

NORD FILE EXTRACT
UTILITY COMMAND
User's Guide

NORSK DATA A.S



**NORD FILE EXTRACT
UTILITY COMMAND
User's Guide**

NORD FILE EXTRACT UTILITY COMMAND
User's Guide
Publication No. ND-60.102.02



TABLE OF CONTENTS

+ + +

<i>Section:</i>		<i>Page:</i>
1	INTRODUCTION	1-1
2	PURPOSE	2-1
3	COMMAND STRUCTURE	3-1
3.1	Input File	3-1
3.1.1	Mode File Save Option	3-2
3.1.2	Limited Automatic Command Input	3-2
3.1.3	Fixed Record Length Input File Option	3-3
3.1.4	Indexed Access via KEY File	3-3
3.2	Output File	3-4
3.2.1	Output File Append Option	3-4
3.2.2	File Split Option	3-4
3.2.3	Output File Organization Change (X Option)	3-5
3.3	Extract Selecton Specifications	3-6
3.3.1	Numeric Field Evaluation	3-7
3.3.2	Text Field Evaluation	3-8
3.3.3	Text String Search	3-9
3.3.4	Limited Text String Search	3-10
3.3.5	Logical Operands	3-11
3.3.6	Parentheses Nesting	3-11
3.3.7	Input File Record Intervals	3-12
3.3.8	Show First Input File Record Option	3-12
3.3.9	Command Line Continuation Option	3-13
3.4	Output Specifications	3-13
3.4.1	Input Record Subset Specification	3-14
3.4.2	Output Record Constants	3-15
3.4.3	Input Record Number Inclusion	3-15
3.4.4	Output Record Number Inclusion	3-16
3.4.5	Random Key Inclusion Option	3-16
3.4.6	Terminal Output Wait Option	3-17
3.4.7	Line Printer/Terminal Output Heading Option	3-18
3.4.8	Line Printer/Terminal Page Numbering Option	3-19
3.4.9	Predefined Heading as Extract Command Line	3-19
3.4.10	Predefined Heading as Position Mask	3-20
3.4.11	Split File Copy Option	3-20
3.4.12	Show First Input File Record Option	3-20
3.4.13	Command Line Continuation Option	3-21
3.4.14	Skip Output Record Trailing Spaces	3-21
4	RUN-TIME STATUS MESSAGES	4-1

1

INTRODUCTION

File extract is a general purpose UTILITY program which can extract records from one file and write onto another file or output device.

In addition, by using the split option, records not satisfying given extract selection criteria can be placed in a second output file, thus providing a complete file split possibility.

The program provides for complex record selections invoked by simple parameters. The user may define his output record layout in several ways. Also, a wide range of output environment choices are available.

These facilities make the program useful in various data processing situations. Its use may range from very simple runs to rather sophisticated processing.

FILE-EXTRACT is written in FORTRAN. It handles standard SINTRAN III text files, including variable record length files. Maximum record size is set to 1024 bytes.

2

PURPOSE

A lot of administrative data processing may be expressed as the manipulation of big data files. Programming languages and Data Base Management Systems (DBMS) have been developed to take care of such things.

All the same, a great deal of processing takes place without a DBMS, using more conventional methods like sequential and random file structures.

Spontaneous, unproven needs are difficult to handle in such systems. They usually require special programming and test efforts, resulting in a time lag. The data of interest may then be too old when they are provided, or the expenses of providing these data too high.

FILE-EXTRACT is a tool enabling NORD computer users to do a lot of processing without having to do the programming for every specific need. Also, the program may be implemented as part of a system, just like SORT or other necessary utilities. Finally, the program may be of great help for functions like program development and testing, the correction of erroneous processing, and the fetching of transactions for rush processing, etc.

