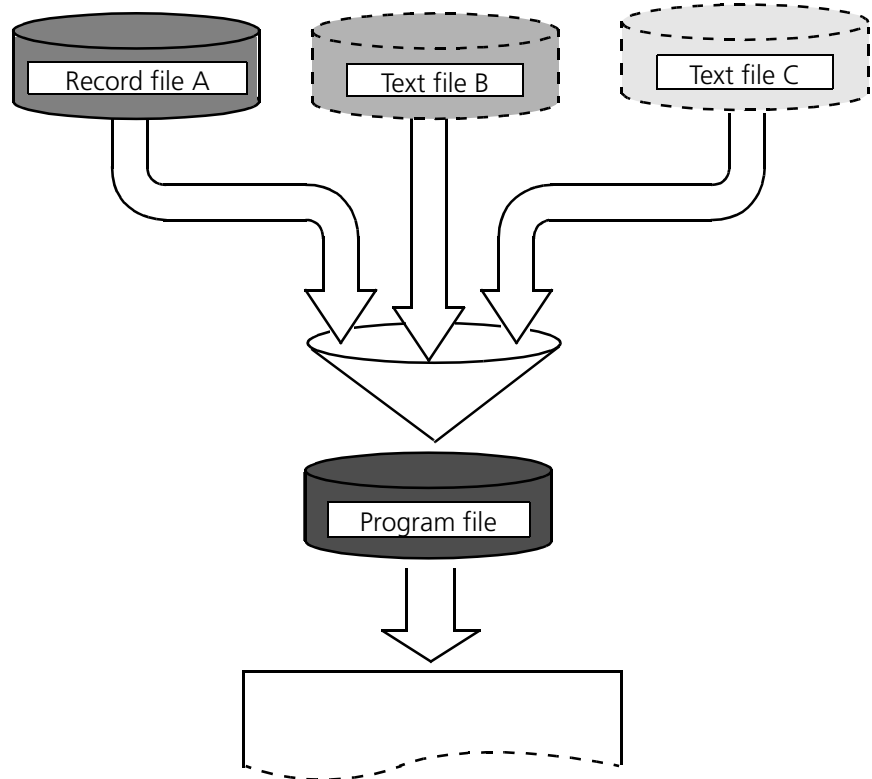


UMESS

Option 1 for UNIX and LINUX Extended Display of Measurement Results



Operating Instructions

The dissemination or copying of this manual, use or disclosure of its contents are prohibited without express permission. Violations will be prosecuted.

All rights reserved, especially regarding the granting of patents or the registration of utility models.

We reserve the right to revise this manual and make changes in the version and scope of delivery of the CMM, the software packages and the corresponding documentation.

Carl Zeiss does not guarantee this manual or provide any implied guarantee of its merchantable quality or suitability for any specific purpose whatsoever.

Carl Zeiss shall in no case be held liable for any errors contained in this manual or for any accidental or consequential damage in connection with its provision, function or use.

All product names are registered trademarks or trademarks of the corresponding proprietors.

Carl Zeiss
Industrial Metrology
Business Group
D-73446 Oberkochen

Document type: . . .Operating Instructions
Version:8.x
Date:04/2000
Order no.:61212-1020102

Preface

These operating instructions are based on the assumption that you are already familiar with the operation of the corresponding coordinate measuring machine and its components. Please keep all of the documents included in delivery within easy reach at all times.

Conventions in these operating instructions

Before beginning work with these operating instructions you should first familiarize yourself with the conventions they employ.

The following text will provide you with information on the fonts, characters and symbols used in this manual.

Typographical conventions

The fonts and type-faces used in these operating instructions have the following significance:

- **bold**
 - A dialog element displayed on the screen
Example: "... the **<TERMIN>** button"
 - A term
Example: "during the calculation the spatial position of a **part feature** is determined in reference to a **reference element**."
 - File and directory names
Example: **/home/zeiss/UB**
- *italics*
 - A highlighted text containing information of special importance
Example: "Click with the *right* mouse button ..."
 - Cross reference
Example: "..., see also ➤ „*Typographical conventions*“ on page -3"
- Courier
Program code, file contents
- **Courier bold**
Text in dialog boxes and protocols (records)

Characters and symbols

Special characters and symbols are used in these operating instructions.

Symbols for warnings and notes



Danger!

Special caution is required in this case. The warning triangle on the left indicates a danger of injury. If you do not observe this warning you may possibly be injured.



Important!

This symbol is used to warn the reader of situations involving a possible data loss, measuring errors, faults during a measuring run, collisions, or damage to the CMM or the workpiece.



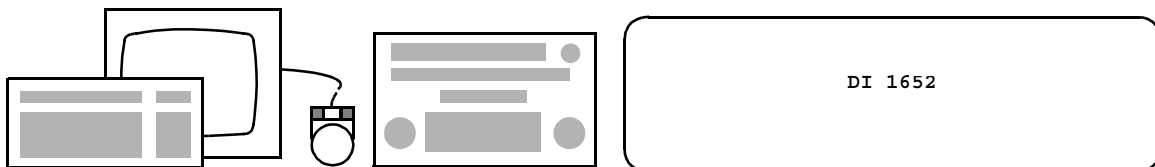
The **NOTE** symbol is placed next to important text passages and helpful additional information.

Symbol for function call-in

Several possibilities exist:

- Direct selection via a DI number
- Function selection via the pull-down menu
- Selection via a pictograph display

Example:



Symbol for softkey

References to softkeys in dialogs are displayed as follows.

Chapter overview

These operating instructions describe the function, handling and possible applications of the UMESS Opt.1 measuring program.

The following topics are covered:

- „Variable record“ on page 1-1
- „Plotting actual values“ on page 2-1
- „Outputting HPGL files“ on page 3-1

Table of contents

Conventions in these operating instructions	3
Typographical conventions	3
Characters and symbols	4
Chapter overview	5

Chapter 1 Variable record

Principle	1-2
The program file	1-3
The text files	1-3
Record file A	1-4
Files B to J	1-6
Creating the variable record	1-7
Procedure	1-7
Call and functions of <DI 1687>	1-8
Description of EQL (Easy Quality Language)	1-10
Variables	1-10
The WRITE command	1-11
DO loops	1-13
IF branchings	1-15
Program Structure	1-19
Changing the type size	1-19
Application examples	1-21
Application example 1	1-21
Record file A	1-22
Program file	1-22
Variable record	1-23
Application example 2	1-24
Record file A	1-25
Text file B	1-25
Text file C	1-26

Program file	1-26
Variable record.	1-27

Chapter 2 Plotting actual values

General	2-2
Procedure	2-3
Options for plot display	2-4
Plotting the results of nominal-actual (variance) comparisons 2-5	
Selection page	2-5
Reserving the nominal-actual table	2-6
Setting P1, P2 on the plotter	2-6
Setting the plotter scale	2-7
Nominal-actual table with measurement program	2-7
Nom/act table for DIN program	2-9

