"Mastering Software Management" Cheat Sheet

1. Searching the Cache

Ubuntu's package manager (apt) uses the /var/lib/apt/lists as a cache. A cache is basically a place to store a local copy of some files in order to improve computational performance.

This cache (known as the apt-cache) contains lists of packages available in the Ubuntu repositories.

The Ubuntu Repositories are:

Main	Free and Open Source Software maintained by Canonical
Universe	Free and Open Source Software maintained by the Ubuntu Community.
Restricted	Proprietary software and drivers for company-specific devices such as wireless cards etc.
Multiverse	Software that is restricted either by copyright or legal issues.

To search the apt-cache for packages match certain search term you can use:

```
apt-cache search <search term>
```

This will list out all the packages that match a certain search term, and also give a snippet of information about each result.

To find more information about a specific package you can do:

```
apt-cache show <package name>
```

2. Updating the Cache

There is a configuration file called sources.list located in the /etc/apt directory that tells the package manager which package lists to download.

These package lists are stored on archive.ubuntu.com.

An important factor for maintaining any cache is to ensure that it is up to date with the original source, and updating your systems cache is fortunately very easy to do.

Just do:

```
sudo apt-get update
```

This will cause the apt package manager to compare your package lists in your cache to those available on archive.ubuntu.com and ensure you have the most up to date versions.

3. Upgrading Software

Overtime, software becomes outdated as newer versions are released and new functionality is produced.

The apt package manager has a very elegant way of using the package lists in your apt-cache to ensure that all the software on your system is fully up to date.

Doing so not only makes your software more functional and modern, but it also enhances system security.

To update all the software on your system to the most up-to-date versions mentioned in the package lists in your apt-cache use:

```
sudo apt-get upgrade
```

Because this process uses the package lists in your apt-cache, in order for this to be effective, your package lists should be completely up-to-date.

Therefore, *you must ensure that your package lists are up to date before upgrading software* by using:

```
sudo apt-get update
```

Note: This will only update software installed via the package manager. It won't work for software compiled and installed manually. Therefore, *always aim to install software from the repositories wherever possible.*

4. Installing Software

Installing software is a very easy task using the apt package manager.

When you have found a package that you would like to install simply do:

```
sudo apt-get install <package name>
```

This will download and install the package, as well as any dependencies that are required in order to make the package function.

Again, in order to install the most up to date version of the package, *ensure that you have updated your cache prior to installing* by executing sudo apt-get update.

5. Installing Source code

The sources.list file in the /etc/apt folder contains lines telling the package manager what package lists it should download.

By default the lines that allow the apt package manager to download source code packages are "commented out" and are preceded by a hash symbol (#).

The lines for source code packages start with deb-src for example:

```
#deb-src gb.archive.ubuntu.com/ubuntu artful main restricted
```

To enable source code lists to be accessed by your package manager, open the configuration file using sudo nano /etc/apt/sources.list and remove the hash symbol from the front of each line so that for example:

```
#deb-src gb.archive.ubuntu.com/ubuntu artful main restricted
```

Would become

```
deb-src gb.archive.ubuntu.com/ubuntu artful main restricted
```

Once done for all lines starting with deb-src, you should save the file and update your cache so that the new source package lists can be downloaded.

Update your cache using:

```
sudo apt-get update
```

Once these new lists are downloaded in your cache, there is one more step to allow you to download source code.

You need to install the dpkg-dev package using:

```
sudo apt-get install dpkg-dev
```

Once that is installed, you are now ready to download source code from the Ubuntu Repositories.

To download source code, simply do:

```
sudo apt-get source <package name>
```

6. Compiling and Installing from Source Code

With the source code downloaded, you are free to make your own edits to the code, recompile it and install it on your machine.

To compile and install code written in the C programming language, you will need the Gnu C Compiler (gcc), which you can install by running:

```
sudo apt-get install gcc
```

With the Gnu C Compiler installed, the compiler must be configured correctly prior to compiling a package.

Inside the package directory, there will be a configure file. From within the same directory as the configure file, simply run *configure*, and the Gnu C Compiler will be configured.

This will generate a new file in the directory called the MakeFile.

In order to operate on the make file, you will need to install one more package using:

```
sudo apt-get install make
```

When that has installed, then from within the same directory as the MakeFile, you can execute the *make* command and all the software in the package will be ran through the Gnu C Compiler and compiled into binary code ready for installation.

In order to install the compiled binary files, simply run:

```
sudo make install
```

7. Uninstalling Software

When software packages are no longer required it is possible to remove them from your system with a few simple commands.

To remove a package completely (including any configuration files) simply run:

```
sudo apt-get purge <package name>
```

Oftentimes when a package is installed, many other packages required for the successfully operation of that package will be installed alongside it. These packages are known as dependencies.

In order to remove any dependencies that are no longer required by the system you can do:

```
sudo apt-get autoremove
```

Whenever a package is installed, the archive it came from is also downloaded on the system, and stored in the /var/cache/apt/archives directory.

The amount of space consumed by the archives in that directory can become quite large, so to clean it out, you can run:

```
sudo apt-get clean
```

Sometimes even Gigabytes of space can be saved by this process, so it is worth doing often.

Perhaps, however, that you don't want to clean out *all* the packages from that directory, but instead only want to remove the ones that are so old that they can no longer be found in the up-to-date repositories.

To delete only those packages, you can run:

```
sudo apt-get autoclean
```

"Well done! You are now a master of software management in the Gnu/Linux Operating System! I hope that this cheat sheet is useful to you if you every need a refresher!

Best wishes, Ziyad.