FILE-EXTRACT contains facilities such as:

- the extraction of subsets from files based on record numbering
- the extraction of subsets from the files based on individual record contents
- the rearranging of files
- the appending of files or subsets of files to other files
- file splitting by one run
- reformatting of files according to record layout, length and organization
- providing output records containing input record number
- providing output records containing the master record's physical address (see Section 3.4.4)
- conversion of transactions from various systems to a common layout
- generation of readable reports containing heading and page numbering routed to a terminal or a line printer
- saving of parameter input in MODE files for later automatic processing (see Section 3.1.1)
- building or procedures to be processed with limited run-time parameter input (see Section 3.1.2)

These facilities may be combined in various ways thus meeting new demands as they occur.

3 COMMAND STRUCTURE

FILE-EXTRACT may be called by it's name from a terminal when in the SINTRAN III command mode.

Command Structure:

@FILE EXTRACT

— NORD FILE EXTRACT UTILITY COMMAND, VER. DD MM YY —

INPUT FILE: <\$MODE><\$AUTO><\$KEY><,Fnnn>

OUTPUT FILE: _____<,X><,A><:>

<SPLIT OPTION OUTPUT FILE 2: _____<,A>>

EXTRACT SPECIFICATIONS:

<<SHOW> <extract selection criteria> _____<:>>

<_____>

OUTPUT RECORD LAYOUT SPECIFICATIONS:

<<SHOW> <Wnn> <L> <LO> <Hnn> <PAGE[="xxxx"]> <R> <E> <P> <C>

<T> <record layout> <:>>

<_____>

INPUT RECORDS: 99999, OUTPUT RECORDS: 99999 | = = = >----*----|

The program will request input from the user as shown above.

All input fields, except for INPUT-FILE, accept default values. Thus, a "default run" will cause the input file to be listed on the terminal.

The default value is indicated by typing CARRIAGE RETURN in the specific input field.

However, the command structure is made in such a way that the required options may be activated by use of simple parameters. Any other functions are automatically avoided.

3.1 INPUT FILE

The input file may be specified as any randomly accessible SINTRAN III text file. A default file type <:SYMB> is assumed when type is not specified. The file is immediately checked for legal access. If not obtained, an error message will be written to the terminal before program termination.

3.1.1 *Mode File Save Option*

The mode file save option may be invoked by typing <\$MODE> in response to the input file question. The following text will be written on the terminal:

MODE SAVE FILE:

In the file specified in answer to this question, all COMMAND INPUT will be saved as a SINTRAN III MODE file. In this way, specifications given for an extract run may be saved for later automatic processing, thus enabling the user to generate procedures under the guidance of the program.

3.1.2 *Limited Automatic Command Input*

The LIMITED AUTOMATIC COMMAND INPUT option may be invoked by typing <\$AUTO> in response to INPUT FILE. The program will immediately ask for:

AUTO RUN-TIME COMMAND FILE:

and then read the command input lines from the file specified here. This facility is quite similar to the execution of FILE-EXTRACT from a MODE file. The difference is that a command line in the AUTO RUN-TIME COMMAND FILE may contain the text \$TERM, meaning that this line is to be prompted from the terminal.

This option is very useful for complex predefined procedures, where some features are to be requested at run-time. An example could be a pregenerated report procedure where the user is to specify, at run-time, the output device as terminal or line printer, or perhaps some additional extract selection criteria to be read in. All other parameters and the report layout will automatically be read from the command file.

Such a command file may be generated by the MODE FILE SAVE OPTION (see Section 3.1.1) and then edited by QED. Remember to remove tabs when in QED (command M TO(0)).

3.1.3 *Fixed Record Length Input File Option*

To process a fixed record length input file not containing record delimiting characters (octal 015, 012, i.e., CR, LF), the F option must be used. The parameter should follow input file name and be specified as follows:

<,Fnnnn>

where nnnn specifies input file record length in bytes (maximum 1024 bytes).

Note that the output file, as a rule, will receive/have the same organization as the input file.

