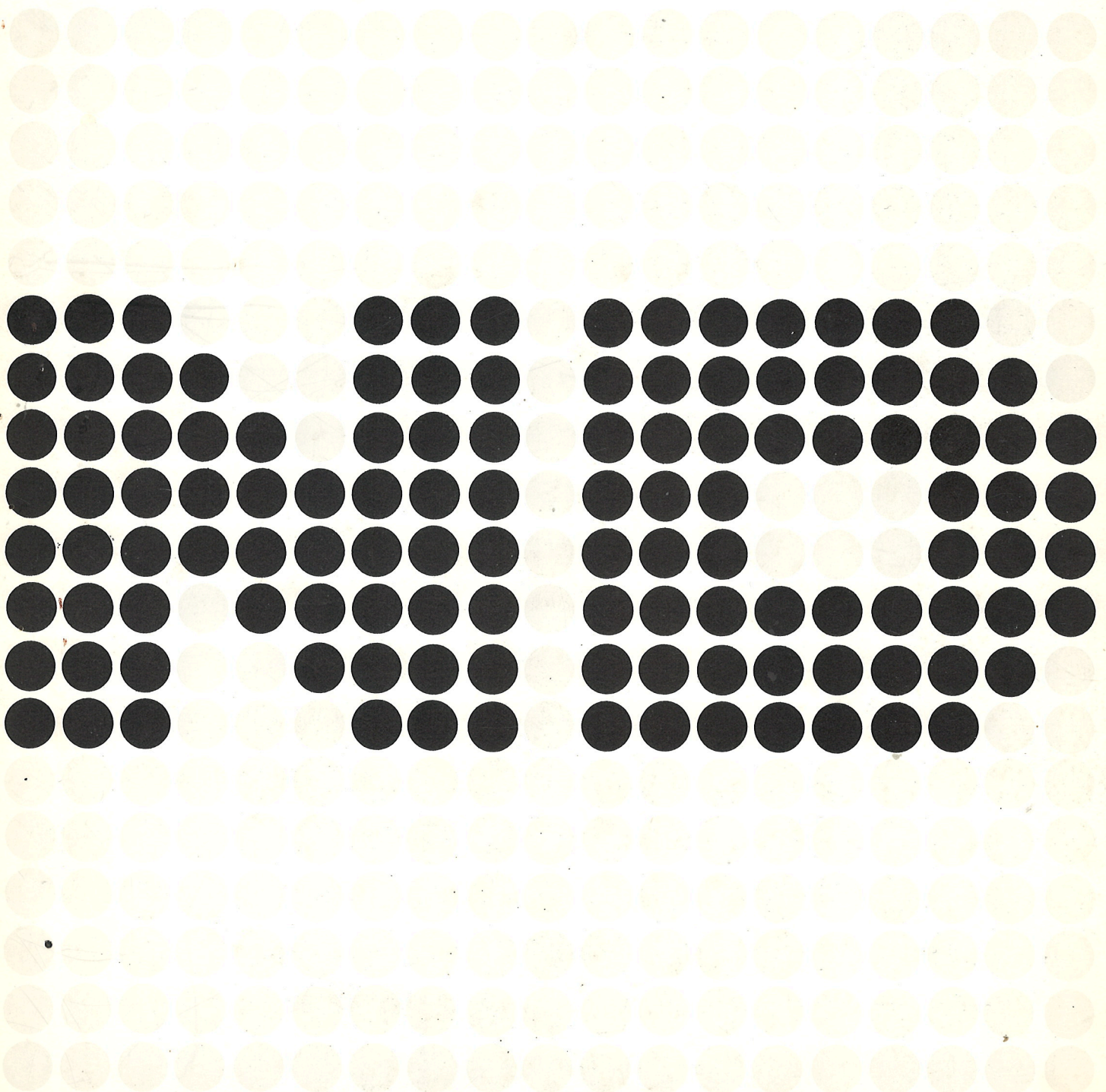


**NORD-50 LOADER**  
**User's Guide**

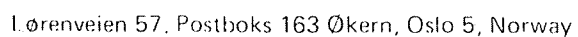
**NORSK DATA A.S**



# **NORD-50 LOADER**

## **User's Guide**

NORD-50 LOADER — User's Guide  
Publication No. ND-60.083.02



## TABLE OF CONTENTS

+ + +

<i>Section:</i>		<i>Page:</i>
1	GENERAL	1—1
1.1	How to Load and Execute a BRF Program	1—2
1.2	Load Address Control Commands	1—3
1.3	Commands Affecting the Symbol Table	1—4
1.4	Saving Executable Format Programs	1—4
1.5	Memory Image File Examination Commands	1—5
1.6	Setting Break Conditions for the Program	1—7
1.7	Overlay Segmentation of FORTRAN Programs	1—8
1.8	Communication between NORD-50 Programs and NORD-RT Programs through Shared Memory	1—11
1.9	Listing (Tracing) of Loader BRF Input	1—12
2	LOADER INSTALLATION	2—1
3	THE RELOCATING LOADER	3—1
3.1	Binary Relocatable Format	3—1
<i>Appendix:</i>		<i>Page:</i>
A	LOADER COMMAND SUMMARY	A—1
B	THE LOADER ERROR MESSAGES	B—1

# 1 GENERAL

Programs transformed by our assembler or compiler into Binary Relocatable Format (BRF) must be read and processed by a loader in order to be executed by the machine. Relocatable programs may be loaded anywhere in memory according to default system addresses or according to load addresses specified by the user.

