marquer ce message pour suppression

- **From**: EL KHATIB Ryad <ryad.elkhatib@meteo.fr>
- **To**: alabobo2 <alabobo2@meteo.fr>
- **Subject**: [alabobo2] Dependencies analysis in gmkpack
- **Date**: Wed, 3 Oct 2018 16:06:48 +0200 (CEST)

Dear all,

Some of you must have noticed that sometimes the recompilation of fortran modules goes wrong with gmkpack (especially for deeply nested modules), and that a workaround is to clean the pack. Sometimes cleanpack is even not enough and a recompilation from scratch as a main pack can be necessary.
Recently I had to fix a severe issue which was concerning the dependency analysis of the C++ code for OOPS in gmkpack, and this helped me understand the problem with the dependencies of fortran code : the trouble seems to be that the recursive analysis of modules dependencies is not performed at the time the source code is analysed : it is done later, to find out in which order files should be compiled and which files are outdated. This makes the algorithm very complex. The proper way to do may be to perform the recursive dependencies analysis while the source code is analysed ; so that we would have an independent and "flat" analysis of each file, and it would be then easy to find out what is outdated. I had to do so for the C++ code, fortunately with the help of cpp ; otherwise the compilation was deadlocked. But cpp can't be used to analyse the recursivity of modules. You can also consider very strange that include files are treated like modules and "compilable" files in gmkpack. With that "flat" analysis of the source code they would be just passive files.
I intend to fix this issue one of these days, but you know, it takes time to develop, and even more time to validate, ...

Best regards,
Ryad

----- Météo-France -----
EL KHATIB RYAD
CNRM/GMAP/ALGO
ryad.elkhatib@meteo.fr
Fixe : +33 561078466

- **[alabobo2] Dependencies analysis in gmkpack**, *EL KHATIB Ryad, 03/10/2018*