

Compiling Grisbi from Source on Mint 19.3

By kalwisti

July 2022

Contents

1	Introduction	1
2	Download the Source Files	2
3	Move File to /home Directory and Extract It	3
4	Install Required Packages and Dependencies	3
5	Compile the Source Code	4
6	Launch Grisbi	8
7	Edit the Mint Menu	8

1 Introduction

The version of Grisbi in the official Mint 19.3 repositories is very outdated (1.0.2, released in 2017). The current stable version of the program is 2.0.5 (released in August 2021). As far as I know, neither a PPA nor a Flatpak is available, so if you want to run the most recent version, you must compile it yourself from source.

The process is not as intimidating as it sounds; I was pleasantly surprised that I was able to compile Grisbi 2.0.5 without much difficulty on Mint 19.3 (Cinnamon 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 2.0.5 ...

2 Download the Source Files

Go to Grisbi's SourceForge website

<https://sourceforge.net/projects/grisbi/files/grisbi%20stable/2.0.x/2.0.5/>

and download the `grisbi-2.0.5.tar.bz2` file (7.5 MB in size, modified on 2021-08-31).

Grisbi Files
Brought to you by: [benj2](#), [carcam](#), [cedric_grisbi](#), [gegeweb](#), and 5 others

Summary **Files** Reviews Support Wiki News Code

[Download Latest Version](#)
grisbi_2.0.5-1_amd64.deb (629.7 kB)

[Get Updates](#)

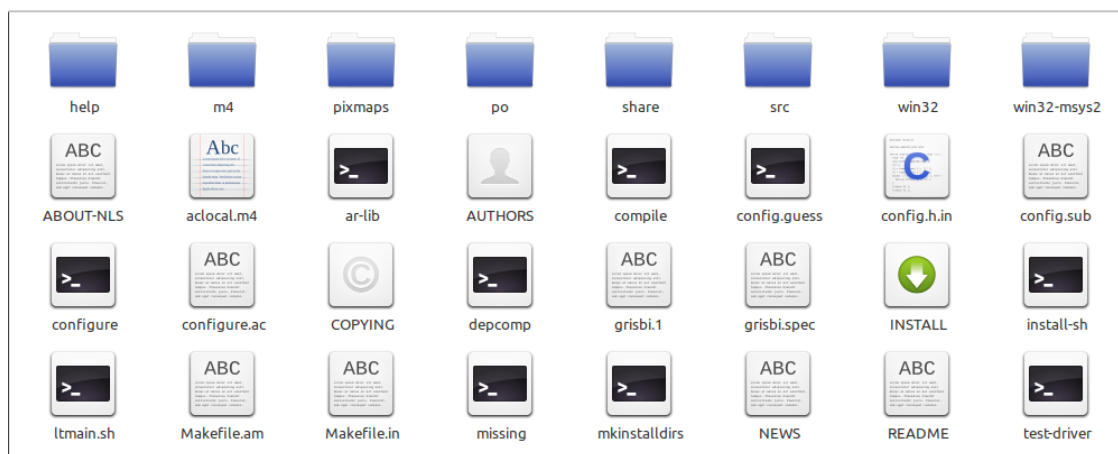
[Home / grisbi stable / 2.0.x / 2.0.5](#)

Name	Modified	Size	Downloads / Week
Parent folder			
Ubuntu 20.04	2021-09-06		41
Grisbi-2.0.5.dmg	2022-03-19	27.8 MB	17
Grisbi-64bit-2.0.5-setup.exe	2021-09-01	48.6 MB	255
Grisbi-32bit-2.0.5-setup.exe	2021-09-01	48.5 MB	6
grisbi-2.0.5.tar.bz2	2021-08-31	7.5 MB	22

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-2.0.5` which is 32 MB in size. That folder should contain 8 subfolders as well as 24 assorted files (illustrated below):



4 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	12.4
checkinstall	1.6.2-4
libgl1-mesa-glu	2.56.4-0
libgl1-mesa-dev	2.56.4-0
libgl1-mesa-glu-dev	2.56.4-0
libgl1-mesa-glu-bin	2.56.4-0
libgl1-mesa-dev	3.22.30-1
libgl1-mesa-dev	3.22.30-1
libgl1-mesa-dev	1.14.41-2
libgl1-mesa-dev	1.14.41-2
libgl1-mesa-dev	0.10.39
libgl1-mesa-dev	1:0.9.12-1
libgl1-mesa-dev	1.0.2n-1
libgl1-mesa-dev	2.9.4+dfsg1-6.1
libgl1-mesa-dev	0.51.0-5

**Note:**

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

5 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-2.0.5
```

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, 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 PC has an Intel Core i3-6100 CPU (3.7 GHz), with 16 GB of RAM).¹

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. (I left the majority of the fields blank; however, it 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]:
```

```
Preparing package documentation...OK
```

```
Please write a description for the package.
```

¹As a comparison, I compiled Grisbi 2.0.5 from source on another distro (Salix OS), on an old netbook (Intel Atom N455 CPU with 2 GB RAM). That process took approximately 25 minutes.

