Procedure POURSUITE

Usage

Progress a study starting from the backup of its base 'GLOBALE' with format JEVEUX or format HDF.

User should not worry about the apparent complex syntax of this procedure as a simple call of POURSUITE() is sufficient in most cases by default.

The use of this command is completely similar to that of DEBUT. For the keywords which are identical between DEBUT and POURSUITE, refer to [U4.11.01]

Table of Contents

Syntax

```
POURSUITE(
      ◇PAR_LOT        = / 'OUI',                              [DEFAULT]
                        / 'NON',
      ◇IMPR_MACRO     = / 'NON',                              [DEFAULT]
                        / 'OUI',
      ◇LANG = Lang,                                           [TXM]
      ◇BASE = _F ( ◆FICHIER = 'VOLATILE',
                           ◇ /| LONG_ENRE = lenr,             [I]
                              | NMAX_ENRE = nenr,             [I]
                              | LONG_REPE = lrep,             [I]
                 ),
      ◇CODE = / 'NON',                                        [DEFAULT]
              / 'OUI',
      ◇ERREUR = _F ( ERREUR_F = / 'ABORT',                    [DEFAULT]
                                / 'EXCEPTION',
                   ),
      ◇IGNORE_ALARM = l_vale,                                 [l_Kn]
      ◇DEBUG = _F ( ◇JXVERI =         / 'OUI',
                                      / 'NON',
                    ◇ENVIMA =           'TEST',               [l_Kn]
                    ◇JEVEUX =         / 'OUI',
                                      / 'NON',
                    ◇SDVERI =         / 'OUI',
                                      / 'NON',
                    ◇HIST_ETAPE =     / 'NON',
                                      / 'OUI',
                  ),
      ◇MESURE_TEMPS = _F ( ◇NIVE_DETAIL = / 0,                [DEFAULT]
                                          / 2, /1, /3
                           ◇MOYENNE = / 'NON',                [DEFAULT]
                                      / 'OUI',
                         ),
      ◇MOMOIRE = _F ( ◇TAILLE_BLOC = / 800.,                  [DEFAULT]
                                     = / tbloc,               [R]
                    ),
      ◇RESERVE_CPU = _F ( /VALE = vale,                       [R]
                          /POURCENTAGE = pcent,               [R]
                          ◇BORNE = / bv,                      [R]
                                   / 180,                     [DEFAULT]
                        ),
      ◇FORMAT_HDF = / 'NON',                                  [DEFAULT]
```

```
                    / 'OUI',
);
```

Principle of operation

This procedure affects, moreover, the necessary memory resources required for the continuation of computation.

The operands of this command are similar to those of DEBUT [U4.11.01]. Here it is possible to specify certain resources assigned to the new study.

The study carried out previously continues with a set of commands starting with POURSUITE and ending with FIN [U4.11.02].

The commands placed before POURSUITE (except obviously DEBUT) or after FIN are ignored even if they are syntactically correct.

When the supervisor executes the command POURSUITE, it carries out the following tasks:

- definition of the logical units of the files used for printing
- assignment of the files associated with the databases managed by JEVEUX
- reading of the catalogues of commands (catalogues of elements which were recopied on database during first execution are not read)

The operands are to be used only to divert various files on a different logical unit than default numbers or to adjust certain parameters of files.

The simple concepts of python (of the type variable) created during a preceding execution are preserved in a file associated with base JEVEUX (pick.1). During the execution of procedure POURSUITE these concepts are regenerated and can thus be used under the variable name they were created

Note:

Python objects of the type classified and type function are not saved in the pick.1

Operands

The operand PAR_LOT and keywords LANG and DEBUG are identical to those of procedure DEBUT [U4.11.01].

Keywords BASE and HDF which are different in procedure POURSUITE are discussed here.

Keyword BASE

BASE =

This keyword redefines the values of the parameters of the random access file associated with "database" if the user does not want to use the default parameters.

POURSUITE cannot modify characteristics of database GLOBALE.

Default values of the parameters associated with databases.

```
VOLATILE
      NMAX_ENRE       62914
      LONG_ENRE       100        Kmots
      LONG_REPE       the 2000
```

word is worth 8 bytes in 64 bits under LINUX64, TRU64 an IRIX64. It is worth 4 bytes in 32 bits under SOLARIS, HP-UX, WINDOWS-NT and LINUX.

Procedure `POURSUITE` will allocate a file of random access for base '`VOLATILE`' a record of `100 Kmots` (the K is worth 1024) to more than 62914 records on a LINUX64 machine by Default.

Note:

The real size of the file is dynamic and it depends on the volume of information to store. But this size is limited by the operating conditions and parameters present among the values characterising the platform.

On a 64 bit platform, the maximum size is fixed at 48 Go. This value can be modified while passing an argument on the command line behind the key word "`- max_base size`" where size is an actual value measured out of Mo.

On a 32 bit platform, the maximum size is fixed at 2,047 Go (2,147,483,647), but the code manages several files to go beyond this limit when the parameter "`- max_base`" is passed in the argument.

For the base GLOBALE, which can be saved and re-used in computation data, the maximum size in "POURSUITE" is preserved such as it is if the parameter "`- max_base`" is not used, but perhaps redefined with the need for this manner.

Operand FICHIER

◆FICHIER =

Symbolic name of the base considered.

Only the parameter of database "VOLATILE" can be redefined.

Operands LONG_ENRE / NMAX_ENRE / LONG_REPE

Definitions of the parameters

/ | LONG_ENRE = lenr
lenr is the length of the records in Kmots of the files of random access used.

Note:

The memory manager JEVEUX uses this parameter to determine two types of objects, the large object which will be cut out in as many records as necessary, and the small objects which will be accumulated in a buffer of the size of a record before being discharged.

| NMAX_ENRE = nenr
nenr is the number of records per defect, if this value is not modified by the use of the keyword "- max_base" then this value is given starting from LONG_ENER and of an operating parameter on the reference platform of LINUX64 it is fixed at 12 Go (51,539,607,552 bytes)

Note:

Two operands LONG_ENER and NMAX_ENER must be used with precaution. Bad use can lead to a critical stop of the program by saturation of the files of random access. Coherence between the maximum size of the file and the value resulting from the product of two parameters LONG_ENRE and NMAX_ENRE is checked at the beginning of the execution.

| LONG_REPE = lrep
lrep is the initial length of the directory (maximum number of addressable objects by JEVEUX). It is managed dynamically by the memory manager which extends the size of the directory and all the associated system objects as needs.

Keyword CODE

CODE =
This keyword makes it possible to activate the generation of the file .code. held with the benchmarks, this keyword is a version simplified from the keyword code of DEBUT.

Keyword FORMAT_HDF

FORMAT_HDF = 'OUI',
Allows reading database "GLOBALE" again which is saved in the file with format HDF (look at FIN [U4.11.02]). The base is then rebuilt starting from object JEVEUX stored in the file, this

file can be built on a different platform (operating system, platform 32 or 64 bits). The characters of the original base are read again in the file and the base is rebuilt identically (the length of records is preserved).

The file with format HDF associated with GLOBALE database is named bhdf.1 in the directory of execution of the code.

Example of use

The standard use of this procedure is:

POURSUITE();

The tests yyy100a and yyy100b illustrate the use of RESERVE_CPU. The tests forma04b, ssnv156a, ssnv166b and yyy108 illustrate the use of MEMOIRE.