The following conditions will, however, make a sequential output file out of a "fixed" input file:

- output file organization change option specified (see Section 3.2.3)
- terminal output wait option specified (see Section 3.4.6)
- line printer/terminal heading option specified (see Sections 3.4.7, 3.4.8, 3.4.9 and 3.4.10)

3.1.4 *Indexed Access via KEY file*

Indexed access via KEY file is initiated by typing <\$KEY> in response to the input file question. The program will then ask for:

KEY FILE NAME:

The KEY file is only supposed to indicate which records of the input file are to be read and in which order. The KEY file must be a symbolic file, each record starting with a pointer to a corresponding record within the main input file. Any trailing contents of a KEY file record will be ignored by FILE-EXTRACT. A KEY file will normally be output of a FILE-EXTRACT run using the "Random Key Inclusion Option" and must follow the format used here (see Section 3.4.5). The file could then be sorted or processed in any way before being utilized as KEY file.

For situations which could benefit from this option, see examples mentioned in Section 3.4.5.

3.2 OUTPUT FILE

Output file may be any existing/nonexisting SINTRAN III disk file or an output device such as line printer or terminal.

The file name is specified due to the standard SINTRAN syntax. That is, nonexisting files must be enclosed by double quotes, etc.

Note that random write is always utilized unless output file TERM (terminal) is selected or the WAIT option (see Section 3.4.6) is switched on. So, when writing to any other sequential only output device, a dummy WAIT option must be used.

Default output file is the terminal.

3.2.1 Output File Append Option

The parameter <A> following output file name, invokes the output file append option. This means that the output will be appended at the end of the given file.

Note that this option requires an existing output file and is not valid for such output devices as terminal or line printer.

3.2.2 File Split Option

A <:> at the end of the output file input line invokes the file split option. The following test will be written to the terminal:

SPLIT OPTION OUTPUT FILE:

Records read, but not qualifying to be written to the main output file according to the extract selection criteria given (see Section 3.3) will now be written to the SPLIT OPTION OUTPUT FILE. If this option is not specified, those records will simply be bypassed by FILE-EXTRACT.

The append option <A> is also available for the split file (see Section 3.2.1).

3.2.3 *Output File Organization Change (X Option)*

The X option is used to switch the output file organization, thus making a sequential file containing end of record characters out of a random, fixed length record file and vice versa.

Consider a sequential, variable record length input file. By using the X option, a random, fixed length record output file will be produced. The output record length will automatically be computed from the output record layout specifications given (see Section 3.4). Note that X option switch to random file organization will be ignored when used together with certain other options (see Section 3.1.3).

Sequential records, delimited by End of Record characters will be produced when the X option is specified in conjunction with the fixed record length input file option (see Section 3.1.3).

Output file organization change may be useful in several situations. Consider a fixed length random data file needing some special editing. The X option can produce a QED editable version of the file, which could then be edited and finally reconverted to its original organization using the X option once again.

3.3 *EXTRACT SELECTION SPECIFICATIONS*

One or two input lines are available for extract selection specifications. The commands given here determine which records are to be written to the output file.

There are four types of selections available:

- specification of input file record intervals in question (see Section 3.3.7)
- specification of input record field values to be satisfied/not satisfied (see Sections 3.3.1 and 3.3.2)
- specification at text strings which are to occur/not occur within a record (see Section 3.3.3)
- specification of a text string which is to occur/not occur within a specified subset of a record (see Section 3.3.4)

The selection criteria specified may be connected by the logical operands `<.AND.>` and `<.OR.>` (see Section 3.3.5).

Finally, parentheses nesting on groups of selection criteria are allowed (see Section 3.3.6).

Together, these options provide a sophisticated data selection tool that may be used for the diverse tasks.

Note that extract criteria, logical operands, values and parentheses must not be separated by spaces. Spaces are treated as command line terminators.

