

Operator EXEC_LOGICIEL

1 Goal

The goal of this operator is to carry out an external software since Code_Aster on the local machine. That can be any program or achievable script.

A first alternative is to carry out a maillor instead of an unspecified program in order to produce a file of grid. In this case, one starts SALOMÉ, GMSH or GIBI while passing to him a data file in argument and one recovers then a file of grid.

The second alternative consists in calling on a session SALOMÉ already open on the local machine or a distant machine and carrying out there a script (visualization for example).

2 Syntax

```
EXEC_LOGICIEL (
# or if GRID is present
e-mail = EXEC_LOGICIEL ( [grid]

    ◇ | SOFTWARE = nom_exe, [TX]
      ◇ ARGUMENT = (arg1, arg2,... ) [l_TX]
      ◇ SHELL = / 'NOT', [DEFECT]
              / 'YES',

    | / GRID = _F (
      ◆ FORMAT = / 'GMSH', [TX]
              / 'GIBI',
              / 'SALOMÉ'
      ◇ UNITE_GEOM = / igeom, [I]
                  = / 16 [DEFECT]
    ),

    | / SALOMÉ = _F (
      ◆ CHEMIN_SCRIPT = script, [TX]

      ◇ MACHINE = machine , [TX]
        # if MACHINE is indicated
        ◆ USER = user, [TX]
      ◇ PORT = / port, [I]
          / 2810, [DEFECT]
      ◇ FICHIERS_ENTREE = L_arg , [l_TX]
      ◇ FICHIERS_SORTIE = L_arg , [l_TX]
      ◇ ◆ NOM_PARA = L_arg , [l_TX]
        ◆ VALE = L_arg , [l_TX]
    ),

    ◇ CODE_RETOUR_MAXI = / icode, [I]
                        / 0, [DEFECT]
    ◇ INFORMATION = / 1 , [I]
                  / 2 , [DEFECT]
)
)
```

3 Operands

3.1 Operand SOFTWARE

◇ SOFTWARE = nom_exe

Name of the order or achievable to call. It is a character string (between '), it is necessary to specify way complete to reach the achievable one.

This operand can also to be used to overload the order by default during the creation of a grid or of the call of SALOMÉ.

3.2 Keyword ARGUMENT

◇ ARGUMENT = l_arg

Allows to define the list of the arguments passed in the achievable one. To each element of the list an argument provided to the achievable one corresponds. If an argument contains a space, the achievable one will recover a character string containing space (see keyword SHELL).

This keyword is obligatory if one creates a grid with the format SALOMÉ. In this case, it is necessary to provide a single argument which is the name of the file which is produced by the script of grid.

3.3 Keyword SHELL

◇ SHELL = 'YES' / 'NOT'

The good practice consists in carrying out a program (keyword SOFTWARE) with a list of arguments (keyword ARGUMENT). One leaves then SHELL with its value by default which is worth 'NOT'.

Sometimes, one wishes to not carry out a program to which one passes from the arguments but a command line supplements (for example starting with a test yew). In this case, it is necessary to use SHELL='NON'. The keyword SOFTWARE and the possible ones ARGUMENT[S] additional are joined with a space to compose the complete command line.

See the documentation of the module subprocess of Python for shelves of SHELL='NON' (False).

3.4 Keyword GRID

◇ GRID = _F (

Allows to produce a grid by calling directly since the command file one of the following tools: GMHS, GIBI or SALOMÉ. Grid e-mail is turned over by the operator.

By default, the order to carry out the maillor is foundE in the repertoire of installation of Code_Aster + /outils (for example /opt/aster/outils).

For the formats quoted previously, one will call respectively: ... /outils/gmsh, ... /outils/gibi, ... /outils/salome.

One can use SOFTWARE (and ARGUMENT/SHELL) to call another achievable.

3.4.1 Operand FORMAT

/ FORMAT = 'GMSH'

Creation of a grid to format GMSH.

/ FORMAT = 'GIBI'

Creation of a grid to format GIBI.

/ FORMAT = 'SALOMÉ'

Creation of a grid to the format SALOMÉ, or more generally a file with format MED.

In this case the keyword `ARGUMENT` is obligatory and contains the name of file `MED` produced by script.

3.4.2 Operand `UNITE_GEOM`

```
UNITE_GEOM = igeom
```

Logical number of unit associated with the data file used to create the grid.