By the load process, the BRF programs are transformed into an absolute binary format and the quality of relocatability is lost.

The program is loaded into memory image file where it resides in executable binary form. For the format of the memory image file and how to execute the program on the NORD-50 refer to the manual "NORD-50 Monitor, System Documentation" (ND.60.076).

In the text and in the examples, some symbols are used:

↵	carriage return
<...>	parameter to a command
[...]	the parameters enclosed are optional

In the examples, underlined text is user input.

## 1.1 *HOW TO LOAD AND EXECUTE A BRF PROGRAM*

The NORD-50 LOADER is executed on the NORD-10/SINTRAN III system independent of the NORD-50 CPU and memory.

The subsystem is started by typing @N50-LOADER. The loader asks for a memory image file to be used for building the executable format NORD-50 program. Default file number for the memory image file is 100<sub>8</sub> (the terminal scratch file) and default file type is :NOR5.

Loader input is obtained from one or more files/library files. The loading is initiated by the command:

\*LOAD <file name> [<file name> ...]

Each of the files specified will be searched and routines loaded until end-of-file is detected, then the control is transferred to the loader command processor (types \*) which is then ready to accept another command. The bracket contents denote optional parameters. The default file type is :BRF5.

To obtain the entry-point addresses of the loaded program, use the command:

\*ENTRIES-DEFINED [<file name>]

which will give you a printout of the entry names along with their octal address in memory. If no file/device name is specified, the printout will appear at your terminal.

Referenced but not defined entry-points may be displayed by the command:

\*ENTRIES-UNDEFINED [<file name>]

If you have loaded a FORTRAN program and some references still remain, the FORTRAN run-time/library system file should be loaded. If any of these routines are necessary for the execution they will be selected by the loader and connected with their corresponding references.

If you wish to leave the loader and enter the operating system you simply write:

\*EXIT

The loader operation is then ended and the memory image file is closed.

You may re-enter the loader by using the system command:

@CONTINUE.

If you are loading a FORTRAN program, the NORD-50 Formatted Input/Output system must be loaded and the NORD-50 library should be loaded as the last file. This is done automatically by the loader.

The file defined by the command

\*DEFINE-FIO <file name>

is loaded when the first LOAD or LIBRARY command is issued and the files defined by the command

```
*AUTOMATIC
<file name 1>
<file name 2>
```

```
<file name n>
```

are loaded when the command

```
*EXIT
```

or

```
*LIBRARY
```

is issued. Default file name for FIO is N50-FIO and default file name in the AUTOMATIC buffer is N50-LIB. If the loading of the files is defined by the DEFINE-FIO or AUTOMATIC commands are not wanted, the FIO and AUTOMATIC buffers have to be cleared by the commands:

```
*DEFINE-FIO,,,
```

or

```
*AUTOMATIC
```

```
.
```

```
*
```

*Example (loading a FORTRAN program):*

```
@N50-LOADER
MEMORY-IMAGE FILE: N50PROG
*LOAD MAIN-ROUT, USER-LIB-1, USER-LIB-2
*EXIT
@
```

## 1.2 LOAD ADDRESS CONTROL COMMANDS

If you wish to load your program at a specific address you may obtain this by typing:

```
*SET-LOAD-ADDRESS <octal address>
```

Subsequent loading will then be performed from the address specified.

Also, the absolute upper load limit may be redefined with:

```
*UPPER-LIMIT <octal address>
```

Be certain that no overlapping occurs when manipulating load addresses.

### 1.3 *COMMANDS AFFECTING THE SYMBOL TABLE*

Symbol table-entry-points may be created, renamed or deleted by the user. An entry is created by:

\*DEFINE <entry name> <octal value/address>

Symbol names may be renamed by

\*RENAME <old symbol name> <new symbol name>

and an entry is deleted by

\*KILL <symbol name>.

The associated address/value of an entry-point may be examined by typing:

\*VALUE <symbol>

The loader then prints the octal number on the terminal.

The associated address/value of an entry-point may be added to memory location by the command:

\*REFERENCE <symbol> <octal memory address>

It doesn't matter if the referenced entry-point is present in the table or not, as the correct address will be filled in when the symbol value is defined.

All table contents are removed by typing:

\*RESET

### 1.4 *SAVING EXECUTABLE FORMAT PROGRAMS*

The necessary information for running the program on the NORD-50 is saved on the memory image file when the loading is terminated by the EXIT command.

