
Procedure TEST_FICHER

1 Goal

This macro-order makes it possible to test to it not regression of files produced by the Code_Aster orders, mainly `IMPR_RESU`, but one can use it on any textual file.

To test a file, one extracts the list from the real numbers and whole present in the file, the cardinal of this list, and the whole of the remaining text (once extracted numbers).

For the numbers, one checks to it not regression of the sum, the absolute sum or the min/max of the values except for a precision.

To test the text of the file, the module Python `md5` is used which provides a "signature" of the text (called `md5sum`).

Turn over `OK` if the file is identical to the file which was used as reference, `NOOK` if not.

This order is used primarily by the cases tests to validate the not-regression of the results.

2 Syntax

```
TEST_FICHER (
    ◆ FILE = fich, [KN]
    ◇ EXPR_IGNORE = regexp, [1_Kn]
      TYPE_TEST = / 'SOMM', [DEFECT]
                  / 'SOMM_ABS',
                  / 'MAXIMUM',
                  / 'MINI',
                  / 'MAXI_ABS',
                  / 'MINI_ABS',
    ◆ NB_VALE = nbval, [I]

    # Definition of the value of reference:
    # to see TEST_RESU [u4.92.01]

    ◇ INFORMATION = / 1,
      [DEFECT]
                  / 2,

    )
```

3 General information

This procedure makes it possible to test to it not regression of files produced by the orders Aster, mainly IMPR_RESU, but one can use it on any textual file.

To test a file, one extracts the list from the real numbers and the list of entireties present in the file, the cardinal of these listS, and the whole of the remaining text (once extracted numbers).

For the numbers, one checks to it not regression of the sum, the absolute sum or the min/max of the values except for a precision. The real numbers and entireties are treated separately.

One also checks (and it is the only obligatory test) the cardinal of the list of the real numbers identified in the file.

To test the text of the file, the module Python is used md5 who provides a "signature" of the text (called md5sum). Thus, any variation of a text (title, name of a component) compared to a file of reference will involve a change of the signature of the file.

Notice 1:

| The file must be closed so that the value is reliable (with for example DEFI_FICHER, ACTION = 'TO RELEASE').

Notice 2:

| The order does not give a relevant information about binary files.

The function `test_file` can be called out of *Code_Aster* to obtain the value of md5sum of a file:

```
iret, sum = test_file (filename=' nom_fichier',  
                      type_test=' SOMM',  
                      verbose=True)
```

Notice 3

| For the integers, the value tested on entireties is truncated with the greatest entirety which can be represented in 32 bits (approximately 2.e9).

4 Operands

4.1 Operand FILE

◆ FILE = fich,

One provides the name of the file here to be analyzed with his access path into relative compared to the repertoire of execution or absolute. Examples: 'fort.37' or './REPE_OUT/resultats.pos'.

4.2 Operand EXPR_IGNORE

◇ EXPR_IGNORE = regexp

The lines of the file satisfying the regular expressions given behind this keyword will be ignored in the analysis of the file.

Example: *

```
EXPR_IGNORE = ( 'DATE= [0-9] {2}/[0-9] {2}/[0-9] {4}',  
               '^VERSION' )
```

Here lines containing DATE=jj/mm/aaaa where *j*, *m*, *a* are figures or starting with VERSION are ignored.

4.3 Operand TYPE_TEST

Type of the test made on the actual values and whole of the file. The possible values are:

- SOMM : nap of the values
- SOMM_ABS : nap of the absolute values
- MAXIMUM : maximum value
- MINIS : minimal value
- MAXI_ABS : maximum of the absolute values
- MINI_ABS : minimum of the absolute values

4.4 OperandS NB_VALE and NB_VALE_I

NB_VALE is it many actual values expected in the file. It is the only obligatory test.

NB_VALE_I is the number of whole values expected in the file.

4.5 Keywords common to the orders TEST_XXX

The definition of the values of nonregression and reference, as well as acceptable tolerances, the comparison criterion is detailed in the documentation [u4.92.01] of the order TEST_RESU.

Specificities of TEST_FICHER are:

- complex pas de valeurs,
- pas de test in absolute value (except with TYPE_TEST),
- VALE_CALC, VALE_CALC_I and VALE_CALC_K can be provided together,
- pas de tolerance on the value of the parameter (TOLE_MACHINE and CRITERION takes only one value).
- VALE_CALC provides the value of reference expected for the test on the actual values (the sum or maximum or...).
- VALE_CALC_I provides the value of reference expected for the test on the whole values (the sum or maximum or...).
- VALE_CALC_K the signature md5 text of the file provides after the all actual values and whole were withdrawn. To obtain the value on the file of reference, it is enough to launch the order TEST_FICHER and to record the displayed value (it is about a continuation of 32 hexadecimal natures).

4.6 Operand INFORMATION

◇ INFORMATION = inf

Specify the detail of the information printed in the file message.

If inf=1, there is the following summary:

(extracted from ssls108a)

```
Name of the file      : ./REPE_OUT/DEP12.pos

                Calculated                Reference
Entireties:
Many values: 176                176
Somme of the values: 878                878
Realities:
Many values: 10375                10375
Somme of the values: 1.5553683808e+04    1.5553683808e+04
Checksum: e5050b2a3517728c4cc0e23af2b16ba5 not_tested
```

If inf=2, there is the list of the all values recorded in the file, read per block (5897 values real and 118 whole values in the 3rd block of this example):

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

Real numbers: 5897

['2.1774365E-003', '2.4554530E-003', '2.4552944E-003', '7.0000000E+000',...]

Integers: 118

['1', '0', '0', '0', '0', '0', '0', '0', '144', '0', '0', '0', '0', '0', '0', '0',...]

as well as the whole of the text remaining once all extracted numbers:

Text:

\$EndView\$ViewDEFZ_DRX