

Operator LIRE_TABLE

1 Goal

To see a file containing one `table`.

This order makes it possible to read a table written in a file under the format `TABLE`, `ASTER` or a format known as `FREE`.

With the formats `TABLE` and `ASTER`, the file can come directly from one `IMPR_TABLE` with the same format.

The format `FREE`, as its name indicates it, is more flexible. It is necessary however to observe certain conditions so that the columns are correctly identified and to get the expected result.

Product a concept of the type `table`.

2 Syntax

```
Tb [table] = LIRE_TABLE(  
    ♦ UNIT          = ific ,                [I]  
    ♦ FORMAT        = / 'TABLE' ,          [DEFECT]  
                    / 'ASTER' ,  
                    / 'FREE' ,  
    ♦ NUME_TABLE    = / 1 ,                [DEFECT]  
                    / digital ,           [I]  
    ♦ SEPARATOR     = / ``                [DEFECT]  
                    / sep ,              [K]  
    ♦ RENOMME_PARA  = 'SINGLE'             [K]  
    ♦ TITLE         = title ,              [1_K]  
    ♦ INFORMATION   = / 1 ,                [DEFECT]  
                    / 2 ,                [I]  
)
```

3 Operands

3.1 Operand UNIT

Logical number of unit of the file containing the tables with reading.

3.2 Operand FORMAT

Name of the format under which the table of the file was written.

3.2.1 Remarks

During the reading of the values, one tries to interpret the value like whole, real, real with the comma instead of the point like decimal separator, failing this like a character string.

The value of blank cell is not the same one with the format TABLE, ASTER and FREE.

3.2.2 Format TABLE

It is the format by default.

With the format TABLE, the order estimates the number of columns present in the file. For that, the lines are cut out by using it SEPARATOR. The number of columns of the table is then supposed being the maximum found on the unit of the lines.

The lines starting with character # are regarded as comments and are added to the title of the table.

Among the values one can find (-), who means that there is no value for this parameter.

There are no delimitations to separate several tables in a file. It is considered that the end of a table was reached when the number of columns falls.

The format TABLE must allow to read again the tables printed with IMPR_TABLE with the same format. However, contrary to the format ASTER, the type is automatically given according to the values read. The type of character strings can thus be different.

3.2.3 Format ASTER

The format ASTER requires certain characteristics illustrated in the following table:

```
#DEBUT_TABLE
#TITRE line 1 of the text of the title
#TITRE line 2 and following
ENTITLE   CHEMIN   SEGMENT  RESU      NOM_CHAM  INST      SIXX
K8        K16          I         K8        K16       R          R
TOTAL    CHEM1       1         SIG      SIGM_ELNO 0.00000E+00 - 1.48981E+07
TOTAL    CHEM1       -         SIG      -         1.00000E+00 - 2.48981E+07
TOTAL    CHEM1       2         SIG      SIGM_ELNO 1.00000E+00 - 3.48981E+07
...
#FIN_TABLE
```

Any table with the format ASTER begin with the keyword #DEBUT_TABLE and ends in #FIN_TABLE. In fact the delimitations make it possible to know where the reading starts and where it stops.

#DEBUT_TABLE is then followed by lines of titles which are obligatorily preceded by the keyword #TITRE.

Follow then a line containing all the names of the parameters, then a line containing all the types of these parameters. These types belong obligatorily to the following list: K8, K16, K24, K32, K80, I, R.

The following lines give the value of all the parameters.

Among the values one can find (-), who means that there is no value for this parameter.

The format ASTER must allow to read again the tables printed with IMPR_TABLE with the same format.

3.2.4 Format **FREE**

The format **FREE** is very close to the format **TABLE** except that character **#** does not have particular meaning and that the absence of value for a parameter corresponds to a null string.

3.3 Operand **NUME_TABLE**

It is possible to read a table among several in a file. One indicates the number of the *i*ème table to reading. By default, the first table is read.

3.4 Operand **SEPARATOR**

Several consecutive separators count only for one. The separator by default is a space.
To indicate that the fields are separated by a tabulation, one will indicate: `SEPARATEUR=' \ you`.

3.5 Operand **RENOMME_PARA**

In a table, all the parameters must be distinct. If it is not the case in the file with reading, the reading fails because the table cannot be produced.
While specifying, `RENOMME_PARA = 'SINGLE'` (only possible choice if the keyword is indicated), if a parameter read were already met, one adds a numbered suffix of the form to him: "`_i`".
Example: two columns are named `INST` in the file. The table will be produced with a named parameter `INST` and the named second `INST_1`.

3.6 Operand **TITLE**

Title attached to the concept produced by this operand [U4.03.01].
This title replaces that of the table read.

3.7 Operand **INFORMATION**

With `INFORMATION = 2`, details on the data read are displayed. They are very numerous and may find it beneficial only to understand why a file is not read as it was expected.

4 Examples

4.1 Creation of a table to the format '**ASTER**' with **IMPR_TABLE**

In this example (extracted from zzz128b), one writes several tables with the format '**ASTER**' in a file.

```
TOUT=POST_RELEVET (ACTION=_F (ENTITLES = 'RESU_U2',  
                                NODE = ('B', 'IT', 'OF', 'E', 'F',),  
                                RESULT = U2,  
                                NOM_CHAM = 'DEPL',  
                                TOUT_ORDRE = 'YES',  
                                TOUT_CMP = 'YES',  
                                OPERATION = 'EXTRACTION'))
```

```
IMPR_TABLE (UNITE=37,  
            TABLE=TOUT,  
            FORMAT=' ASTER')
```

```
IMPR_TABLE (UNITE=37,  
            TABLE=TOUT,
```

```
FORMAT=' ASTER',  
NOM_PARA= ('NODE', 'INST', 'ABSC_CURV',  
           'DX', 'DY', 'DZ', 'DRX', )  
)
```

4.2 Reading of tables to the format 'ASTER'with LIRE_TABLE

```
TT2=LIRE_TABLE (UNITE=37,  
               FORMAT=' ASTER',  
               NUME_TABLE=1,  
               TITRE=' VALIDATION OF LIRE_TABLE',)  
  
TT_RED=LIRE_TABLE (UNITE=37,  
                  FORMAT=' ASTER',  
                  NUME_TABLE=2,  
                  TITRE=' VALIDATION OF LIRE_TABLE',  
                  )
```

4.3 Reading of table to the FREE format

That is to say for example a table such as this one in OpenOffice Calc:

| | | |
|---|----------|--------------|
| Table produced in OpenOffice | | |
| recorded with the format csv with the semicolon like separator | | |
| and without separator of text | | |
| | | |
| Column A | Column B | Column C |
| label1 | 111 | 0.09 |
| name 2 | 222 | 10.09 |
| valeur3 | 444 | 2240.18 |
| exemple4 | 888 | 994640.73 |
| ligne5 | 1776 | 883240965.82 |

One records it in format text (format CSV) with the semicolon like separator of columns and without framing the character strings (without separator of text).

The following textual file is obtained:

```
Table produced in OpenOffice; ;  
record with the format csv with the semicolon like separator; ;  
and without separator of text; ;  
; ;  
Column A; Column B; Column C  
label1; 111; 0.09  
name 2; 222; 10.09  
valeur3; 444; 2240.18  
exemple4; 888; 994640.73  
ligne5; 1776; 883240965.82
```

To read it in Code_Aster, one can make:

```
= LIRE_TABLE (UNITE=55,  
             FORMAT=' LIBRE',  
             SEPARATEUR='; ',)
```