3.3.1 *Numeric Field Evaluation*

A numeric field evaluation criterion is to be specified in the following manner:

<STARTPOS> [-ENDPOS] <operation code> <MIN VALUE> [-MAX VALUE]

where

STARTPOS

is the start byte number of numeric field within input record.

ENDPOS

End byte number of numeric field within input record. May be omitted for 1 digit fields.

OPERATION CODE

One of the following operation codes must be specified:

=	equal to
≠	not equal to
>	greater than
<	less than

MIN VALUE

is the numeric value for operation codes =, ≠ or the value to compare with the codes < and >.

MAX VALUE

is the maximum value that may be specified for operation codes = or ≠. It then specifies the upper numeric limit for a range specification, thus providing the additional operation codes "inbetween" and "not inbetween".

Example:

15 - 18 = 1590 - 1862

This means that if this particular extract selection criterion is to be satisfied, byte 15 through 18, within an input record, must contain a numeric value within the range 1590 to 1862.

3.3.2 Text Field Evaluation

A text field evaluation criterion is specified as follows:

<STARTPOS> [-ENDPOS] <operation code> <"text string">

where:

STARTPOS

is the start byte number within input record to be evaluated.

ENDPOS

is the end byte number within input record to be evaluated. May be omitted for one byte field.

OPERATION CODE

The two following operation codes are allowed:

= equal to
 ≠ unequal to

TEXT STRING

The text string may contain any character and must be surrounded by double quotes.

Note that the length of the text string must be the same as the field length specified by the STARTPOS/ENDPOS elements.

If shorter, a limited text string search will be assumed (refer to Section 3.3.4).

If longer, the specification will not be accepted and the program terminated with an error message.

Example:

45 - 50 = "OSLO 5"

3.3.3 *Text String Search*

A text string search specification will cause the entire input record to be scanned for the existence of the given text string.

A text string search is specified as follows:

TEXT <operation code> <"text string">

where:

TEXT

specifies search within the entire record.

OPERATION CODE

The two following operation codes are allowed:

= equal to
≠ unequal to

TEXT STRING

Any text enclosed by double quotes may be specified.

Example:

TEXT = "COMMUNICATION"

3.3.4 *Limited Text String Search*

A limited text string search will cause the specified subset of the input record to be scanned for the existence of the given text string.

Syntax:

<STARTPOS><-ENDPOS><operation code><"text string">

where:

STARTPOS

is the start byte number within input record where the text search is to be done.

ENDPOS

is the end byte number limiting search area within input record.

operation code

The two following operation codes are allowed:

= equal to
≠ unequal to

text string

The search text string may contain any characters (except double quote) and must be enclosed by double quotes.

Note: the length of the text string must be less than the record subset specified by startpos/endpos.

Example:

45 - 90 = "BOX"

This may extract those customer records having a P.O. Box address within the address fields subset of the record.

3.3.5 *Logical Operands*

A logical operand is used to connect two extract selection criteria of any kind.

Together with the parentheses nesting (see Section 3.3.6) this facility enables complex extract selections to be made.

Syntax:

<extract criterion A> <logical operand> <extract criterion B>

where:

extract criterion A and B

is the same as Sections 3.3.1, 3.3.2, 3.3.3 or 3.3.4 except for the input file record interval option as in Section 3.3.7.

logical operand

The two following operands are allowed:

.AND.	both criterion A and criterion B must be fulfilled
.OR.	either criterion A or B must be fulfilled

Example:

15 - 18 = 1590 - 8260 .OR. 45 - 50 = "OSLO 5"

3.3.6 *Parentheses Nesting*

Parentheses nesting is available for expressing more complex selections.

Extract criteria/groups of extract criteria connected with logical operands may be surrounded by parentheses/levels of parentheses.

Example:

((1 - 2 = "T1" .OR. 1 - 2 = "T2") .AND. 10 = 2) .AND. (15 - 22 > 90000 .OR. 23 = "**")

This could mean something like, "select those records of type T1 or T2 having status code 2 and either have a balance over 90,000 or are marked with a start in position 23".

