

13 Software Management

1. 13.1 Introduction

antiX offers three complementary methods of software management:

- **Install Programs** which allows one-click installation/removal of apps selected from the distribution's standard repositories as useful and popular.
- **Package Manager (Synaptic)**, a complete graphical tool that allows a whole range of Debian package management operations.
- **Cli-aptiX**, a complete semi-graphic tool

Install Programs, compared to Synaptic, has a few advantages:

- It's much faster!
- The Popular Applications tab is limited to the most used packages, so they are easier to find.
- It installs some packages that would otherwise be complicated to install correctly (e.g. Wine).

Synaptic, on its side, has these strengths compared to Installa Programs:

- It allows you to install a lot more software, not just the popular apps.
- It allows you to set a large number of advanced filters, such as sections (categories), status, etc.
- Offers detailed information on specific packages.
- It makes it very easy to add new repositories.

Cli-aptiX (Command Line Package Manager), is a semi-graphical tool with similar functionality to Synaptic, more spartan but very fast:

- Being semi-graphical it is less attractive than Synaptic, but simple enough and equally powerful.
- Great for older PCs that are slowed down by the use of "heavy" software like Synaptic.

This chapter focuses on Synaptic and cli-aptiX since Install Programs is very intuitive and doesn't need a guide. Synaptic is the recommended method for beginners to manage software packages when they need to go beyond the functions of Install Programs. If Synaptic is slow Cli-atpiX may be a good choice.

Other available methods that may be indicated in certain situations will also be analyzed.

In particular, it is important to know how to use **Apt** through the terminal because it is very handy to install a sequence of software packages with a single command (if the sequence is long, with Synaptic and Cli-aptiX it would take much longer). **Gdebi** is also useful, if you need to install a Debian package (with adequate features), downloaded by other routes than the repositories (software archives) of this distribution.

Note. Beware, there may be malware and stability risks to downloading from sources other than the distro repos.

2. 13.1.1 Packages

Software management in antiX is done through the [APT](#) system: [Advanced Package Tool \(APT\)](#) which operates behind the scenes. Software comes in the form of a **package**: a discrete set of non-executable data where you will find installation instructions for the package manager to follow. Packages are stored on servers called **repositories** (repos), and can be viewed, downloaded, and installed using special client software called a **package manager**. The recommended package manager for antiX is Synaptic, but you can use apt from the command line if you prefer to use the terminal. For *.deb files that we will have obtained in a different way than downloading them from the repositories through Apt-Synaptic (maybe downloaded from some website or copied from some magazine CDs) we can use the graphical utility Gdebi that is started with a simple click on the package file.

Most packages have one or more **dependencies**, which means that they need one or more supporting packages to work. The APT system is designed to automatically handle dependencies for you; in other words, when you try to install a package whose dependencies are not already installed, the APT package manager automatically marks the dependencies for installation as well. However, it may happen (rarely) that these dependencies cannot be met, thus preventing the installation of a package.

3. 13.2 Repositories (Repo)

APT repositories (repos) are more than just websites with downloadable software. The packages on these repository-sites are specially organized and indexed to be accessible through a package manager, rather than browsing them directly.

WARNING: Don't add more repositories randomly on antiX Linux! This is especially true for Debian Sid, which is very likely to break the installation of packages with no possibility of repair.

4. 13.2.1 Standard repos

antiX Linux comes with a number of enabled repositories that provide security and a wide choice of software. If you are new to antiX (and especially if you are new to Gnu/Linux), it is generally recommended that you use the default repositories at first.

For security reasons, these repositories are digitally signed, which means the packages are authenticated with an encryption key to make sure they are authentic. If you install packages from non-Debian repositories without a key, you'll get a warning message that they can't be authenticated.

To get rid of this warning and make sure your installations are secure, you need to install the missing keys. You can get directions on how to do this a little later in the sub-section at the end of this paragraph. Repositories are easily added, enabled/disabled, removed, or modified via Synaptic, but can also be edited manually by editing files in `/etc/apt/` in a root terminal. In Synaptic, select **Settings> Repositories**, then click the New button and add the information. The repository information is expressed in a single line, like this:

```
deb http://it.mxrepo.com/antix/bullseye/ bullseye main nosystemd nonfree
```

Be careful to correctly enter the spaces that separate the information in this row into four parts. These 4 parts that make up the repository's information row are defined in the 4 fields at the bottom of the window that opens by selecting in Synaptic **Settings> Repository**

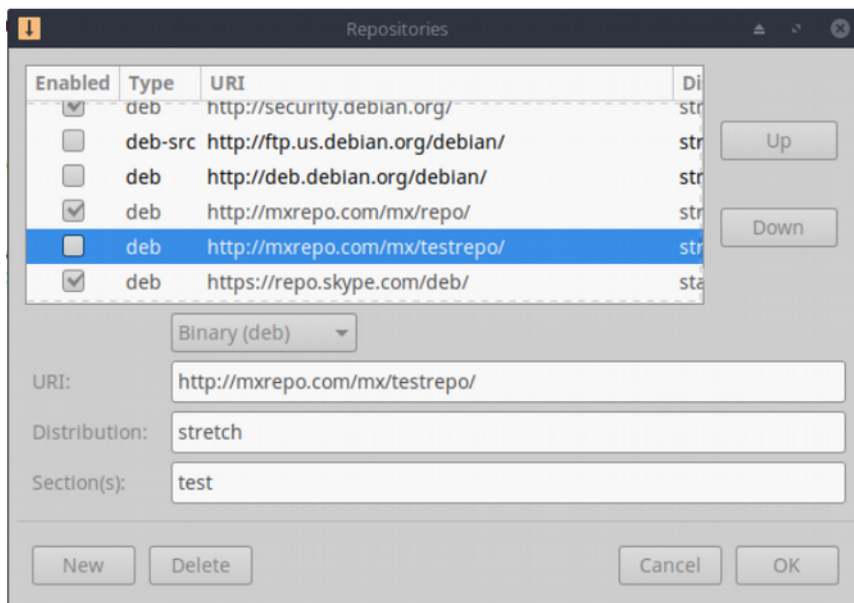


Figure: Repository

Some repositories carry special labels depending on the type of packages they contain:

- **contribute**, depend on or are package-accessory to the non-free packages.
- **non-free**, do not meet the Debian Free Software Guidelines (DFSG).
- **security**, contain only security-related update packages.
- **backports**, contain packages from the repositories of the most up-to-date versions of Debian that have been recompiled to be compatible with Debian Stable (the version of Debian on which antiX is built) to keep your operating system up-to-date.
- **antix**, contain the special, distribution-specific packages that make antix what it is.