The format of the memory image file is described in the manual "NORD-50 Monitor, System Documentation" (ND-60.076) in Sections 6.5.1 and 6.2.2.



## 1.5 MEMORY-IMAGE FILE EXAMINATION COMMANDS

The command

\*[LOOK-AT] <address or register>

or

<address>

or

<register>

enters the "Look-at mode" to display and/or change the NORD-50 memory image file or registers. Output formats are set in the command FORMAT. In the "look at mode" the system will respond like this:

<op. code>, <number>, — — —, <number>

or

<number> ↘ contents of memory or register are changed

<address or register> / contents are displayed

↘ the next memory location or register is displayed

@ returns to normal command mode

(Refer to the following example.)

### \*STATUS

prints some information about the NORD-50 status.

### \*FORMAT <formats>

sets output formats to be used in the commands PRINT and LOOK-AT. Formats may be O, D, F, S, I, A, B, T or any combination of these characters.

O	octal
D	decimal
F	floating point double precision (64 bits)
S	floating point single precision (32 bits)
I	instructions (disassembled)
A	ASCII (one word = 4 ASCII characters, non-printable characters are replaced by space)
B	binary
T	NORD-10 format (the two half-words as 16-bit octal numbers)

To obtain printouts of larger parts of memory the command

\*PRINT <low address or register>, <high address or register>, [<file name or number>]

could be used.

The contents of the address or register interval are printed on the specified file. Output formats used are determined by the last FORMAT command given.

If the loader is used for examining or changing a memory-image file prepared in a previous load session or written by the SAVE command in the NORD-50 Monitor (ND-60.076), Section 3.1.3.8, the command

\*DEBUG

must be issued before the examination is started. This command retrieves the status information and defined and undefined symbols for the program from the memory-image file.

Example:

- 1) \*FORMAT OI ↵
  - 2) \*PRINT )  
LOW ADDRESS: 0 ↵  
HIGH ADDRESS: 10 ↵  
FILE NAME OR NUMBER: TELE ↵
- |            |             |     |     |       |     |     |   |
|------------|-------------|-----|-----|-------|-----|-----|---|
| 000000:    | 20227050000 | STR | 05, | 0000, | 04, | 00, | I |
| 000001:    | 00223050000 | LDR | 05, | 0000, | 04, | 00  |   |
| 000002:    | 00267040001 | STR | 04, | 0001, | 05, | 00  |   |
| 000003:    | 00267060002 | STR | 06, | 0002, | 05, | 00  |   |
| 000004:    | 00201000001 | RTJ | 00, | 0001, | 04, | 00  |   |
| 000005:    | 00012060013 | JRZ | 06, | 0013, | 00, | 00  |   |
| 000006:    | 14000070006 | RAD | 07, | 00,   | 06  |     |   |
| 000007:    | 00263050000 | LDR | 05, | 0000, | 05, | 00  |   |
| 3) 000010: | 00263040001 | LDR | 04, | 0001, | 05, | 00  |   |
- 3) \*FORMAT ↵  
FORMATS (O, I, D, A, F, S): DOSA ↵  
\*PRINT 0 10 1 ↵
- |         |             |            |    |              |
|---------|-------------|------------|----|--------------|
| 000000: | 20227050000 | -210789424 | \P | -3.18891E-75 |
| 000001: | 00223050000 | 38555648   | LP | 2.63619E-75  |
| 000002: | 00267040001 | 47988737   | \@ | 1.27470E-74  |
| 000003: | 00267060002 | 47996930   | \  | 1.27643E-74  |
| 000004: | 00201000001 | 33816577   |    | 1.17452E-75  |
| 000005: | 00012060013 | 2646027    | (  | 7.04220E-78  |
| 000006: | 14000070006 | 1610641414 |    | 1.71305E+38  |
| 000007: | 00263050000 | 46944256   | LP | 1.05448E-74  |
| 000010: | 00263040001 | 46940161   | L@ | 1.05361E-74  |
- 4) \*LOOK-AT 0  
000000: 20227050000 -2107879424 \P -3.18891E-75 ↵  
000001: 00223050000 38555648 LP 2.63619E-75 @
  - \*FORMAT OI ↵  
\*0 ↵
- 5) 000000: 20227050000 STR 05, 0000, 04, 00, I LDR 15 ↵  
000001: 00223050000 LDR 05, 0000, 04, 00 STR 16 ↵  
000002: 00267040001 STR 04, 0001, 05, 00 STOP 134 ↵  
000003: 00267060002 STR 06, 0002, 05, 00 177 ↵  
000004: 00201000001 RTJ 00, 0001, 04, 00  
000005: 00012060013 JRZ 06, 0013, 00, 00 FCN 2.334 ↵  
000006: 14000070006 RAD 07, 00, 06 5/ ↵  
000005: 10042530040 EXC 53, 0040, 01, 10 @ ↵  
F O S ↵  
\*5 ↵  
000005: 2.33400E+00 @  
\*  
\*EXIT ↵  
@