Chapter 3 Outputting HPGL files

Chapter

1

Variable record

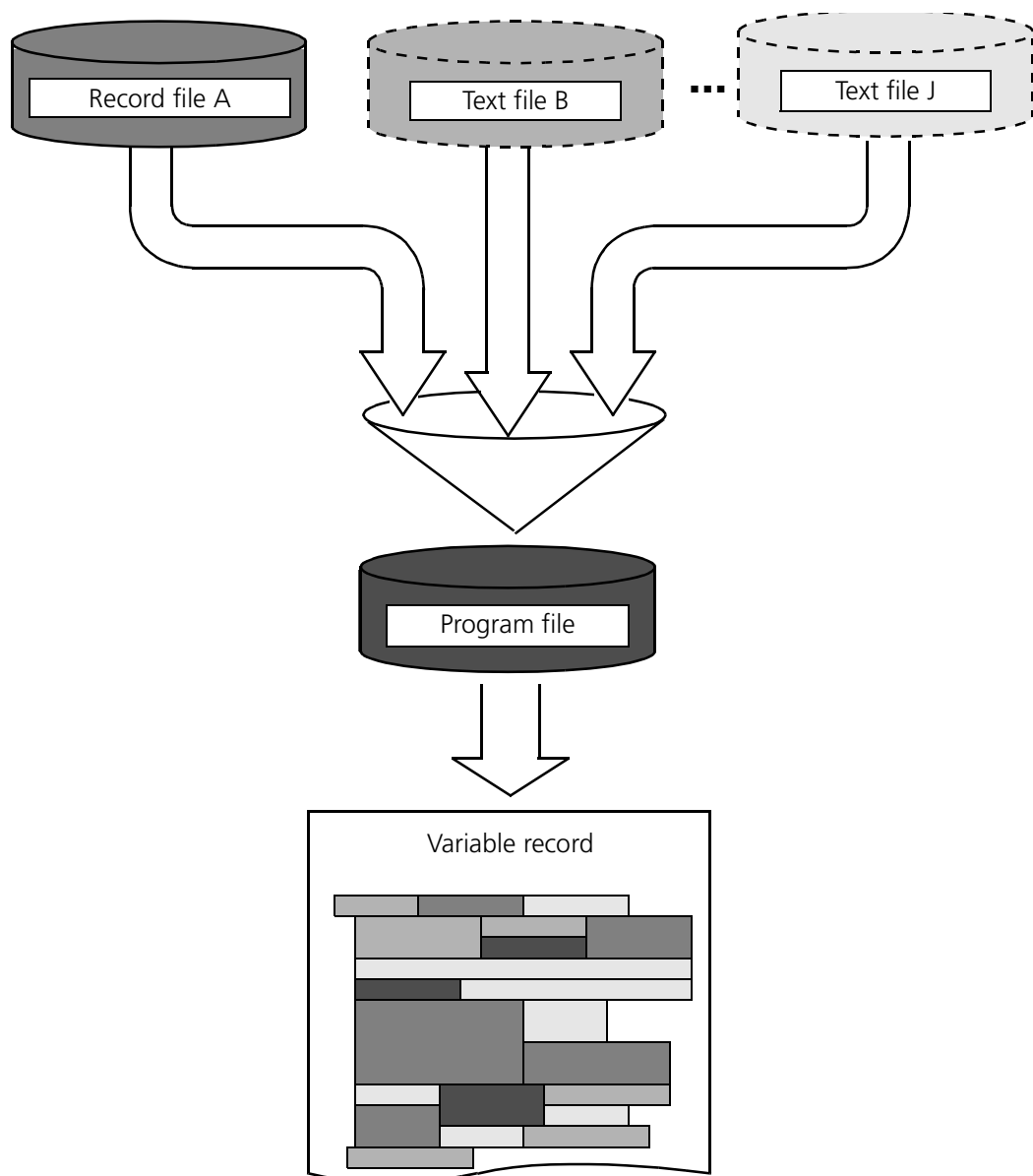
This chapter contains:

Principle.	1-2
The program file	1-3
The text files	1-3
Creating the variable record.	1-7
Description of EQL (Easy Quality Language)	1-10
Application examples.	1-21

Principle

In the variable record, text blocks are mixed from a max. of 10 text files (ASCII format) and subsequently output.

The rules according to which the text modules from the different ASCII files are interlinked are stored in the program file.



The program file

The program file contains the rules according to which the text modules of the different text files are interlinked. These rules are stored as an EQL (Easy Quality Language) program .

EQL) is a simple user language.

NOTE

- The program file with the EQL program must be created by the user in an ASCII editor.
- The name of the program file must comprise max. ten characters.
- An unlimited number of program files can be created.

The text files

The text files contain the text modules which make up the variable record (e.g. standard measurement record, test plan data, tables).

A distinction is made between two types of text files:

- Record file A
- Text files B, C, D, E, F, G, H, I and J.

Record file A

In general, record file A contains a UMESS measurement record in ASCII format.

Definition of line sections:

The lines of the text files are divided into sections to enable specific accessing of certain areas. These sections are marked with lower-case letters.

This division is specified with record file A.:

	Section	from character	to character
	a	1	4
*	b	6 (0)	9 (0)
*	c	11 (0)	17 (0)
*	d	19 (6)	23 (15)
	e	25	26
	f	28	37
	g	39	48
	h	50	56
	i	58	64
	j	66	72
	k	74	80
	l	1	19
	m	21	39
	n	41	59
	o	61	80
	p	1	17
	q	19	31
	r	33	47

*) with extended record output (10 characters)

Standard record

l	m	n	o
=====			
MEASUREMENT RECORD		UMESS	
Housing	MANUAL MEASUREMENT		
=====			
DRAWING NUMBER	ORDER NUMBER	SUPPLIER/CUSTOMER	OPERATION
			Manual measurement
OPERATOR	DATE	PART NO.	
	28.06.1994	2	
p	q	r	
=====			
a	b	c	d
e	f	g	h
i	j	k	
=====			
ADR	REC	TASK	IDF
SY	ACTUAL	NOMINAL	L.TOL
L.TOL	DEV	EXC	
=====			
123456789	123456789	123456789	123456789
123456789	123456789	123456789	123456789
10	20	30	40
50	60	70	80

Extended record

a	d	e	f	g	h	i	j	k
=====								
ADR	REC	TASK	IDF	SY	ACTUAL	NOMINAL	U.TOL	L.TOL
DEV	EXC							
=====								
123456789	123456789	123456789	123456789	123456789	123456789	123456789	123456789	123456789
10	20	30	40	50	60	70	80	

NOTE

- You can change between the column sectioning for standard record and extended record using **<DI 1667>**.
- The preset sectioning can be changed by following the procedure described below.

Files B to J

Files B to J contain random texts in ASCII format created with an ASCII editor.

Definition of line sections

Unlike the record file, you define the line sections here yourself.

Format

```
##DEF = letter (from character, to character).
```

Example

```
##DEF = a(1,4),b(5,8),c(10,25)
```

Column a contains characters 1 ... 4 of a line

Column b contains characters 5 ... 8 of a line

Column c contains characters 10 ... 25 of a line

If one line is not sufficient for a column definition, a new line must be created with "##DEF=".

