5 Software Management

5.1 Introduction

5.1.1 Methods

Synaptic is the recommended method for beginners to manage software packages, although other methods are also available and may be required for certain situations.

5.1.2 Packages

Software operations in MX are accomplished through the Advanced Package Tool (APT) system. Software is provided in the form of a package: a discrete, non-executable bundle of data that includes instructions for your package manager about installation. They are stored on servers called repositories, and can be browsed, downloaded, and installed through special client software called a package manager. The recommended package manager for MX is Synaptic, though the command-line utility apt is also included for those who prefer it. The graphical utility Gdebi is launched for downloaded *.deb files with a single click on the file name; an alternative is to open a terminal and use the command dpkg -i packagename.deb

The majority of packages have one or more **dependencies**, meaning that they have one or more packages that must also be installed in order for them 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, your APT package manager will automatically mark those dependencies for installation as well. It can happen that these dependencies can not be met, preventing the installation of a package.

5.2 Repositories

APT repositories (repos) are much more than just web sites with downloadable software. The packages on repository sites are specially organized and indexed to be accessed through a package manager, rather than browsed directly.

5.2.1 Standard repos

MX Linux comes with a set of enabled repositories that offer you both security and choice. If you are new to MX Linux (and especially if you are new to Linux), it is recommended that in general you stick with the default repositories at first. For security reasons, these repositories are digitally signed, meaning that packages are authenticated with an encryption key to make sure they are authentic. If you install packages from non-Debian repos without the key, you will get a warning that they could not be

authenticated. To get rid of this warning and make sure your installations are secure, you need to install missing keys using Check Apt GPG (MX Tools).

Repositories are most easily added, removed, or edited through Synaptic, though they can also be altered by hand by editing the files in **/etc/apt/** in a root terminal. In Synaptic, click **Settings > Repositories**, then click the button New and add the information. The repo information is often given as a single line, like this:

```
deb http://main.mepis-deb.org/mepiscr/mx-test/ mx-16 test
```

Be careful to note the location of the spaces, which separate the information into four chunks that are then entered into separate lines in Synaptic.

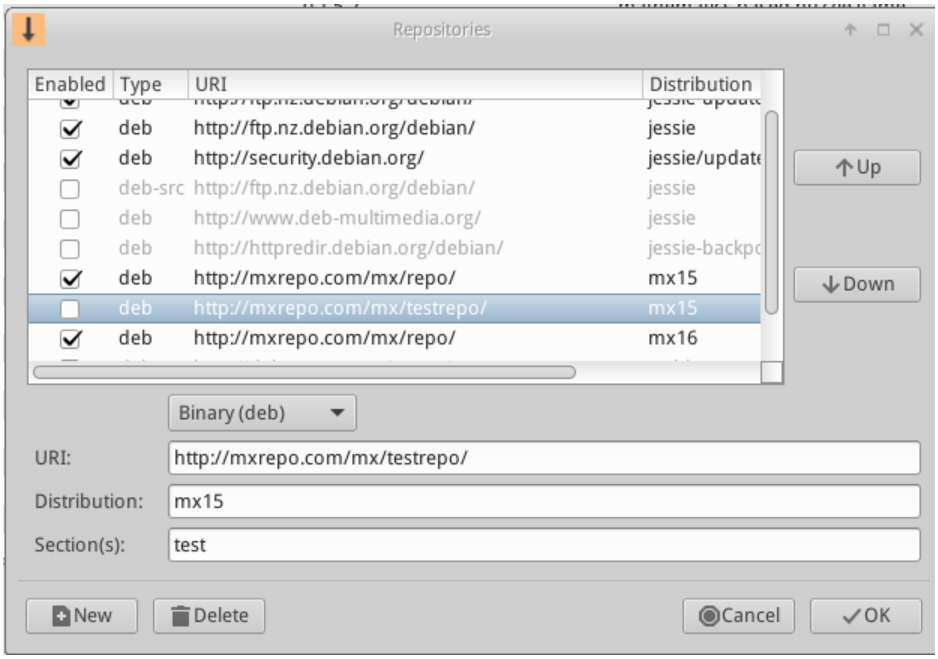

Figure 5-1: Repos, with the main MX-16 test repo highlighted

Some repositories carry special labels:

- **contrib**, which depend on or are accessory to non-free packages.
- **non-free**, which do not meet the Debian free software guidelines (DFSG).
- **security**, which contain security-related updates only.
- **backports**, which contain packages from newer versions of Debian that have been backwards compiled for Debian Stable (the version MX is built with) to keep your OS up-to-date.