²<https://forums.linuxmint.com/viewtopic.php?t=230142>. 20 Sept. 2016.

End your description with an empty line or EOF.
>>

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

This package will be built according to these values:

```
0 - Maintainer: [ root@david-ThinkCentre-M800 ]
1 - Summary: [ Grisbi ]
2 - Name: [ grisbi ]
3 - Version: [ 2.0.5 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ amd64 ]
8 - Source location: [ grisbi-2.0.5 ]
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](#)

After you press the Enter key, a wall of text will scroll rapidly by, which should finish with the message: "Installation successful."

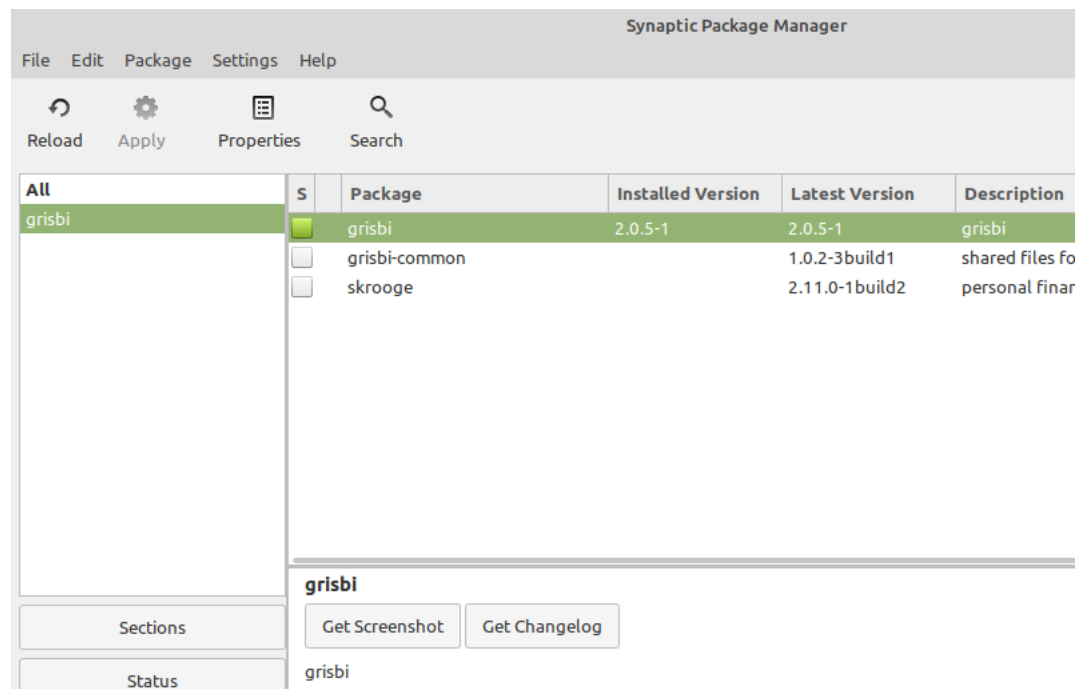
