

FILE COMPARE

Reference Manual

ND-60.184.4 EN

THE PRODUCT

This manual describes the product

FILE-COMPARE ND-210603B

FILE-COMPARE compares two sets of data on mass storage, and irrespective of the data format and record organization, it will describe the differences between them.

THE READER

This manual is intended for all users of ND-500 computers who need to compare two files to find discrepancies between them.

PREREQUISITE KNOWLEDGE

The reader should have a basic knowledge of general data-processing techniques and of the use of SINTRAN file systems.

THE MANUAL

This manual is a Reference manual. It provides a description of the product, the commands, and some examples of the ways in which the product may be used. There is also a section which describes how to start and use only the simplest aspects of FILE-COMPARE.

TABLE OF CONTENTS

1	INTRODUCTION TO FILE-COMPARE _____	1
1.1	Notation	3
2	STARTING FILE-COMPARE _____	5
3	CONCEPTS AND SPECIAL TERMINOLOGY _____	11
4	MAIN COMMANDS _____	15
4.1	COMPARE	17
4.2	EXIT	17
4.3	HELP	17
5	TWO DIFFERENT WAYS OF USING FILE-COMPARE _____	19
5.1	How to Switch between Modes	21
5.1.1	UNIT-OF-COMPARE	22
5.2	Command Description when Unit is SOURCE or RECORD	22
5.2.1	INCLUDE-MODE	23
5.2.2	TRIM-SPACES	23
5.2.3	UPPERCASE-MODE	23
5.3	Command Description when Unit is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER	24
5.3.1	DATE-CHECK	24
5.3.2	FORMAT-DISPLAY	25
5.3.3	HALT-MODE	26
5.3.4	MESSAGES	26
5.3.5	SEARCH-LENGTH	26
5.3.6	SIGNIFICANT-UNITS	27
5.3.7	TERMINATE	27
6	LOGFILE FORMAT _____	29
6.1	The Logfile Format when Unit is SOURCE or RECORD	31
6.2	The Logfile Format when Unit is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER	31
7	THE COMPARISON PROCESS _____	33
7.1	Comparing Files when Unit is SOURCE or RECORD	35

7.2	Comparing Files when Unit is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER	35
7.3	Equality of Units	37
8	COMMAND SUMMARY _____	39
8.1	Main Commands	41
8.2	Command to Switch Mode	41
8.3	Commands when Unit is SOURCE	41
8.4	Commands when Unit is BYTE, CHARACTER, DOUBLE, HALFWORD, LINE or WORD	42
9	EXAMPLES OF THE USE OF FILE-COMPARE _____	43
9.1	Example 1	45
9.2	Example 2	46
9.3	Example 3	48

TABLE OF APPENDICES

APPENDIX 1: ERROR MESSAGES FROM FILE-COMPARE	49
INDEX	53

1 : INTRODUCTION TO FILE-COMPARE

FILE-COMPARE is an ND-500 program to discover and describe differences between two sets of data/source code on mass storage. The system can handle data/source code organized in virtually any kind of format or record structure.

The system can be used in a very simple way, but if necessary the user can specify the logical partitions of the files, how the data fields are to be interpreted, and how the results are to be presented. To use the system at its simplest, you need only read chapter 2.

The system will detect the change of a single field, as well as the insertion and/or deletion of one or several blocks of data. A description of the changes that are discovered is written to a logfile specified by the user.

FILE-COMPARE may, for example, be used to find out what changes have been made from one version of a file to another, or to check the resulting output from two versions of a program.

NOTE! You cannot use the NOTIS function keys (like the HELP key) when using FILE-COMPARE.

1.1 NOTATION

The notation used throughout this manual is listed below:

< >	<ul style="list-style-type: none"> • Brackets indicate a required parameter.
[< >]	<ul style="list-style-type: none"> • Brackets enclosed by square brackets indicate an optional parameter.
<u>name</u>	<ul style="list-style-type: none"> • Underlined words are those <u>you</u> type in.

2 : STARTING FILE-COMPARE

You can start FILE-COMPARE by typing:

and FILE-COMPARE

or alternatively,

and FILE-COMPARE <OLDFILE>, <NEWFILE>, [<LOGFILE>]

In the first case, the program will prompt you for further information. In the second case, the program will validate the parameters and will then run straight away if they are correct. If they are incorrect, the program will issue an error message, and enter the command processor.