The address of the antiX repo, can be seen here <https://antixlinux.com/antix-19-packages/>

If there is a change in the official repositories, it is usually communicated in the antiX forum

<https://www.antixforum.com/>

5.

6. 13.2.2 Dedicated Repositories

In addition to the general repositories such as Debian and antiX-specific repositories, there are also a number of repositories associated with a single application. By adding one, either directly or via Synaptic, you can then receive future updates for that software. Some are already preloaded but not enabled, others you can add yourself.

Here is a classic example (VirtualBox):

```
deb http://download.virtualbox.org/virtualbox/debian/stable contrib
```

New users coming from Ubuntu or a derivative often ask about PPA repositories; Ubuntu is quite far from standard Debian, so such repositories should be treated with caution. See the [MX/antiX Wiki](#).

7. 13.2.3 Development Repositories

There is a category of repository that allows you to capture the most recent (and therefore less stable) development of an application. This can be achieved through the [Git](#) portal, which can be consulted by the end user to stay up-to-date on development through version control. The user can download and check on a local computer a copy of the source code of the application he is interested in. The [GitHub](#) portal is a convenient method for managing projects using Git and antiX Linux maintains most of its code in its GitHub repository.

More info: [Wikipedia: software repositories](#)

8. 13.2.4 Mirrors

The antiX repositories for both software packages and ISOs are stored at sites hosted on different servers around the world, the same being true for Debian repositories. These sites (called mirrors because they reproduce exactly the same contents as an original site) allow providing multiple sources of the same information, and are intended to reduce download time, improve reliability, and provide some resilience in case of server failure. During installation, the server-mirror from which you download the software will most likely be automatically selected based on your geographic location and language. But you may have reason to prefer a different one:

- In some cases the automatic assignment during installation may be wrong.
- The user can change the residence.
- A new mirror may become available that may be closer, faster, or more reliable.
- An existing mirror might change its URL.
- It may no longer be online.

antiX Repo Manager (Control Center → Software tab → Repo Manager) makes it easy to change the mirrors you use to download software, allowing you to choose the one that works best for you.

The easiest way to find it is to use the button that selects the fastest mirror for your location. In the "Debian Repos" and "Individual Sources" tabs, you can enable/disable the repositories in the source list.

9. 13.2.5 Obtaining Repo Authentication Keys.

Sometimes, after a new repo has been added, or if there has been a change in the repositories from which our software managers download packages, a message may appear warning that packages cannot be authenticated. To get rid of this warning and make sure your installations are secure, you need to install the missing keys.

You can get the authentication key **by simply installing the package that contains it**. Most repositories allow you to install a special package with the key. Search for "keyring" in Synaptic, and you will see a number of packages with names similar to these (which are among the most common debian key packages):

gnome-keyring

debian-archive-keyring

debian-multimedia-keyring

Checking in "Properties", on Synaptic's function bar, we can see if, among the selected ones, there is the key package we are looking for.

Another way to get the keys is to use the **SMXI** script. It is an interactive script that you start from the terminal but then continue from command prompt outside the graphical environment. When you ask for a software update, smxi will automatically do a search for repo keys.

A detailed description of how to use this script can be found in the appropriate chapter of this guide.

Note. The script will not work if the system is booted in Live mode.

Another option is to make a note of the message warning us that the key is missing.

Suppose the message notifying us of authentication failure looks like this:

"An error has occurred.

W: GPG error: https://mirror.noreply.org lenny Release: The following signatures could not be verified because the public key is not available: NO_PUBKEY CFF71CB3AFA44BDD"

This means that you have to authenticate the repo (in this case "noreply") by downloading the relative gpg key. To do this we use the alphanumeric number (in this case CFF71CB3AFA44BDD) that we see at the end of the message, which represents the **public "key id"** of that repository.

Since **gpg authentication keys** are usually published by the repository maintainer within special public servers for managing these keys, we will query this site by providing it with the public key id we copied from the warning message.

The command uses the following syntax:

```
$ gpg --keyserver [server name] --recv-keys [key id]
```

So, considering that one of the most used **server** as a repository for these keys is **hkp://keys.gnupg.net**, in the case of the repository shown before, the command to give will be:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys CFF71CB3AFA44BDD
```

Once obtained the gpg key from the server, it will have to be inserted into the key database of our system, with this command, this time with administration privileges:

```
# gpg --export --armor [key id] | apt-key add -
```

continuing with the example from before will be:

```
# gpg --export --armor CFF71CB3AFA44BDD | apt-key add -
```

Watch out for the hyphen at the end of the command!

At this point you have to do the update, from Synaptic or terminal, like this: `# apt-get update`

Note. If you can't get anything from the reported server, you can try changing servers, e.g.

