

TM5-MP - Support #20341

Python 2 becoming obsolete

07/26/2019 10:42 AM - Tommi Bergman

Status:	Closed	Start date:	07/26/2019
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
Just realised that the new supercomputer in Finland will have python 3 as default. Officially Python 2 will be retired at the end of this year. Probably after this Computing centres start to force updating to python 3. Therefore, we should think about updating the pycasso processing to python 3. It's not urgent but probably needs to be done by some not so distant future date.			

History

#1 - 07/26/2019 01:52 PM - Philippe Le Sager

- Status changed from New to In Progress

- % Done changed from 0 to 10

May be more urgent than we think. At ECMWF, python 2 is already not supported (although available), but will not be available after the migration to Bologna. Of course we could always install it ourselves (pretty straightforward with conda), but I think it is better to switch to python3.

Should we have two sets of py scripts, one for python 2.7.x and the other for 3.6+? I don't think so, but maybe there are arguments for it. I'm really not in favor of having one set that handle both major versions.

I gave a quick try at porting to python3. I run 2to3 on all python files in the bin dir and did not get any error. But trying to run TM5 did not work right away. I made a couple of small obvious fixes (eg, remove import exceptions), but then an issue popped up with rc.py when trying to parse the included rc files. To be continued... Feel free to give a go, I won't be able to look closer into it for a long time.

#2 - 07/26/2019 01:55 PM - Tommi Bergman

What do you mean by error. Since it sounds a bit funny, since my 2to3 gives the following(only the last bit not the diff):

```
RefactoringTool: Files that need to be modified:
RefactoringTool: bin/go_logging.py
RefactoringTool: bin/go_subprocess.py
RefactoringTool: bin/pycasso.py
RefactoringTool: bin/pycasso_user_scripts_template.py
RefactoringTool: bin/pycasso_user_scripts_tm5.py
RefactoringTool: bin/rc.py
RefactoringTool: bin/submit_tm5_setup_rcfile.py
RefactoringTool: bin/submit_tm5_tools.py
```

#3 - 07/26/2019 02:18 PM - Philippe Le Sager

By "no error", I mean the writing of the modified files went ok. But yeah, it that doesn't mean the files are ok ;)

I had to pass a list of files to 2to3 to be sure that all python scripts are parsed, i.e. to include the ones not ending with ".py" like some of the "submit_tm5".

The problem with the include files I mentioned above is gone; it was an issue with my environment. After replacing a couple of:

```
if key in rcf:
```

with

```
if rcf.has_key(key):
```

I was able to get a bit further. Now I have a Unicode issue:

```
[INFO ] copy files from source directories...
[INFO ] remove "__<name>" parts from sources files ...
[INFO ] scanning base ...
[INFO ] scanning levels/ml60 ...
[INFO ] scanning levels/ml60/tropo34 ...
[INFO ] scanning proj/output ...
[INFO ] scanning proj/budget10 ...
[INFO ] scanning proj/cb05 ...
[ERROR ] Traceback (most recent call last):
[ERROR ] File "setup_tm5", line 63, in <module>
[ERROR ] pycasso.Main( args, pycasso_user_scripts )
[ERROR ] File "/home/ms/nl/nm6/TM5MP/bin3/pycasso.py", line 253, in Main
[ERROR ] Build_Copy( rcf, pycasso_user_scripts )
[ERROR ] File "/home/ms/nl/nm6/TM5MP/bin3/pycasso.py", line 786, in Build_Copy
[ERROR ] if pycasso_tools.diff_text_files(sourcefile,targetfile) :
[ERROR ] File "/home/ms/nl/nm6/TM5MP/bin3/pycasso_tools.py", line 55, in diff_text_files
[ERROR ] lines1 = f.readlines()
[ERROR ] File "/usr/local/apps/python3/3.6.8-01/lib/python3.6/codecs.py", line 321, in decode
[ERROR ] (result, consumed) = self._buffer_decode(data, self.errors, final)
[ERROR ] UnicodeDecodeError: 'utf-8' codec can't decode byte 0xb5 in position 1046: invalid start byte
[ERROR ]
[ERROR ] exception from pycasso.Main
```

#4 - 07/26/2019 02:25 PM - Arjo Segers

Alternative is to switch to the 'utopya' scripting, now already used in the 'adjoint' branch.

Works fine under py2 and py3.

Advantage is that it creates sequence of jobs that can be configured individually, for example "install", "build", "run", and "post" each with different queue options (cores, memory, etc).

#5 - 08/30/2019 09:31 AM - Philippe Le Sager

Arjo Segers wrote:

Alternative is to switch to the 'utopya' scripting, now already used in the 'adjoint' branch.

Works fine under py2 and py3.

Advantage is that it creates sequence of jobs that can be configured individually, for example "install", "build", "run", and "post" each with different queue options (cores, memory, etc).

Sounds very much like the current pycasso, but with a better handling of the different steps. This may be an option for the long term, but for the model being coupled to EC-Earth we need to stick to the current set of scripts, since I had to modify it significantly to be able to pass many variables/parameters through the command line and I suspect I will have to do it again with utopya.

#6 - 01/27/2022 11:53 AM - Philippe Le Sager

- *Status changed from In Progress to Closed*

- *% Done changed from 10 to 100*

Fixed with the merging of adjoint code.