If prompts are requested, then the program responds with:

FCOM:

Start the comparison by typing the command "compare", and give the parameters:

FCOM: COMPARE <OLDFILE>, <NEWFILE>, [<LOGFILE>]

You get information on the terminal telling you how newfile differs from oldfile, and the discrepancies are written to the specific logfile.

Exit from FILE-COMPARE by typing

FCOM: EXIT

and look at the logfile through an editor.

- <oldfile> - is an "old" file containing the old program.
Default file type: :symb.
- <newfile> - is a "new" file containing the updated program that you want to see what changes has been done to compared with oldfile.
Default file type: :symb.
- <logfile> - is the output file that the discrepancies are written to. Give the name in double quotes if the file does not already exist.
Default file type: :symb.
Default file: the terminal.

If you only intend to use FILE-COMPARE in a straightforward way, you need not read any further in this manual.

EXAMPLE

We want to compare the file TEST-2 with TEST-1, and to write the differences between them to a file called LOG.

This is what the files look like:

	TEST-1 (the old file):	TEST-2 (the new file):
1	MUCH	A
2	WRITING	MASS
3	IS	OF
4	LIKE	LATIN
5	SNOW	WORDS
6	,	FALLS
7	A	UPON
8	MASS	THE
9	OF	RELEVANT
10	LONG	FACTS
11	WORDS	LIKE
12	AND	SOFT
13	PHRASES	SNOW
14	FALLS	,
15	UPON	COVERING
16	THE	UP
17	RELEVANT	THE
18	FACTS	DETAILS
19	COVERING	.
20	UP	
21	THE	
22	DETAILS	
23	.	

NOTE! The line numbers to the left are added here in the manual.

This is what you type and the messages you get on the screen:

```
@ND FILE-COMPARE
---  ND FILE-COMPARE, vers. ND-10603B  DECEMBER 10, 1986  ---
FCOM:COMPARE TEST-1 TEST-2 LOG
Number of units read in the oldfile:      23
Number of units read in the newfile:      19
Deleted 5, Inserted 1, Replaced 1 by 1, Moved 3 units.
-- End of compare --
FCOM:EXIT
- EXIT -
```

This is an explanation of the line that indicates what has been done to TEST-2 compared with TEST-1 (units are in this example lines):

Deleted 5	- 5 of the original units have been deleted.
Inserted 1	- 1 new unit has been inserted in the new file.
Replaced 1 by 1	- 1 unit in the old file has been replaced by a new one.
Moved 3 units	- 3 of the old units have been moved to a different location.

The logfile looks like this:

```
Name of oldfile : (user-area)TEST-1
Name of newfile : (user-area)TEST-2

**** Delete **** Deleted unit 1 to unit 3:
    MUCH
    WRITING
    IS
**** Replace **** Replaced unit 10:
    LONG
++++ Replace ++++ The unit(s) are replaced by:
    LATIN
**** Delete **** Deleted unit 12 and unit 13:
    AND
    PHRASES
**** Insert **** Inserted before unit 5:
    SOFT
**** Move **** Moved unit 4 to unit 6 before unit 19.

Deleted 5, Inserted 1, Replaced 1 by 1, Moved 3 units.
```

NOTE! The unit numbers refer to the oldfile, in this case TEST-1.

For further examples, see chapter 9.

3 : CONCEPTS AND SPECIAL TERMINOLOGY

This chapter explains certain terms and concepts related to FILE-COMPARE before continuing with the description of the program and its commands.

The two files to be compared are regarded as two sets of logical UNITS. Below are the UNIT types to be defined by the user.

Units are compared character by character (see example of output in section 9.2 and 9.3):

SOURCE = the default unit. Every file that consists of lines terminated by CR and/or LF can be regarded as a file with unit = SOURCE. This unit type is also used for NOTIS "S" and 16-bit representations.

RECORD = fixed-length groups of bytes

The great advantage of using the SOURCE unit type is that it enables files prepared with NOTIS-WP to be processed.

Units are compared byte by byte (see example of output in section 9.1):

BYTE = 8-bit groups

CHARACTER = 8-bit groups; parity (bit 7) cleared

HALFWORD = 16-bit groups

WORD = 32-bit groups

DOUBLE = 64-bit groups

LINE = strings of ASCII characters separated by CR and/or LF

For any normal symbolic file, the unit may be SOURCE. The unit type which is chosen applies to both of the files.