```
HKP="--keyserver hkp://eu.pool.sks-keyservers.net" # European pool
```

```
HKP="--keyserver hkp://na.pool.sks-keyservers.net" # North American pool
```

```
HKP="--keyserver hkp://pool.sks-keyservers.net" # Global pool
```

```
HKP="--keyserver hkp://keys.gnupg.net" # Gnupg
```

```
HKP="--keyserver hkp://keyserver.ubuntu.com" # Ubuntu keyserver
```

10.13.3 Synaptic

The following section attempts to provide an up-to-date overview of the use of Synaptic.

This is a graphical tool that allows you to install, remove, update or get information about all software packages stored in the repositories. Please note that you will be asked for the root password and, of course, you need to be connected to the Internet.

11.13.3.1 Installing and removing packages

Installation

Here are the basic steps for installing the software in Synaptic:

- Click **Start> System> Synaptic Package Manager** , providing the root password if asked.
- Press the *Update* button. This button causes Synaptic to contact the online repository servers and download a new file-index with information regarding what packages are available, in what versions, and what other packages (dependency packages) are required for them to be installed. If you get a message that you were unable to contact any of the repositories, wait a minute and then try again.
- If you already know the name of the package you are looking for, just click on the box on the right and start typing; Synaptic will do an incremental search as you type.
- If you don't know the name of the package, use the "search" button at the top right to locate the software by name or keywords. This is one of the biggest advantages of Synaptic over other methods.
- Alternatively, use one of the filter buttons in the lower left corner:

- **Sections** provide sub-areas, such as editors, games and entertainment, utilities, etc. You will see a description of each package in the bottom pane, and you can use the different tabs to find more information.
- **State** groups of packages categorized by whether they are installed or not, to be updated, etc.
- **Source** will show the packages from each specific repository.
- **Custom filters** are given the ability to use search filters with various options.
- **Search through the results** will show a list of previous searches for the session Synaptic is in.
- Click on the empty box to the left of the package you want to install, then select "Install" from the menu that appears, the box will be marked with a view. If the package has dependencies, a notification will be shown and these additional packages will be automatically marked for installation.
- Some packages come with "Suggested" and "Recommended" packages which can be viewed by right-clicking on the package name. These are additional packages that add functionality to the selected package, and it is a good idea to take them into account by checking out their features and functionality to decide whether, and if so what, to install.
- Click Apply to start the installation. If you get a message like this: "You are about to install software packages that cannot be authenticated! " don't worry, it can be safely ignored.
- There may be additional steps to take - just pay attention to the messages and follow the instructions that accompany them until the installation is complete.

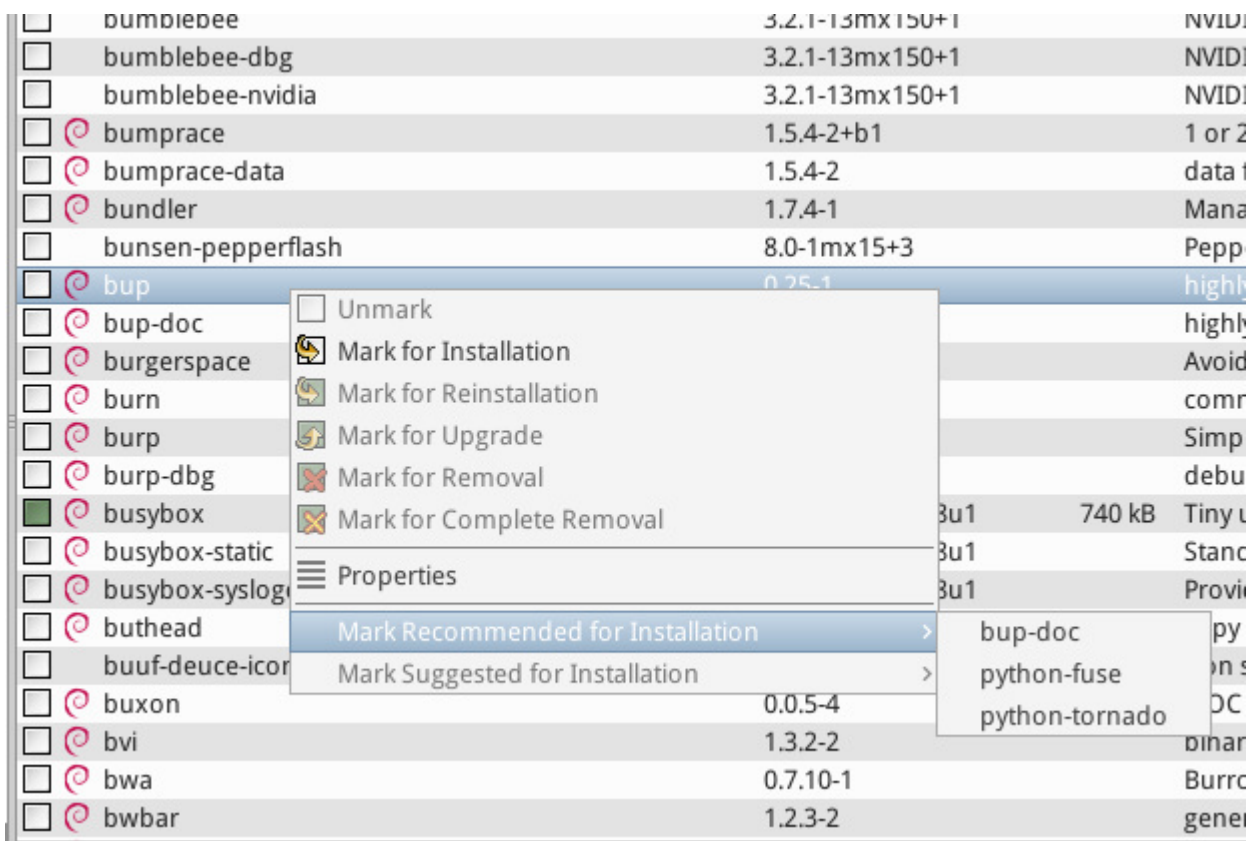


Figure: Checking recommended packages with a package installation.

Removal

Synaptic's removal of the software from your system is just as simple as installation, but there's more to it than meets the eye:

- To remove a package, simply click on the same box used for installation and select Remove or Remove completely.
 - Removal uninstalls the software, but leaves the system configuration files in case you want to keep the settings.
