GÄLLER FR O M:	DISTRIBUERAD TILL:	TEKNISKA PÄRMEN TS	
1981-05-28	TS		
ERSÄTTER UTGÅVA AV:	UTFÄRDAD AV:	FLIK, GRUPP, LÖP-NR:	SID AV:
---	SAB	6 3 1.	1 13

META FORTRAN - USER'S MANUAL

MET

1/29/76 FREDRIK STADLER

81-08-12 TOMMY LUNDH
81-10-12 WILLY SVENNINGSSON
VERSION V06

BRIEF DESCRIPTION OF META FORTRAN PRECOMPILER

1 INTRODUCTION

MET FORTRAN PRECOMPILER CONVERTS SOURCE-FILES WRITTEN IN THE META-FORTRAN LANGUAGE TO STANDARD-FORTRAN SOURCE-FILES AND PRODUCES A LIST-FILE.

THE MAIN FEATURES OF MET ARE:

1. STRUCTURED PROGRAMMING FACILITIES
2. COPY TEXT IMPLEMENTATION
3. COMMENT FACILITIES

WITH THIS TOOL YOU CAN:

MAKE CODE WITHOUT ERRORS

MAKE PROGRAMS EASY TO READ.

WRITE IN A STANDARD WAY

SEE THE EFFECTS OF CHANGES

```
MCR>MET
MET><FILE1>,<FILE2>=<FILE3>

(FOR /PD, PD-MET, /VX AND VX-MET SEE 'GENERATION')

FILE1:  OUTPUT FORTRAN FILE
        DEFAULT:                    SY:.FTN (PD-MET)
                                    SY:.FOR (VX-MET)
        SWITCHES:                   /PD OR /VX

FILE2:  OUTPUT LIST FILE
        DEFAULT:                    SY:.LST (PD-MET)
                                    SY:.LIS (VX-MET)
        SWITCHES:
                /VX                 SEE 'GENERATION'
                /PD                 SEE 'GENERATION'
                /SP  ON:            LIST FILE WILL BE SPOOLED
                    OFF:            LIST FILE NOT SPOOLED
                DEFAULT: ON

                /CY  ON:            LIST FILE WILL CONTAIN TOTAL
                                    CONTENTS OF ANY COPY TEXT FILE
                    OFF:            LIST FILE WILL CONTAIN ONLY
                                    FIRST RECORD OF COPY TEXT FILES
                DEFAULT:  OFF

                /TM                 TAB MARKS (DOTS) WILL BE
                                    PRINTED ON LIST FILE TO
                                    SIMPLIFY READING OF HEAVILY
                                    NESTED MODULES
                DEFAULT:  ON

FILE3:  INPUT META FILE
        DEFAULT:                    SY:.MET

/ID     (NO FILES OR OTHER SWITCHES ALLOWED)
        WILL CAUSE THE VERSION NUMBER OF MET
        TO BE LOGGED
```

3 STRUCTURED PROGRAMMING

THE FOLLOWING LOGICAL BLOCKS ARE SUPPORTED:

´IF´ - ´ELSIF´ - .. - ´ELSE´ - ´ENDIF´

´CASE´ - ´WHEN´ - .. - ´OTHER´ - ´ENDCASE´

´POSIT´ - ´ADMIT´ - ´ENDPOSIT´

´WHILE´ - ´ENDDO´

´REPEAT´ - ´UNTIL´

´DOLOOP´ - ´ENDLOOP´

´BREAK´

´PERFORM´

´SECTION´ - ´ENDSECT´

3.1 ´IF´

```
´IF´(LOG1)                EXECUTE CODE1 IF LOG1 IS TRUE
   <CODE1>
´ELSIF´ (LOG2)            EXECUTE CODE2 IF LOG2 IS TRUE
   <CODE2>
.
.
´ELSE´                    EXECUTE CODE9 IF LOG1 AND LOG2 ...
   <CODE9>                ARE FALSE
´ENDIF´
```

NOTE: ´ELSIF´ AND/OR ´ELSE´ MAY BE OMITTED

3.2 ´CASE´

```
´CASE´ VAR
´WHEN´ VALUE1            EXECUTE CODE1 IF VALUE1 = VAR
   <CODE1>
´WHEN´ VALUE2            EXECUTE CODE2 IF VALUE2 = VAR
   <CODE2>
.
.
´OTHER´                  EXECUTE CODE9  OTHERWISE
   <CODE9>
´ENDCSE´
```

3.3 ´POSIT´

```
´POSIT´                  EXECUTE CODE1 AS LONG AS A CERTAIN
                         ASSUMPTION IS TRUE
   <CODE1>
    ´IF´ (LOG2)          CHECK THE ASSUMPTION
       ´QUIT´            INTERRUPT AND PROCEED AT THE ADMIT PART
    ´ENDIF´
   <CODE2>
´ADMIT´                  EXECUTE CODE9 IF THE ASSUMPTION WAS WRONG
   <CODE9>
´ENDPOSIT´
```

´QUIT´ INTERRUPTS EXECUTION OF THE CODE IN THE ´POSIT´-PART AND EXECUTION PROCEEDES AT THE FIRST STATEMENT IN THE ´ADMIT´-PART

IF NO ´QUIT´ IS ENCOUNTERED IN THE ´POSIT´-PART THEN THE ´ADMIT´-PART IS NOT EXECUTED

3.4 ´WHILE´

```
´WHILE´(LOG)            EXECUTE CODE WHILE LOG IS TRUE
   <CODE>
´ENDDO´
```

3.5 ´REPEAT´

```
´REPEAT´                EXECUTE CODE AND REPEAT THIS
   <CODE>               UNTIL LOG IS TRUE.
´UNTIL´(LOG)            I.E. AT LEAST ONE TIME !
```

3.6 ´DOLOOP´

```
´DOLOOP´ I=START,STOP,STEP
   <CODE>
´ENDLOOP´
```

3.7 ´BREAK´

´BREAK´ INSIDE A ´WHILE´-, ´REPEAT´- OR ´DOLOOP´-BLOCK CAUSES AN IMMEDIATE JUMP TO THE FIRST STAEMENT AFTER THE ´ENDDO´, ´UNTIL´ OR ´ENDLOOP´ RESPECTIVLEY

3.8 ´PERFORM´

```
E.G.    ´PERFORM´       FUNCTION-25
        .
        .
        .
        ´SECTION´       FUNCTION-25
            <CODE>
        ´ENDSECT´
```

NOTES: NAME OF SECTION CAN BE UP TO 12. CHARACTERS LONG.
ALL CHARACTERS ARE ALLOWED EXCEPT SPACE.
SECTION SHOULD BE PUT AT THE END OF THE MODULE.
THEY CAN NOT BE ENTERED IN-LINE (A ´GOTO´ PAST
IT WILL OCCOUR).

NOTES: LOG IS ANY LEGAL FORTRAN EXPRESSION.
CODE IS ANY NUMBER OF STATEMENTS
INCLUDING OTHER LOGICAL BLOCKS.

4 COPY TEXTS

COPY TEXTS MAY BE INSERTED ANYWHERE IN THE META PROGRAM

SYNTAX:

C&<DEL><FILE4>Ä<DEL>&=XÅ

C IN COL 1
& IN COL 2
<DEL> IS ANY COMBINATION OF TABS OR SPACES (AT LEAST 1)
<FILE4> IS ANY LEGAL DATA SET SPECIFIER
DEFAULT DEVICE: SY:

DEFAULT UIC : ÄN,310Å
WHERE ´N´ IS THE UIC GROUP CODE OF MET AT RUN TIME. E.G.: ASSUME UIC HAS BEEN SET TO Ä130,120Å USING THE ´HEL´ FUNCTION. DEFAULT UIC FOR COPY TEXT FILES WILL THEN BE Ä130,310Å.

DEFAULT EXT.: .CPY

IF &=X IS SPECIFIED ALL &´S IN THE THE COPY TEXT FILE WILL BE REPLACED BY X.
DEFAULT IS REFPLACEMENT BY SPACE.
META EXPRESSION IN COPY TEXT FILES ARE NOT PRE COMPILED.

COMPLETE NAME INCLUDING DEVICE, UIC, AND VERSION NUMBER OF COPY TEXT FILE AS WELL AS CREATION DATE AND TIME WILL BE OUTPUT ON LIST FILE.

5 ´ASCII´

SYNTAX:

```
BYTE IB(3)

´ASCII´ IB/HEJ/
´ASCII´ SLASH ////
```

WILL RESULT IN THE FOLLOWING FORTRAN CODE:

```
DATA IB/´H´,´E´,´J´/
DATA SLASH /´/´/
```

TAB CHARACTERS ARE LEGAL CARACTERS ANYWHERE
(CONVERTED TO SPACES FOR FORTRAN FILE)

6 ´TABVEC´

THIS IS A "MACRO" USED TO SET UP A 2-16 WORDS INTEGER ARRAY FOR A VARIABLE TAB CALL TO THE MODIFIED TT-HANDLER

(VTF).

EXAMPLE OF SYNTAX:

´TABVEC´ IA /T123B53,3A5,25I6,34F2,60S5/

IA IS THE ARRAY NAME.

T - LINES CAN BE TERMINATED WITH FUNCTION KEY 1,2 OR 3

B - EMPTY LINES CAN BE TERMINATED WITH FUNCTION KEY 5 OR 3

A TAB HOLD IS DESCRIBED. FOR EXAMPLE 3A5 WHICH MEANS POSITION 3, A FORMAT, FIELD SIZE 5.

REF. SEPARATE VTF DOCUMENTATION FOR ALL FORMATS ETC

THE ARRAY HAS TO BE DECLARED IN THE NORMAL WAY.

NOTE: THE SIZE OF THE ARRAY IS THE NUMBER OF TAB HOLDS PLUS ONE EXCEPT WHEN USING 16 TABHOLDS IN WHICH CASE THE ARRAY SIZE SHOULD BE 16.

.7 OTHER FACILITIES

1. TABULALATED LIST FILE

2. COMMENTS ADDED TO ANY LINE

3. <CODE><DEL><COMMENT> <DEL> IS ANY COMBINATION OF TABS OR SPACES (AT LEAST ONE) FOLLOWED BY <
   K=K+1 < COMMENT

4. NEW PAGE FORCED BY 'PAGE'

5. STATEMENT NUMBERING AS IN FORTRAN

NOTE: LABELS 1-9999 CAN BE USED WITHOUT INTERFERING WITH META.

.8 GENERATION

EDIT METPRE.MAC TO FIT YOUR INSTALLATION. THE FOLLOWING OPTIONS MAY BE CHOSEN:

1. DVAX=0 - A MET SUITABLE FOR RSX STANDARDS IS GENERATED. FORTRAN FILE IS OF TYPE .FTN, LIST FILE IS OF TYPE .LST AND 'TCSENTRY' XXX IS TRANSLATED TO SUBROUTINE USER. THIS DEFAULT SET-UP CAN BE OVERRIDED TO VMS-TYPE BY /VX.

2. DVAX=1 - A MET SUITABLE FOR VMS STANDARDS IS GENERATED. FORTRAN FILE IS OF TYPE .FOR, LIST FILE IS OF TYPE .LIS AND 'TCSENTRY' XXX IS TRANSLATED TO SUBROUTINE XXX. THIS DEFAULT SET-UP CAN BE OVERRIDED TO RSX-TYPE BY /PD.

3. RSX11M=1. SHOULD BE DEFINED WHEN RUNNING ON RSX-11M OR VMS. OTHERWISE THE SYMBOL SHOULD NOT BE DEFINED

EDIT THE ABOVE FILE TO CHANGE THE FOLLOWING PARAMETERS:

1. PAGSIZ=48. PAPER SIZE (NUMBER OF LINES/PAGE)

2. MTAB=4 NUMBER OF SPACES PER 'TAB' IN THE MET-LIST

1.2 COMPILING (FOR 77)

Syntax:

KFO <source code file> <switch>/<switch2>/...<switchn>

source code file — Source code's file name (with or without extension). Default extension is MET.

switchn — switches for changing standard compile options.

- I4 - Default integer is a integer*4 type.
- TR - put "trace-back" information (line number mm.)
- DB - tag with support for FORTRAN 77 symbolic debugger.
- RO - divides the code into separate read-only, and read/write parts (facilitates RSX11M+ multiuser tasks.
- CK - Checks field limits and widths.
- NO - do not optimise object code
- NT - compile list without tab marks.
- CY - list copy text files
- DE - fortran compiles with debug statements.
- WR - show warnings from fortran compile.
- TI - show compile list on terminal.
- LP - show compile list on printer.
- LI - change fortran compile list attribute e.g. specify LI:3 (for description see FORTRAN 77 user's guide).
  Default is obtained /LI:0.
- NQ - show executed command line.
- RP - create report file with information with the outcome of the command (CMDRPT.TMP)

1.3 ASSEMBLY OF MACRO MODULE

Syntax:

KMA <source code file><switch1>/<switch2>/...<switchn><prefix file(s)>

source code file	Source code file name (with or without extension). Default extension is MAC.
switchn	Switches for changing standard assembly options. LP - output assembly list on printer. TI - output assembly list on terminal. NQ - list executed command line. LI - add desired macro list attribute e.g. LI:MEB:SRC (see attributes in MACRO-11 reference manual). NL - delete macro list attribute e.g. NL:MEB:SRC. RP - create report file containing information about the outcome of the command (CMDRPT.TMP).
prefix file(s)	One or more (max. 4) source code prefix files can be specified separated by (,) character. Default extension is MAC.

1.6 COMPILING AND LINKING OF FOR 77

Syntax:

KLF <source code file><switch1>/<switch2>/...<switchn>

Compiles and links fortran 77 module. The task created is placed on ZT:

source code file See command KFO

switchn See command KFO
LM - List map.
RE - set up the new task.
RP - Create report file containing information about the outcome of the command (CMDRPT.TMP)

1.7 COMPILING AND LINKING OF MACRO (MACRO11)

Syntax:

KLM <source code file><switch1>/<switch2>/...<switchn><prefix file(s)>

Assembles and links macro module. The task created is placed on ZT:

source code file	See command KMA
switchn	See command KMA LM - List map RE - Set up the new task RP - Create report file containing information about the outcome of the command (CMDRPT.TMP)
prefix file(s)	See command KMA

1.10 COMPILATION OF FORTRAN MODULE AND PICKING IN LIBRARY

Syntax:

KBF <source code file><library name><switch1>/.../...<switchn>

Compiles fortran 77 module and places object code in an object library.

source code file	See command KFO
library name	Name of desired library. Presupposes that the library file has extension OLB.
switchn	See also command KFO CL - create library if not already created. RP - create report file containing information about the outcome of the command (CMDRPT.TMP).