4 : MAIN COMMANDS

The three main commands are:

```
COMPARE <oldfile>,<newfile>,[<logfile>]
```

```
EXIT
```

```
HELP <command>
```

4.1 COMPARE

Form:

```
COMPARE <oldfile>,<newfile>,[<logfile>]
```

Function:

The contents of <newfile> are compared with the contents of <oldfile>. The names of the files and the discrepancies between them, will be written to <logfile>. Write the name of the logfile in double quotes if it does not already exist.

Default logfile: the terminal.

Default file types: :symb.

4.2 EXIT

Form:

```
EXIT
```

Function:

Leave FILE-COMPARE.

4.3 HELP

Form:

```
HELP <command>
```

Function:

List the available commands. If a command name is given after the HELP command, then information will be shown for that command only.

You can influence the way in which "compare" works by giving special commands to FILE-COMPARE before you start the program. These special commands are described in chapter 5.

5 : TWO DIFFERENT WAYS OF USING FILE-COMPARE

You can use FILE-COMPARE in two different modes. When the unit type is SOURCE or RECORD you are in one mode, and when it is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER you are in the other. The files are compared in different ways in the two modes. In addition, the commands in the two modes are different, as is the format of the logfile. These differences are described in sections 5.2 and 5.3. The default mode is with unit=SOURCE.

When the unit is SOURCE or RECORD the differences between the files are reported absolutely correctly. When the unit is SOURCE it is also possible to compare NOTIS files, but graphic characters will not be displayed correctly on the terminal. (When read into NOTIS-WP the logfile will be displayed correctly.) It is recommended that SOURCE or RECORD units are used wherever possible.

When the units BYTE, CHARACTER, HALFWORD, WORD, DOUBLE or LINE are used, the result is slightly less predictable.

The main commands COMPARE, EXIT and HELP may be used in either of the two modes.

5.1 HOW TO SWITCH BETWEEN MODES

To switch between the modes, use the command

```
FCOM: UNIT-OF-COMPARE <unit-type>,[<length>]
```

This command can be used to switch between unit type SOURCE or RECORD, and unit type LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER. The command description is given in section 5.1.1.

5.1.1 UNIT-OF-COMPARE

Form:

```
UNIT-OF-COMPARE <unit-type>,[<length>]
```

Function:

A UNIT is the element used for each individual comparison.

The <unit-type> parameter may take one of the following values:

```
SOURCE      : lines terminated by CR and/or LF.
RECORD      : fixed-length groups of bytes
BYTE        : 8-bit groups
CHARACTER   : as above, but parity bit (bit 7) is
              cleared before comparison
HALFWORD    : 16-bit groups
WORD        : 32-bit groups
DOUBLE      : 64-bit groups
LINE        : string of ASCII characters separated
              by CR and LF. This can, for example,
              be used with FORTRAN formatted
              records and COBOL 'TEXT' type files.
```

Default unit: SOURCE.

It is recommended that the SOURCE unit type is used wherever possible.

Length is a parameter giving the length of the fixed-length record. If the unit type is RECORD, this parameter is mandatory, otherwise it is ignored.

5.2 COMMAND DESCRIPTION WHEN UNIT IS SOURCE OR RECORD

When the unit is SOURCE you can use these 3 commands:

```
INCLUDE-MODE <on/off>
TRIM-SPACES <on/off>
UPPERCASE-MODE <on/off>
```

The description of these commands is written in sections 5.2.1 - 5.2.3. Note that these commands only apply when the unit is SOURCE. The INCLUDE-MODE command does not apply if the SOURCE file is a NOTIS file.

When the unit is RECORD you can only use the main commands COMPARE, HELP and EXIT.

5.2.1 INCLUDE-MODE

Form:

INCLUDE-MODE <on/off>

Function:

When INCLUDE-MODE is ON, "include" statements in the files will be executed. This means that the contents of the included files will also be compared. If INCLUDE-MODE is OFF, the contents of the included files will not be compared.

Default is OFF.

5.2.2 TRIM-SPACES

Form:

TRIM-SPACES <on/off>

Function:

When TRIM-SPACES is ON, the lines in the files will be trimmed so that spaces at the beginning and end of the lines will not be considered when comparing the files. In addition, multiple spaces within lines will be trimmed down to a single space. If TRIM-SPACES is OFF, spaces at the beginning and end of the lines will be taken into account.

