Compiling Grisbi 3.0.0 from Source
on Mint 21.2 Xfce

By `kalwisti`

29 Nov. 2023

Contents

1 Introduction

The current stable version of the program is 3.0.0 (just released on 17 Nov. 2023).[1] Although Grisbi has a Flatpak on Flathub.org, it is still at ver. 2.0.5. To the best of my knowledge, there is not a PPA available, so if you want to run the most recent version, you must compile it yourself from source.

---

[1]According to the News text file in the source directory, Grisbi 3.0.0 includes these new features: **a)** a change in the Payee search; **b)** added a new type of consumer loan; **c)** added all transactions in archive when the account is closed; **d)** bug fixes; **e)** code cleanup; **f)** preparing for the transition to GTK 4.

The process is not as intimidating as it sounds; I was pleasantly surprised that I was able to compile Grisbi 3.0.0 without much difficulty on Mint 21.2 (Xfce DE). If you are interested in trying, this tutorial covers the procedure that I followed.

> **Caveats:**
>
> I should mention that by compiling your own version of Grisbi, the program will not be upgraded automatically.
>
> It is also possible that the functionality of your compiled Grisbi package might break when its dependencies and software are upgraded during your normal update process. If this happens, you might need to recompile. Using self-compiled software means that you are responsible for maintaining it, as well as the libraries that it depends on.

That said, compiling Grisbi from source is safe and unlikely to break your system. It will take you on a mini-adventure and provide a sense of accomplishment when you balance your checkbook with version 3.0.0 …

2   Download the Source Files

Go to Grisbi's SourceForge website

https://sourceforge.net/projects/grisbi/files/grisbi stable/3.0.x/

and download the `grisbi-3.0.0.tar.bz2` file (12.4 MB in size, modified on 2023-11-17).

3   Move File to `/home` Directory and Extract It

Move the tarball (i.e., the compressed Grisbi `.tar` file) that you just downloaded to your `/home` folder.

Then, use Archive Manager — or a similar utility — to extract the compressed files. When you have extracted the tarball, you should have a folder named `grisbi-3.0.0`. That folder should contain 8 subfolders as well as 24 assorted files (illustrated below):
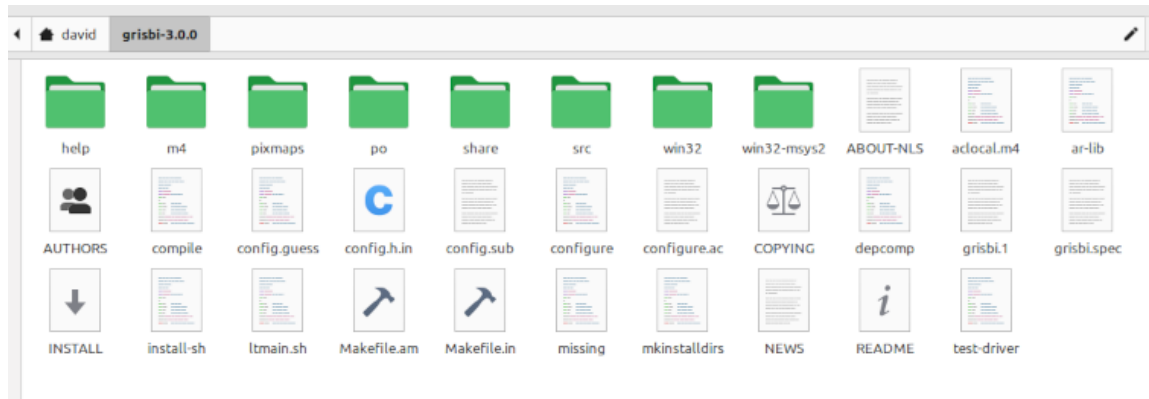
4   Remove Grisbi 2.0.5 (If Installed)
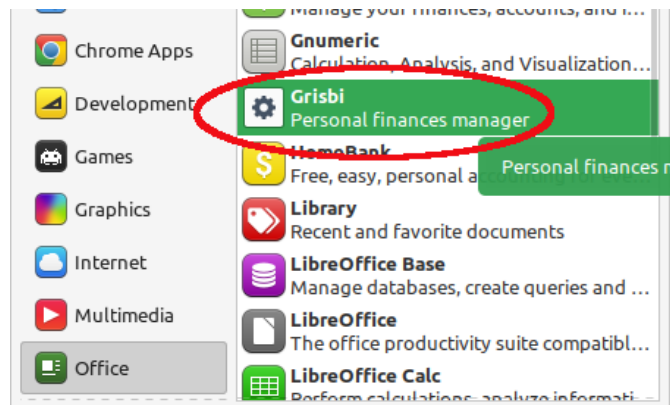
If you have ver. 2.0.5 installed on your system, you should remove it before trying to compile ver. 3.0.0.