Rules:

A start parenthesis must be placed before an extract criterion or together with another start parenthesis.

An end parenthesis must be placed after an extract criterion or together with another end parenthesis.

3.3.7 *Input File Record Intervals*

By specifying input file record intervals, one may select subsets of the input file to be evaluated.

Also, this option provides a file rearranging possibility due to the fact that the program will process input file records in the same order as indicated in the command line.

If a record interval is followed by another one specifying records already bypassed, the input file will be rewound before those records are processed.

Syntax:

<start record no.> — <end record no.> ,

where:

record no.

Record no. is specified with 1 to 9 digits

—

is start/end delimiter

is interval terminator. May be followed by parentheses or any other extract selection criterion including another input file record interval specification.

Note:

When record intervals are used to rearrange a file and the file split option is active (see Section 3.2.2) split file records will be duplicated every time the input file is rewound.

3.3.8 *Show First Input File Record Option*

Typing "SHOW" and the RETURN button at the beginning of the command line, the first input file record will be written to the terminal together with a position mask line such as:

```
123456789.123456789.123456789.123456789.1234....
7205PETTERSEN,PER    OSLO 5    223652    80000
```

This information is meant to be of assistance to the operator to see the position number for the different fields to be made extract selections from and has nothing to do with the actual output from the run.

The program will immediately accept input of extract selection specifications.

Note:

By typing another SHOW, the next record will be shown, thus providing selection of a record type layout representative record.

3.3.9 *Command Line Continuation Option*

Terminating the first command line with a <:> will provide another line for extract selection input.

Note:

Used together with the limited automatic command input (see Section 3.1.2) the first line may be specified beforehand, while the second may be used for additional operator selections at run-time.

3.4 *OUTPUT SPECIFICATIONS*

One or two input lines are available for various output specifications. A number of parameters are available to specify how records selected by the extract specifications are to be written (refer also to Section 3.3).

There are two main types of specifications available:

1. Specification of output record layout as one or more of the following elements:
 - a copy of input record
 - subsets of input record
 - imbedded constants
 - input record number
 - output record number
 - input record random address
2. Specification of output environment such as:
 - terminal output wait at full screen option
 - line printer/terminal heading specification
 - line printer/terminal predefined headings
 - page numbering
 - split file record as a copy of input record in spite of output specifications

Default (CR) makes the output record a copy of input record.

3.4.1 *Input Record Subsets Specification*

Subsets of input record can be specified to build the output record or to be a part of it.

Syntax:

<start position> [—end position] [,]

where:

start position

starts the position within input record to be copied to the output record.

end position

ends the position within input record to be copied. May be omitted when only one character is to be copied.

is specification delimiter in case of more specifications.

Example:

50 — 55, 1 — 20

This will produce an output record containing position 50 through 55 and finally the first 20 characters of the input record.

Note:

When the output record is specified to contain subsets of the input record, input records shorter than the subsets specified will result in an output record filled with spaces as a substitution for the missing input characters.

As a result, this facility can provide a file reformatting possibility, e.g., produce a fixed record length file out of a variable length one.

3.4.2 *Output Record Constants*

Constants may be imbedded in any position of output record.

Syntax:

"text" [,]

where:

text

may be any character except for double quotes.

[,]

is used as delimiter in case of more specifications.

Example:

50 — 55, "ABC", 1 — 26

This will insert the string "ABC" within the input record subsets specified.

3.4.3 *Input Record Number Inclusion*

The input record number may be specified to be the first element of the output record.

Syntax:

<L>[,]

The command will result in a 5 digit line number indicating source record number of input file.

Note: It cannot be used together with the <LO> or <R> options.

3.4.4 *Output Record Number Inclusion*

The output record number inclusion option will produce an output record containing a successive 5 digit record numbering as it's first element.

Syntax:

<LO>[,]

Note: It cannot be used together with the <L> or <R> options.