Default is OFF.

5.2.3 UPPERCASE-MODE

Form:

UPPERCASE-MODE <on/off>

Function:

To distinguish between uppercase and lowercase, UPPERCASE-MODE must be OFF. If UPPERCASE-MODE is ON, FILE-COMPARE will not differentiate between uppercase and lowercase.

Default is OFF.

5.3 COMMAND DESCRIPTION WHEN UNIT IS LINE, WORD, HALFWORD, DOUBLE, BYTE OR CHARACTER

When the unit is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER, there are seven subcommands you can give to the COMPARE command:

```
DATE-CHECK <on/off>
FORMAT-DISPLAY <alpha/dec/oct/float/double-float>
HALT-MODE <on/off>
MESSAGES <on/off>
SEARCH-LENGTH <parallel>,<general>
SIGNIFICANT-UNITS <number-of-units>
TERMINATE <maximum-number-of-discrepancies>
```

5.3.1 DATE-CHECK

Form:

```
DATE-CHECK <on/off>
```

Function:

Output from programs often contains the date/time of the run. This information is, of course, different each time, but you do not usually want the fact to be reported.

With DATE-CHECK OFF, combined with unit = LINE or RECORD, two date/time printings in the same position will always be considered equal. The system will recognize dates and times in either of the three following formats:

```
XXX9:X9
or :
XXX9:X9:X9
or:
x9:x9:xxx9
```

where:

```
9 - numeric field
X - numeric or nothing
: - could also be / or . or -
```

If the first separation character is a period, only the last format will be accepted as a date or time. *Default is OFF.*

5.3.2 FORMAT-DISPLAY

Form:

FORMAT-DISPLAY <format>

Function:

The units that are found to be changed, can be displayed in one of the following formats:

ALPHA : alphanumeric
DECIMAL : decimal value
DOUBLE-FLOATING : 64-bit floating
FLOATING: the unit will be interpreted as
a set of 32-bit floating numbers
OCTAL : octal value of the unit

*Default format is ALPHA for LINE and
CHARACTER units, otherwise OCTAL.*

5.3.3 HALT-MODE

Form:

HALT-MODE <on/off>

Function:

If the option is ON, the system will halt after each discrepancy and ask whether the comparison should continue or not.

Default is OFF.

5.3.4 MESSAGES

Form:

MESSAGES <maximum-number>

Function:

After the given number of discrepancies has been reported on the logfile, the printing will be suspended.

The default is 4000.

5.3.5 SEARCH-LENGTH

Form:

SEARCH-LENGTH <parallel>,<general>

Function:

This command defines how much further in the file the program should search, once it has found a discrepancy. Both parallel searching and general searching may be limited in this way.

(See section 7.2).

Default is parallel=10 and general=100.

5.3.6 SIGNIFICANT-UNITS

Form:

SIGNIFICANT-UNITS <number-of-units>

Function:

If consecutive units corresponding to <number-of-units> are found to be unchanged in the files, the two areas are considered to be the same. *The default value is 3 for LINE and 7 for other units.* If the default values are not suitable for the files in question, the numbers may be adjusted according to the way in which the units are distributed. Very uneven distribution requires a high number.

There are both advantages and disadvantages associated with the value you give to the <number-of-units> parameter. If you choose a high number, the program will probably succeed in bringing the two files into phase, but the comparison will be slower and there is a greater possibility of units being overlooked. But a lower number may result in the program trying to phase the files together at the wrong place.

<number-of-units>=0 will give a parallel unit-by-unit comparison in the fastest way.

5.3.7 TERMINATE

Form:

TERMINATE <maximum-number-of-discrepancies>

Function:

The comparison will stop when the given number of discrepancies has been found. *The default is 4000.*

6 : LOGFILE FORMAT

The logfile describes the discrepancies between the two files. The format of the logfile is different in the two modes. There is one logfile format when the unit is SOURCE or RECORD, and another format when the unit is one of the other unit types (LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER).

6.1 THE LOGFILE FORMAT WHEN UNIT IS SOURCE OR RECORD

When the unit is SOURCE or RECORD, the logfile describes which units you have deleted, inserted, moved or replaced from <oldfile> to produce <newfile>.