- **MX**, which contain the special packages that make MX what it is.

The current list of standard MX repositories is kept in the MX/antiX Wiki.

5.2.2 Community repos

MX Linux has its own Community Repos with packages that our Packagers build and maintain. These packages are distinct from official MX packages coming from Debian Stable, and may have been incorporated from Debian versions in development (testing or even experimental), from antiX developments or from other sources. The Community Repos are critical to MX Linux, since they permit an OS based on Debian Stable to stay abreast of important software developments.

The purpose of the MX Test repo is to get feedback from users before the packages are moved to MX Main. The easiest way to install from MX Test is with the MX Package Installer (Section 3.2.14), as it handles many steps automatically.

To find out more about what is available, who the packagers are, and even how to get involved, see MX Community Packaging Project.

5.2.3 Dedicated repos

In addition to the general repositories such as Debian, MX, and Community, there also exist a certain number of dedicated repositories associated with a single application. When you add one of them, either directly or through Synaptic, then you will receive updates. Some are preloaded but not enabled, others you will add yourself.

Here is a common example (VirtualBox):

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

5.2.4 Development repos

One final category of repository exists for acquiring the most recent (and thus least stable) build of an application. This is done through a version control system such as Git that can be used by the end user to stay current with development. The user can check out a copy of the application source code into a directory on a local machine. The portal GitHub is a convenient method of managing projects using Git, and MX Linux keeps most of its code in its GitHub repo.

More: Wikipedia: Software repository

5.2.5 Mirrors

MX Linux repositories for both packages and ISOs are “mirrored” on servers at different sites around the world. These mirror sites provide multiple sources of the same information, and function to reduce download time, improve reliability, and provide a certain resiliency in case of server failure. During installation, the most likely mirror will be automatically selected for you based on location and language. But the user may have reasons to prefer another:

- the automatic assignment at installation may be wrong in some cases
- the user may change residency
- a new mirror may become available that is much closer, faster or more reliable
- an existing mirror may change its URL the mirror being used may go offline

MX Repo Manager (Section 3.2.15) makes it easy to switch mirrors, enabling you to choose the one that works best for you.

5.3 Synaptic

Synaptic is a friendly, easy-to-use frontend (GUI) to the APT packaging system. It is a graphical tool that allows you to install, remove, upgrade, downgrade, or get information on all the software packages available in the online repositories on your repository list. Note that your root password is required and, naturally, you will need to be connected to the Internet.

5.3.1 Installing and removing packages

Installing

Here are the basic steps for installing software in Synaptic:

- Click **Start menu > System > Synaptic Package Manager**, supplying the root password if necessary.
- Hit the Reload button. This button causes Synaptic to contact the online repository servers and download a new index file with information on what packages are available, what versions they are, and what other packages are required for them to be installed. If you get a message that some of the repositories failed to be contacted, wait a minute and then try again.

- If you already know the name of the package you are looking for, just click in the pane on the right and start typing, and Synaptic will incrementally search for what you type.

- If you don't know the package's name, use the Search box in the upper right corner to locate software based on name or keywords. This is one of Synaptic's greatest advantages over other methods.

- Alternatively, use one of the filter buttons in the bottom left corner:

    - **Sections** provides subareas such as Editors, Games and Amusement, Utilities, etc. You will see a description of each package in the bottom pane, and can use the tabs to discover more information about it.

    - **Status** groups packages by their installation situation.

    - **Origin** will show packages from a specific repository.

    - **Custom Filter**s provides various filter options

    - **Search Results** will show a list of previous searches for the Synaptic session you are in.

- Click the empty box next to the package you want and select Mark for Installation. If the package has dependencies, you will be notified and they will automatically be marked for installation as well. You can also just double-click the package if it is the only one you are installing.