1.11 COMPILATION OF MACRO MODULE AND PLACING (MACRO11) IN LIBRARY

Syntax:

KBM <source code file><library name><switch1>/.../...<switchn>
<prefix file(s)>

Assembles macro module and places object code in an object library.

source code file	See command KMA
library name	Name of desired library. Presupposes that the library file has extension OLB.
switchn	See also command KMA CL - create library if not already created. RP - create report file containing information about the outcome of the command (CMDRPT.TMP)
prefix file(s)	See command KMA

CHAPTER 5

SUBROUTINE LIBRARY

1. RECOMMENDED GEN-LIB ROUTINES

1.1 Character handling routines

1.1.1 Right justification of ASCII string
1.1.2 Left justification of ASCII string
1.1.3 Convert to "uppercase" character
1.1.4 Increment ASCII String
1.1.5 Check if ASCII String is numeric
1.1.6 Change "A", "A" and "O" for sorting (SWEDISH)
1.1.7 Change back after CHACH1 (SWEDISH)
1.1.8 Fill ASCII String with specified character
1.1.9 Check ASCII String content

1.2 VT100 DISPLAY ROUTINES

1.2.1 Set VT100 to "Bold" mode
1.2.2 Change line on VT100 to double width
1.2.3 Position cursor on VT100 Terminal
1.2.4 Set Scrolling region on VT100
1.2.5 Change VT100 over to "Smooth Scroll"
1.2.6 Clear whole or parts of VT100 screen
1.2.7 Decide whether terminal is of VT100 type or not
1.2.8 Change serial VT100 attributes simultaneously
1.2.9 Change VT100 over to keypad mode
1.2.10 Change terminal over to full-duplex mode
1.2.11 General VT100 screen handling routine
1.2.12 Change VT100 over to reserved video
1.2.13 Change VT100 over to graphic mode
1.2.14 Revert VT100 from graphic mode

1.3 INPUT FROM TERMINAL

1.3.1 Input ASCII buffer
1.3.2 Analysis of input from terminal
1.3.3 Input and check of ASCII String
1.3.4 Input and check of integer *2 value
1.3.5 Input and check of integer *4 value
1.3.6 Input and check of real number
1.3.7 Input and check of real number in double precision
1.3.8 Input and check of Integerlist
1.3.9 "Attach" terminal and save any desired characters
1.3.10 Read characters saved by subroutine ATTEX
1.3.11 Input of individual characters from terminal

1.4 FILE MANAGEMENT ROUTINES

1.4.1 "Attach" unit
1.4.2 Check whether file exists
1.4.3 Find file name from logical unit
1.4.4 Find file version number
1.4.5 "Spool" file on optional printer (Version 1)
1.4.6 "Spool" file on optional printer (Version 2)
1.4.7 Delete file
1.4.8 Create block I/O datafile
1.4.9 Open block I/O datafile
1.4.10 Read block on block I/O datafile
1.4.11 Write block on block I/O datafile
1.4.12 Close block on I/O datafile

1.5 BIT HANDLING ROUTINES

1.5.1 Set or reset bit in word (Version 1)
1.5.2 Test whether bit in word is set or not (Version 1)
1.5.3 Set or reset bit in word (Version 2)
1.5.4 Test whether bit in word is set or not (Version 2)
1.5.5 Read bit value on bit map
1.5.6 Set/reset bit on bit map
1.5.7 Convert word to 16 bytes ASCII (0 or 1)
1.5.8 Convert word to 16-word vector

1.6 CONCERSION ROUTINES

1.6.1 Convert ASCII String to integer *2 number
1.6.2 Convert ASCII String to octal format
1.6.3 Convert ASCII String to integer *4 format
1.6.4 Convert ASCII String to real number
1.6.5 Convert ASCII String to real number in double precision
1.6.6 Convert real number to ASCII
1.6.7 Convert real number (in double precision) to ASCII
1.6.8 Convert ASCII String to integer list
1.6.9 Convert decimal integer to octal integer

1.7 TIME ROUTINES

1.7.1 Get system time in binary format (Format 1)
1.7.2 System date in packed format (Format 4)
1.7.3 System time in packed format (Format 4)
1.7.4 System date in ASCII format (Format 3)
1.7.5 System time in ASCII format (Format 3)
1.7.6 Check date in ASCII format (Format 3)
1.7.7 Check time in ASCII format (Format 3)
1.7.8 Pack date in ASCII format to integer *2 number
.7.9 Pack date in integer format to integer *2 number
1.7.10 Unpack packed date to ASCII format (4-3)
1.7.11 Unpack packed time to ASCII format (4-3)

1.8 OTHER ROUTINES

2.1 NON-RECOMMENDED SUBROUTINES
STRING HANDLING ROUTINES

2.1.1 Locate specified Byte String
2.1.2 Right justify ASCII String
2.1.3 Right justify String (Version 2)
2.1.4 left justify ASCII String
2.1.5 Conversion of characters in ASCII String to upper case character
2.1.6 Increment ASCII String
2.1.7 Merge several ASCII Strings to one with zero fill
2.1.8 Divide ASCII string into substrings
2.1.9 Compare two ASCII String
2.1.10 Compare two ASCII fields
2.1.11 Check whether ASCII String is numeric
2.1.12 Check if string only includes numeric " "
2.1.13 Check if string only includes numeric " 0 "
2.1.14 Fill ASCII string with blanks
2.1.15 Fills ASCII string with ASCII Zeros
2.1.16 Fill Byte string with specified character
2.1.17 Change "A", "A" and "0" for sorting
2.1.18 Change back after ASCCH1
2.1.19 Edits numeric ASCII string
2.1.20
2.1.21
2.1.22
2.1.23
2.1.24 Locate first and last writable character in buffer

2.2 TIME ROUTINES

2.2.1 System time in binary format (format 2)
2.2.2 Pack date in integer format to integer *2 (2-4)
2.2.3 Pack time in integer format to integer *number (2-4)
2.2.4 Unpack packed date to integer format (4-2)
2.2.5 Unpack packed time to integer format (4-2)

FORTRAN 77 general subroutine library comprises of two parts:

1 recommended subroutines - subroutines tested which should be used if possible. Described in chapter 1.

2 subroutines which remain for reasons of compatibility. If the corresponding function is present under 1 it should be used. Described in chapter 2.

Fortran 77 subroutine library is called GENF77.OLB and is in UFD [40,1]. It is linked to the user program with:

```
MAIN=MAIN,...,ZG:[40,1]GENff/LB,....
```

Parameters within <...>can be omitted.

.... means the variable number of parameters.
The variables are of integer*2 type if no others are specified.

1 RECOMMENDED GEN-LIB ROUTINES

Description of the recommended general subroutines.

1.1 CHARACTER HANDLING ROUTINES.

Below are the routines intended for the handling of variables in FORTRAN-77 character format.
If a routine appears to be missing, first see what can be done with FORTRAN-77 standard statements and standard routines.
A part string of character variables is specified for all the calls stated below.

1.1.1 RIGHT JUSTIFICATION OF ASCII STRING

Subroutine (MACRO) which right justifies ascii string.

CALL

CHARJU (CHAR,LENGTH)

PARAMETERS

CHAR string which should be right justified.

LENGTH length of ascii string.

1.1.2 LEFT JUSTIFICATION OF ASCII STRING

Subroutine (MACRO) which left justifies ascii string.

CALL

CHARJU (CHAR,LENGTH)

PARAMETERS

CHAR string which should be left justified.

LENGTH length of ascii string which should be operated.

1.1.3 CONVERT TO "UPPER CASE" CHARACTER.

Subroutine (MACRO) converts lower case characters to upper case.

CALL

CHAUPP (CHAR,LENGTH)

PARAMETERS

CHAR Ascii string.

LENGTH Length of ascii string.

1.1.4 INCREMENT ASCII STRING.

This subroutine (MACRO) numerically increments an ASCII string. If overflow occurs (i.e. all characters before call = 9) it restarts at 0 (all characters in the string = 0).

CALL

CHAINC (CHAR,LENGTH)

PARAMETERS

CHAR ASCII string (numeric)

LENGTH length of ASCII string.

1.1.5 CHECK IF ASCII STRING IS NUMERIC

Logical function (MACRO)

CALL

CHANUM(CHAR,LENGTH)

PARAMETERS

CHAR ASCII string

LENGTH length of ASCII string.

CHANUM function value:

.TRUE. If CHAR only contains numeric characters.

.FALSE. If CHAR contains non-numeric characters.

1.1.6 CHANGE " A ", " A " and " O " FOR SORTING (SWEDISH).

The subroutine (MACRO) changes " A ", " A " and " O " to characters which can be sorted correctly.

CALL

CHACH1(CHAR,LENGTH)

PARAMETERS

CHAR ASCII string.

LENGTH length of ASCII string.

1.1.7 CHANGE BACK AFTER CHACH1 (SWEDISH)

The subroutine (MACRO) changes " A ", " A " and " O " after CHACH1 consists of ASCII zeros.

CALL

CHACH2(CHAR,LENGTH)

PARAMETERS

CHAR ASCII string

LENGTH length of ASCII string.

1.1.8 FILL ASCII STRINGS WITH SPECIFIED CHARACTER.

The subroutine (MACRO) fills the character in FILL into the ASCII string CHAR.

CALL

CHAFIL(CHAR,LENGTH,FILL)

PARAMETERS

CHAR ASCII string.

LENGTH Length of ASCII string.

FILL The character to fill out with. Only one character. The first character is used if several are specified.

1.1.9 CHECK ASCII STRING CONTENT

Logical function (MACRO) which checks if the ASCII string CHAR comprises only of the character VERI.

CALL

CHAVER(CHAR,LENGTH,VERI)

PARAMETERS

CHAR ASCII string.

LENGTH Length of ASCII string.

VERI The character to be checked. Only one character. The first character is used if several are specified.

CHAVER function value :

.TRUE. The string CHAR consists solely of the character VERI.

.FALSE. The string CHAR does not consist solely of the character VERI.

1.2 VT100 DISPLAY ROUINTES

Subroutines to operate VT100 and VT100 compatible terminal attributes from the FORTRAN programs.

1.2.1 SET VT100 TO "BOLD" MODE

Engages/disengages " bold " mode.

CALL

```
BOLD(UNIT,ONOFF)
```

PARAMETERS

UNIT	Logical unit
ONOFF	Control parameter
	1 = engage "bold" mode
	0 = disengage "bold" mode

1.2.2 CHANGE LINE ON VT100 TO DOUBLE WIDTH

Changes current line to text in double width.

CALL

```
DOUBLE(UNIT,ONOFF)
```

PARAMETERS

UNIT	Logical unit
ONOFF	Control parameter
	0 = engage double width
	1 = disengage double width

1.2.3 POSITION CURSOR ON VT100 TERMINAL

Sets cursor to line LINE and column COL.

CALL

```
CURSOR(UNIT,LINE,COL)
```

PARAMETERS

:	UNIT	Logical unit
	COL	Column number (1-24)
	LINE	Line number (1-80)

1.2.4 SET SCROLLING REGION ON VT100

Sets desired scrolling region on VT100.

CALL

```
SCROLL(UNIT,TOP,BOT)
```

PARAMETERS

UNIT	Logical unit
TOP	Top line in scrolling region
BOT	Bottom line in scrolling region

1.2.5 CHANGE VT100 OVER TO "SMOOTH SCROLL"

Engages/disengages " smooth scroll "

CALL

```
SMOOTH(UNIT,ONOFF)
```

PARAMETERS

UNIT	Logical unit
ONOFF	Control parameter
	0 = disengage "smooth scroll"
	1 = engage "smooth scroll"

1.2.6 CLEAR WHOLE OR PARTS OF VT100 SCREEN

Subroutine (FORTRAN).

CALL

CLEAR(UNIT,TYPE<,LINE,COL,NCHAR>)

PARAMETERS

UNIT : Logical unit

TYPE : Defines different types of clearing

1 - Clear whole screen and set cursor to home position

2 - Clear screen from line LINE downwards, and set cursor to first position on line LINE

3 - Clear line LINE from column COL to end of line and set cursor to position LINE,COL

4 - Clear NCHAR character positions from postion LINE,COL and set cursor to position LINE,COL

LINE Line number

COL Column number

NCHAR number of characters

1.2.7 DECIDE WHETHER TERMINAL IS OF VT100 TYPE OR NOT

Logical function (macro)

CALL

VT100(UNIT)

PARAMETERS

UNIT : Logical unit

VT100 function value :
.TRUE. Terminal is of VT100 type.

.FALSE. Terminal is not of VT100 type.

1.2.8 CHANGE SEVERAL VT100 ATTRIBUTES SIMULTANEOUSLY

Subroutine (fortran)

CALL

ATTRIB(UNIT,NPARM,BOLD <,UNDLIN,FLASH,REVERS >)

PARAMETERS

UNIT : Logical unit

NPARM : Number of attributes specified

BOLD : Bold mode on/off (1/0)

UNDLIN : Underscore on/off (1/0)

FLASH : Flash on/off (1/0)

REVERS : Reversed video on/off (1/0)

1.2.9 CHANGE VT100 OVER TO KEYPAD MODE

Subroutine (fortran)

CALL

KEYPAD(UNIT,ONOFF)

PARAMETERS

UNIT : Logical unit

ONOFF : Control parameter

0 = disengage keypad mode

1 = engage keypad mode

1.2.10 CHANGE TERMINAL OVER TO FULL-DUPLEX MODE

Subroutine (macro)

CALL

FULDPX(UNIT,ONOFF)

PARAMETERS

UNIT : Logical unit

ONOFF : Control Parameter

0 = half-duplex mode

1 = full-duplex mode

1.2.11 GENERAL VT100 SCREEN HANDLING ROUTINE

Subroutine (macro) used by other VT100 video routines.

CALL

VIDEO(UNIT,TYPE, < ,P1,P2,P3>)

PARAMETERS

UNIT : Logical unit

TYPE : Function code

Pn : Additional parameters (depend on TYPE)

1.2.12 CHANGE VT100 OVER TO REVERSED VIDEO

Subroutine (fortran)

CALL

REVERS(UNIT,ONOFF)

PARAMETERS

UNIT : Logical unit

ONOFF : Control parameter

0 = disengage reversed video

1 = engage reversed video

1.2.13 CHANGE VT100 OVER TO GRAPHIC MODE

Subroutine (MACRO)

CALL

: GRAFON(NUNIT)

PARAMETERS

: UNIT logical unit

1.2.14 REVERT VT100 FROM GRAPHIC MODE

Subroutine (MACRO)

CALL

: GRAFOF(UNIT)

PARAMETERS

: UNIT logical unit

1.3 INPUT FROM TERMINAL

Routines for input from TTY terminal.

1.3.1 INPUT ASCII BUFFER

Subroutine (fortran) which reads in an ASCII buffer.

CALL

```
INPUT(UNIT,TIMOUT,BUFFER,LENGTH,IDS)
```

PARAMETERS

UNIT : Logical unit

TIMOUT : Number of seconds to timeout

BUFFER : ASCII buffer (80 Bytes long)

LENGTH : Number of characters input

IDS : Return code

0 = input IK
1 = illegal input
2 = timeout
3 = control z specified
4 = <ESC> input

1.3.2 ANALYSIS OF INPUT FROM TERMINAL

Subroutine (fortran) which analyses input buffer after subroutine INPUT has been used.

CALL

```
ANSWER(BUFFER,LENGTH,IDS)
```

PARAMETERS

BUFFER ASCII buffer

LENGTH Number of characters input

IDS Return code

0 = normal input
1 = character Y input
2 = character N input
3 = only <CR> (length = 0)
4 = character X input

1.3.3 INPUT AND CHECK OF ASCII STRING

Subroutine (fortran) which inputs ASCII string from terminal. Checks max. and min. limits.

CALL

```
TTGETI(UNIT,TIMOUT,BUFFER,LENGTH,MIN,MAX,IDS,ERROR)
```

PARAMETERS

UNIT : Logical Unit

TIMOUT : Timeout time in number of seconds

BUFFER : ASCII buffer

LENGTH : Number of characters input

MIN : Minimum number of characters allowed

MAX : Maximum number of characters allowed

IDS : See IDS under subroutine ANSWER (chap.5.2)

ERROR : Return code

0 = input OK
1 = illegal input
2 = timeout
3 = control z specified

1.3.4 INPUT AND CHECK OF INTEGER *2 VALUE

Subroutine (fortran) which inputs integer*2 value from terminal. Checks max. and min. limits.

NB: Task built with option FMTBUF > =10

CALL

```
TTGETI(UNIT,TIMOUT,VALUE,MIN,MAX,IDS,ERROR)
```

PARAMETER

UNIT : Logical unit

TIMOUT : Timeout time in seconds

VALUE : Integer *2 variable. VALUE does not change if an input error is encountered.

MIN : Minimum permitted value

MAX : Maximum permitted value

IDS : See IDS under subroutine ANSWER (chap.5.2)

ERROR : Return code

0 = input OK
1 = illegal input
2 = timeout
3 = control z specified

1.3.5 INPUT AND CHECK OF INTEGER *4 VALUE

Subroutine (fortran) which inputs integer*4 from terminal. Checks max. and min. limits.

CALL

```
TTGETJ(UNIT, TIMOUT,VALUE,MIN,MAX,ID,ERROR)
```

PARAMETER

UNIT : Logical unit

TIMOUT : Timeout time in seconds

VALUE : Input integer *4 variable. VALUE does not change if an input error is encountered.

MIN : Minimum permitted value (integer *4)

MAX : Maximum permitted vlaue (integer *4)

IDS : See IDS under subroutine ANSWER (Chap.5.2.)

ERROR : Return code

0 = input OK
1 = illegal input
2 = timeout
3 = control z specified

1.3.6 INPUT AND CHECK OF REAL NUMBER

Subroutine (fortran) which inputs real value from terminal. Checks max. and min. limits.

NB: Task-build with option FMTBUF > = 10

CALL

TTGETR(UNIT,TIMOUT,NINT,NDEC,VALUE,MIN,MAX,IDS,ERR)

PARAMETERS

UNIT : Logical unit

TIMOUT : Timeout time in seconds

NINT : Max. number of integer positions

NDEC : Max. number of decimal postions

VALUE : VALUE does not change if an input

MIN : Minimum permitted value (real number)

MAX : Maximum permitted value (real number)

IDS : See IDS under subroutine ANSWER (chap.5.2)

ERR : Return code

0 = input OK
1 = illegal input
2 = timeout
3 = control z specified

1.3.7 INPUT AND CHECK OF REAL NUMBER IN DOUBLE PRECISION

Subroutine (fortran) which inputs real value from terminal. Checks max. and min. limits.

CALL

: TTGETD(UNIT,TIMOUT,NINT,NDEC,VALUE,MIN,MAX,IDS,ERR)

PARAMETERS

UNIT : Logical Unit

TIMOUT : Timeout time in seconds

NINT : Max. number of integer positions

NDEC : Max. number of decimal postions

VALUE : Input real number in double precision. VALUE does not change if an input error is encountered.