Attention the data file `GIBI` must end in the order:

```
OPTI SAUV FORMS `fort.8`
```

3.5 Keyword `SALOMÉ`

```
SALOMÉ = _F (
```

Allows to carry out a script in an authority of `SALOMÉ` who must already to be impetus (`Code_Aster` does not launch `SALOMÉ` in this case and does not finish it either), on the same machine as the object computer `DE Code_Aster` or on a distant machine. It is the launcher `Salome` who manages the contact of a distant authority.

Script `SALOMÉ` must be with syntax Python and being `exécutwhitebait` since `SALOMÉ` via one `FileLoad Script`. It must follow a certain number of conventions of writing, in particular on the variables used for the input files (i.e. used by script) and the output files (i.e. generated by script), bus of the adjustments/replacements are operated before the execution in `SALOMÉ`.

By default, one uses script ... `/outis/salome` but one can overload this value by using the keyword `SOFTWARE`.

In the event of connection to a distant session, it is necessary to indicate, with the keyword `SOFTWARE`, the site of script on the distant machine. Failing this, it will be supposed that it is to install at the same place as on the local machine (and thus that the repertoire of installation of `Code_Aster` is the same one).

3.5.1 Operand `CHEMIN_SCRIPT`

```
CHEMIN_SCRIPT = './script.py'
```

This keyword makes it possible to specify the way of script `SALOMÉ`. One can use an absolute way (`/home/user/mon-script.py`) or relative (`./fort.99` will open the file `fort.99` contents in the temporary repertoire of execution `DE Code_Aster`).

If script needs files as starter and/or creates output files, the keywords are used `FICHIERS_ENTREE` and `FICHIERS_SORTIE`. See the two paragraphs hereafter.

3.5.2 Operand `MACHINE`

```
MACHINE = machine
```

This keyword makes it possible to specify the name of the machine or sound address IP on which is openE the session of `SALOMÉ` to be contacted. Louse to use the local machine, this keyword should not be used.

3.5.3 Operand `USER`

```
USER = user
```

This keyword is obligatory if `MACHINE` is well informed. It is the name of the user who started the authority `SALOMÉ` on the distant machine.

3.5.4 Operand `PORT`

```
PORT = port
```

This keyword makes it possible to specify the port of the authority SALOMÉ which one seeks to be attached. This port is displayed during the launching of SALOMÉ starting from a terminal. By default, it is worth 2810 but it should be checked!

3.5.5 Operand FICHIERS_ENTREE

◇ FICHIERS_ENTREE = ['/. /fichier_in1', '/. /fichier_in2', ...]

This keyword makes it possible to specify the list of the data files who could be used by script SALOMÉ (for example a file MED if script corresponds to a postprocessing).

For that, script will be modified by replacing the character strings INPUTFILE1 with INPUTFILEn by N files of FICHIERS_ENTREE. If necessary, the launcher Salome gives the responsibility itself to copy these files from the distant machine, then of (Re) modifying script to write the name of the files there after recopy.

3.5.6 Operand FICHIERS_SORTIE

◇ FICHIERS_SORTIE = ['/. /fichier_out1', '/. /fichier_out2', ...]

In a symmetrical way with FICHIERS_ENTREE, one can recover the files generated by script SALOMÉ. Script produces files under the names OUTPUTFILE1 with OUTPUTFILEp. These character strings will be replaced by the files of FICHIERS_SORTIE.

In the event of distant launching, LE launcher Salome takes care of (Re) modifying script to insert there the names used on the distant machine, then to repatriate the produced files by using the names of FICHIERS_SORTIE.

3.5.7 Operands NOM_PARA and VALE

◇ NOM_PARA = ['para1', 'para2', ...]
◇ VALE = ['vale1', 'vale2', ...]

These two keywords allow paramétriser script SALOMÉ by replacing each occurrence of para1 by vale1, para2 by vale2, etc.

3.6 Operand INFORMATION

◇ INFORMATION = information

If INFO=2, the messages coming from the order carried out are printed in the file MESSAGE. It is the value by default. That makes it possible to preserve the trace of the execution.

3.7 Operand CODE_RETOUR_MAXI

◇ CODE_RETOUR_MAXI = icode

Maximum value of the code return returned by the order or the software which is tolerated to consider that the execution proceeded well. By default this value is worth 0, if it is affected to - 1, the code return of the order is ignored.

4 Examples

One can find examples of use in the CAS-test *zzzz151a*.

The macro-orders are also of good examples of use of the various possibilities, in particular STANLEY for the execution of script in a session SALOMÉ.