
Procedure IMPR_CO

1 Goal

To print all the objects `JEVEUX` who constitute an existing concept.

This order is intended above all for the “debugage”. Contents of the objects `JEVEUX` that one prints comprehensible in theory only by developers is not provided with their documentation D.

In particular, the format of these objects can change without dialogue with the users. It is thus very likely to use these writings in “postprocessings” or “chainings” towards other codes.

2 Syntax

```
IMPR_CO      (
# selection of the objects to be printed:
♦ /  CONCEPT = _F (
    ♦ NAME      = concept,
      [1_nom_concept]
    )
  /  ALL        = 'YES'
  /  ♦ CHAIN    = chain,          [K24]
    ♦ POSITION   = / ipos,        [I]
                      / 1 ,      [DEFECT]

♦ ATTRIBUTE = / 'NOT',          [DEFECT]
              / 'YES',

♦ CONTENTS  = / 'YES',          [DEFECT]
              / 'NOT',

♦ BASE      = / 'G',            [DEFECT]
              / 'V',
              / '',

♦ LEVEL     = / 2,              [DEFECT]
              / 1,
              / 0,
              / -1,

# If LEVEL = -1
    ♦ PERMUTATION = / 'YES',    [DEFECT]
                      / 'NOT'

♦ UNIT      = / numfi          [I]
              / 8,            [DEFECT]
)
```

3 Operands

3.1 Keyword factor **CONCEPT**

The user indicates behind the keyword `NAME` , the list of the concepts which he wants to print.

3.2 Operand **ALL = 'YES'**

Allows to print all the objects present on a basis (or on all the bases). Attention, these impressions can be very bulky.

3.3 Operands **CHAIN / POSITION**

/ ♦ `CHAIN`

The user can give a text (between "quotes") length lower than 24 characters: the order will then print all the objects having in their name the character string: `chain` in position `ipos` (see keyword `POSITION`).

Examples:

1) `CHAIN = 'TOTO1234' , POSITION = 1`
is equivalent to `CO = TOTO1234`

2) `CHAIN = '.DESC' , POSITION = 20,`
allows to print all the objects whose name ends in `'.DESC'` .

♦ `POSITION = ipos,`

This entirety defines the position of the beginning of the chain to be sought in the name of the existing objects.

3.4 Operand **ATTRIBUTE**

♦ `ATTRIBUTE =`

One can choose to print (or not) the attributes of the objects `JEVEUX` selected.

3.5 Operand **CONTENTS**

♦ `CONTENTS =`

One can choose to print (or not) the contents of the selected objects.

3.6 Operand **BASE**

♦ `BASE = bases`

Only the objects will be printed `JEVEUX` found on the basis `base`. If one wants to obtain the objects being on the whole of the bases (`'G'` and `'V'`) , one will use `BASE = ' ('` ("white" character).

3.7 Operand LEVEL

◇ LEVEL =

0 : only the names of the objects are printed,

1 : one prints only the first 5 objects of each collection of objects JEVEUX,

2 : all is printed.

-1 : One prints for each object a single line containing:

- the name of the object
- some characteristics: LONMAX, LONUTI, TYPE
- a number (SOMMI or SOMMR) who "summarizes" the whole of the object.

These impressions make it possible for example to help to compare 2 versions different from the code. For that:

- ◆ To add at the end of the command file: IMPR_CO (TOUT=' OUI', NIVEAU=-1)
- ◆ to make "turn" the 2 versions of the code with this command set
- ◆ to make a "diff" of the 2 files produced results.

3.8 Operand PERMUTATION

◇ PERMUTATION = / 'YES', [DEFECT]
/ 'NOT'

This keyword can be present only when LEVEL = -1. It makes it possible to make depend the numbers SOMMI or SOMMR (mentioned in the paragraph of the operand LEVEL) order in which the elements of the object are arranged.

- 'YES' : SOMMI or SOMMR will depend of this kind
- 'NOT' : SOMMI or SOMMR will not depend of this kind

3.9 Operand UNIT

◇ UNIT =

Logical number of unit associated with the file on which must be printed the selected objects

By default, the impression is carried out on the unit which is associated with the file `fort.8`.

4 Examples

Example 1

```
IMPR_CO (CONCEPT = _F (NOM= (my, Mo, ch1)))
```

Impression of the concepts: `my, Mo, ch1`. The attributes of the objects will not be written.

Example 2

```
IMPR_CO (CHAIN = '&', POSITION = 1, LEVEL = 0, BASE = 'V',)
```

Impression of the names of the objects of the volatile base starting with '&'.