Command SEXECUTE

PURPOSE Execute a list of SATAN commands automatically

PARAMETERS

FILE Name of a file containing the commands to be executed.

&1 … &9 Character strings that will substitute the parameters **&1.**, **&2.**, … **&9.**, respectively, wherever they appear in the command list.

/PATH(p) Path of the command file to be executed.

/BEGIN(I) If specified, execution begins with the statement “STOP I” or “LABEL I” of the command list; “I” denotes a characters string.

/NOLIST If specified, the given commands are not listed at the terminal, but executed.

REMARKS The command SEXECUTE is also invoked by dropping an SCOM file from the WINDOWS EXPLORER in the SATAN dialog window by the mouse.

A common pre-selection of the path or a list of paths for SCOM files to be executed may be performed by the command
SET / EXECPATH(…). SATAN will search in this list for the file name indicated in the SEXECUTE command in the sequence they appeared in the EXECPATH parameter of the SET command..

A command list may call another one, and so on, up to any level..

A SATAN command list may contain up to 9 local symbolic parameters (e.g. used as parameters of commands), specified as **&1.**, **&2.**, …, **&9.** , respectively. They are substituted by the character strings given in the corresponding positions of the SEXECUTE command. The user is prompted if a symbolic parameter is encountered having no value.

Furthermore, any command substring may be substituted by a global parameter preceded by an ampersand “**&**”, and followed by a period “**.**” As delimiter. Global parameters are created by the commands IPAR and IPOPER, or by the Macro PARDCL in the user-supplied analysis procedure. Global parameters and local parameters may be nested up to any level, e g. **&CALIB(&1.).** , **&A&1..**The values of the parameters are inserted in several passages, starting from the innermost level.

All SATAN commands may be used within a command list. Additionally, the following special statements are supported:

LABEL string	defines a label
STOP or **STOP string**	stops and evtl. defines a label
RETURN	returns one level up (to the calling procedure or command prompt)
END	returns to the command prompt

SELECT(string)

Define the beginning of a select group. The string eventually given as argument is evaluated and saved. The statements listed below control the flow of execution within a select group; commands between these statements are called “units”.

WHEN(string)
Statement
Statement
…

Specify the unit to be executed if the string given as argument matches the string of the corresponding SELECT statement.

OTHERWISE
Statement
Statement
…

Specify the unit to be executed when every test of the preceding WHEN statements fails.

ENDSELECT

Terminate the select group

An alternative syntax of the SELECT group:

SELECT
WHEN(Boolean expression)

The Boolean expression may contain any arithmetic expression. The result zero is equivalent to “false”, any other result is equal to “true”. Boolean operators “&”, “|”, “^”, “=” (like in PL/I) are supported.

Statement
Statement
…
OTHERWISE
Statement
Statement
…
ENDSELECT

DO i **=** $\mathbf{n_1}$ **TO** $\mathbf{n_2}$ **BY** $\mathbf{n_3}$

Define an iterative loop. Start value, last value and step are specified. If omitted, the default step value is one.

ENDDO

IF Boolean expression THEN statement

Define an IF clause, consisting of one statement.

IF Boolean expression THEN DO

Define an IF clause, consisting of a number of statements

Statement
Statement
…

ELSE

Statement

Statement

…

ENDIF

LEAVE

Leave an iterative DO loop. May be used after an IF statement inside an iterative DO loop to leave the loop in a certain case.

Comments are denoted by /* …*/ as in PL/I, however, they can be nested. Alternatively, one line is commented if it begins with a “*”sing.

EXAMPLE

```
EX  GET  E1  E2
```

The command list in the file GET.SCOM is executed. Assume the following content:

```
* Fetch analyzers, evaluate the sum or difference and dump the result.
AFETCH &1.
AFETCH &2.
STOP A
SELECT(&3.)
        WHEN(SUM)
                AOPER ASUM = &1. + &2.
                ADUMP ASUM
        WHEN(DIFF)
               AOPER ADIFF = &1. - &2.
                ADUMP ADIFF
        OTHERWISE
                PUT UNDEFINED OPTION
ENDSELECT
```

The character string &1. is replaced by “E1” and &2. by “E2”; parameter &3. is prompted. The analyzers “E1” and “E2 are fetched from the dumpfiles E1.DMP and E2.DMP, respectively. According to the value of &3., the sum or the difference is evaluated and dumped into the file ASUM.DMP or ADIFF.DMP. IF &3 has a value different from “SUM” and “DIFF”, an error message is given. Entering
EX GET E1 E2 SUM / BEGIN(A), the execution begins at the statement “STOP A”.

```
EX  ZBAR  DISP
```

The command list in file ZBAR.SCOM is executed: It defines a combination of two figures from different GRAF files on one display.

```
gr ZH-BAR
G&1 / YREL(1.05) FORM(0.7) NOTITLE AXF(1.8) NOXN  +
   SCA(0.35) SY(5) YMIN(51) YMAX(57) PAL(MONO)
GR NH-BAR
Gover / FORM(0.7) NOTITLE AXF(1.8) SCREL(1) SYM(5) +
   YMIN(78.5) YMAX(84.5)
```