
Operator EXEC_LOGICIEL

1 Drank

To call since *Code_Aster*, a software or a command system. External software and command system called can reside and will be carried out on the same machine as *Code_Aster*, or on a distant machine. It is more particularly intended for the development of macro-commands. This macro python leans on the modulus python `bone` and more particularly on the command `os.system`. This operator also allows to generate directly since the command file a mesh of the type GMSH, GIBI or SALOME. This possibility was introduced primarily to be used in the tests of NON-regression of the code, the checking of the mesh remaining essential to be able to launch a study. Lastly, this command makes it possible to carry out a script Python with syntax SALOME in instance of SALOME, on the current machine or a distant machine. Syntax

2 EXEC

```

    _LOGICIEL (
        |LOGICIEL=nom_exe
    ] |MAILLAGE

        =_F (
            FORMAT="/
                GMSH",
                GIBI"/"
                SALOME",
            UNITE_GEOM=igeom
            UNITE=unite
            MAILLAGE=ma
                [MACHINE

        _DISTANTE=_F (
            SSH_ADRESSE=
            SSH_LOGIN=
            SSH_PORT=
                [SALOME

        =_F (
            /CHEMIN_SCRIPT=
                /script.py',
        /UNITE_SCRIPT

            =unite
                [ I]
                SALOME

            _HOST= `adresse'
            _PORT=port,
            _RUNAPPLI= `/.
                /runSalomeScript',
            _ENTREE=l_arg
            _SORTIE=l_arg
            _PARA=l_arg
            =l_arg
                ARGUMENT

            =l_arg ,
                [I_TX]
                CODE

            _RETOUR_MAXI=/icode
        [DEFAULT
            ]
            INFO
                =/1,
                [I ]
                /2
                ,
                [DEFAULT
            ]
            Operands
    
```

3 Operand

3.1 LOGICIEL \diamond LOGICIEL

`=nom_exe` Name of the command

or of executable to call. It is a character string (between “), it is necessary to specify L” path (path) to together reach L`executable. This operand can be used to overload the command by default during the creation of a mesh. Note:

The officially

referred software must have an entrance point under a particular directory managed by the administrator (directory tools of the installation). The developer of macro-command must take the precaution to build the name by concaténant the name of the directory of installation which it will recover using a utility in order to ensure the portability. Key word

3.2 ARGUMENT \diamond ARGUMENT

`=l_arg` Makes it possible

to define the list of the arguments passed in the executable one. This key word is compulsory if one creates a mesh with format SALOME/MED. The values passed in the form of character strings, to load of executable to decode them, they are transmitted such as they are, the character “”(white) being used as separator. Key word

3.3 MAILLAGE \diamond MAILLAGE

`=_F` (Allows

D” to generate a mesh by calling directly since the command file L” one of the following tools: GMSH, GIBI or SALOME by providing the data file in a file. The various formats are related to the presence of the command associated in the directory of installation of Code_Aster . *Operand*

3.3.1 FORMAT /FORMAT

`= ' GMSH'` Creation

of a mesh to format GMSH. /FORMAT

`= ' GIBI'` Creation

of a mesh to format GIBI. /FORMAT

`= ' SALOME'` Creation

of a mesh to format SALOME, or more generally a file with med format. In this case key word ARGUMENT is compulsory and contains the name of the MED file produced by script. Operand

3.3.2 UNITE_GEOM \diamond UNITE

`_GEOM = igeom` logical

Number of unit associated with the data file used to create the mesh. Attention

the data file GIBI must finish by the command: OPTI

SAUV FORMS “fort.8”. Operand

3.3.3 UNITE \diamond UNITE

=unite logical

Number of unit associated with the results file produced by the tool for mesh. Operand

3.3.4 MAILLAGE \blacklozenge MAILLAGE

=ma Name of

the concept produced mesh. The name must be indicated in the syntactic form: CO (" MY") Key word

3.4 MACHINE_DISTANTE \diamond MACHINE

_DISTANTE=_F (Makes it possible

to carry out command defined by the key word LOGICIEL on a distant machine. The communication protocol SSH network will be used for connection with the distant machine, it is thus necessary that the accounts users are correctly configured for an access without password (use of SSH word). Operand

3.4.1 SSH_ADRESSE /SSH_ADRESSE

= ' adresse' It

acts of the address "network" of the machine: either the IP address (130.98.x.y example on the EDF internal network), or the complete name of the machine (clau5aaa.der.edf.fr example). The name runs (or hostname) can also be sufficient if the two machines are on the same network (clau5aaa example). Operand SSH_LOGIN