- 1) The FORMAT command sets the print/inspect output to be given as octal value and instruction format.
- 2) The PRINT command outputs a memory area specified by the lower and upper address to the file, which could be any output device or user file. Default file (if only a carriage return is given) will be the users terminal.
- 3) The same area is printed with octal, decimal, ASCII and single precision floating point format specified.
- 4) The LOOK-AT command displays one location at a time, allowing the user to change the contents, continue to the next location by typing carriage return, or terminate the command by typing the @ character.
- 5) The user may change the contents of a location in several formats:
  - instructions (address 0 to 2)
  - octal value
  - floating constant (address 5, using the pseudo instruction FCN: floating constant)

A new location to be displayed can be entered by an address followed by a slash (/), and the contents of that address are displayed.

## 1.6 SETTING BREAK CONDITIONS FOR THE PROGRAM

Default break conditions are selected by the loader when the loading session is terminated by the EXIT command: BP to the top of the program, BQ to the lower end of the common and break on store and stop on parity error. This may be overridden by the command:

`*BREAK-CONDITIONS <BP address>, <BQ address>, <break conditions>`

which sets the NORD-50 breakpoint registers and break conditions. <Break conditions> may be A, D, F, S, O, U, P or any combination of these characters.

A	stop on any reference in BP ≤ BQ
D	stop on data reference in BP ≤ BQ
S	stop on store reference in BP ≤ BQ
F	stop on fetch reference in BP ≤ BQ
O	stop on overflow
U	stop on underflow
P	stop on parity error in memory

The command will affect the Load Address and Upper Limit (Section 1.2).

If only the break conditions are to be changed and not the limit registers BP and BQ, default values of BP and BQ may be used.

*Example:*

`*BREAK-CONDITIONS,,AP`

For further information about the NORD-50 address violation system refer to "NORD-10/NORD-50 Communication System" (ND-06.005), Chapters 1 and 2 and "NORD-50 Monitor, System Documentation" (ND-60.076), Section 3.1.3.9.

## 1.7

## OVERLAY SEGMENTATION OF FORTRAN PROGRAMS

As a program may be too large to fit in the available memory space, the programmer may decide to divide his program into several overlay-segment modules. When the program system is generated in this way, only certain portions (root-segment + one overlay-segment) of the executing program need to be in memory concurrently. The various overlay-segments reside in the same area of memory at different times, and during time of execution they are loaded automatically (in binary form) by the run-time system when the control is transferred to one of its entry point. The overlay structure consists of a main program (referred to as the root segment) and one level of associated overlay segments.

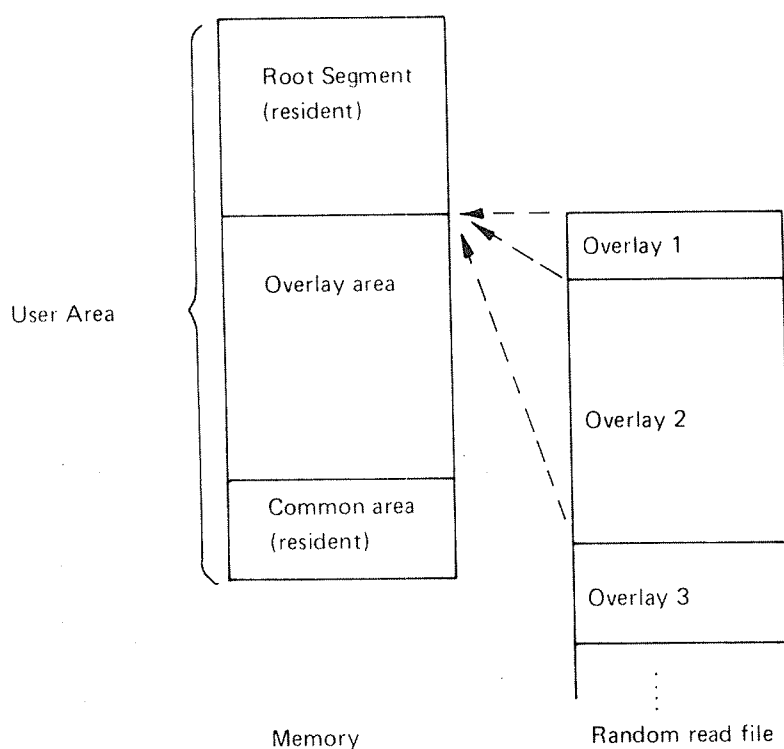


Figure 1.1: The Overlay Structure

The root segment and the common area reside in memory throughout the entire execution, while the overlays reside on a random read file. This file is specified with the OVERLAY-FILE command which also acts as an overlay-modus setting. The OVERLAY-FILE command should, therefore, be the first directive given after recovering the loader from the operating system. Default type of the overlay-file is OVL5.

Note that overlay-segment transfers from mass storage to memory, during program execution, will be much faster for contiguous files than for indexed files.