MIN : Minimum permitted value (real No.,double prec)

MAX : Maximum permitted value (real No.,double prec)

IDS : See IDS under subroutine ANSWER (chap.5.2)

ERR : Return code

0 = input OK
1 = illegal input
2 = timeout
3 = control z specified

1.3.8 INPUT AND CHECK OF INTEGER LIST

The subroutine (FORTRAN) reads an integer list from unit UNIT. The elements in the list are seperated by comma, blank or tab character. Number of input integers is obtained in NVAL. The operator can specify a whole sequence by seperating two integers by a minus sign (e.g. 1-8). Use subroutine DOLOOP to find the correct loop index.

CALL

```
TTGETL(UNIT,TIMOUT,VALUE,NVAL, IDS,ERROR)
```

PARAMETERS

UNIT : Logical unit

TIMOUT : Timeout time in seconds

VALUE : Integer *2 field with input integer (40 words).VALUE can not be changed if incorrect input is encountered.

NVAL : Number of input integers

IDS : See IDS under subroutine ANSWER (chap.5.2)

ERROR : Return code

0 = input IK
1 = illegal input
2 = timeout
3 = control z specified

1.3.9 'ATTACH' TERMINAL AND SAVE ANY DESIRED CHARACTERS

Integer function (macro) attaches terminal and saves any characters input in temporary buffer (max 3). Use subroutine COMINP (chap.5.9) to decide whether any character has been input.

CALL

```
ATTEX()
```

PARAMETERS

ATTEX Function value:

If subroutine is used as function, 0 is obtained if directive error, <0 I/O error or 1 successful call.

1.3.10 READ CHARACTERS SAVED BY SUBROUTINE ATTEX

Logical function (macro)
Subroutine ATTEX must be called before COMINP can be used.

CALL

: COMINP(BUFF)

PARAMETERS

: BUFF : ascii buffer containing any characters input
: Logical function which assumes the value true if any character has been input since the last time COMINP was called.

1.3.11 INPUT OF INDIVIDUAL CHARACERS FROM TERMINAL

Integer function (macro) reads individual characters from terminal. Input characters are not echoed.

CALL

ICHAR(UNIT,FUNC)

PARMETERS

UNIT : Logical unit

ICHAR : Function value

FUNC : Function code:

If FUNC = 0 the function assumes input character as value. Otherwise FUNC assumes the following values (if VT100 is set to keypad mode)

FUNC	KEY	FUNC	KEY	FUNC	KEY
1	PF1	2	PF2	3	PF3
4	PF4	5	Keypad 7	6	Keypad 8
7	Keypad 9	8	Keypad -	9	Keypad 4
10	Keypad 5	11	Keypad 6	12	Keypad ,
13	Keypad 1	14	Keypad 2	15	Keypad 3
16	Keypad Ent	17	Keypad 0	18	Keypad .
19	Cursor up	20	Cursor down	21	<-
22	->	23	DELETE	24	<RETURN>
25	Tab key				

1.4 FILE MANAGEMENT ROUTINES

1.4.1 "ATTACH" UNIT

Subroutine (macro) attempts to "attach" logical unit. Allows typehead on terminal and prevents more than one program using the same terminal at one time.

CALL

```
ATTACH(UNIT,IDS)
```

PARAMETERS

UNIT : Logical unit

IDS : Return code:

0 = successful call
1 = failed call

1.4.2 CHECK WHETHER FILE EXISTS

Subroutine (macro) examines whether file assigned to UNIT exists.

CALL

```
FIND(UNIT,IDS)
```

PARAMETERS

UNIT : Logical unit

IDS : Return code

0 if file exists, otherwise 1.

1.4.3 FIND FILE NAME FROM LOGICAL UNIT

Subroutine (macro) finds which file is "assigned" to a specified logical unit.

CALL

GETFNB(UNIT,FILNAM)

PARAMETERS

UNIT : Logical unit

FILNAM : 24-Byte ASCII string containing file name

Finds file name and places in FILENAM in format:

DDnn:xxxxxxxxx.xxx;vvvvv

1.4.4 FIND FILE VERSION NUMBER

Subroutine (macro) finds version number of already opened file.

CALL

```
GETVSN(UNIT,VERS)
```

PARAMETERS

UNIT : Logical unit

VERS : Version number

1.4.5 "SPOOL" FILE ON OPTIONAL PRINTER (ALTERNATIVE 1)

Subroutine (fortran) spool file on printer. Printer must have spool queue "assigned". File to be spooled must be on default UIC of calling task.

CALL

```
SPOOL(UNIT,LPDEV,FLAG)
```

PARAMETERS

UNIT : Logical unit

LPDEV : Printer name in format TTnn: (ASCII)

FLAG : Delete flag

FLAG indicates whether file is to be deleted after spool

0 = do not delete
1 = delete

1.4.6 SPOOL FILE ON OPTIONAL PRINTER (ALTERNATIVE 2)

Subroutine (fortran)spools file on printer. Printer must have spool queue "assigned".

CALL

```
PRINT(LPDEV,FLAG,NAME,SIZE)
```

PARAMETERS

LPDEV : Printer name in format xxxxx (ASCII)

FLAG : Delete flag. Flag indicates whether file is to be deleted after spool.

0 = do not delete
1 = delete

NAME : Name of file to be spooled (ASCII)

SIZE : Number of bytes in NAME

1.4.7 DELETE FILE

Subroutine (macro) deletes files open on logical unit UNIT.

CALL

```
DELETE(UNIT,IDS)
```

PARAMETERS

UNIT : Open file's logical unit

IDS Return code:

0 = successful call
1 = unsuccessful call

1.4.8 CREATE BLOCK I/O DATA FILE

Subroutine (MACRO) creates and resets block I/O data file.

CALL

: CRBIOF(FNAM,FNAMZ,RECNO,RECSZ,IDS)

PARAMETERS

: FNAM : File name(in)

FNAMZ : Number characters in file name (in)

RECNO : Number records in the file (in)

RECSZ : Record size in bytes (in)

IDS : Return code:

>0 successful call
otherwise unsuccesssful call.

1.4.9 OPEN BLOCK I/O DATA FILE

Subroutine (MACRO) opens block I/O data file.

CALL

:BLKOPN(FDB,LUN,FNAM,FNAMZ,RACC,IDS)

PARAMETERS

: FDB : 50 words internal workspace (out).(Must not be affected by the user program).

LUN : Logical unit (in)

FNAM : Name of file to open (in)

FNAMZ: Number of characters in file name (in)

RACC : Access code:

RO read only
RW both read and write
IDS return code:
>0 successful call
otherwise unsuccessful call.

1.4.10 READ BLOCK ON BLOCK I/O DATA FILE

Subroutine (MACRO) reads block from block I/O data file.

CALL

:GETBLK(FDB,BLKNR,BUFFA,BUFFZ,IDS)

PARAMETERS

: FDB : 50 words internal workspace. Output data from BLKOPN(in)

BLKNR : Desired block number (in)

BUFFA : Block buffers address (out)

BUFFZ : Block buffers size in numbers of blocks (in)

IDS : Return code:

>0 successful call otherwise unsuccessful.

1.4.11 WRITE BLOCK ON BLOCK I/O DATA FILE

Subroutine (MACRO) writes block on block I/O data file.

CALL

:PUTBLK(FDB,BLKNR,BUFFA,IDS)

PARAMETERS

:FDB 50 words internal workspace. Output data from BLKOPN (in).

BLKNR : Desired block number (in)

BUFFA : Block buffers address (in)

BUFFZ : Block buffers size in number of blocks (in)

IDS : Return code:

>0 successful call otherwise unsuccessful.

1.4.12 CLOSE BLOCK IN DATA FILE TO DO

Subroutine (MACRO) closes block I/O data file opened by BLKOPN.

CALL

:BLKCLO(FDB,IDS)

PARAMETERS

: FDB 50 words internal workspace. Output data from BLKOPN (in).

IDS Return code :

>0 successful call otherwise unsuccessful.

1.5 BIT HANDLING ROUTINES

Routines for handling of bits in words and bit maps, also conversion routines for translating bitmaps.

1.5.1 SET OR RESET BIT IN WORD

Subroutine (macro) sets or resets bit depending on value of VALUE.

CALL

```
SETBIT(WORD,BIT,VALUE)
```

PARAMETERS

WORD : Word containing bit to be set or reset

BIT : Bit number in word (1-16). BIT = 1 means change of least significant bit etc.

VALUE : Bit value (0-1)

1.5.2 TEST WHETHER BIT IN WORD IS SET OR NOT (VERSION 1)

Logical function (macro) tests if a bit in a word is set.

CALL

```
TESTBIT(WORD,BIT,VALUE)
```

PARAMETERS

WORD : Word containing bit to be tested

BIT : Bit number in word (1-16). Test bit assumes value true if desired bit has value VALUE, otherwise false. BIT = 1 means test on least insignificant bit etc.

VALUE : Bit value (0 or 1)

1.5.5 READ BIT VALUE IN BIT MAP

Integer function (macro) gives the bit value of a bit in a field.

CALL

GETBIT(BITMAP,BITNUM)

PARAMETERS

BITMAP : Field containing bit map

BITNUM : Bit number (1-32767). BITNUM = 1 means reading of least significant bit in first word of BITMAP. BITNUM = 32 means reading of bit 1 in second word of bitmap etc.

GETBIT function value : Assumes value 1 if bit is set, otherwise 0.

1.5.6 SET/RESET BIT IN BIT MAP

Subroutine (macro) sets or resets bit in bitmap

CALL

PUTBIT(BITMAP,BITNUM,VALUE)

PARAMETERS

BITMAP : Field containing bit map

BITNUM : Bit number (1-32767) BITNUM = 1 means least significant bit in first word of bitmap is set. BITNUM = 32 means that bit in the second word of bitmap is set etc.

VALUE : Bit value (0 or 1)

1.5.7 CONVERT WORD TO 16 BYTES ASCII

Subroutine (macro) converts word to 16 byte bitmap with 0 or 1.

CALL

BITASC(WORD,BUFF)

PARAMETERS

WORD : Word (integer *2) to be converted

BUFF : 16 Byte output buffer

1.5.8 CONVERT WORD TO 16-WORD VECTOR

Subroutine (FORTRAN) converts word to 16-word vector containing zeros or ones.

CALL

: WRITE(PWORD,UBITMAP)

PARAMETERS

: PWORD : Word to be converted

: UBITMAP : Vector (16 words) containing the converted word.

3.1 DATABASE HANDLER - INTERFACE

The database handler is called from a user program using sub-routine calls. The sub-routines are accessible once the user program is linked with an interface module DBSIFC.OBJ. The interface module uses standard RSX11M methods for calls.

A copy text file, DBSDEF.CPY (fortran 4) or DBSPAR.CPY (fortran 77) can be copied onto the user program in order to link symbolical names with record and variable numbers.

IMPORTANT CONCEPTS AND DEFINITIONS

Ids = return code. Positive when call successful

Label = address specified in database call and to which exit can be made if the data value using is unsuccessful. Allocated a value using assign statement in call program. If label is omitted then control returns, irrespective of the value of the return code, to the line following the sub-routine in the call program.

Father = record in superior register

Brother = record in the same register as the record in question

Son = record in inferior register

Father Pointer = pointer from record in inferior register (son) to record in superior register (father)

Son Pointer = pointer from record in superior register (father) to record in inferior register (son)

Brother Pointer = pointer from one record to another record in the same register (brother to brother)

RAF register = register addressed using a key. A pointer to a RAF-register consists of 4 bytes

DAF register = register addressed using record number pointers to DAF-registers are the same as record numbers and consist of two bytes.

DATABASE CALL: RETURN CODES

Positive return code for successful calls. Negative return code for unsuccessful calls.

Valid return codes:

1 Successfully completed call.

-1 Incorrect parameter list; incorrect database description or system pool full.

-2 Not part of implemented function of database handler or database handler not yet installed.

-6 Incorrect data buffer in user program.

-26 Incorrect SFM file no. e.g. SFM file not installed.

-81 Database handler not initiated or forcibly stopped.

-90 Incorrect register number

-91 Incorrect variable number

-92 Incorrect structure

-93 Incorrect usage of variable as pointer

ADDRESSING OF RECORD AND/OR VARIABLE

Record = register no., record identifier

Record identifier = key (, pointer, (pointer value)...)

Register No = desired register register no.

Key = key value if a RAF-register or record no if a DAF register

Pointer = variable no. of the variable acting as a pointer to another record. May be omitted.

Pointer value = variable where the selected record's record no. or key value is put. May be omitted.

Variable = record, variable no., variable value (, old value)

Record = as above

Variable no. = variable no. of required variable

Variable value = contains value of the variable

Old value = If specified, testing of old value occurs i.e. when reading takes place the value is also entered into the old value when writing takes pace in the database it is checked to see if there has been any change in the old value. Should this have been the case then a special return code is given.

N.B.: applies only to DBGET and DBPUT (DBEPUT) calls (see below).

In the calls described below, <...> indicates that parameters may be omitted. All other parameters must be included.

3.1.1 READING VARIABLE(S) IN RECORD

3.1.1.1 DBGET - READ INDIVIDUAL VARIABLE

Call DBGET (<Label>, Ids, Variable)

Parameters: as above

Special return codes:

-46 Record does not exist
-66 Variable format error
-83 Attempt to read via pointer with pointer value 0 or -1

3.1.1.2 DBGTM - READ GROUP OF VARIABLES

Call DBGTM (<Label>, Ids, Register no., Key<, Variable no, value ...>)

Parameters: as above. A maximum of 4 groups of variable no., values can be specified

Special return codes:

-46 Record does not exist
-66 Variable format error

3.1.2 UPDATING VARIABLE(S) IN RECORD

3.1.2.1 DBPUT - UPDATING INDIVIDUAL VARIABLE

Call DBPUT (<Label>, Ids, Variable)

Call DBEPUT (<Label>, Ids, Variable)

If the transaction log option was selected (see above), then the following are also available:

CALL DBBPUT (<Label>, Ids, Variable)

CALL DBXPUT (<Label>, Ids, Variable)

CALL DBCPUT (<Label>, Ids, Variable)

DBPUT does not immediately update the disc file; it only updates the variable in a primary memory buffer.

DBEPUT ensures automatic writing on disc immediately after updating. However, if the last operation in the record was a linking or unlinking operation, then the use of DBEPUT is not necessary as all such operations are automatically followed by writing on disc. This does not apply if the transaction log option was selected. updating of the disc is internally controlled by the database handler.

Parameters: See above

Special return codes:

```
  2 Change in old value (see above)
-46 Record does not exist
-66 Variable format error
-83 Attempt to read via pointer with pointer value 0 or -1
```

3.1.2.2 DBPTM - UPDATING GROUPS OF VARIABLES

CALL DBPTM (<Label>, Ids, Register no., Key<, Variable No., Value..>)

CALL DBEPTM (<Label>, Ids, Register no., Key<, Variable no., Value..>)

If the transaction log option was selected (see above) then the following are also available:

CALL DBBPTM (<Label>, Ids, Register no., Key<, Variable no., value..>)

CALL DBBPTM (<Label>, Ids, Register no, Key<, variable no., Value..>)

CALL DBCPTM (<Label>, Ids, Register no., Key<, Variable no., Value..>)

DBPTM does not immediately update the disc file; it only updates the variable in a primary memory buffer.

DBEPTUM ensures automatic writing aon disc immediately after updating. However, If the last operation in the record was a linking or unlinking operation, then the use of DBEPTM is not necessary as all such operations are automatically followed by writing a disc.

This does not apply if the transaction log option was selected. Updating of the disc is internally controlled by the database handler.

Param: See above Max 4 groups of variable no. value can be specified

Variable nos., values can be specified.

Special return codes:

-46 Record does not exist
-66 Variable format error

3.1.3 CREATING AND DELETING RECORDS IN/FROM THE DATABASE

3.1.3.1 DBNEW CREATE NEW RECORD

CALL DBNEW (<Label>, Ids, Record<, Variable no., Value..>)

CALL DBENEW (<Label>, Ids, Record<, Variable no, Value..>)

If the transaction log option was selected (see above), then the following are also available:

CALL DBBNEW (<Label>, Ids, Record<, Variable no, Value..>)

CALL DBXNEW (<Label>, Ids, Record<, Variable no, Value..>)

CALL DBCNEW (<Label>, Ids, Record<, Variable no, Value..>)

Creates records for both RAF and DAF registers. If the register is a RAF-register then the key value is specified in the record identifier. A DAF-register is allocated the first free record record no. in the record identifier. DBENEW is used when the record is to be immediately written on disc i.e. when the DBNEW-call is not followed by an updating or link-operation on the same record.

This does not apply if the transaction log option was selected. Updating of the disc is internally controlled by the database handler.

Parameter: See above. A maximum of 4 groups of variable nos., values can be specified.