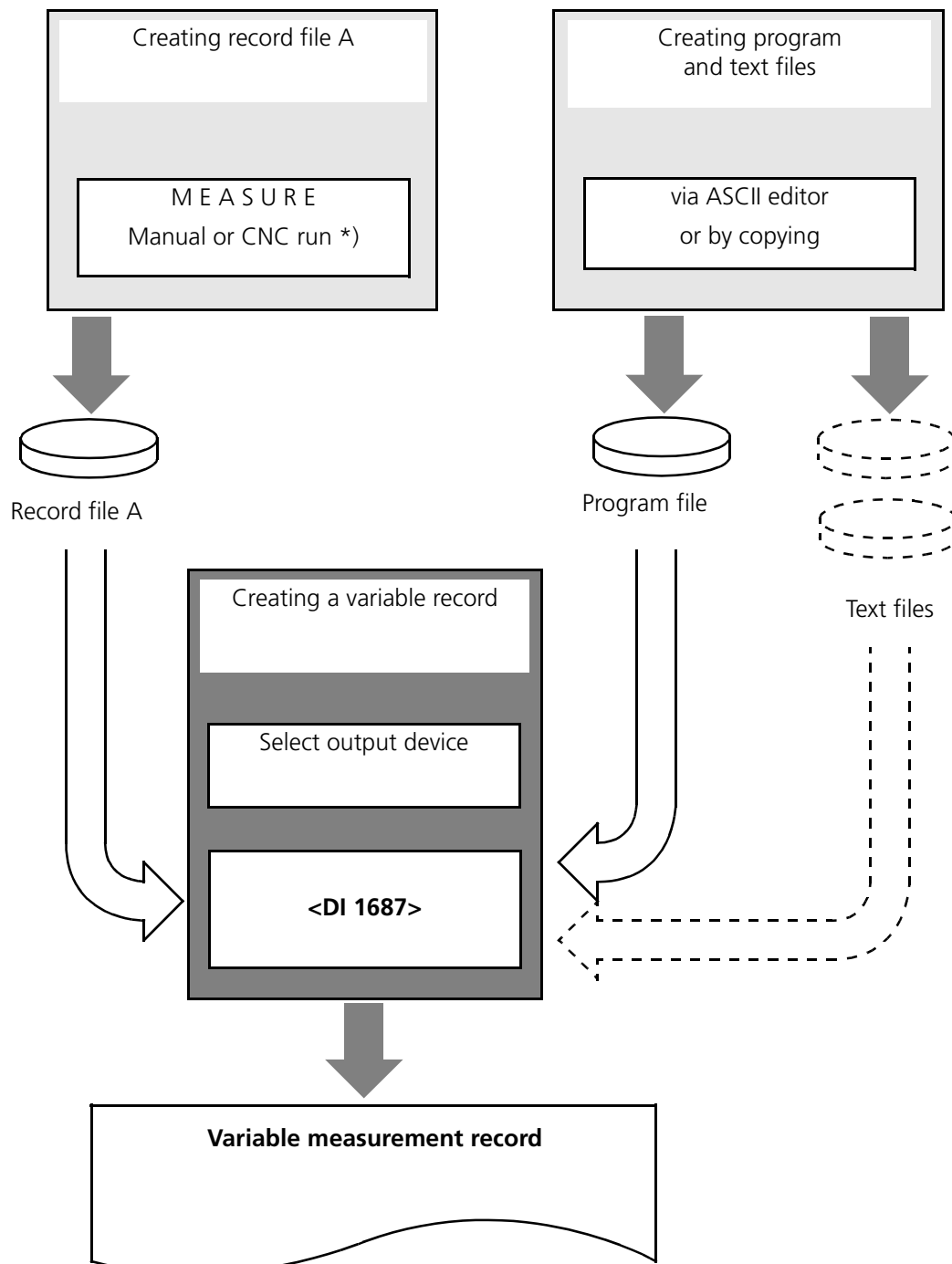
Example

```
##DEF= a(1,4),b(5,8),c(10,14),d(15,18),e(20,25),f(26,28),g(30,40)
```

```
##DEF= h(41,45),i(46,50),j(51,60),k(61,80)
```

Creating the variable record

Procedure



*) A measurement record can be saved with **<DI 1684>**. You can access files stored in this way only by specifying the file name.

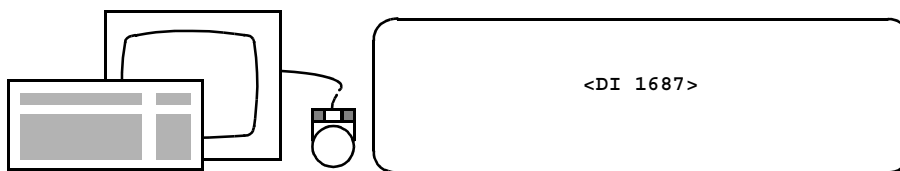
Call and functions of <DI 1687>

Creation of the variable record can be started by calling <DI 1687> in the MAN or PROG mode. The files required must be created before the function call.

Direct input <DI 1687> performs the following tasks:

- Definition of the program, record and text files
- Syntax check of the entire run and, where required, output of error information
- Selection of the output unit
- Execution of the program file

Function call



Input mask

Variable test record			
Output	Default output device ?	*	
	and / or file	?	File name = VPA B.
Control with program file	VPPVDAP	B.	
Meas. record			
current	?	*	
or saved record			File name = B.
Text files			
B= VPTVDAB	B	C= VPTVDAC	B
E= VPT	B	F= VPT	B
H= VPT	B	I= VPT	B
D= VPT	B.	G= VPT	B.
J= VPT	B.		
* YES	NO	1. PATT	* TERMIN
BACK			INFO

1. PATT

The names of the following standards are entered in the input mask:

VPPVDAP_____B for program file

VPTVDAB_____B for text file B

VPTVDAC_____B for text file C

The standards are supplied with the program.

How to create the program and text files

You must have good knowledge of the operating system and the vi editor before using this program.

NOTE

- All of the required program and text files must be available in the **/home/zeiss/UC** directory and the record files must be stored in the **UF** directory.
- The file name of each file must always have 14 characters.

The first three characters and the last character of the file name for the program file, the text file and the output file are preset. The remaining 10 characters of the file name may be selected freely by the user:

Program file: **VPP_____B**

Text file: **VPT_____B**

Output file: **VPA_____B**

The first two characters and the last character of the file name are preset for the record file. The places marked by **xxx** must be replaced by the workpiece number and the places marked by **yyyyy** are allocated to the part number.

Record file: **PD_xxx_yyyyy_B**

Description of EQL (Easy Quality Language)

NOTE

All commands are written in large letters!

Dual-arm mode

Start column 2 first, and then column 1.

Wait until all windows have appeared.

If necessary, adjust the size and layout of the windows to suit your needs.

Variables

In EQL "A" to "Z" are allowed as variables. They can take the place of fixed numeric values in statements.

Before a variable is used in a statement, a defined value must first be assigned to it. This value can be changed within the program later on.

Value assignment

A value can either be assigned directly as a number or as simple addition or subtraction.

Example

X=10 Variable X is assigned the value 10
Y=X+5 Variable Y is assigned the value 15
Z=Z+1 The value of variable Z is increased by 1

Multiplication and division are also possible.

Example:

A=5*B
Y=Z:2 Integer division (example: 3:2=1)

Since the variables can only represent integers (whole numbers), only the integer part of the variable is assigned for division.

If a division by zero occurs, the program is canceled and the corresponding error message appears.

Within a DO loop (► „DO loops“ on page 1-13) the loop counter (!) can also be included in the assignment.

Example

```
DO 1,10
```

$Z=I+1$ Variable Z is assigned the value of the loop counter plus 1 for each loop cycle.

```
.  
ENDDO
```

The following value assignments are ***not permitted***:

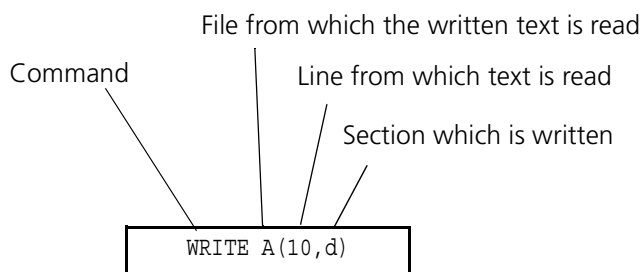
$X=Y+5-2$ More than one arithmetical operation

$Z=I+1$ Loop counter outside of a DO loop

The WRITE command

The WRITE command outputs sections of the text files or individual characters in the variable record.

Format



Examples of WRITE commands which can access text files A, B and C

WRITE a(10,d)

Section d of line 10 is read from record file A and output in the variable record.

WRITE B(17)

Line 17 is read from text file b and output in the variable record.

WRITE C(4/7)

Lines 4 to 7 are read from file C and output one beneath the other in the variable record.

WRITE A(3/END)

All lines are read starting with line 3 and output one beneath the other in the variable record.

WRITE B(1/5,a)

Sections a of lines 1 to 5 are read from file B and output in succession in the variable record.

WRITE B(2,a/d)

Columns a to d of line 2 are read from file B and output in succession in the variable record.

Instead of the line numbers variables (➤ „Variables“ on page 1-10) and simple arithmetic operations can be included directly in the **WRITE** command, e.g.:

```
WRITE A(X,c)
```

```
WRITE B(Y+3,j)
```

```
WRITE C(Z-!,d)
```

Examples of WRITE commands which output texts directly in the variable record

WRITE "*" "

Output: *

WRITE 5 "*" "

Output: *****

WRITE 80 " "

A blank line (80 blanks) is output.

