Oracle VM VirtualBox Manager
File
Machine
Help
Preferences... Ctrl+G
Import Appliance... Ctrl+I
Export Appliance... Ctrl+E
New Cloud VM...
Virtual Media Manager... Ctrl+D
Host Network Manager... Ctrl+H
Cloud Profile Manager... Ctrl+P
Network Operations Manager...
Reset All Warnings
Exit Ctrl+Q
Import
Export
New
Add

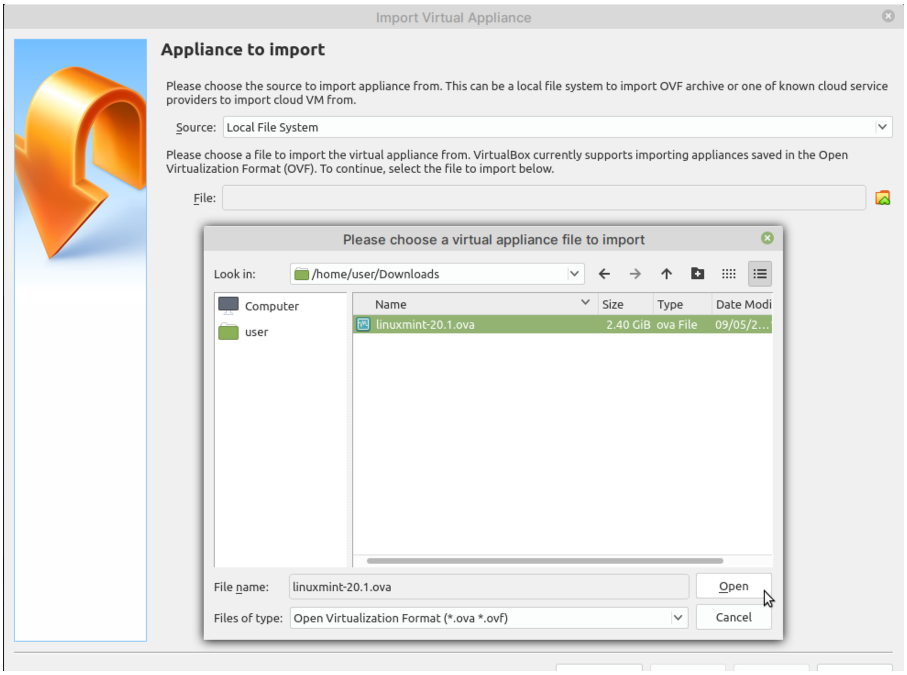
to VirtualBox!

t of application window
bal tools and lists all virtual
d virtual machine groups on
ter. You can import, add and
VMs using corresponding
ons. You can popup a tools of
lected element using
ng element button.

ss the **F1** key to get instant help,
.virtualbox.org for more
and latest news.

Fig. 9: Select File –> Import Appliance from the menu

In the Appliance to import screen, click on the button that appears next to the File entry field (Fig. 10).



Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File:

Please choose a virtual appliance file to import
Look in: /home/user/Downloads
Computer
user

Name	Size	Type	Date Modi
linuxmint-20.1.ova	2.40 GiB	ova File	09/05/2…

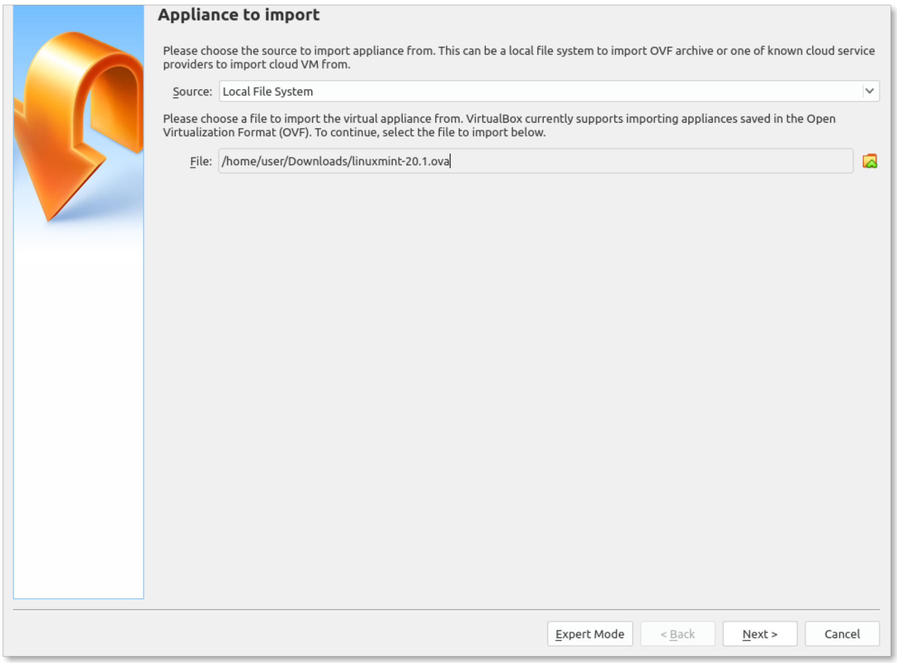
File name: linuxmint-20.1.ova

Files of type: Open Virtualization Format (*.ova *.ovf)

Fig. 10: Click on the button that appears next to the File entry field

Browse to the location of the `linuxmint-20.1.ova` file, select it, and then click on the Open button.

The File entry field will now have the complete path to the selected file (Fig. 11).


Appliance to import

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File: /home/user/Downloads/linuxmint-20.1.ova

Fig. 11: Click on the Next button to proceed

Click on the Next button to proceed.

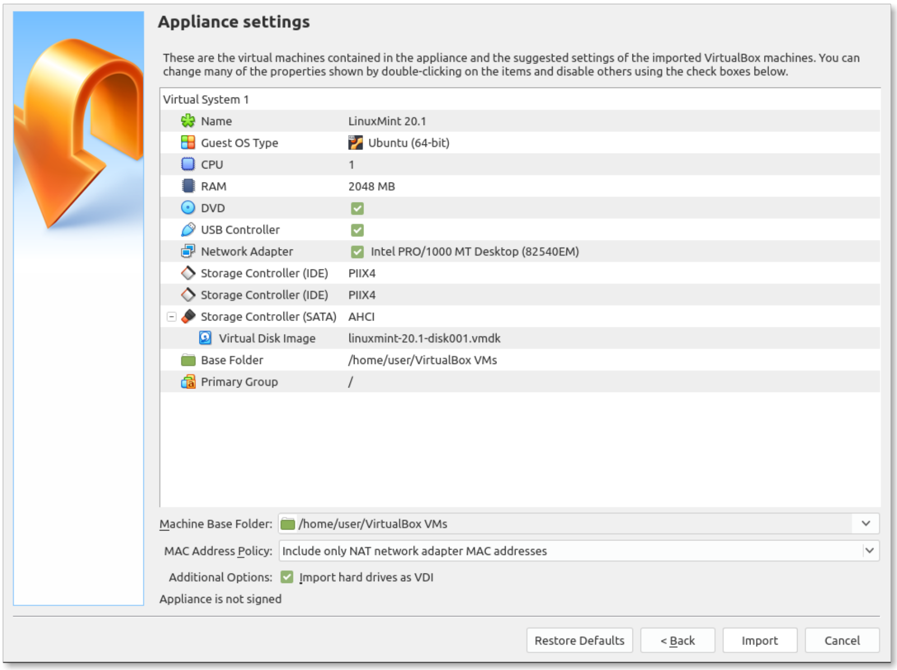
Step 2 — Adjust appliance settings (optional)

In the next screen — Appliance settings (Fig. 12), you can modify the configuration of this virtual machine.

Note: *This step is optional.*

You can click on the Import button at the bottom to proceed to the *next step*.

For example, you can add more CPU or RAM for this virtual machine, depending on the configuration of your computer.



Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Virtual System 1	
Name	LinuxMint 20.1
Guest OS Type	Ubuntu (64-bit)
CPU	1
RAM	2048 MB
DVD	✓
USB Controller	✓
Network Adapter	✓ Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4
Storage Controller (IDE)	PIIX4
Storage Controller (SATA)	AHCI
Virtual Disk Image	linuxmint-20.1-disk001.vmdk
Base Folder	/home/user/VirtualBox VMs
Primary Group	/

Machine Base Folder: /home/user/VirtualBox VMs

MAC Address Policy: Include only NAT network adapter MAC addresses

Additional Options: ✓ Import hard drives as VDI

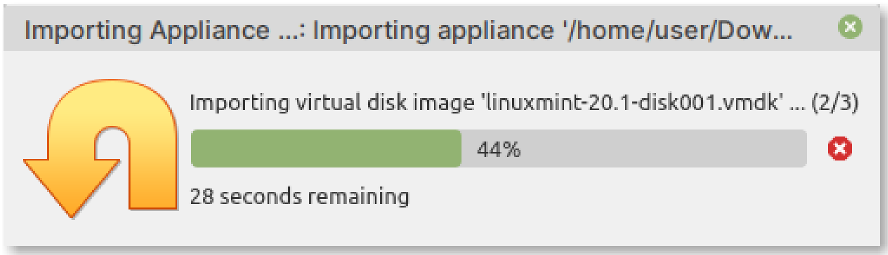
Appliance is not signed

Fig. 12: Appliance settings screen

Click on the Import button to proceed.

Step 3 — Import progress and completion

You will notice a progress bar at this stage (Fig. 13).


Importing Appliance ...: Importing appliance '/home/user/Dow...

Importing virtual disk image 'linuxmint-20.1-disk001.vmdk' ... (2/3)

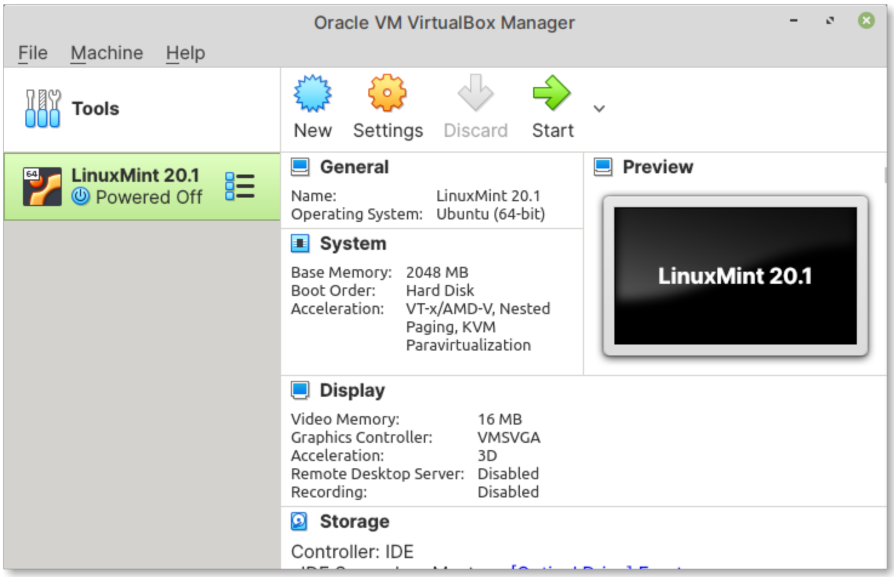
44%

28 seconds remaining

Fig. 13: Virtual machine import progress

When import is complete, you will be taken back to the main window of VirtualBox.

The imported virtual machine — LinuxMint 20.1, will appear in the panel on the left (Fig. 14).



Name: LinuxMint 20.1
Operating System: Ubuntu (64-bit)

Base Memory: 2048 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

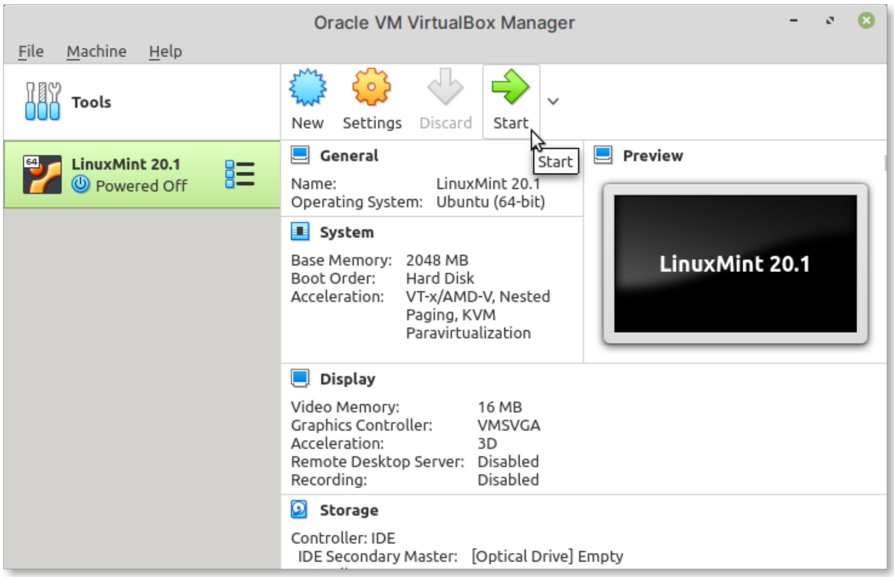
Video Memory: 16 MB
Graphics Controller: VMSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

Controller: IDE

Fig. 14: Imported virtual machine

2.2.3 Starting the virtual machine

To start the virtual machine, select LinuxMint 20.1 and then click on the Start button in the main toolbar (Fig. 15).



General

Name: LinuxMint 20.1
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

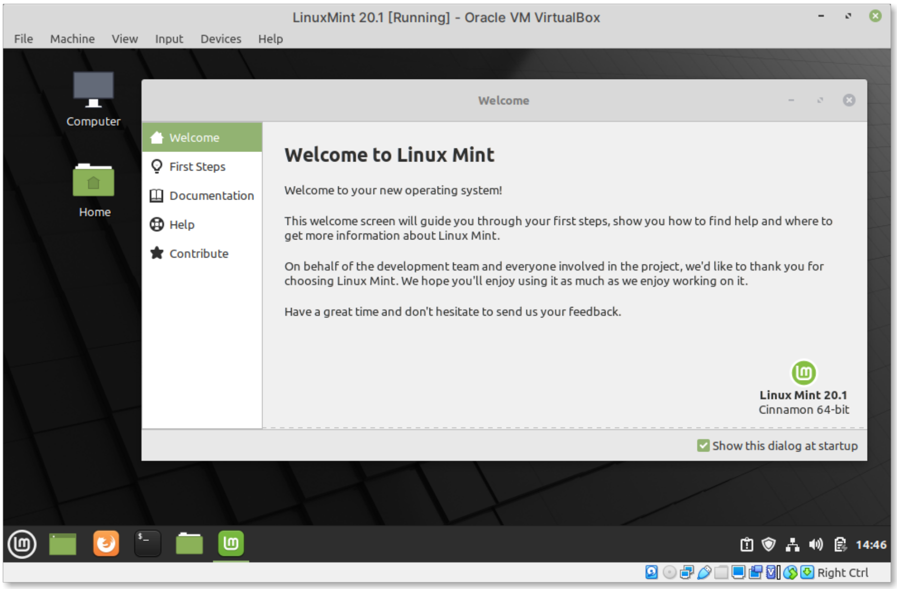
Video Memory: 16 MB
Graphics Controller: VMSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] Empty

Fig. 15: Start virtual machine

Once the boot process is complete, you will be logged in directly to the desktop (Fig. 16).



Welcome to Linux Mint

Welcome to your new operating system!

This welcome screen will guide you through your first steps, show you how to find help and where to get more information about Linux Mint.

On behalf of the development team and everyone involved in the project, we'd like to thank you for choosing Linux Mint. We hope you'll enjoy using it as much as we enjoy working on it.

Have a great time and don't hesitate to send us your feedback.

Fig. 16: The Linux Mint desktop

For reference, the default username is `user` and password is also `user`.

2.2.4 Stopping the virtual machine

When your work is complete, you can shut down the virtual machine by clicking on the Quit button in Linux Mint's Applications Menu.

This will display a confirmation dialog (Fig. 17).

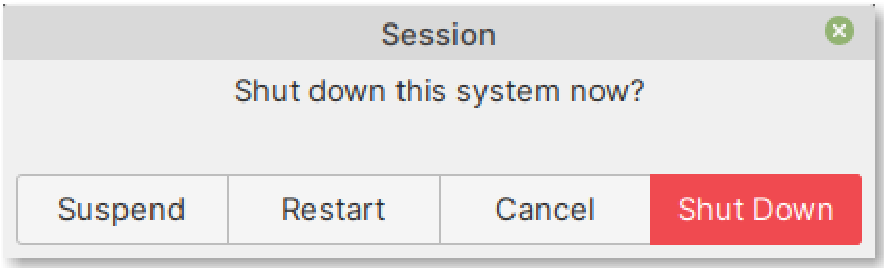


Fig. 17: Confirmation dialog for system shutdown

Click on the Shut Down button to stop the virtual machine.

2.3 The desktop

Once you have started the virtual machine, you will be logged in directly to the desktop.

For reference, the default user account for logging in to the virtual machine, is:

Username: `user`

Password: `user`

2.3.1 The Cinnamon desktop

In its default configuration, the Cinnamon desktop includes the following components (Fig. 18):

Applications Menu — search and launch installed applications, logout, shutdown or restart system.

Panel — launch pinned applications, manage open applications windows.

System Tray — contains applets for managing the network, sound, power, software updates etc.,

Calendar — provides current date and time, and a calendar.

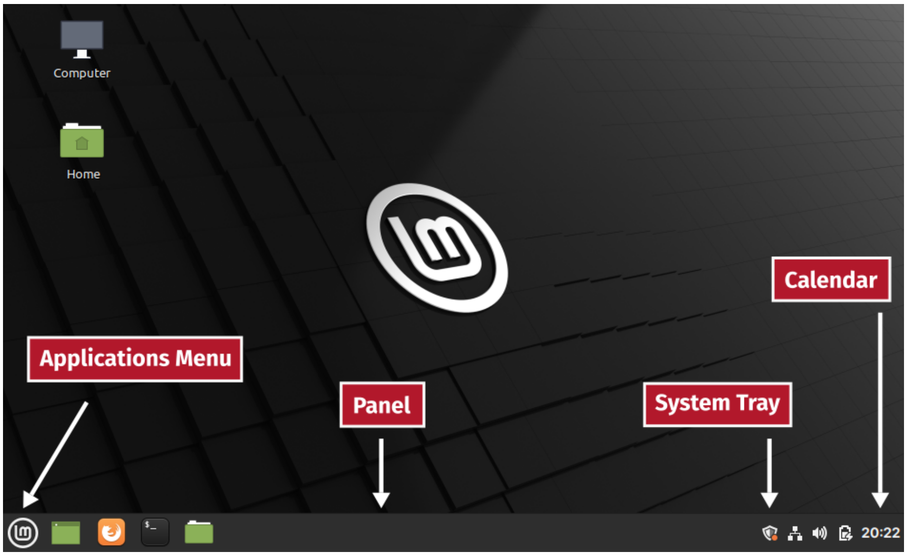


Fig. 18: Components of the Cinnamon desktop

2.3.2 Changing system settings

The System Settings application can be used to customize the desktop, the operating system and hardware (Fig. 19).

You can launch it from the applications' menu.

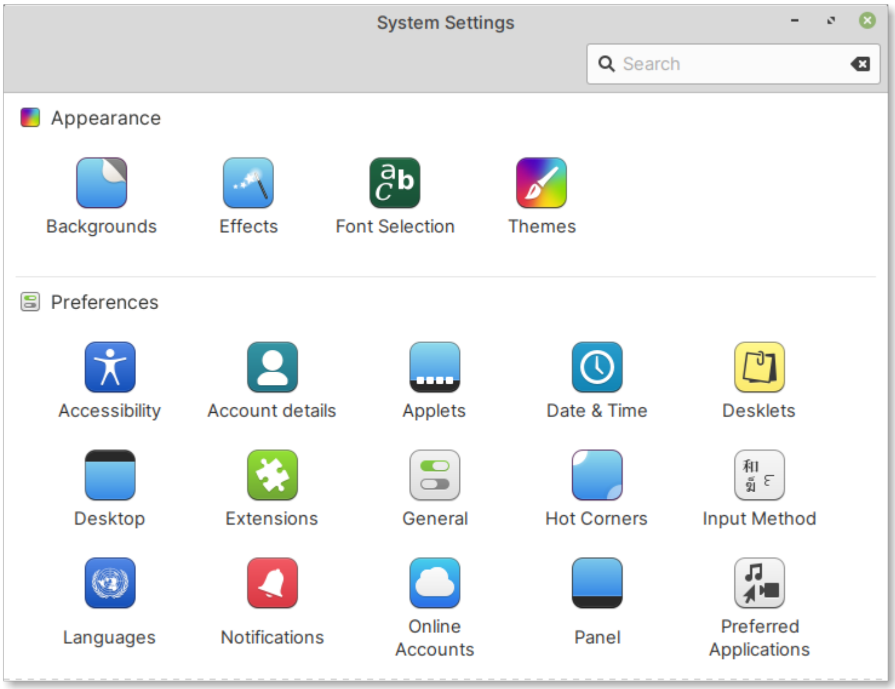


Fig. 19: System Settings application

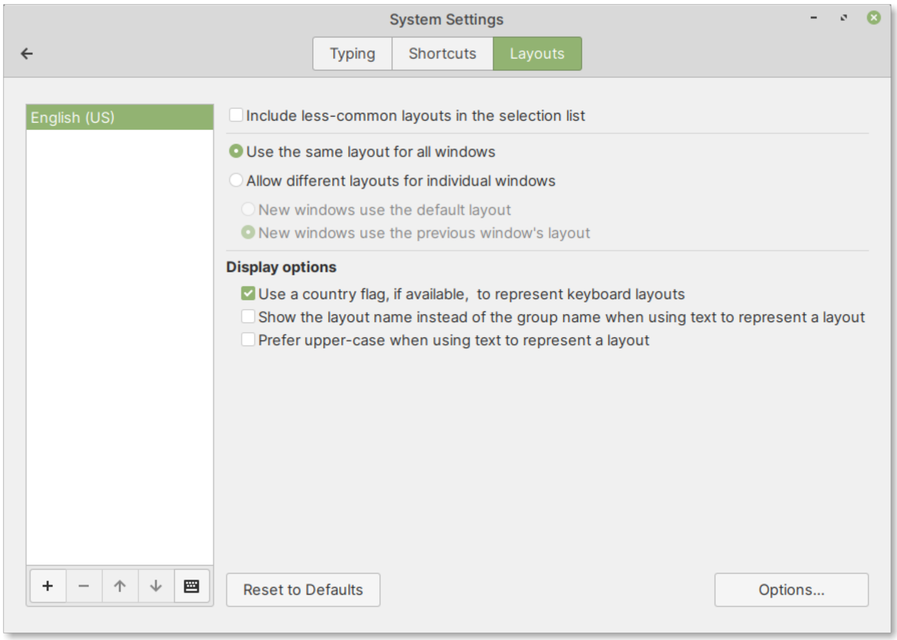
The settings you can configure here include accessibility, user accounts, date and time, keyboard layout, language, network, display and more.

Changing keyboard layout

The default keyboard layout of the virtual machine is English (US).

You can change this under:

Hardware → Keyboard → Layouts



English (US)

[ ] Include less-common layouts in the selection list

(●) Use the same layout for all windows

( ) Allow different layouts for individual windows

( ) New windows use the default layout

(●) New windows use the previous window's layout

Display options

[✓] Use a country flag, if available, to represent keyboard layouts

[ ] Show the layout name instead of the group name when using text to represent a layout

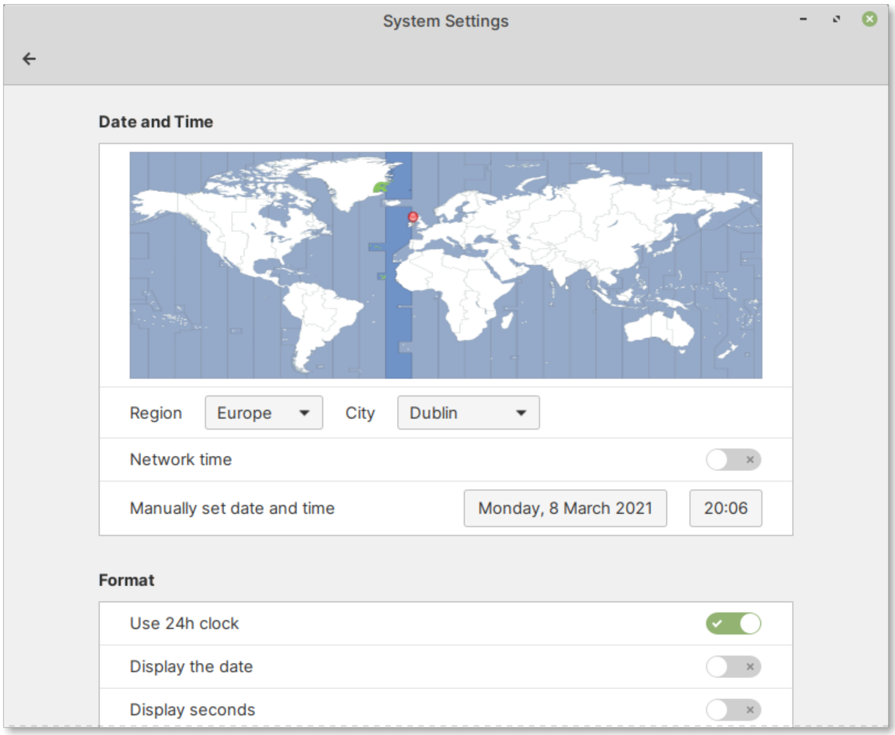
[ ] Prefer upper-case when using text to represent a layout

Fig. 20: Keyboard layouts in System Settings

Changing system date and time

The system date and time can be updated by selecting:

Preferences → Date and Time



Date and Time

Region: Europe
City: Dublin
Network time: off
Manually set date and time: Monday, 8 March 2021 — 20:06

Format

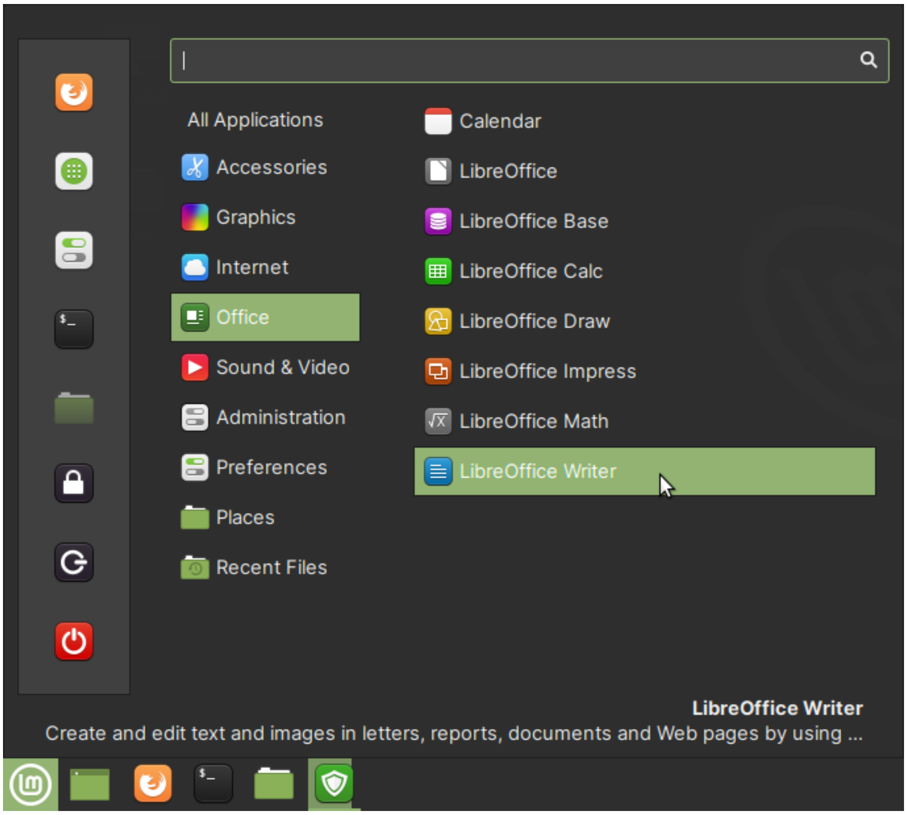
Use 24h clock: on
Display the date: off
Display seconds: off

Fig. 21: Date and time settings

2.4 Available software

This is a brief overview of applications included in Linux Mint 20.1 Cinnamon.

You can launch them from the Applications Menu (Fig. 22). You can search the list of installed applications using the search bar.



LibreOffice Writer
Create and edit text and images in letters, reports, documents and Web pages by using ...

Fig. 22: Launching an application

Note: Some applications discussed here are not included in the default installation.

This includes *GNU Image Manipulation Program* and *Inkscape*. You can install them using *Software Manager*.

2.4.1 Files — manage files and directories

Files is the default file manager (Fig. 23) in Linux Mint.

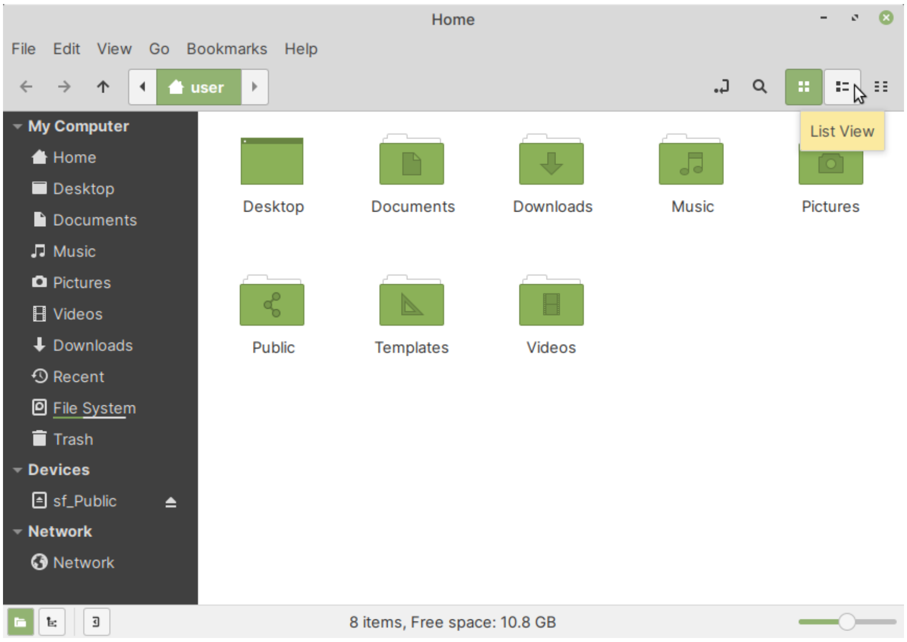


Fig. 23: Main window of Files

Using Files

You can launch Files from the Applications Menu or by clicking on its icon in the Panel.

These are some tips on using Files.

Toggle icon and list views

You can quickly switch between Icon and List View using icons in the toolbar.

Toggle sidebar

You can open the sidebar view or hide it using icons in the status bar.

Open an extra pane

You can open an extra pane using:

View → Extra Pane

This is especially useful when you are copying files between local directories or when connected to remote systems.

Bookmark locations

When you are browsing a directory, you can bookmark it if you will access it frequently. This function is available under:

Bookmark → Add Bookmark

Connect to servers

Files can be used to connect to external servers using FTP, SSH, Samba and Webdav. This option is available under:

File → Connect to Server

To learn more about this feature, read the following section in this book: *Transferring files between systems*.

2.4.2 Firefox — browse the web

Firefox is the default web browser (Fig. 24).

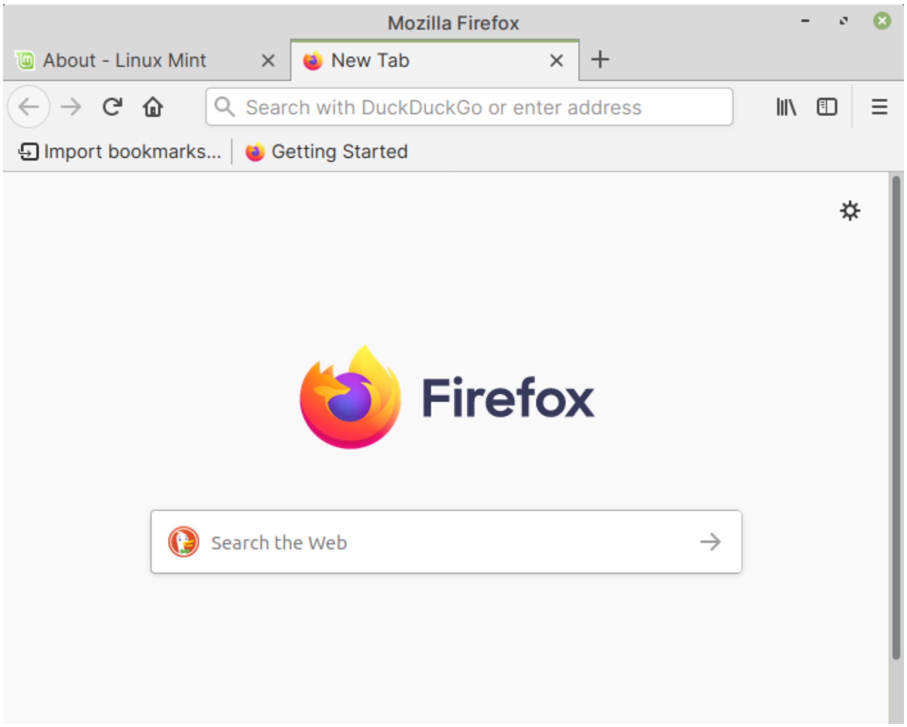
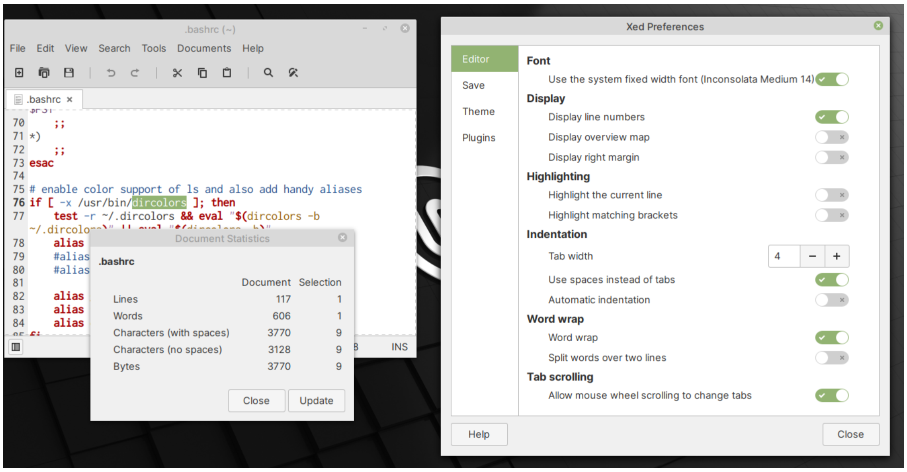


Fig. 24: Firefox web browser

Other web browsers can be installed using *Software Manager*.

2.4.3 Text Editor — create and edit text files

The Text Editor can be used for working with files in plain text formats like `.txt`, `.md` and `.fasta`, or editing configuration files like `.bashrc` (Fig. 25).



```
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b
~/.dircolo
    alias
    #alias
    #alias

    alias
    alias
    alias
```

Document Statistics

.bashrc

	Document	Selection
Lines	117	1
Words	606	1
Characters (with spaces)	3770	9
Characters (no spaces)	3128	9
Bytes	3770	9

Close Update

Xed Preferences

Font

Use the system fixed width font (Inconsolata Medium 14)	✓

Display

Display line numbers	✓
Display overview map	✗
Display right margin	✗

Highlighting

Highlight the current line	✗
Highlight matching brackets	✗

Indentation

Tab width	4
Use spaces instead of tabs	✓
Automatic indentation	✗

Word wrap

Word wrap	✓
Split words over two lines	✗

Tab scrolling

Allow mouse wheel scrolling to change tabs	✓

Fig. 25: Text Editor with Document Statistics and Preferences windows open

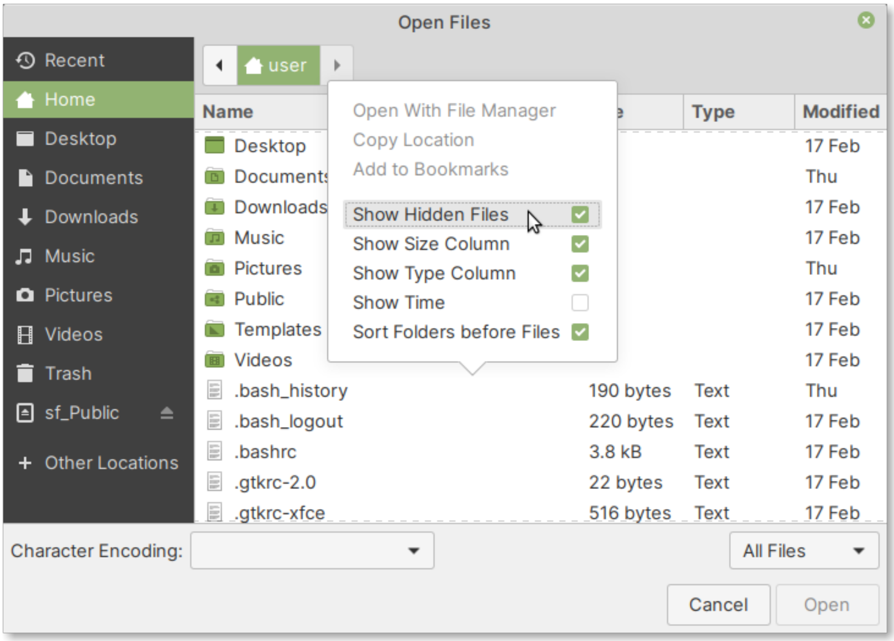
Using Edit → Preferences, you can customize settings like the font used, display of line numbers, indentation, word wrap etc.,

The Tools → Document Statistics feature is useful for counting the numbers of words and characters in a document.

Opening hidden files in Text Editor

To open hidden files in Text Editor:

1. Use File → Open to select a file
2. In the Open Files dialog that appears, right-click in the middle pane and select Show Hidden Files (Fig. 26).



Show Hidden Files ✓
Show Size Column ✓
Show Type Column ✓
Show Time ☐
Sort Folders before Files ✓

Name	e	Type	Modified
Desktop			17 Feb
Documents			Thu
Downloads			17 Feb
Music			17 Feb
Pictures			Thu
Public			17 Feb
Templates			17 Feb
Videos			17 Feb
.bash_history	190 bytes	Text	Thu
.bash_logout	220 bytes	Text	17 Feb
.bashrc	3.8 kB	Text	17 Feb
.gtkrc-2.0	22 bytes	Text	17 Feb
.gtkrc-xfce	516 bytes	Text	17 Feb

Fig. 26: Opening hidden files in Text Editor

You can also use the `Ctrl + h` keyboard shortcut.

2.4.4 LibreOffice — edit documents and spreadsheets

LibreOffice Writer and LibreOffice Calc can be used to create and edit documents and spreadsheets, respectively (Fig. 27).

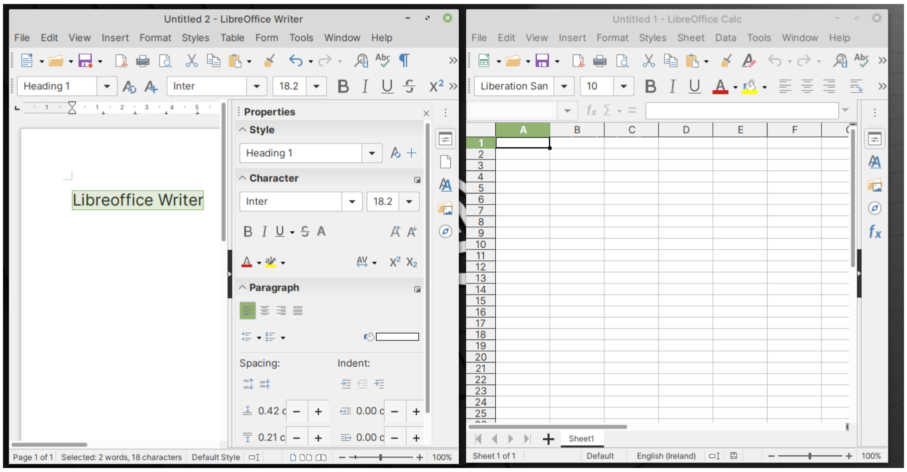

Untitled 2 - LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help
Heading 1
Inter
18.2
Libreoffice Writer
Properties
Style
Character
Paragraph
Spacing:
Indent:
Page 1 of 1 Selected: 2 words, 18 characters Default Style
Untitled 1 - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation San
10
Sheet1
Sheet 1 of 1 Default English (Ireland) 100%

Fig. 27: LibreOffice Writer and Calc programs

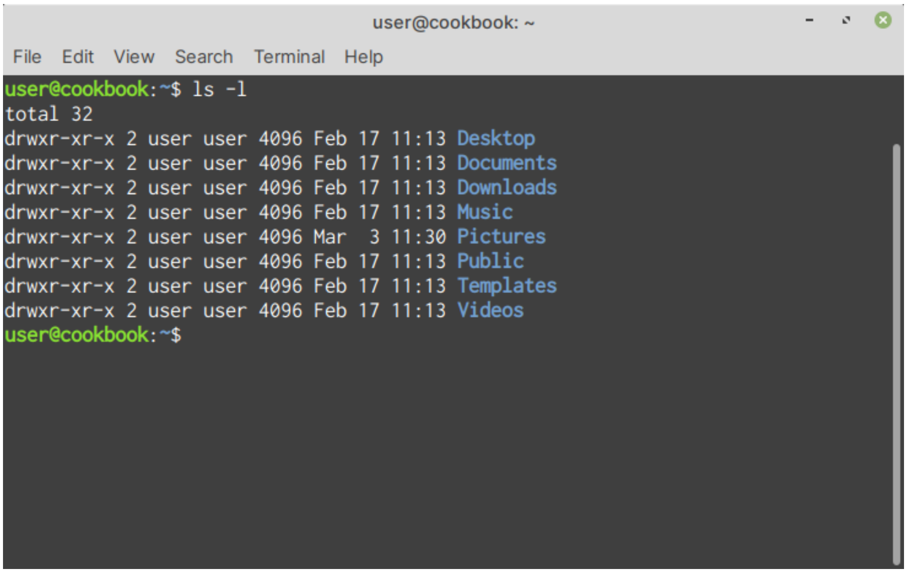
The default file formats used by Writer and Calc are ODF Text Document (.odt) and ODF Spreadsheet (.ods) respectively.

These applications also support reading and writing files in Microsoft Office formats like `.docx` or `.doc` (Word) and `.xlsx` or `.xls` (Excel).

There are additional applications in the LibreOffice suite with functionality to create and edit slide shows or presentations, scientific formulas, diagrams and file-based databases.

2.4.5 Terminal — access the command-line

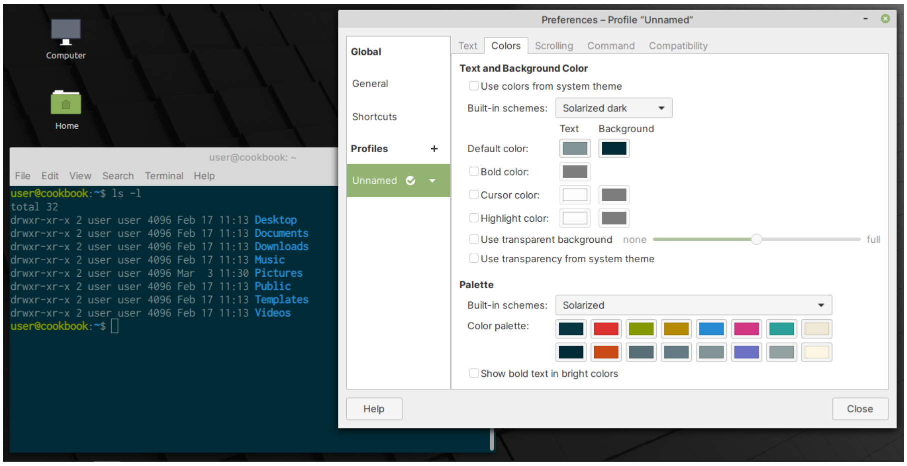
The Terminal application (Fig. 28) provides the possibility of working with the command-line interface. The default login shell for users on Linux Mint is *Bash*.



```
user@cookbook:~$ ls -l
total 32
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Desktop
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Documents
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Downloads
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Music
drwxr-xr-x 2 user user 4096 Mar  3 11:30 Pictures
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Public
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Templates
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Videos
user@cookbook:~$
```

Fig. 28: Terminal

By selecting the `Edit` → `Preferences` menu item, you can customize Terminal's appearance and functionality (Fig. 29).