³Warren Post, "Creating Packages from Source with checkinstall," *A Maze of Twisty Little Passages* (blog), Jan. 9, 2010, <https://warrenpost.wordpress.com/2010/01/09/checkinstall/>.

When the checkinstall process completed, the Terminal displayed this final message:

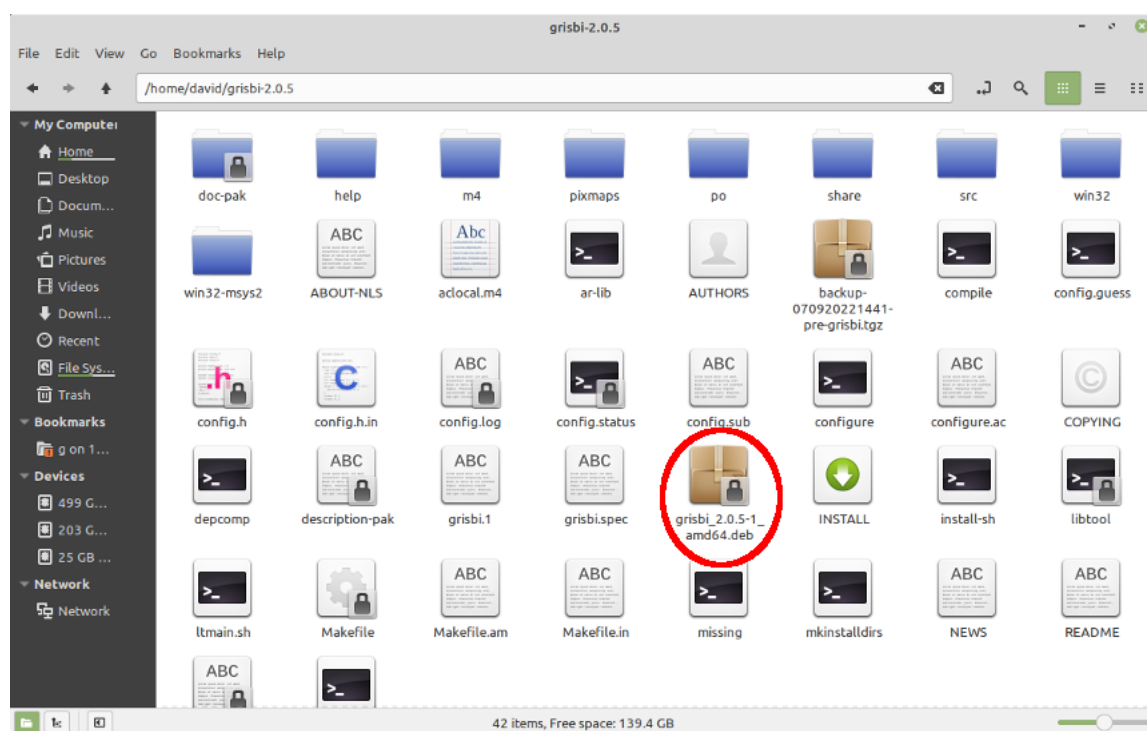
Done. The new package has been installed and saved to
/home/david/grisbi-2.0.5/grisbi_2.0.5-1_amd64.deb

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 had been installed:



In addition, a 64-bit .deb package is now present in my grisbi-2.0.5 source folder:



6 Launch Grisbi

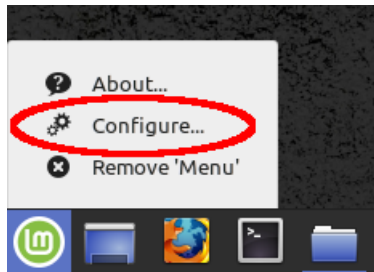
It took a few minutes for Grisbi's entry to appear in the Mint menu (under the Office section). Initially, I had