

Opérateur EXEC_LOGICIEL

1 But

Appeler depuis *Code_Aster*, un logiciel ou une commande système. Logiciel externe et commande système appelés doivent résider et seront exécutés sur la même machine que *Code_Aster*. Il est plus particulièrement destiné au développement de macro-commandes. Cette macro python s'appuie sur le module python `os` et plus particulièrement sur la commande `os.system`.

Cet opérateur permet aussi d'engendrer directement depuis le fichier de commande un maillage de type GMSH, GIBI ou SALOME. Cette possibilité a été introduite essentiellement pour être utilisée dans les tests de non-régression du code, la vérification du maillage restant indispensable pour pouvoir lancer une étude.

2 Syntaxe

```
EXEC_LOGICIEL (
    ◇ | LOGICIEL = nom_exe, [TX]
    | MAILLAGE = _F (
        ◆ FORMAT = / 'GMSH', [TX]
                    / 'GIBI',
                    / 'SALOME',
        ◇ UNITE_GEOM = igeom, [I]
        ◇ UNITE = unite, [I]
        ◆ MAILLAGE = ma, [maillage]
    )

    ◇ ARGUMENT = l_arg, [l_TX]

    ◇ CODE_RETOUR_MAXI = / icode, [I]
                        / 0, [DEFAULT]
    ◇ INFO = / 1, [I]
            / 2, [DEFAULT]
)
```

3 Opérandes

3.1 Opérande LOGICIEL

◇ LOGICIEL = nom_exe

Nom de la commande ou de l'exécutable à appeler. C'est une chaîne de caractères (entre '), il faut préciser l'ensemble du chemin (path) pour atteindre l'exécutable.

Cet opérande peut être utilisé pour surcharger la commande par défaut lors de la création d'un maillage.

Remarque :

Les logiciels référencés officiellement doivent posséder un point d'entrée sous un répertoire particulier géré par l'administrateur (répertoire `outils` de l'installation). Le développeur de macro-commande doit prendre la précaution de construire le nom en concaténant le nom du répertoire d'installation qu'il récupérera à l'aide d'un utilitaire afin d'en assurer la portabilité.

3.2 Mot clé ARGUMENT

◇ ARGUMENT = l_arg

Permet de définir la liste des arguments passés à l'exécutable. Ce mot clé est obligatoire dans le cas où l'on crée un maillage au format SALOME/MED.

Les valeurs sont passées sous forme de chaînes de caractères, à charge de l'exécutable de les décoder, elles sont transmises telles quelles, le caractère " (blanc) servant de séparateur.

3.3 Mot clé MAILLAGE

◇ MAILLAGE = _F (

Permet d'engendrer un maillage en appelant directement depuis le fichier de commandes l'un des outils suivants : GMSH, GIBI ou SALOME en fournissant le jeu de données dans un fichier. Les différents formats sont liés à la présence de la commande associée dans le répertoire d'installation de *Code_Aster*.

3.3.1 Opérande FORMAT

/ FORMAT = 'GMSH'

Création d'un maillage au format GMSH .

/ FORMAT = 'GIBI'

Création d'un maillage au format GIBI.

/ FORMAT = 'SALOME'

Création d'un maillage au format SALOME, ou plus généralement un fichier au format MED.

Dans ce cas le mot clé ARGUMENT est obligatoire et contient le nom du fichier MED produit par le script.

3.3.2 Opérande UNITE_GEOM

◇ UNITE_GEOM = igeom

Numéro d'unité logique associé au fichier de données utilisé pour créer le maillage.

Attention le fichier de données GIBI doit se terminer par la commande :

OPTI SAUV FORM 'fort.8'.

3.3.3 Opérande UNITE

◇ UNITE = unite

Numéro d'unité logique associé au fichier de résultat produit par l'outil de maillage.

3.3.4 Opérande MAILLAGE

◆ MAILLAGE = ma

Nom du concept maillage produit. Le nom doit être indiqué sous la forme syntaxique : CO ('MA')

3.4 Opérande INFO

◇ INFO = info

Dans le cas où INFO=2, les messages provenant de la commande exécutée sont imprimés dans le fichier MESSAGE. C'est la valeur par défaut. Cela permet de conserver la trace de l'exécution lors de l'utilisation EXEC_LOGICIEL/MAILLAGE, particulièrement dans les cas-tests.

3.5 Opérande CODE_RETOUR_MAXI

◇ CODE_RETOUR_MAXI = icode

Valeur maximum du code retour renvoyé par la commande ou le logiciel qui est tolérée pour considérer que l'exécution s'est bien déroulée. Par défaut cette valeur vaut 0, si elle est affectée à -1, le code retour de la commande ou du logiciel est ignoré.

4 Exemples

EXEC_LOGICIEL n'est pas la seule solution pour appeler une commande ou un programme, il est possible en insérant une commande python `os.system` d'effectuer le même type d'opération.

Appel d'une commande unix de base :

```
EXEC_LOGICIEL(LOGICIEL='ls -la', ARGUMENT='/tmp',)
```

Création d'un maillage :

```
EXEC_LOGICIEL(MAILLAGE=_F(FORMAT='GIBI',  
                           UNITE_GEOM=17, UNITE=18,  
                           MAILLAGE=CO('magibi')),  
              LOGICIEL='gibi'  
              CODE_RETOUR_MAXI = 2,  
              INFO=2,)
```

Création d'un maillage au format SALOME.

```
EXEC_LOGICIEL(MAILLAGE=_F(FORMAT='SALOME',  
                           UNITE_GEOM=15,  
                           UNITE=21,  
                           MAILLAGE=CO('mamed')),  
              ARGUMENT='cube.mmed',  
              INFO=1)
```