Here is an example:

```

Old source-file name : (user-area)TEST-1
New source-file name : (user-area)TEST-2

**** Delete **** Deleted unit 1 to unit 3:
      MUCH
      WRITING
      IS
**** Replace **** Replaced unit 10:
      LONG
++++ Replace ++++ The unit(s) are replaced by:
      LATIN
**** Delete **** Deleted unit 12 and unit 13:
      AND
      PHRASES
**** Insert **** Inserted before unit 5:
      SOFT
**** Move **** Moved unit 4 to unit 6 before unit 19.

Deleted 5, Inserted 1, Replaced 1 by 1, Moved 3 units.
```

6.2 THE LOGFILE FORMAT WHEN UNIT IS LINE, WORD, HALFWORD, DOUBLE, BYTE OR CHARACTER

The logfile in this format describes differences between the two files when the unit is LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER. The files are compared unit by unit, and when two units differ, they are both written to the logfile together with the unit address in oldfile. If a unit exists only in oldfile, it is regarded as missing in newfile.

If the unit type is LINE, the unit address in oldfile is equal to the line number.

Here is an example of what this format may look like:

Comparison unit is LINE, 2 units were significant for recover

Byte ad- dress in file-1	Unit ad- dress in file-1	Bias in units file-1 -> file-2	File-1 "test1:symb"	File-2 "test2:symb"
000B	1	-1	MUCH	.missing
006B	2	-2	WRITING	.missing
017B	3	-3	IS	.missing
023B	4	-4	LIKE	.missing
031B	5	-5	SNOW	.missing
037B	6	-6	,	.missing
057B	10	-6	LONG	LATIN
101B	13	-7	PHRASES	.missing
112B	14	-8	FALLS	.missing
154B	18	-7	.missing	LIKE
154B	18	-6	.missing	SOFT
154B	18	-5	.missing	SNOW
154B	18	-4	.missing	,

23 units checked, 13 discrepancies found

7 : THE COMPARISON PROCESS

The program uses two methods for comparing files, depending on whether the unit is SOURCE or RECORD, or one of the other unit types (LINE, WORD, HALFWORD, DOUBLE, BYTE or CHARACTER).

7.1 COMPARING FILES WHEN UNIT IS SOURCE OR RECORD

This method for comparing files applies when the unit is SOURCE or RECORD. One expression needs to be explained: a "unique unit" is a unit (line or record) which occurs once only in both files. The program searches the files to see if they contain any equal unique units. If so, the program considers these units to be identical. It then compares the units on either side of the equal unique ones. If these are equal (they do not have to be unique), they are considered to be identical. And thus, this "block" of units is considered to be the same in both files. There is usually more than one equal unique unit in the two files, so this process may be repeated several times.

This method depends on the files having a number of unique units. If the files contain only a few, or none at all, the method will work badly.

7.2 COMPARING FILES WHEN UNIT IS LINE, WORD, HALFWORD, DOUBLE, BYTE OR CHARACTER

FILE-COMPARE scans the two files in parallel until a pair of unequal units is found. (See section 7.3). The system will then try to find the next point where the two files again match. This process is called PHASE-IN.

Once a discrepancy has been discovered, for instance at unit number X in one file and Y in the other, the attempt to bring them back into phase will be carried out in two steps:

1. The two files are checked in parallel forwards through the data from units X+1 and Y+1, to see if an unchanged area is found. The maximum length of the forward search is determined by the current value of the PARALLEL parameter in the SEARCH-LENGTH command.
2. The files are then systematically checked from X and Y up to X+n and Y+m respectively, for all combinations of m and n between 0 and GENERAL, where GENERAL is the current value of the second parameter in the SEARCH-LENGTH command.

During this process, the program may, in some cases, consider records to match where they should not, particularly if there are many blanks or zeros. If the user is familiar with the data, this possibility can be decreased by adjusting the length of the parallel search and the general search.

The first attempt, that is the parallel check, will only discover changes where no units have been deleted or inserted. The general search will find changes where units have been deleted and/or inserted.

It is possible to change the length of the two search steps with the help of system parameters. If you know that there have been no insertions or deletions, the length of the second step can be set to zero, and the first one made large.

7.3 EQUALITY OF UNITS

This section is only relevant when the unit is BYTE, CHARACTER, DOUBLE, HALFWORD, LINE or WORD.

