Short overview of the multiple usages of SVN, aka Subversion. The goal is to have a depository on the web, which maintains the scripts that are part of ComboCode (excluding any new versions of *MCMax* or *GASTRoNOoM*).

1 The ComboCode home folder

The home folder for ComboCode is ∼/ComboCode/. For now this cannot be changed, and you will have to make the folder now yourself. Make the folder, and then run the first check out of SVN (next section) in it. When you do this, a new folder called `cc` is made automatically in that home folder.

The inputfiles for ComboCode can be anywhere you want them to be. However, if you store them in ∼/ComboCode/, you don't have to specify the path. If you store them elsewhere (fi ∼/ComboCode/input/), always specify the path as well when calling ComboCode.

2 First check-out

In the example here, I'm using the home folder `/home/robinl/ComboCode/` in which I run the first check out command. When downloading the scripts for the first time, the command you should use is:

```
svn co http://subversion.ster.kuleuven.be/repos/astrophysicalcodes/cc/
```

This will create a new folder called `cc` in the folder `/home/robinl/ComboCode/`. From then on, that folder is recognized as a sort of working folder for SVN, from which you can add, update, check the status and so on for every file in the folder, i.e. including all the ComboCode scripts that you need.

If you want to run these scripts in python or ipython, whichever python shell you prefer, python needs to know where to find the scripts. Include `/home/robinl/ComboCode/` in the PYTHONPATH in your .bash_profile. The command in .bash_profile should be something like this, replacing my name with yours (or whatever your home folder is):

```
PYTHONPATH=/home/robinl/ComboCode/:$PYTHONPATH
export PYTHONPATH
```

Make sure you remember to source the new .bash_profile before using any of the scripts or importing them into a (i)python shell. Furthermore, you will need an __init__.py file in ∼`/ComboCode/` for the scripts to be importable in python. This is not provided by SVN, simply because you would otherwise have to download every other astrophysical code on SVN (for the IvS) as well, which is not very practical.

3 Updating your scripts

If at some time you want to update the scripts in the folder to the most recent version (it's probably safest to update the scripts regularly, even if nothing changed, then you're sure you're always using the version you should be using), do

```
cd ∼/ComboCode/cc/
```

and then you can use

```
svn up
```

There is no need to use the `co` command again. Only use that command when you make an entirely new cc folder that isn't known yet by SVN. If you want information on what's been updated as compared to a last version:

```
svn log
```

or

```
svn log script.py
```

for a log on a specific file. I will try to keep the log as updated as possible, so you can always check what's been updated or what hasn't, if at any time that would be important. You can ask for the current status of the files in the `cc` folder as well:

```
svn status
```

inside the `/home/robinl/ComboCode/cc/` folder. Every file will then be listed. If a file is not known by the SVN server, it will be listed as '?'. If it's just been updated from an older version to the most recent one, it will be listed as 'U'. If you added a file to the SVN server or if you made a change to a file before doing a commit (only for those who have writing access), it will be listed as 'A' and 'M' respectively. If the file has not been updated and is the same as the version on the SVN server, it is not listed at all.

4 Managing files in the repository

All of you currently have read-only access to the scripts. If that changes at any point, or if you want to set up your own SVN account, you may want to use following commands as well (to be honest, though, I'm including those here so I can remember them myself... :-)):

```
svn add script.py
```

to add a script for the first time to the SVN server.

```
svn commit
```

to update all scripts known by the SVN server with the versions currently in your SVN folder. An Emacs window will pop up, allowing you to give a description of the changes you've made in the updated scripts, since the last version. Save, and you're done!

5 Some final remarks concerning ComboCode

Whenever you do any of the above, you may be prompted for your login and your password. Your login is always your full capitalized name without spaces: RobinLombaert, for instance. I will e-mail you your password in a separate mail. I think it's possible to have it changed, but then you will have to talk to Wim De Meester, of the system group at the IvS.

Please, remember to update your CC scripts regularly! If major changes have been made, I will let you know. Other files on SVN:

- I'll make sure I keep an updated input file for ComboCode (named inputComboCode.dat in cc/) with all possible keywords in the SVN folder. Copy that file to /home/robinl/ComboCode/ (or wherever you store your inputfiles) and name it whatever you want, but don't use this file itself. The inputComboCode.dat file is updated by SVN and will not remember any changes you make. Furthermore, it's always a good idea to keep inputfiles separate from code.
- The CC_Structure file is also in here. Check that file to see what folders you have to copy over from my home at /home/robinl/.

- The inputfiles that manage the keywords vs databases (Input_Keywords_*.dat), and the interaction between GASTRoNOoM and MCMax (Mutable_Parameters_*.dat) are also here, so they'll remain up-to-date. Don't touch them!
- The other inputfiles that you need for ComboCode (Dust.dat, Molecule.dat, Star.dat and Pacs_Resolution.dat in ∼/ComboCode/Data) are not on SVN: They should remain the same as before, and you can make your own changes to some of them if you want. If anything changes in them as far as syntax is concerned, I will let you know and send you a new example file manually.
- Any special inputfiles for *MCMax* and *GASTRoNOoM* (usually templates for running the codes) are not on SVN either. They should not be changed ever, so it shouldn't be a problem anyhow. These are:
  - inputMCMax.dat, Spec.out and visibilities_baseline.out in ∼/MCMax/src/;
  - vic_job_example.sh, vic_run_jobs_example.sh, inputGASTRoNOoM.dat and execGASTRoNOoM_example in ∼/GASTRoNOoM/;
  - any other code-specific stuff you need regardless of using ComboCode (executables, collision rates, opacity files, ... you should get them when you copy the MCMax and GASTRoNOoM folders over from me anyway).

Happy modelling!