3.4.2 /SSH_LOGIN = ' login'

the login of the user

on the distant machine. If it is not specified, same the login as on the server of execution of Aster will be used. Operand SSH_PORT

3.4.3 /SSH_PORT = port This key word

makes it possible

to redefine the port of server HS. It is a advanced function making it possible to adapt to particular IT environments. In the majority of the cases, the users do not have to specify the port SSH, the port by default (22) being used. Key word SALOME \diamond

3.5 SALOME=_F (Makes it possible

to carry out

a script in instance of Salome, which must be launched in addition (Aster does not launch Salome), on the same machine as the object computer of Aster or on a distant machine. Script Salome must be with syntax Python of Salome, i.e. it is a script which can be carried out since Salome via Load 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 Salome. Operand CHEMIN_SCRIPT

3.5.1 \diamond CHEMIN_SCRIPT = ' '

`./script.py` This key word makes it possible to specify the path of script Salome. One can use an absolute path (`/home/user/mon-script.py`) or relative (`./fort.99` will open the file `fort.99` contained in the temporary directory of execution of Aster).
Operand UNITE_SCRIPT \diamond

3.5.2 UNITE_SCRIPT =unite This

key word makes it possible to specify the logical unit of script Salome. That makes it possible to integrate script Salome like input file into the profile of execution of ASTK or the file `.export` (to use the `libr` type associated with the logical unit defined by this key word).
Operand SALOME_HOST \diamond

3.5.3 SALOME_HOST = ' adresse'

This key word makes it possible to specify the IP address to specify or the name of machine (according to the same rules that key word `SSH_ADRESSE`) on which is open Salome. If the key word is not specified, the local machine will be used.
Operand SALOME_PORT \diamond

3.5.4 SALOME_PORT =port This key word

makes it possible instance to specify the port of the Salome to which one seeks to be attached. This port is given during the launching of Salome from a terminal: `assire@claut629: ~$ /local00/salome/SALOME-MECA-2011.1/runSalomeMeca`

```
Loading environment for
python 2.4 [OK]. /local00/salome/SALOME-MECA-2011.1/SALOME/SALOME5/V5_1_5/prerequis-V5_1_5.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME/SALOME5/V5_1_5/envSalome-V5_1_5.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME-MECA/V5_1_5/envSalomeMeca.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME-MECA/V5_1_5/prerequis-SalomeMeca.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME-MECA/V5_1_5/prerequis-Aster.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME-MECA/V5_1_5/prerequis-Eficas.sh
[OK]. /local00/salome/SALOME-MECA-2011.1/SALOME-MECA/V5_1_5/prerequis-OM.sh
Checking... LD_LIBRARY_PATH

Checking... PATH Checking

... PYTHONPATH CHECK

/home/assire Searching
for free port for
the SALOME Naming Service: 2810 - Ok runSalome running one claut
629 the value by default is
```

2810, but if several instances of Salome turn on the machine, the others will use successively the following ports: 2811,2812, etc. Operand SALOME_RUNAPPLI

3.5.5 \diamond SALOME_RUNAPPLI = './.'

`runSalomeScript` This key word makes it possible to specify the launcher of scripts in the environment Salome. This launcher is part of the distribution of Salome and is contained in indexes Salome. If it is not specified, one will use the `restrain` contained in the directory `tools` of Aster, which corresponds to the version by default of Salome.
Operand FICHIERS_ENTREE

3.5.6 \diamond FICHIERS_ENTREE = ["/

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

`. /fichier_in1 ", "/ . /fichier_in2 ",...]` This key word makes it possible to specify the list of the data files of script Salome (for example med file if script corresponds to a postprocessing). So that script can function, and in particular remotely, it is necessary to follow the convention of following writing: each input file must appear in script Salome in the form of the variable: INPUTFILE1= (for the first file) , INPUTFILE2= (for the second) , etc.: INPUTFILE1 = louse INPUTFILE2 = tutu the tutu and

character strings louse will then be replaced by the paths of the files defined by FICHIERS_ENTREE before the execution in Salome. Operand FICHIERS_SORTIE

3.5.7 \diamond FICHIERS_SORTIE = ["/

`. /fichier_out1 ", "/ . /fichier_out2",...]` This key word makes it possible to specify the list of the files which will be generated by script Salome (for example med file if script corresponds to an operation of mesh). Same convention as for the output and input files must be respected for script: each file must appear in script Salome in the form of the variable: OUTPUTFILE1= (for the first file) , OUTPUTFILE2= (for the second) , etc.: OUTPUTFILE1 = toto2 OUTPUTFILE2 = tutu2 the tutu2 and

character strings toto2 will then be replaced by the paths of the files defined by FICHIERS_SORTIE before the execution in Salome. Operand NOM_PARA \diamond NOM

