

Rules concerning the inputs/outputs

Summary:

This document lists the rules concerning the inputs/outputs which the developers must respect of *Aster*.

Contents

1 Introduction.....	3
2 Various files of Aster and their use.....	3
2.1 Databases: 'TOTAL', 'VOLATILE'.....	3
2.2 Files generals and essential.....	3
2.3 Particular files.....	4
3 Various types of messages.....	4
3.1 The error.....	4
3.2 Alarm.....	4
3.3 The result.....	4
3.4 The echo of the data.....	4
3.5 INFORMATION.....	5
4 Read/write in a file different from 'ERROR' or 'MESSAGE'.....	5
5 Emission of error message or alarm.....	6
6 Writing of messages of information, keyword INFORMATION.....	6

1 Introduction

The orders Aster exchange data. Generally these data are Structures of Data (or concepts "user") for which, the programmer does not have reasons to make "READ" (for its data) nor of "WRITE" (for its results). In this case, the "left" "entrances"/are made by JEVEUX [D6.02.01].

It happens however that certain orders (in general of the procedures) have to read data on a file or to write a result. The rules concerning these orders are given to the §4.

With the occasion of a calculation, an order can want to transmit an error message or of alarm, in this case, it will use the "package" UTMESS [D6.04.01] . One will speak again about it to the §5.

Lastly, an order can want to write messages of information concerning the course of calculation. These impressions are controlled by the keyword INFORMATION order. These impressions are the object of the §3.5 for which one uses the package INFXXX [D6.04.02]).

2 Various files of Aster and their use

2.1 Databases: 'TOTAL', 'VOLATILE'

(IN/OUT) they are the files with direct access managed by JEVEUX. JEVEUX is the only one with reading/writing on these files.

2.2 Files generals and essential

- (IN) command file (.COM) and (of include) : these files are only read by the supervisor,
- (OUT) file 'error': only UTMESS can write there; one finds a trace of the errors there,
- (OUT) file 'MESSAGE': one finds there:
 - impressions of UTMESS,
 - impressions of INFORMATION (see §3.5),
 - echoes of the orders by the supervisor,
- (OUT) file 'RESULT': one finds there information asked explicitly by L' user: impression with the format "Aster" of the results (orders IMPR_RESU,...), is added to that:
 - impressions of UTMESS : alarms and errors,
 - the summary of the times spent in the various orders.

Note:

Approximately, the difference between the files 'MESSAGE ' and 'RESULT ' is the following one:
→ the file 'RESULTAT' contains information interesting the silent partner of the study,
→ the file 'MESSAGE' contains information interesting that which carries out L' study to control the courses of calculations.

2.3 Particular files

They are the other files. They are known as “individuals” because they are used only by some orders or procedures. For example:

- file grid Aster (.e-mail),
- file Gibi grid (.mgib),
- ...

3 Various types of messages

When a programmer wishes to print information in a file, it is important for him to try “to typify” this information, because from this type, the name rises from the file where one prints and the way of printing: `UTMESS` or `WRITE`.

3.1 The error

It prevents the continuation of calculation. One must always use `UTMESS <F>` (or `<E>`) (cf [D6.04.01]).

3.2 Alarm

The continuation of calculation is possible but the use is strongly disadvised;

Examples: modulus Young negative, affections in double,...

It is necessary to use `UTMESS <A>`

3.3 The result

It is a result of the study. He is asked explicitly by the user.

It is necessary to use `WRITE (IFR, ...)`...

where `IFR` is the logical unit of the file result:

- ‘RESULT’ (with the format *Aster*),
- ‘IDEAS’ (with format *IDEAS*),
- ...

3.4 The echo of the data

Examples:

“You chose the method `TRICK`”

“For the option ‘`SMALLER`’, the frequencies are ignored.

These impressions must be avoided: they do nothing but repeat what the user wrote or what must be known as in the Documentation of Use (Handbook U).

3.5 INFORMATION

It is information which relates to the course of calculation.

Examples:

- iteration count to converge,
- coefficient of conditioning of Lagranges,
- convergence criteria reached,...

It can be also "data-processing" information more:

- size memory (or disc) of a matrix
- time spent in the factorization of a matrix,
- ...

It can be finally information intended to reassure the user (confirmation of its data):

- many nodes, of meshes,
- list of nodes likely to make contact.

The writing of `INFORMATION` is made by `WRITE` on the file 'message' it is necessary to comply with the rules of the §6.

Information of the type `INFORMATION` are associated with a level (1 or 2): one `INFORMATION` of level 1 is more important than one `INFORMATION` of level 2. `INFORMATION` of level 1 will be described systematically in the Instruction manual (cf [D6.04.02]) They are contractual.

- When the user asks `INFORMATION : 1` (defect), one prints them `INFORMATION` of level 1.
- When he asks `INFORMATION : 2`, they are printed `INFORMATION` of level 1 and level 2.

4 Read/write in a file different from 'ERROR' or 'MESSAGE'

Orders making of the read/writes in a file different from the files 'ERROR' or 'MESSAGE' are of limited number.

- If an order reads a file, one will try to give him a name of the form `LIRE_XXX` :
`LIRE_MALLAGE, LIRE_FONCTION,...`
- if an order writes in a file, one will try to give him a name of the form `IMPR_XXX` :
`IMPR_RESU, IMPR_TABLE,...`
- orders of interface of entry: `PRE_IDEAS, PRE_GIBI,...` read in general on a file and write in another.
- The orders should not read and write in files except if they are conceived for that (cf R2). The files should not be used to exchange information between orders (the SD are there for that) nor to make "pagination" memory (it is `JEVEUX` who undertakes some).
- The orders which read/write in files do it always explicitly. It be-have-to say that they use standard keyword:
`/ FILE` : when the file is named
`/ UNITE_XXX` : when one uses the logical unit of the file except the orders `LIRE_XXX` (which is operators) the other orders which read and write in files are all the procedures (`IMPR_XXX, PRE_XXX,...`) what wants to say that their role is to read and/or write.

5 Emission of error message or alarm

- All order can transmit an error message (more or less immediate stop) or of alarm (one continues the execution). For that, it will use exclusively routine `UTMESS [D6.04.01]`.
The transmitted messages will go then automatically in "the good" given pre files: 'ERROR', 'MESSAGE' and 'RESULT'.

6 Writing of messages of information, keyword `INFORMATION`

In this paragraph, one calls `INFORMATION`, a message of information.

- One `INFORMATION` is always written by one `WRITE`. The logical unit (of the file 'MESSAGE') is always recovered by the routine `INFNIV`.
- One `INFORMATION` a level has: 1 or 2. `INFORMATION` of level 1 are contractual; they are described in the Documentation of Use. The person in charge of the order must present in meeting EDA any evolution of `INFORMATION` of level 1.
- The level of impression chosen by the user (1 or 2) is always via the keyword `INFORMATION : / 1 / 2`. Value 1 is always the value by default.
When the user asks `INFORMATION : 2`, that wants to say that it wishes them `INFORMATION` of level 1 and level 2.
- Treatment of the keyword `INFORMATION` in an order `OP00XX : CAL INFMAJ ()`
- Impression of one `INFORMATION` of level 1:

```
CAL INFNIV (INF, NIV) ! recovery of the logical unit and the level of  
impression asked  
IF (NIV.GE.1) WRITE (INF, FMT) ...
```
- Impression of `INFORMATION` of level 2:

```
CAL INFNN (INF, NIV)  
IF (NIV.EQ.2) WRITE (INF, FMT)...
```