No variables may be used for the number of repetitions.

WRITE [42]

The ASCII value 42 is output: *

Values from 0 to 255 are permitted. No repetition factor is permitted!

WRITE [12],[65],[66],[67]

An ASCII character can be entered in square brackets.

Output in this example following a page feed: **ABC**

Several WRITE statements can be combined in a single line (separated by commas):

**WRITE
A(2,f),4"***",B(7,a),4"***",
Nominal"**

The following line is printed (if $A(2,f) = 2.458$ and $B(7,a) = 5.00$):

2.458*****5.00*****Nominal

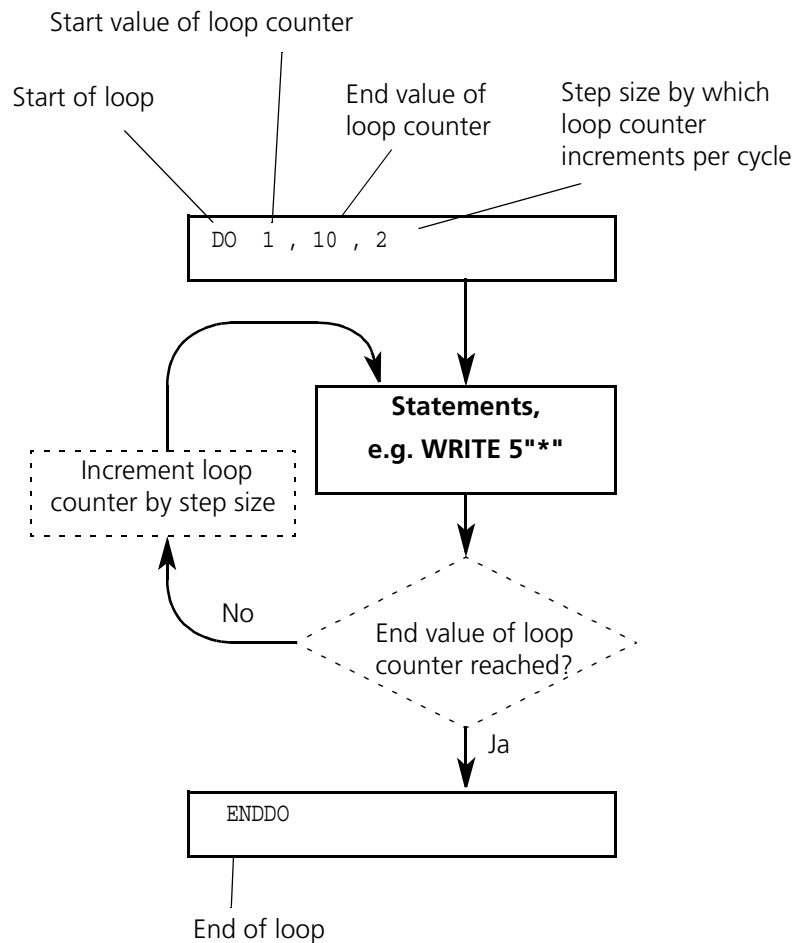
If a WRITE command is too long for a line, a continuation line must be tagged with the "\$" character. Example:

**WRITE C(2,h),B(36,c),"Form deviation",A(26,g),"!!!"
\$ B(2,c/d),4"***"**

DO loops

In DO loops, the execution of a statement is repeated until a "loop counter" reaches a defined value.

Structure of a DO loop



NOTE

- A maximum of three DO loops can be nested.
- The loop counter (!) can be used as a variable in statements within the loop.
- If DO loops are nested, the loop counter as shown as follows:
 - !!!: innermost DO loop
 - !!: second innermost DO loop
 - !: third innermost (outside) DO-Schleife
- A step size of 1 need not be specified.

Examples

```
DO 1,5,2
  WRITE "Line"
ENDDO
```

The word **Line** is output three times one below the other.

```
DO 3,7,2
  WRITE A(!,d)
ENDDO
```

The "**d**" sections of lines 3, 5 and 7 are read from text file A and output below one other.

```
DO 5,8
  WRITE C(!)
ENDDO
```

This command has the same meaning as **WRITE C(5/8)**

"**END**" can also be used as a loop end value:

```
DO 1,END(A),2
  WRITE B(!)
ENDO
```

This command outputs text file B until the loop counter reaches the line number of text file A.

IF branchings

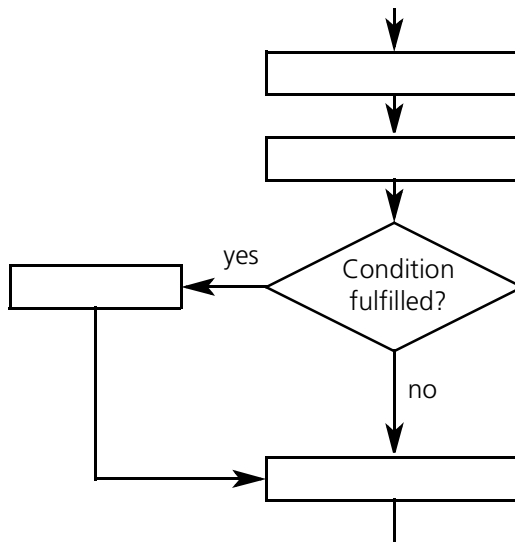
IF branching can be used to formulate a condition under which defined statements will be executed.

Types of IF branching

Simple IF branching

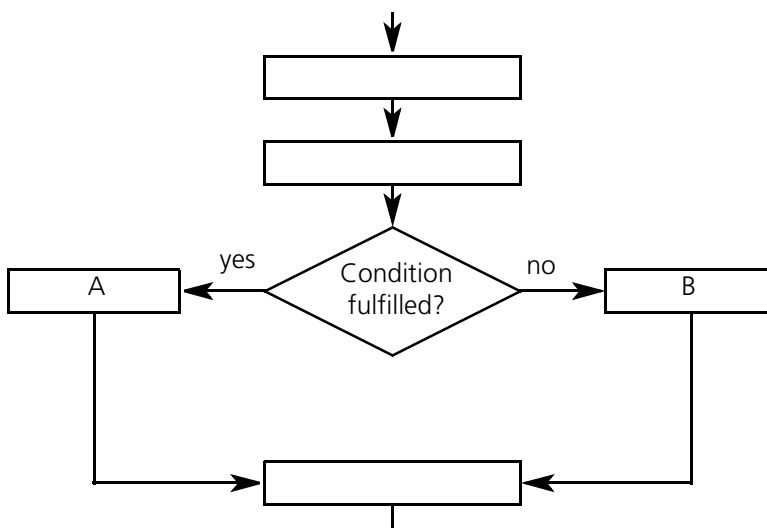
If a condition has been fulfilled, a subprogram is processed.

If the condition is not fulfilled, the main program is continued.



IF-ELSE branching

If a condition is fulfilled, subprogram A is processed; if the condition is not fulfilled, subprogram B is processed. The main program is then continued.