Special return codes:

-24 RAF-register full
-46 Record already exists
-66 Key format error or variable format error
-78 Daf-register full

3.1.3.2 DBDEL - DELETE RECORD

CALL DBDEL (<Label>, Ids, Record)

If the transaction log option was selected (see above), then the following are also available:

CALL DBBDEL (<Label>, Ids, Record)
CALL DBXDEL (<Label>, Ids, Record)
CALL DBCDEL (<Label>, Ids, Record)

Deletes records from the database for both RAF and DAF registers. DBDEL ensures automatic writing of the record on the disc. DBDEL checks that there are no pointer in the record not set to 0.

Special return codes:

2 Record not deleted as pointer updated in the record
-46 Record does not exist
-66 Key format error (applies only to RAF-registers)

3.1.4 LINKING AND UNLINKING RECORD

3.1.4.1 DBLNK - LINKING OF RECORD (WITH SORTING)

CALL DBLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

CALL DBFLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

If the transaction log option was selected (see above), that the following are also available:

CALL DBBLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

CALL DBXLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

CALL DBCLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

AND

CALL DBBFLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

CALL DBFLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

CALL DBCFLNK (<Label>, Ids, Son record, Father record 1<, Father record 2..>)

Links son record to one or more linked lists. if the required linked list has been sorted, then DBFLNK commences its search for the correct position from the beginning of the list. the search initiated by DBLNK, on the other hand, begins from the end of the list. Should the list be unsorted, then the record is entered into the last position in the list.

Parameters: See above

Special return codes:

-46 Record does not exist
-66 Key format error (applies only to RAF-registers)
-78 Linking error (e.g. record already linked)

3.1.4.3 DBUNL - UNLINKING OF RECORD

CALL DBUNL (<Label>, Ids, Son record, Father record 1 < Father record 2..>)

If the transaction log option was selected (as above), then then following are also available:

CALL DBBUNL (<Label>, Ids, Son record, Father record 1 < Father record 2..>)

CALL DBXUNL (<Label>, Ids, Son record, Father record 1 < Father record 2..>)

CALL DBCUNL (<Label>, Ids, Son record, Father record 1 < Father record 2..>)

Unlink son record from one or more linked lists.

Parameters: see above

Special return codes:

-46 record does not exist
-66 Key format error (applies only to RAF-registers)
-78 Unlinking error (e.g. record already unlinked)

3.1.4.4 DBUND - SIMULTANEOUS UNLINKING AND DELETION OF RECORD

CALL DBUND (<Label>,Ids, Son record, Father record 1<, Father record 2..>)

If the transaction log option was selected (see above), then the following are also available:

CALL DBBUND (<Label>,Ids, Son record, Father record 1<, Father record 2..>)

CALL DBXUND (<Label>,Ids, Son record, Father record 1<, Father record 2..>)

CALL DBCUND (<Label>,Ids, Son record, Father record 1<, Father record 2..>)

Unlinks son record from on or more linked lists. After the unlinking has taken place, then the son records is deleted from the database (the son record must be a DAF-type).

Parameters: See above

Special return codes:

2 Record not unlinked and deleted as the record is linked to another linked list
-46 Record does not exist
-66 Key format error (applies only to RAF-registers).
-78 Unlinking error (e.g. record already unlinked)

3.1.6 OTHER DATABASE CALLS

3.1.6.1 DBCHK - CHECK THAT THE DATABASE VARIABLE HAS THE CURRENT FORMAT

DBCHK (<Label>, Ids, Register no, variable no, Value)

DBCHK is the logical function which makes the value true if no format error is detected in the variable variable no.

Parameters: See above

Special return codes:

3 Format errors

3.1.6.2 DBFND - CHECK THAT THE RECORD EXISTS

DBFND is the logical function which makes the value true if the record has already been created.

parameters: See above

Special return codes:

2 Record does not exist
3 Format error on specified key (applies only to RAF-registers)

3.1.6.6 IERR - ROUTINE ERROR

IERR()

IERR is a function which is called if exit has taken place to the label given in the database call. IERR specifies the FORTRAN line containing the erroneous database call.

Parameters: Name Name of the fortran module error is in. Is specified as six ASCII character. Fortran module must be constructed with fortran-77 compiler and have trace-information included (the switch /TR at compiling).

CHAPTER 9

TERMINAL HANDLING ROUTINES

2. SUBROUTINES

2 SUB ROUTINES

2.1 General Functions

There are a number of general sub routines for writing in a simple manner dialogue routines which comply with the screen dialogue-standard 830111. The following functions are available:

Heading is written out with STRUBR.

For input, display and changes there are different routine depending on the type of variable to be input. STGETA, STGETI, STGETJ, STGETL, STGETN, STGETR, STGETS and STGETV deal with the most usual types and carry out normal checks before the input is acknowledged. If an input has to be made of a type which cannot be handled by these subroutines, you can write a new subroutine which manages this based on one of the existing subroutines.

Display of data is possible with the subroutines, but of course write-statements should be the most common manner

The same subroutines as used for input can be used for changes by suggesting a value and afterwards changing it.

STERRI is used for outputting error messages. This routine is also used internally by the input subroutines.

STEXIT is called for terminating screen dialogue routines. This subroutine is used by the input routines in the case of timeout or reply X or cntrlZ. Should also be called by the dialogue routine in the case of normal termination.

All text written out by these subroutines can be modified in order to obtain the desired layout. Use of the subroutines means a standardized layout of the dialogue routines for a system with a number of dialogue routines.

2.2 Subroutine Descriptions

The following subroutine descriptions are intended for a user who will use but also modify the routines. To describe these subroutines simply, function, input and output are given. The structure of the routines is simple. STRUBR, STERRI, STEXIT and STPRMT have a simple sequential structure. The input routines STGETA, STGETI, STGETJ, STGETL, STGETN, STGETR, STGETS and STGETV have the same basic structure (see fig. 2.1). STPRMT is an internal routine which is called by all input routines.

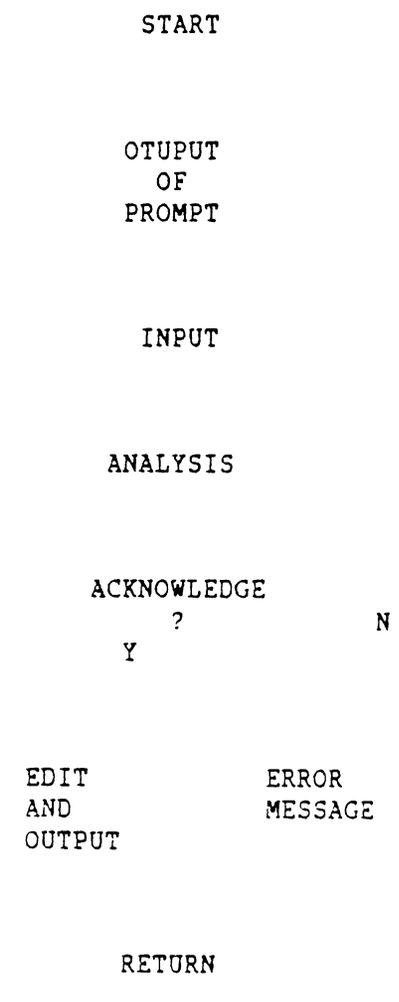


Figure 2.1 Flowchart for input routines

2.2.1 STRUBR Heading

Attaches the logical unit which output is to take place. Fortran input error message is blocked. Screen is cleared and then a heading line is written out. Heading consists of text, data time and name of executive program. Text comprises 1 -54 ASCII characters describing function of dialogue. Date is in format 15-MAR-83, time 13:07:25, and the name of the executive program comprises three letters. Heading line is written on line 1 beginning in column 1.

Call: CALL STRUBR (UNIT,TEXT,TEXTLENGTH)

Input:

UNIT	The logical unit on which heading is to be written out.
TEXT	Text string which begins the heading. 1-54 ASCII characters
TEXTLENGTH	Number of characters in TEXT

Output: None

Call:
ATTACH
CLEAR
ERROFF
DATE
TIME
GETTSK
R50ASC

Example: CALL STRUBR (1,"ARTICLE REGISTRATION" 20)
Produces:

ARTICLE REGISTRATION 15-MAR-83 12:23:30 ART

2.2.2 STERRI Error Messages

Error message line is written out on the screen. The error message comprises 1 - 65 ASCII characters. The error message is written on line 24 after the line has been cleared. The message always includes a BEL. This can be easily deleted if not desired.

Call: CALL STERRI (UNIT,TEXT,TEXTLENGTH)

Input:

UNIT	The logical unit on which error message is to be written out
TEXT	Text string of which error message consists. 1 - 65 ASCII characters
TEXTLENGTH	Number of characters in TEXT

Output: None

Calls: MBYTES
CLEAR

Example: CALL STERRI (1,"INPUT ERROR",11) produces message on 24:

*** INPUT ERROR ***

2.2.3 STEXIT Exit from Dialogue

Terminates execution of dialoge routine. This routine can be complemented by functions which you always wish to execute before a dialogue routine is terminated. The present function is to position cursor at line 24 column 1. It is important for all dialogue routines to use this subroutine.

Call: CALL STEXIT(UNIT)

Input:

UNIT	The logical unit on which the dialogue routine works

Output: None

Call: CURSOR
EXIT

2.2.4 STGETA Input of ASCII String

A prompt and possibly a suggested ASCII string variable are written out on a specified line. The pormpt and string variable are separated by a separator in a specified column. Then an ASCII string is input. The input string is anlysed and if accepted is returned together with a return code. If the string is not accpeted when analyzed, an error message is output and the input must be repeated. If an ASCII string is suggested at input, it is separated from the new input by a separator.

There are four ways of terminating the input routine besides an accepted input. These are by X followed by RETURN or only cntrlZ or by timeout when the routine is terminated if no reply is received after a given time. In these three cases an EXIT takes place, i.e. the dialogue routine calling the subroutine ends its execution. The fourth way is to enter <ESC> character. This deletes the default value.

Some of the functions of the input subroutine can be changed by means of a function bit map.

Call: CALL STGETA(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH,DEVALUE, VALUELENGTH,FUNCTION,TIMEOUT,VALUE,MIN,MAX,IDS)

Input:

UNIT	The logical unit on which input is to take place
LINE	Line on which input take place
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	ASCII string suggested at input
VALUELENGTH	Number of characters between separators, i.e. equal to or longer than MAX.

FUNCTION — Bit map which states changes of function in subroutine.

Bit	Function if bit is 1:
1	DEVALUE is written out and used. Input takes place after second separator instead of after first.
2	Only return or <ESC> are not permitted (as) replies
3	No input, only outout . VALUE is not affected.
4	TEXT is not written out
5	Separators are not written out.

Input: TIMEOUT — Time in seconds. If no character is entered within this time, input is aborted.

MIN — Minimum number of characters in VALUE

MAX — Maximum number of characters in VALUE

Output: VALUE — Input and accepted ASCII string if input has taken place

ID — Return code

0	VALUE is input
1	VALUE is not changed, only return is input

Call: ANSWER
CLEAR
CURSOR
INPUT
MBYTES
STERRI
STEXIT
STPRMT
TESTBIT

Example: CALL STGETA(1,3,17, "Name"4,Name 20,2,60, Name,1,20,IDS) produces on the screen:

Name: :HANDBRAKE CASING

2.2.5 STGETI Input of Integer * 2 Variable

Same function as STGETA. Input of a integer*2 variable instead of an ASCII string. After acceptance of the input, the input value is rewritten right-justified.

Call: CALL STGETI(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH, DEVALUE,VALUELENGTH,FUNCTION,TIMEOUT,VALUE,MIN, MAX,IDS)

Input:

UNIT	The logical unit on which input is to take place
LINE	Line on which input takes place
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	Integer*2 variable suggested at input
VALUELENGTH	Number of characters between separators
FUNCTION	Bit maps which states changes of function in subroutine.

Bit	Function if bit is 1:
1	DEVALUE is written out ans used. Input takes place after the second separator instead of the first.
2	Only return is not permitted as answer
3	No input, only output. VALUE is not affected
4	TEXT is not written out
5	Separators are not written out

TIMEOUT	Time in seconds. If no character is entered within this time, input is aborted.

	MIN	Minimum value of VALUE
	MAX	Maximum value of VALUE
Output:	VALUE	Input and accepted integer*2 variable if input has taken place.
	IDS	Return code
		0 VALUE is input 1 VALUE is not changed, only return is input
Calls:	ANSWER CLEAR CURSOR INCONV INPUT RJUST STERRI STEXIT STPRMT TESTBIT	

Example: CALL STGETI(1,2,17,"CRANE NUMBER",12,CRANE,
5,2,60,CRANE,1,5,IDS) produces on the screen:

Crane No: 3

after input

2.2.6 STGETJ

Input of integer*4 variable.
Same function as STGETI. Input of an integer*4 variable instead of an integer*2 variable.

Call: CALL STGETJ(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH, DEVALUE,VALUELENGTH,FUNCTION,TIMEOUT,VALUE, MIN,MAX,IDS

Input:

UNIT	The logical unit on which input is to take place
LINE	Line on which the input takes place
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	Integer*4 variable suggested at input
VALUE LENGTH	Number of chara. between separators NB: min 10 characters.
FUNCTION	Bit map which states changes of function in subroutine.

Bit	Function if bit is 1:
1	DEVALUE is written out and used. Input takes place after the second separator instead of the first.
2	Only return is not permitted as answer
3	No input, only output. VALUE is not affected.
4	TEXT is not written out
5	Separators are not written out

TIMEOUT	Time in seconds. If no character is entered within this time, input is aborted.

	MIN	Minimum value of VALUE
	MAX	Maximum value of VALUE
Output:	VALUE	Input and accepted integer*4 variable if input has taken place.
	IDS	Return code
	0	VALUE is input
	1	VALUE is not changes, only return is input

Call:
ADD14
ANSWER
ASBI10
BIAS4
CLEAR
CMP44
CURSOR
INPUT
NUMCHK
REDO1
RJUST
STERRI
STEXIT
STPRMT
TESTBIT

Example: CALL STEGETI(1,3,17,"Number",6,Number,11,2,60, NUMBER,MINIMUM,MAXIMUM,IDS) produces on the screen:

Number: 190000

after input

2.2.7 STGETL

Yes/no question as logial function.

A prompt and possbily a suggested reply (Y or N) are written out on a specified line. The prompt and reply are separated by a separator in a specified column. Then an ASCII character is input. The input character is analysed and if it is "Y", the function assumes the value true. If the input character is "N", the function assumes the value false. Only RETURN also means false. An accepted input is rewritten right-justified. if the reply is not approved when analysed, an error message is output and input must be repeated. If a reply is suggested at input, it is separated from the new input by a separator.

There are three ways of terminating the input routine besides an accepted input. These are by X followed by RETURN or only cntrlZ or by timeout when the routine is terminated if no reply is received after a specified time. In these three cases an EXIT takes place, i.e. the dialogue routine calling the subroutine ends its execution.

Some of the functions of the input subroutine can be changes by means of a function bit map.

Call: STGETL(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH,DEVALUE, VALUELENGTH,FUNCTION,TIMEOUT,IDS)

Input:

UNIT	The logical unit on which input is to take place.
LINE	Line on which input takes place.
COLUMN	Column in which the first sepatator is written
TEXT	Prompt which begines input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	Reply suggested at input (Y or N)
VALUELENGTH	Number of characters between separators

FUNCTION Bit map which states changes of function in subroutine

Bit	Function if bit is 1:
1	DEVALUE is written out and used. Input takes place after second separator instead of after first.
2	Only return is not permitted as reply
3	No input, only output
4	TEXT is not written out
5	Separators are not written out

Input: TIMEOUT Time in seconds. If no character is entered within this time, input is aborted.

Output: IDS Return code

0 "Y" or "N" is input
1 Only RETURN is input

Calls: ANSWER
CLEAR
CURSOR
INPUT
MBYTES
RIGHT
STERRI
STEXIT
STPRMT
TESTBIT

Example: IF (STGETL(1,3,17,"NEW PAGE",8,"Y",5,1,60,IDS))
produces on the screen:

New Page: Y/ and if only RETURN is input the function is true

2.2.8 STGETN Input of Numeric ASCII String

Same type of function as STGETA. Input of a numeric ASCII string. After approval of input, the input ASCII string is rewritten right-justified.