*Example:*

OVERLAY-FILE OVLAY1

The root segment is generated by loading the main program, along with some (user selected) frequently used function/subprograms.

Usually, when the root-segment is completed, some undefined subprograms are referenced. Such referenced subprograms may be grouped into overlay-segments in various ways. In generating overlays, the programmer should organize his program to retain the commonly used subprograms in the root segment and the less used routines in the overlay-segments, which reside in memory only temporarily, one at a time. The set of subprograms on an overlay-segment is specified by the loader command:

**\*OVERLAY-ENTRY** <name 1> <name 2> ... <name n>

where the names refer to subprograms called from the root segment.

When this command is given, the specified subprograms can be loaded from one or more BRF files. It is recommended that the overlay subprograms be kept on a separate BRF file compiled in library mode (refer to the FORTRAN Reference Manual). In this way, the specified set of subprograms may be selected into the overlay independently on the compilation sequence.

When all specified entry points are defined and no other undefined references occur on this overlay, the message

**OVERLAY COMPLETED**

is given.

When all specified entry points are defined but other references occurred during the load process, the message

**UNDEFINED SYMBOLS ON OVERLAY**

is given. The file(s) containing these entry points may then be loaded in order to complete the overlay.

When an overlay is completed another one may be specified and created according to the outline above.

An overlay system is considered to be complete if no undefined reference occurs on the entry map (UNDEFINED-ENTRY command).

The user should consider the following restrictions:

- Only one level of overlays is possible, thus, a routine in the root segment may reference any other routines in the root segment or overlay subprograms, while an overlay subprogram may only reference subprograms in its associated overlay or in the root segment.
- The FORTRAN debugging option can only be used for the root segment.

*An Example of Overlay Generation*

In the following example the root segment is compiled into the file ROOT-BRF5, and the subprogram into LIBSUB:BRF5 (in library mode) in the sequence SUBR1, SUBR2, SUBR3, SUBR4. To generate a program system with SUBR1, SUBR4 on overlay 1 and SUBR2, SUBR3 on overlay 2, the following command sequence will apply:

```
*OV-FI OVERLAY-SYSTEM
*LOAD ROOT
*OV-ENT SUBR1 SUBR4
*LOAD LIBSUB
OVERLAY COMPLETED
*OV-ENT SUBR2 SUBR3
*LOAD LIBSUB
OVERLAY COMPLETED
*EXIT
@
```

```
*LOCAL-COMMON
```

Common areas defined after this command is issued will be local to the overlay.

## 1.8 COMMUNICATION BETWEEN NORD-50 PROGRAMS AND NORD-10 RT PROGRAMS THROUGH SHARED MEMORY

### RT-COMMON-DEFINE

All RT-COMMON labels defined by the RT-LOADER (ND-670.051) will be defined as common labels in the loader Symbol Table. Labels defined to odd addresses in the NORD-10 SINTRAN III RT-COMMON will be listed and *ignored*. The addresses are transformed from NORD-10 addresses according to the information given in the NORD-50 LOADER memory Configuration Table.

The RT-COMMON-DEFINE command should be used before the program units referring to the RT-COMMON area are loaded.

When this command has been used, information about the size and address of RT-COMMON is put on the Memory Image File. This information is checked by the NORD-50 MONITOR before the program is allowed to run on the NORD-50.

### RT-PROGRAM-DEFINE

Define RT-PROGRAM names. The command should be used after the program units referring to the RT program names are loaded.

Only RT program names which are undefined symbols to the NORD-50 LOADER are defined by the command.

### SEGMENT-COMMON-DEFINE <segment no.>

All Segment Common Labels defined by the RT-LOADER (ND-60.051) on the segment specified will be defined as common labels in the Loader Symbol Table. Labels defined to odd addresses in the NORD-10 SINTRAN III Segment Common will be listed and *ignored*. The addresses are transformed from NORD-10 to NORD-50 addresses according to the information given in the NORD-50 LOADER Memory Configuration Table.

The SEGMENT-COMMON-DEFINE command should be used before the program units referring to the Segment Common are loaded.

When this command has been used, information about the contents of the Loader Memory Configuration Table (Chapter 2) is put on the Memory Image File. The NORD-50 MONITOR will then run the program with the memory configuration defined by the contents of the Memory Image File ignoring the default memory configuration for the actual installation as defined by the SET-MEMORY command in the NORD-50 MONITOR.

## 1.9 LISTING (TRACING) OF LOADER BRF INPUT

Listing (tracing) of loader BRF input may be turned on by the command

\*SET-LIST-MODE <output file>

and turned off by the command

\*RESET-LIST-MODE.

