

## Procédure DEFI\_FICHER

---

### 1 Goal

---

To open or close a file associated with a number with logical unit. This number can be indicated in the procedure or obtained in return of the latter. This action can be carried out constantly during work. This procedure makes it possible to associate a logical number of unit with a file FORTRAN of the ASCII type, but this concept can also apply with certain restrictions on files of the binary type.

## 2 Syntax

---

```
[nfic [whole] =] DEFI_FICHER (  
    ◇ ACTION      = / 'TO ASSOCIATE',           [DEFECT]  
                  / 'TO RELEASE',  
                  / 'TO RESERVE',  
  
    ◇ FILE        = nomfic,                       [K255]  
  
    ◇ UNIT        = numul,                        [I]  
  
    ◇ TYPE        = / 'ASCII',                   [DEFECT]  
                  / 'FREE',  
    ◇ ACCESS     = / 'NEW',                       [DEFECT]  
                  / 'OLD',  
                  / 'SUSPENDS',  
  
    ◇ INFORMATION = / 1,  
                  / 2  
  
)
```

The obligatory or optional character of certain operands depends on the presence or the value associated with the keywords indicated previously.

This operator has the characteristic to function at the same time like an order by providing a result of the whole type reusable behind a keyword of a further order, that is to say like a procedure.

## 3 Operation of `DEFI_FICHIER`

Inputs/outputs on the ASCII files since *Code\_Aster* are realized for the majority, using instructions FORTRAN using the concept of **logical unit**, it is a variable entirety from 1 to 99 associate using an instruction of the "open" type to a file. By default this number `xx` is associated with the name `fort.xx`, but it is possible to associate it with an unspecified file name. The logical number of unit is used in the interface of access to the code `astk` to recopy the files as starter and in exit.

In the command file *Aster* the various operators carrying out of the readings or the writings admit the keyword `UNIT` like argument.

logical unit 6 is reserved by the code at the time of the phase of initialization. It is not authorized to modify this number of logical unit 6 which makes it possible to trace the course of an execution.

logical unit	name of the file	authorized modification
6	MESSAGE	not

The user can wish to add or **to modify** associations using the procedure `DEFI_FICHIER` to use for example new file names, to print certain results or to differently gather them in files. Procedure `DEFI_FICHIER` allows moreover to directly indicate the file of the ASCII type which will be associated with the specified logical unit. It can be specified either by a name in absolute (limited to 255 characters) if the file is localised on the machine, or by a relative name in an agreed repertoire (`./REPE_IN` or `./REPE_OUT`) when the interface takes care of the distant and total transfer of the whole of the files located under the repertoire (standard `repe` in `astk`). One `OPEN` Named FORTRAN is then carried out on the files of the ASCII type. The order makes it possible of more than position either at the top of file, or at the end of the file.

The code manages in-house a structure of data gathering the unit of associations logical unit – file name – type of file - type of access.

## 4 Operands

The modifications of association unit logic-name of file relates to the files of **exit** and **ofentry**.

### 4.1 Operand **ACTION**

◇ `ACTION = 'TO ASSOCIATE'`

The logical number of unit is associated, when that is allowed, the name defined behind the keyword `FILE` if it is indicated, with the name `fort.xx` if not.

It is not possible to redefine associations of the logical numbers 6 and 9.

◇ `ACTION = 'TO RELEASE'`

The logical number of unit is released, it is not more licit to use this number of unit, the associated file, when it is of ASCII type, is the object of an order of closing using instruction FORTRAN `CLOSED`. It then becomes possible to re-use the logical number of unit with an action of the type 'TO RELEASE'. This mechanism is essential if L' one wants to be able D-to immediately exploit the contents of the file associated in the command file in progress, indeed the buffers must be completely emptied and the file must be able to be accessible, possibly at the time of a call to a software since an order python of the type `os.system`.

◇ `ACTION = 'TO RESERVE'`

This kind of action is used in the macros orders and makes it possible to avoid the conflicts of logical number of unit between FORTRAN and python.

The logical number of unit is associated, when that is allowed, the name defined behind the keyword `FILE` if it is indicated, with the name `fort.xx` if not. There is no instruction of the `OPEN` type carried out, with load for the macro order to carry out the various actions necessary.