- Start up the Synaptic Package Manager.

- Remove these two packages:

    - `grisbi`
    - `grisbi-common`

- (Choose the option "`Mark for Complete Removal`".)

Even after removing the aforementioned packages, on my system, there was a 'stray' menu entry for Grisbi:

To remove the menu item, I took these steps:

- Open the Thunar File Manager.

- Make hidden files and directories visible by typing `Ctrl+H`.

 Alternatively, from the `View` menu > choose "`Show Hidden Files`".

- Delete these two directories (folders) from your `/home` folder:

  1. `~/.config/grisbi`
  2. `~/.local/share/grisbi`

  (As a reminder, the tilde [`~`] is used to refer to your home directory. So if your username is "joe", the tilde would refer to /home/joe).

- In addition, delete the file "`grisbi.desktop`" from this folder:

  `~/.local/share/applications/`

Note: I first tried to remove Grisbi's menu item with the MenuLibre editor (as suggested in the Linux Mint Forums), but was unsuccessful. Due to a parsing error, MenuLibre did not display an entry for Grisbi under the **Office** section [as shown in the screenshot below]:
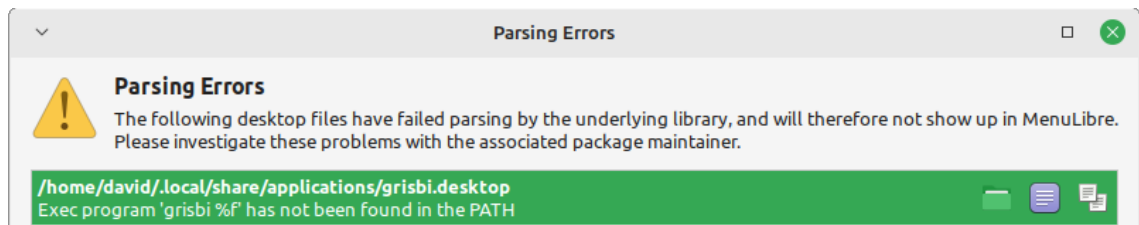
5   Install Required Packages and Dependencies

Use the Synaptic Package Manager to install the following packages, if they are not already present on your system:

Package Name	Version No.
build-essential	12.9
checkinstall	1.6.2+git20170426…
libglib2.0-0	2.72.4-0
libglib2.0-dev	2.72.4-0
libglib2.0-dev-bin	2.72.4-0
libgtk2.0-dev	2.24.33-2
libgtk-3-0	3.24.33-1
libgtk-3-dev	3.24.33-1
libgtk-3-bin	3.24.33-1
libgsf-1-114	1.14.47-1
libgsf-1-dev	1.14.47-1
libgsf-bin	1.14.47-1
libgoffice-0.10-10	0.10.51-1
libgoffice-0.10-dev	0.10.51-1
libofx7	1:0.10.3-1
libofx-dev	1:0.10.3-1
libssl3	3.0.2-0
libssl-dev	3.0.2-0
libxml2	2.9.13+dfsg1
intltool	0.51.0-6

> **Note:**
> The list above is not exhaustive, as some of these packages will pull in dependencies of their own.

6 Compile the Source Code

Now you are ready to compile Grisbi's source code.

Open a Terminal and navigate to the extracted Grisbi folder in your /home directory:

```
$ cd grisbi-3.0.0
```

Type:

```
        $ sudo ./configure
```

(Remember to include the initial period and slash (" ./ ") before the `configure` command.)

This step will perform checks / tasks that will allow your system to compile the software successfully. If the configure script detects a missing package, it will notify you of what to do at the end of the script. You will need to install any missing packages that the configure script has detected.

When that process finishes, you will see output similar to the following:

```
[ ... ]
config.status: executing libtool commands
config.status: executing default-1 commands
config.status: executing po/stamp-it commands


            Build with libxml2:         yes
            Build with OpenSSL:         no
            Build with OFX plugin:      yes
            Build with goffice:         yes
            Build win32:                no
            Build gtk_mac_integration: no
            CUnit/unit tests support:   no
            Install french user doc:    yes
            Enable Debug:               no
            Enable Early Debug:         no
            Enable Config file:
            Enable fail on error:       no
            Enable nooptim:             no
            Compiler flags:             -g -O2
```

When that process finishes, issue the command:

```
        $ sudo make
```

The make command will actually begin compiling Grisbi. This will take some time to complete, depending on your available system resources. (In my case, the

compilation finished in a few minutes because my desktop PC has an Intel Core i3-6100 CPU [3.7 GHz], with 16 GB of RAM).