Example:

```

@N50-LOADER
NORD-50 LOADER
MEMORY-IMAGE FILE: ↓
*LOAD TEST
FREE: 0016315    0175777
*SET-LIST-MODE TERM
*LOAD OBJ
0016315    IDNT    002    002
0016315    DEF     PER      0016315
0016315    LF      20241040007    RTJ    04, 007, 05, 00,I
0016316    EXTR    VBAS
0016316    LF      00000000000    STOP    000000
0016317    LF      24540000001    SETA    13, 000001
0016320    LF      00267130013    STR     13, 0013, 05, 00
0016321    LF      00320060001    ADD     06, 0001, 06, 00
0016322    LF      20241000010    RTJ     00, 0010, 05, 00,I
0016323    LF      00000000001    STOP    000001
0016324    LF      00124400000    ADM     40, 0000, 02, 00
0016325    DEF     VBAS      0016325
0016325    DEF     ABAS      0016342
0016325    LF      00000000000    STOP    000000
0016326    LF      00000000000    STOP    000000
0016327    LF      00000000000    STOP    000000
0016330    LF      00000000000    STOP    000000
0016331    LF      00000000000    STOP    000000
0016332    LF      00000000000    STOP    000000
0016333    EXTR    ABAS
0016333    LF      37777777777    FDV     77, 7777, 17, 17,I
0016334    EXTR    ENTR.
0016334    LF      00000000000    STOP    000000
0016335    EXTR    LEAV.
0016335    LF      00000000000    STOP    000000
0016336    LF      00000000000    STOP    000000
0016337    LF      00000000000    STOP    000000
0016340    LF      00000000000    STOP    000000
0016341    LF      00000000000    STOP    000000
0016342    AFL      0000000
0016342    END
0016342    EOF

FREE: 0016342    0175777
*RESET-LIST-MODE
*EXIT
FREE: 0016342    0177777
@

```



## 2 LOADER INSTALLATION

1. The loader is supplied on a binary file on paper tape or floppy disk. Read this loader into memory by using the SINTRAN III command LOAD-BINARY. The loader will then start and the message:

NORD-50 LOADER  
MEMORY-IMAGE FILE:

2. is printed. Answer with carriage return.

3. Enter the command

\*SET-MEMORY <segment no.> <size> <type>

with the segments defined as the NORD-50 memory by the SET-MEMORY command in the manual "NORD-50 Monitor" (ND-60.076), Section 3.1.6.1.

— 1 is used as segment number of local memory and RT common. Segment number = 0 will end the command. The size must correspond with the actual segment size.

Type is: 0 = dynamic, 1 = static, 2 = RT common, 3 = local, 4 = local with DMA access and 5 = hole. Refer to the manual "NORD-50 Monitor", Section 4.3.

4. The

\*LIST-MEMORY

command may be used to check the memory information given to the loader.

5. Issue the DEFINE-FIO or AUTOMATIC commands if non-default setting for the library files is desired.
6. Enter the SINTRAN III command processor by using the QUIT command.
7. Make the loader a SINTRAN III subsystem by using the SINTRAN III command DUMP.

### 3 THE RELOCATING LOADER

#### 3.1 *BINARY RELOCATABLE FORMAT*

A FORTRAN program is transformed by the compiler into a certain format, Binary Relocatable Format (BRF). This format consists of control information and program information and may be read and processed by a loader only. The control information tells the loader how to act upon the immediately following program information. The BRF format is fully recoverable, thus its information is memory address independent.

The NORD-50 LOADER links the program units together and places the resulting executable absolute binary program on a file.

BRF Structure:

Control byte:                      Additional bytes, the number defined by the control byte:



The control byte consists of 8 bits. The values 0 through 42<sub>8</sub> are legal.

The structure resembles that of NORD-10 BRF. The main difference is that in NORD-50 BRF it is possible to add a symbol value to a location. The NORD-50 BRF uses a few different control bytes.

The information following the control byte is composed of the following constituents:

H group	2 bytes
W group	4 bytes (one NORD-50 word)
S group	6 or 8 bytes, containing a symbol

Six bytes symbol format: 6 bits characters, right adjusted, leading zeros.

Eight bytes symbol format: 8 bits characters, left adjusted trailing spaces.

## The Control Bytes of BRF5:

Control Value	Byte Name	Trailing Information	Meaning
0	FEED	—	Neglect
1	AFL	H	Current location counter (CLC) is modified by H
2	LF	W	Load fixed
3	EXT	S	External symbol or common label is referenced
4	DEF	S, H	Symbol definition
5	LIB	S, H	Library symbol definition
6	END	—	End of program unit
7	SFL	W	Set CLC to W
10	LNF	H, (H) * W	Load fixed H words
11	LR	W	Load relative to program base (PB)
12	EXTD	S	External symbol for displacement is referenced
13	COM	S, W	Define common label
14	MAIN	S, H	Symbol for main entry point is defined
15	INHB	—	Warns that compilation errors have occurred
16	EXTR	S	External symbol is referenced
17	EOF	—	End of file
20	INL	H, W	One word local data
21	EXTC	S	Common label is referenced
22	RLL	H, 2W	Two words local data
23	CXL	H, 4W	Four words local data
24	INC	S, H, W	One word common data
25	IDNT	5H	Compiler identification H <sub>1</sub> : byte 0; 1 - 8 bytes symbols H <sub>1</sub> : byte 0; 2 - 6 bytes symbols H <sub>1</sub> : byte 1; 1 - assembly H <sub>1</sub> : byte 1; 2 - FORTRAN H <sub>2-5</sub> : not used
26	REC	S, H, 2W	Two words common data
27	CXC	S, H, 4H	Four words common data
30	INLW	2W	One word local data
31	RLLW	3W	Two words local data
32	INCW	S, 2W	One word common data
33	RECW	S, 3W	Two words common data
34	CHL	W, H	One character local data H: bit 14 and 15; byte no., bit 7-0, byte data
35	CHC	S, W, H	One character common data. H as for 34 CHL
36	DEB	—	Begin/end debug information
37	AFR	2W	Add W <sub>1</sub> to location W <sub>2</sub> + program base
40	PRI0	H	Program (task) priority
41	SUBR	S	External symbol (subroutine)
42	LIBR	S	Conditional loading, load until END if S undefined