- Right-click the package name again, and check carefully the packages listed under Mark Recommended for Installation; Mark Suggested for Installation is worth looking at as well.

- Some packages also have "Recommended" and "Suggested" packages that can be viewed via right-clicking the package name. These are additional packages that add functionality to the selected package, and it is a good idea to look them over.

- Click Apply to begin the installation. You can safely ignore any warning message: "You are about to install software that can't be authenticated!"

- There may be additional steps: just follow the prompts as you receive them until the installation completes.

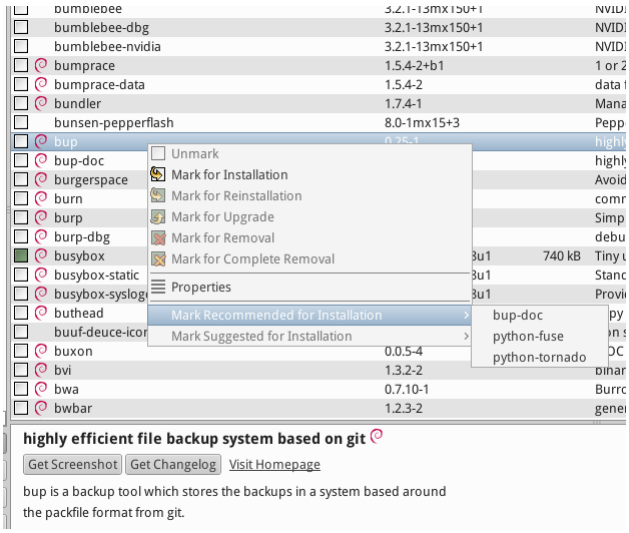

Figure 5-2: Checking recommended packages during package installation.

Removing

Removing software from your system with Synaptic seems as straightforward as installing, but there is more to it than meets the eye:

- To remove a package, simply right-click it and select Mark for Removal or Mark for Complete Removal.
    - Removal uninstalls the software, but leaves system configuration files in case you want to keep your settings.
    - Complete Removal removes the software and the system configuration files as well (purging). Your personal configuration files related to the package will not be removed. Check also for other configuration file remnants in Synaptic, category Not installed (residual config).
- When you have other programs that depend on the package being removed, those packages will have to be removed as well. This usually happens when you remove software libraries, services, or command-line applications that serve as back-ends to other applications. Make sure you read carefully the summary Synaptic gives you before clicking OK.
- Removing large applications that are composed of many packages can bring complications. Many times these packages are installed using a meta-package, which is an empty package that simply depends on all the packages you need for the application. The best way to remove a complicated package like this is to inspect the dependency list for the meta-package, and remove the packages listed there. Take care, however, that you don’t uninstall a dependency of another application you want to keep!

- You may find that the status category Autoremoveable begins to accumulate packages. These were installed by other packages and are no longer needed, so you can click on that status category, highlight all the packages in the right pane, and then right-click them to remove. Be sure to examine the list carefully when the verification box appears, because sometimes you may find that the dependencies listed for removal include packages you actually want to keep. Use ***apt -s autoremove*** to do a simulated (= the -s switch) dry run if you're unsure.

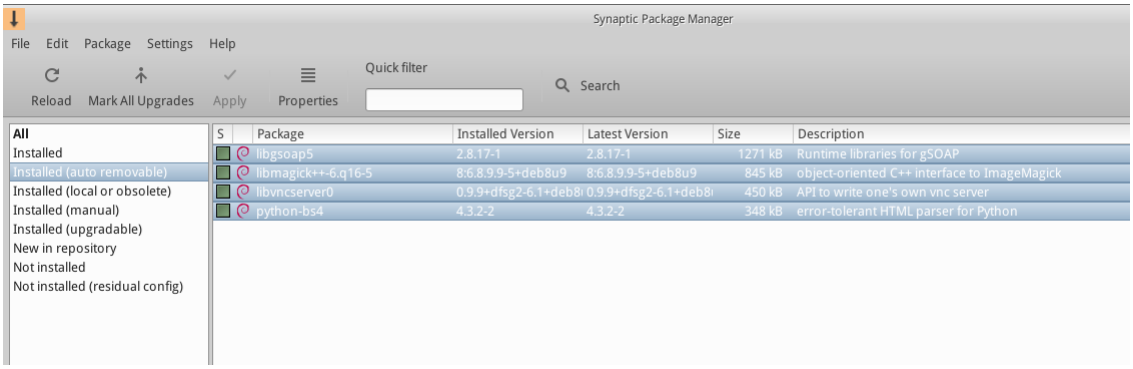

Figure 5-3: Getting ready to clear out the autoremovable packages.

