Operator DEBUG

1 Goal

Allows to activate or disable the functionalities of debugging between the various commands. Certain key words are accessible in the commands `debut` and `POURSUITE`.

It is about a functionality intended for the developers.

2 Syntax

```
DEBUG  (

      ◊   SDVERI=/          "YES",
                            /"NON",

      ◊   JXVERI=/          "YES",
                            /"NON",

      ◊   JEVEUX=/          "YES",
                            /"NON",

      ◊   IMPR_MACRO=/      "YES",
                            /"NON"
     )
```

3 Operands

3.1.1 Operand SDVERI

Attention, this functionality can cause a considerable overcost during the execution.
This key word starts the checking of data structures produced by the operators. It is used in the frame as of procedures of development of the code in the tests of non regression.

3.1.2 Operand JXVERI

Makes it possible to control the integrity of the segments of the memory between two executions of consecutive commands. This functionality relates to only the mode of management in memory known as static, when a memory zone allocated at the beginning of execution is managed by JEVEUX by chaining the various objects allocated during the execution. In way of managing in memory known as dynamic the various objects are allocated directly by the system and sequence does not exist.

3.1.3 Operand JEVEUX

Makes it possible to activate the operating mode in debug of the manager of JEVEUX memory: unloadings on disc not differed and assignment from the segments values to an indefinite value [D6.02.01].

3.1.4 Key word IMPR_MACRO

Authorizes or not the displays produced by the macros in the file of message. The reading of the files of message can be painful when it contains the totality of the echoes of the subcommands generated by macro itself. By default, only the echo of the commands explicitly called by the user in his command set will appear.