Conditions

The following conditions are possible in IF branchings:

- Two text sections with the same contents,
e.g. **IF A(3,b) = B(!,c)**
- Number e.g. **IF A(!,b) = (0.5)**
- Text e.g. **IF A(!,b) = "BO/1"**

You can also check numeric variable values with **IF**. The following relational operations are possible:

- = Check for equal
- <> Check for unequal
- < Check for less than
- <= Check for less than or equal to
- > Check for greater than
- >= Check for greater than or equal to

Examples

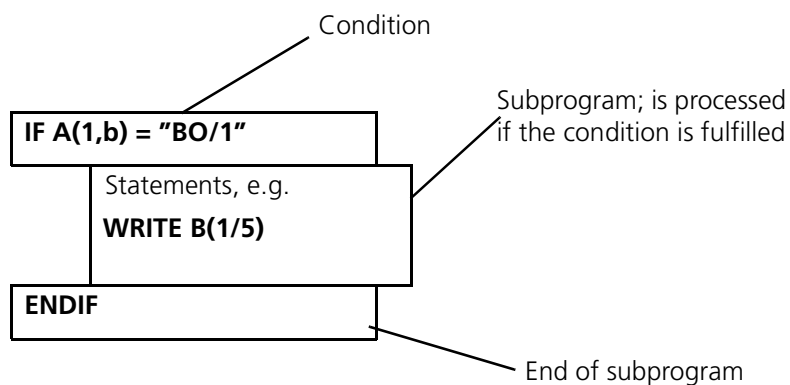
IFC = 3*B

IF! >= 60

IFA <> D:E

Structure of IF branchings

Simple IF branching



Example

IF A(1,b) = "Abwe"

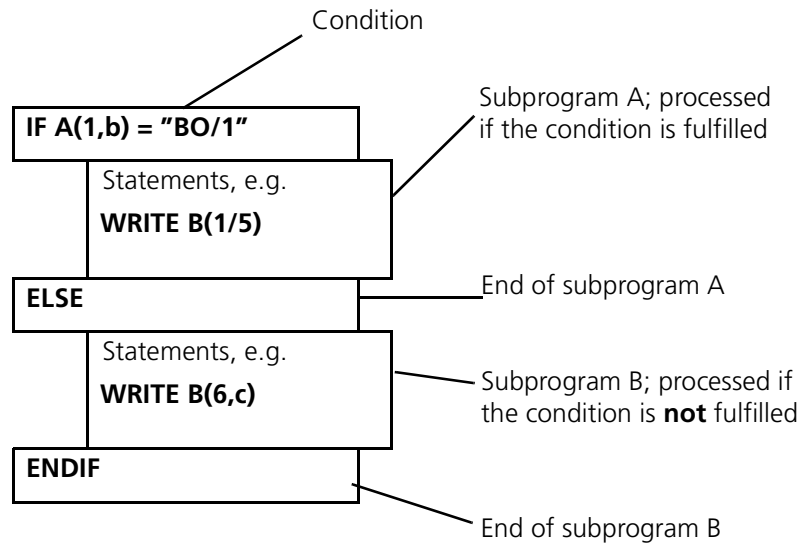
WRITE "***"**

ENDIF

Output

Condition fulfilled: *********

Condition not fulfilled: No output

IF-ELSE branching:**Example**

```
IF A(1,b) = "Abwe"
  WRITE "*****"
ELSE
  WRITE "_____"
ENDIF
```

Output: Condition fulfilled: *****

Condition not fulfilled: _____

Jokers

In EQL the "&" and "@" signs are used as jokers (or wild cards) which can be used to formulate variable IF conditions.

The & sign can be used to replace any single character. It also can be placed anywhere in the text.

Examples

```
IF A(!,b) = "BO&&"
```

The condition is fulfilled for all texts which are four characters long and begin with "BO".

```
IF A(!,b) = "&&&der"
```

The condition is fulfilled for all texts which end in "der" and have a length of six characters.

The @ sign can be used to replace a character string of any length. It may be placed only at the end of a text.

Examples

```
IF B(!,d) = "FL@"
```

The condition is fulfilled for all texts which begin with "FL".

IF C(!,f)

Not permitted!

The @- sign in brackets can be used to replace any number.

Example**IF A(!,b) = (@)**

The condition is fulfilled if there is a number in text section A(!,b).

Application: Used to check whether a tolerance has been exceeded.

The & and @ signs can be combined.

Example**IF A(!!) = "&&CIRCLE@"**

The condition is fulfilled for all texts of any length in which the word "CIRCLE" is in the third position.

NOTE

- Exactly one line section must be completely defined in conditions. For example, the following conditions are not permitted:
IF A(3) = ... Line segment not specified !
IF B(1/3,a) = ... Multiple lines specified.
IF A(5,a/c) = ... Multiple line sections specified.
- All characters located between quotation marks, i.e. including blanks, will be evaluated as text characters. If there are blanks in the column, they must also be entered between quotation marks for IF conditions.

Example

Section A(2,c) contains the characters "**Form deviation**"

IF A(2,c) = "Form@"

WRITE "***"**

No output

ENDIF

IF A(2,c) = " Form@"

WRITE "***"**

Output: *****

ENDIF

Program Structure

Comment lines

Program lines which start with the "%" character are not taken into consideration when the program is executed. The programmer therefore has the option of entering comments.

Examples

```
%Access to file C
% Start of program section II
```

Blanks

Blanks located outside of quotation marks are read over. This makes it possible e.g. to structure DO loops and IF branchings.

Example

```
DO(1,5)
  WRITE 5"*"
  IF A(!,a) = (0.5)
    DO X,X+4
      WRITE "!!!!!"
    ENDO
  ENDIF
ENDDO
```

Changing the type size

Three different type sizes can be selected for printout of the variable record head:

- Compressed (compressed)
- Normal (normal)
- Expanded (expanded)

You can control the printer as follows:

MODE COMPRESSED = ON

The printer is switched to compressed type

MODE COMPRESSED = OFF

The printer is switched back to normal.

The mode setting always applies to a complete line.

The computer can also be set to the required type size via **escape sequences**. You must always type in the latter before the actual text.

Escape sequences are started by pressing the <ESC> key. Activation of the <ESC> key is displayed on the screen by **EC**.

The following escape sequences can be used for special applications with HP printers:

"EC&k0S" (normal)

"EC&k1S" (expanded)

"EC&k2S" (compressed)

NOTE

- An escape sequence may be located either in a program file or in a text file.
- A type size remains valid until it is changed by a new escape sequence.
- A output is made to the screen, the escape sequences will also be output, i.e. the type size will not be changed.
- The number of characters per line in the printout output depends on the type size:
 - Normal 80 characters
 - Expanded 40 characters
 - Compressed 124 characters
- If a uniform type size is required for the entire record, it can be specified with **<DI 1662>**, **<REC OUTP>**.

Example

WRITE"EC&k2ScompressedEC&k0S"

WRITE"normal"

WRITE"EC&k1SexpandedEC&k0S"

Printer output

Compressed

Normal

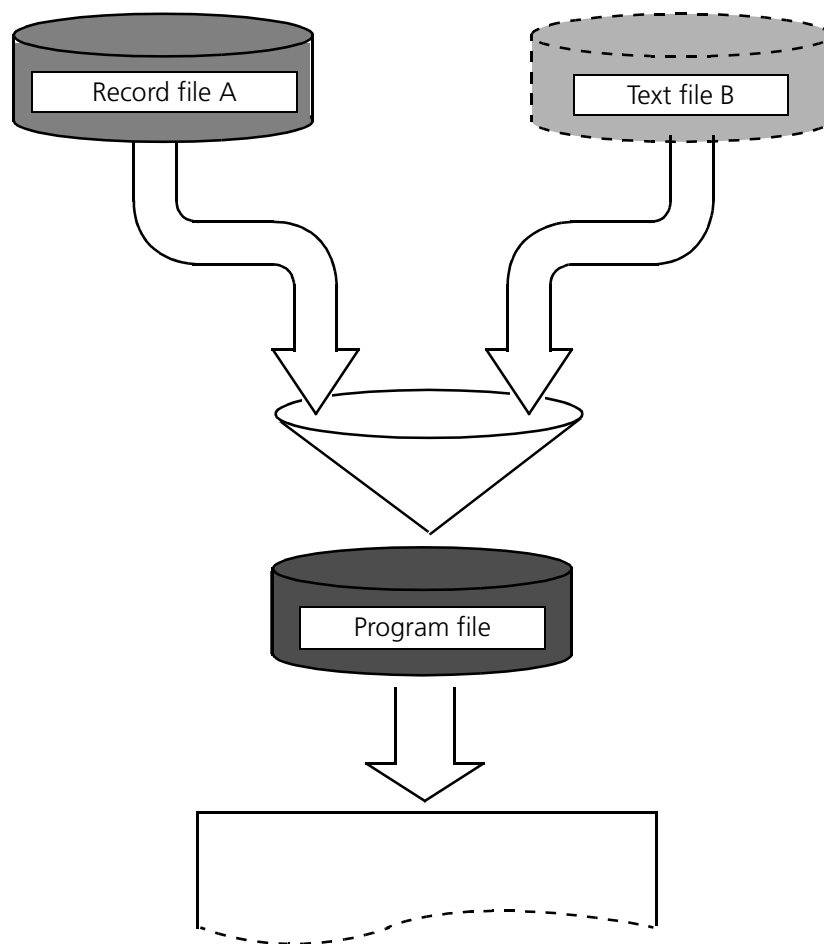
Expanded

Application examples

Application example 1

In this application example a variable record is produced from:

- the measurement results (record file A)
- the inputs for the record structure (text file B)



Record file A

MEASUREMENT RECORD										UMESS	
U M E S S - Test RUN								CNC run			
DRAWING NUMBER		ORDER NUMBER		SUPPLIER/CUSTOMER				OPERATION			
-----		-----		LCV				-----			
OPERATOR		DATE		PART NO.							
MUELLER				1							
=====											
ADR	REC	TASK	IDF	SY	ACTUAL	NOMINAL	U. TOL	L.TOL	DEV	EXC	
=====											
10		CIRCLE I	K1Z	Z	-54.092	53.600	.150	-.200	.492	.342	
			K1Y	Y	32.423	32.000	-.008	-.200	.423	.437	
			K1D	D	29.694	30.500	.500	-.500	-.805	-.305	
		S/MIN/MAX			.008	(3)	-.006	(4)	.006		
11		SURFACE F1X	X	X	-40.204	40.000	.200	-.200	.204	.004	
		F1W1	W1		.029	0.000	.100	-.100	.029	++	
		F1W2	W2		.231	0.000	.100	-.100	.231	.131	

Program file

```
1 WRITE A(1/14)
2 WRITE A(15,d),3" ","|",A(15,g),3" ","|",A(15,f),3" ","|",A(15,k),3" ","|",
  "COMMENT"
3 WRITE A(16)

4 DO 17,END(A)
5 IF A(!,k)=(@)
6 WRITE A(!,d),3" ","|",A(!,g),3" ","|",A(!,f),3" ","|",A(!,k),3" ","|",
7 WRITE 8" ","|",13" ","|",13" ","|",10" ","|",31"_"
8 ENDIF
9 ENDDO

10 WRITE 80" "
11 WRITE "EXTENT OF OUTPUT: FEATURES OUT OF TOLERANCE"
12 WRITE A(16)

13 END
```

Line(s)	Explanation of program file
1, 3, 12	Lines 1 to 14 and 16 are transferred from record file A.
2	Transfer of column designations IDF, NOMINAL, ACTUAL and EXC.

Line(s)	Explanation of program file
4, 9	Record file A is searched through starting with line 17.
5, 8	Only lines with exceeded tolerances are transferred.
6	The preselected columns are transferred from a record line with an exceeded tolerance.
10	Blank line

Variable record

```

=====
                        MEASUREMENT RECORD   ZEISS  UMESS
U M E S S - Test RUN                                CNC run
DRAWING NUMBER      ORDER NUMBER:      SUPPLIER/CUSTOMER:  OPERATION
-----            -----            CARL ZEISS            -----
OPERATOR            DATE      PART NO.      |
MUELLER             |          1          |

=====
IDF      |NOMINAL      |ACTUAL      |EXC      |COMMENT
=====
K1Z      |53.600      |-54.092     |.342     |
K1Y      |32.000      |32.423      |.437     |
K1D      |30.500      |29.694      |-.305     |
F1X      |40.000      |-40.204     |.004     |
F1W1     |0.000      |.029        |.131     |

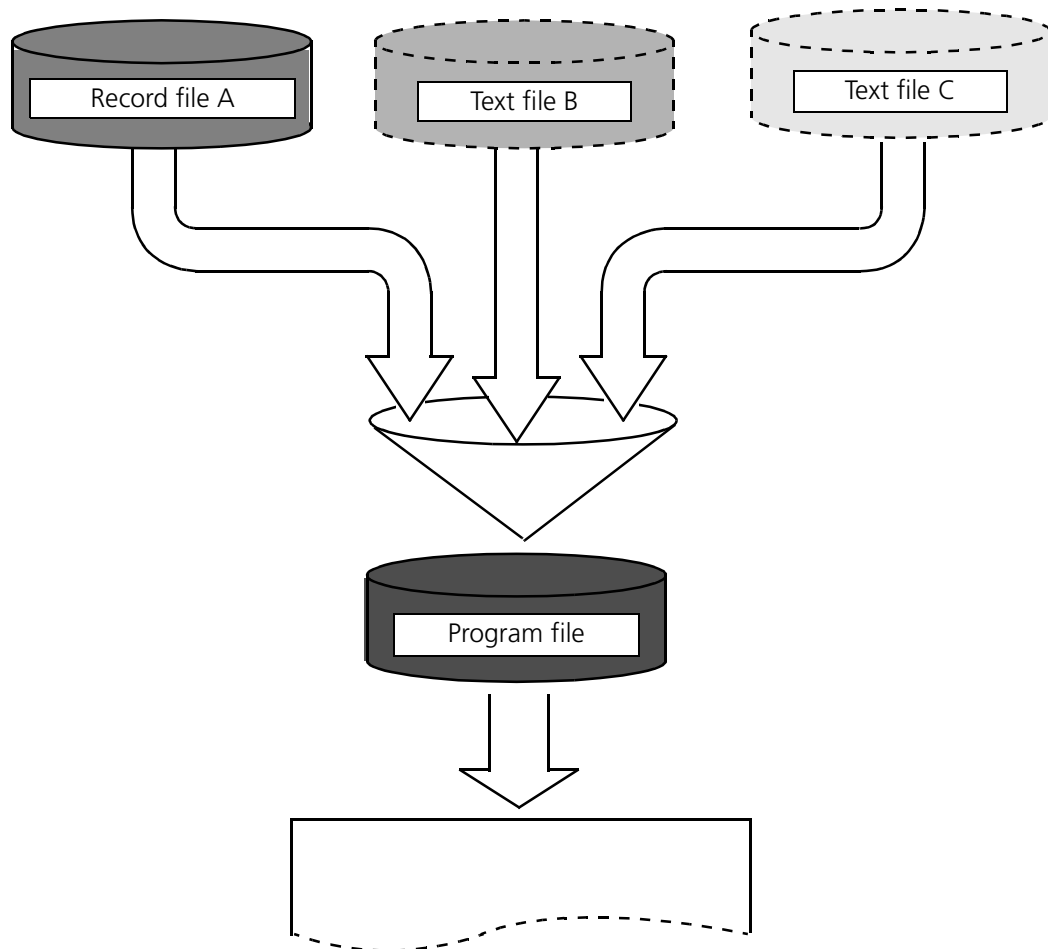
EXTENT OF OUTPUT: FEATURES OUT OF TOLERANCE
=====

```

Application example 2

In this application example, a variable record is made up of:

- The measurement results (record file A)
- The inputs on the record structure (text file B)
- The customer-specific comments (text file C)