Call: CALL STGETN(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH, DEVALUE,VALUELENGTH,FUNCTION,TIMEOUT,VALUE,MIN, MAX,IDS)

Input:

UNIT	The logical unit on which input is to take place
LINE	Line on which input take place
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater then COLUMN
DEVALUE	Numeric ASCII string variable suggested at input
VALUELENGTH	Number of characters between separators, greater than or equal to MAX.
FUNCTION	Bit map which states changes of function in subroutine

Bit	Function if bit is 1:
1	DEVALUE is written out and used. Input takes place after the secons separator instead of after the first.
2	Only return or <ESC> are not permitted as a reply
3	No input, only output. VALUE is not affected
4	TEXT is not written out
5	Separators are not written out

	TIMEOUT	Time in seconds. If no character is entered with this time, input is aborted
	MIN	Minumum number of characters in ASCII string VALUE
	MAX	Maximum number of characters in ASCII string VALUE
Output:	VALUE	Input and accepted numeric ASCII string if input has taken place
	IDS	Return code
	0	VALUE is input
	1	VALUE is not changes, only return is input

Call:
ASCBLA
ANSWER
CLEAR
CURSOR
INPUT
MBYTES
NUMCHK
RIGHT
STERRI
STEXIT
STPRMT
TESTBIT

Example: CALL STGETN91,3,17,"ORDER NUMBER",12,ONO,5,2,60,
ONO,1,5,IDS) produces on the screen

Order No: 88

after input

2.2.9 STGETR

Input of real variable.

Same type of function as STGETI. Input of a real variable instead of an integer*2 variable..

Call: CALL STGETR(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH, DEVALVE,VALUELENGTH,DECLENGTH,FUNCTION,TIMEOUT,VALUE, MIN,MAX,IDS)

Input:

UNIT	The logical unit on which input is to take place.
LINE	Line on which input takes place
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 online LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	Real Variable suggested at input
DECLENGTH	Number of decimal characters
VALUELENGTH	Number of characters between separators
FUNCTION	Bit map which states changes if function in subroutine

Bit	Function If bit is 1:
1	DEVALUE is written out and used. Input takes place after the second separator instead of after the first.
2	Only return is not permitted as a reply
3	No input, only output. VALUE is not affected
4	TEXT is not written out
5	Separators are not written out

	TIMEOUT	Time in seconds. If no character is entered within this time, input is aborted.
	MIN	Minimum value of VALUE
	MAX	Maximum value of VALUE
Output:	VALUE	Input and accepted real variable if input has taken place
	IDS	Return code 0 VALUE is input 1 VALUE us not changes, only return is input

Calls: ANSWER
CLEAR
CONVR
CURSOR
INPUT
RCONV
STERRI
STEXIT
STPRMT
TESTBIT

Example: CALL STGETR(1,3,17,"PRICE",5,STPRICE,10,2,2,60
STPRICE,0,0,100,0,IDS) produces on the screen:

Price: 8.75

after input

2.2.10 STGETS

A prompt and possibly a suggested ASCII string variable are written out on a specified line. The prompt and string variable are separated by a separator in a specified column. Then an ASCII string is input. The input string is analyzed, and for it to be accepted it must agree with one of the permitted command strings. A string can be accepted even if it is not complete. However, it must be unambiguous and must not contain illegal characters . An index to the accepted srting is returned together with a return code. If the string is not accepted when analyzed, an error message is output and the input must be repeated.

If an ASCII string is suggested at input, it is separated from the new input by a separator.

There are three ways of terminating the input routine designs an accepted input. These are with X followed by RETURN or only cntrlZ or by timeout when the routine is terminated if no reply is received after a specified time. In these three cases an EXIT takes place, i.e. the dialogue routine calling the subroutine ends its execution

Some of the functions in the input subroutine can be changed by means of a function bit map.

Call: CALL STGETS(UNIT,LINE,COLUMN,TEXT,TEXTLENGTH, DEVALUE,VALUELENGTH,FUNCTION,TIMEOUT,VALUE, ALLVALUE,STRNGLENGTH,STRNGNUMBER,IDS)

Input:

UNIT	The logical unit on which input is to take place.
LINE	Line on which input takes place
COLUMN	Colmun in which the first separator is written
TEXT	Prompt which begins input. Text beings in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
DEVALUE	Command string suggested at input
VALUELENGTH	Number of characters between separators, i.e. equal to or longer than STRNGLENGTH

FUNCTION Bit map which states changes of function in subroutine

Bit Function If bit is 1:

1 DEVALUE is written out and used. Input takes place after the second separator instead of after the first.

2 Only return is not permitted as a reply.

3 No input, only output. VALUE is not affected.

4 TEXT is not written out

5 Separators are not written out

TIMEOUT Time in seconds. IF no character is entered within this time, input is aborted.

ALLVALUE ASCII string containing all permitted command strings in succession in long line.

STRNGLENGTH Number of characters in a command string

STRNGNUMBER Number of command strings in ALLVALUE

Output: VALUE Index to accept ASCII string in ALLVALUE if input has taken place.

IDS Return code
0 Command string is input
1 Only RETURN is input

Calls: ANSWER
CLEAR
CURSOR
INPUT
MBYTES
RIGHT
STERRI
STEXIT
STPRMT
TESTBIT

Example: CALL,STGETS(1,3,17,"LIST",4," ",6,
2,60,LISTTYPE,"STORE ORDER ARTICLE",5,
3,IDS) produces on the screen:

List: STORE

after input

2.2.11 STPRMT

Internal buffer handling

Prompt and separator are placed in output buffer.

Call: CALL STPRMT(COLUMN,TEXT,TEXTLENGTH,VALUELENGTH, BUFFER)

Input:	COLUMN	Column in which the first separator is written
	TEXT	Prompt which begins input. Text beings in column 1 on line LINE
	TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN
	VALUELENGTH	Number of characters between separators
Output:	BUFFER	Internal output buffer

Calls: ASCBLA
MBYTES

2.2.12 STGETO INPUT OF OCTAL INTEGER*2 VARIABLE

The same function as STGETA. Input an integer variable instead of an ascii string. After and input has been accepted the input value is written if it is right adjusted.

Call: CALL STGETO (UNIT, LINE, COLUMN, TEXT, TEXTLENGTH, DEVALUE, VALUELENGTH, FUNCTION, TIMEOUT, VALUE, MIN, MAX, IDS)

Input:

UNIT	The logical unit on which input is to take place.
LINE	Line on which input takes place.
COLUMN	Column in which the first separator is written.
TEXT	Prompt which begins input. Text begins in column 1 on line LINE.
TEXTLENGTH	Number of characters in TEXT. Must not be greater than COLUMN.
DEVALUE	Integer*2 variable suggested at input.
VALUE LENGTH	Number of characters between separators.
FUNCTION	Bit map which states changes of function in subroutine.

Bit	Function if bit is 1:
1	DEVALUE is written out and used. Input takes place after the second separator instead of the first.
2	Only return is not permitted as answer.
3	NO input, only input. VALUE is not affected.
4	TEXT is not written out.
5	Separators are not written out.

TIMEOUT	Time in seconds. If no character is entered within this time, input is aborted.
MIN	Minimum value of VALUE
MAX	Maximum value of VALUE

Output: VALUE — Input and accepted integer*2 variable if input has taken place.

IDS — Return code

0 Value is input.
1 Value is not changed, only return is input.

CALLS
ANSWER CLEAR CURSOR OCONV INPUT RJUST STERRI STEXIT STPRMT TESTBIT

Example: CALL STGETO (1,3,17,"Crane number", 12,CRANE, 5, 2, 60, CRANE, 1, 5, IDS) produces on screen: Crane number : 3 after input.

2.2.13 STGETB INPUT OF ASCII STRING

A prompt and possibly a suggested ascii string variable are written out on a specified line. The prompt and string variable are separated by a separator in a specified column. Then an ascii string is input. The input string is analysed and if accepted is returned with the return code. If the string is not accepted when analysed, an error message is output and the input must be repeated. After input the string is deleted and written again after it has been right adjusted.

If an ascii string is suggested at input, it is returned, it is separated from the new input by a separator.

There are four ways of terminating the input routine besides an accepted output. These are by X followed by RETURN or only cntrlZ or by timeout when the routine is terminated if no reply is received after a given time. In these three cases an EXIT takes place, i.e. the dialogue routine calling the subroutine ends its execution. The fourth way is to type in the <ESC> character. If necessary the default value is deleted. Some of the functions of the input subroutine are changed with a function bit map.

Call
CALL STGETB (UNIT, LINE, COLUMN, TEXT, TEXTLENGTH, DEVALUE, VALUELENGTH, FUNCTION, TIMEOUT, VALUE, MIN, MAX, IDS)

Input

UNIT	The logical unit on which input is to take place.
LINE	Line on which input takes place.
COLUMN	Column in which the first separator is written
TEXT	Prompt which begins input. Text begins in column 1 on line LINE
TEXTLENGTH	Number of characters in TEXT. Must not be greater than column.
DEVALUE	Ascii string suggested at input.
VALUELENGTH	Number of characters between separators i.e. as long or longer than MAX.

FUNCTION Bit map which states changes of function in subroutine.

Bit	function if bit is 1
1	DEVALUE is written out and used. Input takes place after second separator instead of after first.
2	only return or <ESC> are not permitted as reply.
3	no input, only output.
4	TEXT is not written out.
5	Separators are not written out.

TIMEOUT Time in seconds. If no character is entered within this time, input is aborted.

MIN Minimum number of characters in VALUE

MAX Maximum number of characters in VALUE

OUTPUT

VALUE Input and accepted ascii string if input takes place.

IDS Return code
0 VALUE is entered
1 VALUE is unchanged, only return is input.

CALL
ANSWER CLEAR CURSOR INPUT MBYTES STERRI STEXIT STPRMT TESTBIT

Example 1
CALL STGET (1, 3, 17, 'NAME, 4, NAME, 20, 2, 60, NAME, 1, 20, IDS) produces on the screen:
Name: HANDBRAKE CASING

2.2.14 STERRP - DISPLAY ERROR MESSAGE WITH ERROR CODE

The error message is displayed on the screen. The error message consists of 1 - 65 ascii characters and an integer. The error message is written on line 24 when the line has been cleared. A BEL is always included in the message. If this is not required, it can be deleted.

Call
CALL STERRP (UNIT, TEXT, TEXTLENGTH, ERRCODE)

Input
UNIT The logical unit which the error message should be written out on.
TEXT Text string which the error message comprises of. 1 - 65 ascii characters.
TEXTLENGTH number of characters in TEXT
ERRCODE Integer*2 which is shown.
Output
none
Call
MBYTES CLEAR
Example
CALL STERRI (1, 'DATABASE ERROR IDS '=,19,-46)
produces on line 24:
DATABASE ERROR IDS = -46

2.2.15 STEXST - CONCLUDE EXECUTING AND SPECIFY EXIT STATUS

Works as STEXIT but gives exit status as well.
Call
CALL STEXIT (UNIT, EXSTAT)
Input
UNIT The logical unit that the dialogue routine works on.
EXSTAT Exit status

Output
none
Call
SCROLL CURSOR EXST

2.2.16 STPRUT - DISPLAY TEXT STRING

A specified text is displayed on a current line.
Call
CALL STPRUT (UNIT, TEXT, TEXTLENGTH)
Input

UNIT	The logical unit that the error message will be displayed on.
TEXT	Text string which is displayed (1 - 80 characters)

Output
none

2.2.17 STUTSK - MESSAGE DISPLAY

Works like STERRI but asterisks and BEL are not output.
Call
CALL STUTSK (UNIT, TEXT, TEXTLENGTH)
Input

UNIT	The logical unit that the message will be displayed on.
TEXT	Text string that the message comprises of. 1 - 65 ascii characters.
TEXTLENGTH	number of characters in TEXT

Output
none
Call
MBYTES CLEAR
Example
CALL STUTSK(1, 'IN OPERATION', 12)
produces on line 24:
IN OPERATION

2.2.18 STGETY - INPUT OF DATE

The same function as STGETA. Inputing of date in standard ASCII format (yymmdd). After a successful input the input value is shown if right-adjusted.

Call : CALL STGERY (UNIT, LINE, COLUMN, TEXT, TEXTLENGTH, DEVALUE, VALUE LENGTH, FUNCTION, TIMEOUT, VALUE, MIN, MAX, IDS)

Input :

UNIT	The logical unit that inputs are made on
LINE	Line which input takes place on
COLUMN	Column on which the first separation character is shown on
TEXT	Prompt which prompts the input. The text starts at column 1 on the LINE line.
TEXT LENGTH	Number of characters in TEXT. Must not be greater than COLUMN.
DEVALUE	Integer*2 variable which includes default date in packed format.
VALUE LENGTH	Number of characters between separators.
FUNCTION	Bitmap which specifies subroutine function changes.

Bit	Function if the bit is 1
1	Devalue is shown and used. Input occurs after the second separator instead of after the first.
2	Only return is permitted as reply
3	Input is not done, only output Value is not affected
4	TEXT is not displayed
5	The separators are not shown

TIMEOUT	Time in seconds. If no character is entered within this time input is aborted
MIN	Minimum value of VALUE in packed format
MAX	Maximum value of VALUE packed format

Output

VALUE Input and accepted date in packed format (integer*2 vaiable) if input has occurred.

IDS Return code.
0 VALUE is input
1 VALUE is unchanged, only return is input

Calls

ANSWER CLEAR CURSOR ICONV INPUT RJUST STERRI
STEXIT STPRMT TESTBIT

Example 1

CALL STGETY (1, 3, 17, 'start date', 10, DATE, 5, 2, 60, DATE, 1, 5, IDS) is shown on the screen:

Start date 841012

after input.

2.2.19 STGETT - INPUT OF TIME

Same function as STGETA. Input of time in standard ASCII format (hhmmss). After an accepted input the inputed value is shown if right-adjusted.

Call

```
CALL STGETT (UNIT, LINE, COLUMN, TEXT, TEXTLENGTH, DEVALUE,
             VALUELENGTH, FUNCTION, TIMEOUT, VALUE, MIN, MAX,
             IDS)
```

Input	
UNIT	The logical unit which input takes place in.
LINE	Line which input takes place on
COLUMN	Column which the first separator character is written on.
TEXT	Prompt which prompts the input. The text starts at column 1 on the LINE line.
TEXT LENGTH	Number of characters in TEXT. Must not be greater than COLUMN.
DEVALUE	Integer*2 variable which includes default time in packed format.
VALUE LENGTH	Number of characters between the separators.
FUNCTION	Bitmap which specifies subroutine function changes
Bit	function if bit is 1
1	DEVALUE is shown and used. input after the second separator instead of the first.
2	only return is not permitted as reply
3	input is not done, only output VALUE is not affected
4	TEXT is not shown
5	the separators are not shown
TIMEOUT	Time in seconds. If no character is input within this time input is aborted.
MIN	Minimum value of VALUE in packed format
MAX	Maximium value of VALUE packed format

Output

VALUE	Input and accepted time in packed format (integer*2 variable) if input has occurred.
IDS	Return code 0 VALUE is entered 1 VALUE is unchanged, only return is input

Calls

ANSWER CLEAR CURSOR ICONV INPUT RJUST STERRI
STEXIT STPRMT TESTBIT

Example

CALL STGETT (1, 3, 17, 'starttime', 8, DATE, 5, 2, 60, DATE, 1, 5, IDS) is shown on the screen.

```
       start time :   101012
 after input.
```

1. CATCHALL

The digital command line interpreter MCR calls the Utility task if a specified command is not present in MCR's internal commands, or among the installed ...XXX tasks.

The Catchall task carries out the following steps (assuming command XXX is specified):

1 Is the command among the catchall's internal command chart. If not continue to 2.

2 Is command file XX.CMD in the specified user catchall during catchall UIC (specified during installation) if so execute it, otherwise go to 3.

3 Is the command file XXX.CMD in the system catchall UIC (specified during installation.) If so execute it, otherwise go to 4.