## APPENDIX A

### LOADER COMMAND SUMMARY

The loader is controlled from the terminal or batch input file by the set of commands listed below. The command words may be abbreviated and the parameters (if any) are separated by space or comma.

#### AUTOMATIC

<file name 1>

<file name 2>

<file name n>

The automatically scanned library file names are defined. Separate the file names by CR and end the definition of the AUTOMATIC buffer by a period. The files in the AUTOMATIC buffer are scanned for undefined entries when the EXIT or LIBRARY commands are issued.

Default: N50-LIB.

Default type: BRF5.

#### BREAK-CONDITIONS <BP><BQ><CONDITIONS>

Sets the program limit registers, BP and BQ, and the break conditions to be used when the program is run. (The break conditions may also be changed in the NORD-50 Monitor.) Conditions may be:

A	stop on any reference in BP $\geq$ BQ
D	stop on data reference BP $\geq$ BQ
S	stop on store reference in BP $\geq$ BQ
F	stop on fetch reference in BP $\geq$ BQ
O	stop on overflow
U	stop on underflow
P	stop on parity error in memory

Default is: SP.

#### CC <comment>

Comment line.

#### DEBUG

The old contents of the memory-image file should be preserved for examination.

#### DEFINE <symbol> <octal value>

The symbol will be entered into the loader table. Its value will be equal to the octal number specified.

DEFINE-COMMON <symbol> <octal value> <octal size>

The common label will be entered into the loader table.

DEFINE-FIO <file name>

Define the name for the NORD-50 Formatted Input/Output and Run-time System (FIO). The file is automatically loaded from load address 0 when the first LOAD or LIBRARY command is issued.

Default: N50-FIO.

Default type: BRF5.

ENTRIES-DEFINED [<file name>]

All symbols (defined) present in the loader table will be printed on the terminal. In addition, the current location and the upper bound are displayed in the following format:

FREE: <current location> — <upper bound>

Default file name is the terminal.

ENTRIES-UNDEFINED [<file name>]

This command is similar to ENTRIES-DEFINED. However, only undefined symbols are printed.

Default file name is the terminal.

EXIT

The control is left to the operating system and the loading is terminated. The program and status information is saved on the memory image file.

FORMAT <format>

Set out formats to be used in the PRINT and LOOK-AT commands. Formats may be O, D, F, S, I, A, B, T or any combination of these characters.

O	octal
D	decimal
F	floating point (64 bits)
S	floating point (32 bits)
I	instructions (disassembled)
A	ASCII (4 ASCII characters per word)
B	Binary
T	NORD-10 format (2 x 16 bits octal)

HELP

List the available loader commands on the terminal.

KILL <symbol>

If present, the symbol will be removed from the loader table.

LIBRARY

Scan the files in the AUTOMATIC file name buffer for undefined entries. If the LIBRARY command is issued before any LOAD command the file defined by the DEFINE-FIO command is loaded.

LIST-MEMORY <output file>

List the contents of the NORD-50 Loader Memory Configuration Table as set up by the SET-MEMORY command.

LOAD <file name> [<file name> . . . ]

The file(s) specified will be loaded until end-of-file is encountered. Default file type is BR5.

When the first LOAD command is issued also the file defined in the DEFINE-FIO command is loaded.

LOCAL-COMMON

Common areas defined after this command is issued will be local to the overlay.

[LOOK-AT] <address or register>

Enter the "look-at mode" to examine or change NORD-50 Memory Image File contents. The command may be omitted and just the address or register to be examined given.

OVERLAY-ENTRY <entry name> [<entry name> . . . ]

Specifies the subprograms on the next overlay. These units may be called from the root segment or from the actual overlay itself.

OVERLAY-FILE <file no./name>

Specifies the overlay-file and the loader is put into overlay-mode.

PRINT <low address> <high address> [<file>]

The contents of the address or register interval are printed on the specified file. Output formats are specified by the last FORMAT command.