Computer
Home
user@cookbook: ~
File Edit View Search Terminal Help
Preferences – Profile "Unnamed"
Global
General
Shortcuts
Profiles
Unnamed
Text Colors Scrolling Command Compatibility
Help
Close

```
user@cookbook:~$ ls -l
total 32
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Desktop
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Documents
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Downloads
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Music
drwxr-xr-x 2 user user 4096 Mar  3 11:30 Pictures
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Public
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Templates
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Videos
user@cookbook:~$
```

Text and Background Color

☐ Use colors from system theme

Built-in schemes: Solarized dark

Text Background

Default color:

☐ Bold color:

☐ Cursor color:

☐ Highlight color:

☐ Use transparent background none full

☐ Use transparency from system theme

Palette

Built-in schemes: Solarized

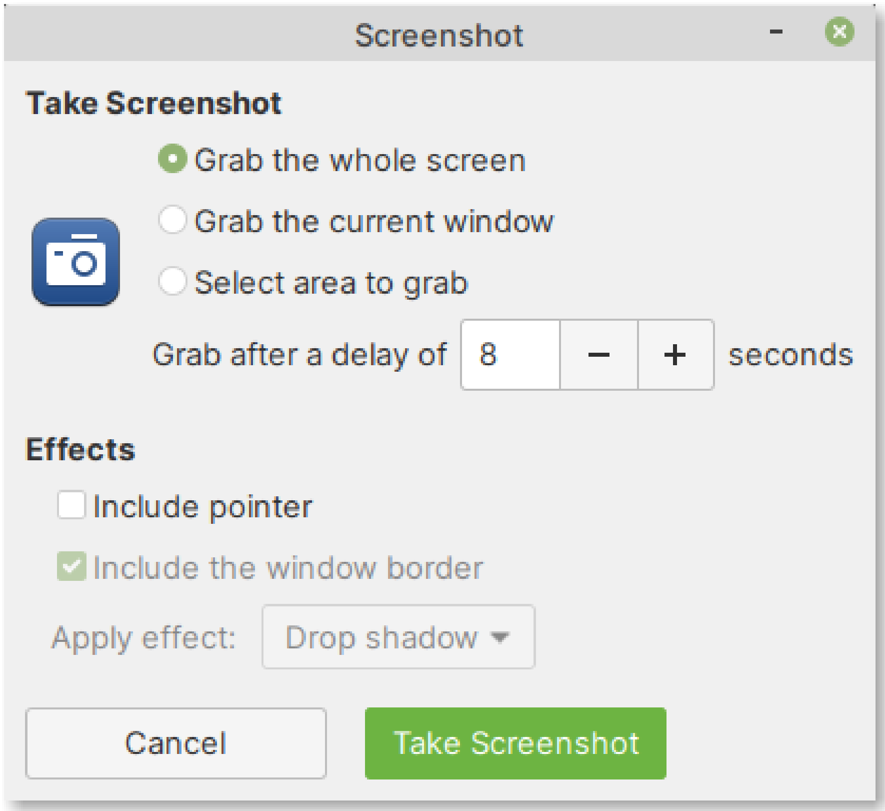
Color palette:

☐ Show bold text in bright colors

Fig. 29: Changing Terminal colour scheme in preferences

2.4.6 Screenshot — take screenshots of desktop or windows

The Screenshot program is useful for taking screenshots of the entire screen, an application window or a selected area of the screen (Fig. 30).



Take Screenshot

[✓ Grab the whole screen] Grab the current window Select area to grab

Grab after a delay of 8 seconds

Effects

☐ Include pointer

☑ Include the window border

Apply effect: Drop shadow

Fig. 30: Screenshot program

2.4.7 Software Manager — install software from repositories

Software Manager can be used to install software from Linux Mint, Ubuntu, Debian and Flathub repositories (Fig. 31).

For a step-by-step procedure for installing software using Software Manager, please consult *The quick and easy method*.

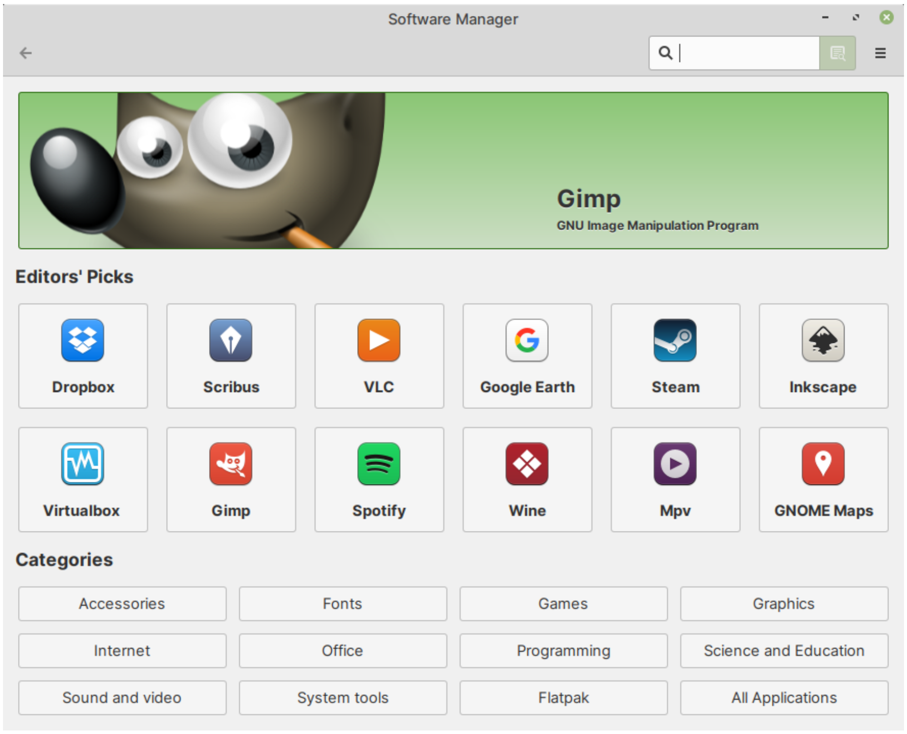
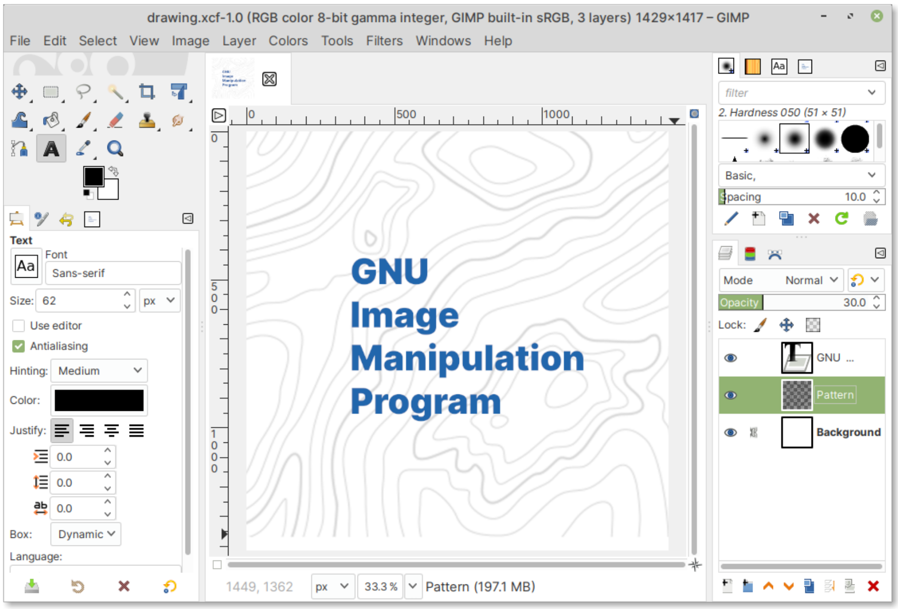


Fig. 31: Software Manager

2.4.8 GNU Image Manipulation Program — edit images

The GIMP program can be used to edit images in various formats like JPG, PNG, TIFF, BMP etc., (Fig. 32).


drawing.xcf-1.0 (RGB color 8-bit gamma integer, GIMP built-in sRGB, 3 layers) 1429×1417 – GIMP

Text
Font: Sans-serif
Size: 62 px
Use editor
Antialiasing
Hinting: Medium
Color:
Justify:
0.0
0.0
0.0
Box: Dynamic
Language:

2. Hardness 050 (51 × 51)
Basic,
Spacing 10.0

Mode Normal
Opacity 30.0
Lock:
GNU ...
Pattern
Background

1449, 1362 px 33.3 % Pattern (197.1 MB)

Fig. 32: Editing an image in GIMP

2.4.9 Inkscape — create vector graphics

Inkscape can be used to create and edit images in the scalable vector graphics (SVG) format (Fig. 33).

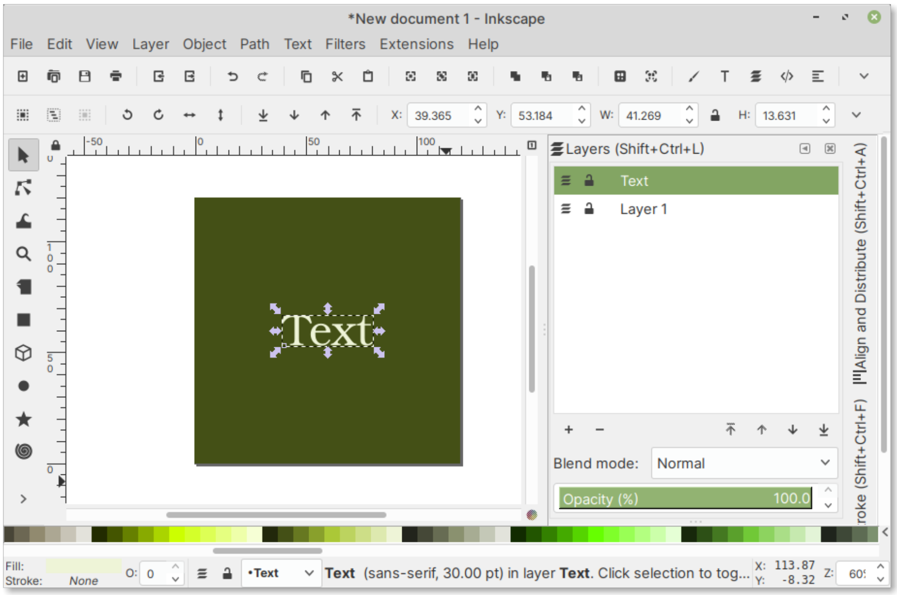

*New document 1 - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X: 39.365 Y: 53.184 W: 41.269 H: 13.631
Layers (Shift+Ctrl+L)
Text
Layer 1
Blend mode: Normal
Opacity (%) 100.0
Align and Distribute (Shift+Ctrl+A)
Text
Fill:
Stroke: None
•Text
Text (sans-serif, 30.00 pt) in layer Text. Click selection to tog...
X: 113.87 Y: -8.32

Fig. 33: Creating vector graphics in Inkscape

2.4.10 Notes

Adding an application to favourites

If you use an application frequently, you can consider adding it to favourites. This list is on the left side of the applications' menu.

To favourite an application, right-click on the application name in applications menu and select Add to favorites.

Alternatively, you can choose the Add to panel action, which will add the application to your Panel. Some applications are visible in the Panel by default — a Show Desktop applet, Firefox browser, Terminal, and Files.

Setting default applications

If you would like to set default applications for certain categories or file types, you can do so in System Settings under:

Preferences → Preferred applications

2.5 Files and directories

This section includes an introduction to your home directory, files, directories, and their functions, the root filesystem and a how-to on transferring files between systems.

2.5.1 Your home directory

When you log in to the desktop and open *Files*, it will display your home directory (Fig. 34).

The following directories will be present in your home directory by default: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos.

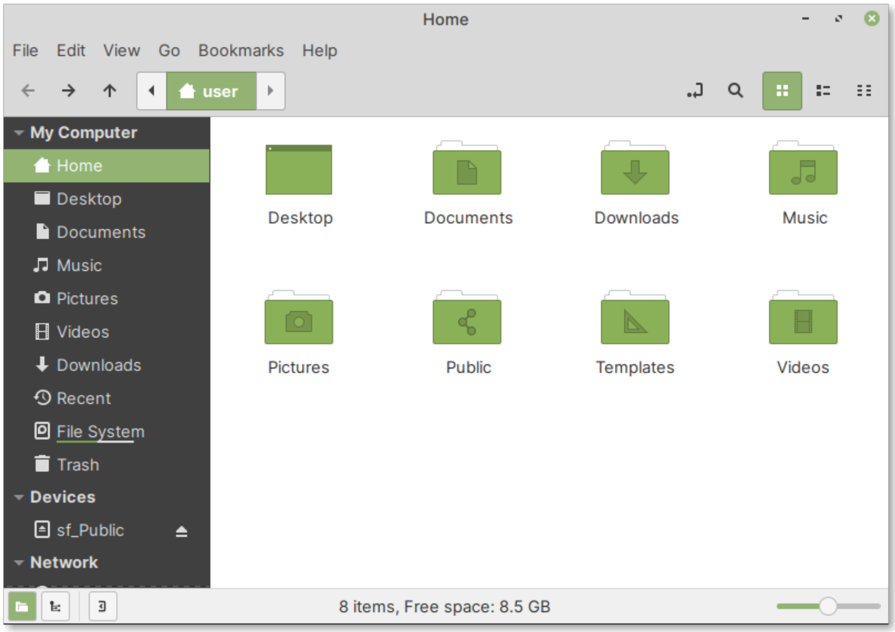


Fig. 34: Home directory

Notes

/home/user

The complete (or absolute) path to your home directory will be of the form `/home/user`, where `user` is your username on the system.

$HOME

The `HOME` environment variable will automatically be set to the complete path to your home directory in the filesystem.

Hence, you can substitute `/home/user` with `$HOME` in your commands or scripts.

2.5.2 Hidden files and directories

In addition to the default directories mentioned earlier, there are hidden files and directories in home directory (Fig. 35). They have a dot (`.`) character at the beginning of their file name.

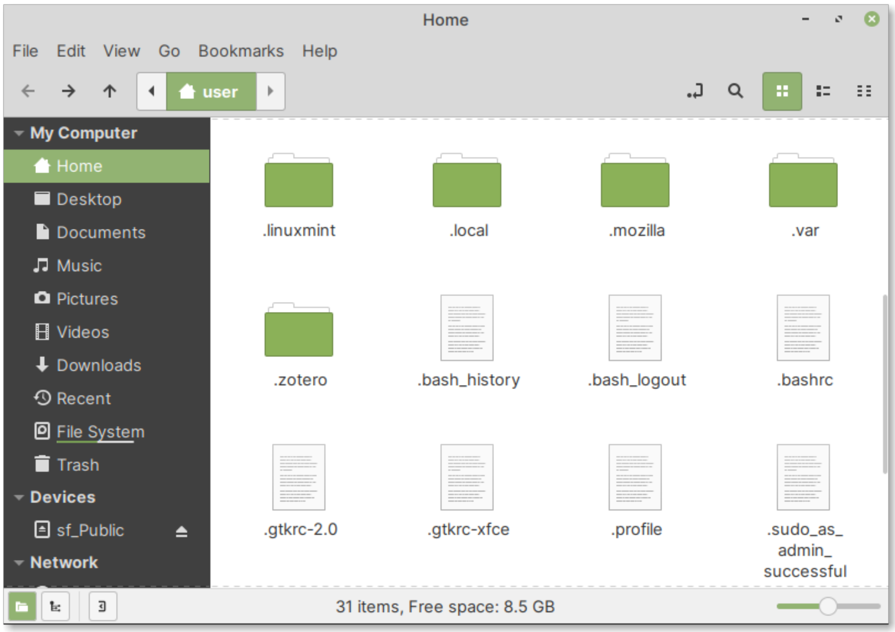


Fig. 35: Hidden files and directories in home directory

You can view (or hide) these files using the following menu entry in file manager:

View → Show Hidden Files

Alternatively, you can use the `Ctrl + h` keyboard shortcut.

2.5.3 Important files in home directory

`.bashrc`

This is the configuration file used by Bash, the default shell for user accounts. You will mostly use this file, when you need to set or modify environment variables like `PATH`.

`.bash_aliases`

You can use the `.bash_aliases` file to set aliases for commands.

For example, here is a commonly used alias:

```
alias l='ls -l'
```

This means when you type `l` at the command-line, the bash shell will execute `ls -l`, which will output a long listing of files, instead of the default.

Note: To learn more about the `ls` command, read the *ls — list files* section.

2.5.4 File and directory names are case-sensitive

One thing you will need to remember is that file and directory names in Linux are case-sensitive.

For example, a file named:

```
sequence.txt
```

is different from:

```
Sequence.txt
```

Similarly, `Documents` and `documents` are two different directories.

Both these files or directories mentioned above, can exist in the same directory and can have different contents.

2.5.5 Accessing the root filesystem

To access the root filesystem (/), click on `File System` in the sidebar of file manager (Fig. 36).

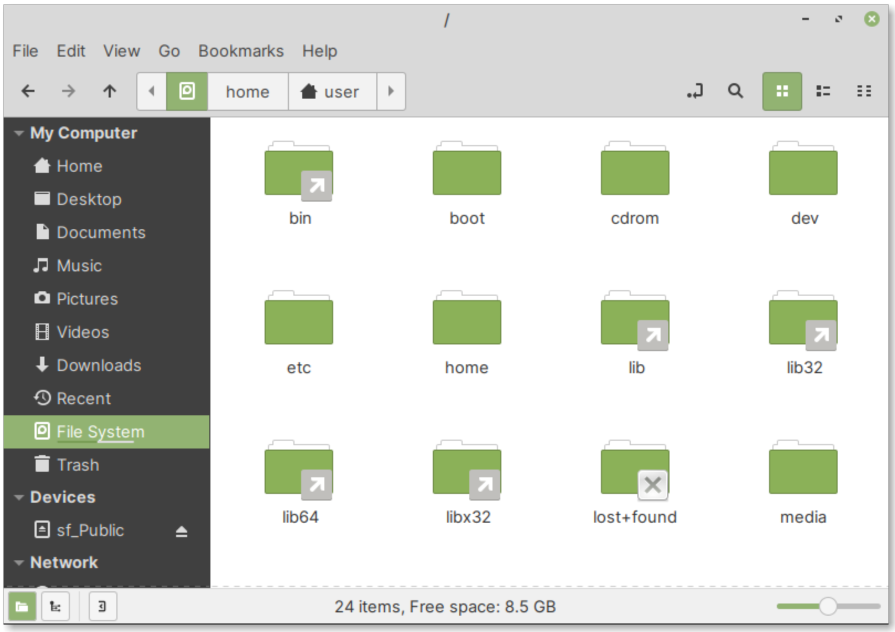


Fig. 36: Click on File System entry to access the root filesystem

Directories under / and their functions

`/etc` — stores configuration files of applications

`/home` — contains home directories of users

`/media` — locations where devices like external disks will be mounted i.e., made available

`/root` — home directory of `root` user (administrator)

`/tmp` — temporary files created by applications

`/usr` — applications and libraries are installed here along with their data and documentation

`/var` — storage for log files, application cache, databases etc.,

Root file system and root home directory are different

/ is the root filesystem

All users on the system can access files and directories here, provided they have the appropriate permissions.

`/root` is home directory of root

Only root will have access to it.

2.5.6 Transferring files between systems

One useful feature of Files is the ability to connect to remote systems using protocols like SSH, FTP, SMB and Webdav. Once connected, you can transfer files from your Linux desktop to the remote system or *vice-versa*.

To connect to a remote system, use the following menu entry in Files (Fig. 37):

File → Connect to Server

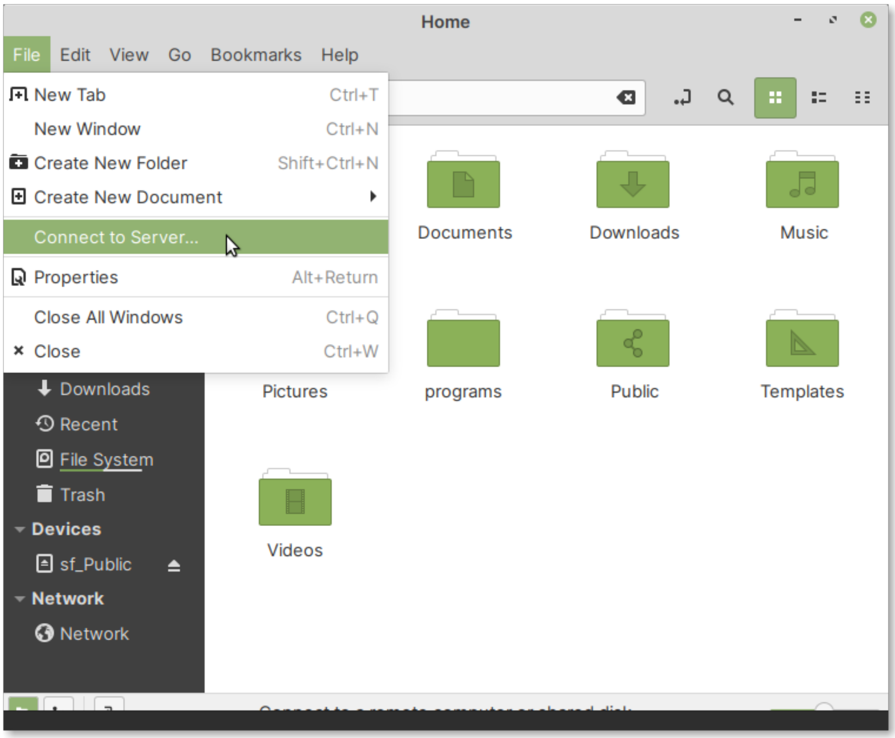
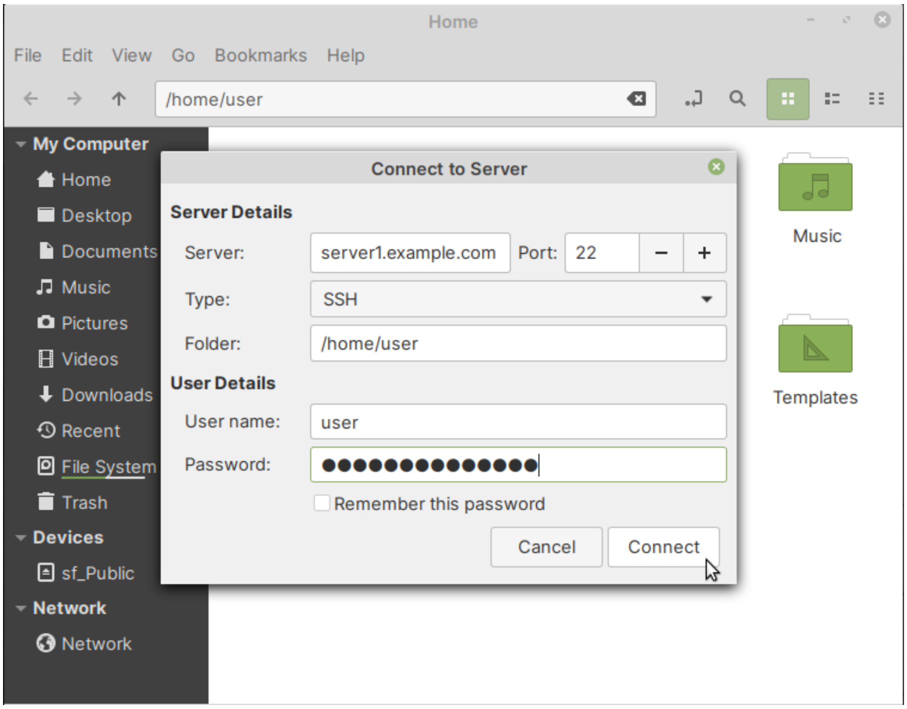


Fig. 37: Connect to Server option in Files

Connecting to an SSH server

Open File → Connect to Server and follow the steps below (Fig. 38):

1. In the field corresponding to Server, enter the domain name or IP address of the server you are connecting to
2. Port 22 is the default for SSH, so you do not need to change that usually
3. From the Type drop-down box, select SSH
4. Under Folder, you can specify a directory (optional) to open, once connection is successful
5. Enter the username and password of your account *on the server*
6. Click on the Connect button



Server: server1.example.com Port: 22
Type: SSH
Folder: /home/user
User name: user
Password: ●●●●●●●●●●●●●●
Remember this password

Fig. 38: Connecting to an SSH server

Once the remote directory is open in Files, you can start transferring files from your desktop or *vice-versa*.

3 Getting software on Linux

In this section, I will discuss how you can get software on your Linux desktop.

For any software you would like to install, you can start with the *quick and easy* method. It will install software available in Linux distribution repositories.

If your software of interest is not available or if you need a different version, you can consider using the other methods.

3.1 The quick and easy method

The *quick and easy* method is to use *Software Manager*, to search and install software on your desktop. It supports installing software from Linux Mint, Debian and Flathub repositories.

Examples of software that can be installed using this method:

Circos[10] — plotter for visualizing data.

Clustal Omega[11] — general purpose multiple sequence alignment program for nucleotide and protein sequences.

EMBOSS[12] — European molecular biology open software suite.

NCBI BLAST+[13] — next generation suite of BLAST sequence search tools.

Bioconductor[14] — tools for the analysis and comprehension of high-throughput genomic data.

CD-HIT[15] — suite of programs designed to quickly group sequences.

UGENE[16] — integrated bioinformatics toolkit.

—

[10] https://circos.ca
[11] https://www.clustal.org/omega
[12] https://emboss.sourceforge.net
[13] https://blast.ncbi.nlm.nih.gov/Blast.cgi
[14] https://www.bioconductor.org
[15] http://weizhongli-lab.org/cd-hit/
[16] http://ugene.net/

3.1.1 Requirements

Attention: *This procedure installs software in system paths and so requires administrator privileges.*

3.1.2 Finding and installing software

To demonstrate the steps involved, I will install *PyMOL* — a software for molecular visualization. You can follow the same procedure for finding and installing other software available in the repositories.

Open Software Manager by clicking on its icon in the applications' menu (Fig. 39).

You can also launch it from the Administration section of the applications' menu.

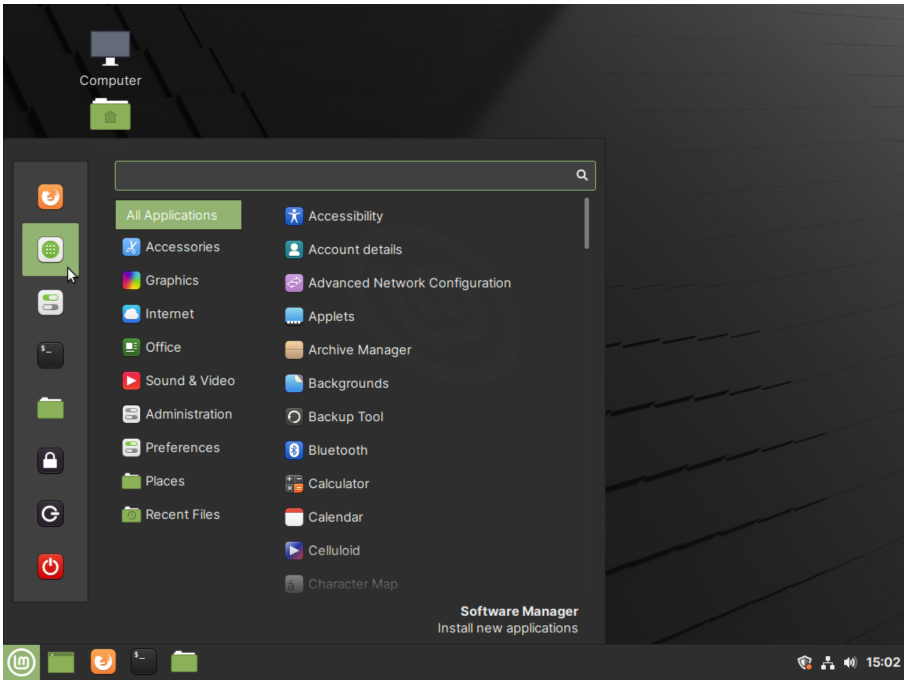


Fig. 39: Click on Software Manager in applications' menu

When launched, you will see its main window (Fig. 40).

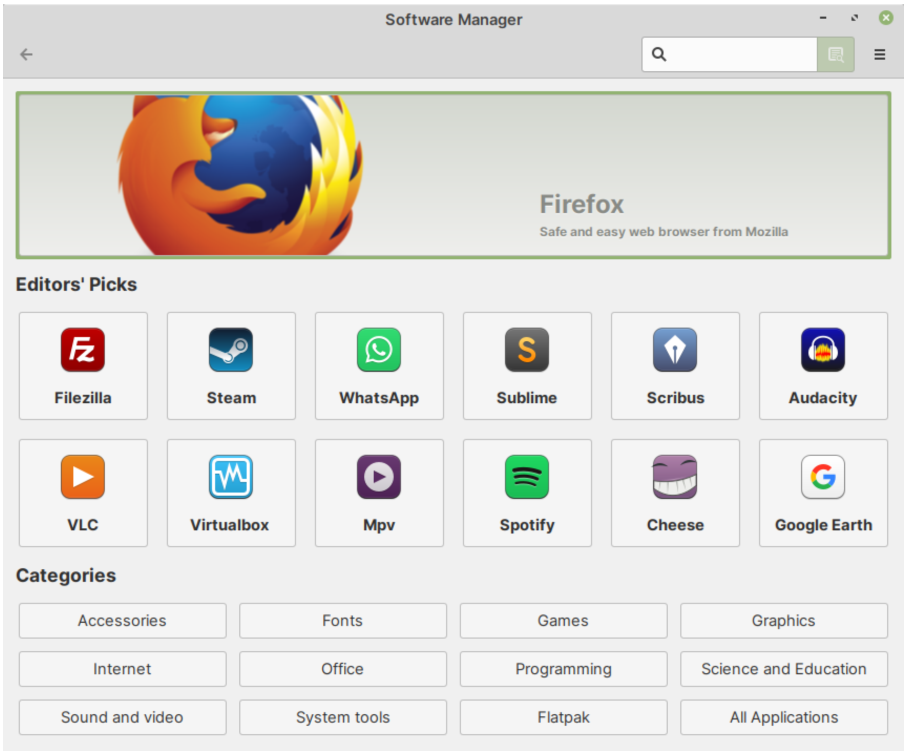


Fig. 40: Main window of Software Manager

In the search bar, type `pymol`.

In some time, packages in repositories that match the search term i.e., pymol, will be displayed below (Fig. 41).

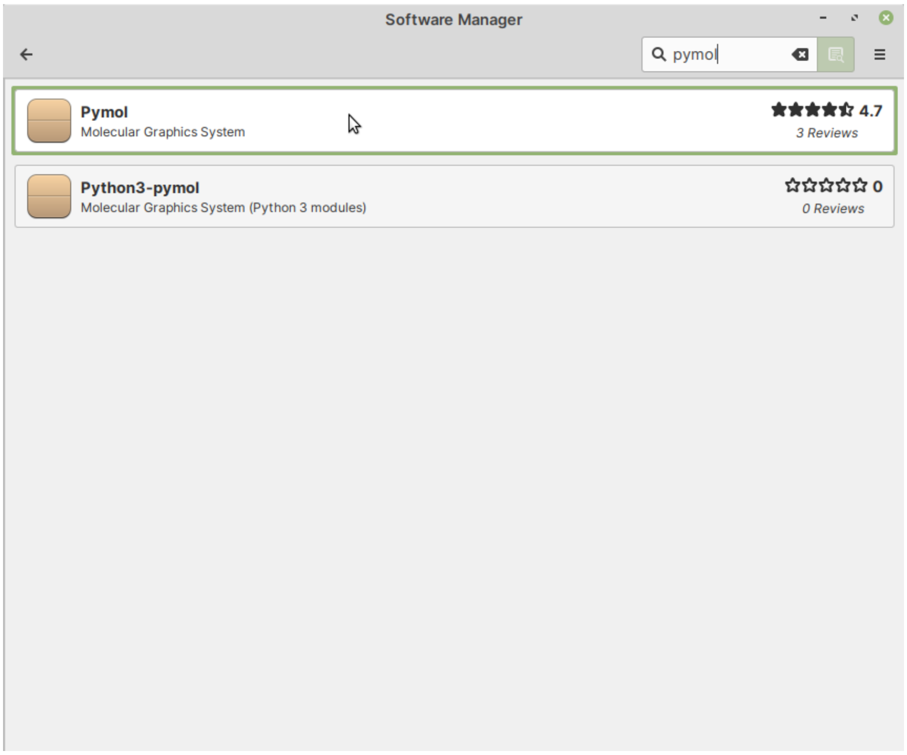

Software Manager
pymol
Pymol
Molecular Graphics System
4.7
3 Reviews
Python3-pymol
Molecular Graphics System (Python 3 modules)
0
0 Reviews

Fig. 41: Search results for pymol

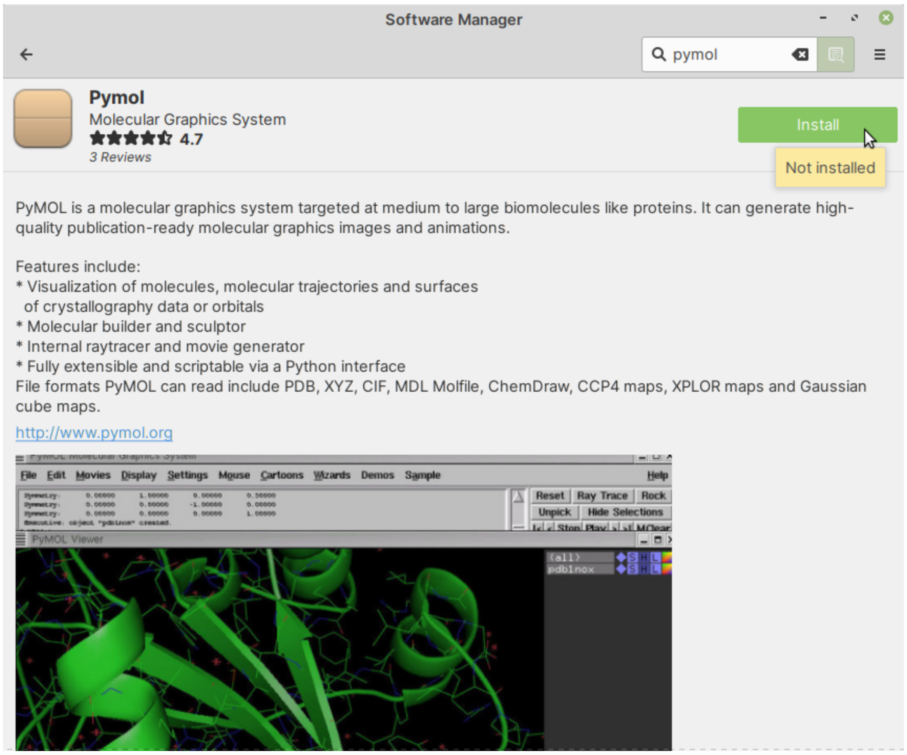
Click on the first result — `Pymol`. You will be taken to the package description page.

Note: Why not select `Python3-pymol`?

The description of the second result, `Python3-pymol` indicates that it contains Python 3 *modules* for working with PyMOL, which is not what we are looking for.

If you are unsure, you can read the complete description of the package on the next page and confirm if this is the package you would like to install.

In the package description page, click on the Install button (Fig. 42).



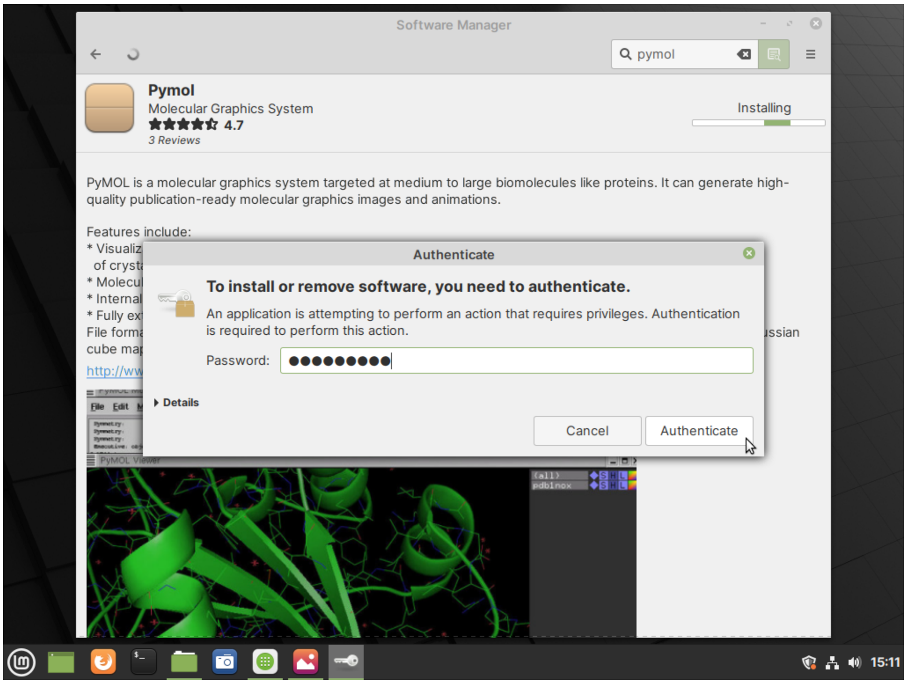
PyMOL is a molecular graphics system targeted at medium to large biomolecules like proteins. It can generate high-quality publication-ready molecular graphics images and animations.

Features include:
* Visualization of molecules, molecular trajectories and surfaces
  of crystallography data or orbitals
* Molecular builder and sculptor
* Internal raytracer and movie generator
* Fully extensible and scriptable via a Python interface
File formats PyMOL can read include PDB, XYZ, CIF, MDL Molfile, ChemDraw, CCP4 maps, XPLOR maps and Gaussian cube maps.
http://www.pymol.org

Fig. 42: Click on the Install button to install this package

An dialog window will now appear, prompting you to enter *your password*.

Type in your password and click on the Authenticate button (Fig. 43).


Software Manager

Pymol
Molecular Graphics System
4.7
3 Reviews

Installing

PyMOL is a molecular graphics system targeted at medium to large biomolecules like proteins. It can generate high-quality publication-ready molecular graphics images and animations.

Features include:

Authenticate

To install or remove software, you need to authenticate.

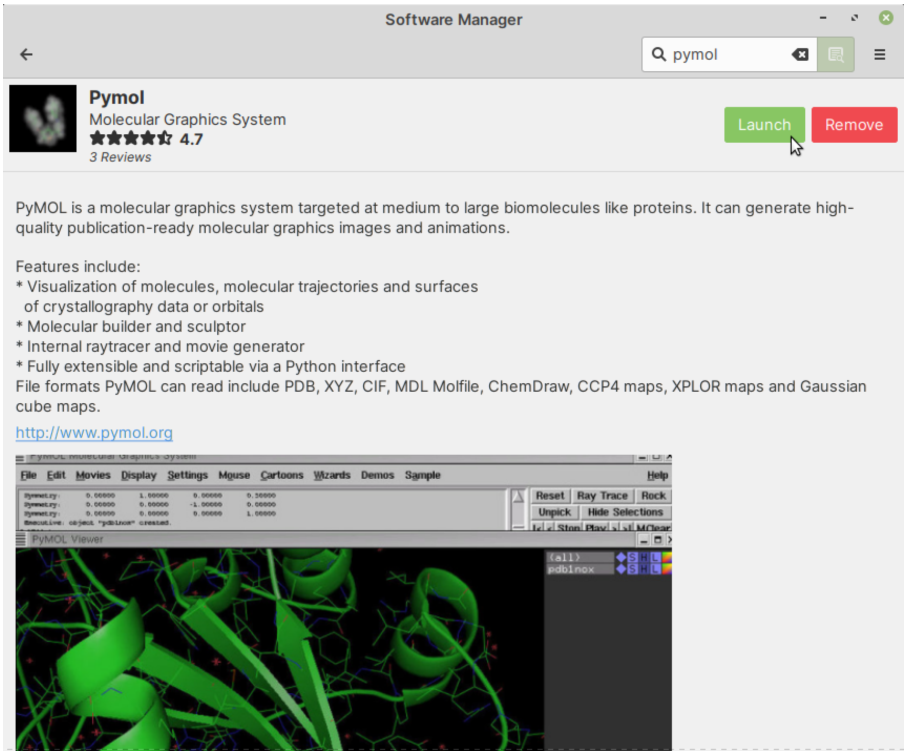
An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password: ●●●●●●●●●

Fig. 43: Type in your password and click on the Authenticate button

Installation will now proceed.

When installation is complete, you will notice two buttons — Launch and Remove, in place of the Install button (Fig. 44).

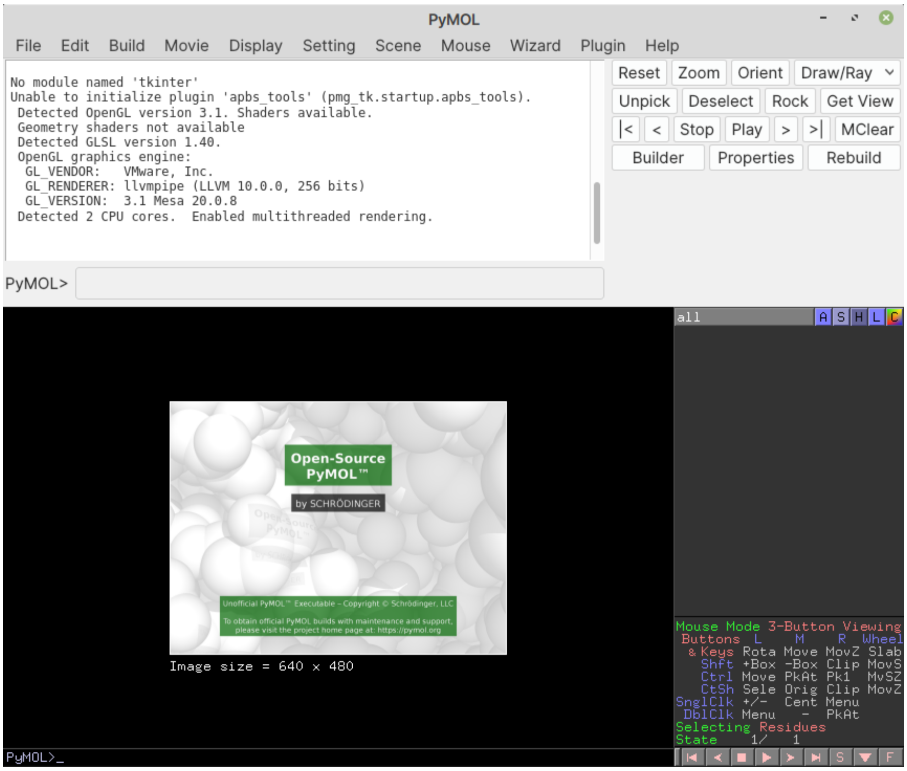


PyMOL is a molecular graphics system targeted at medium to large biomolecules like proteins. It can generate high-quality publication-ready molecular graphics images and animations.

Features include:
* Visualization of molecules, molecular trajectories and surfaces
  of crystallography data or orbitals
* Molecular builder and sculptor
* Internal raytracer and movie generator
* Fully extensible and scriptable via a Python interface
File formats PyMOL can read include PDB, XYZ, CIF, MDL Molfile, ChemDraw, CCP4 maps, XPLOR maps and Gaussian cube maps.
http://www.pymol.org

Fig. 44: Installation complete. Notice the Launch and Remove buttons

Click on the Launch button. This will open the PyMOL program (Fig. 45).