## 4.2 Operand `FILE`

◇ `FILE = nomfic`

Physical name of the file (< 255 characters) which one wishes to associate with a logical unit. This file will be created under the repertoire of execution of the code, but one can directly indicate a file name (respecting UNIX conventions) in the repertoire of the user. Under the repertoire of execution, it is possible to use an additional level of tree structure of conventional name `REPE_IN` (data files) or `REPE_OUT` (files of results) recognized by the interface of access to the code `astk`. This name must be placed between quotes. Although they are not associated with a logical unit by an `OPEN` order `FORTTRAN`, the binary files (for example `MED`) can be treated with this mechanism, it is necessary nevertheless to specify the type of access `NEW` or `OLD` to activate the recopy (by a call system since the code) since the data directory or repertoire as a result.

When the operand is absent, it is by default the file name `fort.ul` where `ul` is the number of unit which is associated with the logical unit definite behind `UNIT`.

For the files of the ASCII type, an instruction `OPEN FORTTRAN` is carried out on the name associated with the logical unit.

## 4.3 Operand `UNIT`

◇ `UNIT = numul`

Number of logical unit associated, its value for the ASCII file opened by instructions lies between 1 and 99 included.

It is possible to re-use an already affected number but in this case it is necessary to take the precaution to release this last before. Certain numbers of logical units cannot be redefined since the orders Aster, it is the numbers 6 and 9 which are respectively allocated with the files `MESSAGE` and `ERROR`.

This number can be used then in all the orders Aster which have the operand `UNIT` (`IMPR_RESU`, `IMPR_TABLE`, `IMPR_FONCTION`, etc).

The operand `UNIT` can sometimes be omitted, it is then the code which will choose to affect a number, according to the availabilities, it is then necessary imperatively to specify the operand `FILE`, the code is given the responsibility then in-house to associate the logical number of unit and the associated file. It is then possible to recover a whole value at exit of the operator.

## 4.4 Operand `TYPE`

◇ `TYPE = 'ASCII'`

The file associated with the logical unit is of ASCII type.

◇ `TYPE = 'FREE'`

The file associated with the logical unit is of unspecified type within the meaning of `FORTTRAN`, that makes it possible to manage in a more flexible way the access to the file, this type is primarily used for the access to files `MED`. The logical unit is not really used in this case, but that makes it possible to have the convention of name `fort.ul` on the file and being able to easily transmit it through the interface of access to the code.

## 4.5 Operand ACCESS

◇ ACCESS = 'NEW'

The file of the "ASCII" type is opened and one positions at the head, a Fortan instruction of the type `REWIND` is carried out. If there exists, the file is crushed. To use for a file as a result.

◇ ACCESS = 'OLD'

The file of the "ASCII" type is opened and one positions such as it is, i.e. possibly with the position at which one was located at the time of the last access.  
To use for a data file.

◇ ACCESS = 'SUSPENDS'

The file is opened and one positions at the end of the file. Is used only in the case of the files of the ASCII type.  
To use for a data file which one wants to supplement (who will be thus a priori as a result).

## 4.6 Operand INFORMATION

◇ INFORMATION = inf

Allows to print in the file `MESSAGE` the list of the open logical units with the order `DEFI_FICHER` as well as the associated parameters. If `INFORMATION = 1`, there is no impression.

## 5 Declaration in the interface of access to the code of the logical unit put in work in `DEFI_FICHER`

Generally the user will call `DEFI_FICHER` in order to carry out postprocessings when it tests the need to physically create several files of results according to the loading cases, of the sizes, the steps or moments of evolution of calculation.

The user must declare the names physical of the files and the associated logical units. This declaration is carried out, in the interface `astk` prior to launching of the completion of the work. It is necessary to add the files in the profile of study by selecting the type `libr` and to associate the number of logical unit to them selected.

The type `repe` is used in the interface `astk` to transmit or receive all the contents of a file directory, with load for the user to call in the command file on `DEFI_FICHER` to carry out association with the logical number of unit.

For convention the data files are transmitted in the repertoire of nonroom `REPE_IN`, the files as a result are transmitted in the repertoire of local name `REPE_OUT`.

In the order `DEFI_FICHER`, the name passed behind `FILE` is form: `./REPE_IN/mon_fichier`.

For a data file, one will make for example:

```
DEFI_FICHER (FICHER=' ./REPE_IN/FILE.DATA',  
            ACCES=' OLD',  
            UNITE=11)  
LIRE_FONCTION (UNITE=11,...)
```

### Notice

*INCLUDE cannot be used with a logical unit associated with a particular file name.*