5.3.2 Upgrading and downgrading

Synaptic enables you to quickly and conveniently keep your system up-to-date.

Upgrading

Unless you are using a manual method in a terminal, upgrading is typically triggered by the appearance of a green arrow on the Apt-notifier in the Notification Area. There are two ways to proceed when this arrow appears.

- Right click the Apt-notifier icon > Upgrade all packages. This is the faster method because there is no wait for software to load, run, etc. Examine the packages that are available for upgrade, then hit Return to complete the process.

- Left click the icon to open Synaptic

    - Click the Mark All Upgrades icon below the menu bar to select all available packages for upgrade, or click on the Installed (upgradable) link in the left panel to review the packages or to select upgrades individually.

    - Click Apply to begin the upgrade, ignoring the warning message. As the installation process begins, you have the option of watching the details in a terminal within Synaptic.

- With some package upgrades, you may be asked to confirm a dialog, enter configuration information, or decide whether or not to overwrite a configuration file you have altered. Pay attention here, and follow the prompts until the upgrade completes.

Downgrading

Sometimes you may want to downgrade an application to an older version, for instance because of problems that arose with the new one. This is easy to do in Synaptic:

1. Open Synaptic, supply the root password, and click Reload.
2. Click on Installed in the panel on the left, then find and highlight the package you want to downgrade in the panel on the right
3. On the menu bar, click Package > Force version...
4. Select from the available versions on the pull-down list
5. Click Force Version, then install in the usual manner.

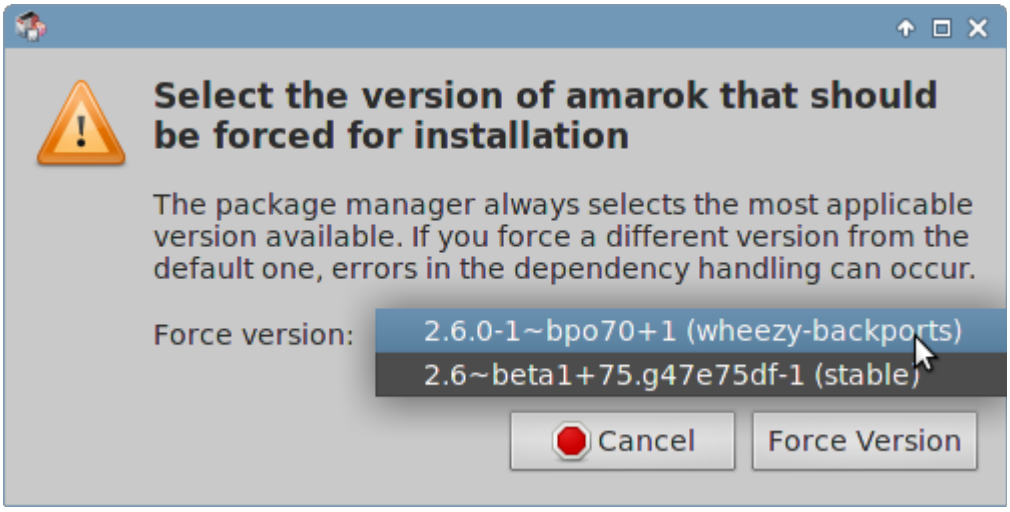

Figure 5-4: Using Force version to downgrade a package

Pinning

Sometimes you may want to pin an application to a specific version to keep it from being upgraded in order to avoid problems with more recent ones. This is easy to do:

1. Open Synaptic, supply the root password, and click Reload.

2. Click on Installed in the panel on the left, then find and highlight the package you want to pin in the panel on the right.