PyMOL
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Reset Zoom Orient Draw/Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Builder Properties Rebuild
PyMOL>
all A S H L C
Open-Source PyMOL™
by SCHRÖDINGER

```
No module named 'tkinter'
Unable to initialize plugin 'apbs_tools' (pmg_tk.startup.apbs_tools).
 Detected OpenGL version 3.1. Shaders available.
 Geometry shaders not available
 Detected GLSL version 1.40.
 OpenGL graphics engine:
  GL_VENDOR:   VMware, Inc.
  GL_RENDERER: llvmpipe (LLVM 10.0.0, 256 bits)
  GL_VERSION:  3.1 Mesa 20.0.8
 Detected 2 CPU cores.  Enabled multithreaded rendering.
```

```
Image size = 640 x 480
```

```
Mouse Mode 3-Button Viewing
 Buttons  L    M    R  Wheel
  & Keys Rota Move MovZ Slab
    Shft +Box -Box Clip MovS
    Ctrl Move PkAt Pk1  MvSZ
    CtSh Sele Orig Clip MovZ
SnglClk +/-  Cent Menu
 DblClk Menu  -   PkAt
Selecting Residues
State    1/   1
```

```
PyMOL>_
```

Fig. 45: Main window of PyMOL

Alternatively, you can search for the application in the Applications Menu and launch it from there.

Note: For applications that do not have a graphical user interface, the Launch button will not be present.

If the program includes any commands, you can access them in a *terminal* session.

3.1.3 Removing installed software

You can follow these steps to remove software installed from repositories.

Open Software Manager.

Click on the settings button in the main application window and then select Show installed applications (Fig. 46).

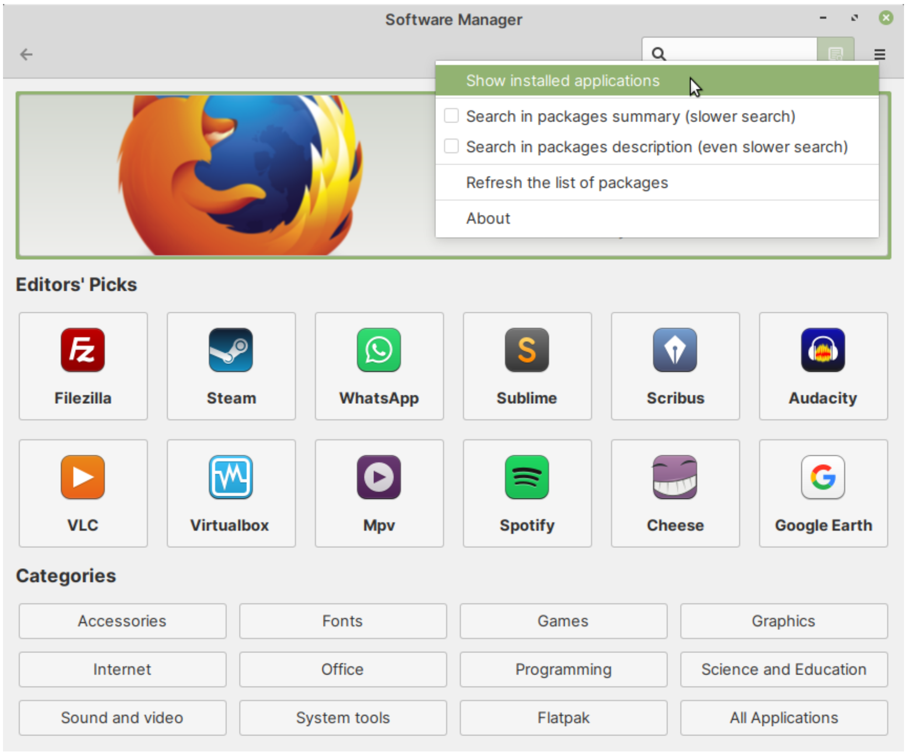
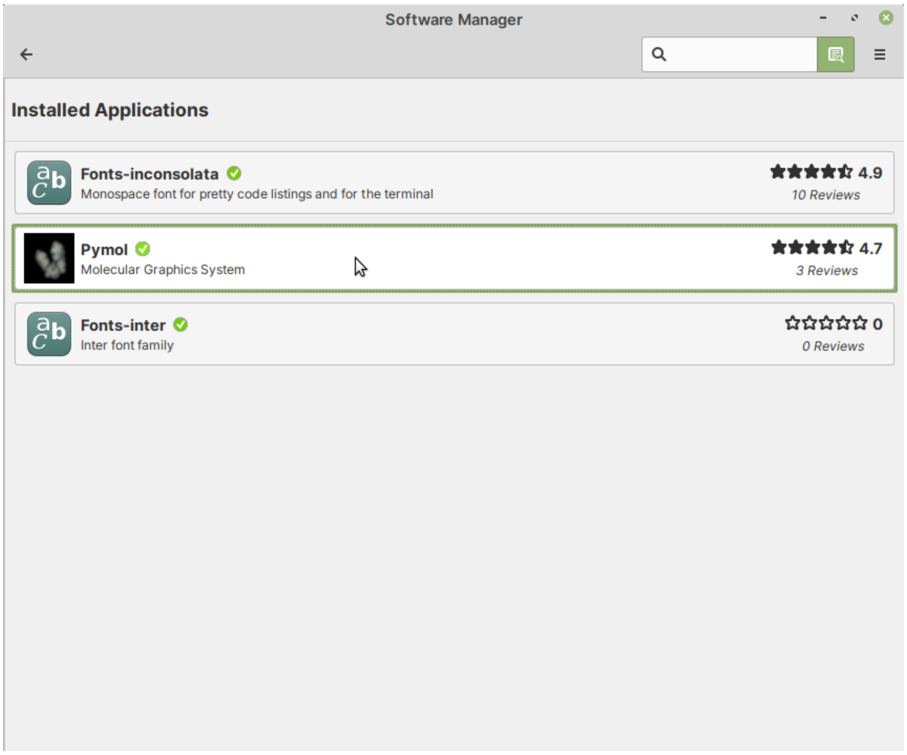


Fig. 46: From Software Manager settings, select Show Installed Applications

Alternatively, you can type the name of the software in the search bar and then press the ENTER key.

In the list of Installed Applications, click on the package you would like to remove — Pymol, in this case (Fig. 47).



Fonts-inconsolata
Monospace font for pretty code listings and for the terminal
4.9
10 Reviews

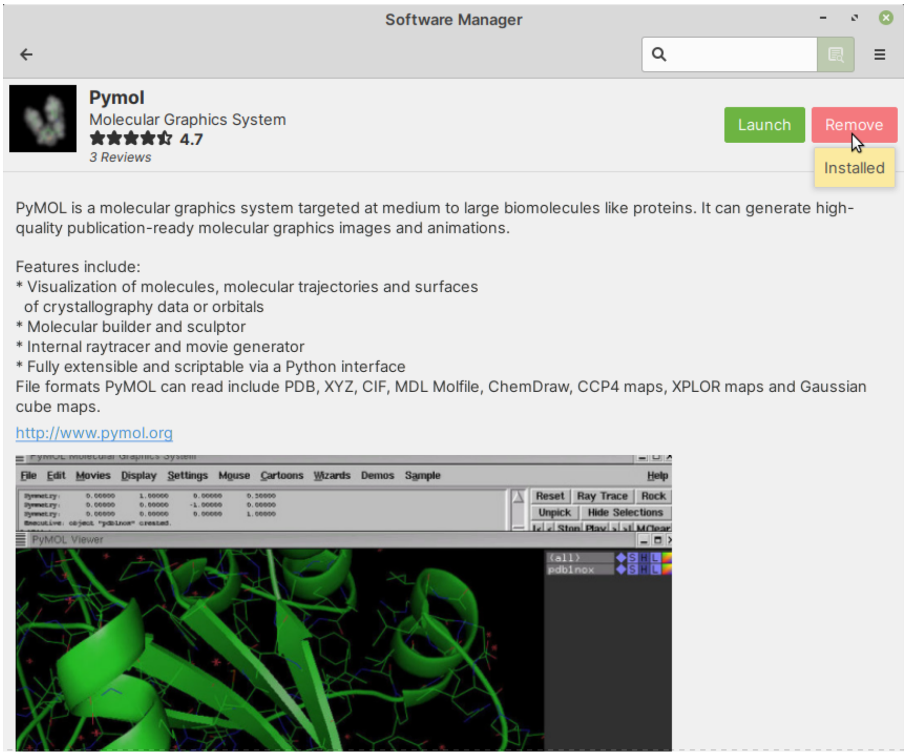
Pymol
Molecular Graphics System
4.7
3 Reviews

Fonts-inter
Inter font family
0
0 Reviews

Fig. 47: Click on the package you would like to remove

You will be taken to the package description page.

Click on the Remove button to remove (or uninstall) the package (Fig. 48).



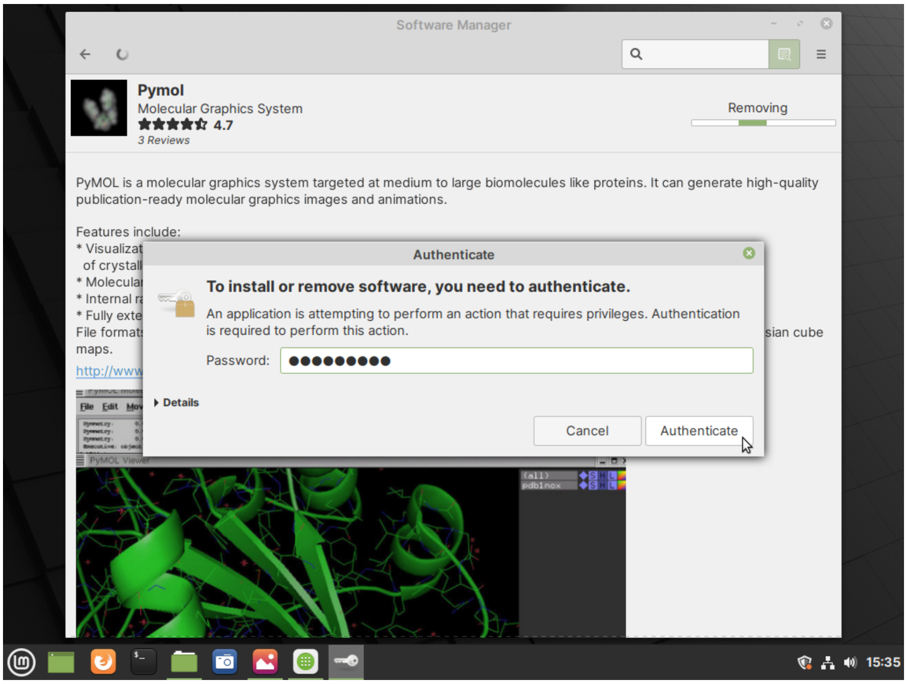
PyMOL is a molecular graphics system targeted at medium to large biomolecules like proteins. It can generate high-quality publication-ready molecular graphics images and animations.

Features include:
* Visualization of molecules, molecular trajectories and surfaces
  of crystallography data or orbitals
* Molecular builder and sculptor
* Internal raytracer and movie generator
* Fully extensible and scriptable via a Python interface
File formats PyMOL can read include PDB, XYZ, CIF, MDL Molfile, ChemDraw, CCP4 maps, XPLOR maps and Gaussian cube maps.
http://www.pymol.org

Fig. 48: Click on the Remove button to remove the package

You will be prompted to enter your password.

Type in *your password* and then click on the Authenticate button to proceed (Fig. 49).



PyMOL is a molecular graphics system targeted at medium to large biomolecules like proteins. It can generate high-quality publication-ready molecular graphics images and animations.

Features include:

To install or remove software, you need to authenticate.

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Fig. 49: Type in your password and click on the Authenticate button

The software will now be removed (or uninstalled) from the system.

3.1.4 Updating installed software

When updates are available for software installed on your system, you will see a notification in the system tray (Fig. 50).

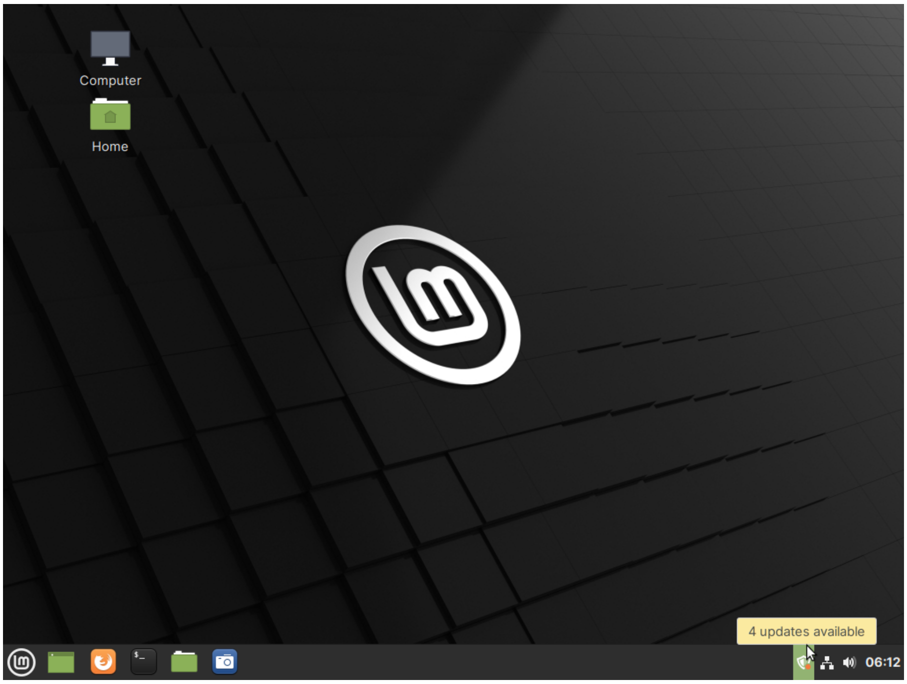
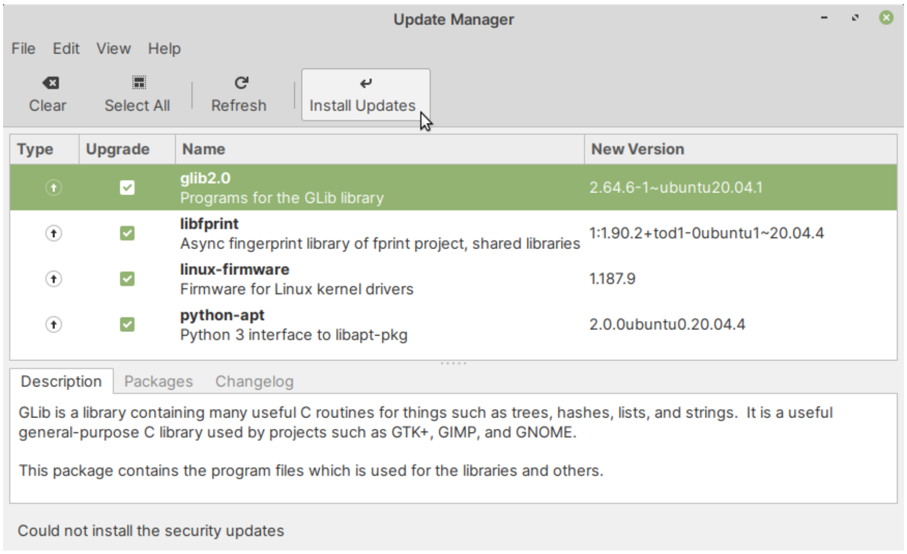


Fig. 50: Notification of software updates in system tray

To apply software updates:

Click on the notification icon in system tray. This will open the Update Manager program.

To install all available software updates, click on the Install Updates button in the toolbar (Fig. 51).



Type	Upgrade	Name	New Version
	✓	glib2.0 Programs for the GLib library	2.64.6-1~ubuntu20.04.1
	✓	libfprint Async fingerprint library of fprint project, shared libraries	1:1.90.2+tod1-0ubuntu1~20.04.4
	✓	linux-firmware Firmware for Linux kernel drivers	1.187.9
	✓	python-apt Python 3 interface to libapt-pkg	2.0.0ubuntu0.20.04.4

GLib is a library containing many useful C routines for things such as trees, hashes, lists, and strings. It is a useful general-purpose C library used by projects such as GTK+, GIMP, and GNOME.

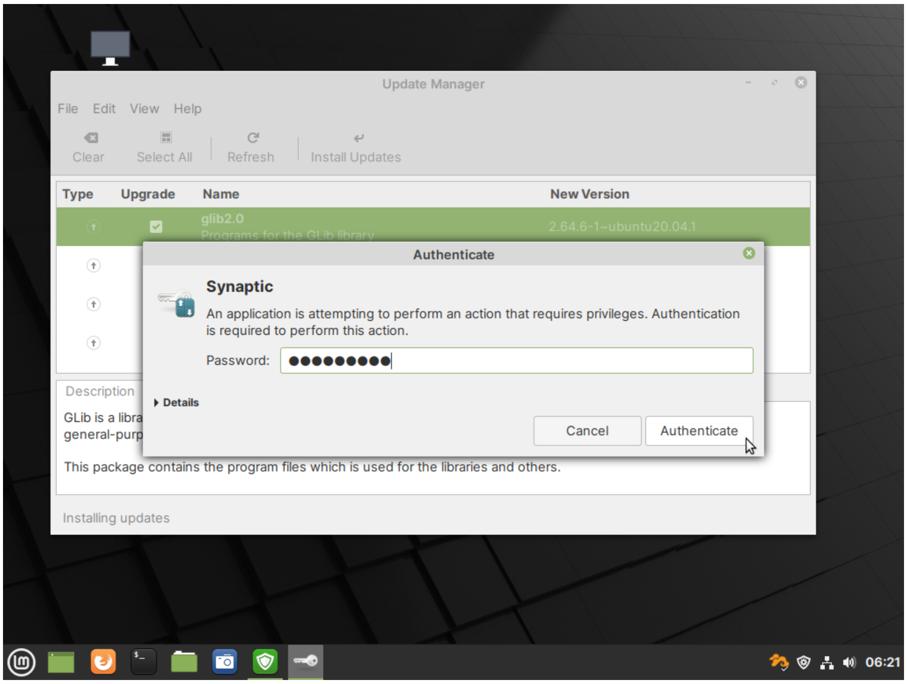
This package contains the program files which is used for the libraries and others.

Could not install the security updates

Fig. 51: Click on the Install Updates button to apply all available updates

You will be prompted to enter your password (Fig. 52).

Type in *your password* and click on the Authenticate button.



An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

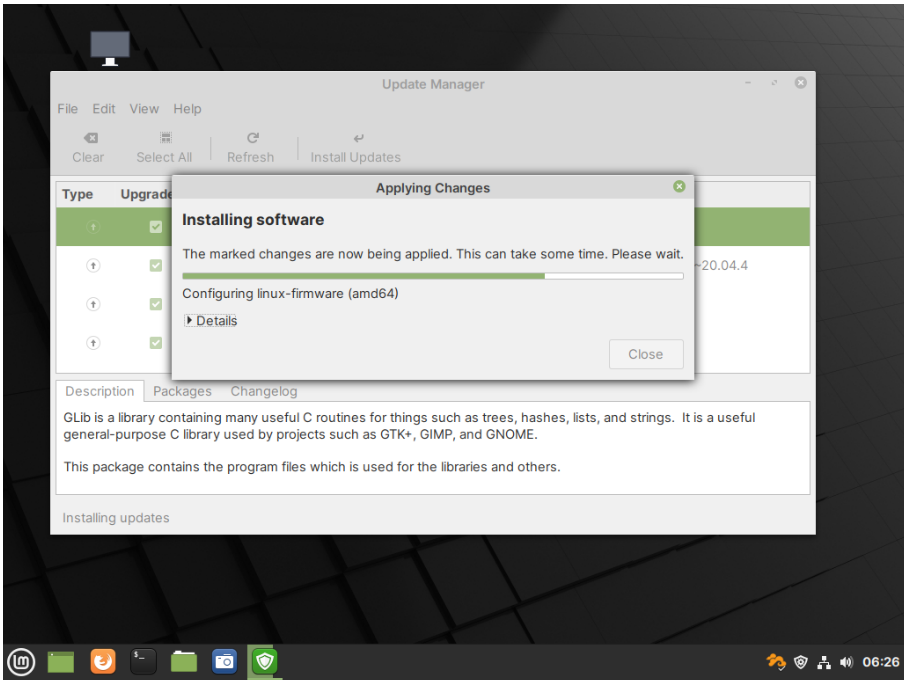
GLib is a libra
general-purp

This package contains the program files which is used for the libraries and others.

Installing updates

Fig. 52: Type in your password and click on the Authenticate button

Update Manager will now continue installing updates (Fig. 53).



Installing software

The marked changes are now being applied. This can take some time. Please wait.

Configuring linux-firmware (amd64)

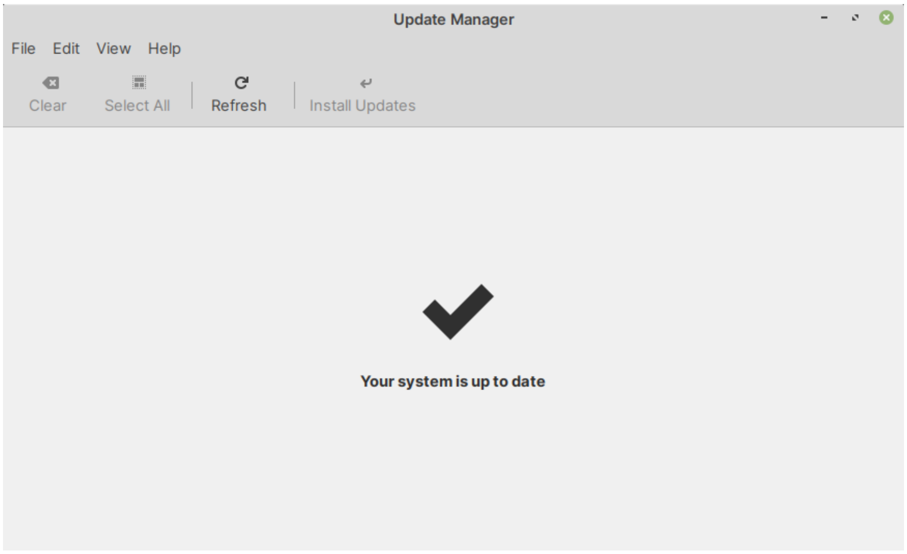
GLib is a library containing many useful C routines for things such as trees, hashes, lists, and strings. It is a useful general-purpose C library used by projects such as GTK+, GIMP, and GNOME.

This package contains the program files which is used for the libraries and others.

Installing updates

Fig. 53: Update Manager installing software updates

When the update process is complete, you will see a message — Your system is up to date (Fig. 54).



Your system is up to date

Fig. 54: Updates installed successfully

3.1.5 Notes

The command-line version (apt)

You can use the *apt* package manager to install software from the command-line.

Searching for software

Open a terminal to perform a search for matching packages in the repositories.

For example, here is a search for `pymol`

```
apt search pymol
```

Note: You do not need to include `sudo` here.

This will output matching results from repositories, if any:

```
p   pymol              - Molecular Graphics System
p   pymol-data         - data files for PyMOL
p   python3-pymol      - Molecular Graphics System (Python 3␣
↪module
```

Once you have identified the correct package name — in this case, it is `pymol` you can proceed towards installing the package.

Installing a package

You can install a package using the following command:

```
sudo apt install pymol
```

Attention: *You need to include sudo here.*

Also, in the commands below, you will be asked to enter *your password*.

You will also need to confirm if you would like to continue with the changes. Type `y` and then press the `ENTER` key to proceed.

```
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  pymol-data python3-pymol
The following NEW packages will be installed:
  pymol pymol-data python3-pymol
0 upgraded, 3 newly installed, 0 to remove and
4 not upgraded.
Need to get 0 B/5,192 kB of archives.
After this operation, 19.8 MB of additional disk
space will be used.
Do you want to continue? [Y/n] y
```

The package will now be installed.

Removing an installed package

To remove an installed package, use the command:

```
sudo apt remove pymol
```

The command above does not remove dependent packages. You can remove them using:

```
sudo apt autoremove
```

Updating packages

To update *all installed packages*, you can use these commands:

```
sudo apt update
sudo apt upgrade
```

To update *a specific package* — if an update is available, you can simply use `apt install` again:

```
sudo apt install package_name
```

Why is this a quick and easy method?

You can use a graphical user interface (Software Manager) or the command-line (`apt`) to install software available in Linux package repositories.

Software installed in this manner will also be kept updated along with the rest of the system.

Additional software repositories

It takes some time for newer versions of software to become available in distribution repositories.

If your software of interest is not available or if you need a more recent version of the software, you could try installing them from these additional sources:

- *Python packages* from *PyPI*
- *Perl modules* from *CPAN*
- *R packages* from *CRAN* or *Bioconductor*
- *Conda packages* from *Anaconda Cloud*

Attention: Software installed using these methods, will also need to be kept updated manually.

3.2 Python packages

If your software of interest is available as a Python package on *PyPI*, you can install it using *pip* — Python package installer.

Note: Your software might already be available using *The quick and easy method*. Search for package names starting with `python-`.

In certain cases, installing packages in a virtual environment might be a better option. Read:

`->` *When should I use a virtual environment?*

3.2.1 Requirements

Python package installer (pip)

To install pip, follow *The quick and easy method* for installing software.

Search and install the `python3-pip` package.

$HOME/.local/bin added to $PATH

Packages that include commands will install them in the `$HOME/.local/bin` directory (*What is $HOME?*).

To be able to run these commands easily, you will need to add this directory to your `$PATH` variable. You can do so by following the steps in *Adding directories to PATH*.

Attention: *You should not add sudo in the commands below.*

This method only installs files in your home directory and so does not require administrator privileges.

3.2.2 Searching for a package on PyPI

Open PyPI website[17] in a web browser.

In the Search projects field, enter the name of the software you would like to install and press the ENTER key or click on the search button (Fig. 55).

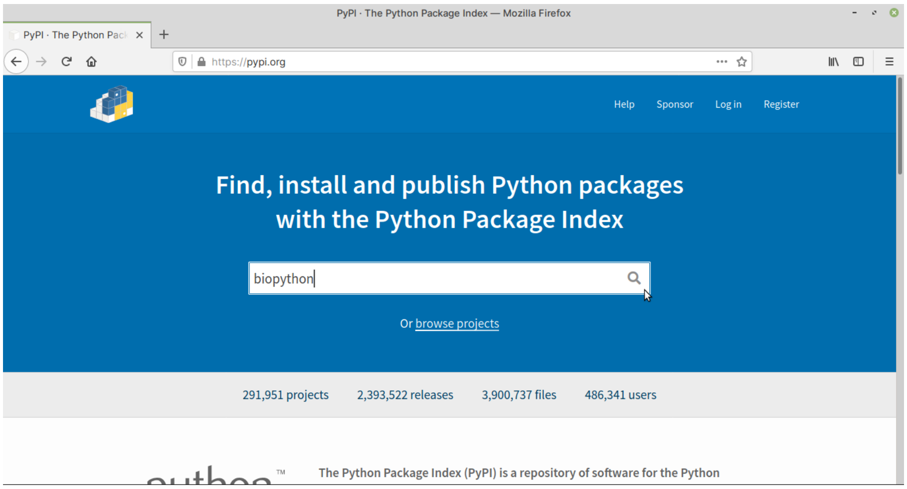
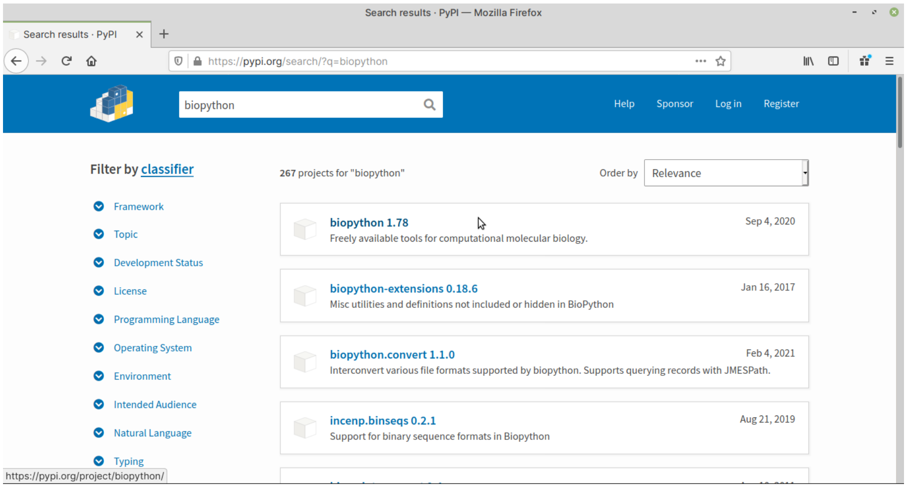

PyPI · The Python Package Index — Mozilla Firefox
PyPI · The Python Pac
https://pypi.org
Help
Sponsor
Log in
Register
Find, install and publish Python packages
with the Python Package Index
biopython
Or browse projects
291,951 projects
2,393,522 releases
3,900,737 files
486,341 users
The Python Package Index (PyPI) is a repository of software for the Python

Fig. 55: Searching for a package on PyPI

A list of packages matching the search term will be displayed (Fig. 56).

[17] https://pypi.org



Search results · PyPI — Mozilla Firefox

Search results · PyPI

https://pypi.org/search/?q=biopython

biopython

Help Sponsor Log in Register

Filter by classifier

- Framework
- Topic
- Development Status
- License
- Programming Language
- Operating System
- Environment
- Intended Audience
- Natural Language
- Typing

267 projects for "biopython"

Order by Relevance

biopython 1.78 Sep 4, 2020
Freely available tools for computational molecular biology.

biopython-extensions 0.18.6 Jan 16, 2017
Misc utilities and definitions not included or hidden in BioPython

biopython.convert 1.1.0 Feb 4, 2021
Interconvert various file formats supported by biopython. Supports querying records with JMESPath.

incenp.binseqs 0.2.1 Aug 21, 2019
Support for binary sequence formats in Biopython

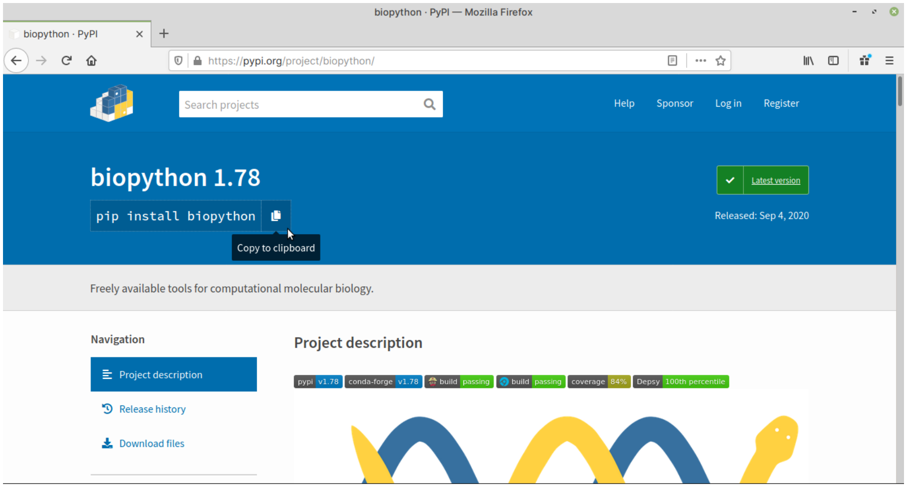
https://pypi.org/project/biopython/

Fig. 56: Search results for biopython. 1.78 is the version number.

Click on the result to proceed to the project description page.

In the project description page, find the package name (Fig. 57) in the `pip install` command.

In this case, it is `biopython`.



biopython · PyPI — Mozilla Firefox

biopython · PyPI

https://pypi.org/project/biopython/

Search projects

Help Sponsor Log in Register

biopython 1.78

```
pip install biopython
```

Copy to clipboard

Latest version

Released: Sep 4, 2020

Freely available tools for computational molecular biology.

Navigation

- Project description
- Release history
- Download files

Project description

pypi v1.78 conda-forge v1.78 build passing build passing coverage 84% Depsy 100th percentile

Fig. 57: Project description page for Biopython

Note: You will need to use `pip3` instead of the `pip` command in the steps below. *Why?*

You can now proceed towards installing the package.

3.2.3 Installing a Python package

Open a terminal.

Use the `pip3 install` command with by the name of the package, you would like to install.

Using biopython as an example:

```
pip3 install biopython
```

Output:

```
 Collecting biopython
  Downloading biopython-1.78-cp38-cp38-manylinux1_x86_64.
↪whl (2.3 MB)
     |********************************| 2.3 MB 2.1 MB/s
Collecting numpy
  Downloading numpy-1.20.1-cp38-cp38-manylinux2010_x86_64.
↪whl (15.4 MB)
     |********************************| 15.4 MB 90 kB/s
Installing collected packages: numpy, biopython
Successfully installed biopython-1.78 numpy-1.20.1
```

Note: One limitation of this method is that, you will not be able to install *multiple versions* of a package. This can be solved using a *virtual environment*.

3.2.4 Updating an installed package

When you notice an update is available for the package, you can use the `install` command with the `-U` option, and the name of the package:

```
pip3 install -U biopython
```

3.2.5 Removing an installed package

Use the `uninstall` command with the name of the package you would like to remove:

```
pip3 uninstall biopython
```

3.2.6 Using installed packages

Where are the files installed?

You can use the `pip3 show` command with the name of the package to identify the path where it is installed:

```
pip3 show biopython
```

Output:

```
Name: biopython
Version: 1.78
Summary: Freely available tools for computational
↪molecular biology.
Home-page: https://biopython.org/
Author: The Biopython Contributors
Author-email: biopython@biopython.org
License: UNKNOWN
Location: /home/user/.local/lib/python3.8/site-packages
Requires: numpy
Required-by:
```

The installation path will be listed next to the `Location` keyword.

Using Python packages and modules

Python packages and modules will be installed in `$HOME/.local/lib/python3.8/site-packages`.

Note: This is the path for Python 3.8.

If you have a different version of Python installed, this value will change.

This directory will be automatically included in `$PYTHONPATH`. So, you can import and use installed packages and modules in your scripts, without any extra effort.

For example, here is a simple Python script to test BioPython installed using pip:

```python
from Bio.Seq import Seq

seq = Seq('ATGC')
comp = seq.complement()

print(f'The complement of {seq} is {comp}')
```

The `Bio.Seq` package is part of BioPython. Copy the code sample above and save it as `biopy_test.py`. Then run it from the terminal like so:

```
python3 biopy_test.py
```

Output:

```
The complement of ATGC is TACG
```

Using included commands

If the package includes commands, those will be installed in `$HOME/.local/bin` directory. For these commands to be easily accessible, you will need to add this directory to `PATH`, as mentioned under *requirements*.

As an example, when you install the Python package of *cutadapt*, the `cutadapt` command will be installed in `$HOME/.local/bin`, which you can then run from a terminal like this:

```
cutadapt --version
```

Output:

```
3.1
```

3.2.7 Python virtual environments

A virtual environment is a *self-contained* directory tree containing Python and some additional packages.

Advantages

These are *some* advantages of using virtual environments.

No administrator privileges

You do not need administrator privileges to create a virtual environment or install packages in an environment.

Multiple environments

Multiple virtual environments can be created with each containing their own sets of packages.

These are isolated and packages in an environment can be installed, updated or removed without affecting other environments.

Share your environment

You can share your configuration with others. They will be able to reproduce your environment, with the exact versions of packages.

Creating a virtual environment

First, using *The quick and easy method*, search and install the `python3-venv` package. This includes the `venv` module necessary for creating virtual environments.

You can create a virtual environment in any directory where you have write privileges, for example, your home directory.

To demonstrate, I will create a virtual environment called `py3env` in my home directory.

```
python3 -m venv py3env
```

If successful, you will find a directory named `py3env` in the current directory. No messages will be displayed.

Note: `python3` is the command to run the Python 3 interpreter. Its complete path is `/usr/bin/python3`.

`venv` is the Python module to create virtual environments.

The `-m` option runs the `venv` module as a script.

Before you can use a virtual environment, you will need to activate it.

Activating a virtual environment

You will need to activate a virtual environment before you can start using it. To do so, use the `source` command with the path to the virtual environment's `activate` script.

For example, to activate `py3env` created in the *previous step*, use:

```
source py3env/bin/activate
```

Your shell prompt will now change to indicate that the virtual environment is now active. Note the `(py3env)` label at the beginning of the prompt:

```
(py3env) user@cookbook:~$
```

You can now start using this virtual environment.

Note: *Before you start installing packages…*

It is a good idea to install (or upgrade) Python build tools — *pip*, *setuptools*, and *wheel* in a new virtual environment.

These build tools are necessary for building and installing packages from *PyPI* and other sources.

Installing them will ensure that additional packages will build and install without errors.

Installing Python build tools

Use `pip3 install` to install or upgrade the required packages:

```
pip3 install -U pip setuptools wheel
```

The `-U` option of `pip3 install`, will upgrade listed packages, if newer versions are available.

Output:

```
Collecting pip
Downloading pip-21.0.1-py3-none-any.whl (1.5 MB)
...
Installing collected packages: pip, setuptools, wheel
...
Successfully installed pip-21.0.1 setuptools-54.2.0 wheel-
↪0.36.2
```

Deactivating a virtual environment

To exit a virtual environment, use the command:

```
deactivate
```

Your shell prompt will change to its original appearance:

```
user@cookbook:~$
```

3.2.8 Notes

You will need to update these packages manually

Packages installed in this manner should also be updated manually.

When you notice there is an update for the package, for example, from the project's website or from their source code repository, follow the steps in *Updating an installed package* to install the latest version.

Why pip3 and not pip?

The *pip* package includes the following commands:

- `pip3` — Python 3 version
- `pip` — Python 2 version

Support for Python 2 ended in January 2020.

Since there is a possibility for both commands to exist on a system, it is safer to use `pip3` when installing packages using this method.

What about programs written in Python 2?

Support for Python 2 ended in January 2020.

If you do need to use a program written only in Python 2, you can create an isolated environment — either using Python venv or Conda and then install the package there.

Related sections:

- *Python virtual environments*
- *Conda packages*

When should I use a virtual environment?

The method described here will not work if the programs you are installing require two different versions of the same package from PyPI.

In that case, you can consider creating an isolated environment — either using Python virtualenv or Conda and then installing the packages there.

Related sections:

- *Python virtual environments*
- *Conda packages*

Older versions of pip

If the version of pip installed in your system is older than 20.0, it will attempt to install packages in system paths by default, resulting in *permission denied* errors.

To avoid that, you will need to add the `--user` option to the `install` and `uninstall` commands, for example:

```
pip3 install --user biopython
```

A better approach is to *upgrade* your local version of pip. Once upgraded, you will no longer need to user `--user`.

To upgrade pip, do:

```
pip3 install --user -U pip
```

You can check the installed version of pip using:

```
pip3 -V
```

3.3 Perl modules

For installing Perl modules, you can configure the *local-lib* module to create and use a local directory. The default location is `$HOME/perl5`.

If your software of interest is available on *CPAN*, you can install it easily using the *cpanminus* script.

Note: Your software of interest, might be available using *The quick and easy method*.

Search for package names containing `perl`.

3.3.1 Requirements

The build-essential and cpanminus packages installed

If you do not have them installed, follow *The quick and easy method*, and search and install the following packages:

1. **`build-essential`** — installs compilers and build tools like `make`
2. **`cpanminus`** — script to obtain, build and install modules from CPAN along with their *dependencies*

Attention: *You should not add sudo in the commands below.*

This method only installs files in your home directory and so does not require administrator privileges.

3.3.2 Configuring local-lib module

To configure local-lib:

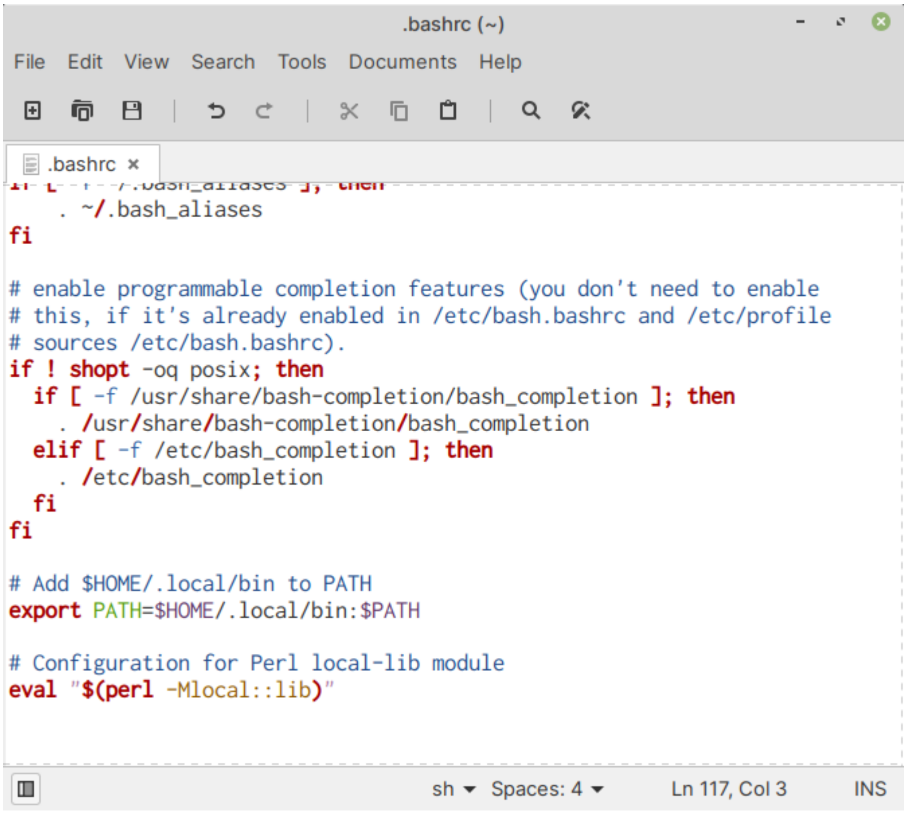
Open the `$HOME/.bashrc` file in a *text editor*.

Add the following text at the end of the file:

```
# Configuration for Perl local-lib module
eval "$(perl -Mlocal::lib)"
```

Save the file.

Here is a sample `.bashrc`, for reference (Fig. 58).

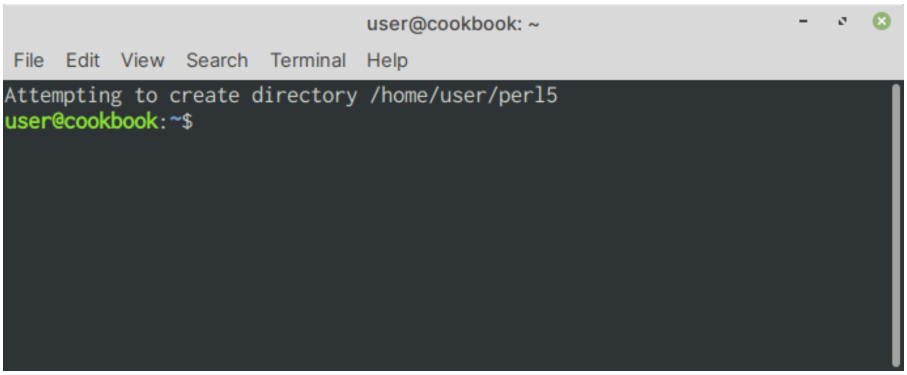

.bashrc (~)
File Edit View Search Tools Documents Help
.bashrc ×
sh Spaces: 4 Ln 117, Col 3 INS

```
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# Add $HOME/.local/bin to PATH
export PATH=$HOME/.local/bin:$PATH

# Configuration for Perl local-lib module
eval "$(perl -Mlocal::lib)"
```

Fig. 58: local-lib configuration in .bashrc

Open a new terminal window.

You will notice the following message displayed (Fig. 59):



```
Attempting to create directory /home/user/perl5
user@cookbook:~$
```

Fig. 59: Attempting to create directory /home/user/perl5

Configuration is now complete.

You can now start installing modules from CPAN and other sources.