3.4.5 *Random Key Inclusion Option*

The random key inclusion option will cause the input record's random address to be included as the first element of the output record.

Syntax:

R [,]

The random address consists of the following two elements:

1. Block number, a 5 digit block number at least containing the first character of the input record. Block size used is 512 words.
2. Byte number, a 4 digit number pointing to the beginning of the actual record within a given block.

This option may be useful for several purposes. It can be used to show where a record (group of records) exist within a file. Also, it may be used for more well defined functions. For example, FILE-EXTRACT may be run to produce an output file containing this random address together with subsets from input records to be used as SORT key. Then, this KEY file may be sorted. The resulting file may then be used as an INDEX file in order to process the input file in quite a different order, without actually having sorted the input file previously. Such an index file may be utilized by FILE-EXTRACT itself through the KEY file option (refer also to Section 3.1.4).

This facility may have several advantages:

1. There may not be enough disk space to sort a bit input file itself.
2. A of a big input file may be very time consuming.
3. When an input file has to be accessed in many ways, this option will avoid the problems with keeping many copies of the same file.

Also, this option makes a limited input file sort possible by using the extract selection possibility to output only those records interesting and then use the SORT utility to produce a suitable index file.

Use your imagination!

3.4.6 *Terminal Output Wait Option*

The WAIT option is intended to be used with the terminal as output file. It simply makes the program wait for an input character for every given number of lines written to the terminal, thus enabling the user to study one screen of information before filling the next one.

The user may, at this point, interrupt the extract run by typing an X (exit). Any other character, including carriage return, will make the process continue.

Syntax:

W [nn] [,]

where:

nn

is a number of lines to be written before waiting for carriage return. Default value is 24 for standard VDU screens.

is specification delimiter in case of more parameters.

3.4.7 *Line Printer/ Terminal Output Heading Option*

The heading option enables the output from FILE-EXTRACT to be generated as simple reports with a one line heading, optionally together with page number (see also Section 3.4.8).

Syntax:

H [nn] [,]

where:

nn

is the number of lines per page. Default value is 24 (VDU terminal).

is parameter delimiter.

Note:

A common line counter is used for the heading and wait options. Therefore, if in doubt, the last line numbering specified in the command line will be used.

When all output specifications are given and the heading option is specified, the program will write a heading mask to the terminal and wait for user input:

```
HEADING MASK:
123456      123456      123456789.123456789.
KUNDENR. KONTONR.      N A V N
```

The first two lines above are produced by the computer. It simply represents a position mask of the output record, dimensioning the input record subsets chosen in the output specifications, corrected with constants if any. This mask indicates where to type the leading text in order to produce a readable report. Used together with the show option (see Section 3.4.12), the heading should have all changes to be correctly specified.

3.4.8 *Line Printer/ Terminal Page Numbering Option*

The page numbering option will provide a page number to be written before each heading. The parameter will have no effect when the heading option is not specified.

Syntax:

PAGE [= "page text"] [,]

where:

PAGE

This text which will invoke the option.

page text

The user may define his own 6 character long page text in his own language. Default text is "PAGE".

Example:

PAGE = "SIDE:"

This will, when used together with the heading option for each page, produce a heading such as:

SIDE: 9999

HEADING LINE

DETAIL OUTPUT LINE 1

DETAIL OUTPUT LINE 2

.
.
.

3.4.9 *Predefined Heading as Extract Command Line*

In some cases, it may be useful to have the extract selection specifications written together with the output. This is provided by the E option, which will automatically produce the extract command line as the heading line.

Syntax:

E [nn] [,]

The option works exactly like the H option (see Section 3.4.7) except it doesn't ask for heading input. Besides, the page numbering option (see Section 3.4.8) will automatically be invoked.

3.4.10 *Predefined Heading as Position Mask*

The P option produces a position mask as a predefined heading. This may be useful when record contents are to be studied in their original compressed format.