Record file A

```

=====
                        MEASUREMENT RECORD                        UMESS

U M E S S - Test RUN                                           CNC run
DRAWING NO      | ORDER NUMBER      | SUPPLIER/CUSTOMER  | OPERATION
-----        | -----        | LCV                | -----

OPERATOR        | DATE          | PART NO.          |
MUELLER         |               | 1                 |

=====
ADR|REC|TASK|IDF|SY|ACTUAL  |NOMINAL  |U. TOL|L.TOL|DEV  |EXC
=====
  10      CIRCLE I K1Z  Z  -54.092    53.600    .150   -.200   .492   .342
                K1Y  Y   32.423     32.000   -.008   -.200   .423   .437
                K1D  D   29.694     30.500    .500   -.500   -.805  -.305
          S/MIN/MAX                .008         (3)   -.006   (4)   .006
  11      SURFACE F1X  X  -40.204     40.000    .200   -.200   .204   .004
                F1W1 W1   .029       0.000    .100   -.100   .029   ++
                F1W2 W2   .231       0.000    .100   -.100   .231   .131

```

Text file B

```

=====
S3=2120 | b=13.970|Q1=49.015| FB=6000 N | K=1.3x45' | MA=40Nm
##DEF=  a(1,5),b(6,31),c(32,43),d(1,71)

a | b | c
K1Z | Z coord. of fixture | Q1=53.600 |
K1Y | Y coord. of fixture | 32.000 |
K1D | Diameter of fixture | 30.500 |
A | Distance | 45 | 85 |
b | Width of fixture | 18.970 |
H | Height of fixture | 100 | 150 |
F1X | X ref. surface bore | 40.000 |
F1W1 | W1 of ref. surface, bore | 0.000 |
F1W2 | W2 of ref. surface, bore | 0.000 |
HP1 | Hardness of fixture | HRC 61 |
HP2 | Hardness of fixture | HRC 61 |
1 | Periphery | Q1*PI/2 |
2 | F1W1 and F1W2 |
3 | Elastic change of H1 | H1*FB/(b*5*10↑6) |
4 | Elastic indents (stop gauge) | FB*3*10↑-5/(S3*b) |

```

Text file C

```

=====
                        TEST AND CALCULATION SHEET
=====

COMPANY          | DRAWING NUMBER    | TRANSFER NO.:     | ORDER NO.:
-----|-----|-----|-----|-----
a      |          b        |          c         |          d         |          e
          f          |          g          |
##DEF=  a(1,5),b(7,31),c(33,42),d(44,53),e(55,80),f(1,31),g(33,53)
=====
IDF |          FEATURE          | NOMINAL | ACTUAL |          REMARKS
=====
All values in mm          | Date          |          Operator:
=====

```

Program file

```

1 WRITE C(1/5)
2 WRITE A(5,n),"|",A(5,1),"|",19" ", "|",A(5,m)
3 WRITE C(4)
4 WRITE B(2)
5 WRITE C(10/12)

6 DO 5,19
7     DO 1,END(A)
8         IF A(!!,d)=B(!,a)
9             WRITE C(6)
10            WRITE B(!,a/c),A(!!,f)
11        ENDIF
12    ENDDO
13 ENDDO

14 WRITE C(12/14)
15 END

```

Line(s)	Explanation
1, 3, 5, 14	Lines 1 - 5, 4, 10 - 12 and 12 - 14 are transferred from text file C.
2	Transfer of column designations IDF, FEATURE, NOMINAL, ACTUAL and REMARKS.
4, 9	Record file A is searched through starting with line 12.
5, 8	Only lines with exceeded tolerances are transferred.
6	The preselected columns are transferred from a record line with exceeded tolerance.
10	Blank line

Variable record

```

=====
                        TEST AND CALCULATION SHEET
=====

COMPANY      | DRAWING NUMBER | TRANSFER NO.: | ORDER NO.:
LCV          | -----      | -----      | -----

S3=2120 | b=13.970 | Q1=49.015 | FB=6000 N | K=1.3x45' | MA=40Nm
=====
IDF | FEATURE | NOMINAL | ACTUAL | REMARKS
=====
K1Z | Z coord. of fixture | Q1=53.600 | -54.092 |
-----
K1Y | Y coord. of fixture | 32.000 | 32.423 |
-----
K1D | Diameter of fixture | 30.500 | 29.694 |
-----
F1X | X ref. surface bore | 40.000 | -40.204 |
-----
F1W1 | W1 of ref. surface bore | 0.000 | .029 |
-----
F1W2 | W2 of ref. surface bore | 0.000 | .231 |
=====
All values in mm | Date | Operator:
=====

```


Chapter

2

Plotting actual values

This chapter contains:

General	2-2
Procedure	2-3
Options for plot display	2-4
Plotting the results of nominal-actual (variance) comparisons	2-5

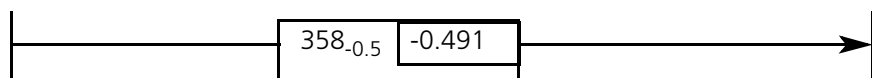
General

With UMESS GRAFICS you can plot the results from variance (nom.-act.) comparisons on preprinted forms.

- The actual values and their deviation from the nominal values can be plotted on the preprinted forms as an actual value table:

IDF	SY	NOM	TOL.	ACT	DEV
B1	Y	300	±0.5	300.059	0.059
	Z	300	±0.5	-299.978	-0.022
	D	44 ^{H7}	+0.025	44.030	0.030

- Deviations can be assigned directly to the drawing size:



- Values within tolerance are clearly differentiated from those exceeding tolerance by an automatic change in color.
- The program is divided into 3 applications:
 - Measurement programs (e.g. **<CIRCLE>**, **<DISTANCE>**, **<INTERSEC>** etc.).
 - DIN programs without MMC (all form and position programs).
 - DIN programs with MMC (e.g. **POS**, **KON**, **KOA**, **REC**).

With DIN programs only the actual values are plotted. The DEV column shows if a tolerance has been exceeded.
For DIN evaluation with MMC, the plotter output is extended by the tolerance specified in the **TOL** column.
- The program always refers to the last record address. In this way it automatically recognizes the appropriate application.
- The display offers the symbols for selection. The parameters defined in the dialog can be assigned as required for each of the symbols selected.
- 7 variance comparisons can be plotted per measured element.

Procedure

- Preparing the printed form
All texts plotted during the CNC run prolong the measurement time. Once the measurement run has been determined, you can print a form containing as many preprinted designations and texts as possible based on the existing test plan.
- Adjusting the plot format
Plotter 2 is the default device for input of **P1** and **P2** (device no. 6, 7 or 8 for **<DI 1625>**). The plotter must be connected directly to the computer.
- Plotter scale
The scale may be different in the x and y direction and is entered in mm.

Options for plot display

The values plotted depend on the type of program involved.

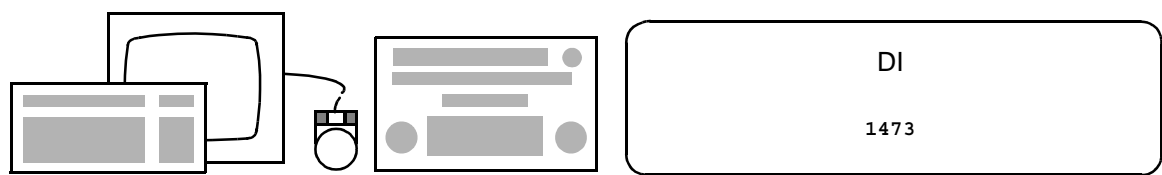
The following table provides an overview. Please note that for DIN programs the deviations correspond to the actual values. The numbers are plotted left-justified. A position is automatically reserved for the +/- sign.