4 Test start task with RUN $XXX.

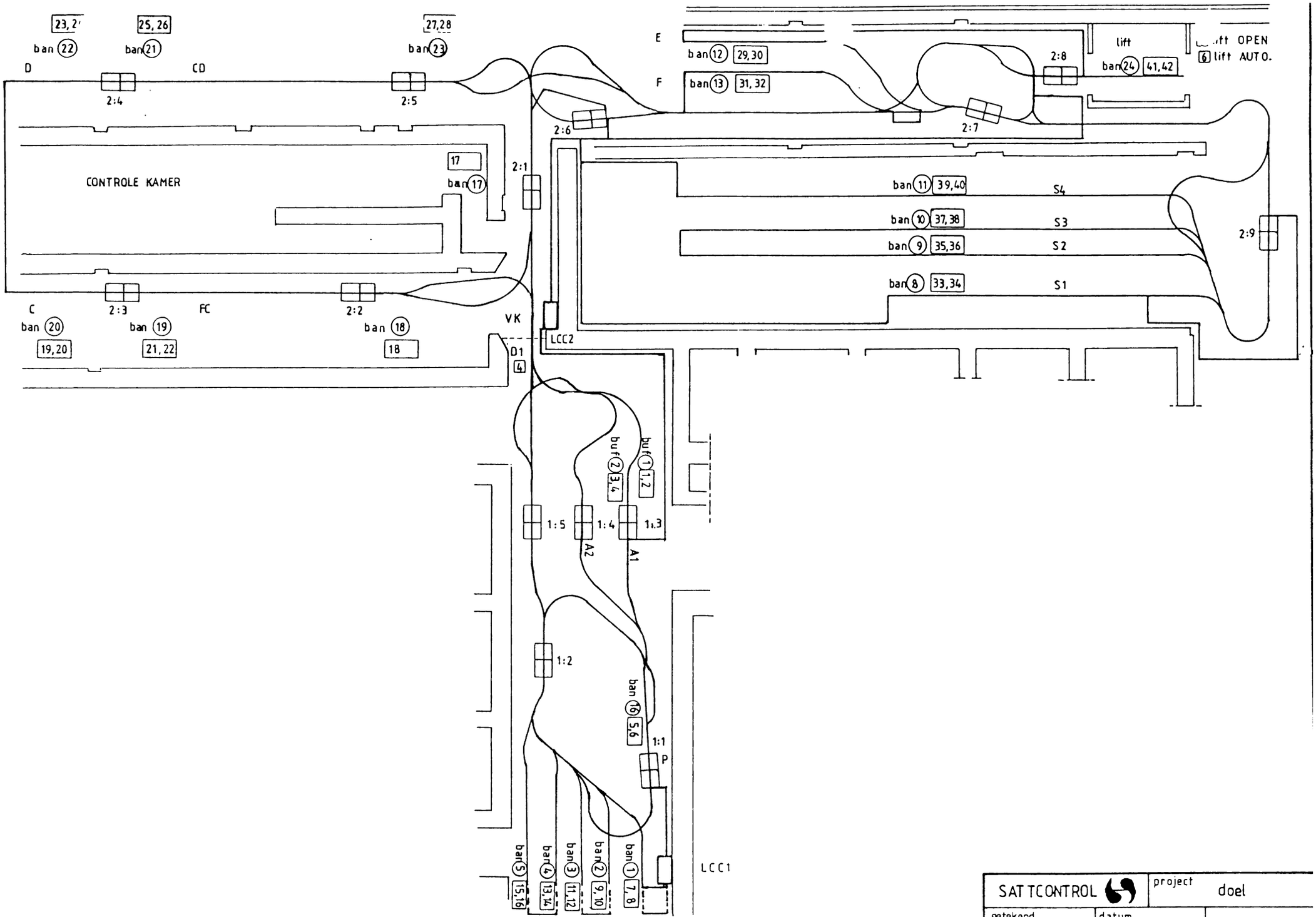
1.1 INTERNAL COMMAND

Parameters inside (...) can be ommited. When (...) is specified directly in the command the abbreviated/short form is used.

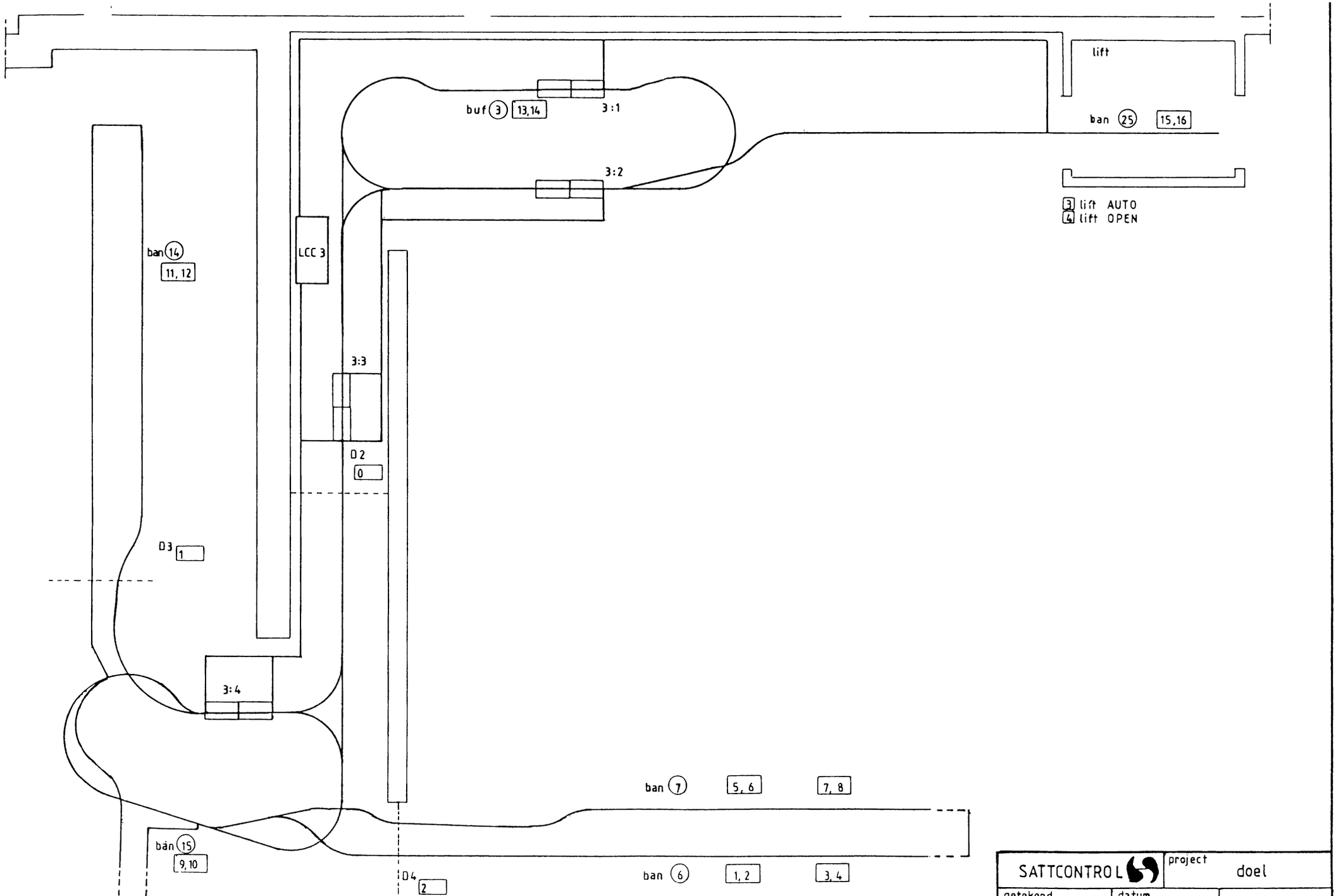
```
DIR (ECTORY)   <filespec>  => PIP <filespec>/LI.
DEL(ETE)       <filespec>  => PIP <filespec>/DE.
TYP(E)         <filespec>  => PIP TI: = <filespec>
CHD            (x,x)       => SET /UIC=[x,x]
                              change/show uic.
ATS            (TTnn(:))   => ACT/ALL if ddnn:
                              is ommited otherwise
                              ACT/TERM=TTnn:
SHQ                        => QUE /LI/FU
                                 List all print queues with
                              complete information.
SHW                        => QUE/LI
                              List all printqueues in brief.
DLG                        => DEV/LOG
                              List all logged terminals.
```

FRE(E)		=> PIP ddnn:/FR ddnn is omitted: replaced by SY:. List number of free blocks.
POO(L)		=> SET /POOL List permissible primary pool.
SYS(TEM)		=> SET /SYSUIC Show current system uic.
ASS(IGN)	(ddnn(:))	=> ASN ddnn: = SY: ddnn is omitted, replaced by ASN. Put default unit or show default unit.
CRE(ATE)	<filespec>	=> KED <filespec>/CR. create new file and start inputing with KED editor.
PUR(GE)	<filespec>	=> PIP <filespec>/PU
DLY	nnnnu	=> delay nnnn time units. u=T-tick S-seconds M-minutes H-hours
HEY	task	"connect" to task, i.e. lock to task and display message when task exits, also give exit status.
ASK	nnnnu,xxxx...	=> put question with time out time nnnu(see DLY above). The reply given is returned in the exit status as follows: 1*400 =Y input 2*400 = N 3*400 = <CR>input 4*400 = <ALT> 5*400 = UZ input 6*400 = timeout 7*400 = other replies
CVT	unnnnnn	=> Converting of nnnnnn where u specifies base. = octal % = radix-50 " = ascii $ = hexadecimal = decimal All the above named types are converted.

STO(P)	ddnn(:)	=>	QUE ddnn:/KIL stops active print job on the ddnn unit.
CLR		=>	clear CRT screen.
TDX		=>	display catchall task's version number and ufd's.(see below)



SATTCONTROL	project	doel
getekend Hvn L	datum 10-9-'85	



SATTCONTROL	project doel	
getekend H.v.L	datum 11-9-85	

```
      COMMON /TCSYRW/
     *  DEBORD              (10),
     *  CTKSR             (13,25),
     *  CTKSRQ                 ,
     *  CTR               (13, 3),
     *  CTRQ                   ,
     *  CTBR              (12, 3),
     *  CTBRQ                  ,
     *  CTBZON            (12, 2),
     *  CTBZOQ                 ,
     *  CTHPR             ( 6, 2),
     *  CTHPRQ                 ,
     *  CTIMER              ( 10),
     *  CTBLOC              (  3),
     *  CTSYSS              (  5),
     *  CTBURB                (1),
     *  CTURBI                (1),
     *  CTUR               (7,10),
     *  CTURQ                  ,
     *  CTSASL            ( 42,2),
     *  CTSAPH            ( 42,2),
     *  CTADRQ                 ,
     *  CTRSET                 ,
     *  CTSERV                (2),
     *  CTTRAN                (2),
     *  CTDTAB               (48),
     *  CTTUST               (20),
     *  CTCREG             (5,2),
     *  CTLEVQ                 ,
     *  CTNAME                (3),
     *  CTLOOP               (2,7),
     *  DUM10                 (18)

      INTEGER DEBORD
      BYTE    CTDTAB
      INTEGER CTKSR
      INTEGER CTKSRQ
      INTEGER CTLOOP
      CHARACTER*6 CTNAME
      INTEGER COR
      INTEGER CORQ
      INTEGER CTR
      INTEGER CTRQ
      INTEGER CTBR
      INTEGER CTBRQ
      BYTE    CTBZON
      INTEGER CTBZOQ
      INTEGER CTHPR
      INTEGER CTHPRQ
      INTEGER CTKGVR
      INTEGER CTKGVQ
      INTEGER CTIMER
      INTEGER CTBLOC
      INTEGER CTSYSS
      INTEGER CTURBI
      INTEGER CTBURB
      INTEGER CTUR
      INTEGER CTURQ
      INTEGER CTSASL
      INTEGER CTSAPH
      INTEGER CTADRQ
      INTEGER CTRSET
      INTEGER CTSERV
      INTEGER CTTRAN
      INTEGER CTTUST
```

!TRUCK REGISTER
=3

```
!
! Sub-buffertregister
! Halt-place register
!
! Timerregister
! Block-bytes from LCC
! System-status
! Bitmap idle ass. in TUR
! Bitmap ass. in TUR
! AGV-assignment register
!
! Conversion F-addr/Comm.point
! Conversion F-addr/PHU
!
!
! Service-address
! Transfer-address
! Run-time parameter set
! Statistics
! Charge-control
! Nr of levels
! Flow-way name
      ! Loop time-out
!
```

```
      INTEGER CTCREG
      INTEGER CTLEVQ
      INTEGER DUM10

      INTEGER CTCMIN
      EQUIVALENCE (CTCMIN,CTIMER(5))
      INTEGER CTCDIF
      EQUIVALENCE (CTCDIF,CTIMER(3))
```

.MAIN. MACRO Y05.02 Friday 31-Jan-86 14:50
Table of contents

```
       3
       4
       5
       6
*************************
       7                          *
       8                          *
       9                          *
      10
*************************
      11
      12
      13
      14
      15
      16
on
      17
 the customer
      18
omer
      19
      20
-----------------------------
      21
      22
      23
```

```
        .ENABL LC
;+
        .IDENT  /N01.01/
;*************************************

;*

;*      Copyright:      SattControl Malm

;*

;*************************************

;
;       Signature:      CF      820622
;
;       Revisions:
;
;       Nxx = Revisions in standardversi

;       Lxx = Revisions made by SATT for

;       Kxx   Revisions made by the cust

;
;-----------------------------------------

; Rev.  Date    Sign.   Description
;
;
```

```
      25                                ;
      26                                ;       Description:
      27                                ;
      28                                ;       Commondata for the AGV-system
      29                                ;-
```

```
      31                                                .IFNDF V01
      32                                                .ERROR 0 Erroneous TSDATA-versio
n
      33                                                .ENDC
      34                                                .SBTTL  Macro-definitions
      35          000052                                MAXADR==DEFADR
      36          000002                                MAXZON==DEFHPL
      37          000002                                MAXLEV==DEFLEV
      38          000003                                TOPLCC==MAXLCC
```

```
      40 000000                                          .PSECT  TCSYRW,RW,D,GBL,REL,OVR
      41 000000                                TCSYRW:
      42                                                 .SBTTL  Debug
      43                                                 ;
      44                                                 ;
      45 000000  177777  177777  177777                  .WORD   -1.,-1.,-1.,-1.
         000006  177777
      46 000010                                          .BLKW   6.
```

```
      48                                        .IF DF DEFID
      49                                        .IF GT DEFID
      50                                        .SBTTL  ID-stationregister
      51                                  ;
      52                                  ;
      53                                  ;ORD
      54                                  ;1      Status
      55                                  ;                         BIT 1   Cargo pr
esent
      56                                  ;                         BIT 2   Processi
ng done
      57                                  ;                         BIT 3   Reject
      58                                  ;                         BIT 4   Request
for reprocessing
      59                                  ;                         BIT 5   Cargo co
mming
      60                                  ;
      61                                  ;2      Result from processing
      62                                  ;
      63                                  ;                         BIT 1   Height 1
      64                                  ;                         BIT 2   Height 2
      65                                  ;3
      66                                  ;4      Address
      67                                  ;5      UNR
      68                                  ;6      Current AGV
      69                                  ;
      70                                  ;
      71                                        I=0
      72                                        .MACRO  ID,SFW,ADRESS
      73                                        .WORD   0
      74                                        .WORD   SFW'
      75                                        .WORD   0
      76                                        .WORD   ADRESS'.
      77                                        .BLKW   2
      78                                        I=I+1
      79                                        .ENDM
      80                                  CIKIR::
      81                                        IDREG
      82                                        .IF NE DEFID-I
      83                                        .ERROR  I       ;Erroneous numbe
r of ID-macros issued
      84                                        .ENDC
      85                                  CIKIRQ::.WORD   I
      86                                        .ENDC
      87                                        .ENDC
```

```
      89                                                .IF DF DEFBAN
      90                                                .IF GT DEFBAN
      91                                                .SBTTL  Transfer Register
      92                                          ;
      93                                          ;
      94                                          ;WORD
      95                                          ;1        Status
      96                                          ;                 BIT 1   Cargo present (P
ick)
      97                                          ;                 BIT 2   Full
      98                                          ;
      99                                          ;                 BIT 15  Cargo present (D
eposit)
     100                                          ;
     101                                          ;2        "Soft-status"
     102                                          ;
     103                                          ;                 BIT 1   Deposit allowed
APM
     104                                          ;                 BIT 2   Pick allowed APM
     105                                          ;                 BIT 3   Deposit allowed
ALL
     106                                          ;                 BIT 4   Pick allowed ALL
     107                                          ;                 BIT 5   FIFO
     108                                          ;                 BIT 6   Assignment sent
to TUADM by TPOLL
     109                                          ;                 BIT 7   Tr.unit connecte
d to crane
     110                                          ;                 BIT 8   Tr.unit NOT conn
ected to AGV-sys
     111                                          ;                 BIT 9   Tr.unit is final
 dest. for assignment.
     112                                          ;
     113                                          ;3        Deposit-address
     114                                          ;4        Pickaddress
     115                                          ;5        UNR
     116                                          ;6        Number of UNR:s
     117                                          ;7        Priority
     118                                          ;8        Timeout-timer
     119                                          ;9        Timeouttime
     120                                          ;10       Max # of assignments
     121                                          ;11       Current # of assignments
     122                                          ;12       Crane #
     123                                          ;13       GK's own word
     124                                          ;
     125                                          ;
     126          000001                                   APML=1
     127          000002                                   APMH=2
     128          000004                                   ALLL=4
     129          000010                                   ALLH=10
     130          000020                                   FIFO=20
     131          000040                                   GUPP=40
     132          000100                                   KRAN=100
     133          000200                                   NSND=200
     134          000400                                   KVIT=400
     135                                          ;
     136                                                   .MACRO  BAN,PADRESS,DADRESS,TMO,
PR~,SFWSTAT,ASSNR
     137                                                   .NARG ANT
     138                                                   .IIF NE ANT-6 .ERROR I+1 Incorre
ct # of parameters in BAN
     139                                                   .WORD   0
     140                                                   .WORD   SFWSTAT
```

```
141                             .WORD   DADRESS'.
142                             .WORD   PADRESS'.
143                             .BLKW   2
144                             .WORD   PRIO'.
145                             .WORD   0
```

```
    146                                         .WORD   TM0'.
    147                                         .WORD   ASSNR'.
    148                                         .BLKW   3.
    149                                         I=I+1
    150                                         .ENDM
    151                                 ;
    152                                 ;
    153 000024                          CTKSR:
    154           000000                        I=0
    155 000024                                  BANREG
    156                                         .IF GT DEFBAN-I
    157                                         .REPT   DEFBAN-I
    158                                         BAN     0,0,0,0,0,0
    159                                         .ENDR
    160                                         .IFF
    161                                         .IF LT DEFBAN-I
    162                                         .ERROR  I       ;Erroneous numbe
r of BAN-macros issued
    163                                         .ENDC
    164                                         .ENDC
    165 001236  000031                  CTKSRQ::.WORD   I
    166                                         .ENDC
    167                                         .ENDC
```

```
    169                                         .SBTTL  AGV-register
    170                                 ;
    171                                 ;ORD:
    172                                 ;1      Status
LDHSTATUS
    173                                 ;       BIT 1   AUTO
LDAUTO
    174                                 ;           2   Cargo present
LDPALLPJ
    175                                 ;           3   ---
    176                                 ;           4   ---
    177                                 ;           5   Temperature
    178                                 ;           6   Charge-warning
    179                                 ;           7   Manual
    180                                 ;           8   ---
    181                                 ;           9   New strategi wanted
    182                                 ;          10   Strategi error
    183                                 ;          11   FWD
    184                                 ;          12   Reversed
    185                                 ;
    186                                 ;2      Software-status
    187                                 ;       BIT 1   Not installed
LDINSTALL
    188                                 ;           2   Man. mode
LDMAN           VFS LOGGFIL
    189                                 ;           3   Not in system
LDSYSTEM        VFS LOGGFIL
    190                                 ;           4   Service
LDSERVICE
    191                                 ;           5   Req. for service
LDSIND
    192                                 ;           6   Req. for manual
LDMIND
    193                                 ;           7   Req. for charge
LDLIND
    194                                 ;           8   Charge not allowed
LDNCRG
    195                                 ;
    196                                 ;          12   Not on comm.point
    197                                 ;
    198                                 ;3      Destination
    199                                 ;4      Reserve
    200                                 ;5      AGV-assignment-nr
LDTUN           VFS TUR
    201                                 ;6      Position
LDPOS
    202                                 ;7      Timer
LDKLOCKA
    203                                 ;8      Charge-timer
LDLADNU
    204                                 ;9      Total charge
LDLADTOT        VFSLOGGFIL
    205                                 ;10     Recentmost funktion-address
LDLAMN
    206                                 ;11     Time-out timer
LDTOTIM
    207                                 ;12     Reserve
    208                                 ;13     Ordered address
    209                                         .MACRO TRUCK
    210                                         I=I+1
    211                                         .WORD   0
    212                                         .IF GE MAXAGV-I
```