An unchanged part of the two files is said to have been found when no changes have been discovered in a significant number of consecutive units.

The method used for deciding whether or not two units are equal, depends on the type of unit involved:

For the types BYTE, WORD, HALFWORD, CHARACTER and DOUBLE, binary equivalence is sufficient.

For type LINE :

- The contents of the lines are split into basic parts, such as:
 - Blanks - a sequence of blanks
 - String - alphabetic string
 - Number - numeric string
 - Date/Time - several formats are recognized for showing the date and/or time (for example 03.01.84 or 84.01.03).

If the records contain the same basic parts, the check continues as follows:

- The values of the basic parts are compared. This comparison is somewhat "fuzzy", in that the different printing formats for numbers, a few extra blanks, and unequal values for dates and times (if required) are considered to be insignificant.

Examples:

- 'AAAA 125 XXX' and 'AAAA 345 XXX' contains the same basic parts but one field is different.
- 'AAAA 0123 XXX' and 'AAAA 123 XXX' are equal.

The number of consecutive equal units needed to establish that two files are in phase, depends on the unit involved. This number may also be modified by the user; the larger it is, the better the chance of the phasing-in process being successful. This will naturally make the system slower. It could even force the system to give up the attempt to bring the files into phase, particularly if many separate small changes have occurred within an area.

8 : COMMAND SUMMARY

8.1 MAIN COMMANDS

COMPARE	<oldfile>,<newfile>,[<logfile>] <i>Default file types: :symb.</i> <i>Default logfile . : the terminal.</i>
HELP	<command>
EXIT	

8.2 COMMAND TO SWITCH MODE

UNIT-OF-COMPARE	<unit-type>,[<length>] SOURCE RECORD BYTE CHARACTER HALFWORD WORD DOUBLE LINE
-----------------	---

8.3 COMMANDS WHEN UNIT IS SOURCE

INCLUDE-MODE	<on/off> <i>Default: OFF</i>
TRIM-SPACES	<on/off> <i>Default: OFF</i>
UPPERCASE-MODE	<on/off> <i>Default: OFF</i>

8.4 COMMANDS WHEN UNIT IS BYTE, CHARACTER, DOUBLE, HALFWORD, LINE OR WORD

DATE-CHECK	<on/off> <i>Default: OFF</i>
FORMAT-DISPLAY	<alpha/dec/oct/float/double-float> <i>Default for LINE and Character: ALPHA</i> <i>Default for others: OCTAL</i>
HALT-MODE	<on/off> <i>Default: OFF</i>
MESSAGES	<maximum-number> <i>Default: 4000</i>
SEARCH-LENGTH	<parallel>,<general> <i>Default for: parallel 10</i> <i> general 100</i>
SIGNIFICANT-UNITS	<number-of-units> <i>Default for: LINE 3</i> <i> other units 7</i>
TERMINATE	<maximum-number-of-discrepancies> <i>Default: 4000</i>

9 : EXAMPLES OF THE USE OF FILE-COMPARE

Examples using FILE-COMPARE on different types of data follow.

9.1 EXAMPLE 1

Two files contain different versions of a source program. For such files, the unit type SOURCE should be used.

In this example, two versions of a PASCAL program are found on files PROG1 and PROG2, and the discrepancies between them are written to the file DIFF.

```
@ND FILE-COMPARE
```

```
--- ND FILE-COMPARE, vers. ND-10603B DECEMBER 10, 1986 ---
```

```
FCOM:COMPARE PROG1 PROG2 DIFF
```

```
Number of units read in the old file: 1918
```

```
Number of units read in the new file: 1920
```

```
Deleted 0, Inserted 2, Replaced 1 by 1, Moved 0 units.
```

```
-- End of compare --
```

The contents of the file DIFF will be:

```
Old source-file name : PROG1:SYMB
```

```
New source-file name : PROG2:SYMB
```

```
**** Replace **** Replaced unit 5:
  for i:=1 to 10 do
++++ Replace +++ The unit is replaced by:
  for i:=1 to 20 do
**** Insert **** Inserted before unit 11:
  if counter>=100 then
    counter:=0;
```

```
Deleted 0, Inserted 2, Replaced 1 by 1, Moved 0 units.
```

9.2 EXAMPLE 2

A test program tests a number of functions and generates a message if an abnormal situation occurs. (This particular program is obviously generating such situations on purpose.) The messages contain the date and time of the run, and an identification of the situation.