Program type	Tolerance	Distance	Plotted values, pen number		
Measurement program		>0		Actual, P	Deviation, P+1/P+2
		=0		Deviation, P+1/P+2	
DIN program without MMC	met	>0		Actual tol., P	
		=0		Deviation, P+1	
	not met	>0		Actual tol., P	Exceeded P+2
		=0		Deviation = Actual tol.,	
DIN program with MMC	met	>0	Tmmc, P	Actual tol., P Actual value,P	Deviation, P+1/P+2
		=0		Deviation, P+1/P+2	
	not met	>0	Tmmc, P	Actual tol., P Actual value,P	Exceeded, P+2 Dev., P+1/P+2
		=0		Deviation, P+1/P+2	

Plotting the results of nominal-actual (variance) comparisons

Selection page

Function call



Input mask

UMESS Graphics: Nom-act table

☐ J Nom-act table
or reserve nom-act table
or set P1, P2 on the plotter
or plotter scale

☐

* YES	NO			*				TERMIN
BACK								INFO

Explanation of the input fields

Nom-act table

This function can be called only after P1 and P2 have been set on the plotter and the plotter has been set to scale.

or reserve nom-act table

The control data lines are reserved in the programming mode. The plot positions can be entered later with the control data correction.

or set P1, P2 on the plotter

The corner points of the required plot format must be set on the printer. This only has to be done once; then set the plotter to scale.

or plotter scale

Input of scale in mm. The scale may be different in the X and Y axes.

Reserving the nominal-actual table

Input mask

UMESS Graphics: Reserve lines for nom-act table

I

No. of nominals

* YES

NO

*

TERMIN

BACK

INFO

Setting P1, P2 on the plotter

Input mask

UMESS Graphics: Set P1, P2 on the plotter

J

Set P1 and P2 on the plotter

* YES

NO

*

TERMIN

BACK

INFO

As soon as this input mask appears, you can set P1 and P2 on the plotter. The values set are accepted by pressing <TERMIN>.

Setting the plotter scale

Input mask

UMESS Graphics: Plotter scaling

D

Xmin

Xmax

Ymin

Ymax

* YES

NO

*

TERMIN

BACK

INFO

NOTE

The plotter queue must be empty and the plotter must not be in use when this function is started.

Nominal-actual table with measurement program

Input mask

UMESS Graphics: Nom-act table

I

Pen number

No. of decimal places

Character size

?

Symbol

Position x

Position y

Distance

Type direction

X

Y

Z

D

W1

W2

Wk

R

* YES

NO

OUTP. MM

*

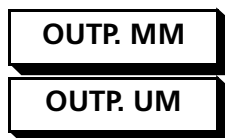
OUTP. UM

TERMIN

BACK

INFO

Softkeys



You can select plotting of the nom/act values in millimeters **<OUTP. MM>** or in micrometers **<OUTP. UM>** by actuating the corresponding softkey. The setting currently selected is the one displayed on the inactive softkey, i.e. only the softkey available for changing the setting is active.

Explanation of input fields

Pen number	The color is changed automatically. The pen number entered refers to the actual value. Deviations are automatically plotted by the pen with the next highest number. The pen number is increased by 2 if the tolerances are exceeded.
Type size	The type size is entered in mm. The type sizes agree with the input values for paper sizes up to DIN A3. The scale and type size are automatically adapted during a CNC run with any other format.
Position X, Y	Plot position of the actual value.
Distance	Distance between actual value and deviation. If the distance = 0 only the deviation will be plotted (► „Procedure“ on page 2-3 - Table of plotted values).
Type direction	The type direction is entered in degrees. Possible values range from -180 to +180 degrees.
Symbols	The symbols listed can be plotted.

Nom/act table for DIN program

Input mask

UMESS Graphics: Nom-act table

I

Pen number

No. of decimal places

Character size

?	Symbol	Position x	Position y	Distance	Type direction
	t				
	tx				
	ty				
	tz				
	td				
	D1				
	D2				

* YES

NO

OUTP. MM

*

OUTP. UM

TERMIN

BACK

Chapter

3

Outputting HPGL files

The results of measurements are easier to interpret if they are displayed together with graphics. One way this can be done is to transfer graphics from a CAD system. The graphics file must contain the graphics to be displayed in their final form and position, since they can not be modified later on.

The graphics file to be output must be stored in the **/home/zeiss/UB** directory. It also must be named according to the following specification:

HGLxxxxxxxxxB

HGL = prefix

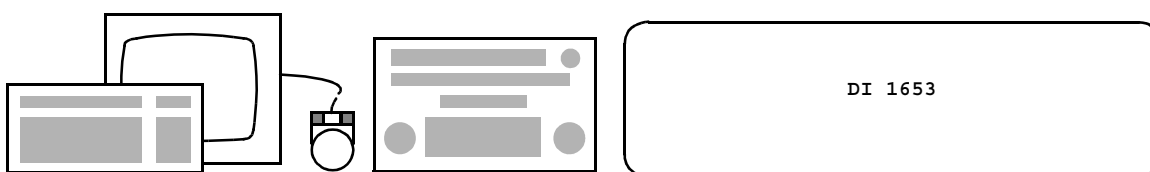
xxxxxxxxxx = file name (=10 characters)

B = suffix

If less than 10 characters are required for the file name underlines "_" must be used for the remaining positions.

These files can be output only to the HP plotter, the record (log) printer, or the Laserjet II in the HPGL mode. Output to the graphics window is not possible.

Function call



Input mask

Output of a graphics file (HPGL format)

File code: CUBE Device code: 4 Enlarge file ? *

* YES	NO			*				TERMIN
BACK								INFO

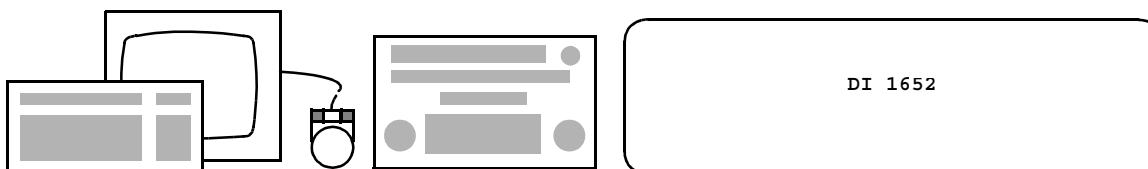
Explanations

- File code** Specification of name under which file is stored in the **/home/zeiss/UB** directory.
- Device code** Number of output device activated with **<DI 1625>**.
- Enlarge file?**
- **<YES>**
If several graphic files in a command block are to be output in succession. The following files are added to the intermediate storage.
 - **<NO>**
If there are several files within a command block, the intermediate storage is overwritten by the following files.

Output to a plotter or graphics-capable printer

Before outputting a graphics file to your plotter or graphics-capable printer, you must execute a **Paper change** by entering **<DI 1652>** after entering **Output HPGL files (DI 1653)**.

Function call



Index

A

Adjusting the plot format 2-3
ASCII editor 1-3
ASCII files 1-2
ASCII format 1-2

B

Branching 1-15

C

CAD system 3-1
Conditions 1-16

D

Definition of line sections 1-4, 1-6
DI 1687 1-8
DIN programs with MMC 2-2
DIN programs without MMC 2-2
Directory /home/zeiss/CZ_MES_UB 3-1
Distance between actual value and deviation 2-8
DO 1-13

E

EQL 1-10
EQL (Easy Quality Language) 1-3
EQL program 1-3
Escape sequences 1-19
Exceeding of tolerance 2-2

F

File name 3-1

G

Graphics file 3-1

H

HPGL file 3-1

I

IF 1-15

L

Loops 1-13

M

Measurement programs 2-2

N

Nom/act table for DIN program 2-9
Nominal-actual (variance) table 2-2

O

Output of a graphics file 3-2

P

Pen number 2-8
Plot display options 2-4
Plot position 2-8
Plotter scale 2-5
Plotting results of nominal-actual comparisons 2-5
Program file 1-3, 1-22

R

Record file 1-4, 1-22
Reserving the nominal-actual table 2-6

S

Selection page 2-5
Setting P1, P2 on the plotter 2-5, 2-6
Setting the plotter scale 2-7
Structuring 1-19
Symbols 2-8

T

Text blocks 1-2
Text files 1-2, 1-3
Type size 1-19
Typographical conventions 3

U

UMESS GRAPHICS 2-2

V

Value assignment 1-10
Variables 1-10

W

WRITE command 1-11

X

X, Y position 2-8