3.5.8 _PARA = [" para1" , "

`para2 ",...]` This key word makes it possible to specify a list of variables (list of character string of text) whose values will be replaced by the values defined by the key word VALE. Same convention as for the output and input files must be respected for script: para1 = titi para2 = touched the lines containing

para1 =" and "para2 =" (for each parameter , only the first line met will be retained) will be identified like line modifying starting from the data of the key word VALE. Operand VALE \diamond VALE =

3.5.9 ["vale1", "vale

`2", ...]` This key word makes it possible to specify the values corresponding to the variables which were specified by NOM_PARA. From the convention of writing of script: para1 = titi para2 = touched the blocks on the right of

equal" sign (titi and touched) will be replaced by the values defined in the list VALE: para1 = vale1 para2 = vale

2 Operand
INFO \diamond INFO=info

3.6 If

```
INFO=2 ,
```

the messages coming of the command carried out are printed in the message file. It is the value by default. That makes it possible to preserve the trace of the execution during use EXEC_LOGICIEL/MAILLAGE, particularly in the benchmarks. Operand CODE_RETOUR_MAXI

3.7 \diamond CODE_maximum

```
RETOUR_MAXI=icode Value of the return code
```

returned by the command 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 return code of the command or of the software is ignored. Examples EXEC_LOGICIEL

4 is not

the only solution to call a command or a program, it is possible while inserting a command python os.system to carry out the same type of operation. Example of launching of a command

4.1.1 Call of basic a Unix

```
command: EXEC_LOGICIEL (LOGICIEL='ls
```

```
-la', ARGUMENT='/tmp',) remote Launching: EXEC
```

```
_LOGICIEL (LOGICIEL='ls -
```

```
al', ARGUMENT='/tmp', MACHINE_DISTANTE=_F (SSH_ADRESSE  
= "clau5aaa.der.edf.fr", SSH_PORT = 22 ,)); Example of  
creation
```

of a mesh Creation

4.1.2 of one mesh: EXEC_LOGICIEL

```
(MAILLAGE=_F (FORMAT='GIBI'
```

```
, UNITE_GEOM=17, UNITE=18, MAILLAGE=CO ("magibi  
"),), LOGICIEL='gibi  
"CODE_RETOUR_MAXI = 2, INFO  
=2,) Creation  
D" a mesh with format  
SALOME
```

```
. EXEC_LOGICIEL (MAILLAGE=_F (FORMAT='SALOME
```

```
", UNITE_GEOM=15, UNITE=21, MAILLAGE=CO ("mamed  
"),), ARGUMENT  
= 'cube.mmed  
", INFO=1) Example of launching  
of a script Salome
```

4.1.3 the following portion of script is given

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

in example: # Script /home/assire/test.py (postprocessing

```
in Salome) importation Salome importation VISU INPUTFILE1 = `/  
home/assire/louse  
` OUTPUTFILE  
  
1 = '' CHOICE = "$CHOIX$" PARA2  
= (continuation of script  
) the continuation of following  
command  
  
: IMPR_RESU (
```

FORMAT=' MED', UNITE=90, RESU=_F (

```
RESULTAT=res,)); EXEC_  
LOGICIEL (  
SALOME=_F (CHEMIN_SCRIPT =  
  
`/home/assire/test.py  
`, SALOME_HOST = `cli 75ca.der.edf.fr`, SALOME_PORT = 2811  
, FICHIERS_ENTREE = [`. /fort.90`],  
FICHIERS_SORTIE = [  
`. /fort.98`], SALOME_RUNAPPLI = `  
path-salome/runSalomeScript`, NOM_PARA  
= [`CHOICE`, `PARA2`], VALE = [`DEPL`, `10`  
]), INFO=2); Allows to send  
the script of postprocessing  
and med file
```

on the distant machine and to launch script in Salome, before recovering result script in the file fort.98.
The script modified by EXEC_LOGICIEL and carried out in

Salome will have the form: # Script /home/assire/test.py (postprocessing in Salome

```
) importation Salome importation VISU INPUTFILE1 = `/tmp/fort.90`  
OUTPUTFILE1 =  
`/tmp/fort.98  
  
` CHOICE = `DEPL` PARA2 = 10  
(continuation of script)
```