3.3.3 Searching for a module on CPAN

Open *MetaCPAN* website (https://metacpan.org) in a web browser.

In the Search field, enter the name of the module you would like to install and click on the Search the CPAN button.

I will use David H. Ardell's *FAST* package as an example. (Fig. 60).

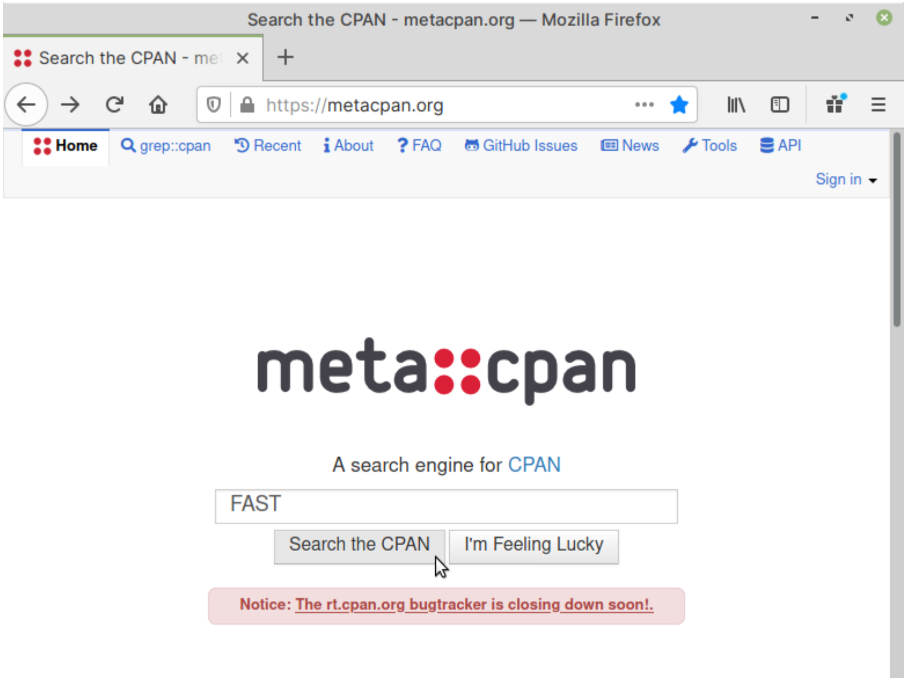
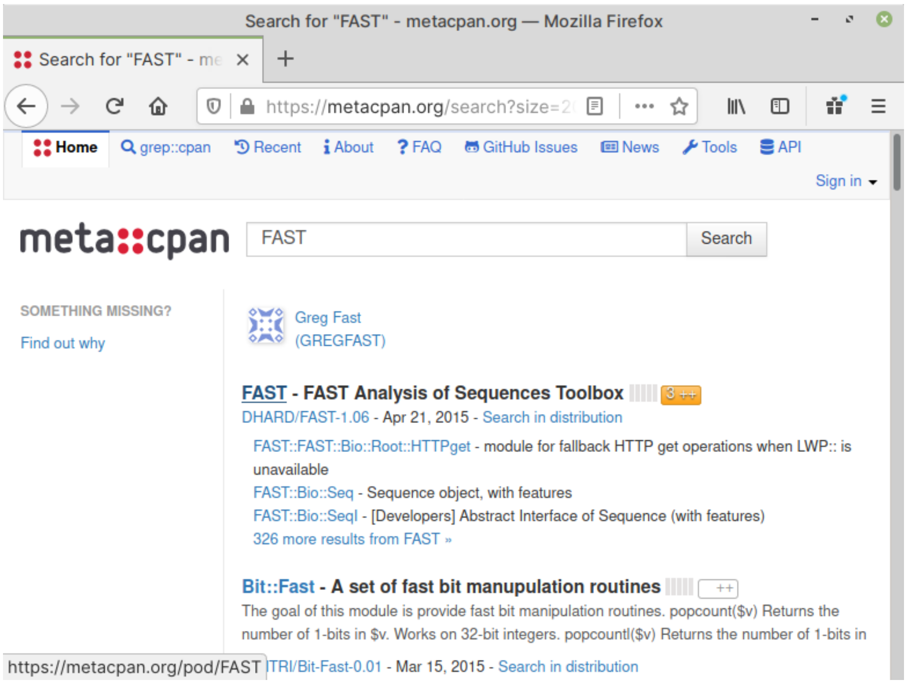


Fig. 60: Searching for a module on MetaCPAN

This should display modules matching the search term — FAST (Fig. 61).


Search for "FAST" - metacpan.org — Mozilla Firefox

https://metacpan.org/search?size=2

Home grep::cpan Recent About FAQ GitHub Issues News Tools API

Sign in

meta::cpan FAST Search

SOMETHING MISSING?
Find out why

Greg Fast
(GREGFAST)

FAST - FAST Analysis of Sequences Toolbox 3 ++
DHARD/FAST-1.06 - Apr 21, 2015 - Search in distribution

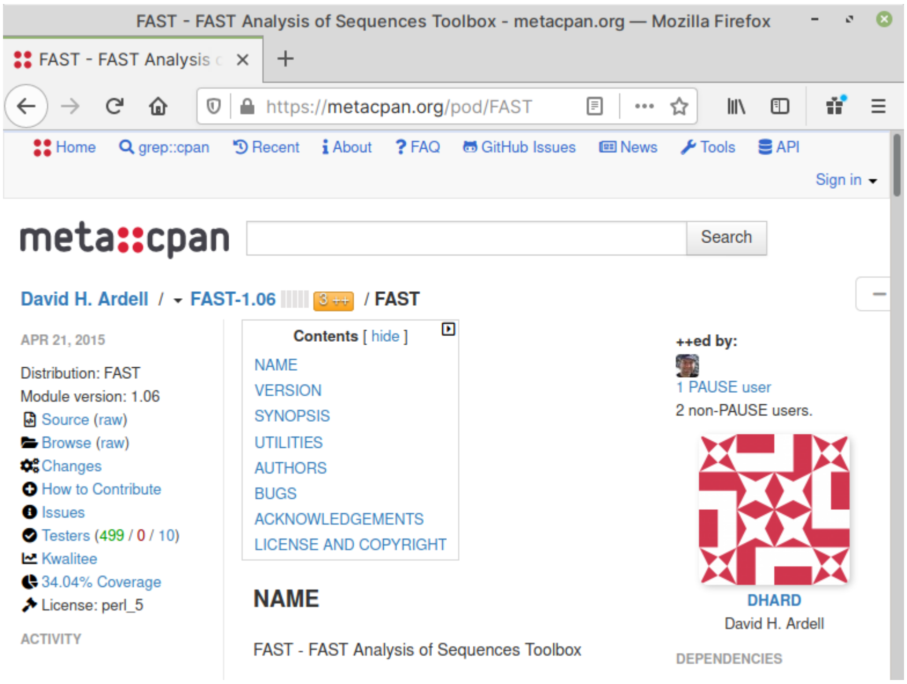
FAST::FAST::Bio::Root::HTTPget - module for fallback HTTP get operations when LWP:: is unavailable
FAST::Bio::Seq - Sequence object, with features
FAST::Bio::SeqI - [Developers] Abstract Interface of Sequence (with features)
326 more results from FAST »

Bit::Fast - A set of fast bit manupulation routines ++
The goal of this module is provide fast bit manipulation routines. popcount($v) Returns the number of 1-bits in $v. Works on 32-bit integers. popcountl($v) Returns the number of 1-bits in

https://metacpan.org/pod/FAST TRI/Bit-Fast-0.01 - Mar 15, 2015 - Search in distribution

Fig. 61: Search results for modules

If you click on the *appropriate* module name, you will be taken to a description page, where you can learn more about the module (Fig. 62).



David H. Ardell / FAST-1.06 3 ++ / FAST

APR 21, 2015

Distribution: FAST
Module version: 1.06
Source (raw)
Browse (raw)
Changes
How to Contribute
Issues
Testers (499 / 0 / 10)
Kwalitee
34.04% Coverage
License: perl_5

++ed by:
1 PAUSE user
2 non-PAUSE users.

DHARD
David H. Ardell

NAME

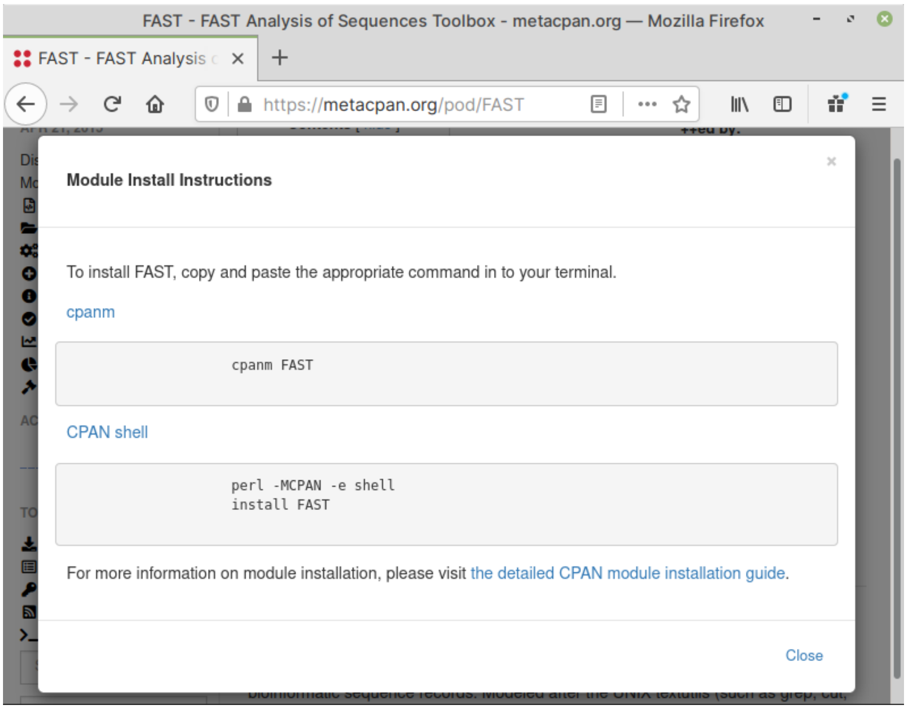
FAST - FAST Analysis of Sequences Toolbox

Fig. 62: Module description page

On this page, you can also verify the module name to use in the next step.

To do that, click on Install Instructions under TOOLS menu in the sidebar. This will open a pop-up window with information on the module name you need to use with the `cpanm` command.

In this case, the module name to use, is simply called `FAST` (Fig. 63).


FAST - FAST Analysis of Sequences Toolbox - metacpan.org — Mozilla Firefox
https://metacpan.org/pod/FAST
Close

Module Install Instructions

To install FAST, copy and paste the appropriate command in to your terminal.

cpanm

```
cpanm FAST
```

CPAN shell

```
perl -MCPAN -e shell
install FAST
```

For more information on module installation, please visit the detailed CPAN module installation guide.

Fig. 63: Module install instructions for FAST

You can now proceed towards installing this module.

3.3.4 Installing a Perl module

Open a terminal window.

Install the module with the `cpanm` command:

```
cpanm FAST
```

This will install the module and its dependencies, if there are any.

Output:

```
--> Working on FAST
Fetching http://www.cpan.org/authors/id/D/DH/DHARD/FAST-1.
↪06.tar.gz ... OK
Configuring FAST-1.06 ... OK
==> Found dependencies: Bit::Vector, Sort::MergeSort,
↪Sort::Key
--> Working on Bit::Vector
Fetching http://www.cpan.org/authors/id/S/ST/STBEY/Bit-
↪Vector-7.4.tar.gz ... OK
Configuring Bit-Vector-7.4 ... OK
...
Building and testing FAST-1.06 ... OK
Successfully installed FAST-1.06
6 distributions installed
```

3.3.5 Using installed modules

Using included commands

If the module includes commands, those will be installed in `$HOME/perl5/bin`. This directory will be included in `$PATH`, when you *configured* local-lib. Hence, you can start using the commands included in a module immediately after installation.

For example, here is the help page of the `fasuniq` command included with the FAST package installed in the previous step:

```
fasuniq -h
Usage:
    fasuniq [options] [MULTIFASTA-FILE]

    [MULTIFASTA-DATA-ON-STDIN] | fasuniq [options]
...
```

Using modules in your scripts

Modules will be installed in:

`$HOME/perl5/lib/perl5/5.30.0`.

Note: This is the module path for Perl 5.30.0.

It you have a different version of Perl installed on your system, it will change.

As local-lib makes these modules discoverable automatically, you can start using them in your scripts immediately.

Here is an example using the *Bio::Phylo* package. After installing the module with `cpanm Bio::Phylo`, create a file using a text editor and copy the following content:

```perl
use Bio::Phylo;

# print version
print Bio::Phylo->VERSION;

# print citation
print Bio::Phylo->CITATION;
```

Save the file as `biophylo_test.pl` and then run it:

```
perl biophylo_test.pl
```

Output:

```
v2.0.1Rutger A Vos, Jason Caravas, Klaas Hartmann, Mark A
↪Jensen and Chase Miller, 2011.
Bio::Phylo - phyloinformatic analysis using Perl. BMC
↪Bioinformatics 12:63.
doi:10.1186/1471-2105-12-63
```

3.4 R packages

If your software of interest is available as an R package on *CRAN* or *Bioconductor*, you can install it using the `install.packages()` or `BiocManager::install()` commands respectively.

Note: Your software might already be available using *The quick and easy method*.

Search for package names starting with `r-cran-` or `r-bioc-`.

3.4.1 Requirements

The r-base package installed

The `r-base` package provides the *GNU R* statistical computation and graphics system.

If you do not have it installed, follow *The quick and easy method*, and search and install the `r-base` package.

Attention: *You should not add sudo in the commands below.*

This method only installs files in your home directory and so does not require administrator privileges.

3.4.2 Starting an R session

To start an R session, open a terminal window.

Type the `R` command and press the `ENTER` key.

```
user@cookbook:~$ R
```

You will be placed inside an R session. Some additional information will be displayed, including the version of R installed.

You can type R commands at the > prompt.

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical
↪Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain
↪conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English
↪locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in
↪publications.
```

(continues on next page)

(continued from previous page)

```
Type 'demo()' for some demos, 'help()' for on-line help,␣
↪or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

3.4.3 Installing a package from CRAN

You can search for R packages on CRAN by consulting the Task Views[18] for a categorized listing or the Table of available packages[19] (currently 17157 packages). Once you have identified a package to install, you can follow the steps below to install it.

I will install the *BiocManager* package as an example below.

1. Install package using `install.packages()` command

Start an R session.

At the prompt, type the `install.packages()` command with the name of the package in quotes. For example:

```
install.packages("BiocManager")
```

Press the ENTER key to proceed.

2. Confirm use of personal library

You will notice a prompt requesting you to confirm if you would like to use a personal library:

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

(continues on next page)

[18] https://cran.r-project.org/web/views/
[19] https://cran.r-project.org/web/packages/available_packages_by_name.html

(continued from previous page)

```
Warning in install.packages("BiocManager") :
'lib = "/usr/local/lib/R/site-library"' is not writable
Would you like to use a personal library instead? (yes/No/
↪cancel) yes
```

Type yes to confirm and press the ENTER key. *Why use a personal library?*.

3. Create personal library if necessary

If you have never installed a package, the personal library directory will not exist. If so, you will be prompted for confirmation to create it:

```
Would you like to create a personal library
'~/R/x86_64-pc-linux-gnu-library/3.6'
to install packages into? (yes/No/cancel) yes
```

Type yes to confirm and press the ENTER key.

Note: The next time you install a package, you will not see this prompt.

4. Package installation summary

Installation should now proceed:

```
trying URL 'https://cloud.r-project.org/src/contrib/
↪BiocManager_1.30.10.tar.gz'
Content type 'application/x-gzip' length 40205 bytes (39␣
↪KB)
==================================================
downloaded 39 KB

* installing *source* package ‘BiocManager’ ...
** package ‘BiocManager’ successfully unpacked and MD5␣
↪sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from␣
↪temporary location
** testing if installed package can be loaded from final␣
↪location
** testing if installed package keeps a record of␣
↪temporary installation path
* DONE (BiocManager)
```

(continues on next page)

(continued from previous page)

```
The downloaded source packages are in
   '/tmp/RtmpDgmOZ8/downloaded_packages'
>
```

If there are no errors during the process, the package installation is successful.

3.4.4 Installing a package from Bioconductor

You can search the Bioconductor software package list[20] on the website (currently has 1974 packages). Once you have identified a package to install, you can follow the steps below to install it.

I will install the *edgeR* package as an example.

Install package using `BiocManager::install()` command

1. *Start an R session*.
2. *Install the BiocManager package*. This is necessary for installing R packages from Bioconductor repository.

   Note: You only need to do this once.

   The next time you wish to install a package from Bioconductor, you can proceed to step 3.

3. At the R prompt, use the `BiocManager::install()` command with the name of the package you would like to install in quotes.

   For example:

   ```
   BiocManager::install("edgeR")
   ```

   Press the `ENTER` key to proceed.

[20] https://www.bioconductor.org/packages/release/BiocViews.html#___Software

The package and its dependencies will now be downloaded, compiled and installed.

Output:

```
Bioconductor version 3.10 (BiocManager 1.30.10), R 3.
↪6.3 (2020-02-29)
Installing package(s) 'BiocVersion', 'edgeR'
also installing the dependencies ‘limma’, ‘locfit’,
↪‘Rcpp’
...
** testing if installed package can be loaded from
↪temporary location
** checking absolute paths in shared objects and
↪dynamic libraries
** testing if installed package can be loaded from
↪final location
** testing if installed package keeps a record of
↪temporary installation path
* DONE (edgeR)
```

If there are no errors during the process, the package installation is successful.

3.4.5 Updating an R package

When an update is available for a package, use the following commands to update.

For packages installed from CRAN:

```
install.packages("BiocManager")
```

For packages installed from Bioconductor:

```
BiocManager::install("edgeR")
```

3.4.6 Removing an R package

To remove an installed package, use the `remove.packages()` command with the name of the package in quotes.

For example:

```r
remove.packages("BiocManager")
```

3.4.7 Notes

The r-base-dev and build-essential packages

If you are installing additional packages from CRAN and other sources, you will also need to install the `r-base-dev` and `build-essential` packages. However, when you install `r-base`, these packages will also be installed as dependencies so, you do not have to install them manually.

Why use a personal library?

When installing packages, R will first attempt to use a system library — for example, `/usr/local/lib/R/site-library`. This path can only be modified by a user with administrator privileges. This is why a personal library in the `$HOME` directory is a better option for users to install packages.

3.5 Conda packages

You can use the *Conda* package manager to create isolated environments and install software available from *Anaconda Cloud* repositories.

These environments can be activated, deactivated and removed when they are no longer needed.

3.5.1 Requirements

Attention: *You should not add sudo in the commands below.*

This method only installs files in your home directory and so does not require administrator privileges.

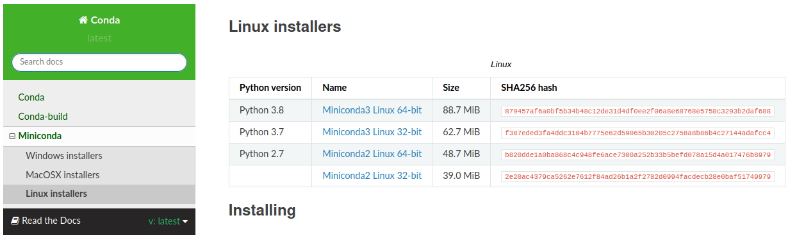
3.5.2 Installing Conda

A minimal installation of Conda can be setup by installing *Miniconda*. You will need to download and run the Miniconda installer following the steps below.

1. Downloading Miniconda

To download the correct version of Miniconda:

1. Open the Conda downloads[21] page in a browser.
2. Navigate to the Linux installers section (Fig. 64)



Linux installers

Linux

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Linux 64-bit	88.7 MiB	879457af6a0bf5b34b48c12de31d4df0ee2f06a8e68768e5758c3293b2daf688
Python 3.7	Miniconda3 Linux 32-bit	62.7 MiB	f387eded3fa4ddc3104b7775e62d59065b30205c2758a8b86b4c27144adafcc4
Python 2.7	Miniconda2 Linux 64-bit	48.7 MiB	b820dde1a0ba868c4c948fe6ace7300a252b33b5befd078a15d4a017476b8979
	Miniconda2 Linux 32-bit	39.0 MiB	2e20ac4379ca5262e7612f84ad26b1a2f2782d0994facdecb28e0baf51749979

Installing

Fig. 64: Miniconda installers for Linux

3. Click on the download link corresponding to:

   Python 3.8 — Miniconda3 Linux 64-bit

   Save the file. It will be saved as:

   `Miniconda3-latest-Linux-x86_64.sh`

Note: You can verify the file's SHA256 checksum to ensure

[21] https://docs.conda.io/en/latest/miniconda.html

it has been downloaded correctly. *How?*

2. Running the installer

Run the installer using the `bash` command:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

This will print a welcome message:

```
Welcome to Miniconda3 py38_4.8.3

In order to continue the installation process, please
review the license agreement.
Please, press ENTER to continue
```

3. Reviewing licence agreement

Press the `ENTER` key to view the licence agreement. Scroll down to read the licence. Towards the end, you will notice a prompt:

```
Do you accept the license terms? [yes|no]
[no] >>> yes
```

You will need to type *yes* to accept the agreement and then press the `ENTER` key to proceed.

4. Confirming Miniconda install location

Next, you will be asked to provide a path to install Miniconda.

```
Miniconda3 will now be installed into this location:
/home/user/miniconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/user/miniconda3] >>>
```

Press `ENTER` here to accept the default value.

Installation will now proceed. When complete, you will notice a prompt asking if you would like to *initialize Miniconda 3*.

```
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

Type `yes` and then press the `ENTER` key. This will add the `conda` command to your `$PATH`.

As a result of this configuration, Conda `base` environment will be activated automatically when you open a terminal session. This can be disabled in the next step. *Why?*

5. Disabling auto-activation of base environment

Open a new terminal. Your shell prompt should now appear like the following:

```
(base) user@cookbook:~$
```

The `(base)` label at the beginning of the prompt, indicates that Conda base environment is now active.

To disable this behaviour, so you can activate the environment manually when you need it, run the following command:

```
conda config --set auto_activate_base false
```

6. Setting up channels

Channels provide additional software for Conda.

Conda's configuration includes a defaults[22] channel. The *bioconda* and conda-forge[23] channels can also be added to access an even larger collection of software. The bioconda channel, for example, provides over 7000 packages of Bioinformatics software.

To add these channels to your configuration, you can run the commands below.

[22] https://repo.anaconda.com/pkgs
[23] https://conda-forge.org/

Attention: *You will need to run these commands in the same order as given below.*

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
```

—

Installation and configuration of Conda is now complete.

You can now start using Conda to create environments and install packages from repositories.

3.5.3 Using Conda

In this short guide on using Conda, I will show you how to:

1. Create an environment
2. Activate the environment
3. Search, install and use packages — using the NCBI BLAST+ program as an example
4. Remove an environment when you no longer need it

Note: A cheat sheet[24] with commonly used commands for working with Conda is available from the project's website.

Creating an environment

To create an environment, use the `conda create` command with the `-n` option, followed by a name for the environment — `blast` in this example:

```
conda create -n blast
```

[24] https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html

Activating an environment

You can get a *list* of all environments using the command:

```
conda envs list
```

To activate an environment, use the `conda activate` command with the *name* of the environment:

```
conda activate blast
```

Searching, installing and using Packages

To search and install packages:

1. First, activate the environment, if you haven't done so already.

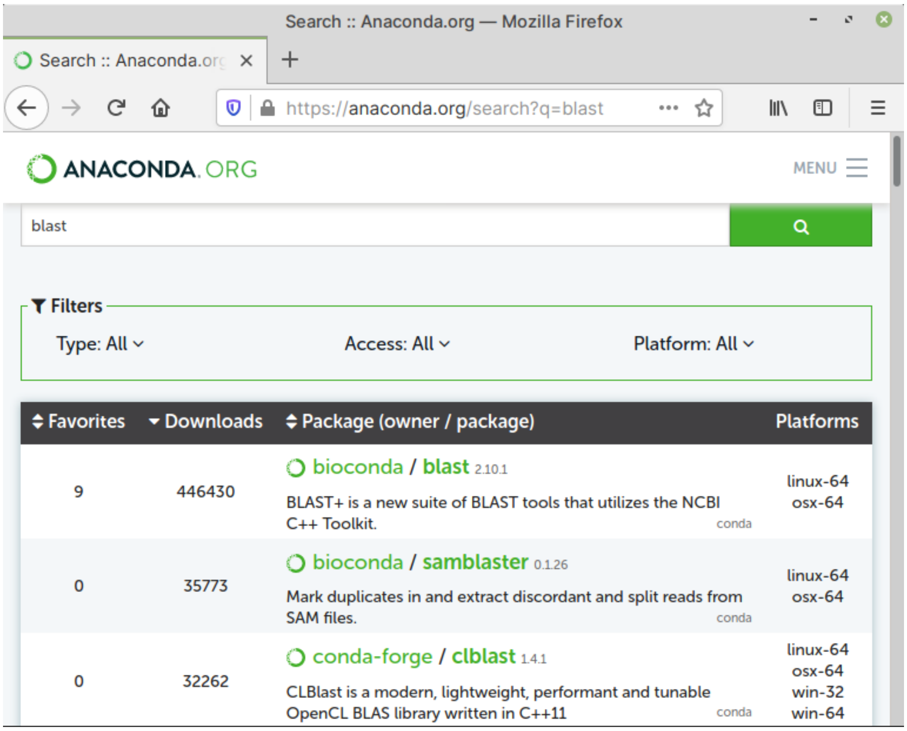
   To demonstrate, I will activate the `blast` environment created earlier:

   ```
   conda activate blast
   ```

2. To search for packages, open Anaconda.org[25] in a browser.

   Type your search term in the Search Packages field and press the ENTER key (Fig. 65).

[25] https://anaconda.org/


Search :: Anaconda.org — Mozilla Firefox
Search :: Anaconda.org
https://anaconda.org/search?q=blast
ANACONDA.ORG
MENU

blast

Filters

Type: All Access: All Platform: All

Favorites	Downloads	Package (owner / package)	Platforms
9	446430	bioconda / blast 2.10.1 BLAST+ is a new suite of BLAST tools that utilizes the NCBI C++ Toolkit. conda	linux-64 osx-64
0	35773	bioconda / samblaster 0.1.26 Mark duplicates in and extract discordant and split reads from SAM files. conda	linux-64 osx-64
0	32262	conda-forge / clblast 1.4.1 CLBlast is a modern, lightweight, performant and tunable OpenCL BLAS library written in C++11 conda	linux-64 osx-64 win-32 win-64

Fig. 65: Searching for a package on Anaconda.org

Alternatively, you can use the `conda search` command:

```
conda search blast
```

This will output matching packages:

```
Loading channels: done
# Name    Version            Build  Channel
blast     2.2.31                 1  bioconda
...
blast      2.9.0 pl526he19e7b1_7  bioconda
```

(continues on next page)

(continued from previous page)

```
...
blast      2.10.1 pl526he19e7b1_2   bioconda
```

3. To install the package, use the `conda install` command with the *name* and *version* number of the package.

   Attention: You will need to use the *highest* version number of the program, obtained from search results. Otherwise, an older version might get installed.

```
conda install blast==2.10.1
```

4. Once installed, you can start using programs included with the package, for example:

```
(blast) user@cookbook:~$ blastn -version
```

   Output:

```
blastn: 2.10.1+
 Package: blast 2.10.1, build Oct 14 2020 11:36:30
```

Deactivating an environment

When your work is complete, you can deactivate an environment. To do so, use the command:

```
conda deactivate
```

Your shell prompt will change to its default state i.e., without the name of the Conda environment — `(blast)` in this case.

Removing an environment

To remove an environment, when you no longer need it, use the command:

```
conda remove -n blast --all
```

Here `-n` is used to indicate the name of environment you would like to remove and `--all` removes all packages installed in that environment.

3.5.4 Notes

How to verify the installer's SHA256 checksum

1. Navigate to the folder containing the installer file — `Miniconda3-latest-Linux-x86_64.sh`:

   ```
   cd Downloads
   ```

2. Use the `sha256sum` command to verify the SHA256 checksum of the file:

   ```
   sha256sum Miniconda3-latest-Linux-x86_64.sh
   ```

   Output:

   ```
   879457af6a0bf5b34b48c12de31d4df0ee2f06a8e68768e5758c3
   ↪293b2daf688  Miniconda3-latest-Linux-x86_64.sh
   ```

The output of the `sha256sum` command should match the value listed on the downloads page.

Why disable auto-activation of base environment?

If you are installing packages using other methods, for example *Installing a Python package* from PyPI, there is a possibility that you might *accidentally* install the package in Conda `base` environment instead of the intended location.

Even if you disable auto-activation of the `base` environment, you can activate it using the `conda activate base` command, when you need it.

3.6 Debian packages

If your software of interest is available as a *Debian package*, you can use the *gdebi* program to install it. If the software has any *dependencies*, they will be installed automatically.

Examples of software that can be installed using this method:

RStudio Desktop — an IDE for the R programming language

Modeller — a program for comparative modelling of protein three-dimensional structures

MEGA — Molecular Evolutionary Genetics Analysis

3.6.1 Requirements

Attention: *This procedure installs software in system paths and so requires administrator privileges.*

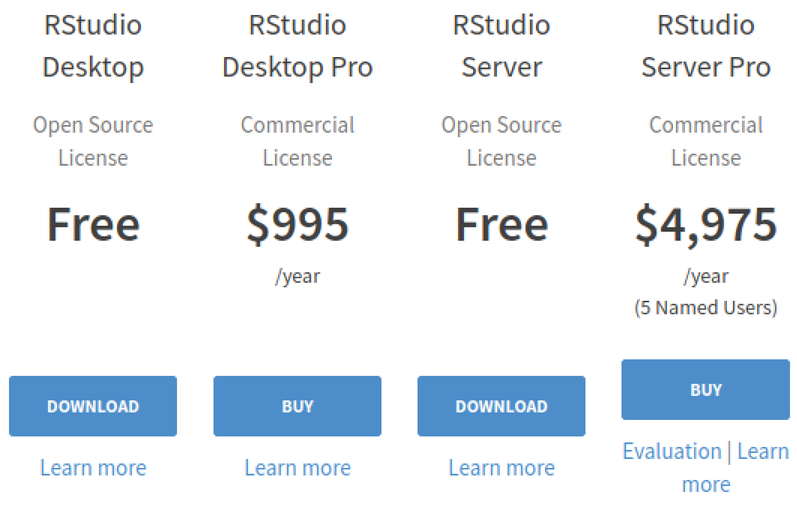
3.6.2 Installing a Debian package

To demonstrate the steps involved in installing a Debian package, I will download and install *RStudio Desktop* — an IDE for R, distributed as a Debian package.

1. Downloading Debian package

Open the RStudio project's download[26] page in a web browser.

Under `Choose Your Version`, click on the download button below `RStudio Desktop` (Open Source License) (Fig. 66).



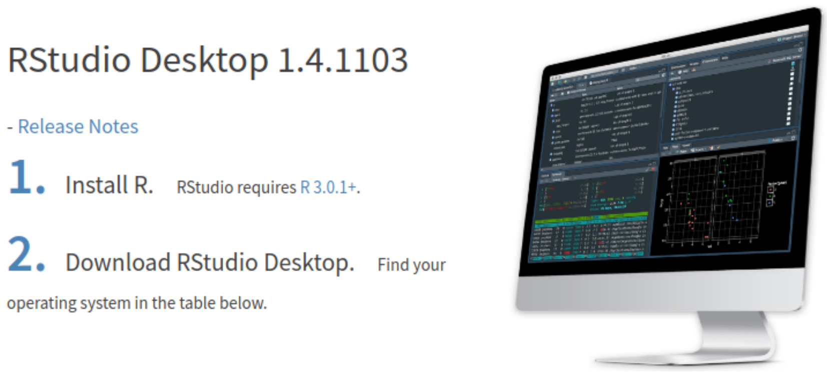
RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Server Pro
Open Source License	Commercial License	Open Source License	Commercial License
Free	$995 /year	Free	$4,975 /year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY
Learn more	Learn more	Learn more	Evaluation | Learn more

Fig. 66: Download RStudio Desktop (Open Source License)

This will take you to the installers section with information on

[26] https://rstudio.com/products/rstudio/download/

the current version (Fig. 67).


RStudio Desktop 1.4.1103

- Release Notes

1. Install R. RStudio requires R 3.0.1+.

2. Download RStudio Desktop. Find your operating system in the table below.

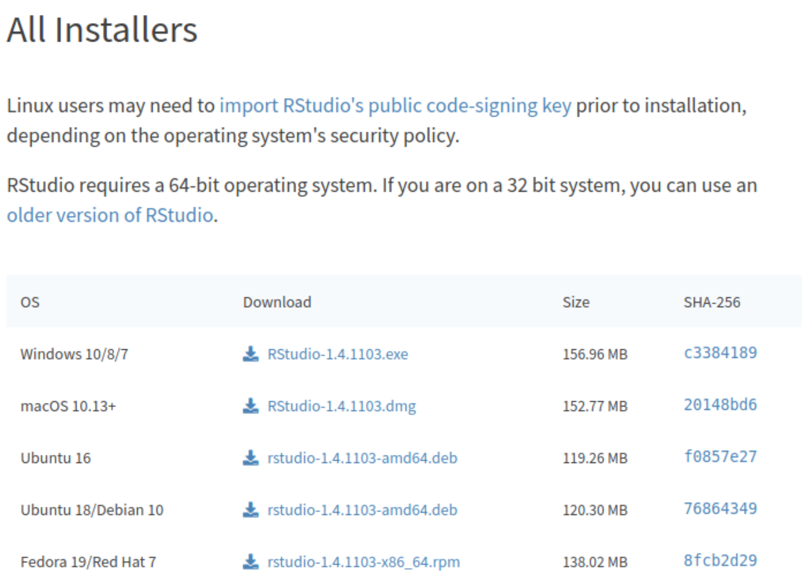
Fig. 67: RStudio Desktop installers

Note: Although the instructions in Fig. 67 state that you will need to install R before installing RStudio, you can safely *skip that step*.

The Gdebi program used here, will automatically install R, while installing RStudio.

Scroll down to the All Installers section.

Click on the download link corresponding to Ubuntu 18/Debian 10 (Fig. 68).



All Installers

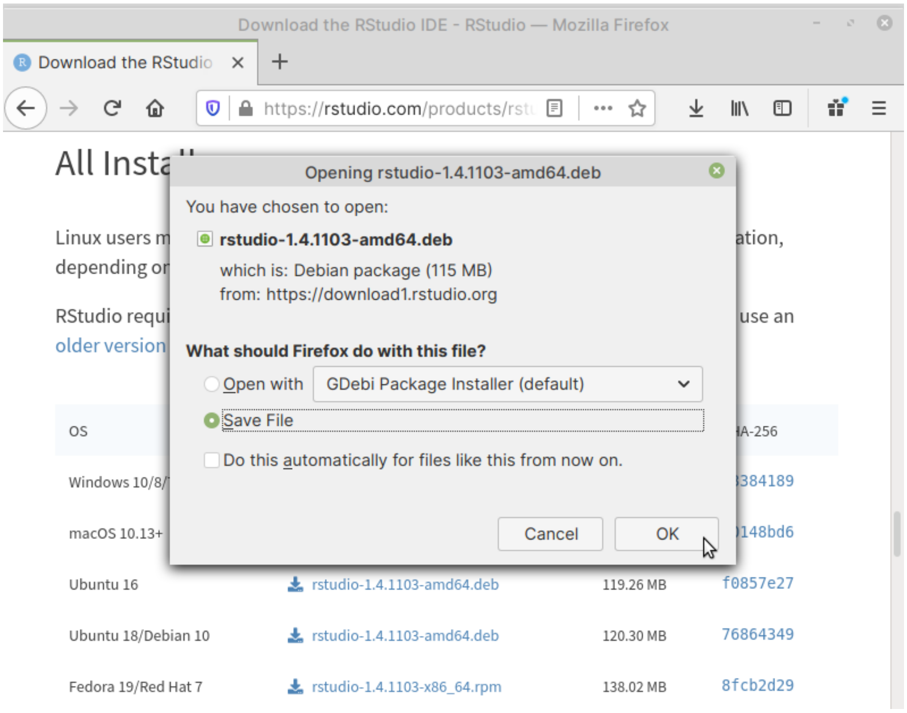
Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.4.1103.exe	156.96 MB	c3384189
macOS 10.13+	RStudio-1.4.1103.dmg	152.77 MB	20148bd6
Ubuntu 16	rstudio-1.4.1103-amd64.deb	119.26 MB	f0857e27
Ubuntu 18/Debian 10	rstudio-1.4.1103-amd64.deb	120.30 MB	76864349
Fedora 19/Red Hat 7	rstudio-1.4.1103-x86_64.rpm	138.02 MB	8fcb2d29

Fig. 68: RStudio installers for different operating systems

In the dialog window that appears, select Save File and then click on the OK button (Fig. 69).



Opening rstudio-1.4.1103-amd64.deb

You have chosen to open:

rstudio-1.4.1103-amd64.deb

which is: Debian package (115 MB)
from: https://download1.rstudio.org

What should Firefox do with this file?

Open with GDebi Package Installer (default) [✓ Save File]

Do this automatically for files like this from now on.

Cancel OK

Ubuntu 16	rstudio-1.4.1103-amd64.deb	119.26 MB	f0857e27
Ubuntu 18/Debian 10	rstudio-1.4.1103-amd64.deb	120.30 MB	76864349
Fedora 19/Red Hat 7	rstudio-1.4.1103-x86_64.rpm	138.02 MB	8fcb2d29

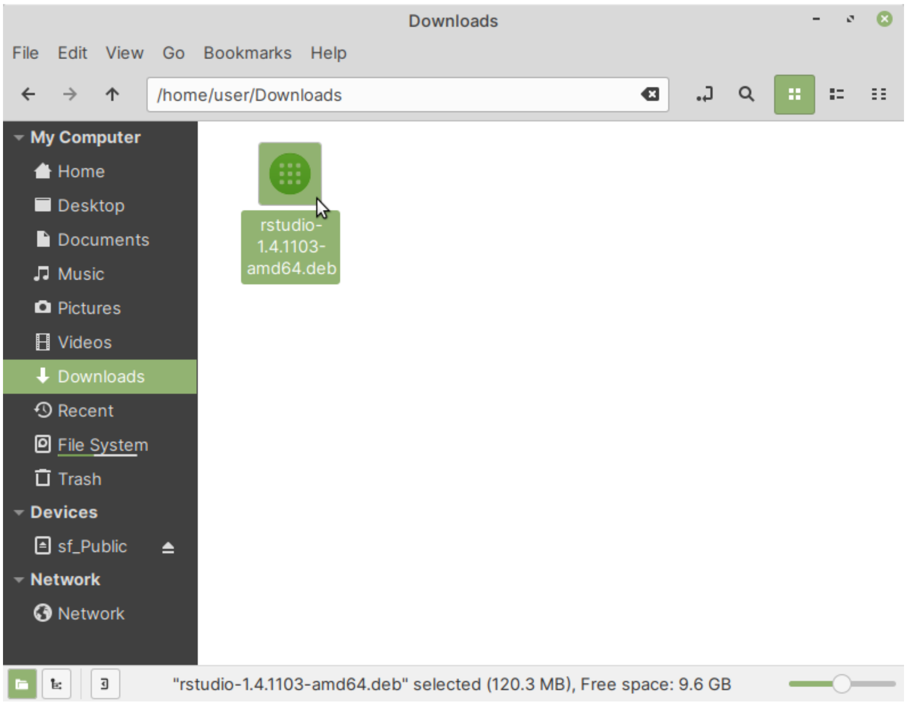
Fig. 69: Save installer file

Once the file download is complete, you can find it in your Downloads directory.

2. Installing downloaded package

Open file manager and navigate to your Downloads directory.

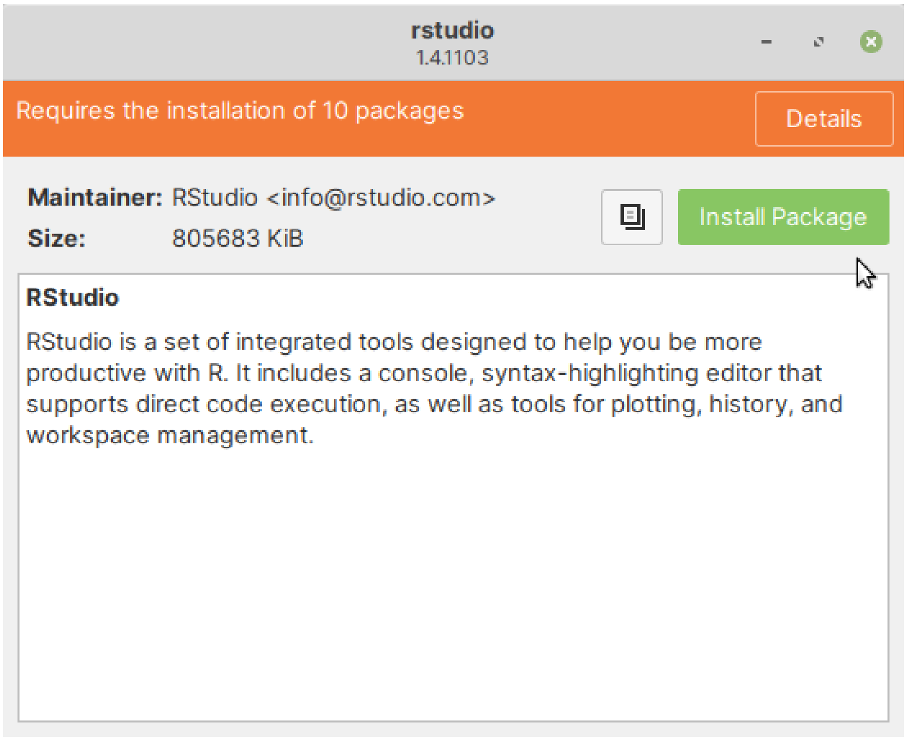
Double-click on the downloaded file (Fig. 70).



"rstudio-1.4.1103-amd64.deb" selected (120.3 MB), Free space: 9.6 GB

Fig. 70: Downloaded Debian package

This will open the Gdebi application. Click on the `Install Package` button (Fig. 71).



Requires the installation of 10 packages

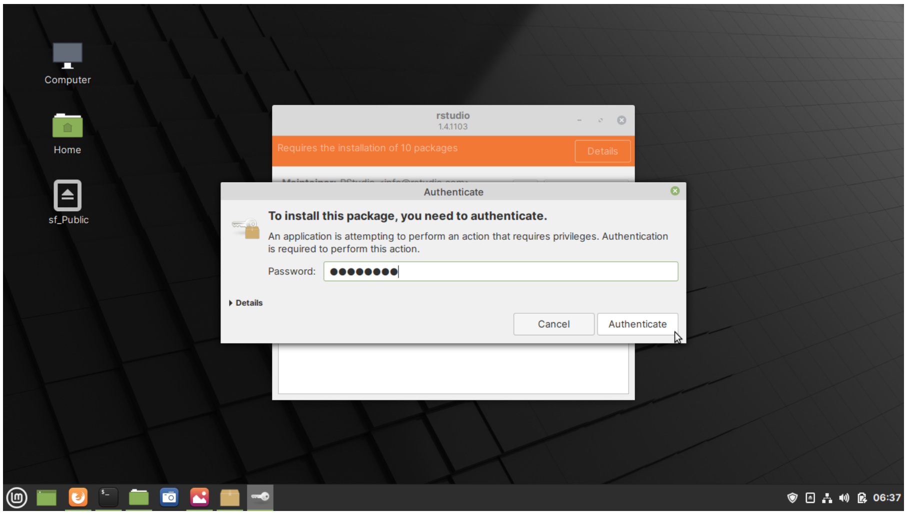
Maintainer: RStudio <info@rstudio.com>
Size: 805683 KiB

RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, and workspace management.

Fig. 71: Gdebi: Click on the Install Package button

You will be prompted to enter *your password*. Type in your password and click on the Authenticate button to proceed (Fig. 72).


rstudio
1.4.1103
Requires the installation of 10 packages

Authenticate

To install this package, you need to authenticate.