pos, xy

pos1 = CTR(6,1)

```
213                                             .WORD   0
214                                             .IFF
215                                             .WORD   1
216                                             .ENDC
217                                             .BLKW   11.
218                                             .ENDM
219 001240                                  CTR::
220             000000                          I=0
221             000003                          .REPT   DEFAGV
222                                             TRUCK
223                                             .ENDR
224 001356      000003                  CTRQ::  .WORD   MAXAGV = 3
```

```
                                                .IF DF DEFBUF
                                                .IF GT DEFBUF
                                                .SBTTL  Buffer register
                                        ;
                                        ;WORD:
                                        ;1      Status
                                        ;                       Bit 1   : ---
                                        ;                       Bit 2   :Charger
 on
                                        ;       Softstatus
                                        ;                       Bit 1   :Charger
 is available
                                        ;                       Bit 2   :Chargin
g permitted
                                        ;
                                        ;3      Address to charger
                                        ;4      Address to buffer
                                        ;5      Link to next buffer "in line"
                                        ;6      Current AGV
                                        ;7      Link to buffergroup
                                        ;8      Link to previous buffer "in line
"
                                        ;9      Buffer reserved for AGV
                                        ;10     Pointer to next buffer "in paral
lell"
                                        ;11     Charge-timer
                                        ;12     Total charging time
                                        ;
                                                I=0
                                                .MACRO  BUF,SFW,CADR,BADR,FLINK,
BLINK,GROUP,NEXT
                                                .WORD   0
                                                .WORD   SFW'
                                                .WORD   CADR'.
                                                .WORD   BADR'.
                                                .WORD   FLINK'.
                                                .WORD   0
                                                .WORD   GROUP'.
                                                .WORD   BLINK'.
                                                .WORD   0
                                                .WORD   NEXT'.
                                                .BLKW   2
                                                I=I+1
                                                .ENDM
                                        ;
                                                BU=0
                                                TL=1
                                        ;
                                        CTBR::
                                                BUFREG
                                                .IF NE DEFBUF-I
                                                .ERROR  I       ;Erroneous numbe
r of BUF-macros issued
                                                .ENDC
                                        ;
                                        CTBRQ::.WORD    I
                                                .ENDC
                                                .ENDC
```

```
                                        .IF DF DEFBZO
                                        .IF GT DEFBZO
                                        .SBTTL  Buffer-group register
                                ;
                                ;
                                ;       Register of buffergroups
                                ;
                                ;
                                ;BYTE
                                ;1      Max. nr of AGV:s in group
    278                 LDBZSZ
                                ;2      Min. nr of AGV:s in group
    279                 LDBZMIN
                                ;3      Current nr of AGV:s in group
    280                 LDBZNU
                                ;4      Pointer to first buffer in group
    281  (to pick AGV)  LDBZB1
                                ;5      Pointer to last buffer in group
    282  (   pick AGV)  LDBZB2
                                ;6      Link to next buffergroup
    283                 LDBZBZ
                                ;7      Pointer to first buffer in group
    284  (to send AGV)  LDBZB3
                                ;8      Pointer to last buffer in group
    285  (to send AGV)  LDBZB4
                                ;9      Max. nr of AGV:s in charger
    286                 LDBCSZ
                                ;10     Min. nr of AGV:s in charger
    287                 LDBCMIN
                                ;11     Current nr of AGV:s in charger
    288                 LDBCNU
                                ;12     ---
                                ;
    289
    290
    291                         .MACRO  BZO,SIZE,MIN,FIRST,LAST,
LINK,SFIRST,SLAST
    292                         .BYTE   SIZE'.
    293                         .BYTE   MIN'.
    294                         .BYTE   0
    295                         .BYTE   FIRST'.
    296                         .BYTE   LAST'.
    297                         .BYTE   LINK'.
    298                         .BYTE   SFIRST'.
    299                         .BYTE   SLAST'.
    300                         .BYTE   SIZE'.
    301                         .BYTE   MIN'.
    302                         .BYTE   0
    303                         .BYTE   0
    304                         I=I+1
    305                         .ENDM
    306
    307                 ;
    308                 ;
    309                 CTBZON::
    310                         I=0
    311 001472  000000          BZON
    312 001472          ;
    313                         .IF NE DEFBZO-I
    314                         .ERROR  I       ;Erroneous bumbe
r of BZO-macros issued
    315                         .ENDC
    316                 ;
    317 001522  000002  CTBZOQ: .WORD   I
```

```
327                                                     .ENDC
328                                                     .ENDC
```

```
    330                                             .IF DF DEFHPL
    331                                             .IF GT DEFHPL
    332                                             .SBTTL  Zone-register
    333                                         ;
    334                                         ;
    335                                         ;WORD:
    336                                         ;
    337                                         ;1      Status
LDSTAT
    338                                         ;2      Buffergroup
    339                                         ;3      Alternative assignment zone
    340                                         ;4      Address
    341                                         ;5      Link to next zone
    342                                         ;6      Current AGV
    343                                         ;
    344          000000                             I=0
    345                                             .MACRO  HPL,GROUP,LINK,ADRESS,AL
TZON
    346                                             .WORD   0
    347                                             .WORD   GROUP'.
    348                                             .WORD   ALTZON'.
    349                                             .WORD   ADRESS'.
    350                                             .WORD   LINK'.
    351                                             .WORD   0
    352                                             I=I+1
    353                                             .ENDM
    354                                         ;
    355                                         ;
    356 001524                                  CTHPR:
    357 001524                                          HPLREG
    358                                             .IF NE DEFHPL-I
    359                                             .ERROR  I       ;Erroneous numbe
r of HPL-macros issued
    360                                             .ENDC
    361                                         ;
    362 001554  000002                          CTHPRQ::.WORD   I
    363                                             .ENDC
    364                                             .ENDC
```

```
                        .IF DF DEFKGV
                        .IF GT DEFKGV
                        .SBTTL  Aisle-selection register
                ;
                ;ORD:
                ;
                ;
                ;1      ---
                ;2      ---
                ;3      Link to next Aisle-sel.
                ;4      Address
                ;5      AGV-assignment number
                ;6      AGV
                ;
                        .MACRO  KGV,SFW,ADRESS,NEXT
                        .WORD   0
                        .WORD   SFW'
                        .WORD   NEXT'.
                        .WORD   ADRESS'.
                        .BLKW   2
                        I=I+1
                        .ENDM
                ;
                CTKGVR::
                        I=0
                        KGVREG
                        .IF GT DEFKGV-I
                        .REPT   DEFKGV-I
                        KGV     0,0,0
                        .ENDR
                        .IFF
                        .IF LT DEFKGV-I
                        .ERROR  I       ;Erroneous numbe
r of KGV-macros issued
                        .ENDC
                        .ENDC
                CTKGVQ::.WORD   I               ;Number
of  isle-selection-points
                        .ENDC
                        .ENDC
```

```
    405                                          .SBTTL Timer register
    406                                 ;
    407                                 ;
    408                                 ;WORD:
    409                                 ;
    410                                 ;1       Time out time AGV:s (between com
.points) in intervals
    411                                 ;2       Timer (in intervals)
    412                                 ;3       Min diff in ch-time when swappin
g AGV:s
    413                                 ;4       Assignment priorityincrement/min
    414                                 ;5       Shortest charging time
    415                                 ;6       Time before charging
    416                                 ;
    417 001556  000017                  CTIMER: .WORD   15.
    418 001560  000000                          .WORD   0
    419 001562  000036                          .WORD   30.
    420 001564  000001                          .WORD   1
    421 001566  000003                          .WORD   3
    422 001570  000001                          .WORD   1
    423 001572                                  .BLKW   4
```

```
    425                                         .SBTTL Blockinformation-register
    426                                 ;
    427                                 ;       Register of the block-status in
the LCC:s
    428                                 ;       The register is updated on deman
d from program TBL
    429                                 ;
    430                                 ;WORD:
    431                                 ;1      LCC-nr
    432                                 ;2      Communicationpoint in LCC (1-15)
    433                                 ;3      Status
    434                                 ;
    435 001602                          CTBLOCKSTATUS:  .BLKW   3.
```

```
    437                                         .SBTTL Systemstatus
    438                                 ;
    439                                 ;CTSYSSTAT(A)
    440                                 ;
    441                                 ;WORD
    442                                 ;1      AGV-polling status
    443                                 ;                                   Bit 15
'One-shot' polling
    444                                 ;                                   Bit 16
Polling not permitted
    445                                 ;
    446                                 ;2      ---
    447                                 ;3      ---
    448                                 ;4      ---
    449                                 ;5      ---
    450                                 ;
    451 001610                          CTSYSSTAT:
    452 001610  000000                          .WORD   0.
    453 001612                                  .BLKW   4.
```

```
                                                        .SBTTL  Assignment-register
                                                ;
                                                ;WORD
                                                ;
                                                ;1      Status
                                                ;2      Local-address
                                                ;3      Final-address
                                                ;4      ---
                                                ;5      Assignment-nr (AD)
                                                ;6      AGV-nr
                                                ;7      Reserve
                                                ;
001622                                          CTBURBIT::
001622                                                  .BLKW   <<DEFASS-1>/16.>+1
001624                                          CTURBIT::
001624                                                  .BLKW   <<DEFASS-1>/16.>+1
                                                ;
001626                                          CTUR:   .BLKW   DEFASS*7
002042  000012                                  CTURQ:: .WORD   DEFASS
                                                ;
                                                ;
```

```
    477                                                 .SBTTL  Conversion-table F-add/P
HU
    478                                         ;
    479                                         ;
    480                                                 .MACRO  SADAT,SLINGA,PHENHEI,FUN
CTION,ZON,SUBZ,NXTZ
    481                                                 I=I+1
    482                                                 .WORD   SLINGA'.
    483                                                 .ENDM
    484 002044                                  CTSASL::
    485 002044                                          SADEF
A       002212  000000G 000304  002115                SIDAT   304,2115,BANAH,2,0
        ;2010
        002220  000000G 000002  000000
P       002326  000051                              .ERROR  I       ;Erroneous numbe
r of addresses issued
```

```
    487                                                 .SBTTL  Conversion-table F-add./
Comm.point
    488                                         ;
    489                                         ;;
    490                                         ;;
    491                                                 .MACRO  SADAT,SLINGA,PHENHET,FUN
CTION,ZON,SUBZ,NXTZ
    492                                                 I=I+1
    493                                                 .WORD   PHENHET'.
    494                                                 .ENDM
    495 002326                                  CTSAPH::
    496 002326                                          SADEF
A       002474  000000G 000304  002115                  SIDAT   304,2115,BANAH,2,0
        ;2010
        002502  000000G 000002  000000
P       002610  000051                              .ERROR  I       ;Erroneous numbe
r of addresses issued
    497                                         ;
    498                                         ;;
    499                                         ;;
    500                                         ;;
    501 002610  000052                          CTADRQ::.WORD   MAXADR
    502 002612  000177                          CTRSET::.WORD   RESET
    503                                         ;
    504                                         ;
```

```
     506                                                .SBTTL  Service-register
     507                                        ;
     508                                        ;       Definition of service points
     509                                        ;
     510                                                .MACRO  SERVICE,SADR
     511                                                .WORD   SADR'.
     512                                                I=I+1
     513                                                .ENDM
     514                                        ;
     515 002614                                 CTSERV:
     516 002614                                         SERVREG
     517                                                .IF NE DEFLEV-I
     518                                                .ERROR  I       ;Erroneous numbe
r of serviceloops
     519                                                .ENDC
     520                                        ;
     521                                        ;
     522                                        ;       Definition of transfer point
     523                                        ;
     524                                                .MACRO  TRANSF,SADR
     525                                                .WORD   SADR'.
     526                                                I=I+1
     527                                                .ENDM
     528                                        ;
     529 002620                                 CTTRAN:
     530          000000                                I=0
     531 002620                                         TRFREG
     532                                                .IF NE DEFLEV-I
     533                                                .ERROR  I       ;Erroneous numbe
r of transferaddresses
     534                                                .ENDC
     535                                        ;
```

```
     537                                ;       Run-time parameter set
     538                                ;
     539          002624'               $S=.
     540 002624                         CTDTAB:
     541 002624                                 DLAGE
     542                                        .IF NE .-$S-48.
     543                                        .ERROR  .-$S    ;Errouneous numb
er of hours
     544                                        .ENDC
     545                                ;
     546                                ;
     547 002704                         CTTUST: .BLKW   20.     ;Statistics
     548                                ;
     549                                ;
```

```
                                        ;
                                        ;
                                        ;       Charge-control data
                                        ;
                                        ;       1       Max # of AGV:s in produc
tion
                                        ;       2       Current # of AGV's in pr
oduction
                                        ;       3       AGV in production witch
lowest chargetimer
                                        ;       4       Charge-timer for Low-chr
g AGV
                                        ;       5       Not used
                                        ;
                                        ;
002754                                  CTCREG::
                000002                          .REPT   DEFLEV
                                                .WORD   DEFAGV
                                                .WORD   0
                                                .WORD   0
                                                .WORD   77777
                                                .WORD   0
                                                .ENDR
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
003000  000002                          CTLEVQ::.WORD   DEFLEV
                                        ;
                                        ;
003002                                  CTNAME::
                000003                          .REPT   DEFNAM
                                                .WORD   0
                                                .WORD   0
                                                .WORD   0
                                                .ENDR
                                        ;
                                        ;
003024                                  CTLOOP::
                000007                          .REPT   DEFLOP
                                                .WORD   0
                                                .WORD   0
                                                .ENDR

                003060                          I=.-TCSYRW
                000060                          J=I&77
                                                .IF NE J
                000020                          I=100-J
003060                                          .BLKB   I
                                                .ENDC
003100  000000                                  .WORD   0
                000001                          .END
```