The final step is to install your now-compiled Grisbi package with the command:

```
$ sudo checkinstall
```

Linux Mint Forums user xenopeek explains the advantage of using "sudo checkinstall" instead of the more common "sudo make install" command:

> "[It] will make the software and its files known to your package managers (e.g., Software Manager and apt) so that you can easily remove the software or replace it with a newer version."[2]

As checkinstall runs, follow its on-screen instructions. I discovered — too late — that I could have filled out some of the requested information fields more completely than I did. (The fields that I left blank did not prevent the installation from completing, nor has it adversely affected Grisbi's performance.)

Sample output is pasted below:

```
The package documentation directory ./doc-pak does not exist.
Should I create a default set of package docs?  [y]:


-----


Please write a description for the package.
End your description with an empty line or EOF.
>> Personal finance manager
>>


*****************************************
**** Debian package creation selected ***
*****************************************


This package will be built according to these values:


0 -  Maintainer: [ root@david-ThinkCentre-M800 ]
1 -  Summary: [ Personal finance manager ]
```

```
2 -  Name:    [ grisbi ]
3 -  Version: [ 3.0.0 ]
4 -  Release: [ 1 ]
5 -  License: [ GPL ]
6 -  Group:   [ checkinstall ]
7 -  Architecture: [ amd64 ]
8 -  Source location: [ grisbi-3.0.0 ]
9 -  Alternate source location: [   ]
10 - Requires: [   ]
11 - Recommends: [   ]
12 - Suggests: [   ]
13 - Provides: [ grisbi ]
14 - Conflicts: [   ]
15 - Replaces: [   ]


Enter a number to change any of them or press ENTER to continue:
```

Based on an entry in Warren Post's blog,[3] these fields could have been filled out as follows:

Description for the package Personal finances manager

Alternate source location grisbi.org

Group Office

When the checkinstall process completes and states "Installation successful", the Terminal displayed this final message:

```
Done. The new package has been installed and saved to
/home/david/grisbi-3.0.0/grisbi_3.0.0-1_amd64.deb

 You can remove it from your system anytime using:
     dpkg -r grisbi
```

As xenopeek mentioned, Synaptic is aware of the new Grisbi package and displays that it has been installed:

[3]Warren Post, "Creating Packages from Source with checkinstall," *A Maze of Twisty Little Passages* (blog), Jan. 9, 2010, https://warrenpost.wordpress.com/2010/01/09/checkinstall/.

S	Package	Installed Version	Latest Version	Description
☐	grisbi	3.0.0-1	3.0.0-1	Personal finance manager
☐	grisbi-common		2.0.5-1build1	shared files for the finance m
☐	skrooge		2.27.0-1build1	personal finance manager fo

In addition, a 64-bit .deb package is now present in my `grisbi-3.0.0` source folder:
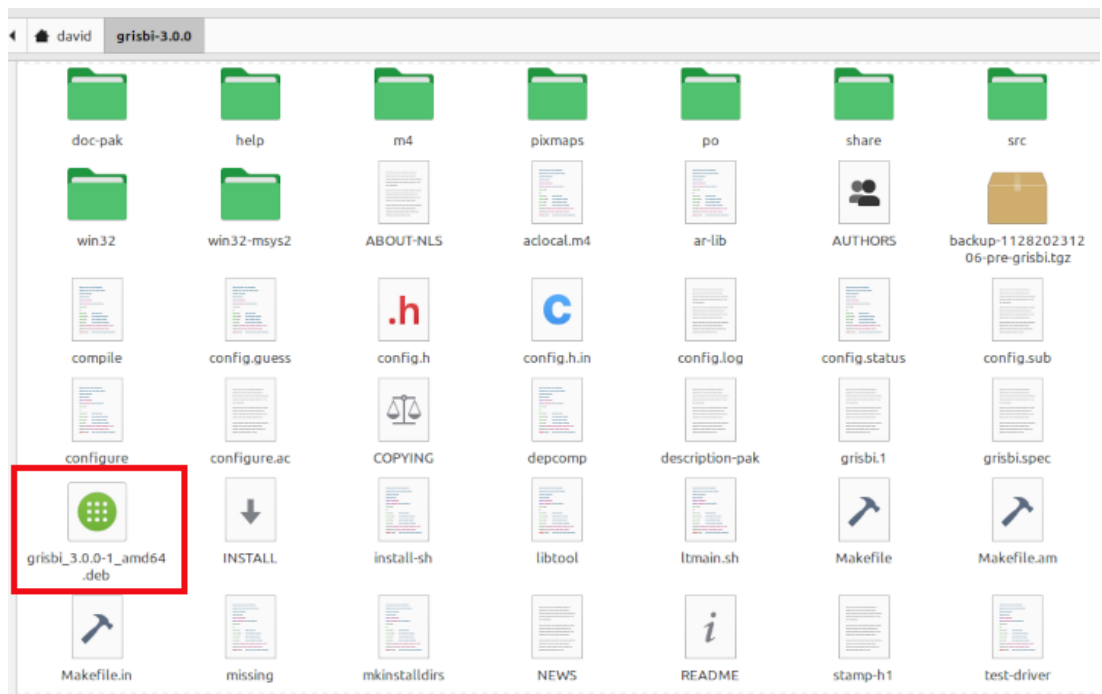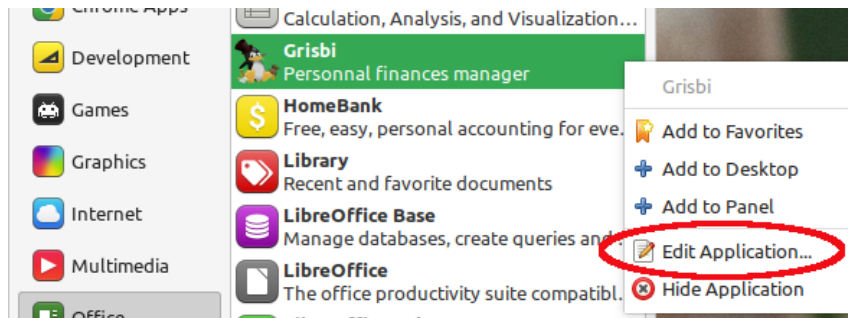