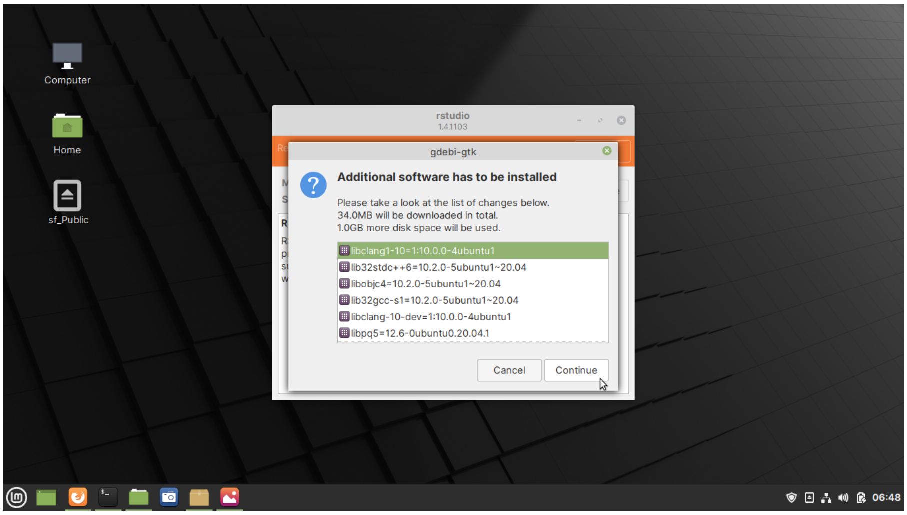
An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password: ●●●●●●●●

Fig. 72: Enter your password when prompted

As this package has additional dependencies, another dialog will appear listing these packages, which need to be installed in order for it to work.

Click on the Continue button to proceed (Fig. 73).



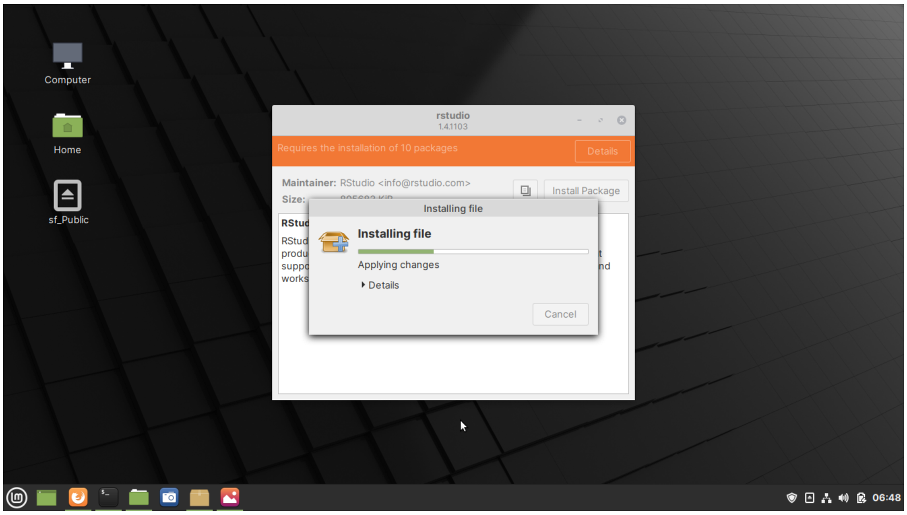
Additional software has to be installed

Please take a look at the list of changes below.
34.0MB will be downloaded in total.
1.0GB more disk space will be used.

```
libclang1-10=1:10.0.0-4ubuntu1
lib32stdc++6=10.2.0-5ubuntu1~20.04
libobjc4=10.2.0-5ubuntu1~20.04
lib32gcc-s1=10.2.0-5ubuntu1~20.04
libclang-10-dev=1:10.0.0-4ubuntu1
libpq5=12.6-0ubuntu0.20.04.1
```

Fig. 73: Additional dependencies to be installed

Installation will now proceed (Fig. 74).



Requires the installation of 10 packages

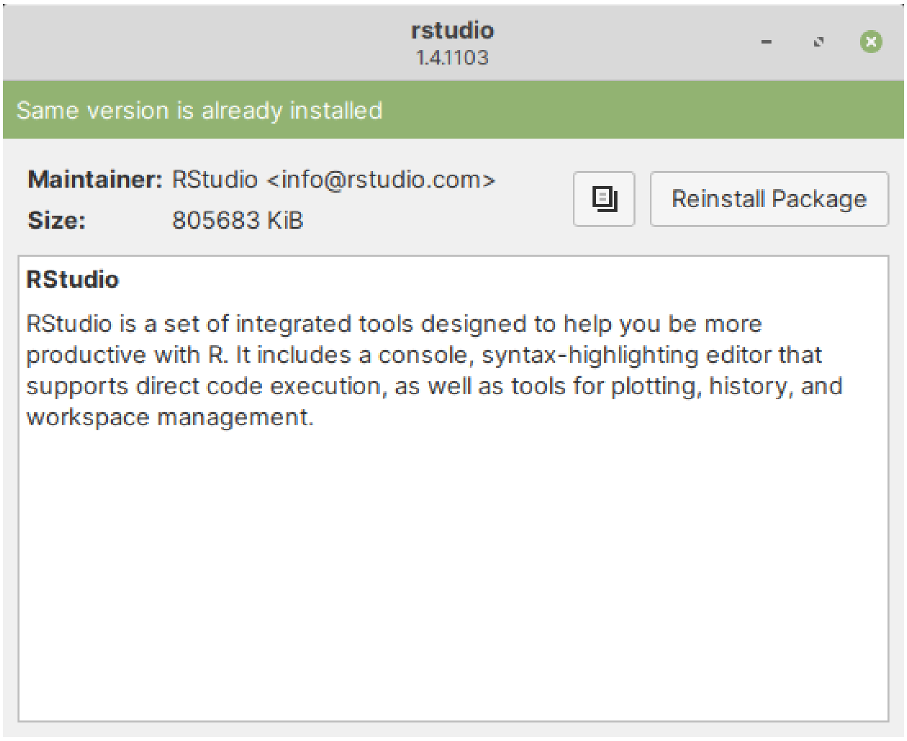
Maintainer: RStudio <info@rstudio.com>

Installing file

Applying changes

Fig. 74: Installation progress

When installation is complete, you will see a message `Same version is already installed`, at the top of the window (Fig. 75).



Same version is already installed

Maintainer: RStudio <info@rstudio.com>
Size: 805683 KiB

RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, and workspace management.

Fig. 75: Package installed successfully

3. Launching installed programs

You can launch installed programs from the `Applications Menu` (Fig. 76).

Note: This only applies to applications with a graphical user interface.

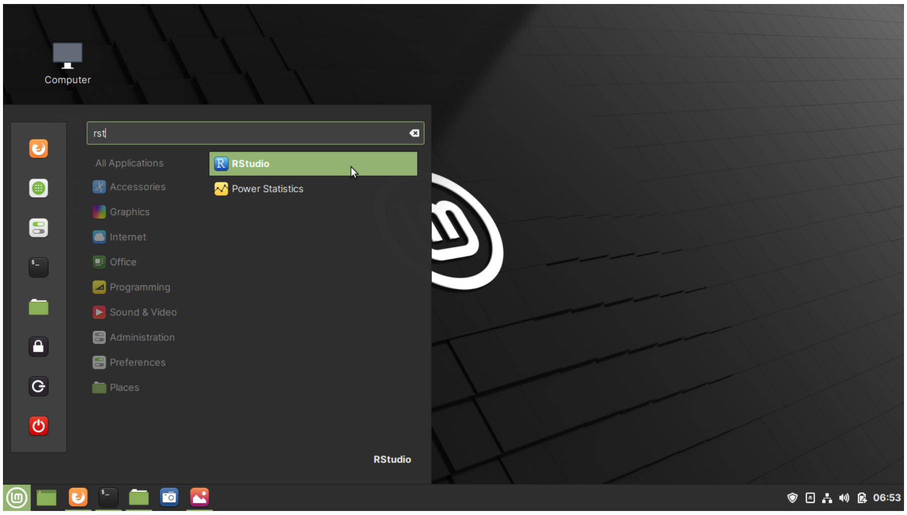
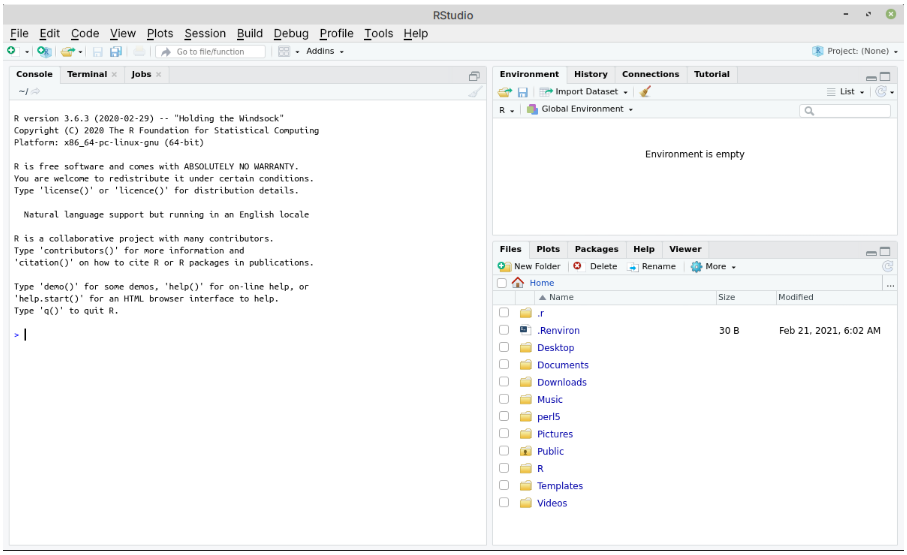


Fig. 76: Launching installed program from applications menu

The main window of the program will be displayed (Fig. 77).



```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name	Size	Modified
.r		
.Renviron	30 B	Feb 21, 2021, 6:02 AM
Desktop		
Documents		
Downloads		
Music		
perl5		
Pictures		
Public		
R		
Templates		
Videos		

Fig. 77: Main window of RStudio Desktop

3.6.3 Upgrading installed software

Upgrading software installed from a Debian package, is similar to installation. When a new version of the program is available, download the package file (`.deb`) and then follow the steps in *Installing a Debian package*.

3.6.4 Removing installed software

You can follow the steps in *Removing installed software*, to search and remove packages installed in this manner.

3.6.5 Notes

The command-line version

You can use the *apt* package manager to install Debian packages on the command-line.

These programs need to be updated manually

Programs installed in this manner will not *usually* be updated automatically. So, this method should only be used for software not available using *The quick and easy method*.

If dependencies cannot be located, installation will fail

Installation of some programs might fail, if dependencies cannot be located in software repositories. In that case, you will need to find and install those packages manually and then attempt to install the program again.

4 Using the Linux command line

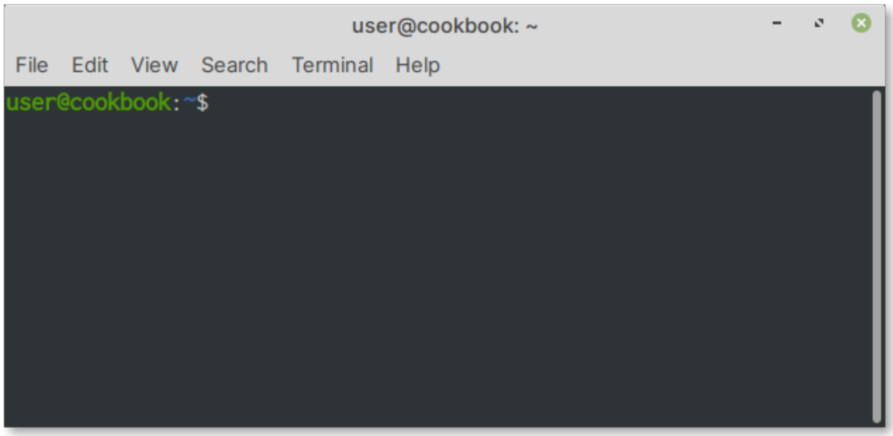
In this section, you will find an overview of the Linux command line, the shell and terminal programs, some commonly used commands, and a sample exercise.

4.1 Shell and Terminal

A *shell* is a program that provides an interface to the operating system's services. The most commonly used shell in Linux distributions is *Bash*.

A *terminal* program runs the shell and provides an interface for users to type commands.

The default terminal application in Linux Mint is Terminal (Fig. 78). You can launch it from the Applications Menu or by clicking on its icon in the Panel.



```
user@cookbook:~$
```

Fig. 78: A Terminal window

Once a terminal window is open, you can start typing commands at the $ prompt.

For example, the following command:

```
pwd
```

Will output the current working directory:

```
/home/user
```

4.1.1 The shell prompt

The prompt `user@cookbook:~$` in Fig. 78, indicates the following:

`user` — username of the account

`cookbook` — the computer's hostname

`~` — short form for the user's home directory i.e., `/home/user`

`$` — a regular user account. For the root user, this will change to `#`

4.2 Commands — an overview

This section provides an overview of some commands that are used frequently on Linux systems.

Familiarity with these will help you navigate the file system and work with files, and directories.

4.2.1 Command options

A command might support additional options.

For example, the command:

```
ls
```

— prints a list of files in a directory, whereas:

```
ls -a
```

— will also print hidden files in that directory.

4.2.2 Getting help on using commands

In most cases, you can see the complete list of options a command supports using `--help`:

```
ls --help
```

Output:

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory
↪by default).
Sort entries alphabetically if none of -cftuvSUX nor --
↪sort is specified.

Mandatory arguments to long options are mandatory for
↪short options too.
  -a, --all                  do not ignore entries
↪starting with .

...
```

Note: To keep this discussion short, the examples in this section, only includes *some* options supported by a command.

4.2.3 `date` — display current date and time

```
date
```

Output:

```
Wed 10 Mar 2021 10:33:11 GMT
```

4.2.4 cd — change directory

To change into a directory, use `cd` followed by the name of that directory.

For example, to change into `/home/user/Downloads`, use:

```
cd Downloads
```

Note: When you log in or open a new terminal window, you will be placed in your home directory — `/home/user`, where `user` is your username.

You can verify your current directory using the `pwd` command discussed *earlier*:

```
pwd
```

```
/home/user/Downloads
```

cd with no arguments

No matter where you are in the file system, typing `cd` *without any arguments* will take you to your home directory.

```
user@cookbook:/usr/share/dict$ cd
user@cookbook:~$
```

Go to parent directory

Use `cd ..` to go to the parent directory.

For example, when you are in `/home/user/Downloads` and type:

```
cd ..
```

You will be taken to `/home/user`.

A single dot (`.`) refers to the *current* directory.

4.2.5 `mkdir` — create new directory

Use `mkdir` followed by the name of the directory you would like to create:

```
mkdir Workspace
```

Attention: *The directory you're creating must not already exist or this command will not work.*

As discussed *earlier*, directory names are case-sensitive, so a directory named `Workspace` is different from another named `workspace`.

Spaces in directory names

If there are spaces in the directory name, use double quotes, for example:

```
mkdir "Project Work"
```

Without that, two separate directories — Project and Work will be created.

4.2.6 `rmdir` — remove empty directory

Use `rmdir` followed by the name of the directory you would like to remove:

```
rmdir Workspace
```

Attention: *This directory must be empty before it can be removed.*

There should be no other files or directories within that directory.

4.2.7 `ls` — list files

To list files in a directory, use the `ls` command:

```
ls
```

Output:

```
Desktop  Documents  Downloads  Music  Pictures  Public 
↪Templates  Videos
```

Listing hidden files

You can provide additional options to `ls`. It will change how the command works.

For example, to display hidden files and directories too, use:

```
ls -a
```

Output:

```
.              .cache     Downloads    .local       Public
..             .cinnamon  .gnupg       .bash_history  .
↪config
.bash_logout   Desktop    Pictures     .var         Videos
.bashrc        Documents  .linuxmint   .profile     .
↪Xauthority
```

Getting a detailed file listing

To get a detailed listing of files, use:

```
ls -l
```

Output:

```
total 32
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Desktop
drwxr-xr-x 2 user user 4096 Mar  4 11:41 Documents
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Downloads
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Music
drwxr-xr-x 2 user user 4096 Mar  4 11:27 Pictures
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Public
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Templates
drwxr-xr-x 2 user user 4096 Feb 17 11:13 Videos
```

Combining multiple options

You can combine multiple options like this:

```
ls -la
```

Output:

```
total 144
drwxr-xr-x 19 user user  4096 Mar 12 10:00 .
drwxr-xr-x  3 root root  4096 Feb 17 10:51 ..
```

(continues on next page)

(continued from previous page)

```
-rw-------  1 user user   190 Mar  4 10:32 .bash_history
-rw-r--r--  1 user user   220 Feb 17 10:51 .bash_logout
-rw-r--r--  1 user user  3771 Feb 17 10:51 .bashrc
drwx------ 17 user user  4096 Mar  3 19:28 .cache
drwxrwxr-x  4 user user  4096 Feb 17 11:40 .cinnamon
```

To see the list of all options that `ls` supports, use:

```
ls --help
```

4.2.8 cp — copy files

With the `cp` command, you can copy files or directories from one place (*source*) to another (*destination*).

Copying one file

To copy one file, use the following format:

```
cp source_file destination
```

Where `source_file` is the file you would like to copy and `destination` can *either* be a file name or a directory.

If destination is a file name

The copied file will have that file name.

For example:

```
cp /usr/share/dict/words /home/user/Documents/dictionary.
↪txt
```

This will copy the `words` file from `/usr/share/dict/` to `/home/user/Documents/` and save it as `dictionary.txt`.

If destination is a directory

The file will be copied *into* that directory with the *same* file name.

For example:

```
cp /usr/share/dict/words /home/user/Documents
```

This will copy the `words` file from `/usr/share/dict/` to `/home/user/Documents/` with the same file name.

Copying multiple files

To copy multiple files, use the following format:

```
cp source_file1 source_file2 destination
```

Where `source_file1` and `source_file2` are the files you would like to copy. You can have *any* number of source files. Here, *destination* is the directory where you would like to copy source files into.

For example:

```
cp /usr/share/dict/words /usr/share/dict/spanish /home/
↪user/Documents
```

This will copy the `words` and `spanish` dictionary files from `/usr/share/dict/` into `/home/user/Documents`.

Copying directories

You can copy entire directories using the `-r` (recursive) option.

For example:

```
cp -r /usr/share/dict /home/user/Documents
```

This will copy the `/usr/share/dict` directory and its contents to the `Documents` directory.

4.2.9 `cat` — display contents of files or combine them

With the `cat` command, you can view the contents of a file on the screen or combine contents of multiple files.

View contents of a file

To view the contents of a file, use `cat`, followed by the name of the file:

```
cat /usr/share/dict/words
```

```
A
A's
AMD
AMD's
AOL
AOL's
AWS
AWS's
Aachen
...
```

This will display the *entire contents* of the file.

Combine contents of files

If you include multiple file names in `cat`, it will combine them and print their contents on the screen.

If you would like to save the output to a file instead, use the following format:

```
cat file1.txt file2.txt > files.txt
```

The > operator redirects *standard output* to the file name following it.

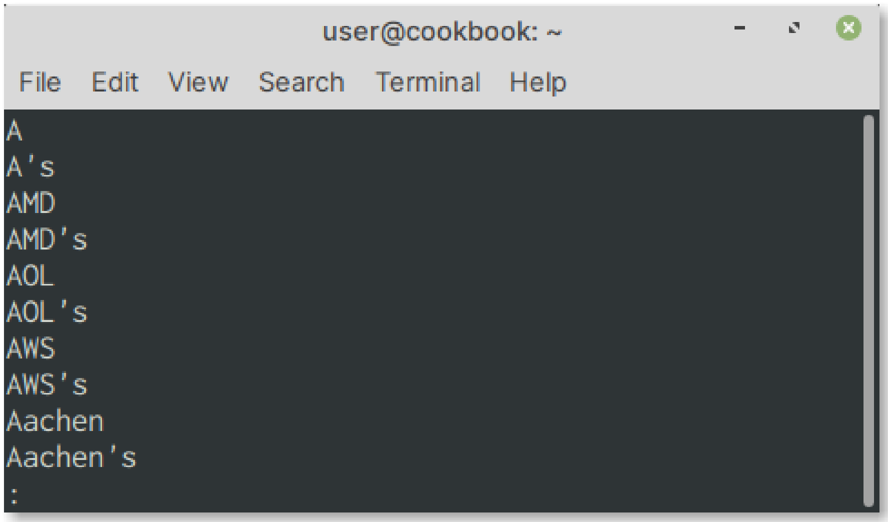
4.2.10 `less` — view and navigate file contents

With the `less` command, you have more control when viewing a file. It allows you to scroll up and down the file and provides features like searching the file using keywords.

To view the contents of a file, use `less`, followed by the name of the file:

```
less /usr/share/dict/words
```

When the file is open, it will appear as in Fig. 79.



```
A
A's
AMD
AMD's
AOL
AOL's
AWS
AWS's
Aachen
Aachen's
:
```

Fig. 79: The `words` file open in `less` command

At the `:` prompt, you can type commands that less will understand.

Navigating the file

You can scroll down the file by pressing the `ENTER` key.

Alternatively, you can use the `PgUp` and `PgDn` keys to scroll one page at a time.

Searching the file

You can search a file while viewing it in `less`.

To do this:

1. Type `/` (forward slash) key
2. Type the text you would like to search
3. Press the `ENTER` key
4. If there are matching results, they will be highlighted

Quitting the `less` command

To quit the less command, type `q` at the prompt.

4.2.11 head — print first few lines of a file

With the `head` command, you can view the first few (default: 10) lines of a file.

For example:

```
head /usr/share/dict/words
```

Will produce the following output:

```
A
A's
AMD
AMD's
AOL
AOL's
AWS
AWS's
Aachen
Aachen's
```

4.2.12 `tail` — print last few lines of a file

With the `tail` command, you can view the last few (default: 10) lines of a file.

For example:

```
tail /usr/share/dict/words
```

```
élan's
émigré
émigré's
émigrés
épée
épée's
épées
étude
étude's
études
```

The `tail` and the `head` command discussed *earlier*, are useful when you want to quickly view the contents of a *large* text file.

In both cases, you can control the *number of lines* displayed, using the `-n` option, instead of the default value (10).

For example:

```
tail -n 5 /usr/share/dict/words
```

will display the last 5 lines of the file.

4.2.13 `mv` — move a file or directory

With the `mv` command, you can move a file or directory from one location (source) to another (destination). You can choose to keep the existing file or directory name or rename them.

The basic format of the command is:

```
mv source destination
```

A safer approach is to add the `-iv` options to the command:

```
mv -iv source destination
```

With `-i` (*interactive*), `mv` will require your confirmation before overwriting a file or directory, if it exists already in destination.

`-v` (*verbose*) will print the command's actions on the screen.

For convenience, you can *add an alias*.

Sample files

To follow the examples below, you will need to copy the following files into your home directory using the *cp* command:

```
cp -v /usr/share/dict/words ~
cp -rv /usr/share/doc/bash ~
```

`~` is a shortcut for home directory.

Moving a file or directory

The simplest use case is to move a file or directory from one location (directory) to another.

Moving a File

For example, to move the `words` file copied *above* into your `Documents` directory, use:

```
mv -iv words Documents/
```

Output:

```
renamed 'words' -> 'Documents/words'
```

Moving a directory

Similarly, to move the `bash` directory copied *above* into your `Documents` directory, use:

```
mv -iv bash Documents/
```

Output:

```
renamed 'bash' -> 'Documents/bash'
```

In both cases, the file or directory name will not be changed.

Renaming a file or directory

In this case, you would like to rename a file or directory.

Renaming a file

To rename the `words` file copied *above* to `dictionary.txt`, use:

```
mv -iv words dictionary.txt
```

Alternatively, to move it into your `Documents` directory and rename it at the same time, use:

```
mv -iv words Documents/dictionary.txt
```

Renaming a directory

To rename the `bash` directory copied *above* into `bash-commands`, use:

```
mv -iv bash bash-commands
```

Notes

What happens if a file exists?

You will notice a prompt requesting you for confirmation to overwrite the file.

Type `y` and press the `ENTER` key to proceed:

```
mv: overwrite 'Documents/words'? y
renamed 'words' -> 'Documents/words'
```

To cancel, simply press `ENTER` key at the prompt.

What happens if a directory exists?

`mv` will overwrite a directory only if it is empty. You can either:

- *copy* files into destination directory or
- rename the destination directory

Adding an alias for `mv`

Rather than typing `mv -iv`, every time you need to use the command, you can add an *alias* for the command in your `~/.bash_aliases` file.

For example:

```
alias mv='mv -iv'
```

Now, when you type `mv`, you will actually be running `mv -iv`.

4.2.14 `rm` — remove files or directories

With the `rm` command, you can remove (or delete) files or directories.

The basic format of the command is:

```
rm source
```

A safer approach is to add the `-iv` options to the command:

```
rm -iv source
```

With `-i` (*interactive*), `rm` will require your confirmation before deleting a file or directory.

`-v` (*verbose*) will print the command's actions on the screen.

For convenience, you can *add an alias*.

Sample files

To follow the examples below, you will need to:

1. Copy the following files into your home directory using the *cp* command:

   ```
   cp -v /usr/share/dict/words ~
   cp -rv /usr/share/doc/bash ~
   ```

   ~ is a shortcut for home directory.

2. Create an empty directory:

```
mkdir empty-dir
```

Removing files

To remove a file, for example, the `words` file copied *above*, you can use:

```
rm -iv words
```

Output:

```
rm: remove regular file 'words'? y
removed 'words'
```

Removing directories

Removing empty directories

If the directory is empty, you can remove it using the `-d` option:

```
rm -d empty-dir
```

Alternatively, you can use the *rmdir* command:

```
rmdir empty-dir
```

Removing directories with content

If the directory has some content i.e., files or subdirectories, you will need to add the `-r` (recursive) option.

For example, using the `bash` directory copied *above*:

```
rm -ivr bash
```

This command will ask for your confirmation for deleting every file in the directory and then delete it:

```
rm: descend into directory 'bash'? y
rm: remove regular file 'bash/RBASH'? y
removed 'bash/RBASH'
...
rm: remove regular file 'bash/README.gz'? y
removed 'bash/README.gz'
rm: remove directory 'bash'? y
removed directory 'bash'
```

Instead of `-i`, you could use the `-I` option, which will only prompt once, when removing directories recursively:

```
rm -Ivr bash
```

Output:

```
rm: remove 1 argument recursively? y
removed 'bash/RBASH'
```

(continues on next page)

(continued from previous page)

```
...
removed 'bash/README.gz'
removed directory 'bash'
```

If you are *completely sure* you do not need the directory and its contents, you can *force* its removal using the `-f` option:

```
rm -vrf bash
```

`rm` will delete the directory without confirmation.

Notes

Adding an alias for `rm`

Rather than typing `rm -iv`, every time you need to use the command, you can add an *alias* for the command in your `~/.bash_aliases` file.

For example:

```
alias rm='rm -iv'
```

Now, when you type `rm`, you will actually be running `rm -iv`.

4.2.15 `echo` — display text or values of variables

With the `echo` command, you can print text or the values of *environment variables* on the screen.

The basic format of the command is:

```
echo text_or_variable
```

Printing text

To print text on the screen, use `echo` followed by the text you would like to display.

```
echo "Hello, This is echo!"
```

Output:

```
Hello, This is echo!
```

Printing values of environment variables

You can also use `echo` to display values of your environment variables.

For example:

```
echo $PATH
```

The $ in PATH indicates, it is a variable.

Output:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
↪bin:/usr/games:/usr/local/games:/snap/bin
```

PATH contains the list of paths (directories) that will be searched for locating programs or commands.

You can set or modify environment variables in your ~/.bashrc.

4.3 Other useful commands

These are some additional commands that you might find useful.

4.3.1 `df` — report disk usage

The `df` command will report used and available disk space on all storage devices connected to the system.

For example, to find the amount of disk space used in `/` — the root filesystem, you can use the command:

```
df -h /
```

Output:

```
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vgmint-root  28G  6.9G   20G  26% /
```

The `-h` option makes the output human-readable. Without this, available disk space will be reported as `20600196` instead of `20G`.

4.3.2 du — estimate disk usage of files or directories

Calculate the amount of disk space used by files or directories.

For example, to calculate the amount of disk space used by files in your home directory, use the command:

```
du -sh ~
```

Output:

```
7.4G .
```

The `-s` option prints a summary instead of printing space used by individual files.

`-h` makes the output human-readable.

4.3.3 `find` — search for files

Search for files using their file name, file type etc.,

For example, to find files in the home directory with `.bash` in their name, use the command:

```
find ~ -name ".bash*"
```

Output:

```
/home/user/.bashrc
/home/user/.bash_history
/home/user/.bash_logout
```

Note: The ~ here is a shortcut for home directory.

The asterisk symbol (*) in `.bash*` acts as a wildcard.

The `find` command does not use a database, so the search will be slower when compared to `locate`. However, it can identify files that were created or modified recently.

4.3.4 `free` — view free and used memory

With `free`, you can find the amount of free and used memory (RAM (Random Access Memory)) and swap space.

Usage:

```
free -h
```

Output:

```
              total        used        free      shared ⍁
↪buff/cache   available
Mem:          3.8Gi       618Mi       1.3Gi       7.0Mi  ⍁
↪     1.9Gi       3.0Gi
Swap:         979Mi          0B       979Mi
```

The `-h` option makes the output human-readable with corresponding units displayed. In the example above, the amount of free physical memory (RAM) is 1.3 GiB (Gibibytes).

4.3.5 `locate` — find files using their names

Search for files using their file names. This method is quick as it uses a file name database.

For example:

```
locate bashrc
```

Output:

```
/etc/bash.bashrc
/etc/skel/.bashrc
/home/user/.bashrc
...
```

Note: This method might not find files that were created or modified recently.

Newer files might not yet be included in the database. To update the database manually, you can run the command:

```
sudo updatedb
```

Alternatively, you can use the *find* command to search for new files.

4.3.6 `which` — locate a command

The `which` command prints the complete path to a command, if it exists in the search path i.e., `$PATH`.

For example:

```
which python3
```

Output:

```
/usr/bin/python3
```

4.4 Editing text files using nano

GNU nano is a command-line text editor with features like undo/redo, syntax colouring, interactive search and replace and more.

To open the program, type `nano` in a terminal session.

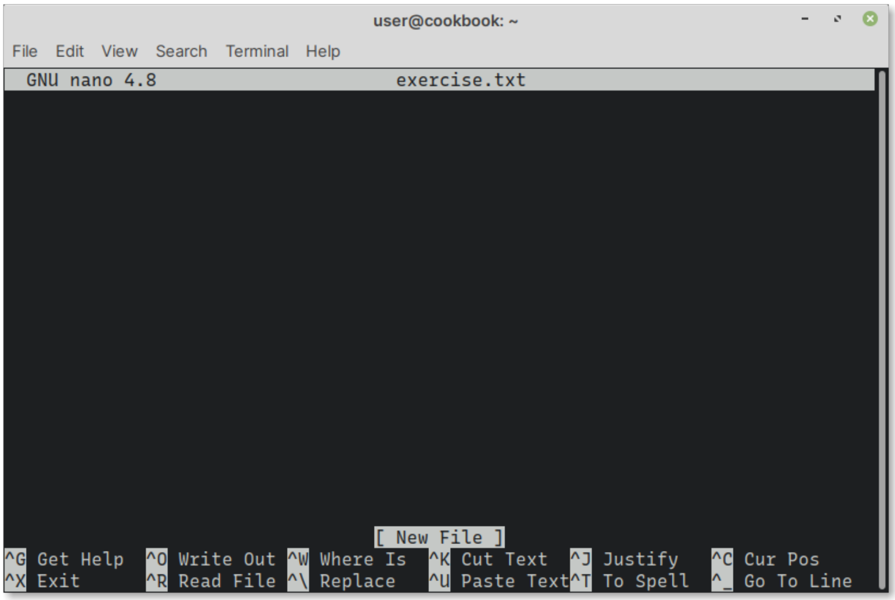
4.4.1 Create new file

To create a new file, use nano followed by the name of the file you would like to create.

For example:

```
nano exercise.txt
```

This will open the file for writing (Fig. 80).



```
  GNU nano 4.8                           exercise.txt







                                  [ New File ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text^T To Spell    ^_ Go To Line
```

Fig. 80: Creating a file in nano

You can start typing text at this stage.

4.4.2 Pasting text from clipboard

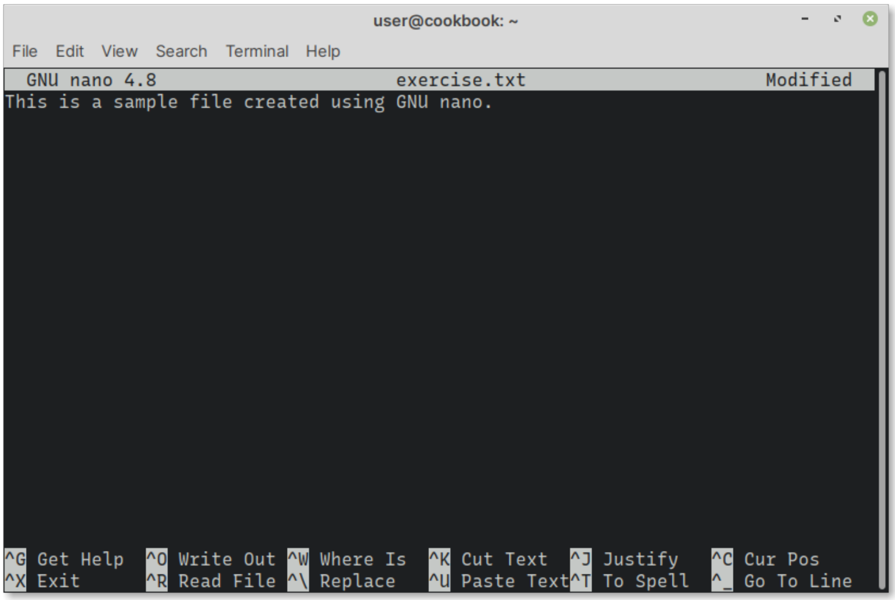
To copy and paste text from a different program, for example, a web browser into a file you are editing in nano, do the following:

1. Copy the text from the program to clipboard using `Ctrl + c`
2. In nano, move the cursor to the position where you would like to paste the text
3. Use the `Shift + Ctrl + v` keyboard shortcut to paste the text

4.4.3 Saving changes

To save the file, use the shortcut `Ctrl + o`.

This corresponds to the `^O Write Out` function displayed at the bottom (Fig. 81). The ^ here refers to the `Ctrl` key.

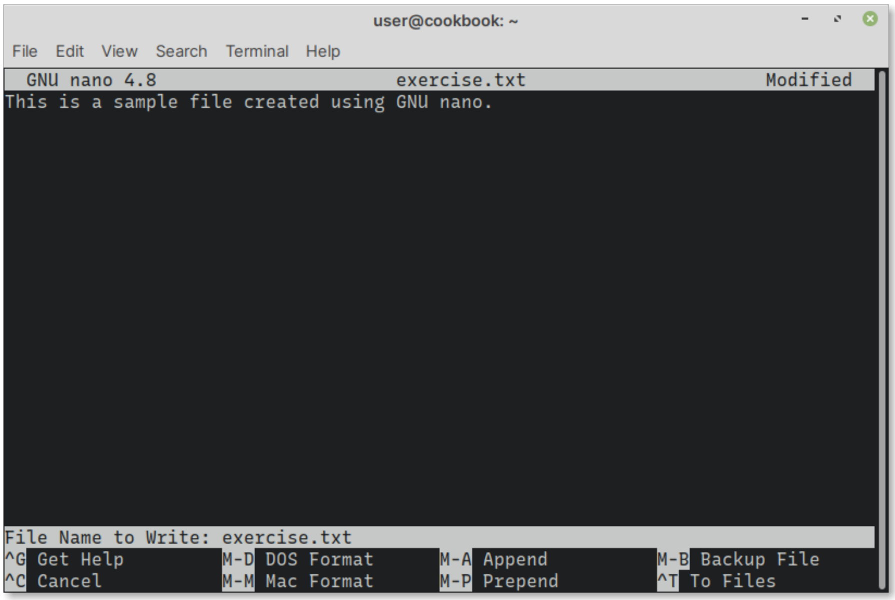


```
  GNU nano 4.8                    exercise.txt                     Modified
This is a sample file created using GNU nano.

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text^T To Spell    ^_ Go To Line
```

Fig. 81: To save changes use the Write Out function

A prompt like the following will be displayed at the bottom (Fig. 82):

```
File Name to Write: exercise.txt
```



```
  GNU nano 4.8                    exercise.txt                    Modified
This is a sample file created using GNU nano.









File Name to Write: exercise.txt
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

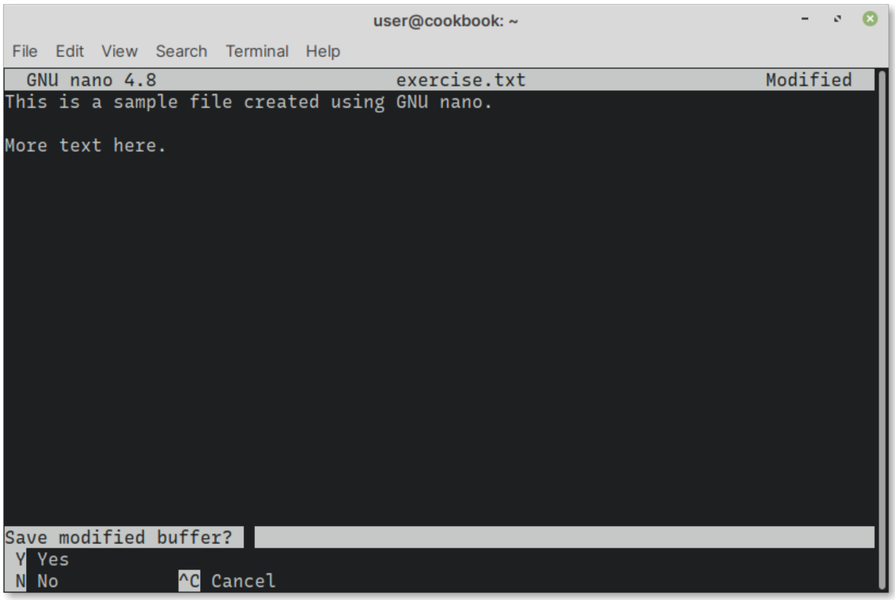
Fig. 82: Enter a file name to save the file

Press the ENTER key to accept the default value — the original file name.

4.4.4 Exit nano

To exit nano, use the `Ctrl + x` shortcut.

If there are no changes, nano will close. If there are unsaved changes, it will prompt you to save them (Fig. 83).



```
  GNU nano 4.8                    exercise.txt                    Modified
This is a sample file created using GNU nano.

More text here.













Save modified buffer?
 Y Yes
 N No             ^C Cancel
```

Fig. 83: Confirmation to save changes to file

Type `y` to save changes or `n` to discard them.

4.5 Exercise — using the command-line

This is an exercise in using the command-line to accomplish a task. You will be making use of the commands *discussed earlier*.

Task

Given a protein sequence, identify matching sequences from a protein sequence database.

Approach

Using programs in the NCBI BLAST+ package, you can search a database of sequences using sequence (query) to identify matching sequences.

4.5.1 Summary of steps

1. Install NCBI BLAST+
2. Download protein query sequence
3. Download protein sequence database and format it
4. Search database using the query sequence

4.5.2 Get sample data

To proceed, you will need to download the protein query sequence and database used in this exercise.

Download query sequence

The protein query sequence used in this exercise is *Spike glycoprotein* from Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It is available from UniProtKB[27] — the protein knowledge base.

The database identifier for this protein is P0DTC2[28]. You can download the sequence in FASTA format from the entry page or using this direct link:

https://www.uniprot.org/uniprot/P0DTC2.fasta

Download protein database

The database used in this exercise is UniProtKB Swiss-Prot[29]. It is a manually annotated database of protein sequences with added functional information.

You can download the entire database as a compressed FASTA format file from the downloads[30] page on the website.

[27] https://uniprot.org/uniprot/
[28] https://www.uniprot.org/uniprot/P0DTC2
[29] https://uniprot.org/uniprot/?query=reviewed:yes
[30] https://www.uniprot.org/downloads

4.5.3 Install NCBI BLAST+ package

Attention: *This procedure installs software in system paths and so requires administrator privileges.*

NCBI BLAST+ is available in the Linux package repositories. You can install it using `apt`:

```
sudo apt install ncbi-blast+
```

Type `y` when prompted to continue.

```
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmbedcrypto3 libmbedtls12 libmbedx509-0 ncbi-data
The following NEW packages will be installed:
  libmbedcrypto3 libmbedtls12 libmbedx509-0 ncbi-blast+
↪ncbi-data
0 upgraded, 5 newly installed, 0 to remove and 0 not
↪upgraded.
Need to get 14.9 MB of archives.
After this operation, 75.0 MB of additional disk space
↪will be used.
Do you want to continue? [Y/n] y
```

4.5.4 Download query sequence

You can follow these steps to download the query sequence:

1. Create new directory
2. Change into it
3. Download query sequence
4. View the downloaded sequence (optional)

Create new directory — `mkdir`

To keep the input and output files related to this project together, create a new directory in your home directory using the *mkdir* command.

```
mkdir -p ~/projects/sars-cov-2
```

Here:

`~` is shortcut for home directory.

`-p` creates parent directories if necessary. In this case, the `projects` directory does not exist, so it is also created.

Change directory — `cd`

Change into the newly created directory using the *cd* command:

```
cd ~/projects/sars-cov-2
```

Download sequence — `wget`

To download the sequence file, you can use the `wget` command with the link to download as the argument. In this case, the link to download is the URL corresponding to the FASTA format file (see *sample data*):

```
wget https://www.uniprot.org/uniprot/P0DTC2.fasta
```

When the download is complete, you can use the *ls* command to verify if the file exists:

```
ls -l
```

Output:

```
total 4
-rw-rw-r-- 1 user user 1414 Feb 10 00:00 P0DTC2.fasta
```

View downloaded sequence — `cat` or `less`

Since `P0DTC2.fasta` is in FASTA format — a plain-text format, you can use the *cat* command to view the file's contents:

```
cat P0DTC2.fasta
```

Output:

```
>sp|P0DTC2|SPIKE_SARS2 Spike glycoprotein OS=Severe acute
↪respiratory syndrome coronavirus 2 OX=2697049 GN=S PE=1
↪SV=1
MFVFLVLLPLVSSQCVNLTTRTQLPPAYTNSFTRGVYYPDKVFRSSVLHSTQDLFLPFFS
NVTWFHAIHVSGTNGTKRFDNPVLPFNDGVYFASTEKSNIIRGWIFGTTLDSKTQSLLIV
NNATNVVIKVCEFQFCNDPFLGVYYHKNNKSWMESEFRVYSSANNCTFEYVSQPFLMDLE
GKQGNFKNLREFVFKNIDGYFKIYSKHTPINLVRDLPQGFSALEPLVDLPIGINITRFQT
LLALHRSYLTPGDSSSGWTAGAAAYYVGYLQPRTFLLKYNENGTITDAVDCALDPLSETK
CTLKSFTVEKGIYQTSNFRVQPTESIVRFPNITNLCPFGEVFNATRFASVYAWNRKRISN
CVADYSVLYNSASFSTFKCYGVSPTKLNDLCFTNVYADSFVIRGDEVRQIAPGQTGKIAD
YNYKLPDDFTGCVIAWNSNNLDSKVGGNYNYLYRLFRKSNLKPFERDISTEIYQAGSTPC
NGVEGFNCYFPLQSYGFQPTNGVGYQPYRVVVLSFELLHAPATVCGPKKSTNLVKNKCVN
FNFNGLTGTGVLTESNKKFLPFQQFGRDIADTTDAVRDPQTLEILDITPCSFGGVSVITP
GTNTSNQVAVLYQDVNCTEVPVAIHADQLTPTWRVYSTGSNVFQTRAGCLIGAEHVNNSY
ECDIPIGAGICASYQTQTNSPRRARSVASQSIIAYTMSLGAENSVAYSNNSIAIPTNFTI
SVTTEILPVSMTKTSVDCTMYICGDSTECSNLLLQYGSFCTQLNRALTGIAVEQDKNTQE
VFAQVKQIYKTPPIKDFGGFNFSQILPDPSKPSKRSFIEDLLFNKVTLADAGFIKQYGDC
LGDIAARDLICAQKFNGLTVLPPLLTDEMIAQYTSALLAGTITSGWTFGAGAALQIPFAM
QMAYRFNGIGVTQNVLYENQKLIANQFNSAIGKIQDSLSSTASALGKLQDVVNQNAQALN
TLVKQLSSNFGAISSVLNDILSRLDKVEAEVQIDRLITGRLQSLQTYVTQQLIRAAEIRA
SANLAATKMSECVLGQSKRVDFCGKGYHLMSFPQSAPHGVVFLHVTYVPAQEKNFTTAPA
ICHDGKAHFPREGVFVSNGTHWFVTQRNFYEPQIITTDNTFVSGNCDVVIGIVNNTVYDP
LQPELDSFKEELDKYFKNHTSPDVDLGDISGINASVVNIQKEIDRLNEVAKNLNESLIDL
QELGKYEQYIKWPWYIWLGFIAGLIAIVMVTIMLCCMTSCCSCLKGCCSCGSCCKFDEDD
SEPVLKGVKLHYT
```

For more control, you can use the *less* command instead of `cat`.

