

Order END

1 Goal

This order properly finishes the execution of a calculation . In this manner, one indicates to the manager of memory that the objects can be released and written on disc in the total base (and allow one CONTINUATION).

The call to this procedure is **obligatory** so that the execution finishes correctly, what requires to finish any command file by:

```
END ( )
```

2 Syntax

```
END      (
          ◇  RETASSAGE =      /  'YES' ,
                               /  'NOT' ,           [DEFECT]
          ◇  INFO_RESU =      /  'YES' ,           [DEFECT]
                               /  'NOT'
          ◇  PROC0      =      /  'YES' ,           [DEFECT]
                               /  'NOT'
        )
```

An argument `exit` is available for the developers (see hereafter)

3 Operands

The order `END` safeguard the whole of the concepts calculated during the execution in the file `glob.1` (and possibly `glob.2`,... if necessary), as well as the whole of the python objects of the context of execution in file `S pick`. These objects will be available for one `CONTINUATION` calculation.

Notice

Are not saved in `pick.codeaster.objects`, the python objects of the type `classifies`, `function` and `type`.

When `END` finished its work, Python continuous to analyze the instructions which follow and which certainly will fail if they call on the `code_aster` objects.

The developer can use `END (exit=True)` to force to leave the interpreter Python.

3.1 Operand `RETASSAGE`

```
◇ RETASSAGE = / 'YES',  
              / 'NOT', [DEFECT]
```

Cause the retassage of the base `'TOTAL'` before writing on the associated file. This makes it possible to preserve smaller bases (removed from the objects associated with the concepts destroyed by the user).

This retassage is carried out as follows by the order:

- closing of the bases,
- opening of the base `'TOTAL'`,
- opening of a base `'VOLATILE'`,
- recopy, nonempty recording by nonempty recording of the base `'TOTAL'` on the basis `'VOLATILE'`,
- renaming by the code of this base `'VOLATILE'` for safeguard as if it were the base `'TOTAL'` classic.

The `TOTAL` base consists of one or more binary files organized in the form of fixed-length recordings. At the time of the operations of destruction, the associated recordings are declared free and can possibly be re-used in the course of execution. At the end of the execution, it can remain of the unoccupied recordings which contribute to the final size of the file. The operation of retassage thus consists in re-using this place by reorganizing the recordings. It is an operation which can involve many readings and writings on disc.

During an execution in parallel (MPI), it is each authority of the `TOTAL` base which is treated, which multiplies the inputs/outputs.

3.2 Operand `INFO_RESU`

```
◇ INFO_RESU = / 'YES', [DEFECT]  
              / 'NOT',
```

Cause the impression in the file `MESSAGE` relative information with the contents of the whole of the structures of data result stored in the base `TOTAL`.

Note:

The use of this keyword can increase in a consequent way the execution time of the order `END`, it is thus advised to modify the value by default when one carries out calculations generating of important quantities of data by their diversity.

3.3 Operand `PROC0`

The operand `PROC0` whose value by default is 'YES', allows to restrict the writing of the structures of data in the TOTAL base on the processor of row 0. If one affects the value to him 'NOT', the operations of safeguards will be carried out on all the processors. This operation can have high costs, it can thus be very penalizing to carry out it on each processor.

It can be necessary to carry out this safeguard on all the processors if one connects several command files (`COM`, `com1`, `com2`, etc...) in the same execution.

This keyword has direction only in parallel calculations and, except typical case, operation is automatic.

In distributed parallel mode (where each process sees only part of the model), all the processes make the writings (`PROC0=' NON'`).

In centralized parallel mode (classical parallelism), during the execution of the last command file, only the process #0 writes the base. During the intermediate executions, all the processes make the writings.

4 Example of impression resulting from the order `END`

The example below is extracted from the file `MESSAGE` associated with the case test `TTNL02A`.

```
# -----  
# ORDER NO: 0026          CONCEPT OF THE TYPE:  
# -----  
END (RETASSAGE=' NON',  
     INFO_RESU=' OUI',  
     PROC0=' OUI',  
     )
```

=====>

STRUCTURE OF THE CONCEPT TEMPLE CALCULATE FOR 15 SEQUENCE NUMBERS

LIST OF THE REFERENCE SYMBOLS:

```
! -----! -----! -----!  
! NUME_ORDRE!      TEMP      !  HYDR_ELGA      !  
! -----! -----! -----!  
!           0!      TEMP_R      !  HYDR_R          !  
!           1!      TEMP_R      !                   !  
!           ...!      ...          !                   !  
!           9!      TEMP_R      !                   !  
!          10!      TEMP_R      !  HYDR_R          !  
!          28!      TEMP_R      !                   !  
!           ...!      ...          !                   !  
!          118!      TEMP_R      !                   !  
! -----! -----! -----!
```

LIST OF THE NAMES OF VARIABLES OF ACCESS:

INST OF TYPE R

LIST OF THE NAMES OF PARAMETERS:

```
! -----! -----! -----! -----!  
! -----!  
! NUME_ORDRE!      MODEL      !  CHAMPMAT      !  CARAELEM      !  EXCIT          !  
! -----! -----! -----! -----!  
! -----!
```

Code_Aster

Version
default

Titre : Procédure FIN
Responsable : COURTOIS Mathieu

Date : 04/11/2021 Page : 5/5
Clé : U4.11.02 Révision :
5ad2df270126

```
!          0!          K8          !          K8          !          K8          !          K24          !  
!          1!          K8          !          K8          !          K8          !          K24          !  
!          ...!          ...          !          ...          !          ...          !          ...          !  
!          118!          K8          !          K8          !          K8          !          K24          !  
! ----- ! ----- ! ----- ! ----- !  
-----!
```