7 Launch Grisbi

Grisbi's entry appeared in the Xfce Whisker menu (under the **Office** section) almost immediately.
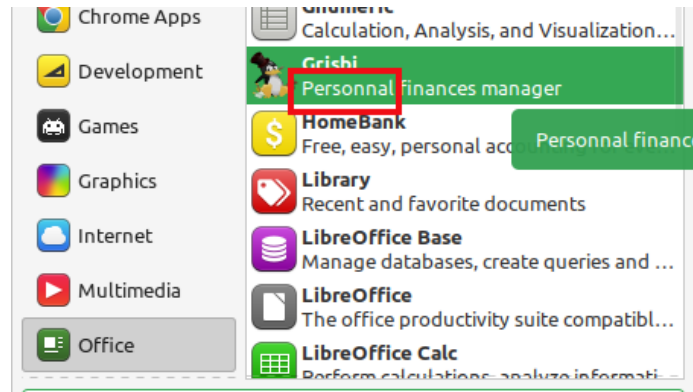
8   Edit the Menu Entry to Fix Typo

Unfortunately, there is a minor typo in Grisbi's package description in the Xfce menu: "Personnal [sic] finances manager." If that misspelling bothers you, it is easy-peasy to fix.[4]

- Right-click on Grisbi's menu entry.

- Choose the option "`Edit Application`":

- In the **Edit Launcher** dialog, in the **Comment** box, delete the extra " n " in "Personnal":

---

[4]If you are using the Cinnamon DE, you can fix this mistake using the Menu Editor. My previous tutorial, "Compiling Grisbi from Source on Mint 20.3" (July 2022, downloadable from `https://tinyurl.com/38a4e3mm`), describes the process (p. 9–14).
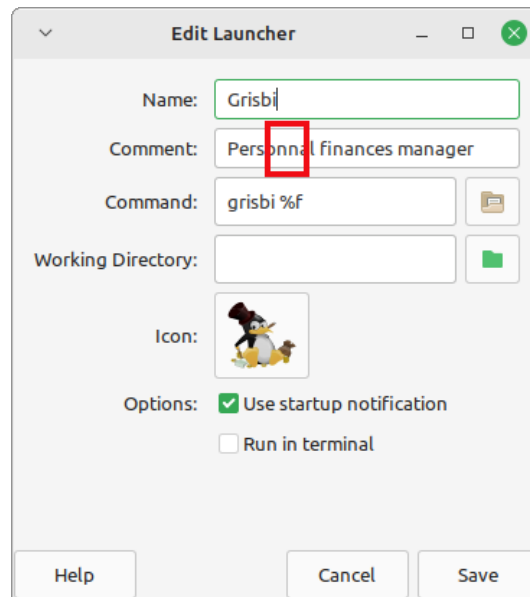
- Click on the Save button.

- Grisbi's description should now be spelled correctly:
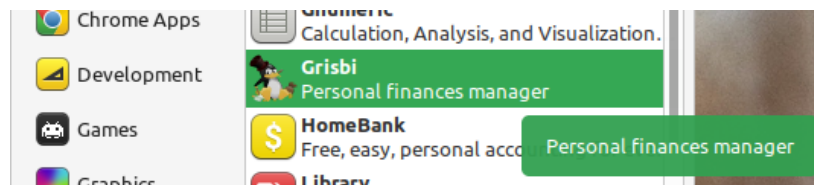
Good luck compiling and Have fun with Grisbi!