3. On the menu bar, click Package > Lock version...

4. Synaptic will highlight the package in red and add a lock icon to the first column.

5. To unlock, highlight the package again and click Package > Lock version (which will have a check mark).

5.4 Troubleshooting

Synaptic is very reliable, but sometimes you may get an error message. A full discussion of such messages may be found in the MX/antiX Wiki, so here we will only mention a couple of the most common.

- You get a message that some repos failed to download repository information: this is usually a transient event, and you simply need to wait and reload.

- If the installation of a package shows that software you have already installed and still want will be removed, click Cancel to back out of the operation.

- It may happen with a new repository that you see an error message after reloading that says something like: W: GPG error: [some repository URL] Release: The following signatures couldn't be verified . This message appears because apt includes package authentication in order to improve security, and the key is not present. To fix this, click **Start menu > System > MX Check Apt GPG** and follow the prompts.

- Occasionally, packages will fail to install because their install scripts fail one or more safety checks; for instance, a package might try to overwrite a file that is part of another package, or require downgrading another package due to dependencies. If you have an install or upgrade that is stuck on one of these errors, it is called a broken package. To fix this, click on the Broken packages entry in the left panel. Highlight the package and try first to fix the problem by clicking Edit>Fix Broken Packages. If that is not successful, then right-click the package to uninstall it.

- Should you uninstall? Occasionally, conflicts in package dependencies can cause the APT system to require the uninstallation of a large number of important packages in order to install some other package. This is rare with the default configuration, but becomes increasingly likely as you add unsupported repositories. **BE VERY ATTENTIVE** whenever installing a package would require that others be removed! If a large number of packages are

going to be removed, you may want to investigate another method of installing this application.

- Should you keep? When upgrading, you may sometimes be informed that a new configuration file is available for a certain package, and be asked whether you want to install the new version or keep your current version.

    - If the package in question is from an MX repository, it is recommended that you "install the maintainer's version"

    - Otherwise, answer "keep the current version" (**N**), which is also the default choice.

5.5 Other methods

5.5.1 Aptitude

Aptitude is a package manager that can be used instead of apt or Synaptic. It is available from the repos, and is particularly helpful when dependency problems arise. Can be run as a straight CLI or as a primitive GUI.