4.5.5 Download protein sequence database

You can follow these steps to download and prepare the protein sequence database:

1. Create new directory
2. Change into it
3. Download the database archive
4. Uncompress (or extract) the database archive
5. Format the database

Create new directory — `mkdir`

In order to keep all BLAST databases in one location, create a directory to store them using the *mkdir* command:

```
mkdir ~/databases
```

Change directory — `cd`

Change into the newly created directory using the *cd* command:

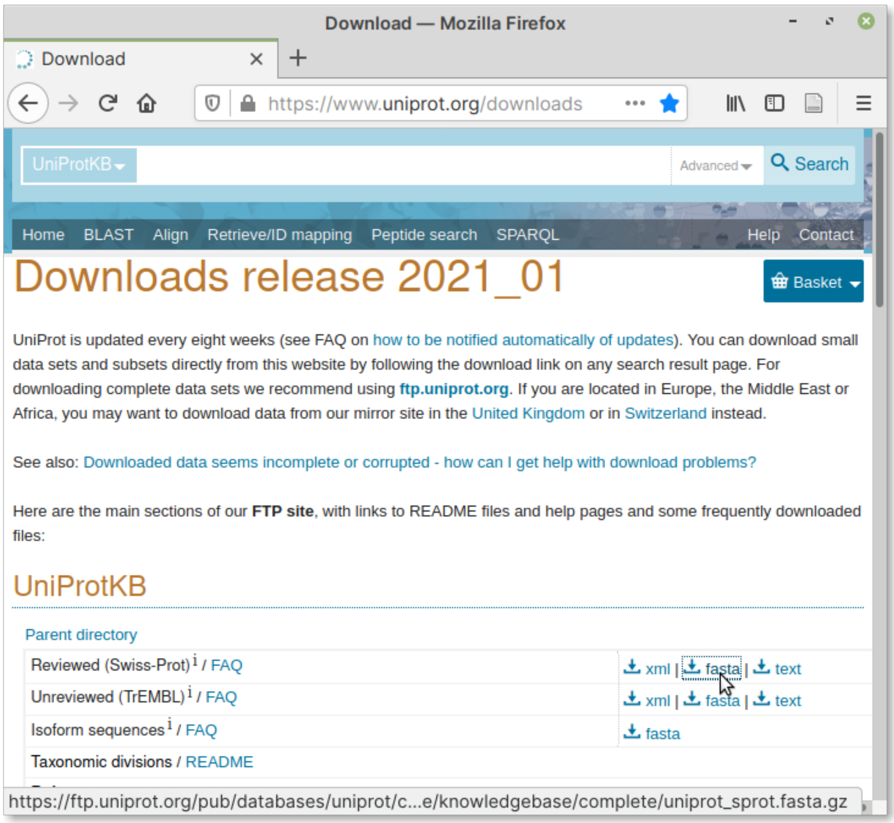
```
cd ~/databases
```

Download the database archive — `wget`

Visit the database downloads[31] page on the UniProt website.

Navigate to the UniProtKB section.

Right-click on the fasta download link corresponding to Reviewed (Swiss-Prot) and then copy it to clipboard (Fig. 84).



Download — Mozilla Firefox

https://www.uniprot.org/downloads

Downloads release 2021_01

UniProt is updated every eight weeks (see FAQ on how to be notified automatically of updates). You can download small data sets and subsets directly from this website by following the download link on any search result page. For downloading complete data sets we recommend using ftp.uniprot.org. If you are located in Europe, the Middle East or Africa, you may want to download data from our mirror site in the United Kingdom or in Switzerland instead.

See also: Downloaded data seems incomplete or corrupted - how can I get help with download problems?

Here are the main sections of our **FTP site**, with links to README files and help pages and some frequently downloaded files:

UniProtKB

Parent directory

Reviewed (Swiss-Prot) i / FAQ	xml | fasta | text
Unreviewed (TrEMBL) i / FAQ	xml | fasta | text
Isoform sequences i / FAQ	fasta
Taxonomic divisions / README	

https://ftp.uniprot.org/pub/databases/uniprot/c...e/knowledgebase/complete/uniprot_sprot.fasta.gz

Fig. 84: Download link for Swiss-Prot database

To download the database, you can use the `wget` command with

[31] https://www.uniprot.org/downloads

the link to download as the argument:

```
wget https://ftp.uniprot.org/pub/databases/uniprot/
↪current_release/knowledgebase/complete/uniprot_sprot.
↪fasta.gz
```

When the download is complete, you will find a file named `uniprot_sprot.fasta.gz` in the current directory. You can use the *ls* command to verify if it exists:

```
ls -lh
```

Output:

```
total 86M
-rw-rw-r-- 1 user user 86M Feb 10 15:00 uniprot_sprot.
↪fasta.gz
```

Since this file is in a compressed format (`.gz`), you will need to uncompress it before proceeding.

Uncompress the database archive — `gunzip`

To uncompress (or extract) the database archive file downloaded in the previous step, you can use the `gunzip` command.

Note: By default, gunzip will remove the original compressed file after extraction.

If you would like to keep the original file (`.gz`), you can include the -k (keep input files) option with gunzip.

Provide the file name of the downloaded file as the argument:

```
gunzip uniprot_sprot.fasta.gz
```

When the extraction is complete, you will find the database file in FASTA format in the same directory:

```
ls -lh
```

Output:

```
total 267M
-rw-rw-r-- 1 user user 267M Feb 10 15:00 uniprot_sprot.
↪fasta
```

View the database

Since this extracted database file is large, you can use the *head* command to view the first few lines of the file:

```
head -n 5 uniprot_sprot.fasta
```

Output:

```
>sp|Q6GZX4|001R_FRG3G Putative transcription factor 001R
↪OS=Frog virus 3 (isolate Goorha) OX=654924 GN=FV3-001R
↪PE=4 SV=1
```

(continues on next page)

(continued from previous page)

```
MAFSAEDVLKEYDRRRRMEALLLSLYYPNDRKLLDYKEWSPPRVQVECPKAPVEWNNPPS
EKGLIVGHFSGIKYKGEKAQASEVDVNKMCCWVSKFKDAMRRYQGIQTCKIPGKVLSDLD
AKIKAYNLTVEGVEGFVRYSRVTKQHVAAFLKELRHSKQYENVNLIHYILTDKRVDIQHL
EKDLVKDFKALVESAHRMRQGHMINVKYILYQLLKKHGHGPDGPDILTVKTGSKGVLYDD
```

Alternatively, you can use the *less* command to view it one page at a time:

```
less uniprot_sprot.fasta
```

If you would like to count the number of sequences in the database, you can use the `grep` command.

```
grep ">" -c uniprot_sprot.fasta
```

Output:

```
564277
```

The `-c` option of `grep`, counts the number of times the given search string (> in this case) occurs in the input file.

Note: A sequence in a FASTA format should start with the > character. Hence, counting the number of times it occurs gives the number of sequences in the file.

You can now proceed towards formatting the database.

Format the database — `makeblastdb`

The database needs to be formatted before it can be used in a BLAST search. You can format it using the `makeblastdb` command, which is part of the NCBI BLAST+ package.

The command has multiple options. Here is an example:

```
makeblastdb -in uniprot_sprot.fasta -parse_seqids \
-title "Swiss-Prot" -dbtype prot -out swissprot
```

Note: The \ character splits the long command into multiple lines.

Output:

```
Building a new DB, current time: 03/24/2021 15:12:50
New DB name:   /home/user/databases/swissprot
New DB title:  Swiss-Prot
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 564277 sequences in 47.
↪507 seconds.
```

What the options mean:

`-in` File name containing input sequences.

`-parse_seqids` Parse sequence identifiers from the input file. These will be displayed in search results.

`-title` A descriptive name for this database.

`-dbtype` The type of input sequences — acceptable values are `prot` (for protein) and `nucl` (for nucleotide) sequences.

`-out` The value here will be used to name the output files. This is also the name you will need to use for the database while doing a search (see *New DB Name*) in output.

When formatting is complete, you will notice the following files in the `databases` directory:

```
ls -lh
```

Output:

```
total 585M
-rw-rw-r-- 1 user user 100M Mar 24 15:13 swissprot.phr
-rw-rw-r-- 1 user user 4.4M Mar 24 15:13 swissprot.pin
-rw-rw-r-- 1 user user 2.2M Mar 24 15:13 swissprot.pog
-rw-rw-r-- 1 user user  18M Mar 24 15:13 swissprot.psd
-rw-rw-r-- 1 user user 411K Mar 24 15:13 swissprot.psi
-rw-rw-r-- 1 user user 195M Mar 24 15:13 swissprot.psq
-rw-rw-r-- 1 user user 267M Feb 10 15:00 uniprot_sprot.
↪fasta
```

4.5.6 Search database using query sequence

With the query sequence downloaded and the database downloaded and formatted, you can start performing a BLAST search.

First, change into the directory containing the query sequence:

```
cd ~/projects/sars-cov-2
```

Now run the `blastp` command using the query sequence and the complete path to the database:

```
blastp -query P0DTC2.fasta \
-db /home/user/databases/swissprot \
-out blastp-results.txt \
-outfmt "7 sacc stitle qlen slen pident"
```

What the options mean:

`-query` Path to the query sequence.

`-db` Complete path of the sequence database.

`-out` File to save results to.

`-outfmt` Format of the output file. This will use format option 7 (tab-delimited text) and include the following information:

- accession number and description of matching sequences (`sacc` and `stitle`),
- query and subject sequence lengths (`qlen` and `slen`)
- percentage identity of the match (`pident`).

When the database search is complete, you can open `blastp-results.txt` to view the results:

```
less -S blastp-results.txt
```

The `-S` option of the `less` command disables word-wrap.

Output:

```
# BLASTP 2.9.0+
# Query: sp|P0DTC2|SPIKE_SARS2 Spike glycoprotein
↪OS=Severe acute respiratory syndrome coronavirus 2
↪OX=2697049 GN=S PE=1 SV=1
# Database: /home/user/databases/swissprot
# Fields: subject acc., subject title, query length,
↪subject length, % identity
# 88 hits found
P0DTC2      Spike glycoprotein OS=Severe acute
↪respiratory syndrome coronavirus 2 OX=2697049 GN=S PE=1
↪SV=1 1273    1273    100.000
P59594      Spike glycoprotein OS=Severe acute
↪respiratory syndrome coronavirus OX=694009 GN=S PE=1
↪SV=1    1273    1255    76.038
Q3LZX1      Spike glycoprotein OS=Bat coronavirus HKU3
↪OX=442736 GN=S PE=3 SV=1     1273    1242    76.041
Q3I5J5      Spike glycoprotein OS=Bat coronavirus Rp3/
↪2004 OX=349344 GN=S PE=1 SV=1 1273    1241    75.334
Q0Q475      Spike glycoprotein OS=Bat coronavirus 279/
↪2005 OX=389167 GN=S PE=3 SV=1 1273    1241    74.745
...
```

4.6 Notes

4.6.1 Adding directories to PATH

The `PATH` variable stores the list of directories that will be searched to locate commands you type. If you have a *command* that is stored in a directory that is not on this list, you will need to add it to `PATH`.

You can display the current value of `PATH` using the *echo* command:

```
echo $PATH
```

Output:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
↪bin:/usr/games:/usr/local/games:/snap/bin
```

To modify `PATH` and add a directory, you can follow the steps below. This example adds `/home/user/.local/bin` to `PATH`.

Attention: *You will need to be careful while editing PATH.*

You will also need to ensure that it includes `$PATH` in the list. Otherwise, you will not be able to access any commands!

1. Open `$HOME/.bashrc` in a *text editor*.
2. Add the following lines at the end of the file. Add the directory you wish to add separated by a colon. The list should end in `$PATH`.

```
PATH=/home/user/.local/bin:$PATH
export PATH
```

Note: If a line with `PATH` exists already in the file, update it instead.

3. Verify the change using `echo`:

   ```
   echo $PATH
   ```

   Output:

   ```
   /home/user/.local/bin:/usr/local/sbin:/usr/local/
   ↪bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
   ↪local/games:/snap/bin
   ```

When you log in again or open a new terminal session, commands in `/home/user/.local/bin` will be accessible.

4.6.2 Dealing with spaces in file names

When you create a new file or directory, it is a good idea to *not use spaces* in the file or directory name. This is especially useful while working with these files or directories in the command-line. Instead, you can use an underscore (_) or hyphen (-) to separate words in file names.

If you do have to work with file or directory names containing spaces, you can use quotes around the file or directory name.

For example:

```
mkdir "Sample Data"
```

To change into the `Sample Data` directory, use:

```
cd "Sample Data"
```

Print current directory:

```
pwd
```

Output:

```
/home/user/Sample Data
```

This would have been simpler if the directory name is `sample-data`. You will not need to use quotes in that case.

5 Getting started with Galaxy

The Galaxy[32] project enables researchers to perform computational analyses using a web interface.

You can use the service online at http://usegalaxy.org/ or any of the other available public servers[33].

The software powering the servers itself, is available under an open-source licence. You can download it and set up a local instance on your computer.

—

[32] https://galaxyproject.org/
[33] https://galaxyproject.org/use/

5.1 Why use Galaxy?

These are *some* advantages of using Galaxy.

You can:

Upload your data — and perform analysis using a *web interface*.

Use included tools — or install additional ones from the Galaxy tool shed (needs a *local instance*).

Use history — functionality to keeps track of the jobs[1] you run. This gives you the possibility to run them again at a later time (reproducibility).

Create workflows — by combining multiple tools. The output of one tool can be used as input for another.

Share your data — with other users. This includes histories, workflows etc.,

[1] Commands run with additional arguments, if any, along with input and output file paths.

5.1.1 What can you use Galaxy for?

These are some tasks you can accomplish using Galaxy:

- Get data from various sources (databases)
- Manipulate text data
- Convert between file formats
- Statistics
- Sort and filter data

Note: The *exercise* in *Using the Linux command line* chapter, can be reproduced using Galaxy, without the need to use the command line or install software.

You only need your input data — query sequence and the protein sequence database.

5.1.2 Why run Galaxy on your computer?

Running Galaxy on your computer i.e., a *local instance*, has some advantages:

Data storage is local — on your computer, whereas, if you use a public instance, it will be stored on a server, possibly unencrypted[2].

Access more tools — that are not installed on public servers by installing them on your local instance.

CPU and storage limits — are dependent on your computer, whereas on public servers, there are limits on CPU and storage usage per user[2].

[2] This is the case on https://usegalaxy.org — the main Galaxy instance. For more information, please read the https://galaxyproject.org/main/#user-data-and-job-quotas.

5.2 Running Galaxy on your computer

To run a local instance of Galaxy on your computer, you will need to:

1. Check if your computer meets all requirements
2. Get the latest release of Galaxy
3. Create a Python virtual environment
4. Start Galaxy

At this stage, you will be able to open Galaxy's web interface in a browser and start using the tools included.

Note: To make use of all the features that Galaxy offers like installing additional tools, you will also need to do the following:

5. Register a user account
6. Grant administrator privileges for user

5.2.1 Check system requirements

Before running Galaxy, you will need to:

1. Install Git and Python virtualenv module
2. Check if there is enough free disk space
3. Ensure that a Conda environment is not active

Install Git and Python virtualenv module

Git is necessary for getting the latest version of Galaxy's source code from GitHub. Python virtualenv module is necessary to set up an isolated virtual environment for running Galaxy.

To install them, use *Software Manager* or *apt*, and search and install the following packages:

1. `git`
2. `python3-venv`

Check free disk space

You will need *at least* 5 GB of free disk space for the initial setup. More space will be needed for your analyses and for the tools you might install from the Galaxy tool shed.

To check available disk space, you can use the Disk Usage Analyzer application or the *df* command.

Ensure a Conda environment is not active

Note: If you are not using Conda, you can ignore this requirement and proceed to the next step:

-> *Getting the latest release of Galaxy*

If you are using Conda, please make sure an environment is not active before the *Starting Galaxy* step.

You can deactivate an active Conda environment using the command:

```
conda deactivate
```

Although it is possible to install Galaxy using an existing Conda installation, it is simpler to create and use a virtual environment using Python's venv module.

—

With all the requirements satisfied, you can proceed towards getting the latest release of Galaxy using Git.

5.2.2 Getting the latest release of Galaxy

Create a directory to set up Galaxy and change into it:

```
mkdir ~/programs
cd ~/programs
```

Here ~ is a shortcut for your home directory.

Note: This directory is not specific for Galaxy.

You can use it for installing or storing other programs too.

Get the latest release of Galaxy from the project's repository on GitHub. The latest release at the time of this writing is 21.01.

You can get (or clone) the repository using the `git clone` command:

```
git clone -b release_21.01 https://github.com/
↪galaxyproject/galaxy.git
```

The `-b` option of `git clone`, checks out the specified Git branch — `release_21.01` in this case.

Output:

```
Cloning into 'galaxy'...
remote: Enumerating objects: 364, done.
remote: Counting objects: 100% (364/364), done.
```

(continues on next page)

(continued from previous page)

```
remote: Compressing objects: 100% (232/232), done.
remote: Total 497693 (delta 184), reused 237 (delta 131),
↪pack-reused 497329
Receiving objects: 100% (497693/497693), 520.44 MiB | 3.
↪89 MiB/s, done.
Resolving deltas: 100% (391502/391502), done.
Updating files: 100% (5826/5826), done.
```

This will take some time depending on your network connection. Once complete, you will find a directory named `galaxy` in the current directory.

Next, you will need to create a Python virtual environment.

5.2.3 Creating a Python virtual environment

In this step, you will use Python 3 `venv` module to create an isolated virtual environment for Galaxy.

Note: For a detailed guide on creating and using virtual environments, read:

-> *Python virtual environments*

To create the virtual environment:

1. Change into the `galaxy` directory:

   ```
   cd galaxy
   ```

2. Create virtual environment using `.venv` as the name.

   ```
   python3 -m venv .venv
   ```

   Attention: The dot (`.`) in `.venv`, is important.

3. Activate the virtual environment:

   ```
   source .venv/bin/activate
   ```

   Your shell prompt will now change to include `(.venv)`:

```
(.venv) user@cookbook:~/programs/galaxy$
```

4. Install or upgrade Python build tools:

   ```
   pip3 install -U pip setuptools wheel
   ```

   This step is necessary, so additional packages will build and install without any issues.

5. Finally, exit the virtual environment:

   ```
   deactivate
   ```

In the *next step*, the start-up script will detect this environment and use it to install all *dependencies* necessary for running Galaxy.

5.2.4 Starting Galaxy

While still in the `galaxy` directory, run the start-up shell script using the command:

```
sh run.sh
```

Note: When run for the first time, this script will take some time to complete.

Subsequent runs will be faster.

Output:

```
Activating virtualenv at .venv
Requirement already satisfied: pip≥8.1 in ./.venv/lib/
↪python3.8/site-packages (20.0.2)
...
```

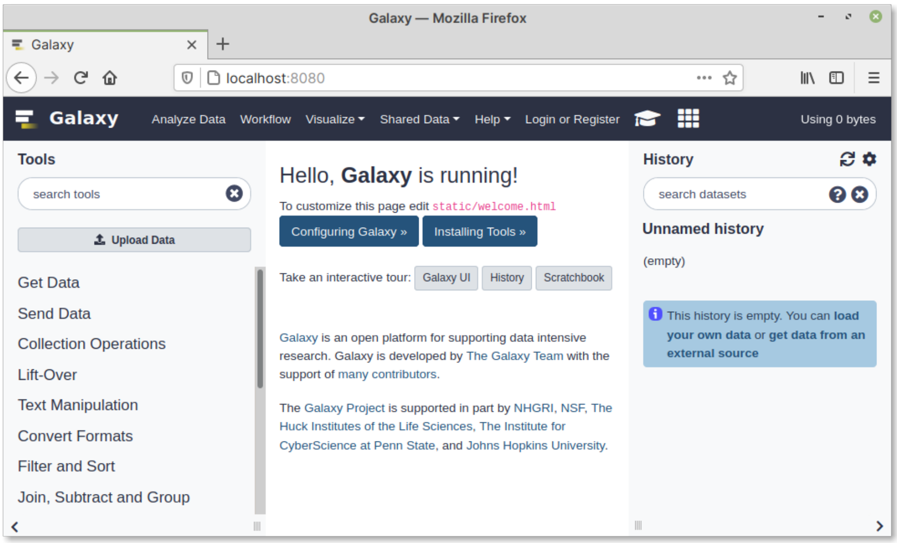
When you see messages like the following in the terminal session, Galaxy is ready to use:

```
Starting server in PID 11466.
serving on http://localhost:8080
```

Leave the terminal window open.

Open a web browser window and type http://localhost:8080 in the address bar.

You will see the home page of your local Galaxy instance (Fig. 85).



Hello, **Galaxy** is running!

To customize this page edit `static/welcome.html`

Take an interactive tour:

Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

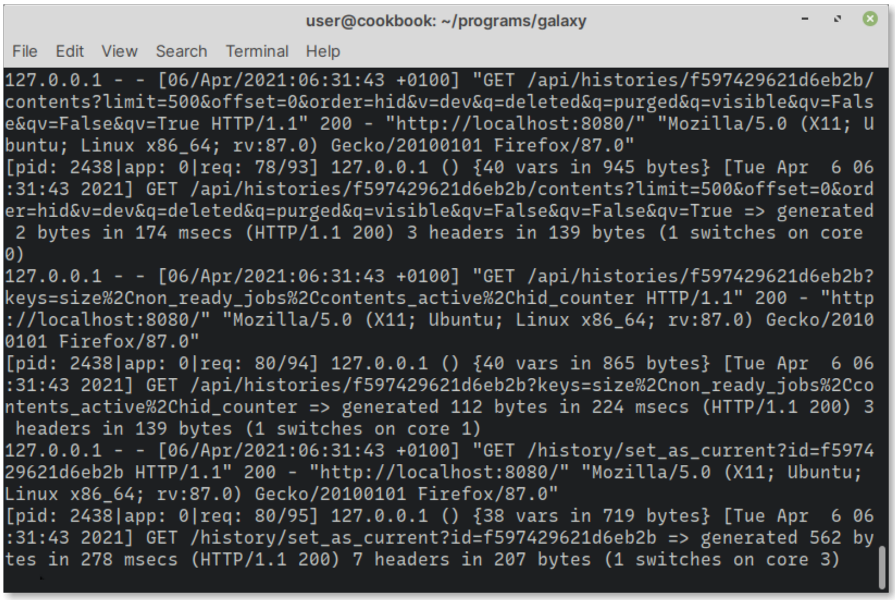
This history is empty. You can **load your own data** or **get data from an external source**

Fig. 85: Home page of your local Galaxy instance

You can start exploring the tools available or even upload data and start using them. However, it is a good idea to *register a user account* now, so you do not lose data, when you clear browser history.

5.2.5 Stopping Galaxy

To stop Galaxy, open the terminal window where you ran the `sh run.sh` command (Fig. 86).

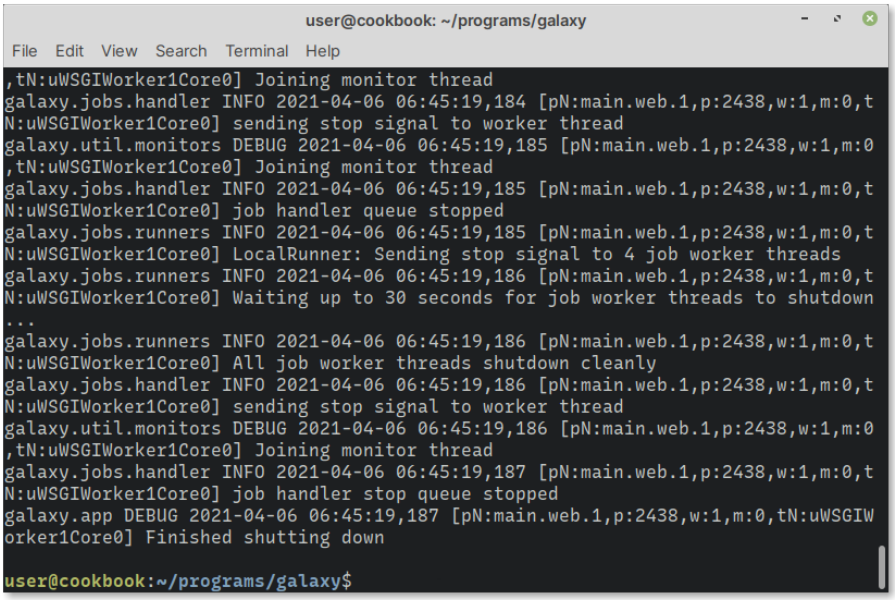

user@cookbook: ~/programs/galaxy
File Edit View Search Terminal Help

```
127.0.0.1 - - [06/Apr/2021:06:31:43 +0100] "GET /api/histories/f597429621d6eb2b/
contents?limit=500&offset=0&order=hid&v=dev&q=deleted&q=purged&q=visible&qv=Fals
e&qv=False&qv=True HTTP/1.1" 200 - "http://localhost:8080/" "Mozilla/5.0 (X11; U
buntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0"
[pid: 2438|app: 0|req: 78/93] 127.0.0.1 () {40 vars in 945 bytes} [Tue Apr  6 06
:31:43 2021] GET /api/histories/f597429621d6eb2b/contents?limit=500&offset=0&ord
er=hid&v=dev&q=deleted&q=purged&q=visible&qv=False&qv=False&qv=True => generated
 2 bytes in 174 msecs (HTTP/1.1 200) 3 headers in 139 bytes (1 switches on core
0)
127.0.0.1 - - [06/Apr/2021:06:31:43 +0100] "GET /api/histories/f597429621d6eb2b?
keys=size%2Cnon_ready_jobs%2Ccontents_active%2Chid_counter HTTP/1.1" 200 - "http
://localhost:8080/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/2010
0101 Firefox/87.0"
[pid: 2438|app: 0|req: 80/94] 127.0.0.1 () {40 vars in 865 bytes} [Tue Apr  6 06
:31:43 2021] GET /api/histories/f597429621d6eb2b?keys=size%2Cnon_ready_jobs%2Cco
ntents_active%2Chid_counter => generated 112 bytes in 224 msecs (HTTP/1.1 200) 3
 headers in 139 bytes (1 switches on core 1)
127.0.0.1 - - [06/Apr/2021:06:31:43 +0100] "GET /history/set_as_current?id=f5974
29621d6eb2b HTTP/1.1" 200 - "http://localhost:8080/" "Mozilla/5.0 (X11; Ubuntu;
Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0"
[pid: 2438|app: 0|req: 80/95] 127.0.0.1 () {38 vars in 719 bytes} [Tue Apr  6 06
:31:43 2021] GET /history/set_as_current?id=f597429621d6eb2b => generated 562 by
tes in 278 msecs (HTTP/1.1 200) 7 headers in 207 bytes (1 switches on core 3)
```

Fig. 86: Terminal window running Galaxy

Use the `Ctrl + c` keyboard shortcut to stop Galaxy i.e., press and hold `Ctrl` key and then press the `c` key.

When Galaxy has stopped running, you will see a message `Finished shutting down` (Fig. 87) in the terminal.


user@cookbook: ~/programs/galaxy
File Edit View Search Terminal Help

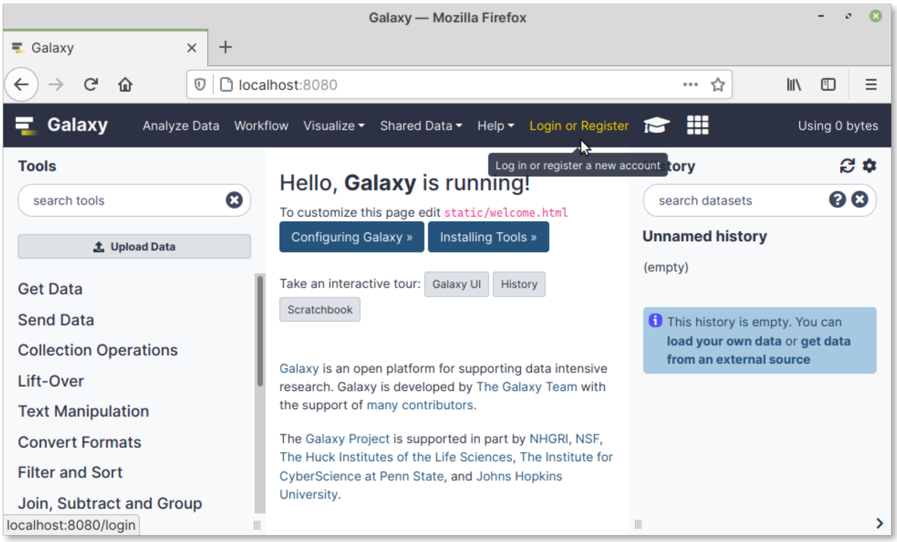
```
,tN:uWSGIWorker1Core0] Joining monitor thread
galaxy.jobs.handler INFO 2021-04-06 06:45:19,184 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] sending stop signal to worker thread
galaxy.util.monitors DEBUG 2021-04-06 06:45:19,185 [pN:main.web.1,p:2438,w:1,m:0
,tN:uWSGIWorker1Core0] Joining monitor thread
galaxy.jobs.handler INFO 2021-04-06 06:45:19,185 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] job handler queue stopped
galaxy.jobs.runners INFO 2021-04-06 06:45:19,185 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] LocalRunner: Sending stop signal to 4 job worker threads
galaxy.jobs.runners INFO 2021-04-06 06:45:19,186 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] Waiting up to 30 seconds for job worker threads to shutdown
...
galaxy.jobs.runners INFO 2021-04-06 06:45:19,186 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] All job worker threads shutdown cleanly
galaxy.jobs.handler INFO 2021-04-06 06:45:19,186 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] sending stop signal to worker thread
galaxy.util.monitors DEBUG 2021-04-06 06:45:19,186 [pN:main.web.1,p:2438,w:1,m:0
,tN:uWSGIWorker1Core0] Joining monitor thread
galaxy.jobs.handler INFO 2021-04-06 06:45:19,187 [pN:main.web.1,p:2438,w:1,m:0,t
N:uWSGIWorker1Core0] job handler stop queue stopped
galaxy.app DEBUG 2021-04-06 06:45:19,187 [pN:main.web.1,p:2438,w:1,m:0,tN:uWSGIW
orker1Core0] Finished shutting down

user@cookbook:~/programs/galaxy$
```

Fig. 87: Galaxy has finished shutting down

You can now close the terminal window.

5.3 Registering a user account

On the Galaxy home page, click on the Login or Register link in the navigation bar (Fig. 88).



Hello, **Galaxy** is running!

To customize this page edit `static/welcome.html`

Take an interactive tour:

Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

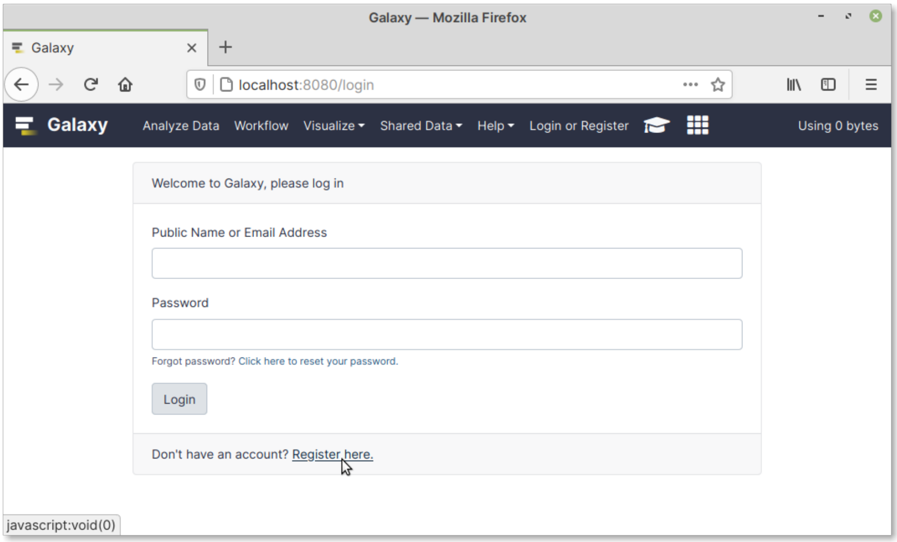
The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

This history is empty. You can **load your own data** or **get data from an external source**

Fig. 88: Click on the Login or Register link

You will be taken to the login page.

Click on the Register here link at the bottom of the page (Fig. 89).



Welcome to Galaxy, please log in

Public Name or Email Address

Password

Forgot password? Click here to reset your password.

Login

Don't have an account? Register here.

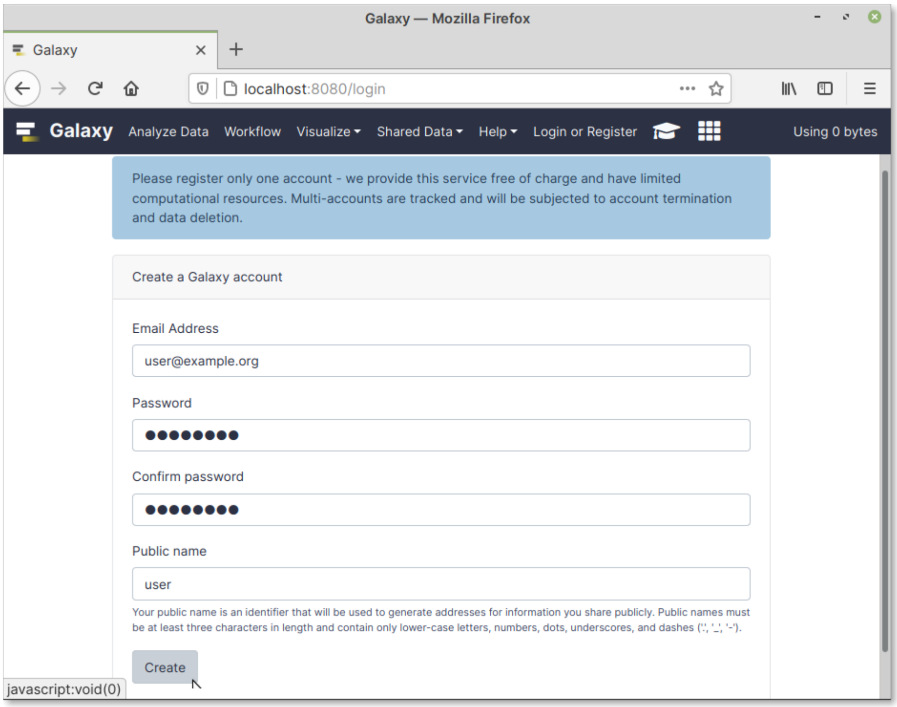
javascript:void(0)

Fig. 89: Click on the Register here link

On the Create a Galaxy account page, type in your email address, a password and its confirmation, and a public name (Fig. 90).

Note: This user account is not related to your accounts on usegalaxy.org or other Galaxy servers.

It exists only in your local instance of Galaxy. You can also ignore the notice at the top of the page, which does not apply to local instances.



Please register only one account - we provide this service free of charge and have limited computational resources. Multi-accounts are tracked and will be subjected to account termination and data deletion.

Create a Galaxy account

Email Address

user@example.org

Password

●●●●●●●●

Confirm password

●●●●●●●●

Public name

user

Your public name is an identifier that will be used to generate addresses for information you share publicly. Public names must be at least three characters in length and contain only lower-case letters, numbers, dots, underscores, and dashes ('.', '_', '-').

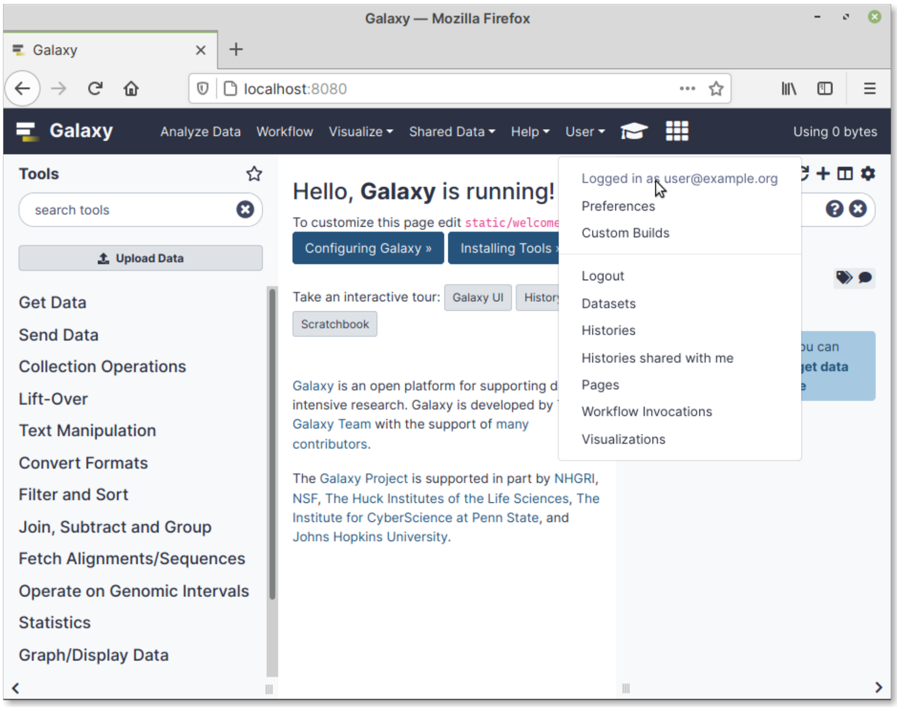
Fig. 90: User account creation page

Click on the Create button to proceed.

If account creation was successful, you will be logged in and redirected to the home page of your Galaxy instance.

5.3.1 Managing your account

You can manage your account by clicking on your username in the navigation bar (Fig. 91).



Hello, Galaxy is running!

To customize this page edit static/welcome

Galaxy is an open platform for supporting d intensive research. Galaxy is developed by Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

Fig. 91: Logged in as user

If you will be installing additional tools from Galaxy tool shed, this user account will need administrator privileges. You can configure that in the *next step*.

5.4 Granting admin privileges for user

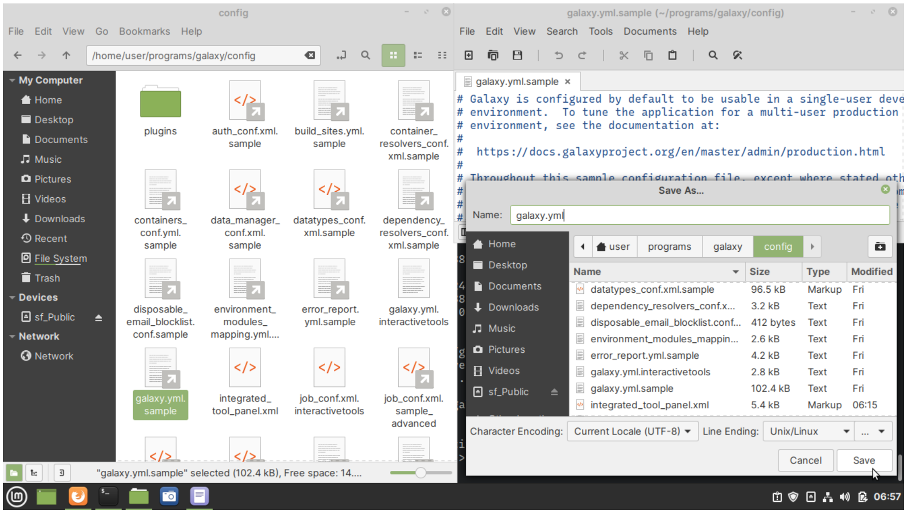
To grant administrator privileges for a user, you will need to create a configuration file for Galaxy and add the user's email address to the list of administrators, and then restart Galaxy for changes to take effect.

5.4.1 Creating a configuration file

A sample configuration file — `galaxy.yml.sample`, is included with Galaxy.

To create a configuration file using the sample file as template, follow the steps below (see Fig. 92):

1. Navigate to the `$HOME/programs/galaxy/config` directory using Files
2. *Double-click* on `galaxy.yml.sample` to open it in Text Editor
3. Save it as `galaxy.yml` in the same directory


config
File Edit View Go Bookmarks Help
/home/user/programs/galaxy/config
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Devices
sf_Public
Network
Network
plugins
auth_conf.xml.sample
build_sites.yml.sample
container_resolvers_conf.xml.sample
containers_conf.yml.sample
data_manager_conf.xml.sample
datatypes_conf.xml.sample
dependency_resolvers_conf.xml.sample
disposable_email_blocklist.conf.sample
environment_modules_mapping.yml....
error_report.yml.sample
galaxy.yml.interactivetools
galaxy.yml.sample
integrated_tool_panel.xml
job_conf.xml.interactivetools
job_conf.xml.sample_advanced
"galaxy.yml.sample" selected (102.4 kB), Free space: 14....
galaxy.yml.sample (~/programs/galaxy/config)
File Edit View Search Tools Documents Help
galaxy.yml.sample
Save As...
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
sf_Public
user
programs
galaxy
config
Cancel
Save
06:57

```
# Galaxy is configured by default to be usable in a single-user deve
# environment.  To tune the application for a multi-user production
# environment, see the documentation at:
#
#  https://docs.galaxyproject.org/en/master/admin/production.html
#
# Throughout this sample configuration file, except where stated oth
```

Name: galaxy.yml

Name	Size	Type	Modified
datatypes_conf.xml.sample	96.5 kB	Markup	Fri
dependency_resolvers_conf.x...	3.2 kB	Text	Fri
disposable_email_blocklist.conf...	412 bytes	Text	Fri
environment_modules_mappin...	2.6 kB	Text	Fri
error_report.yml.sample	4.2 kB	Text	Fri
galaxy.yml.interactivetools	2.8 kB	Text	Fri
galaxy.yml.sample	102.4 kB	Text	Fri
integrated_tool_panel.xml	5.4 kB	Markup	06:15

Character Encoding: Current Locale (UTF-8) Line Ending: Unix/Linux

Fig. 92: Copying sample configuration file of Galaxy

Note: Alternatively, you can use the *cp* command in a terminal to copy the file.

Assuming you are in the `$HOME/programs/galaxy` directory, do:

```
cp config/galaxy.yml.sample config/galaxy.yml
```

5.4.2 Adding user to list of admin users

With `galaxy.yml` open in Text Editor, search for the `admin_users` setting.

You will find the following section:

```
# Administrative users - set this to a comma-separated
# list of valid Galaxy users (email addresses).  These
# users will have access to the Admin section of the
# server, and will have access to create users,
# groups, roles, libraries, and more.  For more
# information, see: https://galaxyproject.org/admin/
#admin_users: null
```

Uncomment `admin_users` by removing the # symbol at the beginning.

Replace `null` with the email address of the user account.

For example:

```
admin_users: user@example.org
```

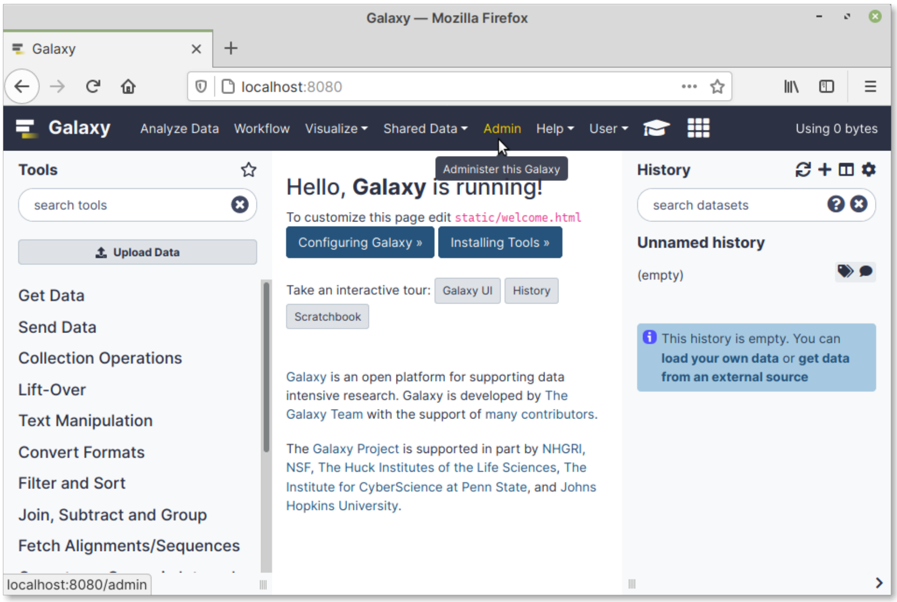
Save the file.

5.4.3 Restarting Galaxy

Stop Galaxy, if it is running and then *start it again*.

5.4.4 Verifying admin access

When you log in again, you will see an Admin link in the navigation bar (Fig. 93).



Hello, Galaxy is running!

To customize this page edit static/welcome.html

Configuring Galaxy » Installing Tools »

Take an interactive tour: Galaxy UI History Scratchbook

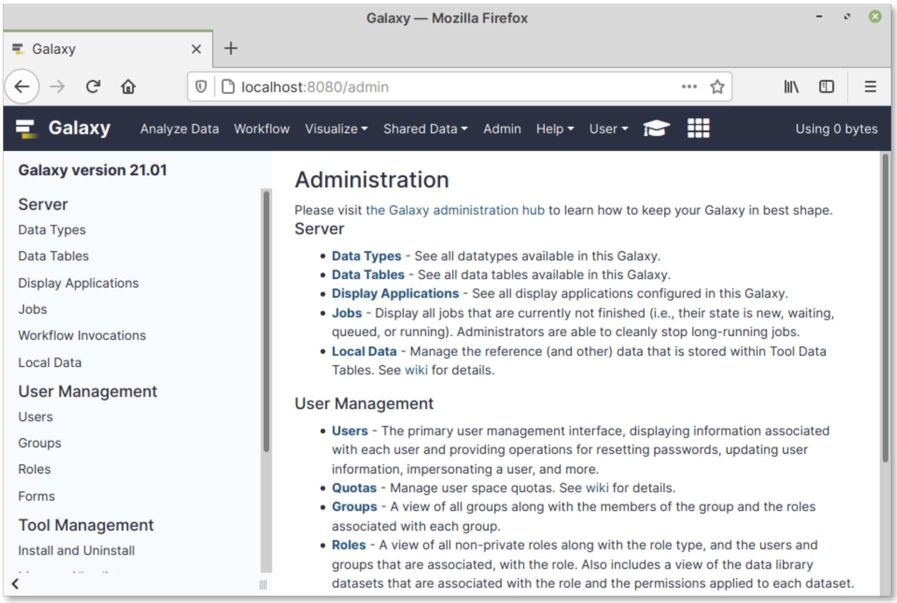
Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

This history is empty. You can load your own data or get data from an external source

Fig. 93: Admin link in navigation bar

If you click on Admin, you will be taken to the administration page (Fig. 94).



Administration

Please visit the Galaxy administration hub to learn how to keep your Galaxy in best shape.

Server

- **Data Types** - See all datatypes available in this Galaxy.
- **Data Tables** - See all data tables available in this Galaxy.
- **Display Applications** - See all display applications configured in this Galaxy.
- **Jobs** - Display all jobs that are currently not finished (i.e., their state is new, waiting, queued, or running). Administrators are able to cleanly stop long-running jobs.
- **Local Data** - Manage the reference (and other) data that is stored within Tool Data Tables. See wiki for details.

User Management

- **Users** - The primary user management interface, displaying information associated with each user and providing operations for resetting passwords, updating user information, impersonating a user, and more.
- **Quotas** - Manage user space quotas. See wiki for details.
- **Groups** - A view of all groups along with the members of the group and the roles associated with each group.
- **Roles** - A view of all non-private roles along with the role type, and the users and groups that are associated, with the role. Also includes a view of the data library datasets that are associated with the role and the permissions applied to each dataset.

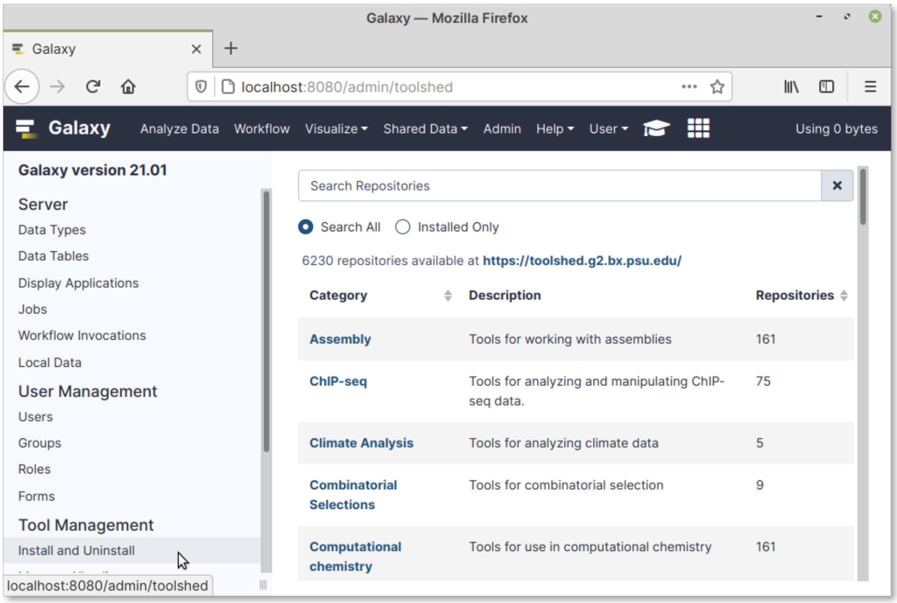
Fig. 94: Galaxy administration page

Installing tools

To install new tools, click on the following link:

Tool Management $\longrightarrow$ Install and Uninstall

You will be taken to the tool management page (Fig. 95).


Galaxy — Mozilla Firefox

localhost:8080/admin/toolshed

Galaxy version 21.01

Server
Data Types
Data Tables
Display Applications
Jobs
Workflow Invocations
Local Data

User Management
Users
Groups
Roles
Forms

Tool Management
Install and Uninstall

Search Repositories

Search All [✓] Installed Only

6230 repositories available at https://toolshed.g2.bx.psu.edu/

Category	Description	Repositories
Assembly	Tools for working with assemblies	161
ChIP-seq	Tools for analyzing and manipulating ChIP-seq data.	75
Climate Analysis	Tools for analyzing climate data	5
Combinatorial Selections	Tools for combinatorial selection	9
Computational chemistry	Tools for use in computational chemistry	161

Fig. 95: Tool management page

From here, you can search and install tools from the Galaxy main tool shed.

6 Documentation

Installing and software for managing references and bibliographies, taking notes etc.,

6.1 Managing references using Zotero

Zotero is a free and open-source software for collecting and organizing scientific research articles and other publications.

Articles can be organized using collections and tags, and can be included as citations in the documents you create.

6.1.1 Installing Zotero

You can search and install Zotero using *Software Manager*.

To save articles from the web and to insert citations in documents, additional add-ons (or extensions) are necessary. You can follow the steps below to install them.

6.1.2 Collecting references from the web

To collect references using a web browser, you will need to install Zotero Connector — a browser add-on.

Installing Zotero Connector

To install Zotero Connector in Firefox:

1. Open Zotero from applications menu.

   A browser window will open, and you will be taken to the add-on installation page[34].

   Click on the Install button (Fig. 96).

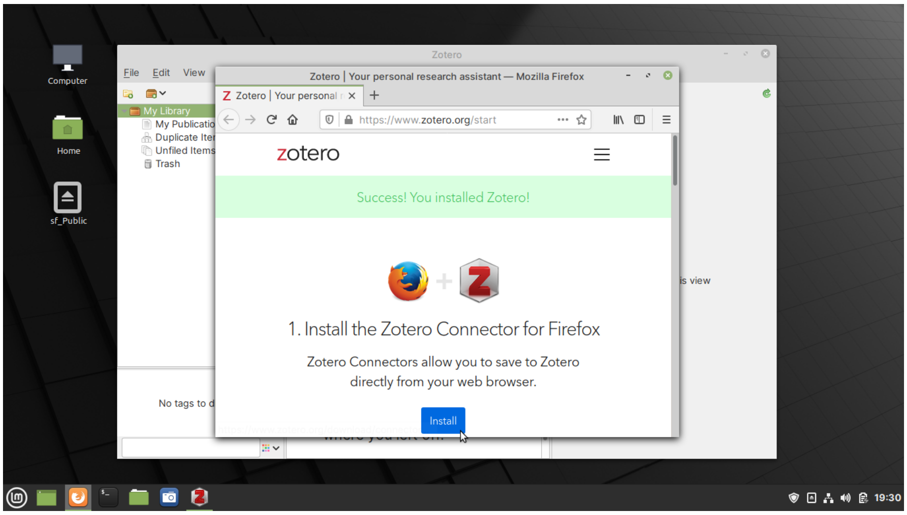

Zotero | Your personal research assistant — Mozilla Firefox
https://www.zotero.org/start
zotero
Success! You installed Zotero!
1. Install the Zotero Connector for Firefox
Zotero Connectors allow you to save to Zotero directly from your web browser.
Install
Computer
Home
sf_Public
My Library
19:30

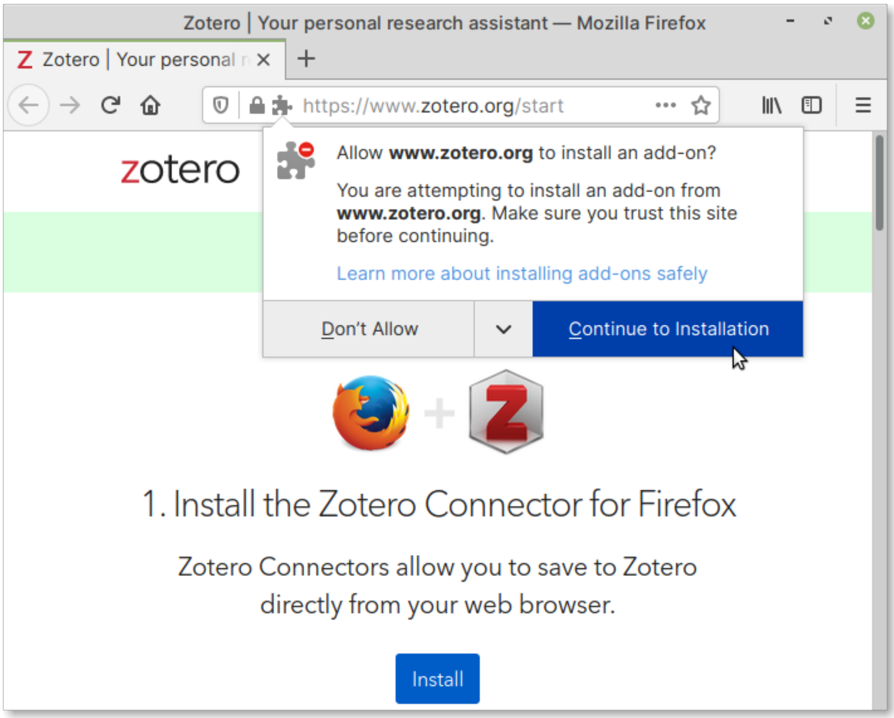
Fig. 96: Zotero add-on installation page

2. You will now see a dialog requesting if you would like to

[34] https://www.zotero.org/start

continue installing this add-on from www.zotero.org.

Click on the Continue to Installation button to proceed (Fig. 97).


Zotero | Your personal research assistant — Mozilla Firefox

https://www.zotero.org/start

Allow **www.zotero.org** to install an add-on?

You are attempting to install an add-on from **www.zotero.org**. Make sure you trust this site before continuing.

Learn more about installing add-ons safely

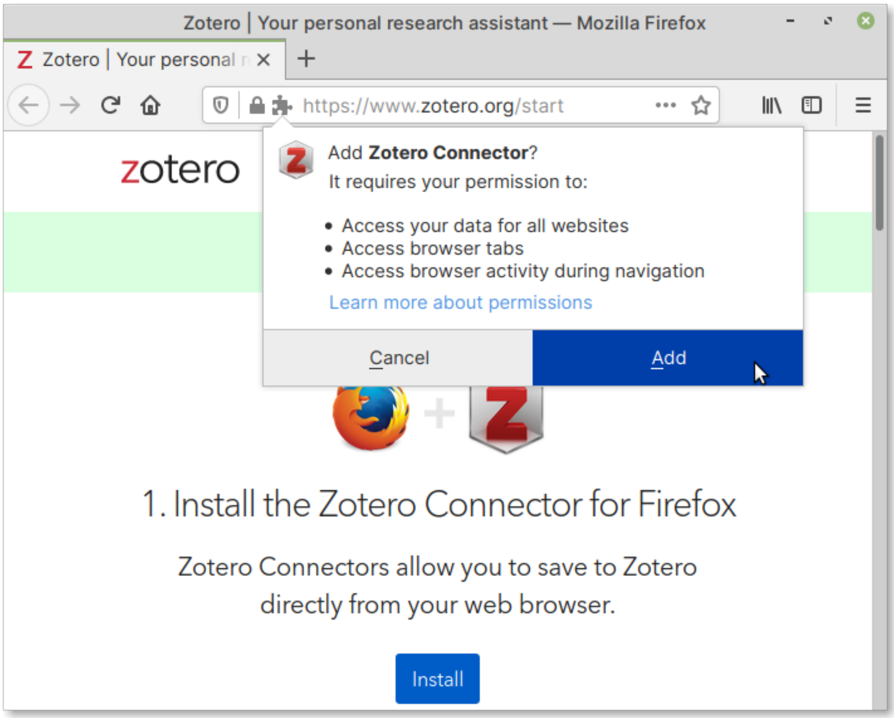
1. Install the Zotero Connector for Firefox

Zotero Connectors allow you to save to Zotero directly from your web browser.

Fig. 97: Click Continue to Installation to proceed

3. You will now see a dialog with information on the permissions required by this add-on.

   Click on the Add button to proceed (Fig. 98).


Zotero | Your personal research assistant — Mozilla Firefox
Zotero | Your personal r
https://www.zotero.org/start
zotero

Add **Zotero Connector**?
It requires your permission to:

- Access your data for all websites
- Access browser tabs
- Access browser activity during navigation

Learn more about permissions

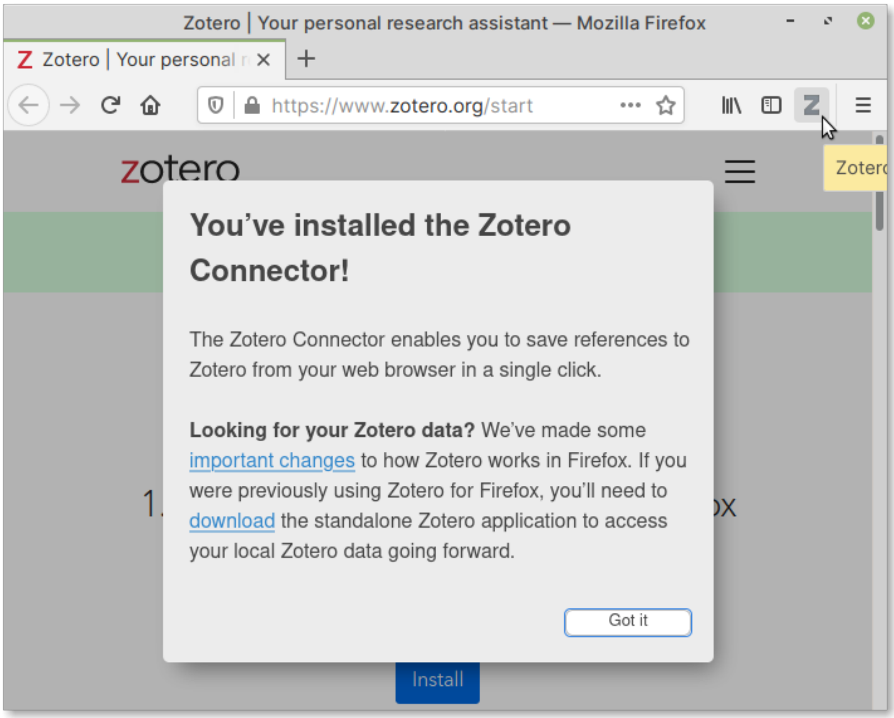
1. Install the Zotero Connector for Firefox

Zotero Connectors allow you to save to Zotero directly from your web browser.

Fig. 98: Click on the Add button to confirm installation

Installation will now proceed.

When complete, you will see a Zotero icon in the browser toolbar (Fig. 99).


Zotero | Your personal research assistant — Mozilla Firefox
Zotero | Your personal r
https://www.zotero.org/start
zotero
Zoterc
1.
ox
Install

You've installed the Zotero Connector!

The Zotero Connector enables you to save references to Zotero from your web browser in a single click.

Looking for your Zotero data? We've made some important changes to how Zotero works in Firefox. If you were previously using Zotero for Firefox, you'll need to download the standalone Zotero application to access your local Zotero data going forward.

Got it

Fig. 99: Zotero Connector installed successfully

You can now start saving articles from the browser to your Zotero library.

6.1.3 Integrating Zotero in LibreOffice

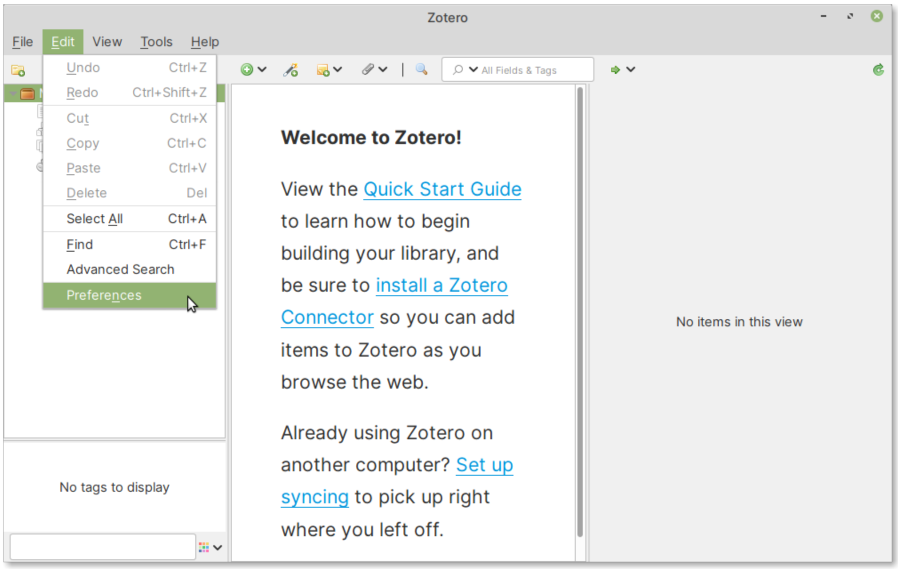
To be able to insert citations and bibliography in your documents, you will need to install the Word Processor Add-in.

Installing Word Processor add-in

To install the add-in in LibreOffice:

1. From the main menu bar of Zotero, select

   Edit → Preferences (Fig. 100).



Welcome to Zotero!

View the Quick Start Guide to learn how to begin building your library, and be sure to install a Zotero Connector so you can add items to Zotero as you browse the web.

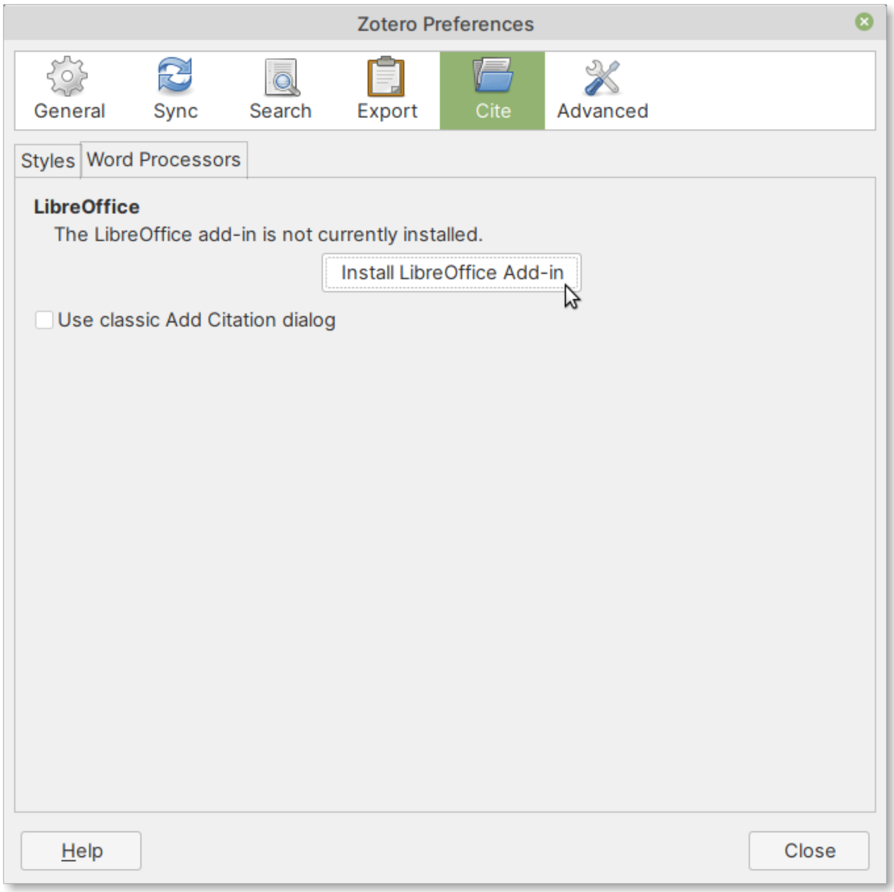
Already using Zotero on another computer? Set up syncing to pick up right where you left off.

Fig. 100: Viewing Zotero preferences

2. In the Zotero Preferences dialog window that opens, click on the Cite tab.

   Now, click on the Word Processors tab to select it.

Click on the Install LibreOffice Add-in button (Fig. 101).



LibreOffice

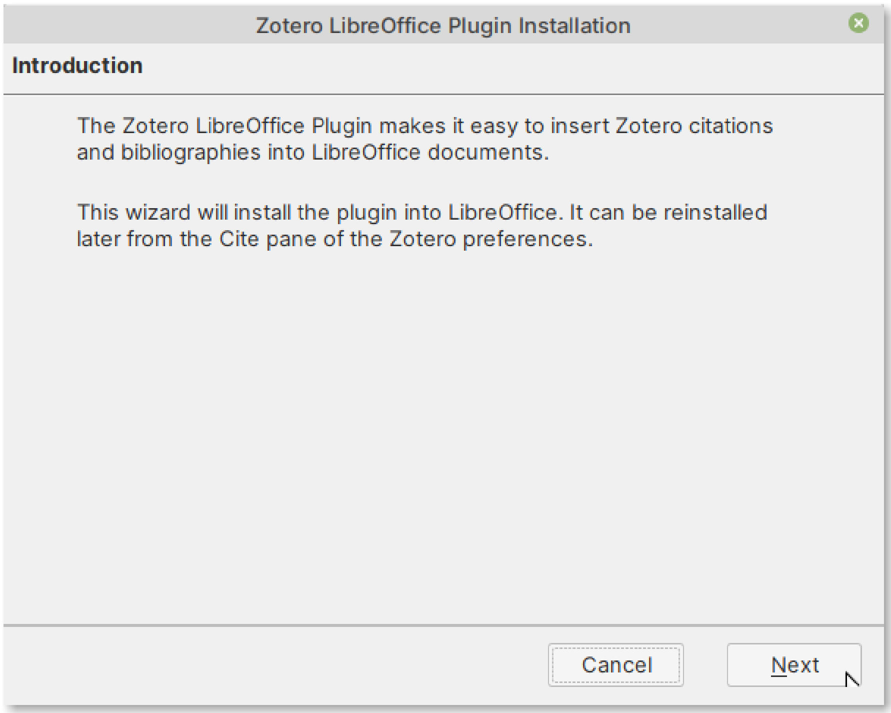
The LibreOffice add-in is not currently installed.

Install LibreOffice Add-in

☐ Use classic Add Citation dialog

Fig. 101: Installing LibreOffice Add-in from Zotero preferences

3. The Zotero LibreOffice Plugin Installation wizard will now open (Fig. 102)


Zotero LibreOffice Plugin Installation

Introduction

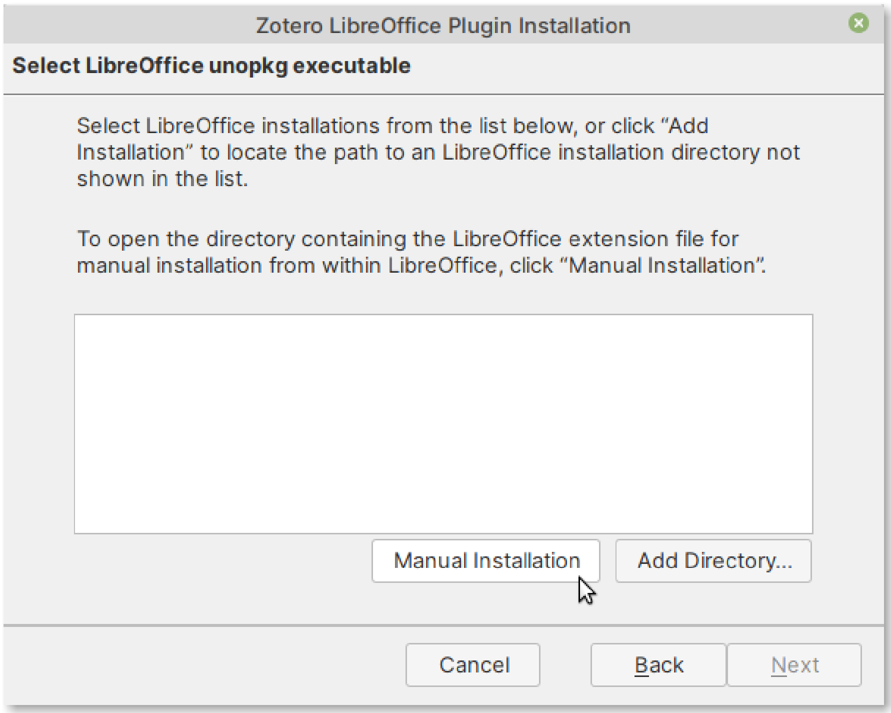
The Zotero LibreOffice Plugin makes it easy to insert Zotero citations and bibliographies into LibreOffice documents.

This wizard will install the plugin into LibreOffice. It can be reinstalled later from the Cite pane of the Zotero preferences.

Fig. 102: Zotero LibreOffice Plugin Installation wizard

Click on the Next button to proceed.

4. Click on Manual Installation in the select LibreOffice installations dialog window (Fig. 103).


Zotero LibreOffice Plugin Installation

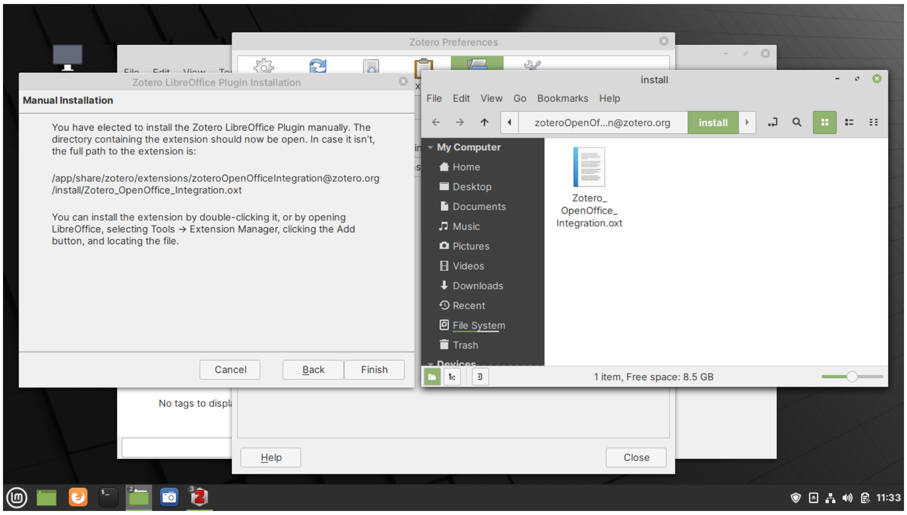
Select LibreOffice unopkg executable

Select LibreOffice installations from the list below, or click "Add Installation" to locate the path to an LibreOffice installation directory not shown in the list.

To open the directory containing the LibreOffice extension file for manual installation from within LibreOffice, click "Manual Installation".

Fig. 103: Click on the Manual Installation button

5. A file browser window will open, with the directory set to the location of the add-in file — Zotero_OpenOffice_Integration.oxt (Fig. 104).



Manual Installation

You have elected to install the Zotero LibreOffice Plugin manually. The directory containing the extension should now be open. In case it isn't, the full path to the extension is:

/app/share/zotero/extensions/zoteroOpenOfficeIntegration@zotero.org /install/Zotero_OpenOffice_Integration.oxt

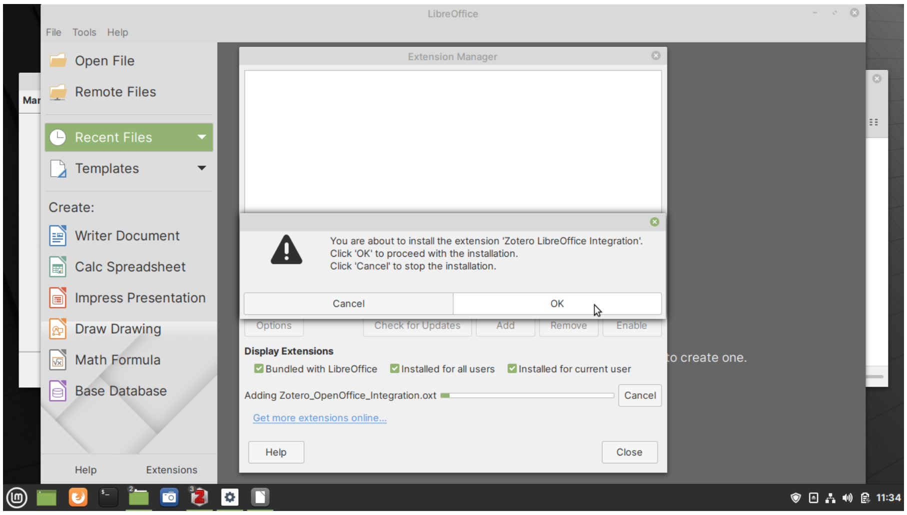
You can install the extension by double-clicking it, or by opening LibreOffice, selecting Tools → Extension Manager, clicking the Add button, and locating the file.

Fig. 104: File browser open with location of the Add-in file

6. *Double-click* on `Zotero_OpenOffice_Integration.oxt`.

   This will open LibreOffice Extension Manager, with a prompt requesting if you would like to install the add-in.

   Click on the `OK` button to confirm (Fig. 105).



You are about to install the extension 'Zotero LibreOffice Integration'.
Click 'OK' to proceed with the installation.
Click 'Cancel' to stop the installation.

Adding Zotero_OpenOffice_Integration.oxt

Fig. 105: Click on the OK button in LibreOffice Extension Manager

The add-in will now be installed (Fig. 106).

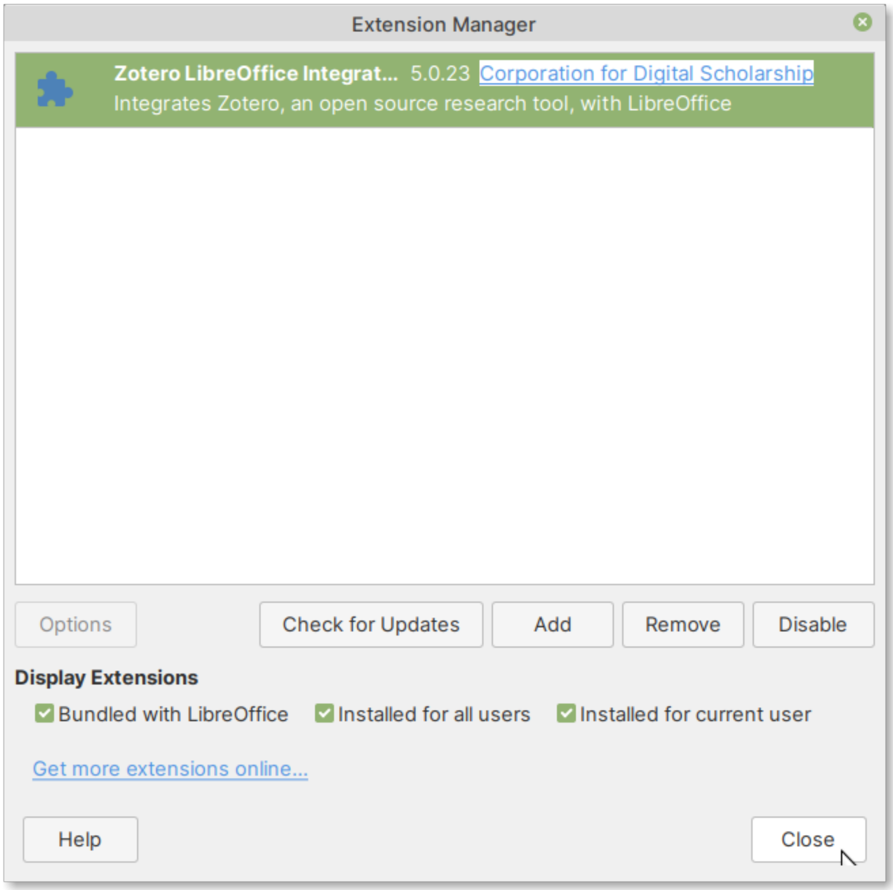

Extension Manager
Zotero LibreOffice Integrat... 5.0.23 Corporation for Digital Scholarship
Integrates Zotero, an open source research tool, with LibreOffice
Options
Check for Updates
Add
Remove
Disable
Display Extensions
✓ Bundled with LibreOffice
✓ Installed for all users
✓ Installed for current user
Get more extensions online...
Help
Close

Fig. 106: Zotero LibreOffice Add-in installed successfully

If you *restart* LibreOffice, you will notice the Zotero toolbar in the main window (Fig. 107).

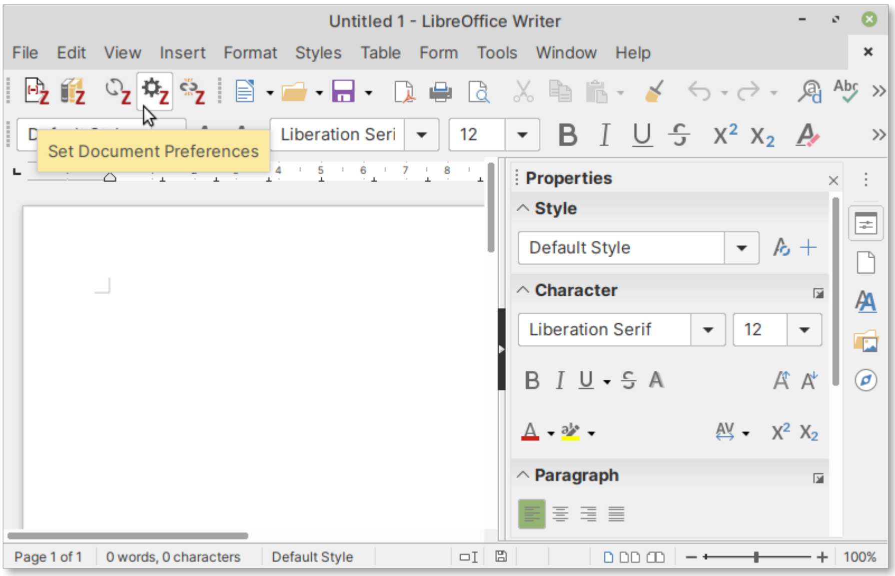


Fig. 107: Zotero toolbar in LibreOffice

—

You can now start inserting citations and bibliographies in your documents.

6.2 Creating a notebook using Zim

Zim is a *GUI* application for taking notes on the desktop. You can use it to create and maintain a personal or laboratory notebook on your computer.

6.2.1 Features of Zim

Using Zim, you can:

Create digital notebooks — add pages, and organize them using an index page.

Take notes quickly — using a WYSIWYG (What You See Is What You Get) interface[1].

Insert images — files, symbols, screenshots, equations and more[2].

Export notes — to formats like HTML, LaTeX, and Markdown.

—

With the help of plugins, you can add even more functionality.

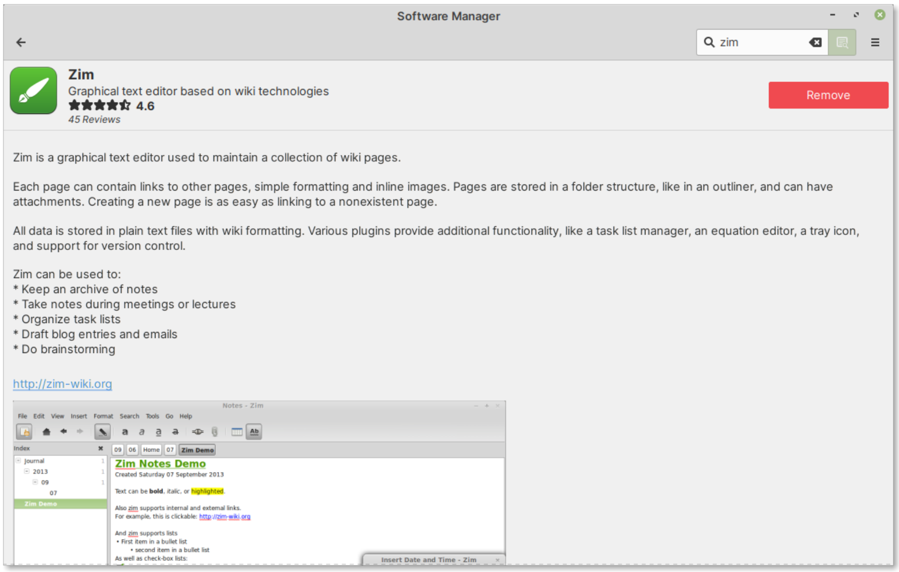
[1] Similar in functionality to that of word processors like LibreOffice Writer. You do not need to learn a new markup language to use Zim.

[2] Requires enabling plugins in preferences.

6.2.2 Installing Zim

Zim is available in the distribution repositories.

You can install it using *Software Manager* (Fig. 108).



Zim
Graphical text editor based on wiki technologies
4.6
45 Reviews

Zim is a graphical text editor used to maintain a collection of wiki pages.

Each page can contain links to other pages, simple formatting and inline images. Pages are stored in a folder structure, like in an outliner, and can have attachments. Creating a new page is as easy as linking to a nonexistent page.

All data is stored in plain text files with wiki formatting. Various plugins provide additional functionality, like a task list manager, an equation editor, a tray icon, and support for version control.

Zim can be used to:
* Keep an archive of notes
* Take notes during meetings or lectures
* Organize task lists
* Draft blog entries and emails
* Do brainstorming

http://zim-wiki.org

Notes - Zim
File Edit View Insert Format Search Tools Go Help
Index
Journal
2013
09
07
Zim Demo
09 06 Home 07 Zim Demo
Insert Date and Time - Zim

Zim Notes Demo
Created Saturday 07 September 2013

Text can be **bold**, *italic*, or highlighted.

Also zim supports internal and external links.
For example, this is clickable: http://zim-wiki.org

And zim supports lists
• First item in a bullet list
• second item in a bullet list
As well as check-box lists:

Fig. 108: Installing Zim using Software Manager

6.2.3 Creating a notebook

Open Zim by clicking on its icon in applications menu (Fig. 109). You can use the search bar to locate it quickly.

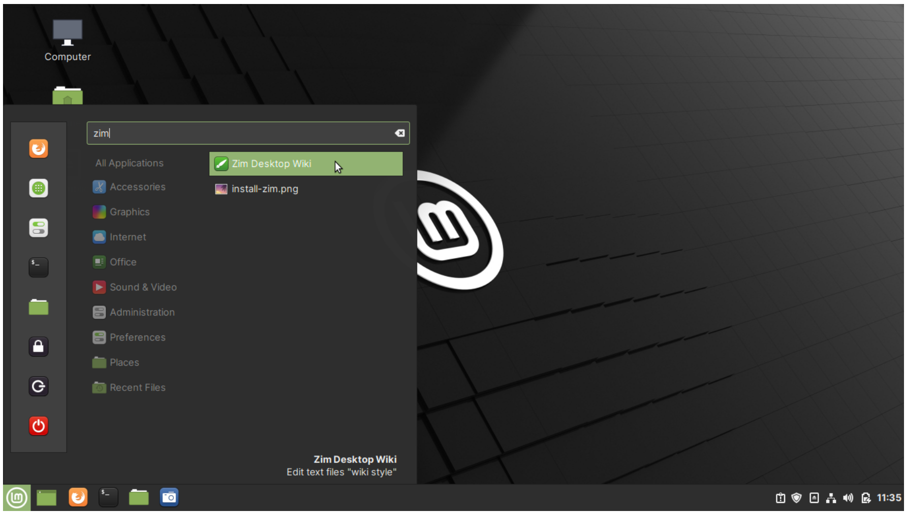
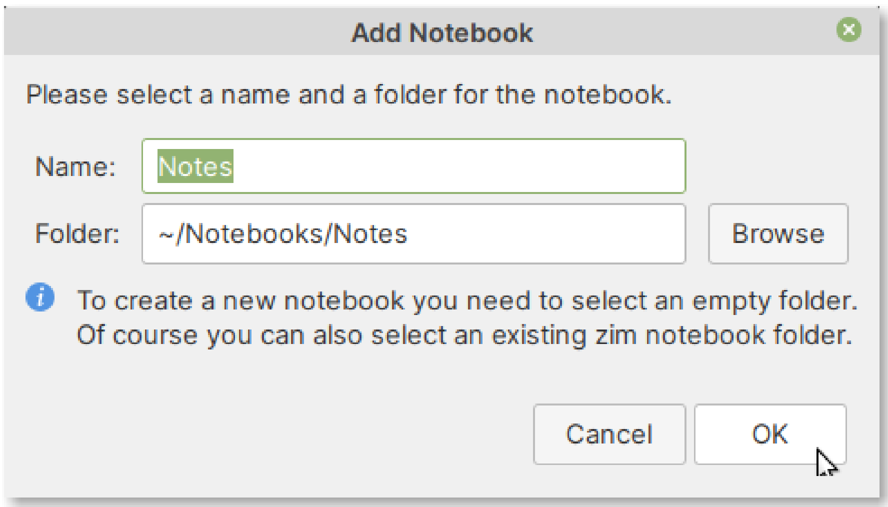


Fig. 109: Opening Zim from applications menu

When you open the program for the first time, you will notice an `Add Notebook` dialog (Fig. 110).

Here, you can set a name for this notebook and select a directory to save notebooks.



Please select a name and a folder for the notebook.

Name: Notes

Folder: ~/Notebooks/Notes

To create a new notebook you need to select an empty folder.
Of course you can also select an existing zim notebook folder.

Fig. 110: Add notebook

You can accept the defaults.

Click on the OK button to proceed.

This will open the main window of the program (Fig. 111).

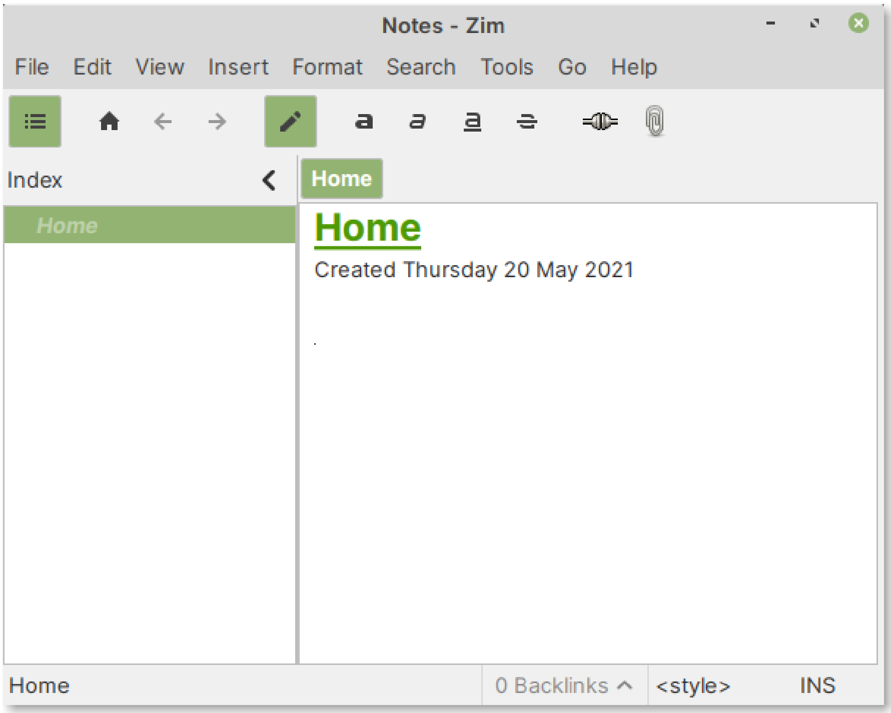


Fig. 111: Main window of Zim

6.2.4 Taking notes

The Home page is the main page of a notebook.

You can start typing your notes here. Zim saves changes automatically.

Basic text formatting

Formatting text in Zim is similar to that of word processor programs.

You can use the Format menu bar entry and select the desired option (Fig. 112).

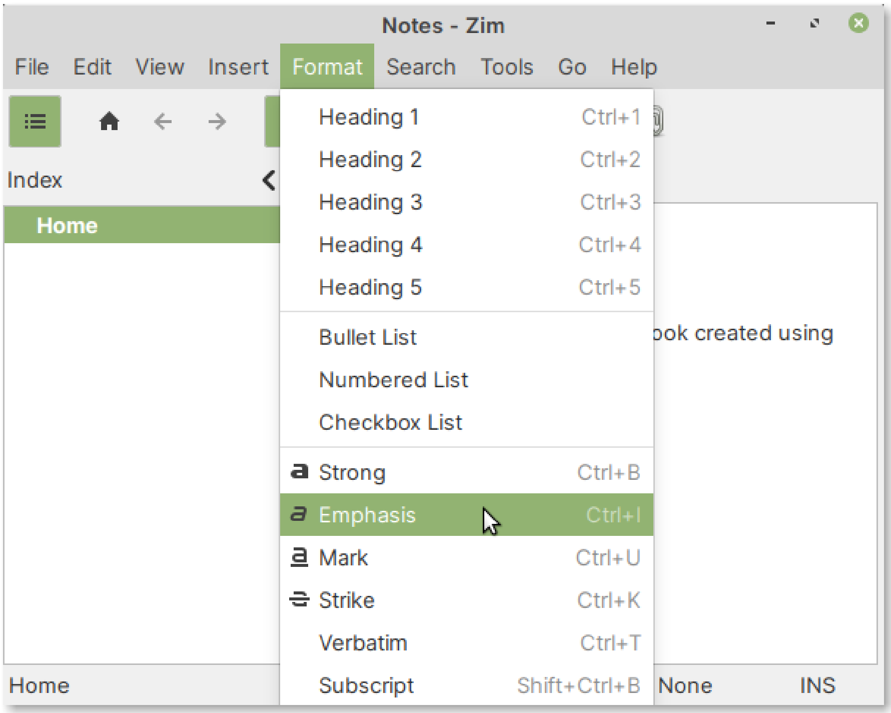


Fig. 112: Options for formatting text

Keyboard shortcuts

Alternatively, you can use keyboard shortcuts to apply formatting to your text.

For example:

Bold : `Ctrl + b`

Italic or *Emphasis* : `Ctrl + i`

$^{\text{Superscript}}$: `Ctrl + Shift + p`

$_{\text{Subscript}}$: `Ctrl + Shift + b`

For headings, use:

Heading 1: `Ctrl + 1`

Heading 2: `Ctrl + 2`

Heading 3: `Ctrl + 3`

Heading 4: `Ctrl + 4`

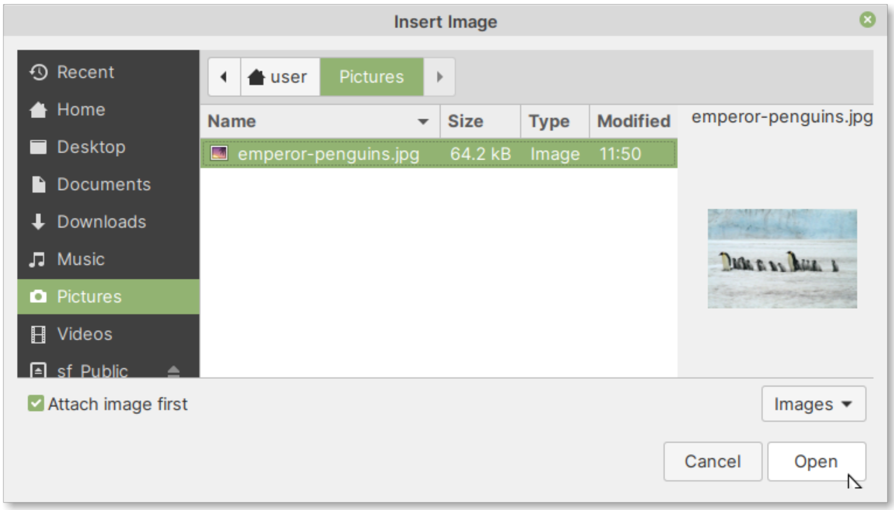
Heading 5: `Ctrl + 5`

Inserting images

To insert an image, you can select the following option from the menu bar:

Insert → Image

In the dialog that opens, browse to the directory containing your image and then select the image (Fig. 113).



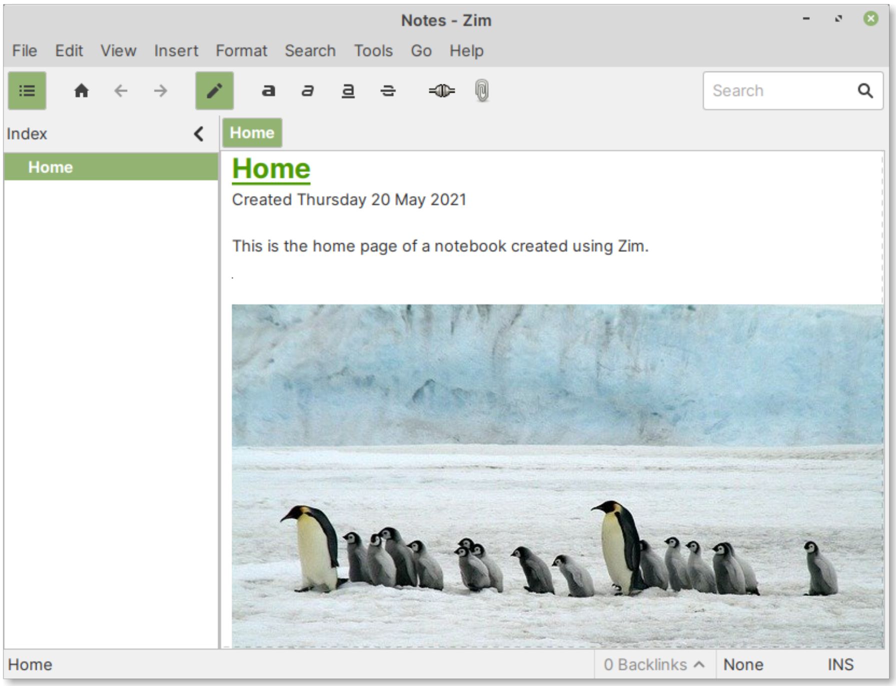
Name	Size	Type	Modified
emperor-penguins.jpg	64.2 kB	Image	11:50

[✓] Attach image first

Fig. 113: Insert image. Check Attach image first option