Symbol table

```
AC  EV= 000000           CTHPRQ  001554RG    002 CTURQ   002042RG    002 J
000060          PPALW = 000001
ALLH  = 000010           CTIMER  001556R     002 DEFADR= 000052          KRAN  =
000100          RESET = 000177
ALLL  = 000004           CTKSR   000024R     002 DEFAGV= 000003          KVIT
000400          SIDAT = ******  GX
ANT   = 000006           CTKSRQ  001236RG    002 DEFASS= 000012          MAXADR=
000052 G        TCSYRW  000000R     002
APMH  = 000002           CTLEVQ  003000RG    002 DEFBAN= 000031          MAXAGV=
000003          TL    = 000001
APML  = 000001           CTLOOP  003024RG    002 DEFBUF= 000003          MAXLCC=
000003          TOPLCC= 000003 G
BANAH = ****** GX        CTNAME  003002RG    002 DEFBZO= 000002          MAXLEV=
000002 G        TYPAGV= 000031
BU    = 000000           CTR     001240RG    002 DEFGRP= 000003          MAXZON=
000002 G        TYPBAN= 000025
CTADRQ  002610RG    002 CTRQ    001356RG    002 DEFHPL= 000002          NSND  =
000200          TYPBUF= 000036
CTPOC   001602R     002 CTRSET  002612RG    002 DEFKGV= 000000          PDOOR1=
00.02           TYPBZO= 000035
CTBR    001360RG    002 CTSAPH  002326RG    002 DEFLCC= 000003          PDOOR4=
000001          TYPCRA= 000030
CTBRQ   001470RG    002 CTSASL  002044RG    002 DEFLEV= 000002          PLA10
000001          TYPHPL= 000044
CTBURB  001622RG    002 CTSERV  002614R     002 DEFLIN= 000001          PLA18
000001          TYPID = 000024
CTBZON  001472RG    002 CTSYSS  001610R     002 DEFLOP= 000007          PL10
000002          TYPKGV= 000047
CTBZOQ  001522R     002 CTTRAN  002620R     002 DEFNAM= 000003          PL18
000002          TYPPVG= 000037
CTCREG  002754RG    002 CTTUST  002704R     002 FIFO  = 000020          PPALFC=
000001          TYPSER= 000041
CTDTAB  002624R     002 CTUR    001626R     002 GUPP  = 000040          PPALFD=
000001          V01   = 000001
CTHPR   001524R     002 CTURBI  001624RG    002 I     = 000020          PPALVK=
000001          $S    = 002624R     002


. ...S.  000000    000   (RW,I,GBL,ABS,OVR)
         000000    001   (RW,I,LCL,REL,CON)
TCSYRW   003102    002   (RW,D,GBL,REL,OVR)
Errors detected:  4

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12552 Words  ( 50 Pages)
Size of core pool: 13480 Words  ( 51 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:44.48
,TI4=TSDATA,TCSYRW
```

```
C
C Symbolic names for register ARSYS
C
      INTEGER ARSYS
      PARAMETER (ARSYS=-1)
      INTEGER SFIFW
      PARAMETER (SFIFW=1)
      INTEGER SLAFW
      PARAMETER (SLAFW=2)
      INTEGER SFACT
      PARAMETER (SFACT=3)
      INTEGER SLACT
      PARAMETER (SLACT=4)
C
C Symbolic names for register APFLW
C
      INTEGER APFLW
      PARAMETER (APFLW=-2)
      INTEGER QFLWNA
      PARAMETER (QFLWNA=1)
      INTEGER QSOFLW
      PARAMETER (QSOFLW=2)
C
C Symbolic names for register ARFLW
C
      INTEGER ARFLW
      PARAMETER (ARFLW=-3)
      INTEGER FFLWNA
      PARAMETER (FFLWNA=1)
      INTEGER FSNR1
      PARAMETER (FSNR1=2)
      INTEGER FSNR2
      PARAMETER (FSNR2=3)
      INTEGER FSNR3
      PARAMETER (FSNR3=4)
      INTEGER FSNR4
      PARAMETER (FSNR4=5)
      INTEGER FSNR5
      PARAMETER (FSNR5=6)
      INTEGER FSNR6
      PARAMETER (FSNR6=7)
      INTEGER FSNR7
      PARAMETER (FSNR7=8)
      INTEGER FSNR8
      PARAMETER (FSNR8=9)
      INTEGER FSNR9
      PARAMETER (FSNR9=10)
      INTEGER FSNR10
      PARAMETER (FSNR10=11)
      INTEGER FSOAFW
      PARAMETER (FSOAFW=12)
      INTEGER FBRPRE
      PARAMETER (FBRPRE=13)
      INTEGER FBRSUC
      PARAMETER (FBRSUC=14)
C
C Symbolic names for register ARASS
C
      INTEGER ARASS
      PARAMETER (ARASS=-4)
      INTEGER UFLWNR
      PARAMETER (UFLWNR=1)
      INTEGER USNR1
      PARAMETER (USNR1=2)
      INTEGER USNR2
```

```
      PARAMETER (USNR2=3)
      INTEGER USNR3
      PARAMETER (USNR3=4)
      INTEGER USNR4
      PARAMETER (USNR4=5)
      INTEGER USNR5
      PARAMETER (USNR5=6)
      INTEGER USNR6
      PARAMETER (USNR6=7)
      INTEGER USNR7
      PARAMETER (USNR7=8)
      INTEGER USNR8
      PARAMETER (USNR8=9)
      INTEGER USNR9
      PARAMETER (USNR9=10)
      INTEGER USNR10
      PARAMETER (USNR10=11)
      INTEGER UPALUN
      PARAMETER (UPALUN=12)
      INTEGER USTPOI
      PARAMETER (USTPOI=13)
      INTEGER UPULIN
      PARAMETER (UPULIN=14)
      INTEGER UDELIN
      PARAMETER (UDELIN=15)
      INTEGER UASTAT
      PARAMETER (UASTAT=16)
      INTEGER UERCOD
      PARAMETER (UERCOD=17)
C
C Symbolic names for register ARLIN
C
      INTEGER ARLIN
      PARAMETER (ARLIN=-5)
      INTEGER LMAX
      PARAMETER (LMAX=1)
      INTEGER LACT
      PARAMETER (LACT=2)
C
C Symbolic names for register ARACT
C
      INTEGER ARACT
      PARAMETER (ARACT=-6)
      INTEGER AFLWNR
      PARAMETER (AFLWNR=1)
      INTEGER ATIME
      PARAMETER (ATIME=2)
      INTEGER APULIN
      PARAMETER (APULIN=3)
      INTEGER ADELIN
      PARAMETER (ADELIN=4)
      INTEGER AFWTYP
      PARAMETER (AFWTYP=5)
      INTEGER AFSTAT
      PARAMETER (AFSTAT=6)
      INTEGER ARQPAL
      PARAMETER (ARQPAL=7)
      INTEGER ASTART
      PARAMETER (ASTART=8)
      INTEGER APUPAL
      PARAMETER (APUPAL=9)
      INTEGER AEND
      PARAMETER (AEND=10)
      INTEGER ALEVL1
      PARAMETER (ALEVL1=11)
```

```
      INTEGER ALEVL2
      PARAMETER (ALEVL2=12)
      INTEGER ABRPRE
      PARAMETER (ABRPRE=13)
      INTEGER ABRSUC
      PARAMETER (ABRSUC=14)
```

```
C*********************************************************************
C
C       This copy file is used in the AD programs
C
CC********************************************************************
C
C
      COMMON /CADMRO/
     * CALINR,                    ! Line record number
     * CAMOLI,                    ! More than one line at station
     * CAMXLI,                    ! Max. number of pallets on line
     * CASTNA,                    ! Station name
     * CASTLE,                    ! Station level
     * TIMOUT,                    ! Time out
     * CAMXST,                    ! Max. P/D-stations
     * RBUFF,                     ! Receive buffer
     * SBUFF,                     ! Send buffer
     * CAGVLI,                    ! AD- AGV conversion lines
     * CAGVST                     ! AD- AGV conversion stations
C
C
      INTEGER          CAGVLI(15)       ! N
      INTEGER          CAGVST(15)       ! O
      INTEGER          CALINR(7)        ! R
      LOGICAL          CAMOLI(7)        ! M
      INTEGER          CAMXLI(7)        ! X
      INTEGER          CASTLE(15)       ! L
      CHARACTER*4      CASTNA(15)       ! S
      INTEGER          TIMOUT
      INTEGER          CAMXST
      INTEGER          RBUFF(15)
      INTEGER          SBUFF(13)
C
      CHARACTER*4
     * S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,
     * S11,S12,S13,S14,S15
C
      INTEGER          M1,M2,M3,M4,M5,M6,M7
      INTEGER          L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,
     *                 L13,L14,L15
      INTEGER          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,
     *                 N13,N14,N15
      INTEGER          O1,O2,O3,O4,O5,O6,O7,O8,O9,O10,O11,O12,
     *                 O13,O14,O15
      INTEGER          R1,R2,R3,R4,R5,R6,R7
      INTEGER          X1,X2,X3,X4,X5,X6,X7
C
C       Station names used for conversion to number
C
      DATA     S1/'B'/
      DATA     S2/'H'/
      DATA     S3/'S'/
      DATA     S4/'E'/
      DATA     S5/'F'/
      DATA     S6/'K'/
      DATA     S7/'I'/
      DATA     S8/'C'/
      DATA     S9/'D'/
      DATA     S10/'EC'/
      DATA     S11/'ED'/
      DATA     S12/'W'/
      DATA     S13/'VK'/
      DATA     S14/'P'/
      DATA     S15/'LIFT'/
C
```

```
C       If more than 1 line M is .TRUE.
C
        DATA    M1/.TRUE./
        DATA    M2/.TRUE./
        DATA    M3/.TRUE./
        DATA    M4/.FALSE./
        DATA    M5/.FALSE./
        DATA    M6/.FALSE./
        DATA    M7/.FALSE./
C
CC      Conversion AD- AGV  system lines
C
        DATA    N1/1007/        ! B1
        DATA    N2/1009/        ! B2
        DATA    N3/1011/        ! B3
        DATA    N4/1013/        ! B4
        DATA    N5/1015/        ! B5
        DATA    N6/2001/        ! H1
        DATA    N7/2005/        ! H2
        DATA    N8/1033/        ! S1
        DATA    N9/1035/        ! S2
        DATA    N10/1037/       ! S3
        DATA    N11/1039/       ! S4
        DATA    N12/1029/       ! E
        DATA    N13/1031/       ! F
        DATA    N14/2011/       ! K
        DATA    N15/2009/       ! I
C
CC      Conversion AD- AGV system stations
C
        DATA    O1/0/
        DATA    O2/0/
        DATA    O3/0/
        DATA    O4/0/
        DATA    O5/0/
        DATA    O6/0/
        DATA    O7/0/
        DATA    O8/1019/
        DATA    O9/1023/
        DATA    O10/1021/
        DATA    O11/1025/
        DATA    O12/1027/
        DATA    O13/1017/
        DATA    O14/1005/
        DATA    O15/0/

C
CC      Sx = CALINR(3) + x   calculation of line record
C
        DATA    R1/0/
        DATA    R2/5/
        DATA    R3/7/
        DATA    R4/11/
        DATA    R5/12/
        DATA    R6/13/
        DATA    R7/14/
C
CC      Stations are on level
C       (1 = level1, 2 = level2, 3   level1 and level2)
C
        DATA    L1/1/
        DATA    L2/2/
        DATA    L3/1/
        DATA    L4/1/
        DATA    L5/1/
```

```
      DATA     L6/2/
      DATA     L7/2/
      DATA     L8/1/
      DATA     L9/1/
      DATA     L10/1/
      DATA     L11/1/
      DATA     L12/1/
      DATA     L13/1/
      DATA     L14/1/
      DATA     L15/3/
C
C     Max. number of lines on the P/D-stations
C
      DATA     X1/5/
      DATA     X2/2/
      DATA     X3/4/
      DATA     X4/1/
      DATA     X5/1/
      DATA     X6/1/
      DATA     X7/1/
C
      DATA     TIMOUT/60/
      DATA     CAMXST/7/
C
      EQUIVALENCE     (CASINA(1),S1)
      EQUIVALENCE     (CASINA(2),S2)
      EQUIVALENCE     (CASINA(3),S3)
      EQUIVALENCE     (CASINA(4),S4)
      EQUIVALENCE     (CASINA(5),S5)
      EQUIVALENCE     (CASINA(6),S6)
      EQUIVALENCE     (CASINA(7),S7)
      EQUIVALENCE     (CASINA(8),S8)
      EQUIVALENCE     (CASINA(9),S9)
      EQUIVALENCE     (CASINA(10),S10)
      EQUIVALENCE     (CASINA(11),S11)
      EQUIVALENCE     (CASINA(12),S12)
      EQUIVALENCE     (CASINA(13),S13)
      EQUIVALENCE     (CASINA(14),S14)
      EQUIVALENCE     (CASINA(15),S15)
C
      EQUIVALENCE     (CASTLE(1),L1)
      EQUIVALENCE     (CASTLE(2),L2)
      EQUIVALENCE     (CASTLE(3),L3)
      EQUIVALENCE     (CASTLE(4),L4)
      EQUIVALENCE     (CASTLE(5),L5)
      EQUIVALENCE     (CASTLE(6),L6)
      EQUIVALENCE     (CASTLE(7),L7)
      EQUIVALENCE     (CASTLE(8),L8)
      EQUIVALENCE     (CASTLE(9),L9)
      EQUIVALENCE     (CASTLE(10),L10)
      EQUIVALENCE     (CASTLE(11),L11)
      EQUIVALENCE     (CASTLE(12),L12)
      EQUIVALENCE     (CASTLE(13),L13)
      EQUIVALENCE     (CASTLE(14),L14)
      EQUIVALENCE     (CASTLE(15),L15)
C
      EQUIVALENCE     (CAMOLI(1),M1)
      EQUIVALENCE     (CAMOLI(2),M2)
      EQUIVALENCE     (CAMOLI(3),M3)
      EQUIVALENCE     (CAMOLI(4),M4)
      EQUIVALENCE     (CAMOLI(5),M5)
      EQUIVALENCE     (CAMOLI(6),M6)
      EQUIVALENCE     (CAMOLI(7),M7)
C
      EQUIVALENCE     (CAGVLI(1),N1)
```

```
      EQUIVALENCE     (CAGVLI(2),N2)
      EQUIVALENCE     (CAGVLI(3),N3)
      EQUIVALENCE     (CAGVLI(4),N4)
      EQUIVALENCE     (CAGVLI(5),N5)
      EQUIVALENCE     (CAGVLI(6),N6)
      EQUIVALENCE     (CAGVLI(7),N7)
      EQUIVALENCE     (CAGVLI(8),N8)
      EQUIVALENCE     (CAGVLI(9),N9)
      EQUIVALENCE     (CAGVLI(10),N10)
      EQUIVALENCE     (CAGVLI(11),N11)
      EQUIVALENCE     (CAGVLI(12),N12)
      EQUIVALENCE     (CAGVLI(13),N13)
      EQUIVALENCE     (CAGVLI(14),N14)
      EQUIVALENCE     (CAGVLI(15),N15)
C
      EQUIVALENCE     (CAGVST(1),O1)
      EQUIVALENCE     (CAGVST(2),O2)
      EQUIVALENCE     (CAGVST(3),O3)
      EQUIVALENCE     (CAGVST(4),O4)
      EQUIVALENCE     (CAGVST(5),O5)
      EQUIVALENCE     (CAGVST(6),O6)
      EQUIVALENCE     (CAGVST(7),O7)
      EQUIVALENCE     (CAGVST(8),O8)
      EQUIVALENCE     (CAGVST(9),O9)
      EQUIVALENCE     (CAGVST(10),O10)
      EQUIVALENCE     (CAGVST(11),O11)
      EQUIVALENCE     (CAGVST(12),O12)
      EQUIVALENCE     (CAGVST(13),O13)
      EQUIVALENCE     (CAGVST(14),O14)
      EQUIVALENCE     (CAGVST(15),O15)
C
      EQUIVALENCE     (CALINR(1),R1)
      EQUIVALENCE     (CALINR(2),R2)
      EQUIVALENCE     (CALINR(3),R3)
      EQUIVALENCE     (CALINR(4),R4)
      EQUIVALENCE     (CALINR(5),R5)
      EQUIVALENCE     (CALINR(6),R6)
      EQUIVALENCE     (CALINR(7),R7)
C
      EQUIVALENCE     (CAMXLI(1),X1)
      EQUIVALENCE     (CAMXLI(2),X2)
      EQUIVALENCE     (CAMXLI(3),X3)
      EQUIVALENCE     (CAMXLI(4),X4)
      EQUIVALENCE     (CAMXLI(5),X5)
      EQUIVALENCE     (CAMXLI(6),X6)
      EQUIVALENCE     (CAMXLI(7),X7)
```