Syntax:

P [nn] [,]

This option works familiar to the E option (see Section 3.4.9).

3.4.11 *Split File Copy Option*

Normally, the split file output (see Section 3.2.2) will contain record layout similar to the main output (no page numbering and no headings). In some cases, it may be useful to provide a split file containing records as a copy of the input records. Thus, the C option will turn off any other output record layout specifications on split file writes.

Syntax:

C [,]

3.4.12 *Show First Input File Record Option*

The "SHOW" option is also provided as a first command to this output specifications input line. It works exactly in the same way as described above (see Section 3.3.8). In this case it is meant as a tool to produce an output record from the right subsets of the input record and also to help design the heading line.

Syntax:

SHOW

3.4.13 *Command Line Continuation Option*

Terminating the first command line with a <:> will provide another line for output specification input.

Note:

Used together with the limited automatic command input (see Section 3.1.2), the first line may be specified previously while the second one may be used for additional operator's choice at run-time.

3.4.14 *Skip Output Record Trailing Spaces*

In order to reduce disk space and increase processing speed, skipping trailing spaces may be desired. The option is supposed to be used in conjunction with variable record length output files.

Syntax:

T [,]

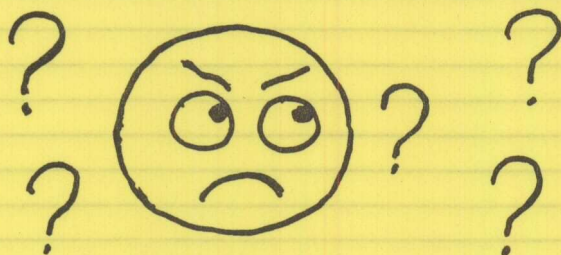
4 RUN-TIME STATUS MESSAGES

In order to enable the user to keep track of the program's progress, a run-time status message line is implemented:

INPUT RECORDS: 99999, OUTPUT RECORDS: 99999 | = = > ---- * ---- |

For every 100 input records processed, this line will be written to the terminal. The right side graph indicates the percentage (in bytes) of the input file being processed, thus enabling the user to estimate when the process will be finished.

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

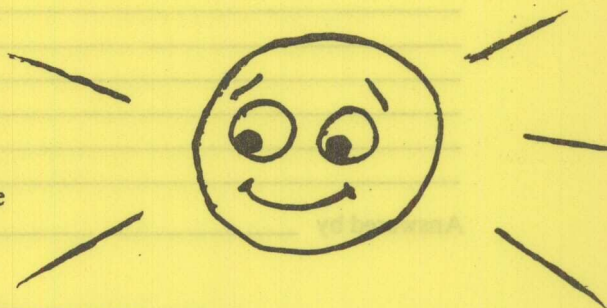


Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card - and an answer to your comments.

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: NORD FILE EXTRACT Utility Command Manual number: ND-60.102.02

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

Do you have suggestions for improving this manual?

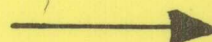
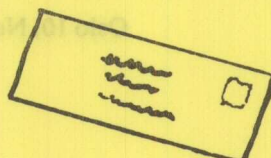
Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

What are you using this manual for?

Send to: Norsk Data A.S.
Documentation Department
P.O. Box 4, Lindeberg Gård
Oslo 10, Norway



Norsk Data's answer will be found on reverse side

Answer from Norsk Data

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

As you know, Norsk Data is a company that is constantly improving its products and services. We are interested in your comments and suggestions. Please let us know if you have any comments or suggestions. We will be happy to hear from you.

Answered by _____ Date _____

Manual name: NORD FILE EXTRACT Utility Command Manual number: ND-60.102.02

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

Do you have suggestions for improving this manual?

Your name: _____
Company: _____
Address: _____
Date: _____
Position: _____

Norsk Data A.S.

Documentation Department

P.O. Box 4, Lindeberg Gård

Oslo 10, Norway

– we make bits for the future