A good test for a system would be to run the test program and then compare the generated output with the output from a correct run.

If the program writes formatted messages to a file, you can define the following:

```
@ND FILE-COMPARE

*UNIT-OF-COMPARE LINE
*FORMAT-DISPLAY ALPHA
*DATE-CHECK OFF
*HALT-MODE OFF
*COMPARE TEST FACIT "LOG"
*EXIT
```

} default values
(you do not need to type these)

The result from the sample run might appear on the file LOG as follows:

Comparison unit is LINE, 3 units were significant for recover.

Unit ad- dress in file-1	Bias in units file-1 -> file-2	File-1 "TEST:SYMB"	File-2 "FACIT:SYMB"
122	0	*** 1983-06-06 15:48:52 SINT TRAN III EXCEPTION: (OB)	..missing..
123	-1	" 0" ERROR CODE OUTSIDE R ANGE	..missing..
213	-2	" 7612" ERROR CODE OUTSIDE R ANGE	STACK OVERFLOW
288	-3	EXCEPTION GROUP NOT ALLOWED (S C=0) "500B"	..missing..
288	-2	..missing..	PROTECT VIOLATION
288	-1	..missing..	RETURN ADDRESS 3232B
288	1	..missing..	*** 1983-06-06 15:48:54 EXCEPTION SYSTEM:(5113B)
288	2	..missing..	EXCEPTION GROUP NOT ALLO WED(SC=0) "500B"

---- 332 units checked, 8 discrepancies found ----

9.3 EXAMPLE 3

Two files contain data that should be equal. The files contain binary numbers in 32-bit integer format.

```

@ND FILE-COMPARE
*UNIT-OF-COMPARE WORD
*FORMAT DECIMAL
*COMPARE TEST-1:DATA TEST-2:DATA LOG
*EXIT

```

The contents of the file LOG are shown below.
Comparison unit is word (32 bits), 7 units were significant for recover.

Unit address in file-1	Bias in units file-1-> file-2	File-1 "TEST-1:DATA;1"	File-2 "TEST-2:DATA;1"
0000001736B	0	0	1
0000001737B	0	0	6
0000002006B	0	0	-1
0000002010B	0	0	-635422

---- 20000 units checked, 4 discrepancies found ----

APPENDIX 1 : ERROR MESSAGES FROM FILE-COMPARE

APPENDIX 1 ERROR MESSAGES FROM FILE-COMPARE

ERROR MESSAGES FROM FILE-COMPARE

FILE-COMPARE may generate the following messages:

ILLEGAL CHARACTER IN DECIMAL NUMBER	- You have entered a letter in a parameter where only digits are expected.
ILLEGAL CHARACTER IN OCTAL NUMBER	- You have entered a letter in a parameter where only octal digits are expected.
ILLEGAL PARAMETER	- You have entered a parameter which is not valid.
INVALID COMMAND	- You have entered an invalid command.
AMBIGUOUS COMMAND	- The command you entered could be one of several commands.
SYSTEM BUFFER EXHAUSTED	- The size of a changed area is so large that the system is unable to analyse it. You should reduce the size of the SEARCH-LENGTH, which will cause the analysis to be performed stepwise.
CANNOT OPEN LOGFILE	- The logfile you have specified probably does not exist.
STACK OVERFLOW. TOO MANY DIFFERENCES BETWEEN THE FILES	- The files are too different to be compared.
TOO MANY INCLUDE-FILES	- You have either too many include files, or too many levels of include files.
TOO LONG UNIT IN FILE	- The line or record is too long to be compared.
END OF FILE IN RECORD	- The record to be compared contains the end-of-file mark. You must adjust the record length.

TOO BIG FILES	- The files are too big to be compared.
TOO SMALL UNIT	- The record size is too small, probably 0.
TOO LONG INPUT STRING	- The input string is too long.
TOO LARGE VALUE, MAX IS 300	- When setting the search length, the maximum value permitted is 300.
TOO SMALL RECORD	- You have set the record size to a value which is too small.
ERROR IN NOTIS FILE FORMAT	- Internal error when reading a NOTIS file, or an attempt has been made to compare files of different formats.
CANNOT OPEN REMOTE FILE	- Remote files cannot be compared using this program.
COMMAND ABORTED	- You have pressed ESCAPE during execution.