to launch Grisbi from the Terminal. However, if you log out from your current desktop session, then log in back in, Grisbi should be listed in the Mint menu.

Grisbi's listing in the menu had the same icon as HomeBank: a small wallet. Although this is not a major problem, I wanted to try configuring the menu to use Grisbi's "in-house" icon: Tux in a top hat, smoking a cigar.

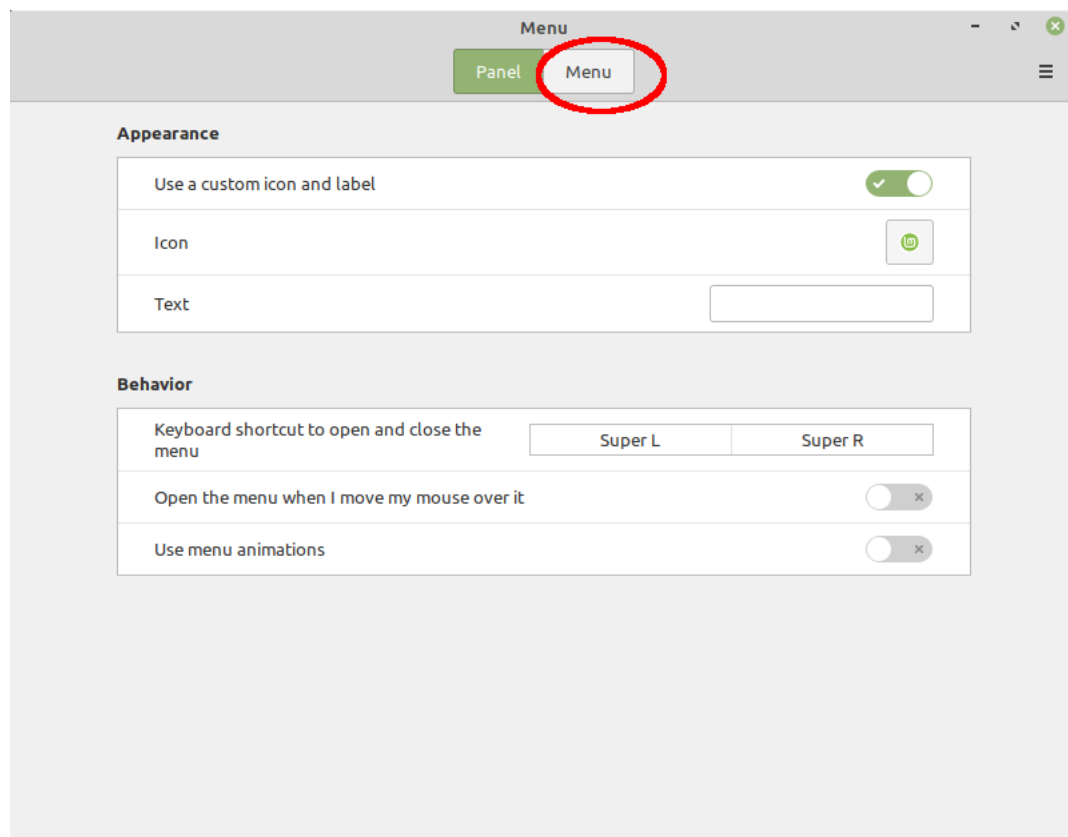
I had never tinkered with the Mint menu before, so I searched for some mini-tutorials on how to edit it. Let's proceed ...

7 Edit the Mint Menu

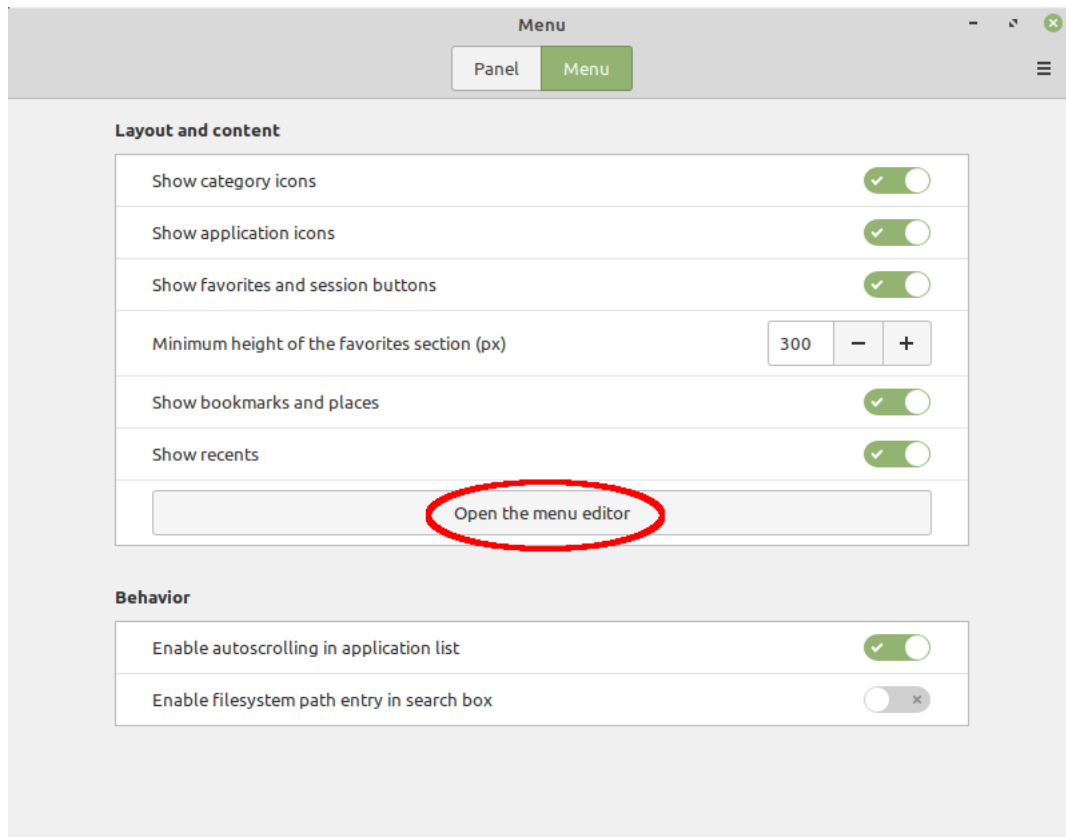
- Right-click on the Menu icon and select Configure:



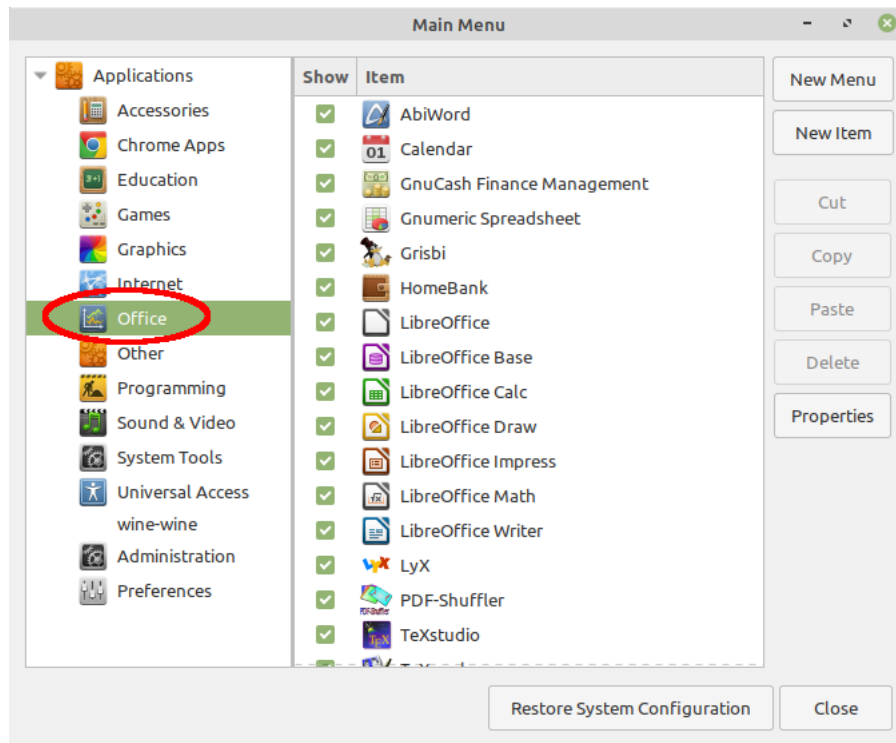
