**Official ASCOT software utilities**

There are a number of software utilities written and maintained by the ASCOT team. I have used them sparingly, having chosen instead to postprocess the ASCOT output with my home-grown Python scripts, over which I have more control. In retrospect, perhaps I should have spent more time learning how to better use the capabilities of these utilities.

Currently the utilities that I use live in directory /global/homes/s/sscott/.local/cori/3.6-anaconda-4.4/bin/

Confusion: there are also copies of these utilities in directory: /project/projectdirs/m3195/ascot/python/a5py/bin/

Presumably, we should be using these utilities rather than the ones in my personal directory.

I’m a little confused about how they are ‘defined’, in the sense that all I need to do is type e.g. a5gui ascot\_CP.h5 (where ascot\_CP.h5 is the name of an ASCOT output file) from any directory, and the utility is magically invoked. I can’t figure out how the system ‘knows’ that the character string a5gui gets mapped to the directory /global/homes/s/ …

If I type ‘which a5gui’ at the command prompt level, the system responds with /global/homes/s/sscott/.local/cori/3.6-anaconda-4.4/bin/a5gui

My .bashrc.ext file makes an alias for a5gui 🡪 /project/projectdirs/m3195/ascot/python/a5py/bin/a5gui

They seem to be simple Python shells that call other Python scripts. Generally, I think that to invoke these utililties, you typically type <utility name> <h5-fielname>

* **a5gui**: This is an all-purpose utility that allows you to view the input options settings, Bfield, Efield, results (inistate, endstate, distributions), wall, plasma data.
* **a5editoptions**: this allows you to view and edit the options settings. I think by default it uses the VIM editor, but I think that by putting an appropriate line in your .bashrc.ext file, you can force it to use other editors including emacs.
* **a5combine**: this allows you to combine multiple ASCOT hdf5 files into a single file.

IMHO, there is IMHO a small limitation to this utility. As described below, the hdf5 file includes sections for ‘marker’, and ‘results’, along with some other sections of data. It seems that the a5combine utility does combine the ‘results’ from multiple simulations, but it does not combine the ‘marker’ information. This does not seem to be a problem for the various ASCOT software utilities, but my home-grown utility process\_ascot.py has some plots that include both marker and results data (for example, it plots the major radius of the markers versus major radius of the inistate).

So the a5combine utility can be used to combine multiple ASCOT5 hdf5 files if the resulting combined file is to be processed by the ASCOT-supplied software utilities, but it cannot be used to create a combined hdf5 file that can be used by my home-grown Python scripts such as process\_ascot.py.

But note that process\_ascot.py has been upgraded to allow it to read multiple hdf5 files directly, and in doing so it properly copies all of the data from multiple runs, including the marker data.

I opened up issue #245 in the ASCOT web portal about this issue. Maybe it will get fixed in the future.

example:

5combine add ascot\_DV.h5 ascot\_39176023.h5 ascot\_39176024.h5 ascot\_39176025.h5

… then type ‘c’ when prompted

* **a5doxygen, a5vol, a5ls, a5setactive, a5continuerun**: … there are a number of additional utilities that I have not used.

Problem: in the past, I have had problems invoking these utilities because the system reports that it can’t find the module ‘unyt’ (which supports processing of units, e.g. eV, in the utilities). I forget the details, but when I try to install the module (need to remember the syntax for doing that), the system reports no errors during the install, but then I get the same error message when I try to invoke the script. The problem is that when you install a module, you must install it into the version of Python that the script actually uses.

pip install –user unyt 🡨 this worked for Alex (2/17/2021)