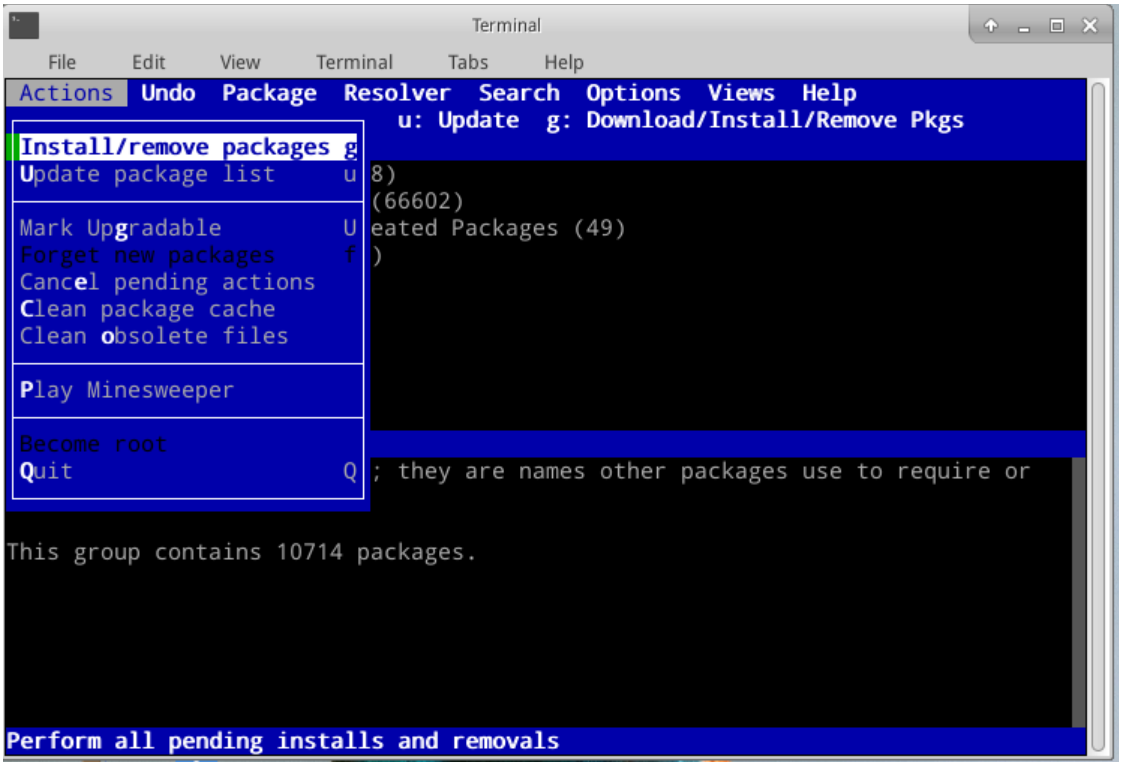

Figure 5-5: Aptitude's home screen (GUI), showing dependency resolver.

For details on this option, see the MX/antiX Wiki.

5.5.2 Deb packages

The software packages installed through Synaptic (and APT behind it) are in a format called deb (short for Debian, the Linux distribution that devised APT). You can manually install 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 dependencies can not be satisfied, you will receive a notice and the program will stop.