REFERENCE <symbol> <octal address>

If the symbol is not defined, a reference to the symbol in the address specified will be linked to the undefined symbols list. When the symbol is defined, the value of the symbol will be added to the previous contents of the address.

RENAME <old symbol name> <new symbol name>

The old symbol name in the loader table will be replaced by the new one. Defined/undefined state and value are left unchanged.

RESET

The loader variables and tables are initialized (symbols removed).

RESET-LIST-MODE

No list of the BRF input will be given.

RT-COMMON-DEFINE

All RT-COMMON labels defined by the RT-LOADER (refer also to the manual "SINTRAN III RT Loader" — ND-60.051) will be defined as common labels in the loader Symbol Table. Labels defined to odd addresses in the NORD-10 SINTRAN III RT-COMMON will be listed and *ignored*. The addresses are transformed from NORD-10 addresses according to the information given in the NORD-50 LOADER Memory Configuration Table.

The RT-COMMON-DEFINE command should be used before the program units referring to the RT-COMMON area are loaded.

When this command has been used, information about the size and address of RT-COMMON is put on the Memory Image File. This information is checked by the NORD-50 MONITOR before the program is allowed to run on the NORD-50.

RT-PROGRAM-DEFINE

Define RT-PROGRAM names. The command should be used after the program units referring to the RT program names are loaded.

Only RT program names which are undefined symbols to the NORD-50 LOADER are defined by the command.

SEGMENT-COMMON-DEFINE <segment no.>

All Segment-Common Labels defined by the RT-LOADER on the segment specified will be defined as common labels in the Loader Symbol Table. Labels defined to odd addresses in the NORD-10 SINTRAN III Segment-Common will be listed and *ignored*. The addresses are transformed from NORD-50 addresses according to the information given in the NORD-50 LOADER Memory Configuration Table.

The SEGMENT-COMMON-DEFINE command should be used before the program units referring to the Segment-Common are loaded.

When this command has been used, information about the contents of the Loader Memory Configuration Table (Chapter 2) is put on the Memory Image File. The NORD-50 MONITOR will then run the program with the memory configuration defined by the contents of the Memory Image File ignoring the default memory configuration for the actual installation as defined by the SET-MEMORY command in the NORD-50 MONITOR.

**SET-LIST-MODE <file>**

A listing of the BRF input will be given on the file specified when program modules are being loaded.

**SET-LOAD-ADDRESS <address>**

Subsequent loading will start from the address specified.

**SET-MEMORY <segment no.> <size> <type>**

Define the NORD-50 Loader Memory Configuration Table (Chapter 2).

**STATUS**

Status information for the program is displayed.

**UPPER-LIMIT <octal address>**

The load area upper limit is set to the value specified.

**VALUE <symbol>**

If defined, the value of the symbol specified will be printed on the terminal.

*Example:*

```
VAL EDMUN
000777
```

**X-LOAD <file name> [<file name> ... ]**

Exclusive load. Library sequences headed with defined symbols are skipped while all other units on the file(s) specified will be loaded until end-of-file is encountered. Default file type is BRF5. This command is somewhat special and is used for system generation.

*Example:*

```
X-LOAD LIBRA
```

**QUIT**

Return to SINTRAN III without ending the loading.



## APPENDIX B

### THE LOADER ERROR MESSAGES

#### COMMON BLOCK EXPANDED

The length of an already defined common block is declared larger in a subsequently loaded program.

#### \*WARNING\* DOUBLY DEFINED

The symbol being defined (either by loading a file or by the DEFINE command) has already been assigned a value.

#### \*WARNING\* MORE THAN ONE MAIN PROGRAM

The first entered main program unit is started by the RUN command.

#### INSUFFICIENT PROGRAM

Error diagnostics have occurred during the compilation process.

#### ILLEGAL BRF CONTROL NO.

Non-interpretive information has appeared on the BRF file due to hardware or software errors.

#### AT UPPER LIMIT

The current load address has reached the absolute upper limit or the beginning of the common area.

#### LOADER-TABLE OVERFLOW

The loader symbol table is filled.

#### MISSING COMMON LABEL

#### TRIED TO LOAD OUTSIDE DEFINED MEMORY

#### UNDEFINED SYMBOLS

If references still not defined at the EXIT command.

#### NO MAIN ENTRY

If a main program unit is missing at the EXIT command.

#### THIS SHOULD NOT HAPPEN

This is system error and should be reported to Norsk Data A.S.



NORSK DATA A.S.

Lørenveien 57 - Postboks 163, Økern

OSLO 1

## COMMENT AND EVALUATION SHEET

NORD-50 LOADER — User's Guide  
AUGUST 1978

Publication No. ND-60.083.02

In order for this manual to develop to the point where it best suits your needs, we must have your comments, corrections, suggestions for additions, etc. Please write down your comments on this pre-addressed form and post it. Please be specific wherever possible.

FROM

---

---

---

**– we make bits for the future**

NORSK DATA A.S LØRENVEIEN 57 OSLO 5 NORWAY PHONE: 21 73 71 TELEX: 18284