 - Full removal removes both the software and the configuration files for that software, however the strictly personal configuration files related to the package will not be removed. It is possible in Synaptic to check for residual configuration files by checking the "Not installed (residual configuration)" category found under "Status" in the right side panel.
- When there are other programs that depend on the package to be removed, the packages of these other programs must also be removed. These programs will not work once you remove the package that was acting as a dependency. This usually happens when you remove software libraries, or command-line applications that serve as the back-end (as the engine underlying a graphical application) to other applications. In these cases it's better not to remove that package, which was acting as a dependency of several programs, so be sure to carefully read the summary that Synaptic provides, with the list of packages that will be installed and those that will be deleted, before clicking OK.
- Removing large applications that are made up of many packages can lead to complications. Many times these packages are installed using a meta-package, which is an empty package that contains a list of packages and simply serves to install the entire set of packages needed for that application. The best way to remove a complicated package like this is to inspect the dependency list of the meta-package, and remove the listed packages. Be careful, though, not to uninstall a package that is also a dependency of another application that you want to keep instead! In the summary of packages to be removed, you can easily see if, in addition to the package of the program you want to uninstall and its support packages, packages of other applications you want to keep are also listed. Do not give the OK and repeat the procedure, one package at a time, deleting only those that do not give problems.
- You may find that several packages accumulate over time in the "Auto removable" status category. These were installed by other program packages which have since been deleted, and are no longer needed. It is therefore possible, and advisable, to click on this status category, highlight all packages in the right-hand pane, and then click on them to remove them. However, be sure to go through the list carefully, and also check it again when the summary window comes up, before giving confirmation because sometimes you may find that the dependencies listed for removal include some package that you actually want to keep. Use:
`apt -s autoremove`
to run a simulation (obtained with the `-s` option) if you are not sure.

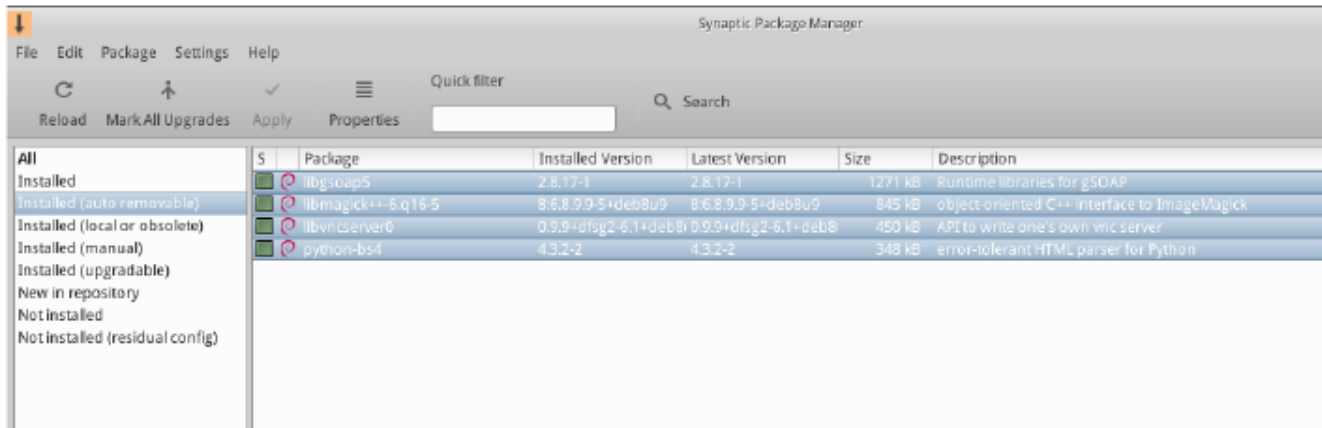


Figure: All set to delete removable car packages.

12.13.3.2 Upgrade and Downgrade

Repositories are subject to continuous updates, the number of packages in them can vary, but most importantly the level of update of each package varies. These repositories keep a list where changes in the update level of all the packages they contain are continuously recorded. Periodically you need to let your software manager know what the update level is of the total number of packages contained in the repos that have been chosen to be the source of your system's software. This allows you to quickly and conveniently keep your system up to date. So to do the update you will first have to update the list with the package information (update) and then proceed to download your packages that you can update. For what concerns the download/configuration phase of the packages you can decide that it happens in "upgrade mode" where if a certain package, with the last update would result to have dependencies to packages not yet installed, or that conflict with other packages present, the update will not be allowed, or you can choose that it happens in "dist-upgrade mode" where the system tries to solve in an intelligent way the conflicts between packages, installing the eventual further necessary packages and preferring those with higher priority. It is recommended to upgrade in dist-upgrade mode.