In addition, all SINTRAN III error messages may be generated.

INDEX

BYTE	13, 22, 37, 42
CHARACTER	13, 22, 37, 42
COMPARE	9, 17, 21, 22, 24, 41
comparison of	
files	35
units	37
DATE-CHECK	24, 42
default file type	17
DOUBLE	13, 22, 37, 42
equality of units	37
EXIT	17, 21, 41
FILE-COMPARE	
examples of use	45, 46, 48
two modes	21
FORMAT-DISPLAY	24, 25, 42
GENERAL	36
HALFWORD	13, 22, 37, 42
HALT-MODE	24, 26, 42
HELP	17, 21, 41
INCLUDE-MODE	22, 23, 41
including files	23
LINE	13, 22, 37, 42
logfile	17, 31
logfile format	31
MESSAGES	24, 26, 42
notation	3
off-track	37
phasing files together	35, 37
program source	13

RECORD	13, 22
record length	22
search general	36
search parallel	36
SEARCH-LENGTH	24, 26, 36, 42
SIGNIFICANT-UNITS	24, 27, 42
SOURCE	13, 22, 41
switch between modes	21
TERMINATE	24, 27, 42
TRIM-SPACES	22, 23, 41
unchanged part of file (definition)	37
unique unit	35
unit of comparison	13, 22
UNIT-OF-COMPARE	21, 22, 41
UPPERCASE-MODE	22, 23, 41
WORD	13, 22, 37, 42

The information in this manual is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this manual. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S. Copyright ©1986 by Norsk Data A.S.

PRINTING RECORD	
PRINTING	NOTES
01/84	Version 1 EN Preliminary
03/84	Version 2 EN
10/85	Version 3 EN
01/87	Version 4 EN

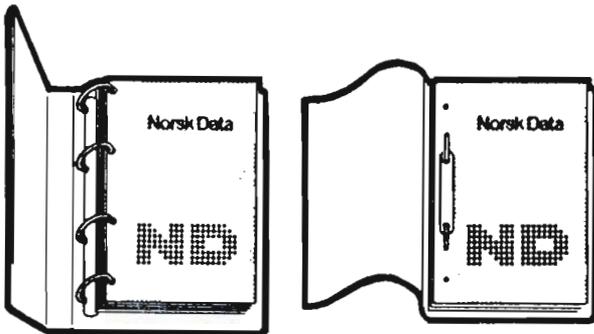
FILE COMPARE
Publ.No. ND -60.184.4 EN

UPDATING

Manuals can be updated in two ways, new versions and revisions. New versions consist of a completely new manual which replaces the old one, and incorporate all revisions since the previous version. Revisions consist of one or more single pages to be merged into the manual by the user, each revised page being listed on the new printing record sent out with the revision. The old printing record should be replaced by the new one.

New versions and revisions are announced in the ND Customer Support Information and can be ordered from the address below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and give an evaluation of the manual. Both detailed and general comments are welcome.



RING BINDER OR PLASTIC COVER

The manual can be placed in a ring binder for greater protection and convenience of use. Ring binders may be ordered at a price of Nkr. 45.- per binder.

The manual may also be placed in a plastic cover. This cover is more suitable for manuals of less than 100 pages than for larger manuals.

Please send your order, as well as all types of inquiries and requests for documentation to the local ND office, or (in Norway) to:

Norsk Data A.S
Graphic Center
P.O.Box 25 BOGERUD
N-0621 OSLO 6 - Norway



I would like to order

..... Ring Binders, 40 mm, at NOK 45.- per binder

..... Plastic Covers, at NOK 10.- per cover

Name:

Company:

Address:

***** **SEND US YOUR COMMENTS!!!** *****



Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card — and an answer to your comments.

Please let us know if you

- find errors
- cannot understand information
- cannot find information
- find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!



***** **HELP YOURSELF BY HELPING US!!** *****

Manual name: FILE COMPARE Reference Manual

Manual number: ND-60.184.4 EN

What problems do you have? (use extra pages if needed) _____

Do you have suggestions for improving this manual? _____

Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

What are you using this manual for? _____

NOTE!
This form is primarily for documentation errors. Software and system errors should be reported on Customer System Reports.

Send to:
Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

—————>
Norsk Data's answer will be found on reverse side