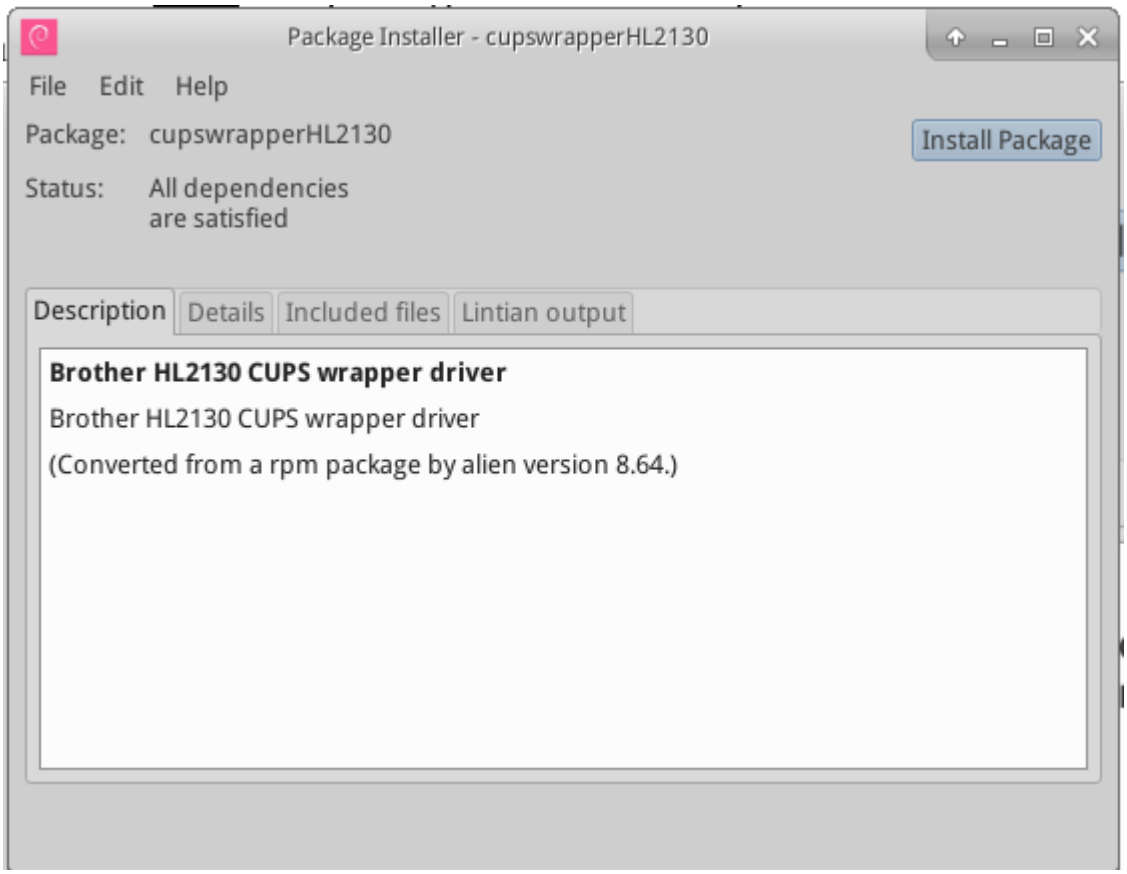

Figure 5-6: Gdebi ready to install.

*Installing *.deb files with Gdebi*

1. Navigate to the deb package you want to install and click on it. Gdebi will open the install dialog.

2. Click Install.

3. Enter your root password when prompted.

4. Gdebi will attempt to install the package, and report the results.

*Installing *.deb files with dpkg*

1. Navigate to the folder containing the deb package you want to install.

2. Right-click an empty space to open a terminal and become root

3. Install the package with the command (substituting the real package name, of course):

   ```
   dpkg -i packagename.deb
   ```

4. If you are installing multiple packages in the same directory at the same time, you can do it all at once using:

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

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

5. If required dependencies are not installed on your system already, you will get unmet dependencies errors as dpkg does not automatically take care of them. To correct these errors and finish the installation, run this code:

   ```
   apt -f install
   ```

6. apt will attempt to rectify the situation by either installing the needed dependencies (if they are available from the repositories), or removing your .deb files (if the dependencies can not be installed).

   NOTE: the command used in Step 5 above reflects the change from the legacy **apt-get**.

5.5.3 Appimages and flatpacks

 Launchers and appimages

Appimages and flatpacks are self-contained packages that do not need to be installed, simply downloaded and made executable (right-click > Permissions). A number of them exist already and it is expected that more will be made available in the future for software distribution.

5.5.4 CLI methods

It is equally possible to use the command line to install, remove, update, switch repositories and generally to manage packages. Instead of launching Synaptic to carry out common tasks, for instance, many users will just open a terminal, become root and use one of these commands (root privileges required).

Table 5: Common commands to manage packages

Command	*Action*
apt install packagename	Install a certain package
apt remove packagename	Remove a certain package
apt purge packagename	Completely remove a certain package (but not configuration/data in /home)
apt autoremove	Clear out leftover packages after a removal
apt update	Refresh the package list from the repos
apt upgrade	Install all available upgrades
apt dist-upgrade	Intelligently handles changing dependencies with new versions of packages

5.5.5 More install methods

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

- RPM packages: Some distributions of Linux use the RPM packaging system. RPM packages are similar to deb packages in many ways, and there is a command-line program available from MX Linux to convert RPM packages to debs called **alien**. It does not come installed with MX Linux, but is available from the default repositories. After you have installed it on your system, you can use it to install an rpm package with this command (as root): **alien -i packagename.rpm**. That will place a deb file with the same name in the location of the rpm file that you can then install as described above. For more detailed information on alien, see the internet version of its man page in the Links section at the bottom of this page.

- Source code: Any open-source program can be compiled from the programmer's original source code if there is no other option. In ideal circumstances, this is actually a pretty simple operation, but sometimes you can run into errors that require more skill to sort through. Source is usually distributed as a tarball (tar.gz or tar.bz2 file). See the Links for a tutorial on compiling programs.

- Miscellaneous: Many software developers package software in their own custom ways, usually distributed as tarballs or zip files. They may contain setup scripts, ready-to-run binaries, or binary installer programs similar to Windows setup.exe programs. In Linux, such

programs often end in .bin. Google Earth, for example, is often distributed this way. When in doubt, consult the installation instructions provided with the software.

5.5.6 Links

- MX/antiX Wiki: Synaptic errors
- MX/antiX Wiki: Installing Software
- MX/antiX Wiki: Compiling
- Gdebi
- The Debian package management tools
- Debian package management tools
- Debian APT Guide
- Debian APT Guide
- Wikipedia: Alien