Upgrade - (upgrade/dist-upgrade)

Periodically, and every time you install a new package) you should check for updates.

There are several ways to go about updating:

- Using **antiX Updater**: Control Center → Software tab → antiX Updater
This is the fastest method because you don't have to wait for the software to boot up, load the package list, etc. A terminal window appears with the packages to be updated; review them carefully, then click OK to complete the process.
At the core of antiX Updater is a script that runs `apt update` followed by `apt dist-upgrade`.
- Use Package Manager (**Synaptic**).
 - Click the "Select Updates" button below the menu bar to select all packages available for update, or select the "Status" section in the left panel at the bottom, and then click the "installed (upgradable)" field at the top, to review packages or to select updates individually.
 - Click Apply to start the update, ignoring any warning message. At the beginning of the installation process, you have the option to look at the details of the process in a terminal within Synaptic.

- Using the **terminal**. Open a terminal and as a normal user give:
 - `apt update` and then `sudo apt dist-upgrade`
(the second command can be `sudo apt upgrade`, but `dist-upgrade` is recommended)

Note. When upgrading certain packages, a dialog may appear asking you to confirm to proceed in a certain way, or to enter configuration information, or to decide whether or not to overwrite a configuration file that comes with differences to the current configuration. Pay attention here, and follow the instructions until the update is complete.

Note. It's possible to insert an icon in the notification area of the panel that warns about the availability of updates. You have to download the package `apt-notifier` then you have to open with a text editor the file `~/.desktop-session/startup` and remove the comment `#`, in the line referring to the program `apt-notifier`. `apt-notifier` icon is not activated by default to avoid burdening the system because `apt-notifier` running in the background consumes about twenty MB of RAM.

Downgrade

Sometimes you may decide to restore the previous version of an application, for example because of problems that arose with the new updated version. This is easy to do in Synaptic:

1. Open Synaptic, provide the root password, and click the "Update" button.
2. Click on Installed, in the left panel, then in the right panel, find and highlight the package on which you want to downgrade
3. On the top menu bar, click Package> Force Version...
4. In the drop-down menu that appears, choose from the available versions, if any.
5. Click Force Version, then install in the usual way.
6. To prevent the latest version from being updated immediately, you need to lock it by going to the top menu bar and then *Package> Lock Version* as we will see in detail in the next point.

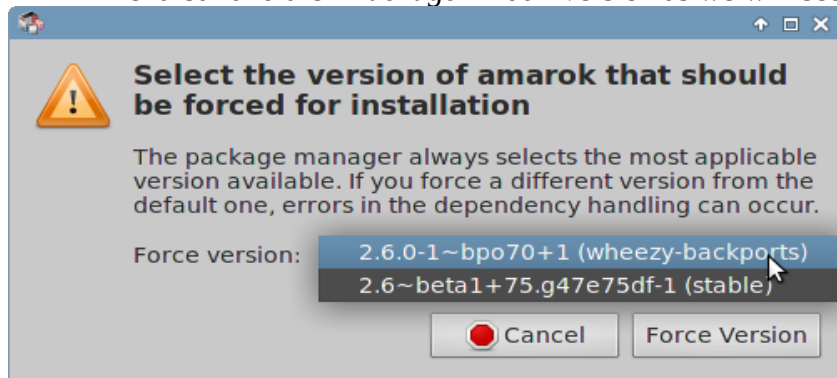


Figure: Using Version Force to downgrade a package

Locking

Sometimes it can be useful to lock an application to a specific version to prevent it from being updated in order to avoid problems with newer ones. This is easy to do:

1. Open Synaptic, provide the root password, and click Update.
2. Click on Installed in the left panel, then in the right panel find and highlight the package you want to block.

3. On the menu bar, click Package> Lock Version ...
4. Synaptic will highlight the package in red and add a lock icon to the box at the beginning of the package's row.
5. To unlock, highlight the package again and click Package> Lock Version (which will have a checkmark).