Check the Attach image first option. This will copy the image to your notebook directory.

Click on the Open button to insert the image (Fig. 114).



Home

Created Thursday 20 May 2021

This is the home page of a notebook created using Zim.

Fig. 114: Image inserted into current page

Resizing an image

If you would like to resize the image, *right-click* on the image and select Edit Properties (Fig. 115).

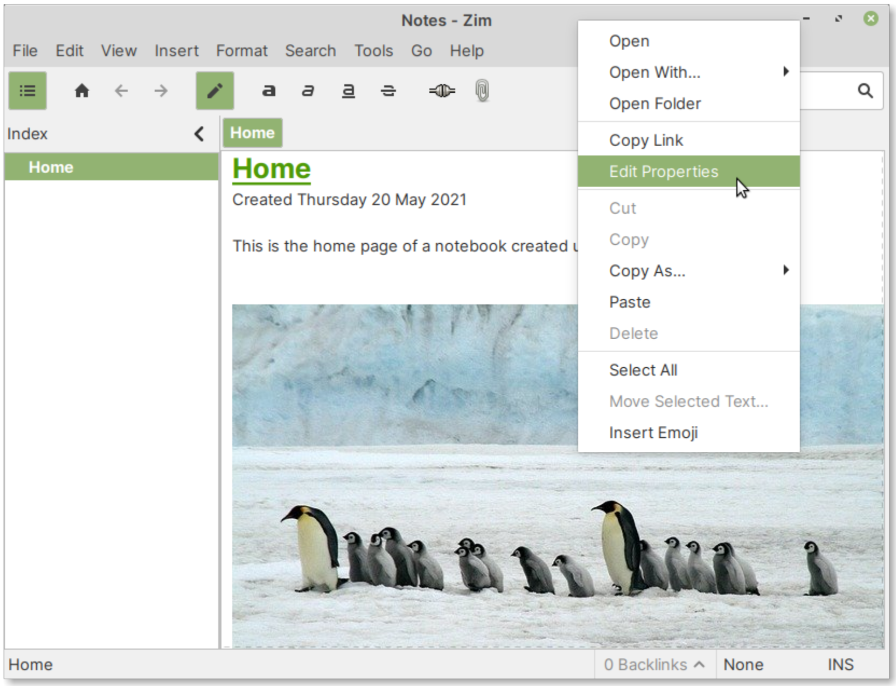
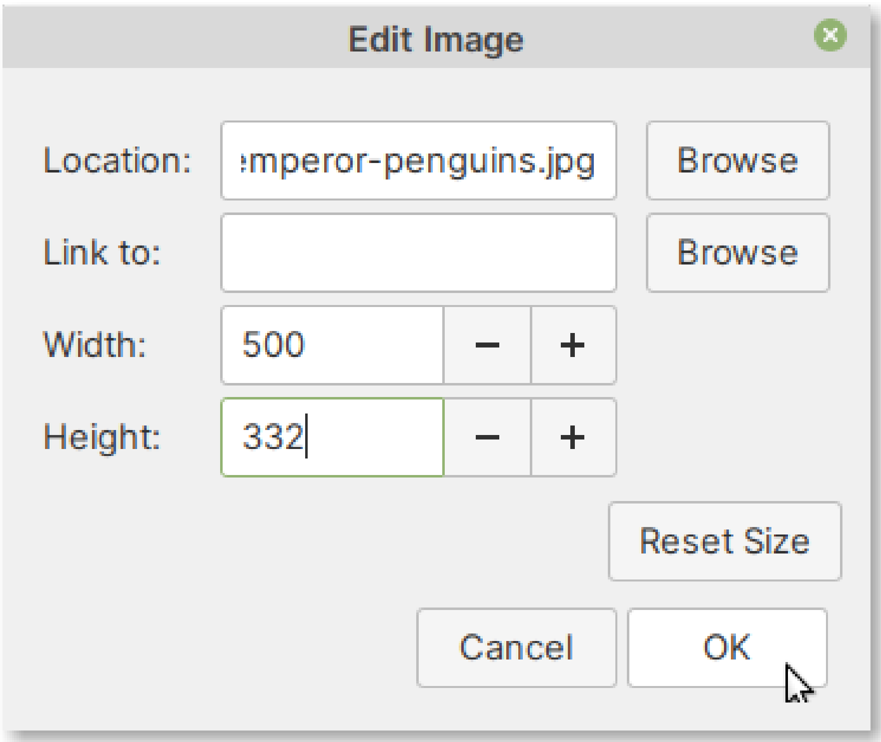


Fig. 115: Edit image properties

In the dialog window that opens, change either the Width or the Height of the image to the desired size and click on the OK button (Fig. 116).


Edit Image

Location: :mperor-penguins.jpg

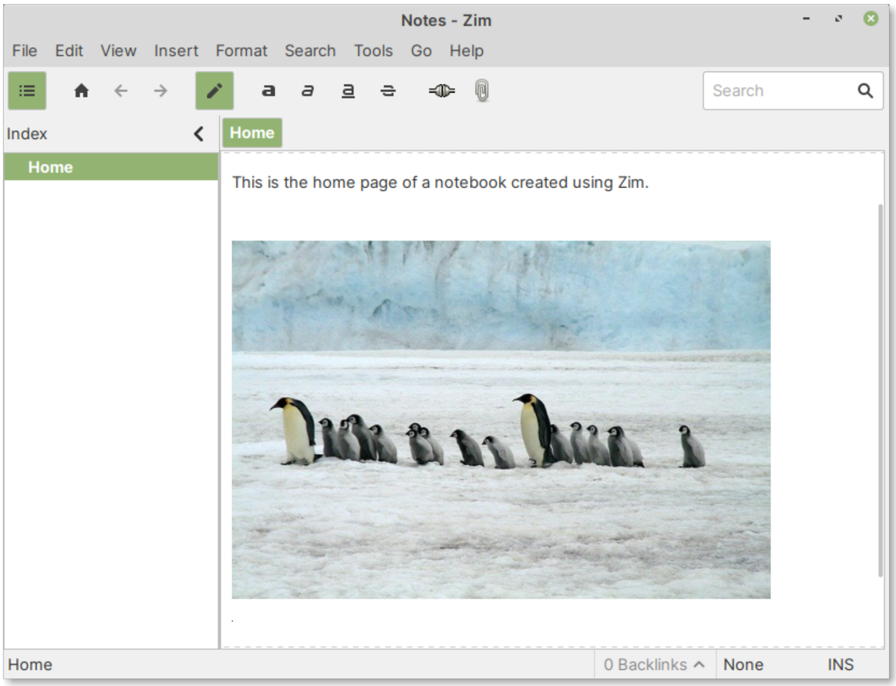
Link to:

Width: 500

Height: 332

Fig. 116: Change the width or height of the image

The image will now be resized (Fig. 117).



This is the home page of a notebook created using Zim.

Fig. 117: Resized image

Adding more pages

To add more pages, select the following option from the menu bar (Fig. 118):

File → New Page

Note: You can also *right-click* on a page in the Index view and select the New Page Here option.

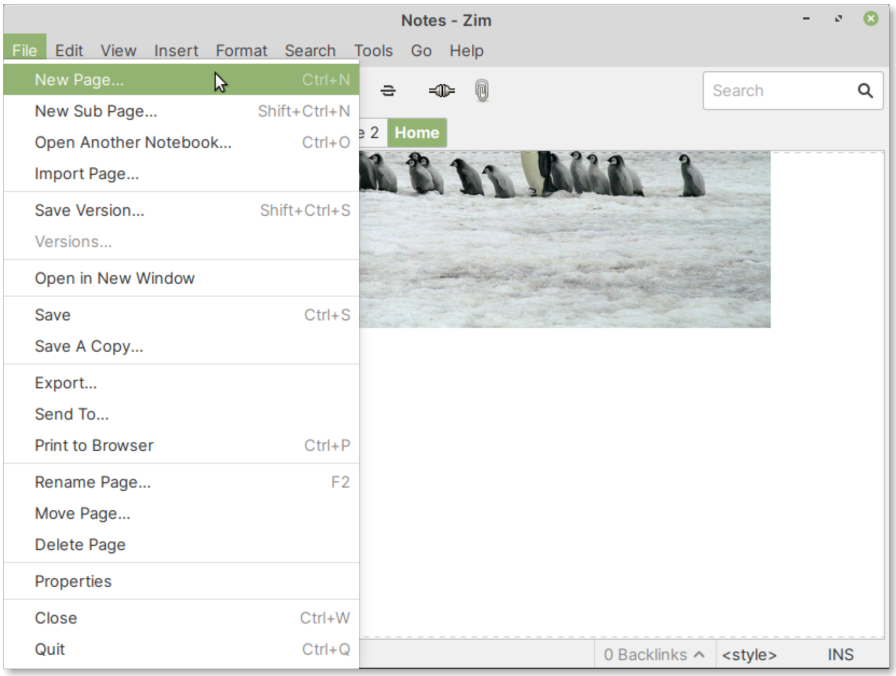
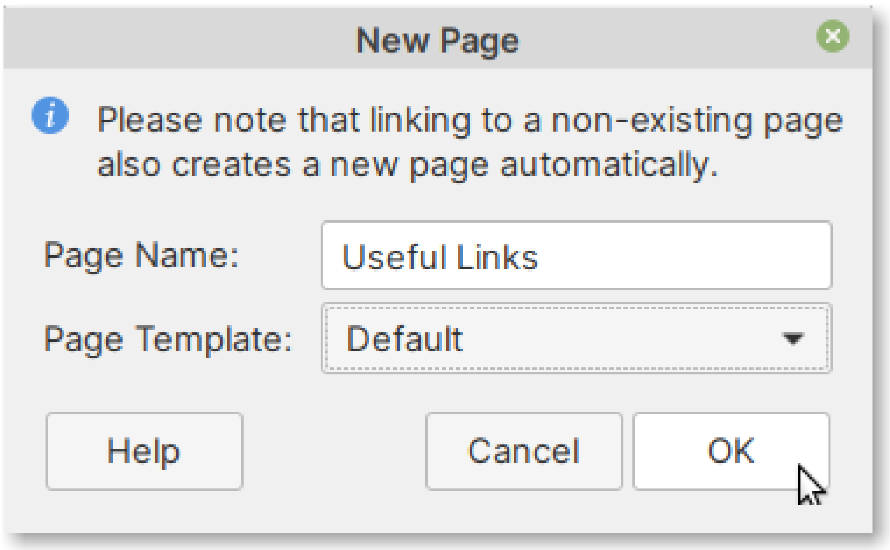


Fig. 118: Adding a new page

Enter a name for the page in the `New Page` dialog window (Fig. 119) and click on the `OK` button.



Please note that linking to a non-existing page also creates a new page automatically.

Page Name: Useful Links

Page Template: Default

Fig. 119: Enter a name for the new page

The new page will appear in the Index view (Fig. 120).

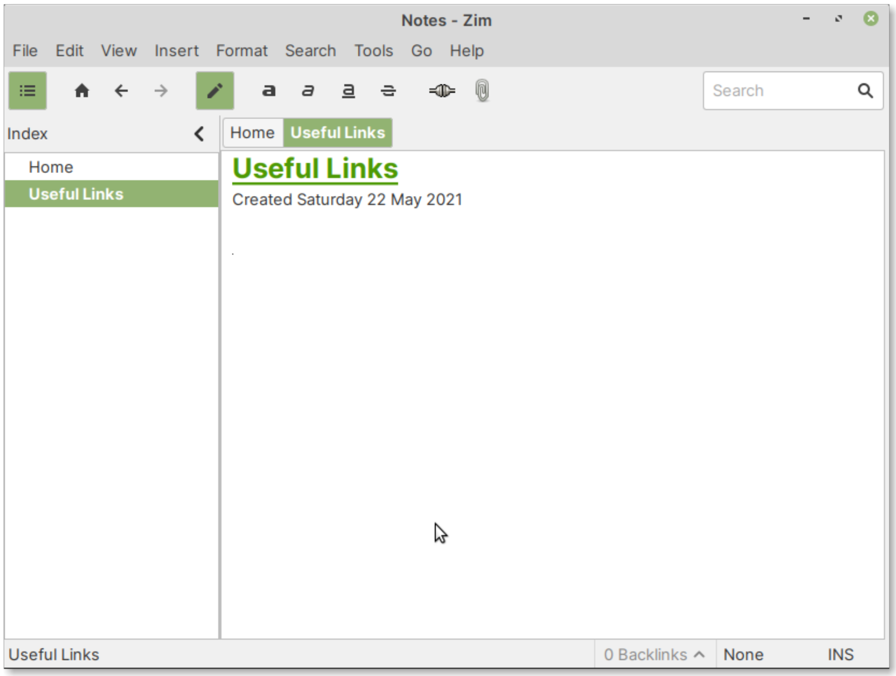

Notes - Zim
File Edit View Insert Format Search Tools Go Help
Search
Index
Home
Useful Links
Useful Links
Created Saturday 22 May 2021
0 Backlinks
None
INS

Fig. 120: Newly added page

Adding subpages

A subpage will be placed under an *existing* page.

To add a new subpage:

1. Select an existing page in Index view
2. Select File → New Sub Page from the menu bar
3. Enter a name for the page and click on the OK button

Deleting or renaming pages

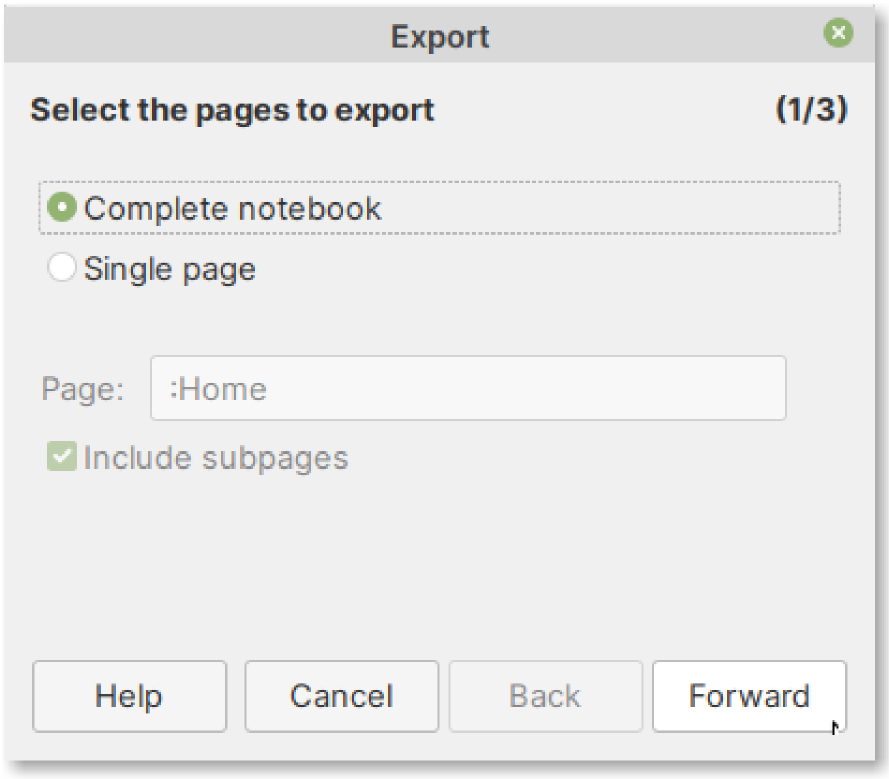
To delete or rename pages, you can *right-click* on the page and select the desired function.

6.2.5 Exporting notes and notebooks

To export the notebook or a page, select:

File → Export

In the dialog window that appears, select the desired option and click on the Forward button (Fig. 121).



Export

Select the pages to export (1/3)

[✓ Complete notebook] Single page

Page: :Home

[✓] Include subpages

Fig. 121: Export notebook

In the next screen, you can select the format of the exported notebook or page in the Format drop-down box (Fig. 122).

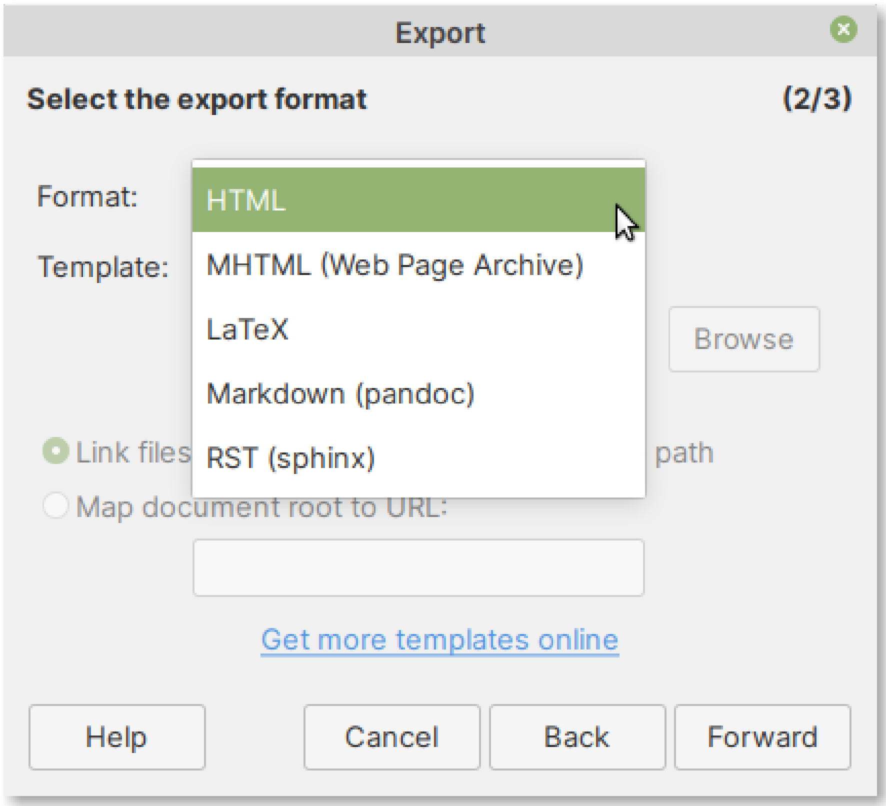


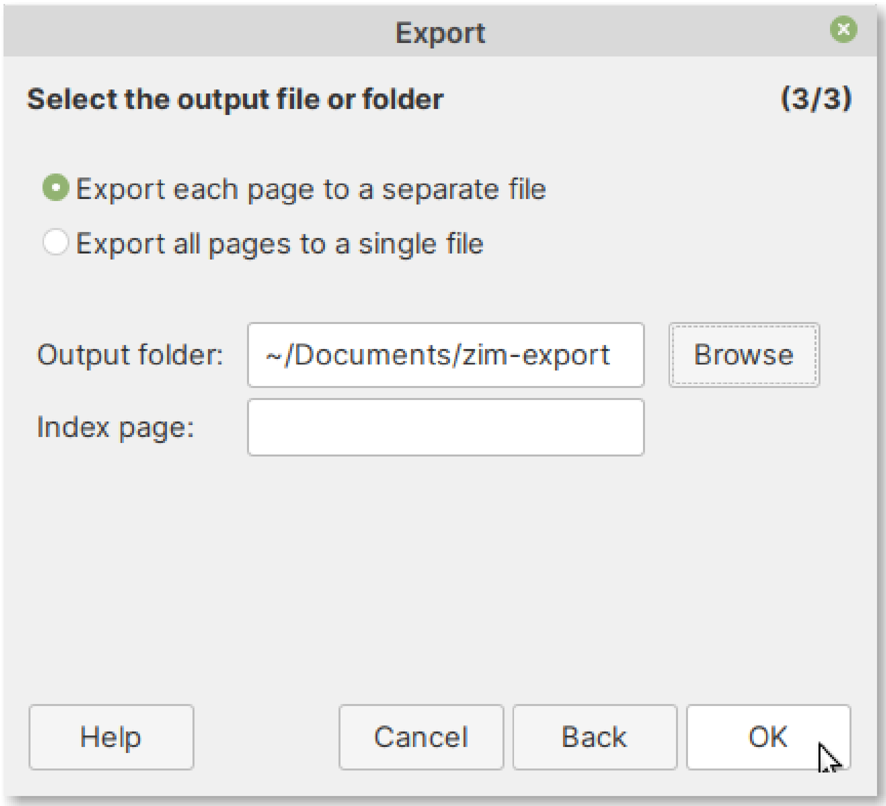
Fig. 122: Select an export format

You can leave it at the default — HTML. Click on the Forward button to proceed.

In the final screen, select a directory to save the exported notebook or page.

Note: If you are using the defaults, it is a good idea to create a new output directory and select it under Output folder.

Click on the OK button complete export.


Export

Select the output file or folder (3/3)

[✓ Export each page to a separate file]

Export all pages to a single file

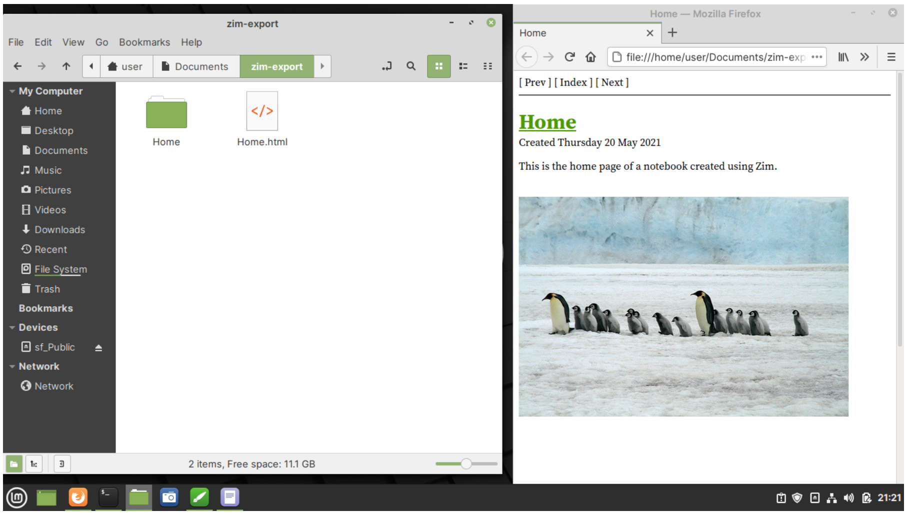
Output folder: ~/Documents/zim-export

Index page:

Fig. 123: Select an output folder

When complete, open the output directory in file manager.

Double-click on Home.html to open it in the browser (Fig. 124).



[Prev] [Index] [Next]

Home

Created Thursday 20 May 2021

This is the home page of a notebook created using Zim.

Fig. 124: Viewing exported notebook

6.2.6 Activating more features using plugins

These are some plugins included with Zim:

Insert Symbol — insert special symbols and characters

Table of Contents — adds a widget displaying the current page's table of contents

Tags — add tags to your notes

Tray Icon — adds an icon to system tray for quick access to your notebooks

Spell Checker — adds support for checking spelling

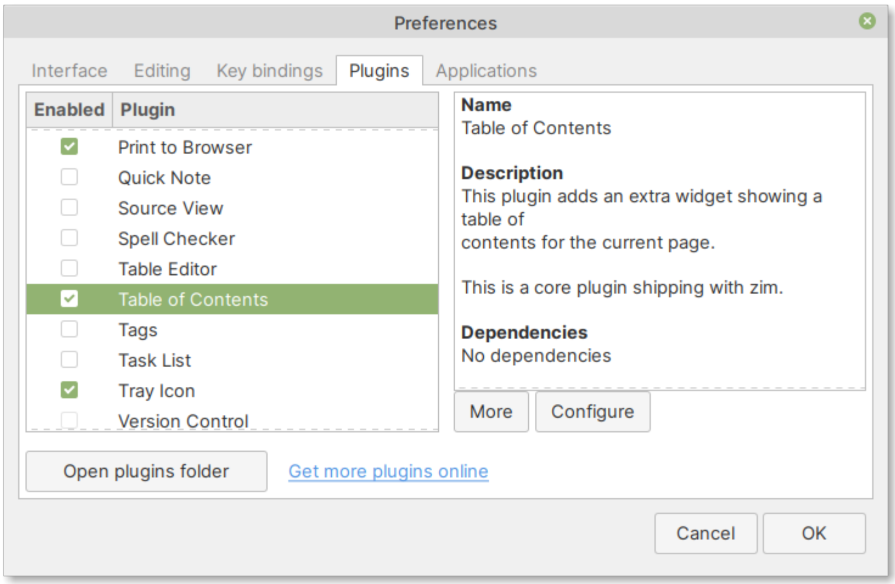
—

You will need to activate (or enable) them in preferences, before you can start using them.

Enabling a plugin

To enable a plugin:

1. Select Edit → Preferences
2. Click on the Plugins tab
3. Check the box corresponding to the plugin you wish to enable (Fig. 125)
4. Click on the OK button



Enabled	Plugin
✓	Print to Browser
	Quick Note
	Source View
	Spell Checker
	Table Editor
✓	Table of Contents
	Tags
	Task List
✓	Tray Icon
	Version Control

Name
Table of Contents

Description
This plugin adds an extra widget showing a table of
contents for the current page.

This is a core plugin shipping with zim.

Dependencies
No dependencies

Fig. 125: Enabling plugins

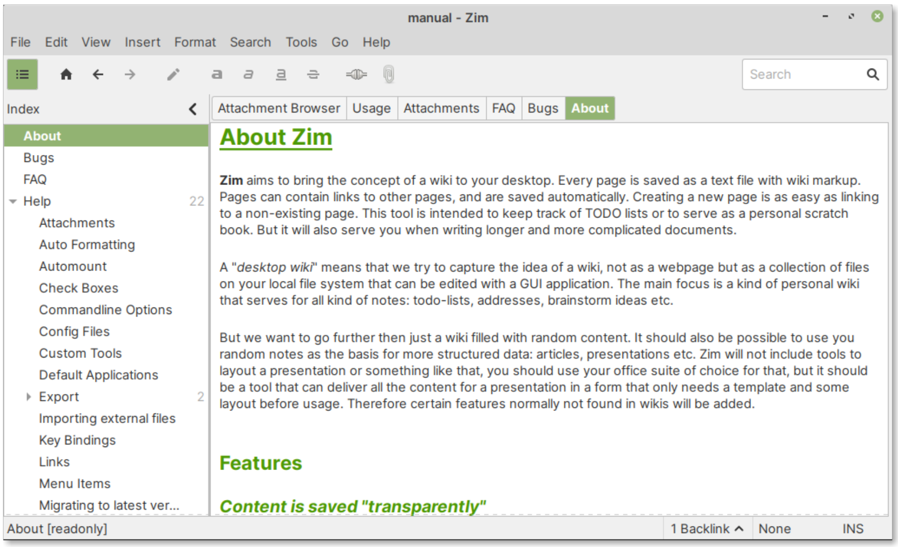
Note: *Some plugins have additional options.*

Click on the Configure button below the plugin description to access these options.

6.2.7 Getting help

Zim includes a detailed user manual. You can access this by selecting (Fig. 126):

Help → Contents



About Zim

Zim aims to bring the concept of a wiki to your desktop. Every page is saved as a text file with wiki markup. Pages can contain links to other pages, and are saved automatically. Creating a new page is as easy as linking to a non-existing page. This tool is intended to keep track of TODO lists or to serve as a personal scratch book. But it will also serve you when writing longer and more complicated documents.

A "*desktop wiki*" means that we try to capture the idea of a wiki, not as a webpage but as a collection of files on your local file system that can be edited with a GUI application. The main focus is a kind of personal wiki that serves for all kind of notes: todo-lists, addresses, brainstorm ideas etc.

But we want to go further then just a wiki filled with random content. It should also be possible to use you random notes as the basis for more structured data: articles, presentations etc. Zim will not include tools to layout a presentation or something like that, you should use your office suite of choice for that, but it should be a tool that can deliver all the content for a presentation in a form that only needs a template and some layout before usage. Therefore certain features normally not found in wikis will be added.

Features

Content is saved "transparently"

Fig. 126: Help on using Zim

7 Glossary

Anaconda A free and open-source (Anaconda Individual Edition) Python and R distribution. It includes *Conda* and more than 250 open-source scientific packages. Additional packages can be installed from *Anaconda Cloud* repositories.

Website: https://www.anaconda.com/products/individual

Anaconda Cloud Repository for Python and R packages and notebooks.

Website: https://anaconda.org

apt A command-line program that handles the installation and removal of software on Debian, Ubuntu and other Linux distributions.

Related section:

-> *The command-line version (apt)*

bioconda A channel for *Conda* package manager with over 7000 packages of bioinformatics software.

Website: https://bioconda.github.io/

build-essential A package that installs software build tools like `make` and compilers like `gcc`, which are required for building modules written in languages like C.

Bash *GNU* project's shell program. The name stands for The Bourne-Again Shell. It is the most commonly used shell in Linux distributions.

Website: https://www.gnu.org/software/bash/

Bioconductor Tools written in the R programming language for the analysis and comprehension of high-throughput genomic data.

Website: https://www.bioconductor.org/

BiocManager An R package to install and update packages from the *Bioconductor* project repository.

Website: https://cran.r-project.org/package=BiocManager

BioPython Python tools for computational molecular biology.

Website: http://biopython.org/

Bio::Phylo Perl package for phylogenetic analysis.

Website: https://metacpan.org/pod/Bio::Phylo

commands Also called as binaries or executables.

Conda Conda is a package and environment manager.

As a *package manager*, it can be used to install software from *Anaconda Cloud*. Software *dependencies* will be installed automatically.

As an *environment manager*, it can be used to create and manage environments containing different sets of packages. You can activate and use these environments as necessary.

cpanminus A Perl script to get, build and install modules from *CPAN*. It provides the `cpanm` command.

Website: https://metacpan.org/pod/App::cpanminus

cutadapt A Python program that finds and removes adapter sequences, primers, poly-A tails and other unwanted sequences from high-throughput sequencing reads.

Website: https://cutadapt.readthedocs.io

CPAN Abbreviation for Comprehensive Perl Archive Network. It provides additional Perl modules for installation (196,752 as of Dec 2020).

Website: https://www.cpan.org/

CRAN Abbreviation for Comprehensive R Archive Network. A network of FTP and web servers providing up-to-date versions of R, packages and documentation.

Website: https://cran.r-project.org/

Debian A Linux distribution made of free and open source software. It is free for anyone to download, use, modify and distribute.

Website: https://www.debian.org/

Debian package An archive of executable files, libraries, and documentation of software. It can be installed on Debian Linux and Debian-based systems like Ubuntu and Linux Mint. These files have the `.deb` file extension.

dependencies Additional programs or libraries that are needed for a program to work.

edgeR An R package for empirical analysis of digital gene expression data. It is available from the *Bioconductor* project repository.

Website: https://www.bioconductor.org/packages/edgeR/

FAST Abbreviation for FAST Analysis of Sequences Toolbox. A set of utilities written in Perl that extend the UNIX paradigm to bioinformatic sequence records.

Website: https://metacpan.org/pod/FAST

Files The default file manager in *Linux Mint* Cinnamon edition. Its original name is Nemo.

gdebi A simple tool to install *Debian package* files along with their *dependencies* (if any).

Website: https://launchpad.net/gdebi

GNU GNU is a recursive acronym for GNU's Not Unix. The goal of the project is to offer a Unix-compatible system that would be 100% free software.

Website: https://www.gnu.org/

GUI Abbreviation for graphical user interface. On a personal computer, it typically includes application windows with buttons to access their functions, icons for launching applications and widgets for managing devices and services.

IDE Abbreviation for integrated development environment. Some examples include PyCharm, RStudio and Eclipse.

local-lib A Perl module to create a local directory structure to install modules with their *dependencies* without requiring administrator privileges.

Website: https://metacpan.org/pod/local::lib

Linux Mint A desktop Linux distribution. It is based on *Debian* and *Ubuntu*.

Website: https://linuxmint.com/

MEGA Software for Molecular Evolutionary Genetics Analysis. Available as *GUI* and command-line versions.

Website: https://megasoftware.net/

MetaCPAN A search engine for Perl packages and modules available on *CPAN*.

Website: https://metacpan.org/

Miniconda A minimal distribution of *Conda*. It is faster to install and also uses less disk space, when compared to *Anaconda* — the alternative installer which comes bundled with additional packages.

Website: https://docs.conda.io/en/latest/miniconda.html

Modeller A program for comparative protein structure modelling by satisfaction of spatial restraints.

Website: https://salilab.org/modeller/

nano A simple command-line based text editor.

Website: https://www.nano-editor.org/

OVA A file format for distributing virtual machine images.

pip The Python package installer. It can be used to install packages from *PyPI* and other Python package indexes.

Website: https://pip.pypa.io/

PyMOL PyMOL is a molecular visualization system originally developed by Warren L. Delano. It is currently maintained by Schrödinger, Inc.

Website: https://pymol.org/2/

PyPI Abbreviation for Python Package Index. Repository for software written in Python (275,161 projects as of Dec 2020).

Website: https://pypi.org/

root The primary administrator account on a Linux system.

R Programming language and free software environment for statistical computing and graphics.

Website: https://r-project.org

RStudio Desktop An *IDE* for *R*. It includes an R console, an editor with syntax-highlighting and tools for plotting, debugging and managing R packages. The open-source version can be downloaded for free from the project website.

Website: https://rstudio.com/products/rstudio/download/

Related sections:

-> *Installing a Debian package*

setuptools Python program for building and installing Python packages.

shell The shell provides a command-line interface (CLI) to the operating system's services.

Software Manager The default application(*GUI*), for installing software on Linux Mint.

Applications with similar functionality are available on other Linux distributions. For example:

- Ubuntu Software on Ubuntu
- Software on Fedora and other GNOME-based distributions
- Discover on Kubuntu and other KDE-based distributions.

Related sections:

-> *The quick and easy method*

Synaptic A *GUI* package manager for systems using *apt*.

Website: https://www.nongnu.org/synaptic/

text editor Program to edit text files.

Examples (*GUI*) - Text Editor (xed) on Linux Mint, Text Editor (gedit) on Ubuntu, Kwrite on KDE Plasma, Geany etc.,

Examples (command-line) — *nano*, VIM or vi, GNU Emacs etc.,

Related sections:

`->` *Text Editor — create and edit text files*

`->` *Editing text files using nano*

Ubuntu Linux distribution developed by Canonical and community of developers. It is based on *Debian*.

Website: https://ubuntu.com/

wheel Python program for installing packages distributed in Python wheel (`.whl`) format.

8 Credits

This book is created on a Linux system using open source software. Fonts and images used are under an open source licence or in the public domain.

Software

Linux Mint — https://linuxmint.com

Sphinx — https://www.sphinx-doc.org/en/master/

sphinx-autobuild extension — https://github.com/executablebooks/sphinx-autobuild

LaTeX for generating PDF — https://www.latex-project.org/

GNOME Builder for editing text — https://wiki.gnome.org/Apps/Builder

Firefox for previewing HTML — https://www.mozilla.org/en-GB/firefox/

Pandoc for converting HTML to ODT — https://pandoc.org/

LibreOffice for proofreading — https://www.libreoffice.org/

LanguageTool extension for spelling and grammar checks — https://languagetool.org/

Inkscape for editing images — https://inkscape.org/

GIMP for editing images — https://www.gimp.org/

Building and testing the virtual machine

Ansible — https://www.ansible.com/

Vagrant — https://www.vagrantup.com/

VirtualBox — https://www.virtualbox.org/

Fonts

Source Serif 4 — https://github.com/adobe-fonts/source-serif/

Fira Sans — https://github.com/mozilla/Fira

Fira Code — https://github.com/tonsky/FiraCode

Recursive Mono — https://github.com/arrowtype/recursive

Images

The emperor penguin image used in the book cover, is obtained from openclipart.org.

https://openclipart.org/detail/47299/emperor-penguin

genindex

Index