- In the new Menu dialog that appears, click on the Menu tab to bring it forward:



- Next, click on the Open the menu editor button:



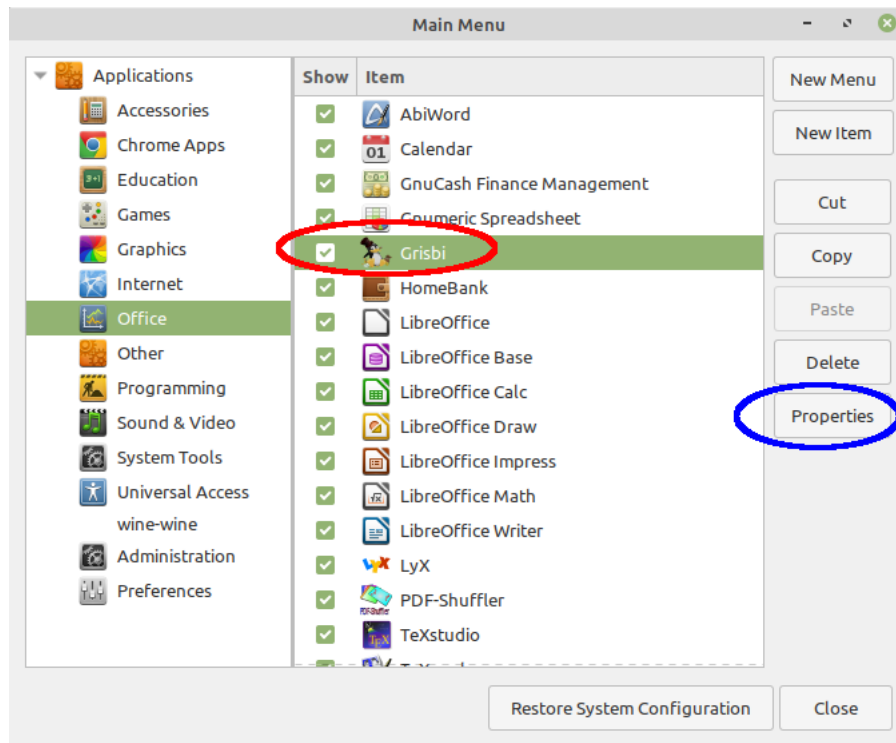
- An Editor window will appear. Click on the Office section and look for Grisbi:



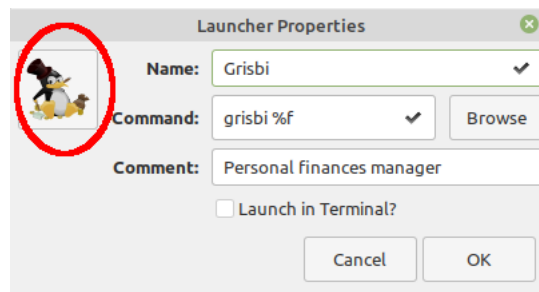
Note:

At first, Grisbi was assigned the same “wallet” icon as HomeBank. (The accompanying screenshots were taken after I successfully edited the menu entry).

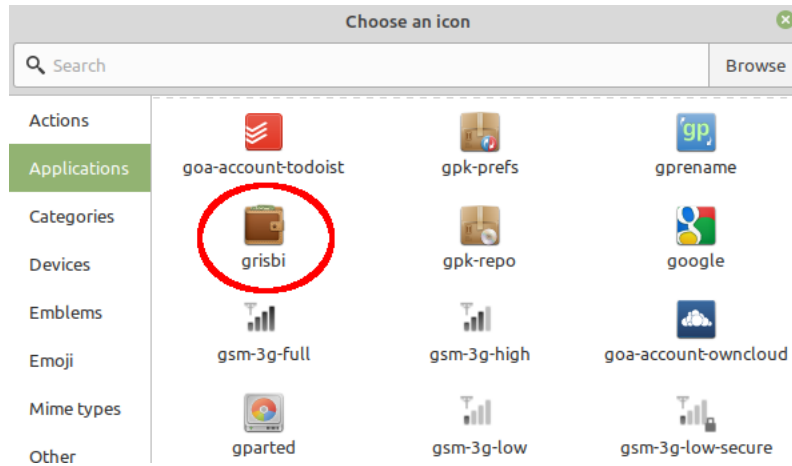
- If you wish to change this to Grisbi’s native icon, click on Grisbi’s menu entry to select it. Next, click on the Properties button:



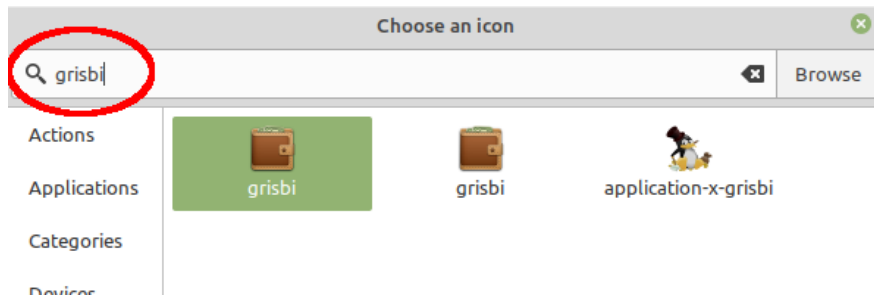
- In the Launcher Properties dialog, click on the rectangle with the current application icon:



- Under the Applications section, you can see the “wallet” icon:



- However, if you type “grisbi” in the Search box and press Enter, you will see the two different icon styles:



- Choose whichever icon you prefer and press the Select button.
- Click the Close button when you are finished.

I am not sure about the exact locations of these icons because they appear to reside in multiple directories, such as:

```
/usr/share/app-install/desktop  
/usr/share/applications  
/usr/share/icons/hicolor/scalable/apps
```

as well as in the Grisbi source folder itself:

```
/home/<yourusername>/grisbi-2.0.5/pixmaps
```

Good luck compiling and Have fun with Grisbi!