6. Note. Locking the package version via Synaptic does not prevent the package from being updated if you use the command line.

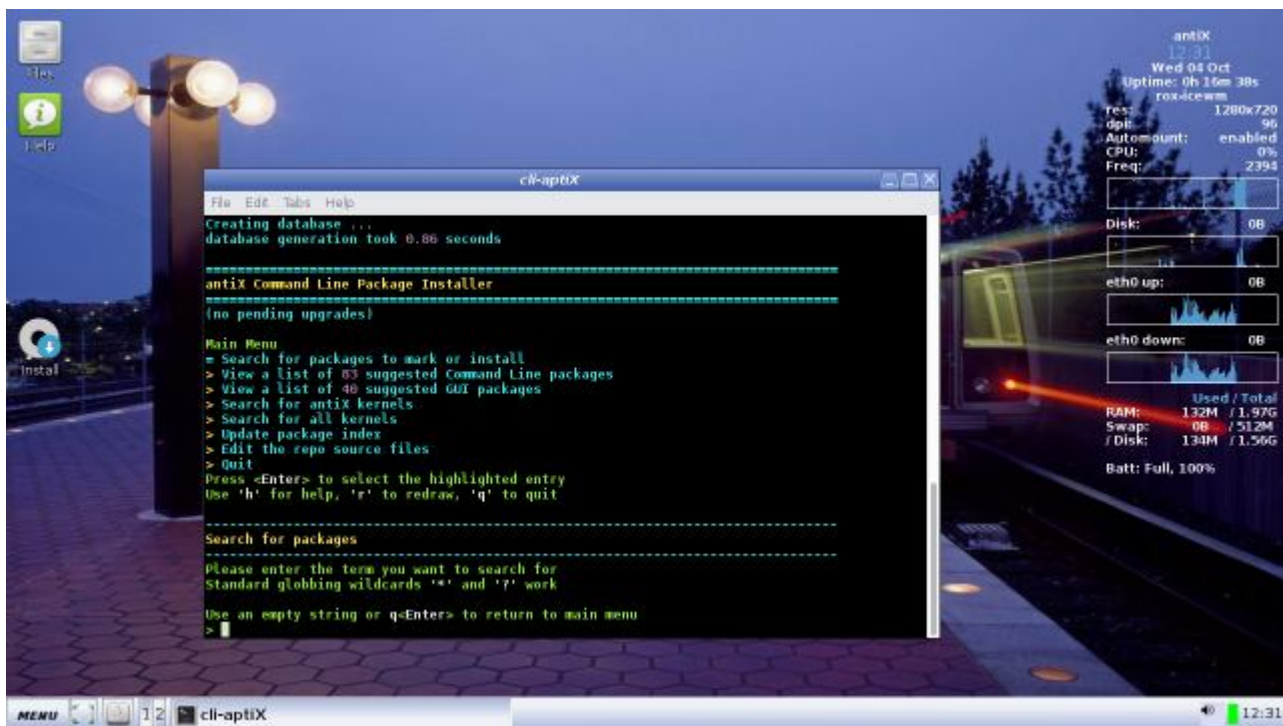
13.13.3.3 Troubleshooting

Synaptic is very reliable, but sometimes you may get an error message. A complete disquisition on these messages can be found in the [MX/antiX Wiki](#), here we just mention a couple of the most common ones.

- You may get a message saying that some repositories have failed to download their own information. This is usually a transient event, and you simply have to wait and then try again by clicking the "Update" button at the top (this corresponds to the *apt update* command) or you can use Repo Manger to change the server for that repository.
- If, while installing a package, you are notified that other software (other packages) you want to keep will be removed, click Cancel to cancel the operation.
- From time to time, packages may fail to install because their installation scripts fail one or more security checks; for example, a package may attempt to overwrite a file that is part of another package, or request to downgrade another package on which it depends. If a package installation or upgrade is stuck on one of these errors, it says that package is "corrupted". To fix this, click on the "Custom Filters" item in the bottom left panel and then on "Damaged" at the top. First highlight the package and then try to fix it by clicking Edit> Repair Damaged Packages. If you can't get the fix, right-click on the package to deselect or uninstall it.
- During installation or uninstallation, messages sometimes appear regarding the process with prompts to choose how to proceed:
 - Uninstall? From time to time, package dependency conflicts can cause the APT system to require uninstalling a large number of important packages in order to install some other package. This is rare with the default configuration, but becomes increasingly likely when adding unsupported repositories. **Be very careful** when installing packages that would require others to be removed! If a large number of packages are going to be removed, you may decide to look for another method of installing that application.

- Keep? During the upgrade, you will sometimes be informed that a new configuration file is available for a certain package, and you will be asked if you want to install the new version or keep the current version.
 - If the package in question comes from an antiX-repository, it is recommended to choose "install the maintainer version".
 - If not, answer "keep current version" (N), which is also the default choice.
- It may happen that you want to install packages with one software manager but you go to open a different one because it is more practical to check some things, then you go back to the initial manager leaving the second one open (e.g. Install Programs and Synaptic). The installation will not start, you will be warned that "...you *can not get an exclusive lock*" just close the second manager.

14.13.4 Semi-graphic method: cli-aptiX



This is a command-line wrapper based on "apt", "apt-cache" and related commands. It is designed to simplify searching for and installing Debian packages.

In some ways it is similar to the GUI program Synaptic. If you wanted something like Synaptic but running on the command line, simple and extremely fast, then "cli-aptiX" could be what you were looking for.

It can be reached from Menu-Start → Applications → antiX → **Command line package manager**.

15.13.4.1 Searching for packages

After the initial "apt update" (package information update), you will normally run a search for packages by name.

You can use "*" to match one or more characters and "?" to match single characters, but these are often not needed unless you want to match multiple parts of a package name, such as:

```
*linux-image*antix
```

which matches all antiX kernels. Most regular expression meta-characters are supported, but are rarely needed.

16.13.4.2 Search Results

For each search term entered, four different sets of results will be available: exact match to name, selective match to name, generic match to name, and match to description. After entering a search term, the number of results for each of the four search patterns will be shown before you have to select which pattern you want to see.

17.13.4.3 Exact name match

The most restrictive result set is the exact match to the name. Only package names that exactly match the search term are listed here. There may be more than one package in this list if you have used the wildcards "*" or "?" but normally there will be at most one match.

18.13.4.4 Selective Name Matching

Includes all package names that start with a match for your search term.

For example, if you use the search term "abc", you will get a list of selective matching results to the name that will look like:

- * 1) (I) abcde - A better CD encoding program
- * 2) abcm2ps - Translates to PostScript, ABC files of music description
- 3) abcmidi - Converts ABC files to MIDI and vice versa.

The "(I)" symbol on the first line indicates that the package is already installed.

Note that all three names *begin* with the search term "abc".

19.13.4.5 Generic name match

This list includes all packages with names that have a part that matches the search term. If the search term is "abc", the generic match results will look like the following:

- * 1) (I) abcde - A better CD encoding program

- * 2) abcm2ps - Translates to PostScript, ABC files of music description
- 3) abcmidi - Converts ABC files to MIDI and vice versa.
- * 4) berkeley-abc - ABC - A System for Sequencing ...
- * 5) grabc - simple program to determine the col ...
- * 6) libakonadi-kabc4 - Akonadi address book acc ...
- * 7) libkabc4 - bo address management library ...
- * 8) python-backports-abc - Backports of the "col ...

Note that this list contains all the package names from the previous list, plus **packages that contain "abc"** in their name, though not at the beginning but later.

20.13.4.6 Matching a name or description

This list contains all packages from the previous list plus any packages that contain the search term in the package description.

21.13.4.7 Selecting a package from a list of search results

As long as the entire list fits the screen, this is simple, just type in the number associated with the package you want to install. But if the list is too long to fit on the screen, you'll have to scroll through the list using "Up Arrow" and "Down Arrow" and "Page-Up" and "Page-Down" to scroll through the list (and also "Left Arrow" and "Right Arrow" to scroll sideways). Scroll through the list so that the package you want to select is at the bottom of the screen, then press *q* . You will be prompted to enter the number for the package you want.

22.13.4.8 Installing a package

Once you have selected a package from the search list, you will be given the option to install it. If the package has already been installed, you will be given additional options (under development).

23.13.5 Deb packages

Software packages installed via Synaptic (and APT, which is its "engine") are in a format called deb (short for Debian, the Linux distribution that invented APT). You can manually install the *downloaded* deb packages, using the graphical tool **Gdebi** or the command line tool **dpkg** . These are simple tools to install local deb packages.

NOTE: If the dependencies cannot be met, you will receive a warning and the program will stop.

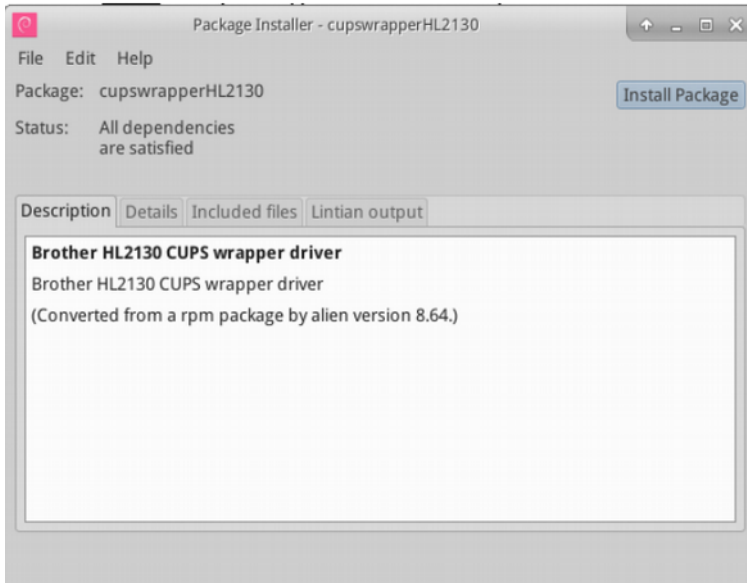


Figure: Gdebi ready to install a package.

Installing deb files with Gdebi

1. Get to the deb package you want to install (usually located in the "Downloaded" folder) and click on it. Gdebi will open the installation window.
2. Click Install.
3. Enter the root password when prompted.
4. Gdebi will attempt to install the package, and report back with the result.

Installing *.deb files with dpkg

1. Navigate to the folder that contains the deb package you wish to install.
2. Click on an empty space in that folder, from the menu that opens choose *Open Terminal Here* and become root. Alternatively, right click on the folder containing the deb package and choose *Open Terminal Here* then become root.
3. Install the package with the command (replacing the real package name, of course):
`dpkg -i packagename.deb`
4. If you need to install multiple packages in the same directory at the same time (for example, manually installing Libreoffice) you can do it all at once using:
`dpkg -i *.deb`

NOTE: In a shell command, the asterisk in the argument is a wild card. In this case, it will cause the program to apply the command to any file whose name ends with *.deb*.

5. If there are required dependencies that are not already installed on your system, you will get an unfulfilled dependencies error, since dpkg does not download them automatically. To correct these errors and complete the installation, run this command to force the installation:
`apt -f install`
6. Apt will attempt to remedy the situation by installing the necessary dependencies (if they are available in the repositories), or by removing the .deb file (if the dependencies cannot be installed).

NOTE: The command used in step 5 reflects the change in installation commands from the old `apt-get` to the current `apt`

24.13.6 Command Line Method: Apt

You can also use the command line to install, remove, update, change repositories and generally manage packages. Instead of launching Synaptic to perform the most common tasks, for example, many users prefer to open a terminal, become root and use one of these commands (administrator privileges are required).

Table: Common commands for package management

Command	Action
apt-cache search program name, package name or string to search for	Search for packages by name, description, etc... This command searches for the keyword "programname" in package lists, including package names and descriptions. You can use multiple keywords, such as "apt-cache search text editor" to get a list of text editors.
apt-cache show package name	Displays a brief description of the package we requested, providing version, usage, size, list of packages it may depend on, and other useful information. Found a program that looks interesting thanks to "apt-cache search", you can view detailed information about it using this command.
apt install package name	Installing a particular package
apt remove package name	Remove a given package (without removing the data configuration in /home)
apt purge package name	Completely remove a particular package
apt autoremove	Clean up remaining packages after removal (dependencies no longer needed)
apt update	Update the list of packages in the repositories
apt upgrade	Install all available updates
apt dist-upgrade	Intelligently handles dependencies that change with new package versions
apt -f install nomepacch. (apt --fix broken install)	This is used when you get an unresolvable dependencies error, even though the package you want to install is in the repositories. This error may occur, for example, if you add a new repo to your source list that contains updated versions of packages already in other repositories. This command should solve your problems.
apt --fix missing install package name	If you run into problems and the installation process aborts without completing (i.e. the package was cached but not installed) this command will allow you to put things back in place.

Examples:

Before attempting to install anything, ALWAYS update the list of available packages, if you haven't done so recently:

```
$ sudo apt update {Enter}
```

Search for the application you want to install:

```
$ apt-cache search [application_name] {Enter}
```

If necessary, scroll down until you see the package you want to install and copy it.

```
Example: $ apt-cache search vlc {Enter}
```

Install the application:

```
$ sudo apt install [package_name] {Enterprise}
```

```
Example: $ sudo apt install vlc {Enter}
```

or, just to be on the safe side, you can install all the recommended packages as well:

```
$ sudo apt install -install-recommends [package_name] {Enterprise}
```

```
Example: $ sudo apt install -install-recommends vlc {Enter}
```

25.13.7 Other methods

26.13.7.1 Self-sufficient packages

Appimages, flatpaks and snaps are self-contained packages (they already contain the libraries and other dependencies they need) that do not need to be installed in the usual way.

They are "**universal**" **application** formats, which means they should work on almost any Linux system.

Program packages installed through traditional Distribution Software Managers are downloaded from repositories that are controlled and secure. These packages, if not downloaded from secure servers, could be dangerous and introduce malware.

They can be huge because they include almost everything they need to work, this means that if an application of this type needs a certain library, the library is already present in the system that other programs access, but it will be replicated that library every time it is needed by a program of this type.

One of their great advantages is that any additional software they need is included, and therefore will not have a negative impact on the software already installed. But it should also be mentioned that they are much larger than traditional installed packages. Installing a lot of them will bloat your system.

In a system like antiX that wants to be thrifty with resources they are to be avoided if they are not really essential.

- **Appimages**: Simply download the file, make it executable and run it. For example: open SpaceFm, navigate to where your AppImage is, right click on the file - Properties - Permissions - Make sure the executable field is selected. You only need to do this once. After that, whenever you want to run the app, just right click and choose the run option. No installation is required.

Pros: one app, one file. If you want to remove the app, just delete the file. It contains almost all the libraries it needs, so you don't risk installing something that interferes with the system libraries. The worst that can happen is that the app needs some dependency that isn't available and so it won't work.

Cons: You have to manually make the Appimage file executable after getting it - this can be inconvenient, but for security reasons it has to be done at least once per AppImage.

Some AppImages do not offer to be automatically added to the menu. Some do not alert you to updates. It can be difficult to get files to run automatically with an AppImage app (file association issues easily fixed)

They start a bit slower, especially on slow machines because they need to be unpacked. If this is a problem, you can extract the package and simply run the application directly.

If you are looking for Appimage apps you can find them here: <https://appimage.github.io/> and here: <https://www.appimagehub.com/>, in some cases you can download them directly from the website of the app in question.

- **Flatpaks**: First you have to install the app. flatpak itself, and then install the apps prepared in flatpak format.

The flatpak file you download to get an application you want, is initially small in size. Then, the moment you run the flatpak to install it, it connects to the server and performs the necessary library transfers/downloads.

Pros: on the user side, you have to install a single file. Also in this case there are no risks to make the system unstable for the inclusion of incompatible libraries, eventually it could happen that the app needs some dependency that is not available and therefore will not work.

Cons: On first startup, flatpak needs to install many hundreds of MB of dependencies, and only after that will it be possible to install applications in the flatpak format.

Flatpak apps can give some integration issues with (for example: sometimes they don't match system decorations, etc).

It can be a little tricky to know how to run apps that aren't added to the menu.

If you are looking for Flatpaks apps you can find them here: <https://flathub.org/home>.

- [Snaps](#): Less reliable on antiX Linux because it doesn't use systemd.

HELP: [MX/antiX Wiki](#)

27.13.7.2 Other installation methods

Sooner or later some software you want to install will not be available in the repositories and you may need to use other installation methods. These methods include:

- **Blobs.** Sometimes what you want to install is not the classic .deb opensource package that you can install using the methods described above, but a collection of pre-compiled data provided as binary data stored as a single "blob" or precompiled entity. Such blobs are usually closed-source and typically go into the /opt directory. Common examples include addons for Firefox, Thunderbird and LibreOffice.

For example, to install the latest version of LibreOffice:

- Search the web for "libreoffice latest version download". Click on :
<https://www.libreoffice.org/download/download/>
- Select the desired version, based on the type of operating system (e.g. Linux x64) and the type of package format (deb).
- Click on the download button; since I use it all the time, I'm making a small donation.
- Once the download is complete, go to the Downloads folder and right-click on the LibreOffice_6.2.3 archive > Extract here.
- Click on the extracted folder, then right-click on the DEBS folder > Open Terminal Here.
- Enter this command

```
sudo dpkg -i *.deb
```

This will install it and make it available in the Office category of the start menu.

- You will probably want to remove the old version. You can do this by using Synaptic to select and remove libreoffice-core, this package will take the other packages with it. Be sure not to remove the new one!
 - Create a launcher by opening the Start menu and right-clicking on LibreOffice > Add to Panel (or: Add to Desktop).
- **RPM packages:** Some Linux distributions use the RPM packaging system. RPM packages are similar to deb packages in many ways, and there is a command-line program available in antiX to convert

RPM packages to DEB called **alien** . It is not installed by default in antiX, but is available in the repositories. After installing it on your system, you can use it to install an rpm package with this command (as root):

alien package name.rpm -i

This will place a deb file with the same name in the same location as the rpm file, which you can then install in the ways described above. For more detailed information on alien, check out the wikipedia version of its man page. See link at the bottom of this page.

- **Source code (sources):** Any open-source program can be compiled from the programmer's original source code, if there is no other option. Under ideal circumstances, this should be a fairly straightforward operation, but sometimes mistakes can be made that require several skills to overcome. The "source" is usually distributed as a tarball (tar.gz or tar.bz2 file). The most convenient thing to do is to make a forum request for that package, or follow the link to a tutorial on compiling programs.
- **Miscellaneous:** Many software developers provide software packages, which are usually distributed as tarball or zip files, in their own way. They may contain installation scripts, or be binary files ready for installation, or contain binary installers similar to the Windows setup.exe programs. In Linux, the installer of these programs often ends up in .bin. Google Earth, for example, is often distributed this way. If in doubt, consult the installation instructions that came with the software.

For compressed binaries like ".tar.bz" just unzip and run the executable file.

For example, you can get the latest version of Mozilla Firefox in this format.

Easy to install and safe to uninstall - simply delete unzipped files.

If you don't download from a secure server it can run foreign software, contain malware, etc.

You must manually configure menu items and file associations.

App ".run": These are almost the same as .tar.bz binaries, but they automatically install everything the app needs to run as soon as you run this type of auto-installer. Some device drivers have this format.

Again there can be dangers of installing malware etc.

- **Use Windows applications:**
Windows software won't work on Linux, but you can try using a program that sets a compatibility level to try to make it work - The program is called **WINE**. There are thousands of Windows applications that run on Wine, some even faster than Windows itself!

- Install WINE (Menu > Control Center > System > Install Programs, then go to Miscellaneous tab → Wine)
 - Set up WINE. If you want, you can install the *q4wine* and/or *playonlinux* packages via Synaptic - these packages help you set up Windows.
 - Open the File Manager (SpaceFM for example), navigate to where you have your Win 98/XP/7 applications. Right click on the ".exe" file you want to run and select the option to use WINE.
- A more compatible but slower option to legally run Windows applications is to boot to a virtual machine (e.g. Virtualbox) MS Windows and use it in antiX. This takes up a lot of storage space and is slow. Weak computers probably can't even run an MS Windows virtual machine due to CPU and RAM constraints.

28.13.8 Links

- [Wiky MX/antiX: s errorsynaptic](#)
- [Wiki MX/antiX: Software installation](#)
- [Wiki MX/antiX: Compilation](#)
- [Gdebi](#)
- [StDebian package management tools](#)
- [Debian APT Guide](#)
- [Guide to Alien](#)